Getting started with python-urx

# Setup

## UR5 robot

* Start up UR5 to the main screen.
* Note the robot IP address by clicking “About” on the Polyscope main page.
* To use the gripper, it must first be activated
  + Program Robot > Installation > Gripper > Activate

### UR5e robot

* In addition to the above steps, ensure the “Remote Control” option is selected from the top-right corner.

## Workstation

* Install Python 3.
* Download and install urx (NOTE: the package available from pip appears to be outdated and cause unexpected errors to be thrown; please download the package directly from GitHub at <https://github.com/SintefManufacturing/python-urx>). One way to download and install the latest version of urx is to:
  + Download the package by going to the Git repository in a browser and clicking “Code” -> “Download ZIP”.
  + Extract the files in your Python working directory.
  + Navigate to the package root directory (*i.e.*, the level ‘setup.py’ is in).
  + Execute >> python setup.py install in the terminal/command line.
* If you installed urx following the previous step, all dependencies (namely numpy and math3d) should have also been installed.
* Connect to the robot router.

# Test code

## Moving the arm

* Ensure the IP address in the move\_arm.py script matches that of the physical robot.
* Run the move\_arm.py script  
  >> python move\_arm.py

## Using the gripper (ENSURE GRIPPER HAS BEEN ACTIVATED AS IN SECT. 1.1)

* Ensure the IP address in the gripper.py script matches that of the physical robot.
* Run the gripper.py script  
  >> python gripper.py

# Capturing an image from the camera

The images captured by the UR5 camera can be accessed from:

http://< UR5 IP address >:4242/current.jpg?annotations=off

To test this, type the above address into a browser while connected to a robot router.

# Additional information

* You may want to set up the tool centre position (TCP) and payload to correspond to the end of the gripper. To do this, call the set\_tcp and set\_payload functions at the beginning of your Python script, for example:  
   rob = urx.Robot("<IP ADDRESS>")  
   rob.set\_tcp((0, 0, 0.1755, 0, 0, 0))  
   rob.set\_payload(1, (-0.0013, 0.0013, 0.0559))  
  Make sure the arguments of the set\_tcp and set\_payload functions are exactly as they appear above [Source: Robotiq support, camera + 2F-85 combo].
* opencv will likely be useful for completing the project. This can be installed as any other Python package, for example, using pip:  
  >> pip install opencv-python  
  or conda:  
  >> conda install -c conda-forge opencv

# Troubleshooting

## ‘Connection timeout’ error

* Make sure you are connected to the correct robot router, and have entered the correct robot IP address into your python script.
* Try running the script several times – sometimes it takes several attempts to establish a connection.

## ‘Robot Stopped’ error

* Make sure you are running the latest version of urx by installing it directly from GitHub, following the instructions proved above.
* Make sure robot control box is powered on, and the arm has been initialised.
* For UR5e: Make sure the robot is in ‘Remote Control’ mode (not ‘Local’).

## Unable to access camera image via URL

* Check the Vision System is running by viewing the camera feed on the Teach Pendant (Installation -> URCaps -> Camera -> Camera tab).
* If a live camera feed is not displayed, restart the robot.

## Unable to control gripper remotely

* Check the gripper has been activated on the Teach Pendant (Installation -> URCaps -> Gripper -> Activate)
* On the same page, check the gripper ID is set to 1.

## Other

* If the above steps do not solve your problem, as always, try turning the robot off and on!

# References

* Python-urx Git repository:

<https://github.com/SintefManufacturing/python-urx>

* <https://dof.robotiq.com/discussion/275/wrist-camera-live-image>