HOW TO USE

THE 2-FINGER ADAPTIVE GRIPPER

WITH UNIVERSAL ROBOTS CONTROLLER

REVISION HISTORY

Rev	Date	Author	Comment
V001	08-05-2013	J. Mangel & N. Lauzier	Initial creation
V002	28-06-2013	J. Mangel	Few precisions concerning the installation
V003	31-07-2013	C. Bernier	Needed version of Polyscope revised (use version 1.7 or later)

REFERENCE DOCUMENTS

Document	Location

TABLE OF CONTENT

1. SUMMARY

This white paper describes the operating mode to use for the 2-Finger Adaptive Robot Gripper from Robotiq with the controller from Universal Robots (named Polyscope).

The 2-Finger Adaptive Robot Gripper uses the Modbus TCP/IP communication protocol. Polyscope is also compatible with this protocol. Robotiq provides the configuration file for the inputs and outputs (I/Os). Therefore, the user has the possibility of applying subroutines which have been previously developed; such as open/close, go to a requested position, and apply speed and force parameters. These routines will allow the user to be able to manage object detection when the Gripper opens and closes. A script file will allow the user to know the position of the fingers.

Two example programs are provided along with the description of their operations.

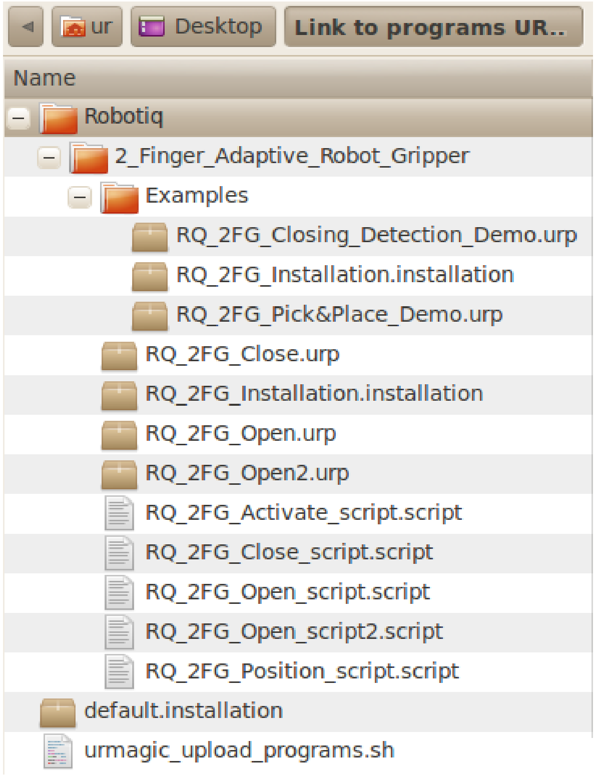
2. AVAILABLE FILES AND METHODOLOGY

WARNING BEFORE STARTING

- Please disconnect the gripper's USB to use it with the Universal Robots and reboot the Universal Robots' controller
- Please use version 1.7 (or later) of the software Polyscope.

Start by unzipping the file **RQ_2FG_URobots.zip**.

Here is an overview of the files included:



- Robotiq
  - 2_Finger_Adaptive_Robot_Gripper
    - Examples
      - RQ_2FG_Closing_Detection_Demo.urp
      - RQ_2FG_Installation.installation
      - RQ_2FG_Pick&Place_Demo.urp
    - RQ_2FG_Close.urp
    - RQ_2FG_Installation.installation
    - RQ_2FG_Open.urp
    - RQ_2FG_Open2.urp
    - RQ_2FG_Activate_script.script
    - RQ_2FG_Close_script.script
    - RQ_2FG_Open_script.script
    - RQ_2FG_Open_script2.script
    - RQ_2FG_Position_script.script
- default.installation
- urmagic_upload_programs.sh

- Installation file of Modbus I/Os
- 3 programs: Open the Gripper (2 versions of the same operation, but using two different parameters) and Close the Gripper: **RQ_2FG_Open, RQ_2FG_Open2** and **RQ_2FG_Close**.
- 1 script file: **RQ_2FG_Activate_script.script** for Gripper initialization (to be used before the 3 previous programs).
- 1 script file: **RQ_2FG_Position_script.script**. to be used in a thread to know the finger position in real time.
- 2 example programs: **RQ_2FG_Closing_Detection_Demo** and **RQ_2FG_Pick&Place_Demo**.

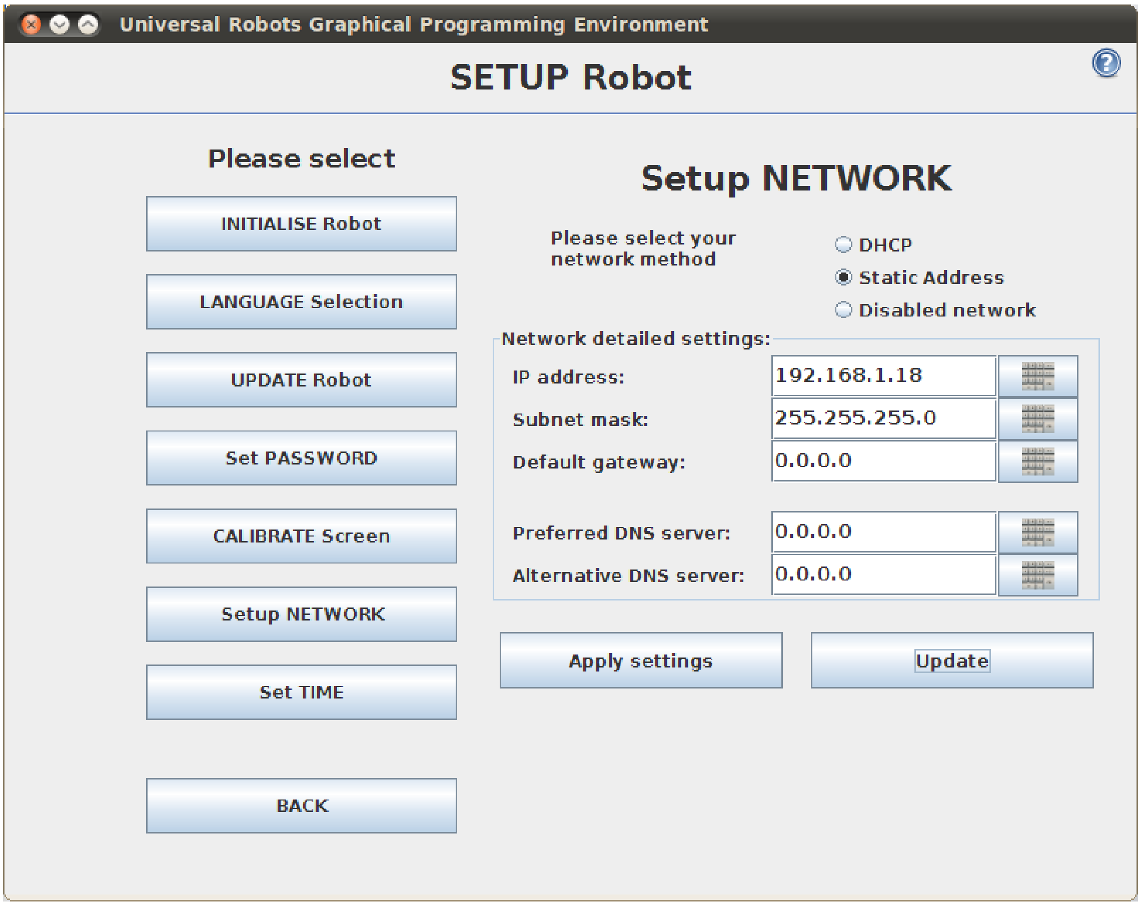
You need to copy files in the Universal Robots controller. To achieve this, Universal Robots provide a file named “urmagic_upload_programs.sh”. All program, sript and installation files will be copied from a USB-stick onto the controller, to the programs folder, subfolders included.

The following instructions explain how to upload files from USB to controller:

1. Copy ''umagic_upload_programs.sh'' file to the root of a USB-stick.
2. Copy program and installation files to the USB-stick.
3. Insert the USB stick in the USB-slot in teach pendant.
4. A red "! USB !" warning sign will appear indicating the upload is in progress.
5. Await a green "<- USB" sign appearing, indicating the upload is completed, it is now safe to remove the USB-stick.

3. HOW TO CONFIGURE THE PARAMETERS OF THE UNIVERSAL ROBOTS NETWORK WITH THE MODBUS I/Os OF THE ROBOTIQ GRIPPER

When starting up Polyscope, click on '*SETUP Robot*' and '*Setup NETWORK*'. Fill in the parameters of the network to which the Gripper is connected and click on '*Apply settings*‘.



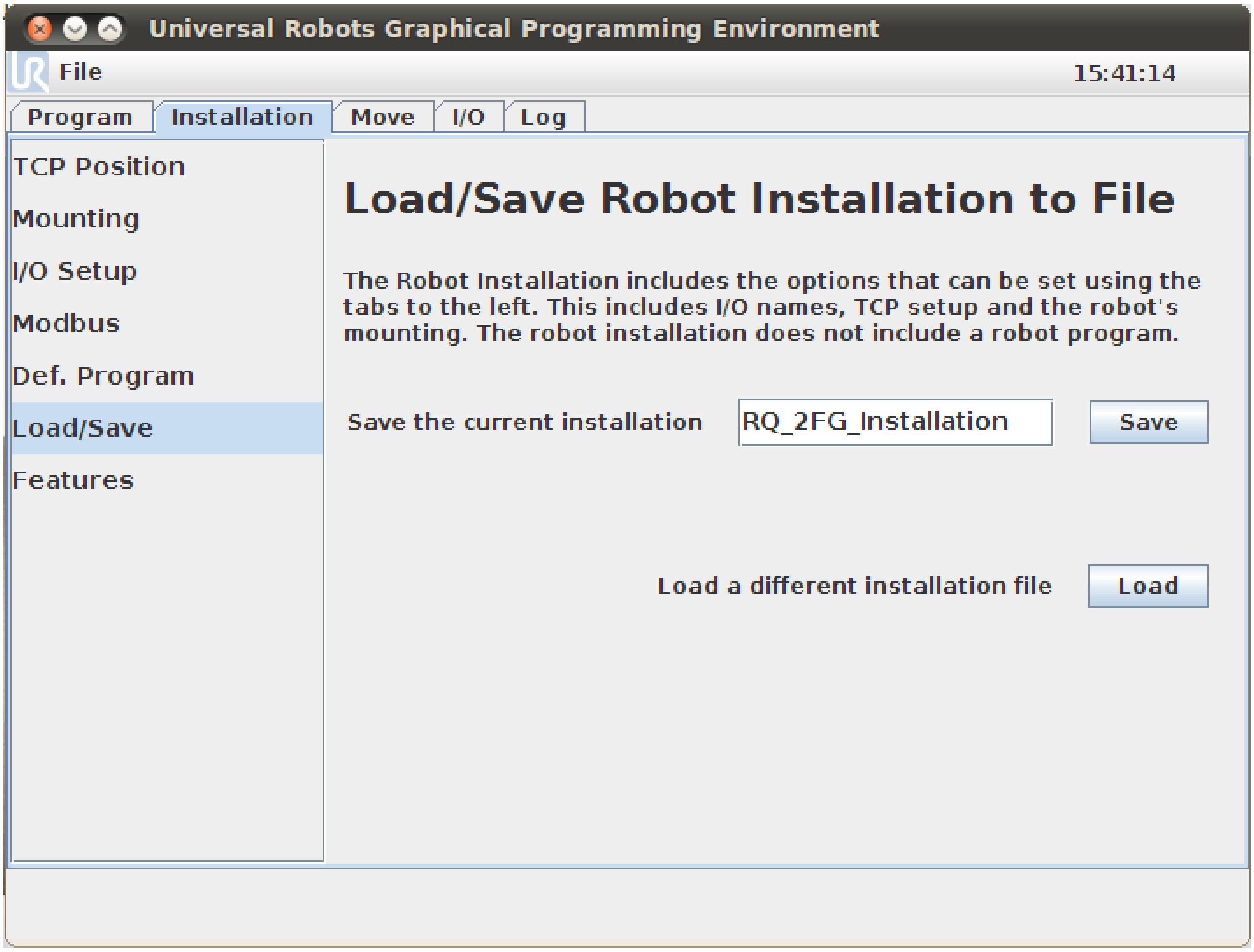
Please select your network method

○ DHCP
● Static Address
○ Disabled network

Network detailed settings:

IP address: 192.168.1.18
Subnet mask: 255.255.255.0
Default gateway: 0.0.0.0
Preferred DNS server: 0.0.0.0
Alternative DNS server: 0.0.0.0

When creating a new program, you must load the installation file with the Modbus parameters for the I/Os of the Gripper. To achieve this, go to the home page, click on '*PROGRAM Robot*' and click on '*Load Program*' in the '*Program*' tab, choose the file **RQ_2FG_Installation.installation** in the folder */Robotiq/2_Finger_Adaptive_Robot_Gripper.*



Load/Save Robot Installation to File

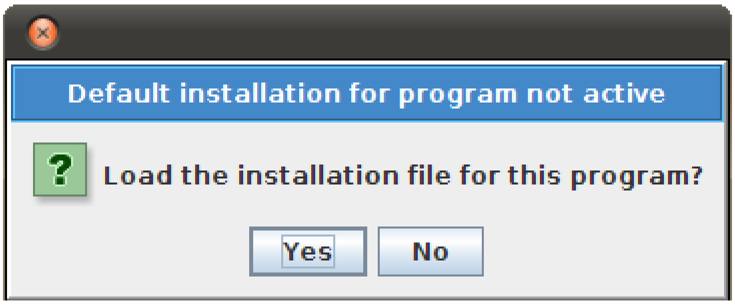
The Robot Installation includes the options that can be set using the tabs to the left. This includes I/O names, TCP setup and the robot's mounting. The robot installation does not include a robot program.

Save the current installation: RQ_2FG_Installation

Load a different installation file

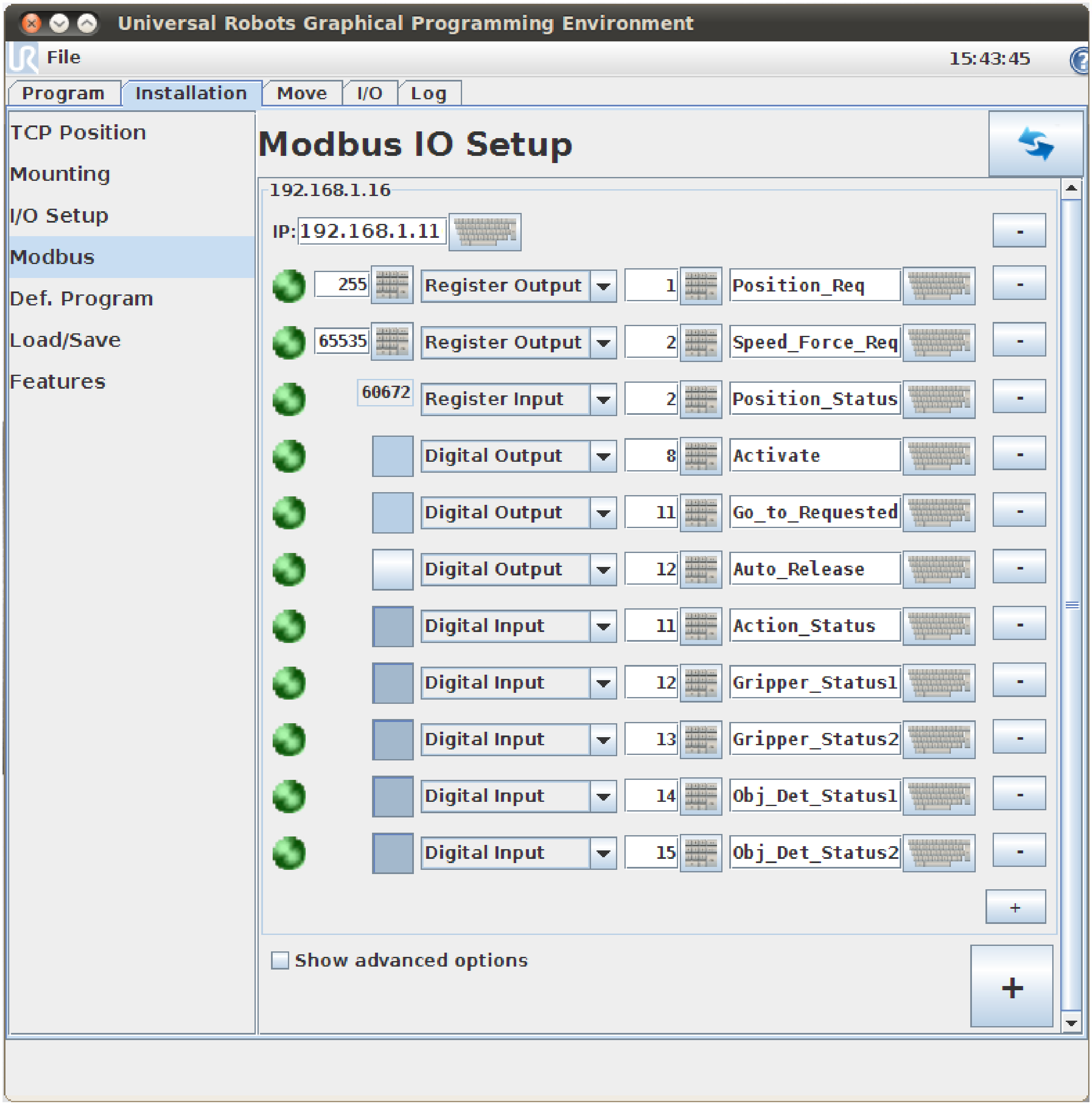
Then go to the '*Modbus*' section of the '*Installation*' tab, the digital I/Os (register bits) and the I/Os registers are configured automatically.

<u>NOTE</u> : When editing a program, the appropriate installation file has not to be reloaded manually. A pop-up will inform you if you want to use the same settings. The previous selected installation file will be loaded by clicking the "Yes" button.



Default installation for program not active

Load the installation file for this program?



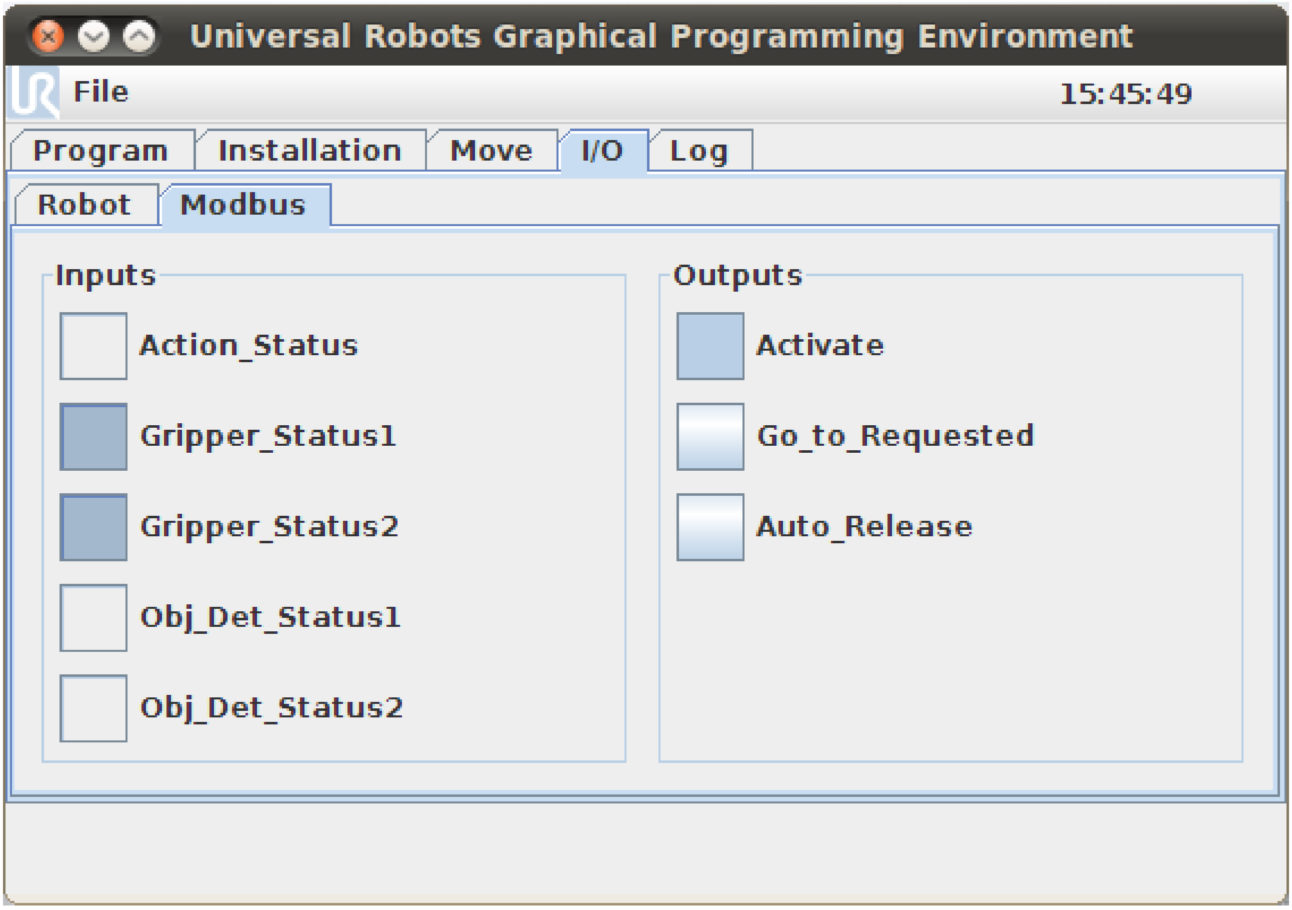
192.168.1.16

IP: 192.168.1.11

255	Register Output	1	Position_Req
65535	Register Output	2	Speed_Force_Req
60672	Register Input	2	Position_Status
	Digital Output	8	Activate
	Digital Output	11	Go_to_Requested
	Digital Output	12	Auto_Release
	Digital Input	11	Action_Status
	Digital Input	12	Gripper_Status1
	Digital Input	13	Gripper_Status2
	Digital Input	14	Obj_Det_Status1
	Digital Input	15	Obj_Det_Status2

CAUTION: You need to set the IP address of the Robotiq Gripper on this page.

You can manually change the refresh frequency of Modbus I/Os by clicking on the '*Show advanced options*' checkbox. To know the digital I/Os' status, and to set the digital outputs, go to the '*Modbus*' tab of the '*I/O*' tab.



Inputs

- ☐ Action_Status
- ☑ Gripper_Status1
- ☑ Gripper_Status2
- ☐ Obj_Det_Status1
- ☐ Obj_Det_Status2

Outputs

- ☑ Activate
- ☐ Go_to_Requested
- ☐ Auto_Release

Digital Outputs:

Activate:	bit n°0 (rACT) from the ACTION REQUEST register
Go_to_Requested:	bit n°3 (rGTO) from the ACTION REQUEST register
Auto_Release:	bit n°4 (rATR) from the ACTION REQUEST register

Digital Inputs:

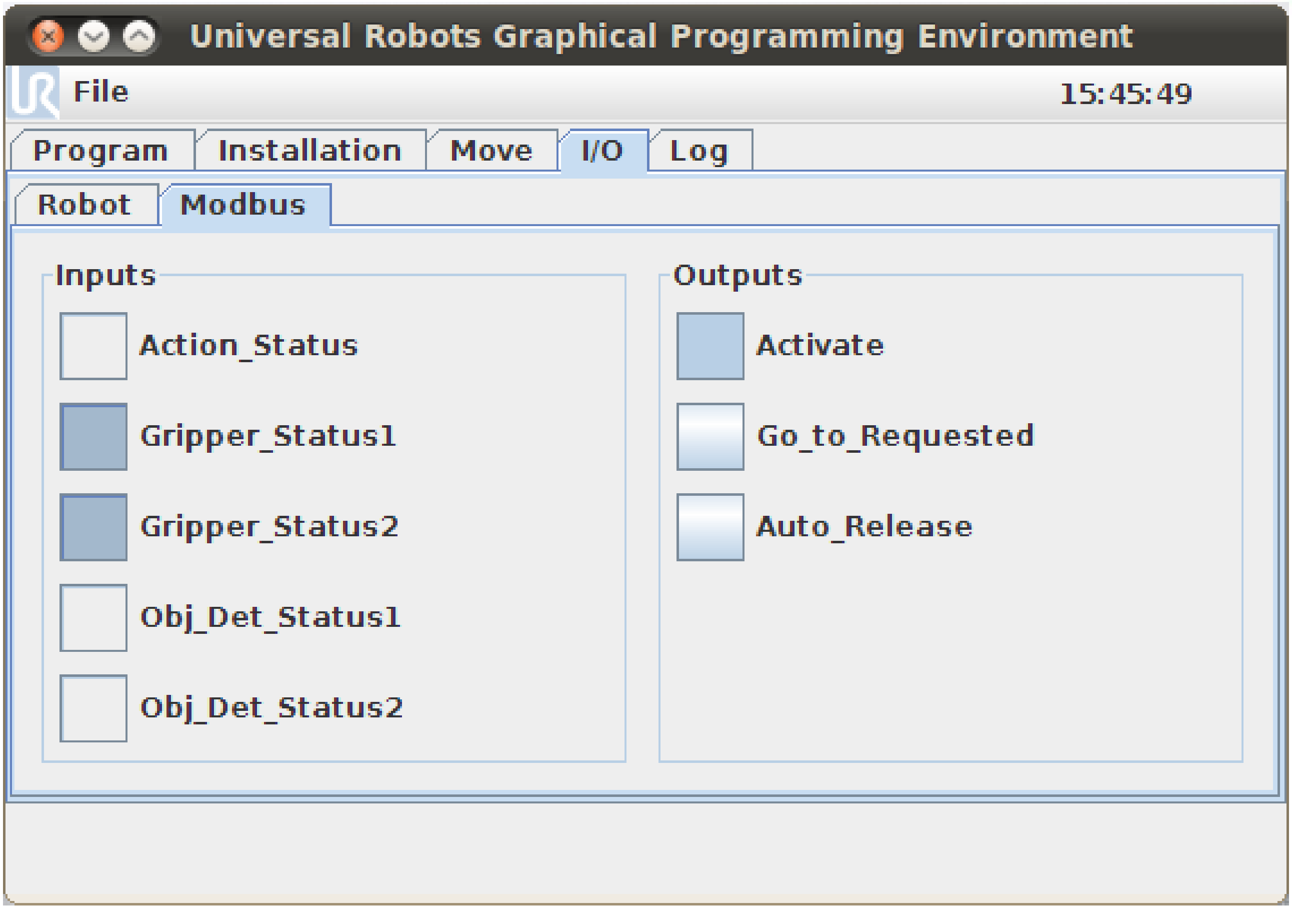
Action_Status:	bit n°3 (gGTO) from the GRIPPER STATUS register
Gripper_StatusX:	bits n°4 & 5 (gSTA) from the GRIPPER STATUS register
Obj_Det_StatusX:	bits n°6 & 7 (gOBJ) from the GRIPPER STATUS register

4. USE OF OPEN AND CLOSE PROGRAMS AS SUBROUTINES

GRIPPER ACTIVATION

In order to be able to use the OPEN and CLOSE programs; each time you start-up the main program you must:

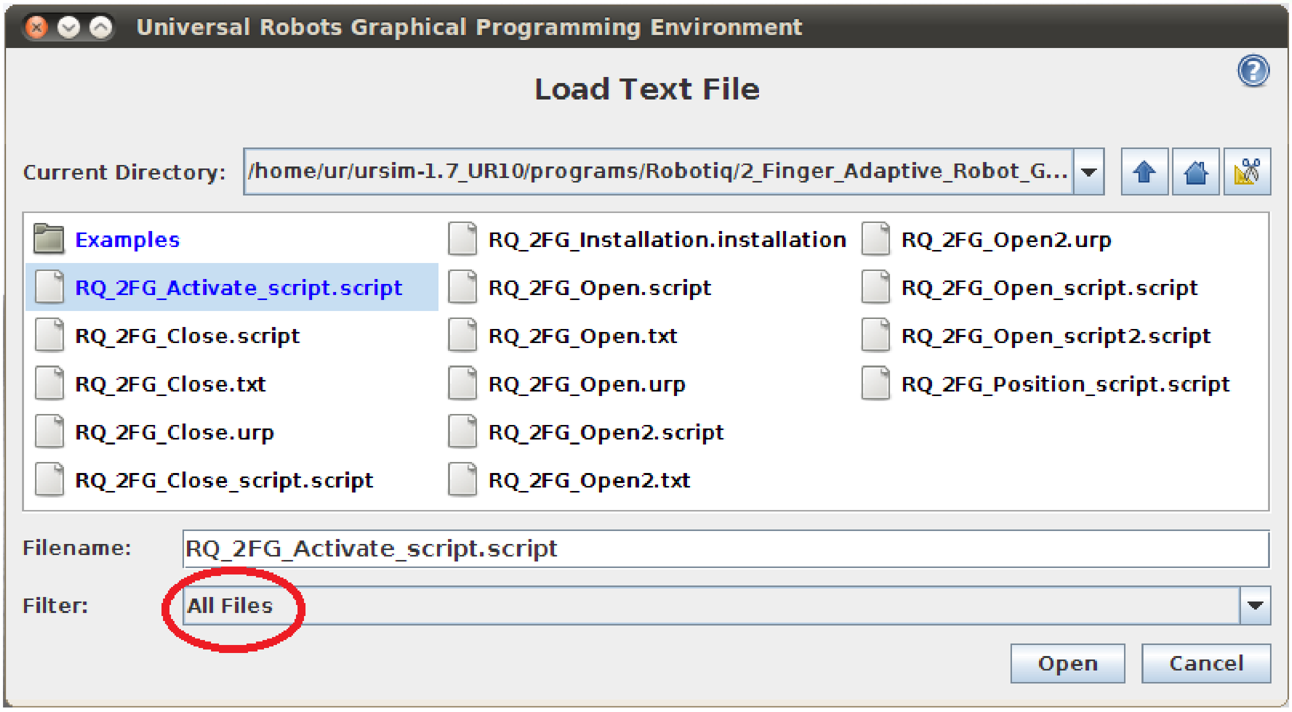
- **Either** manually activate the Robotiq Gripper (*Activate = True*), reset the finger's movement (*Go_to_Requested = False*) by setting the digital outputs as such:



CAUTION: If this option is selected, please wait until the end of the Robotiq Gripper activation sequence (opening-closing-opening) before starting the program.

- **Or** load the file **RQ_2FG_Activate_script.script**, in the folder */Robotiq/2_Finger_Adaptive_Robot_Gripper,* at the beginning of the main program.

To access this file in the previous directory, use the filter '*All Files*', on the bottom of the '*Load Text File*' page:


Universal Robots Graphical Programming Environment
Load Text File
Examples
RQ_2FG_Activate_script.script
RQ_2FG_Close.script
RQ_2FG_Close.txt
RQ_2FG_Close.urp
RQ_2FG_Close_script.script
RQ_2FG_Installation.installation
RQ_2FG_Open.script
RQ_2FG_Open.txt
RQ_2FG_Open.urp
RQ_2FG_Open2.script
RQ_2FG_Open2.txt
RQ_2FG_Open2.urp
RQ_2FG_Open_script.script
RQ_2FG_Open_script2.script
RQ_2FG_Position_script.script
Open
Cancel

Current Directory: /home/ur/ursim-1.7_UR10/programs/Robotiq/2_Finger_Adaptive_Robot_G...

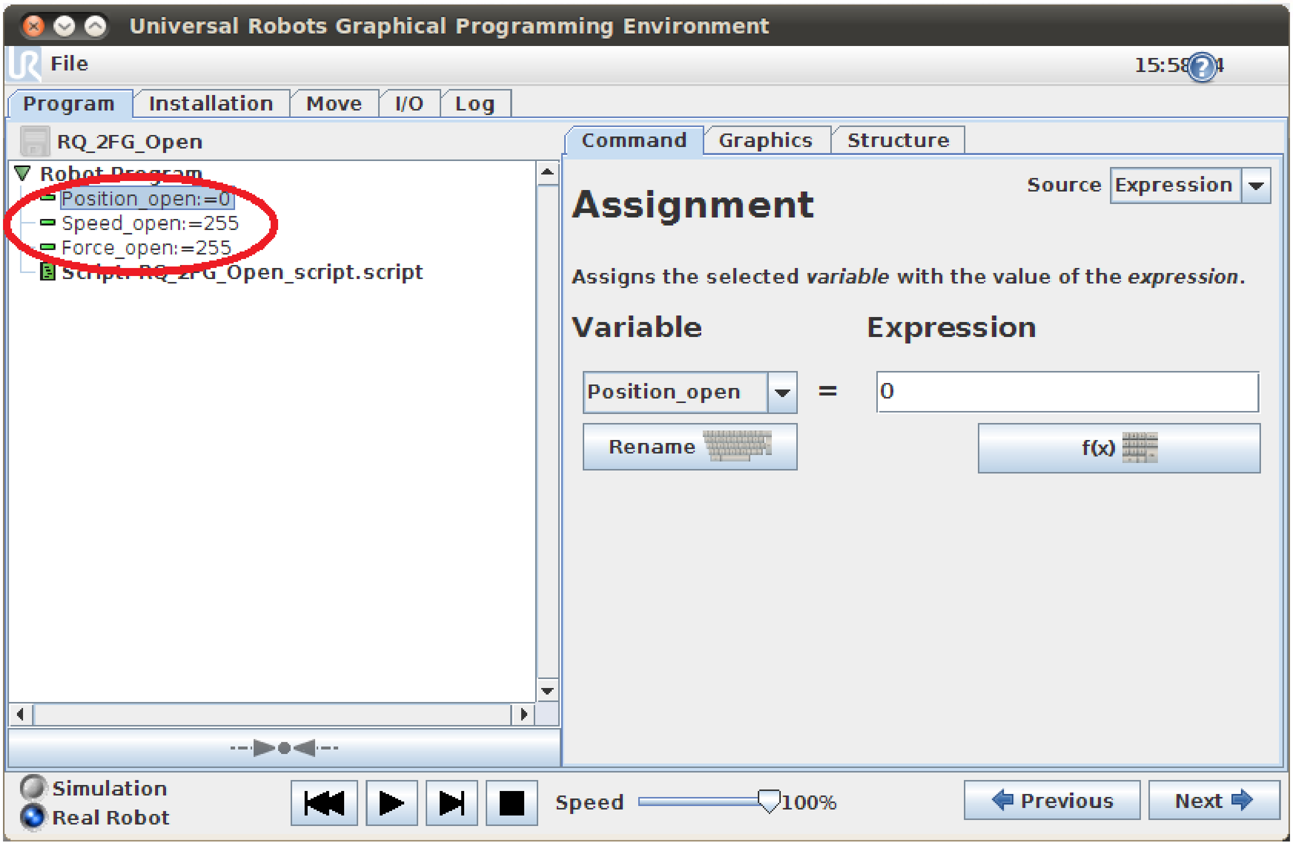
Filename: RQ_2FG_Activate_script.script

Filter: All Files

This script will launch the activation of the Gripper (*Activate = True*), wait until the end of its activation sequence before resetting the finger's movement (*Go_to_Requested = False*).
This script can only be used once in the main program, for example, at the beginning of the program, but before the infinite loop (see the sample programs below).

OPEN PROGRAM

This program file is named **RQ_2FG_Open** and is located in the folder *.../Robotiq/2_Finger_Adaptive_Robot_Gripper.*
This program allows the developer to manually change the motion settings of the Gripper by changing the following variables: *Position_open*, *Speed_open* and *Force_open*, which are accessible on the left in the '*Program*' tab.



RQ_2FG_Open
- Robot Program
  - Position_open:=0
  - Speed_open:=255
  - Force_open:=255
  - Script: RQ_2FG_Open_script.script

Assignment

Source: Expression

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

Variable: Position_open = **Expression**: 0

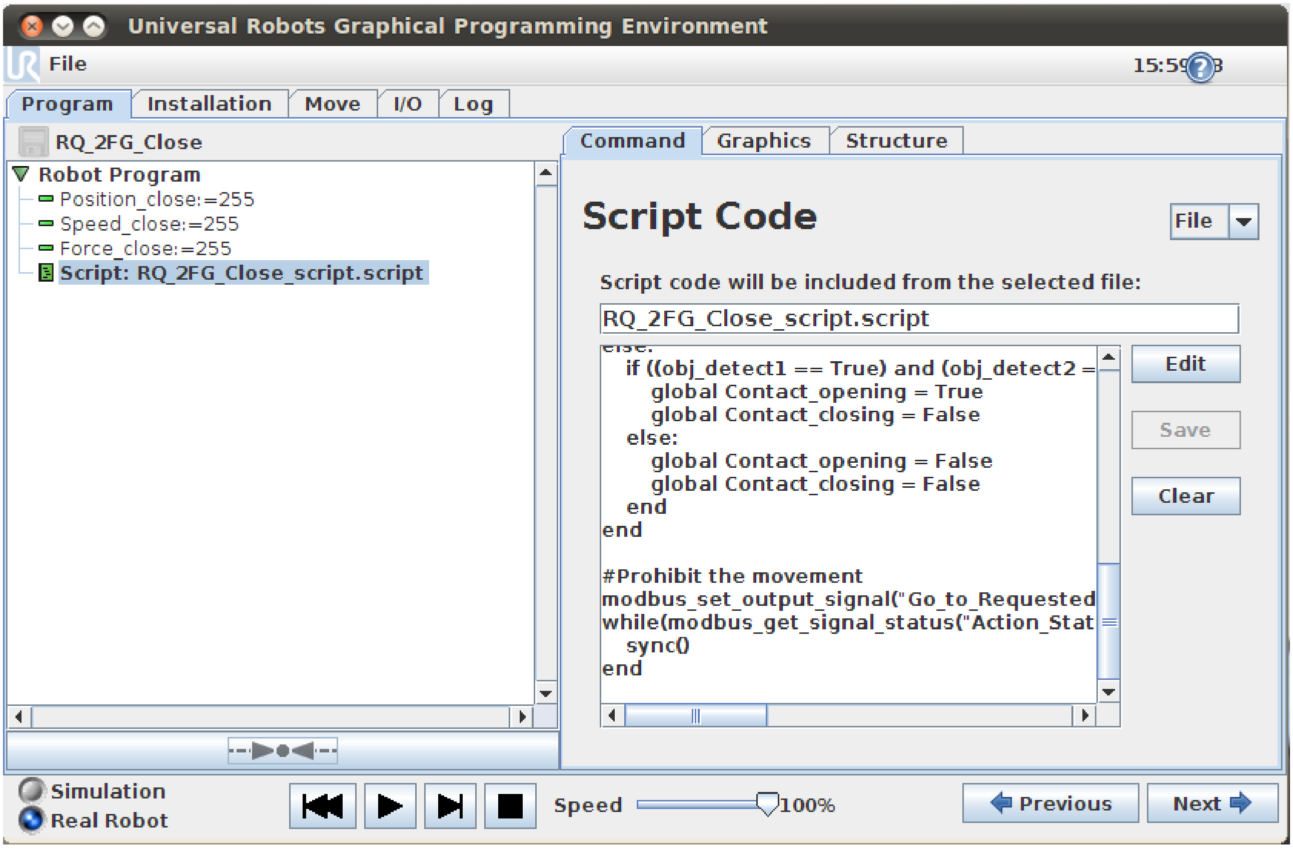
CAUTION : These values must be set between 0 and 255

- *Position_open*: 0 equals the maximum opening value for the fingers and 255 is the Gripper closing value ;
- *Speed_open/Force_open*: 0 setting the minimum speed/force and 255 the maximum ;

CLOSE PROGRAM

This program file is named **RQ_2FG_Close** and is located in the folder *…/Robotiq/2_Finger_Adaptive_Robot_Gripper.*

This program allows the developer to manually change the motion settings of the Gripper by changing the following variables: *Position_close, Speed_close and Force_close,* which are accessible on the left in the '*Program*' tab.



RQ_2FG_Close

```
Robot Program
  Position_close:=255
  Speed_close:=255
  Force_close:=255
  Script: RQ_2FG_Close_script.script
```

Script code will be included from the selected file: RQ_2FG_Close_script.script

```
   if ((obj_detect1 == True) and (obj_detect2 =
      global Contact_opening = True
      global Contact_closing = False
   else:
      global Contact_opening = False
      global Contact_closing = False
   end
end

#Prohibit the movement
modbus_set_output_signal("Go_to_Requested
while(modbus_get_signal_status("Action_Stat
   sync()
end
```

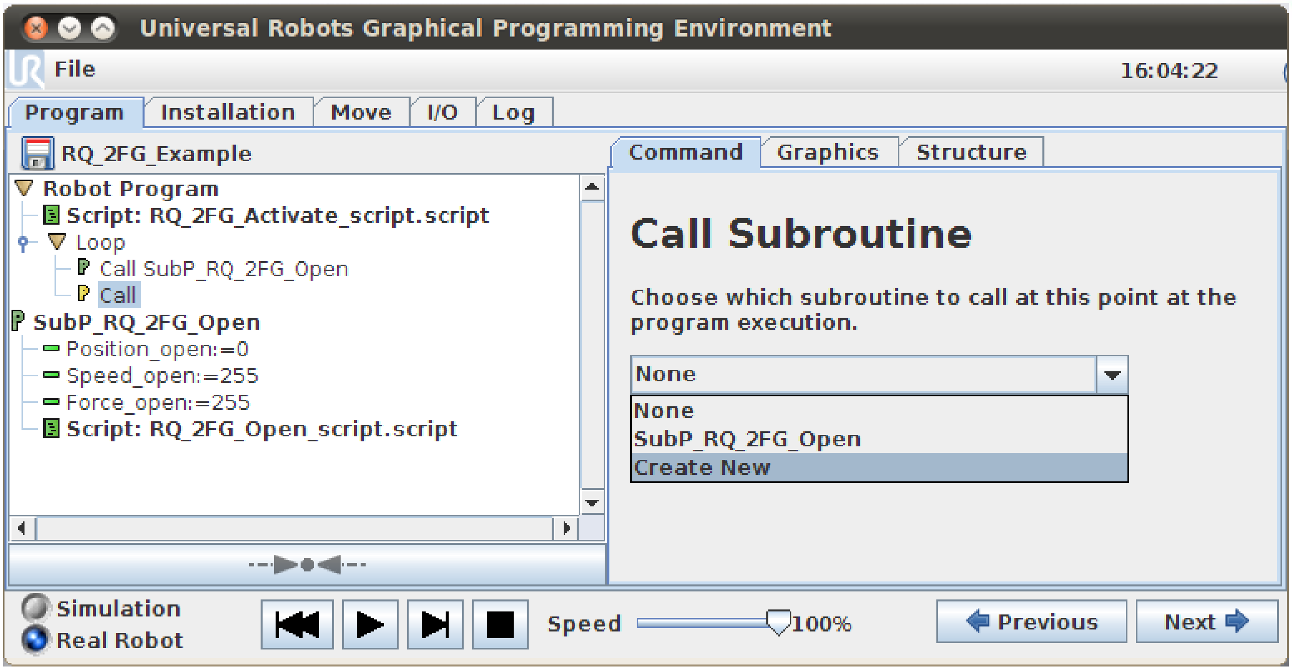
CAUTION: These values must be set between 0 and 255

- *Position_close*: 0 is the maximum opening of the fingers and 255 is the Gripper closing value ;
- *Speed_close/Force_close*: 0 setting the minimum speed/force and 255 the maximum;

USING FUNCTIONS ACTIVATE, OPEN AND CLOSE IN A PROGRAM

Polyscope allows you to use previously developed programs in the development of a new program.

Here is an example program that shows you how the OPEN and CLOSE functions work as subroutines in a program:


Universal Robots Graphical Programming Environment

16:04:22

RQ_2FG_Example

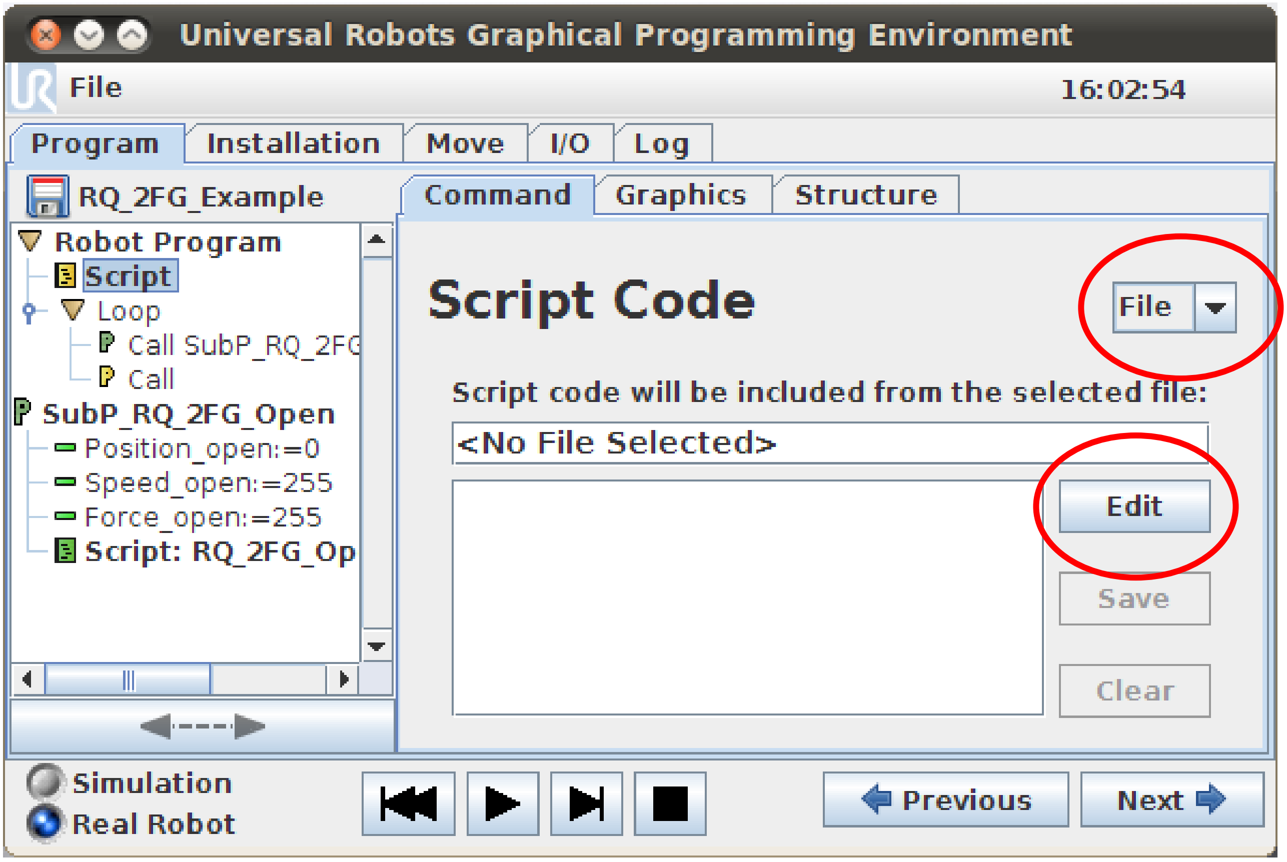
- Robot Program
  - Script: RQ_2FG_Activate_script.script
  - Loop
    - Call SubP_RQ_2FG_Open
    - Call
- SubP_RQ_2FG_Open
  - Position_open:=0
  - Speed_open:=255
  - Force_open:=255
  - Script: RQ_2FG_Open_script.script

Call Subroutine

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

None
- None
- SubP_RQ_2FG_Open
- Create New

Speed 100%


Universal Robots Graphical Programming Environment

16:02:54

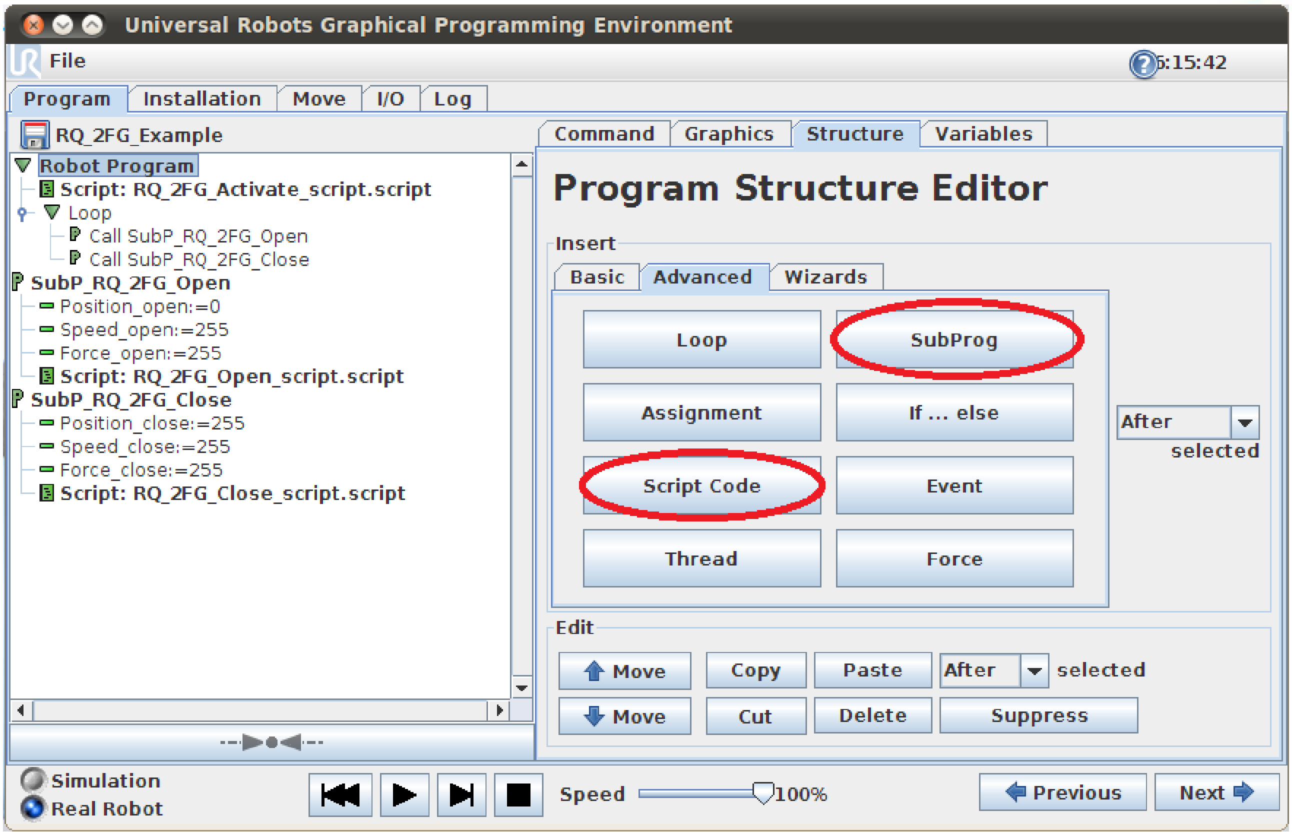
RQ_2FG_Example

- Robot Program
  - Script
  - Loop
    - Call SubP_RQ_2FG
    - Call
- SubP_RQ_2FG_Open
  - Position_open:=0
  - Speed_open:=255
  - Force_open:=255
  - Script: RQ_2FG_Op

Script Code

Script code will be included from the selected file:

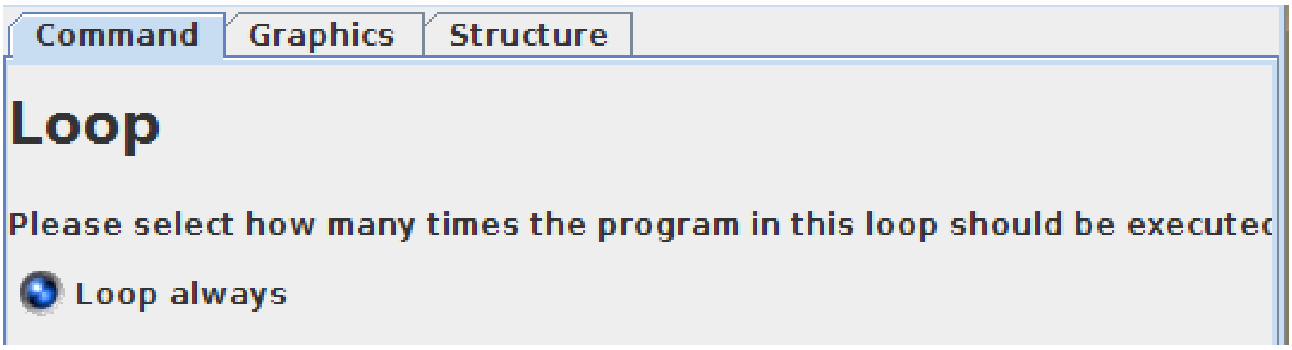
<No File Selected>



```
RQ_2FG_Example
Robot Program
  Script: RQ_2FG_Activate_script.script
  Loop
    Call SubP_RQ_2FG_Open
    Call SubP_RQ_2FG_Close
SubP_RQ_2FG_Open
  Position_open:=0
  Speed_open:=255
  Force_open:=255
  Script: RQ_2FG_Open_script.script
SubP_RQ_2FG_Close
  Position_close:=255
  Speed_close:=255
  Force_close:=255
  Script: RQ_2FG_Close_script.script
```

DESCRIPTIONS:

- At the inception of the program, use the '*Script Code*' and 'SubProg' structures in the '*Advanced*' tab. Then, you have to select your desired files.
- At the execution of the program, the **RQ_2FG_Activate_script.script** script is run and the Gripper will be activated (if it is not already).
- Once the activation is complete, the program will run an infinite loop, set like this:



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

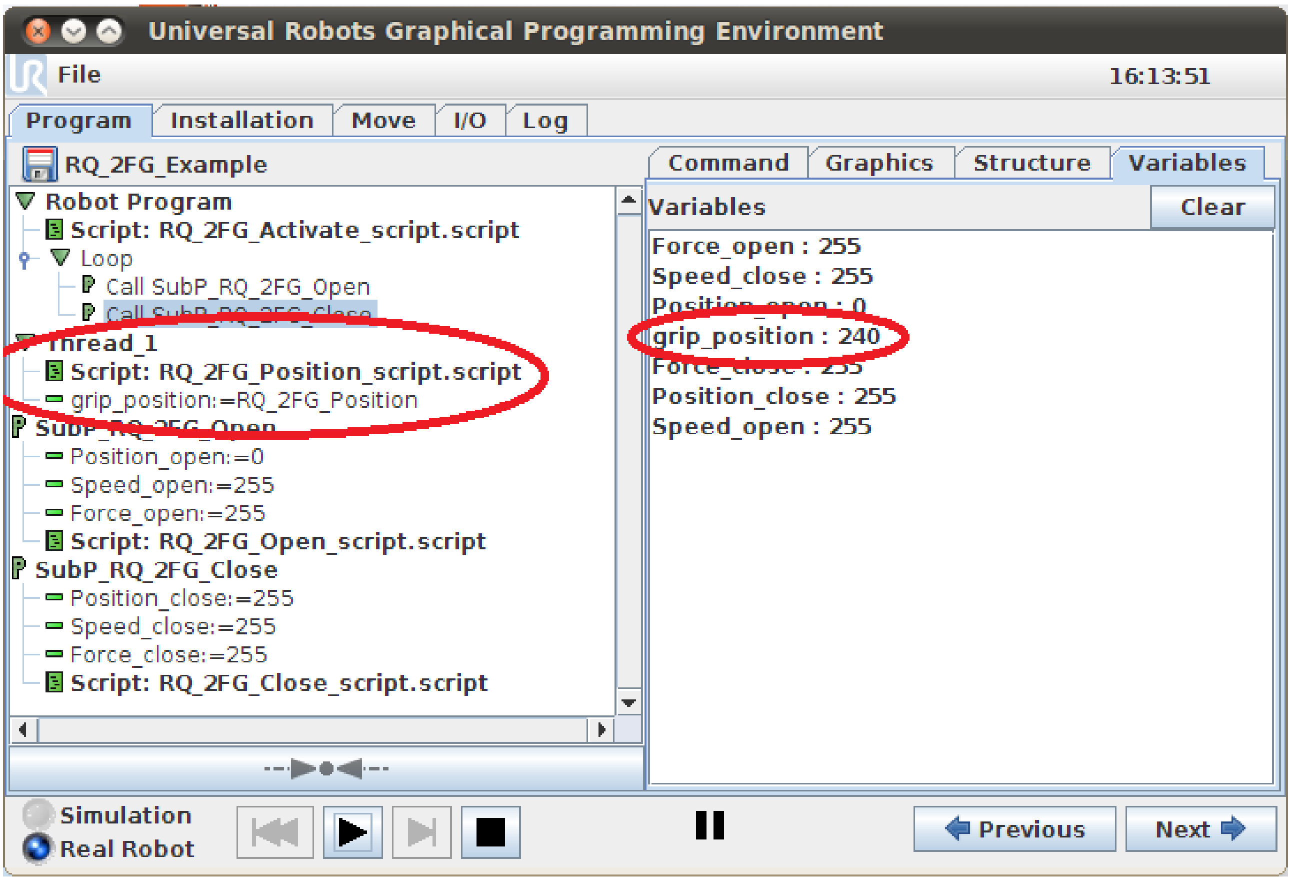
● Loop always

Subroutines **SubP_RQ_2FG_Open** and **SubP_RQ_2FG_Close** will be run one after the other. Then, the Gripper will open and close continuously, regardless of whether an object is detected or not.

5. HOW TO KNOW THE POSITION OF THE ROBOTIQ GRIPPER'S FINGERS

The *Robotiq 2-Finger Adaptive Robot Gripper* allows you to know the position of the Gripper's fingers in real time.

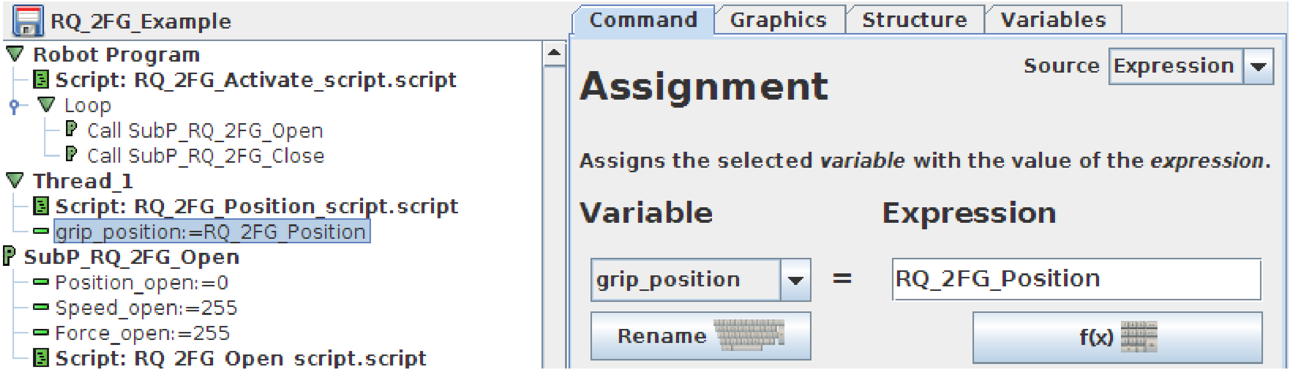
To do this, you have to use a *thread* (a program that runs in parallel with the main program). This structure is available in the '*Advanced*' tab of the '*Structure*' tab.



```
Robot Program
  Script: RQ_2FG_Activate_script.script
  Loop
    Call SubP_RQ_2FG_Open
    Call SubP_RQ_2FG_Close
Thread_1
  Script: RQ_2FG_Position_script.script
  grip_position:=RQ_2FG_Position
SubP_RQ_2FG_Open
  Position_open:=0
  Speed_open:=255
  Force_open:=255
  Script: RQ_2FG_Open_script.script
SubP_RQ_2FG_Close
  Position_close:=255
  Speed_close:=255
  Force_close:=255
  Script: RQ_2FG_Close_script.script
```

```
Variables
Force_open : 255
Speed_close : 255
Position_open : 0
grip_position : 240
Force_close : 255
Position_close : 255
Speed_open : 255
```

The Gripper's finger positions can be displayed in the '*Variables*' tab. The relative value will be assigned to the selected variable, using the global variable **RQ_2FG_Position** (variable '*grip_position*' is used in the example):



```
Robot Program
  Script: RQ_2FG_Activate_script.script
  Loop
    Call SubP_RQ_2FG_Open
    Call SubP_RQ_2FG_Close
Thread_1
  Script: RQ_2FG_Position_script.script
  grip_position:=RQ_2FG_Position
SubP_RQ_2FG_Open
  Position_open:=0
  Speed_open:=255
  Force_open:=255
  Script: RQ_2FG_Open_script.script
```

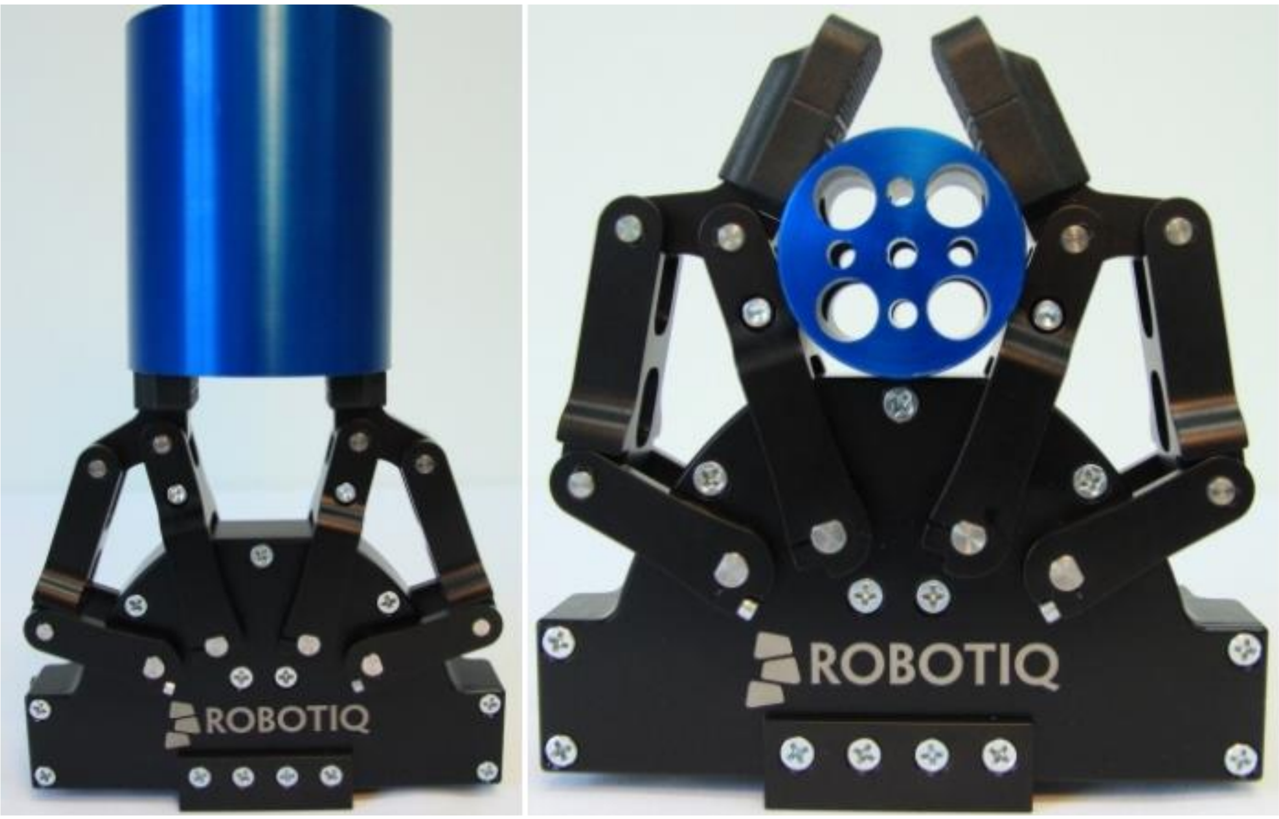
Assigns the selected *variable* with the value of the *expression*.

grip_position = RQ_2FG_Position

Note: The values for extreme positions on either end – opened or closed (0 or 255) – will never be reached because there are internal mechanical margins configured to avoid this (stop components). But, this does not affect the detection of objects.

6. OBJECT DETECTION DURING THE OPENING AND THE CLOSING OF THE GRIPPER

The *Robotiq 2-Finger Adaptive Robot Gripper* allows for the detection of objects at the opening and closing of the Gripper's fingers.

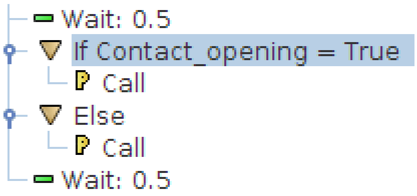


The OPEN and CLOSE programs are designed to recognize object detection.
Two booleans, *Contact_opening* and *Contact_closing* will automatically be recorded in the case of detection or not, during the execution of these programs. Please note that *Contact_closing* is not reliable for thin objects to avoid false detections when the fingers touch themselves.

OPENING DETECTION

During the execution of the OPEN program:

- If an object is detected, the global variable *Contact_opening* will be recorded as *True*.
- If an object is not detected, the global variable *Contact_opening* will be recorded as *False*.



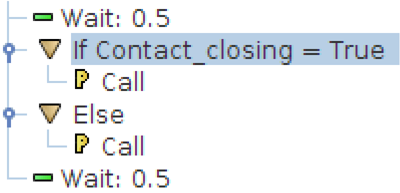
```
Wait: 0.5
If Contact_opening = True
  Call
Else
  Call
Wait: 0.5
```

You can also test this variable in the main program to manage object detections by the Robotiq Gripper.

CLOSING DETECTION

During the execution of the CLOSE program:

- If an object is detected, the global variable *Contact_closing* will be recorded as *True*.
- If an object is not detected, the global variable *Contact_closing* will be recorded as *False*.



```
Wait: 0.5
If Contact_closing = True
  Call
Else
  Call
Wait: 0.5
```

You can also test this variable in the main program to manage object detections by the Robotiq Gripper.

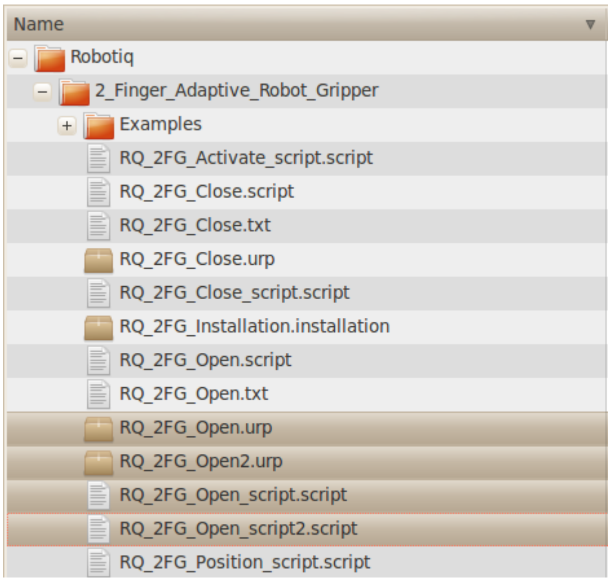
USING SEVERAL SIMILAR OPEN AND CLOSE PROGRAMS AS SUBROUTINES WITH DIFFERENT PARAMETERS IN THE SAME PROGRAM

You might want the Gripper to fit several different types of parts or pieces (different sizes, fragility, etc.) with the same main program. To avoid material damage or to gain execution time, the same subroutine can be used in your program, but with different parameters for: *Position*, *Speed* and *Force*.

To do this you need to manually create a program matching your desired functions, but using a different name and different variables:

First step:

- Copy the following files and rename them as desired.



```
Name
Robotiq
  2_Finger_Adaptive_Robot_Gripper
    Examples
    RQ_2FG_Activate_script.script
    RQ_2FG_Close.script
    RQ_2FG_Close.txt
    RQ_2FG_Close.urp
    RQ_2FG_Close_script.script
    RQ_2FG_Installation.installation
    RQ_2FG_Open.script
    RQ_2FG_Open.txt
    RQ_2FG_Open.urp
    RQ_2FG_Open2.urp
    RQ_2FG_Open_script.script
    RQ_2FG_Open_script2.script
    RQ_2FG_Position_script.script
```

In this example, *RQ_2FG_Open.urp* et *RQ_2FG-Open_script.script has* been copied and renamed *RQ_2FG_Open2.urp* and *RQ_2FG-Open_script2.script*

Second step:

- Open Polyscope and load the copied and renamed program (in this example, *RQ_2FG_Open2.urp*).
- To do this, open the programming page (home page and '*PROGRAM Robot*' button), go to the '*Load / Save*' section of the '*Installation*' tab. Fill in the field for the file *.urp*: Then, choose the related copied and renamed program file.

Third step:

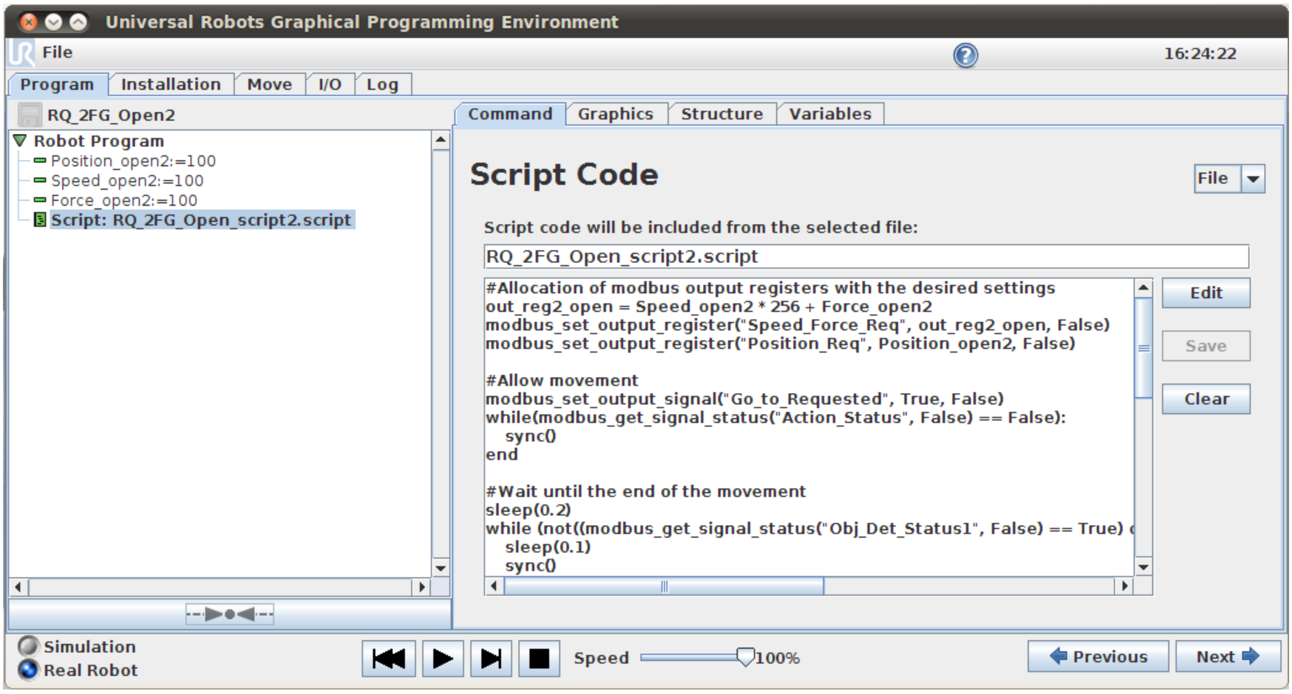
- Edit the copied and renamed program(s):
  - Rename ***Position_x***, ***Speed_x*** and ***Force_ x*** variables (in this example, *Position_open2*, *Speed_ open2* and *Force_ open2*) and change their values.
- Change the *.script* file of the *Script* structure.
  - To do this, remove the file named *Script: RQ_2FG_x_script.script* (select it on the left of the screen, '*Structure*' tab and '*Delete*' button).
- Then, add *Script code* structure in the same place ('*Advanced*' tab of '*Structure*' tab).
- Select the copied and renamed .script file (Select *Script* on the left of the screen, choose '*File*' in the '*Command*' tab, '*Edit*' button, '*Select File*' button, select the file and '*SAVE & EXIT*' button).
- Modify this script in the '*Command*' tab and replace the previously known variables, like this:

```
#Allocation of modbus output registers with the desired settings
out_reg2_open = Speed_open2 * 256 + Force_open2
modbus_set_output_register("Speed_Force_Req", out_reg2_open, False)
modbus_set_output_register("Position_Req", Position_open2, False)
```

Last step:

- Save the *.script* file and the program in the desired folder. It will then be possible to use this program as a subroutine.

Here are the results for this example:



Robot Program
- Position_open2:=100
- Speed_open2:=100
- Force_open2:=100
- Script: RQ_2FG_Open_script2.script

Script Code

Script code will be included from the selected file:

RQ_2FG_Open_script2.script

```
#Allocation of modbus output registers with the desired settings
out_reg2_open = Speed_open2 * 256 + Force_open2
modbus_set_output_register("Speed_Force_Req", out_reg2_open, False)
modbus_set_output_register("Position_Req", Position_open2, False)

#Allow movement
modbus_set_output_signal("Go_to_Requested", True, False)
while(modbus_get_signal_status("Action_Status", False) == False):
   sync()
end

#Wait until the end of the movement
sleep(0.2)
while (not((modbus_get_signal_status("Obj_Det_Status1", False) == True)
   sleep(0.1)
   sync()
```

EXAMPLE: CLOSING DECTECTION *RQ_2FG_Closing_Detection_Demo.urp*

Load the program named **RQ_2FG_Closing_Detection_Demo.urp** onto Polyscope (Home page and '*PROGRAM Robot*' button), click on '*Load Program*' in the '*Program*' tab.

Program description:

- Robotiq Gripper activation
- Once the activation is complete, the program will run an infinite loop:
    - Robotiq Gripper closes
    - Wait for 500 ms
    - If a part is detected when the Robotiq Gripper closes:
        - The Robotiq Gripper slowly opens with a minimum of force and not completely (position = speed = force = 100)
    - Else (If no part is detected when the Robotiq Gripper closes):
        - The Robotiq Gripper opens completely and rapidly with a maximum of force (position = 0, speed = force = 255)
    - Wait for 500 ms
    - (Back to the top of the loop)

Note: The position of the Robotiq Gripper's fingers is given by the *Grip_position* variable.

7. USING SUBROUTINES WITH UNIVERSAL ROBOTS

The main objective is to program *Universal Robots* and the *Robotiq Gripper* from the same program.

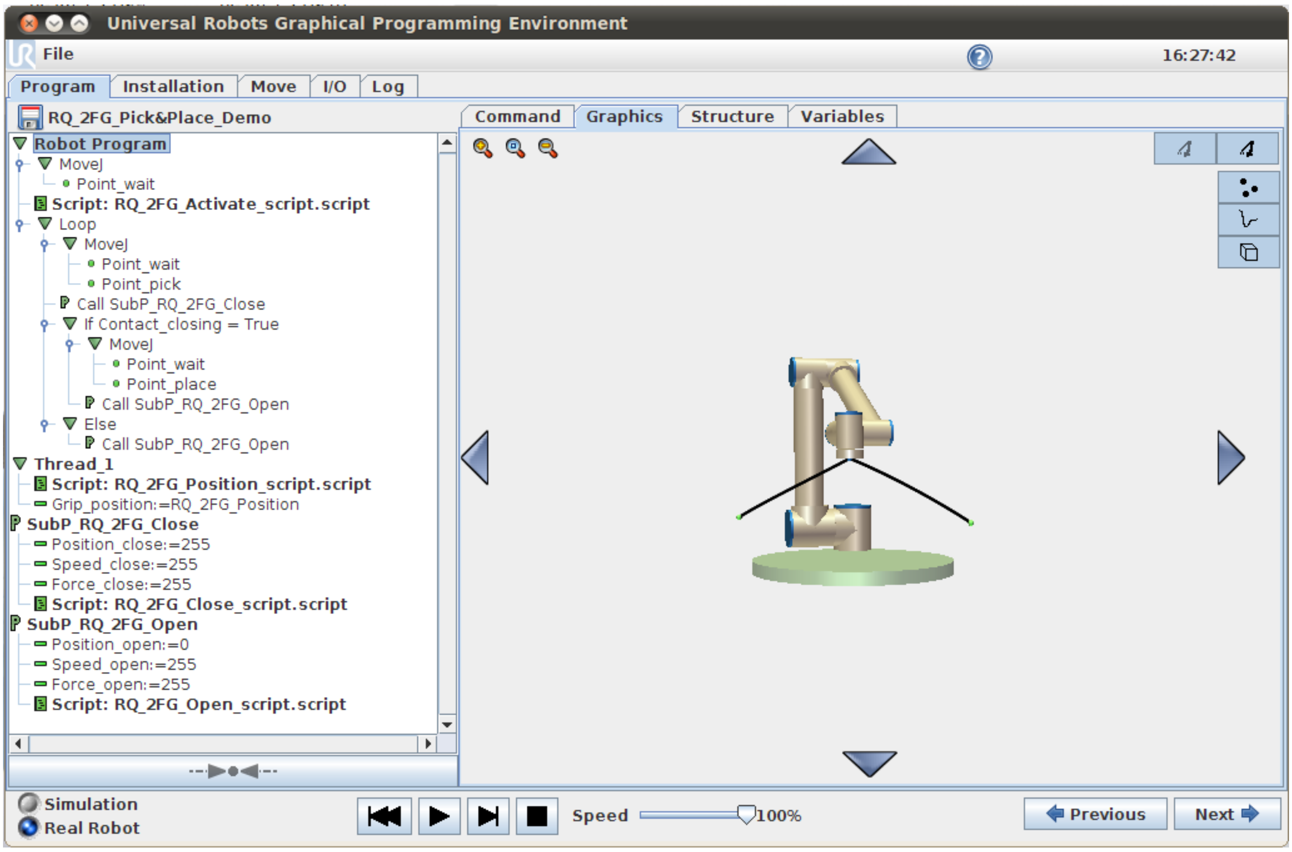
Here is an example of the integration of the Robotiq Gripper into a program which picks and places objects:

Load **RQ_2FG_Pick&Place_Demo.urp** program in Polyscope (Home page and '*PROGRAM Robot*' button), click on '*Load Program*' in the '*Program*' tab.

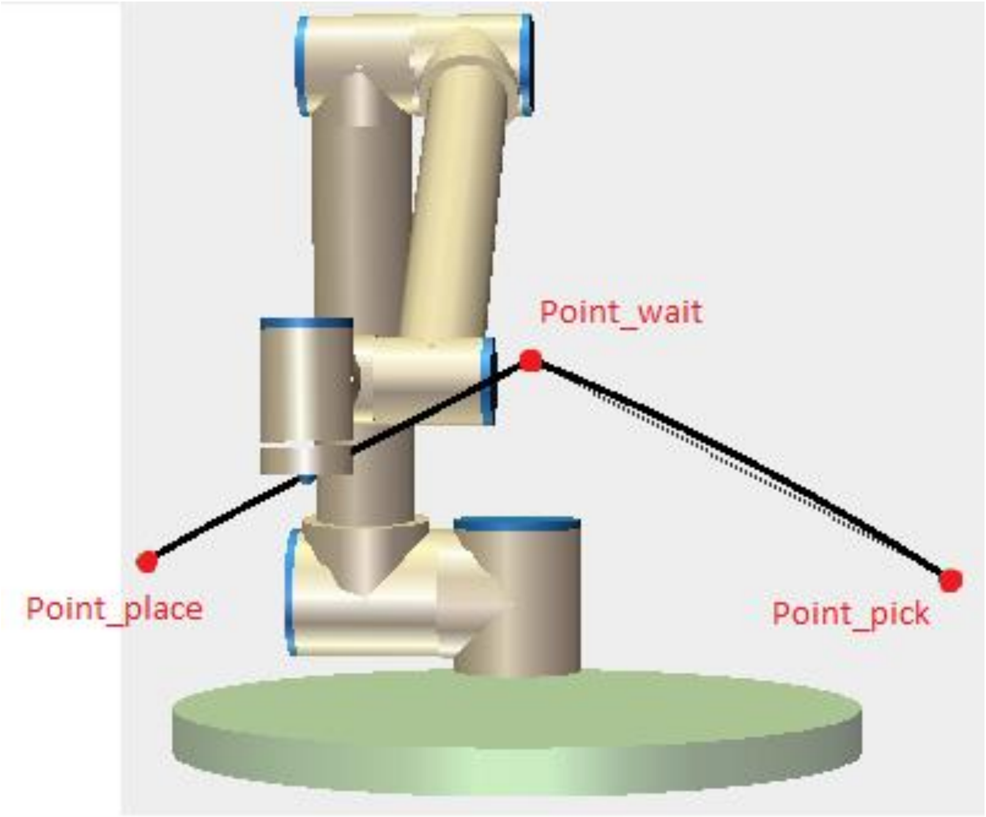
Program description:

- The robot will move to the location of the objects (*Point_wait*)
- Robotiq Gripper activates
- Once the activation is complete, the program will run an infinite loop:
    - The robot moves to the collection point for the objects (*Point_pick*)
    - Robotiq Gripper closes
    - If a part is detected when the Robotiq Gripper is closed:
        - The robot will move to the point for the disposal of the objects (*Point_place*), passing by the holding point of the objects (*Point_wait*)
        - Robotiq Gripper opens
    - Else (If no part is detected when the Robotiq Gripper is closed):
        - Robotiq Gripper opens
    - (Back to the top of the loop)

Note: The position of the Robotiq Gripper's fingers is given by the *Grip_position* variable.



```
Robot Program
  MoveJ
    Point_wait
  Script: RQ_2FG_Activate_script.script
  Loop
    MoveJ
      Point_wait
      Point_pick
    Call SubP_RQ_2FG_Close
    If Contact_closing = True
      MoveJ
        Point_wait
        Point_place
      Call SubP_RQ_2FG_Open
    Else
      Call SubP_RQ_2FG_Open
Thread_1
  Script: RQ_2FG_Position_script.script
  Grip_position:=RQ_2FG_Position
SubP_RQ_2FG_Close
  Position_close:=255
  Speed_close:=255
  Force_close:=255
  Script: RQ_2FG_Close_script.script
SubP_RQ_2FG_Open
  Position_open:=0
  Speed_open:=255
  Force_open:=255
  Script: RQ_2FG_Open_script.script
```


Point_wait
Point_place
Point_pick