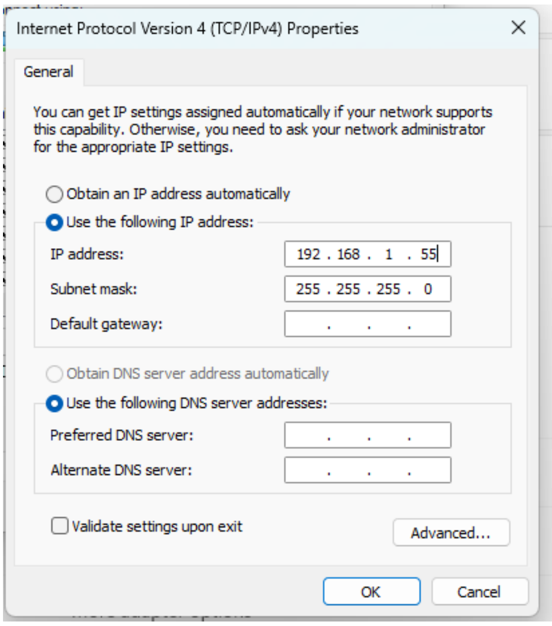
Request response protocol Python example

This manual is step by step manual to run Python example with Photoneo Locator Studio

1. Set of IP address

"Robot controller" ip address in Vision controller Locator Studio must be the same ip address as your computer ipv4

Your PC:



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 55

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

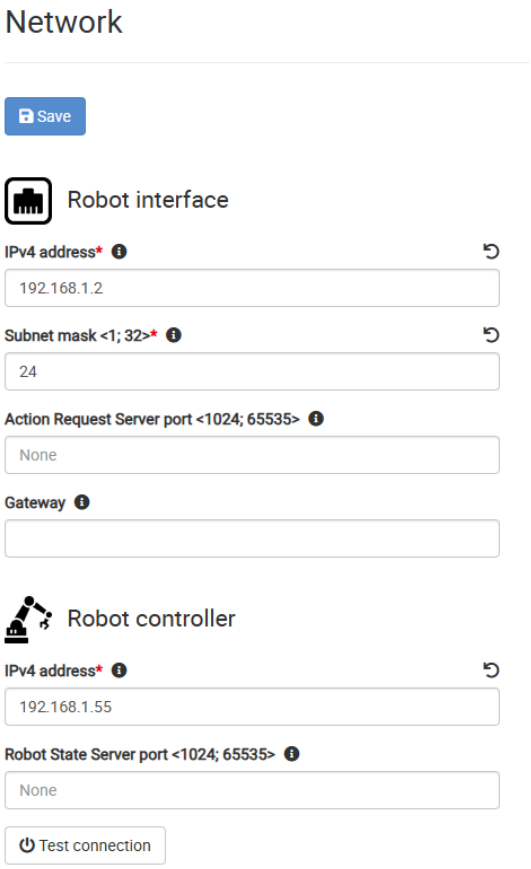
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Vision controller = Locator Studio



Network

Robot interface

IPv4 address*: 192.168.1.2

Subnet mask <1; 32>*: 24

Action Request Server port <1024; 65535>: None

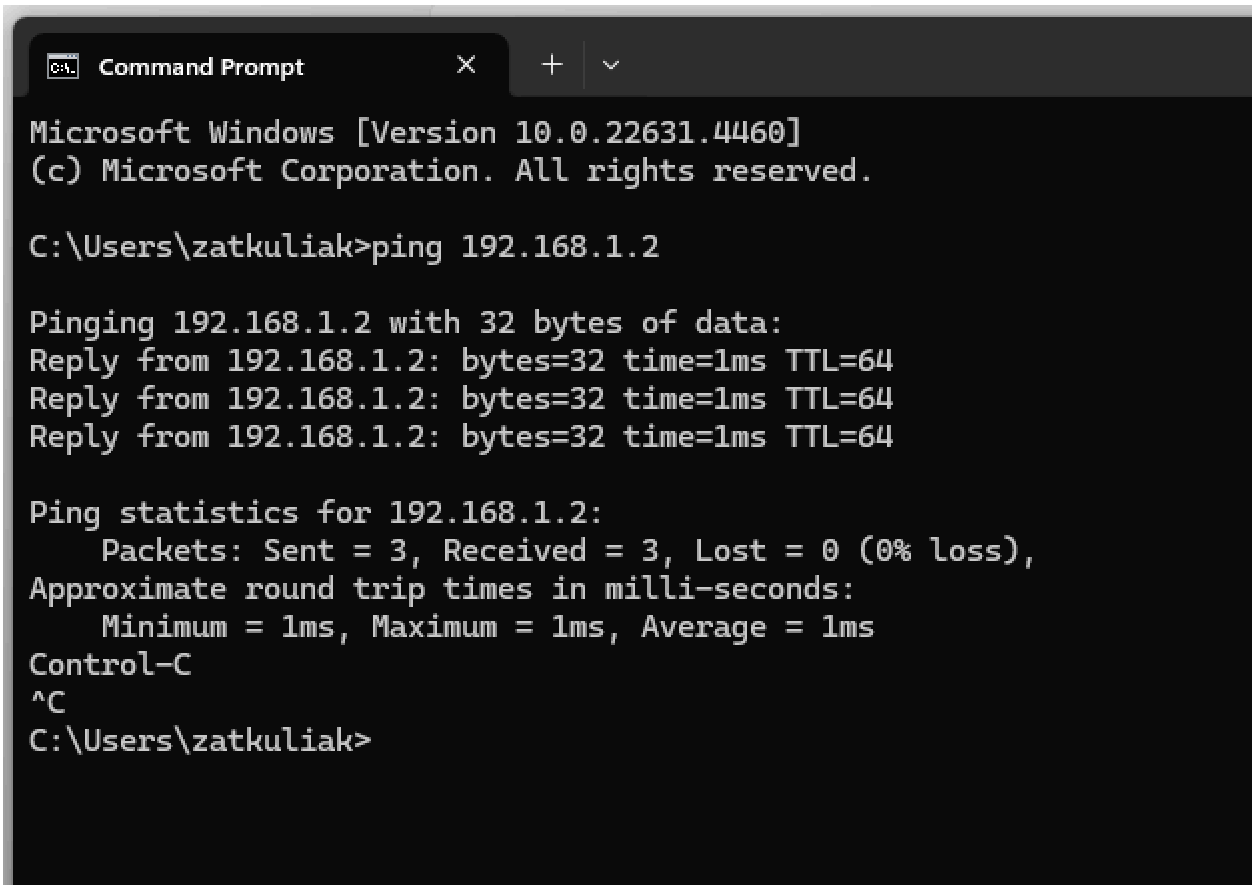
Gateway:

Robot controller

IPv4 address*: 192.168.1.55

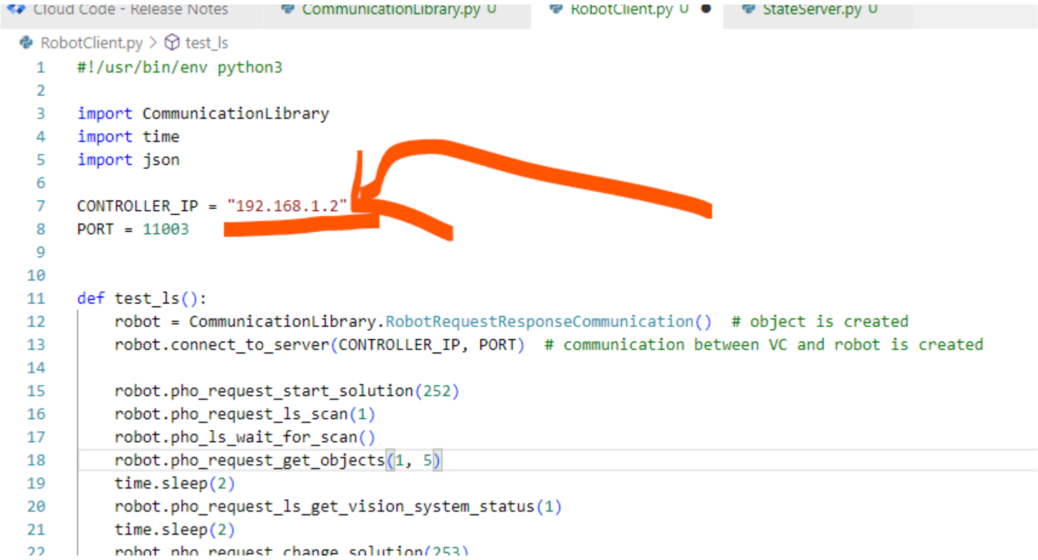
Robot State Server port <1024; 65535>: None

Proof is to test connection from LS or ping ip address of robot interface of LS

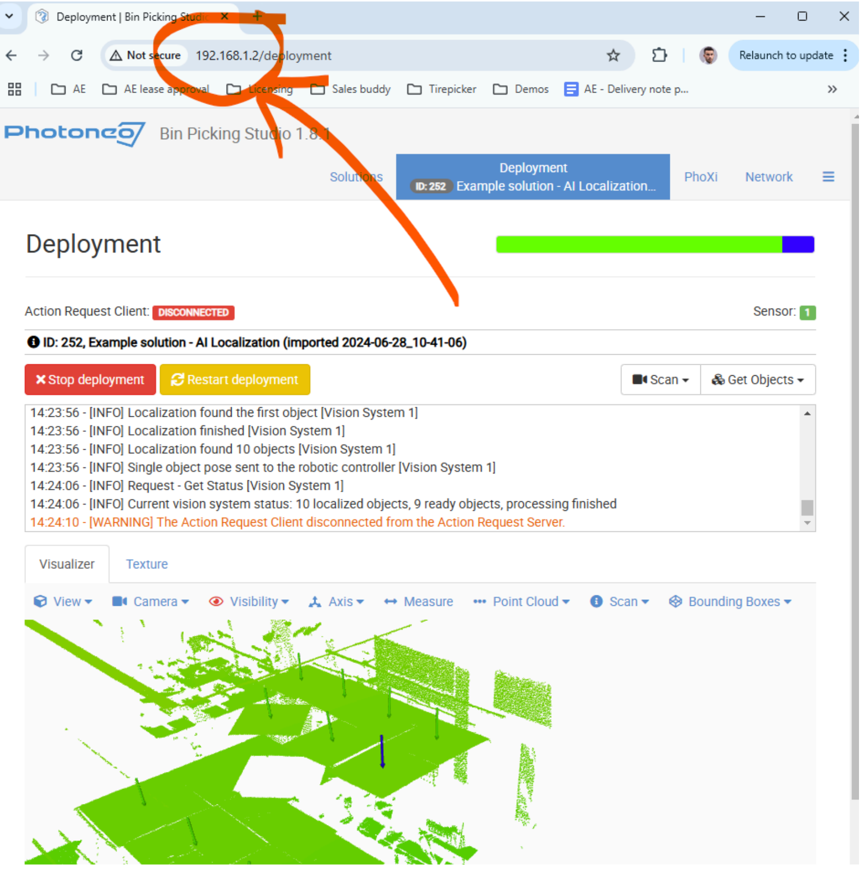


```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\zatkuliak>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=1ms TTL=64
Reply from 192.168.1.2: bytes=32 time=1ms TTL=64
Reply from 192.168.1.2: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.2:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
Control-C
^C
C:\Users\zatkuliak>
```

Optional: You can connect to Vision controller via browser based on IP address:



```
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```



Deployment

Action Request Client: DISCONNECTED

Sensor: 1

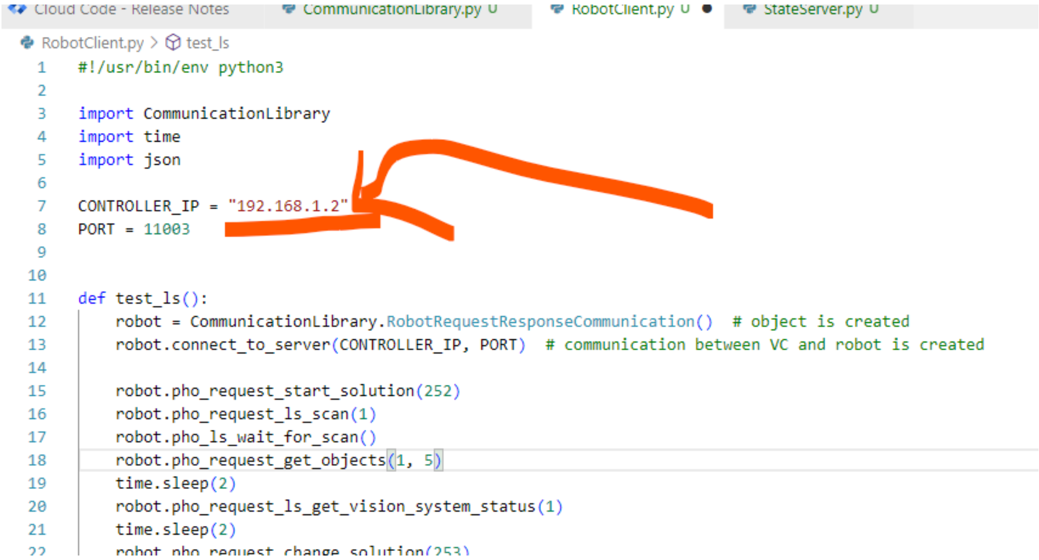
ID: 252, Example solution - AI Localization (imported 2024-06-28_10-41-06)

Stop deployment | Restart deployment | Scan | Get Objects

```
14:23:56 - [INFO] Localization found the first object [Vision System 1]
14:23:56 - [INFO] Localization finished [Vision System 1]
14:23:56 - [INFO] Localization found 10 objects [Vision System 1]
14:23:56 - [INFO] Single object pose sent to the robotic controller [Vision System 1]
14:24:06 - [INFO] Request - Get Status [Vision System 1]
14:24:06 - [INFO] Current vision system status: 10 localized objects, 9 ready objects, processing finished
14:24:10 - [WARNING] The Action Request Client disconnected from the Action Request Server.
```

:

support@photoneo.com Photoneo Request res

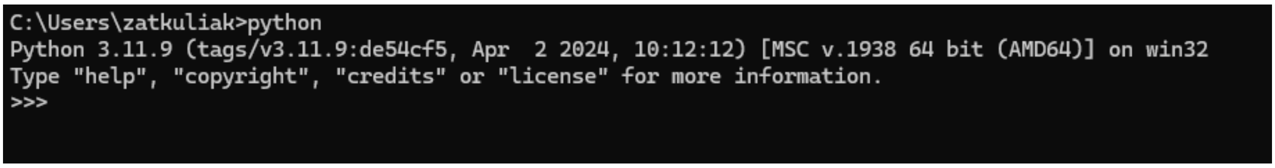


```python
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```

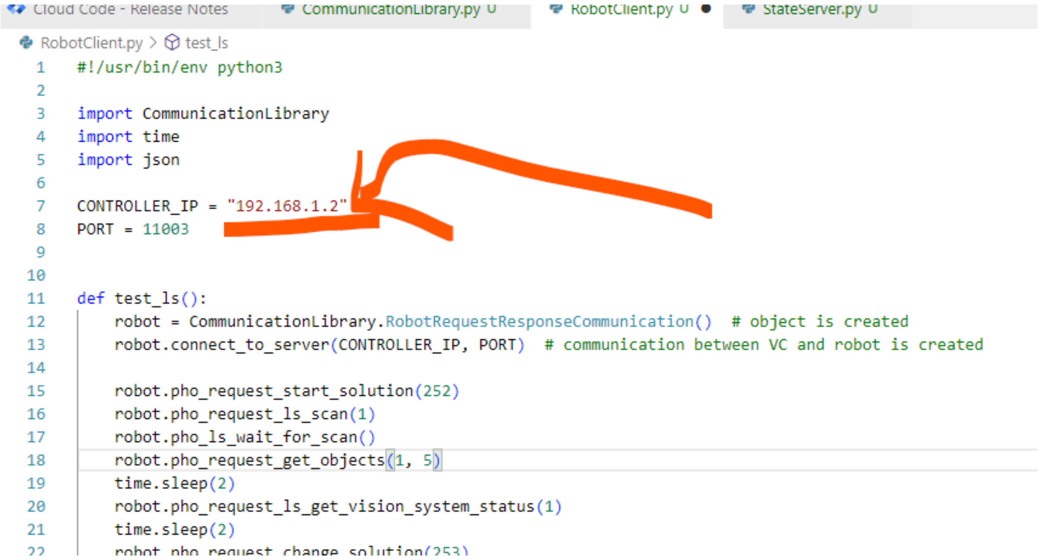
ponse protocol Python

example

2. Check if Python installed on PC



```
C:\Users\zatkuliak>python
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



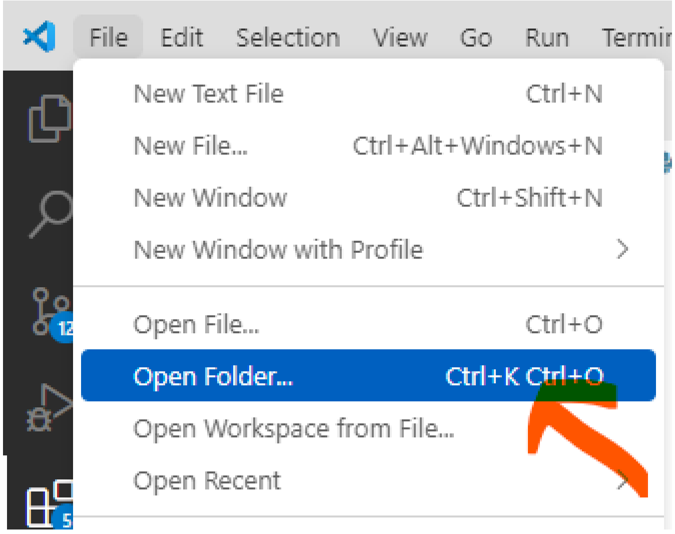
```
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```

3. Download Python example

a. *Extract all,*

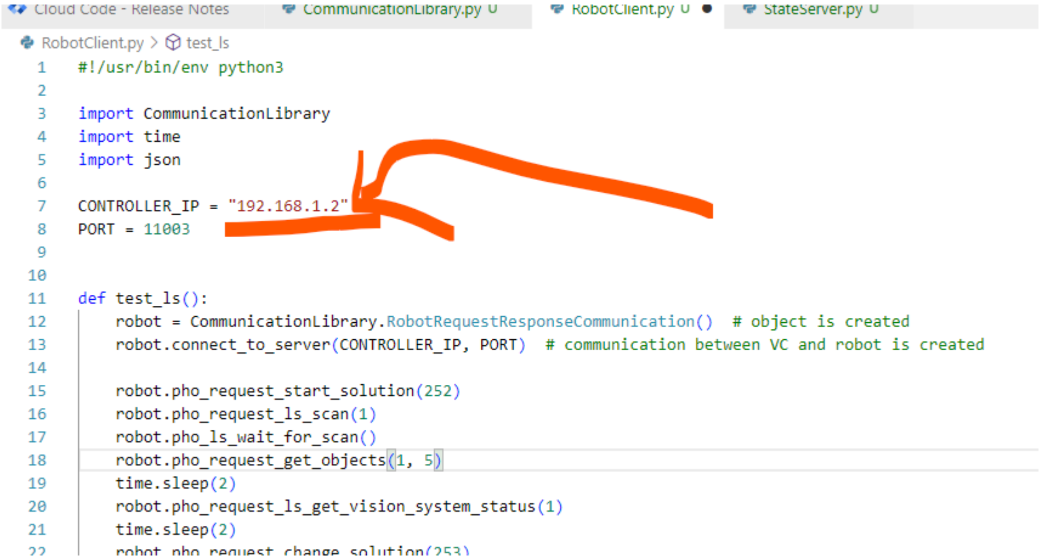
b. *save to folder,*

c. *Open in Visual studio Code this folder:*



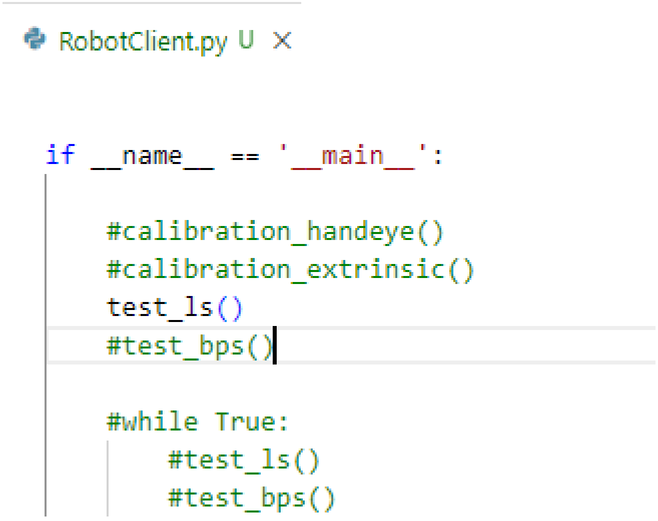
4. Modify Python example

a. *IP of robot interface*



```python
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```

b. *Choose by commenting what product you want to use = in this case ls = Locator Studio*



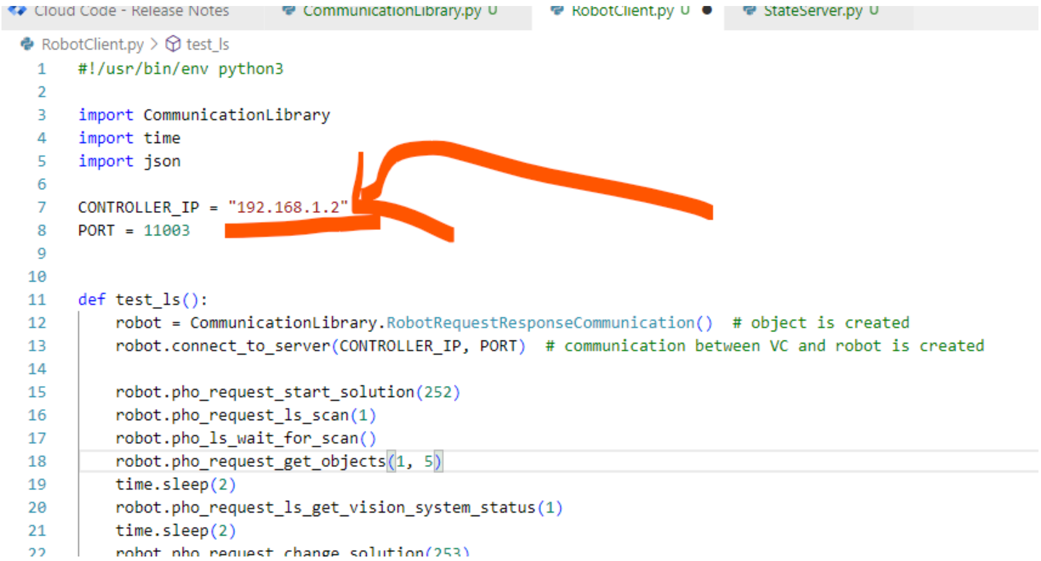
```python
if __name__ == '__main__':

    #calibration_handeye()
    #calibration_extrinsic()
    test_ls()
    #test_bps()

    #while True:
        #test_ls()
        #test_bps()
```

5. Check if Python has all libraries, if missing, install them

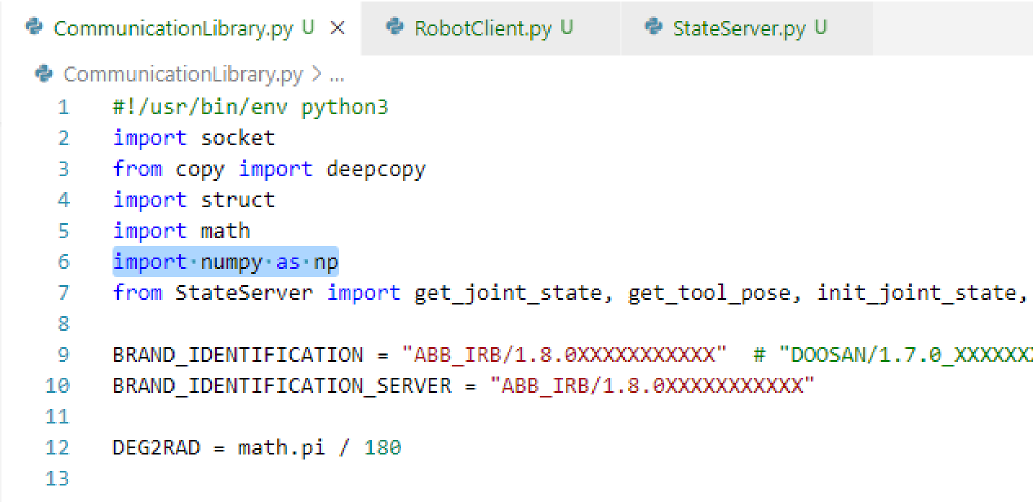
a. You need to have installed NUMPY library, in terminal write:

```
pip install numpy
```

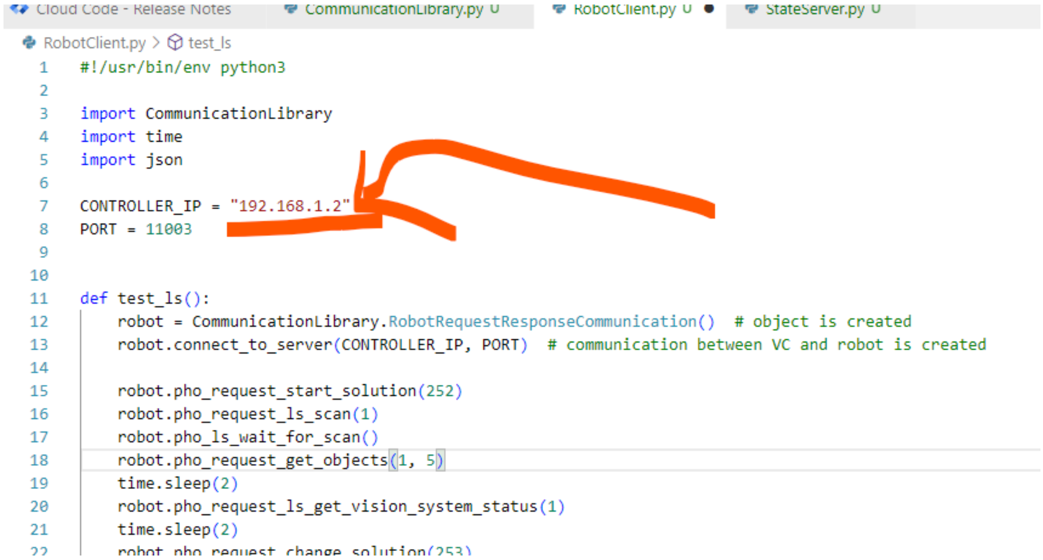
Is good like this:



```python
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```

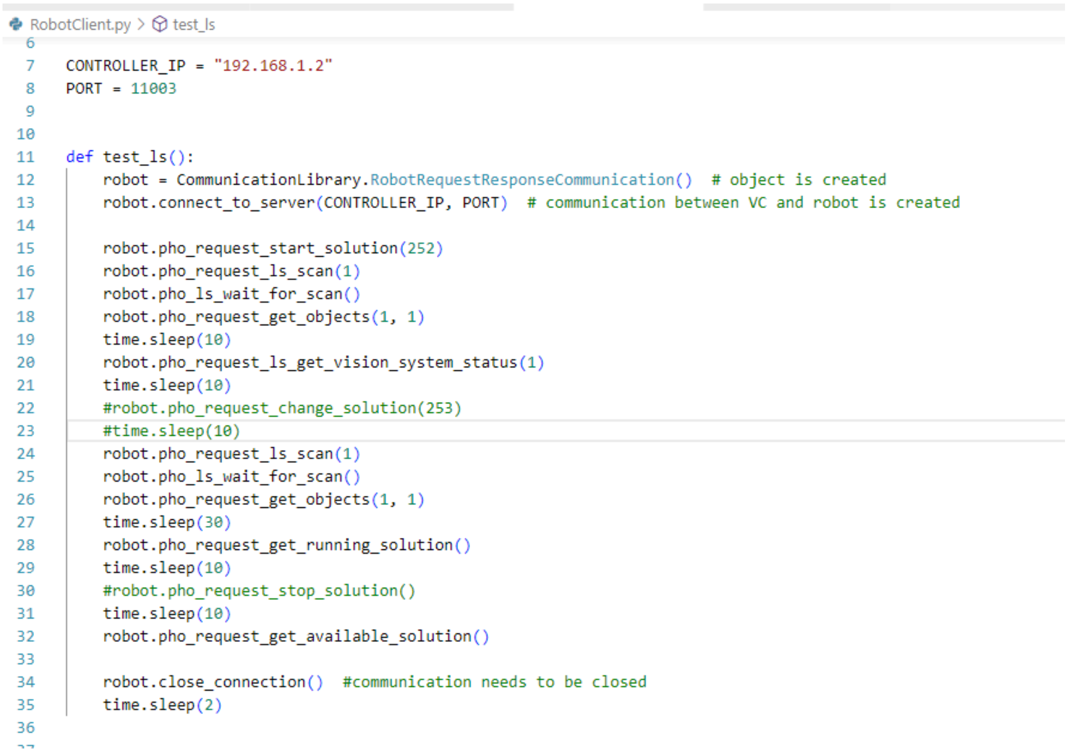


```python
#!/usr/bin/env python3
import socket
from copy import deepcopy
import struct
import math
import numpy as np
from StateServer import get_joint_state, get_tool_pose, init_joint_state,

BRAND_IDENTIFICATION = "ABB_IRB/1.8.0XXXXXXXXXXX"  # "DOOSAN/1.7.0_XXXXXX
BRAND_IDENTIFICATION_SERVER = "ABB_IRB/1.8.0XXXXXXXXXXX"

DEG2RAD = math.pi / 180

```



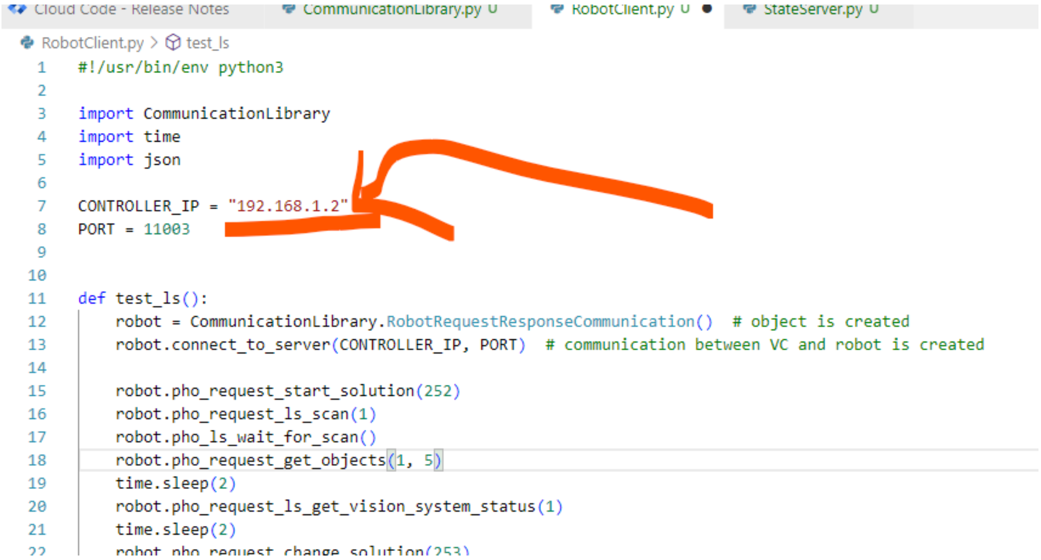
```python
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```

6. Modify your program step by step what you want to:



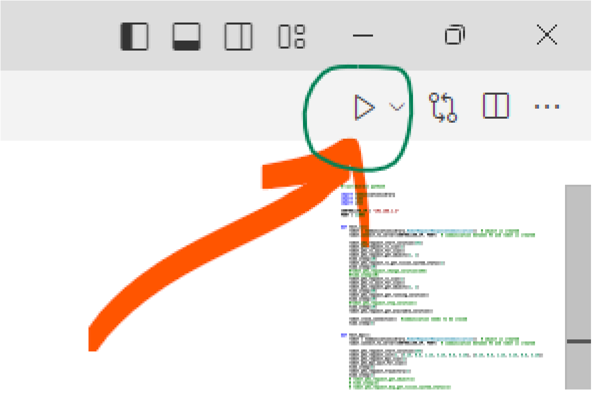
```python

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 1)
    time.sleep(10)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(10)
    #robot.pho_request_change_solution(253)
    #time.sleep(10)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 1)
    time.sleep(30)
    robot.pho_request_get_running_solution()
    time.sleep(10)
    #robot.pho_request_stop_solution()
    time.sleep(10)
    robot.pho_request_get_available_solution()

    robot.close_connection()  #communication needs to be closed
    time.sleep(2)

```

`robot.pho_request_start_solution(252)` = for example represent solution ID on vision controller, you can comment this line if you Deploy solution manually

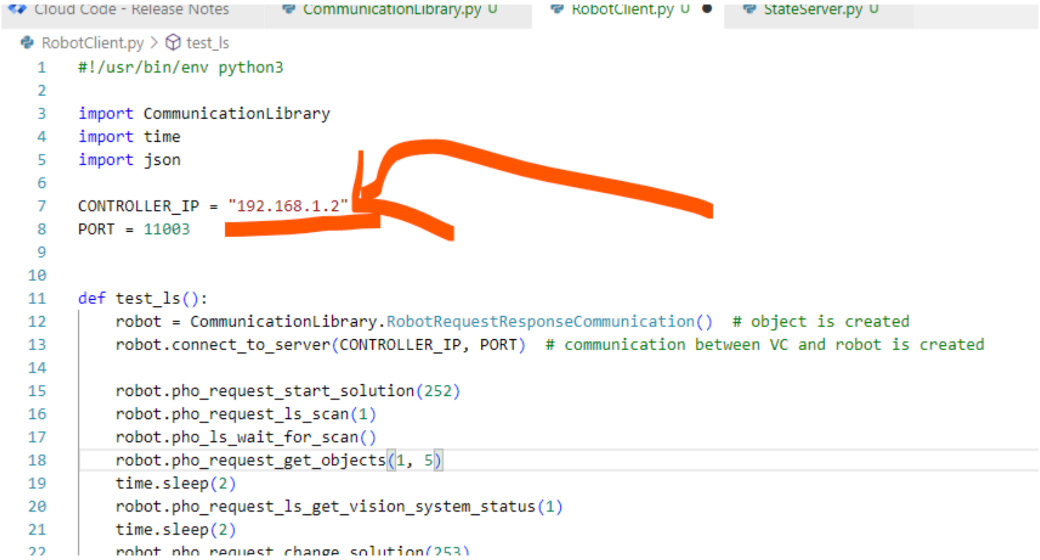


```python
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```

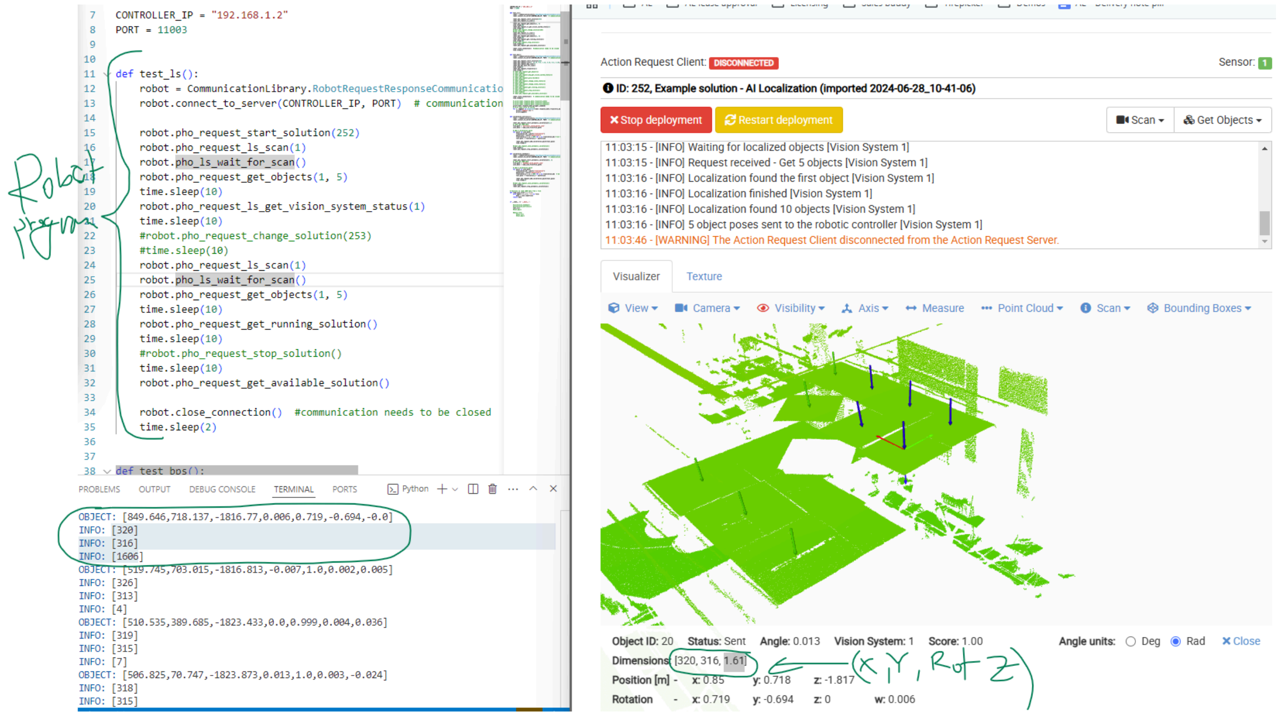
7. RUN the program



8. Enjoy results



```
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```



```python
CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunicatio
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(10)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(10)
    #robot.pho_request_change_solution(253)
    #time.sleep(10)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(10)
    robot.pho_request_get_running_solution()
    time.sleep(10)
    #robot.pho_request_stop_solution()
    time.sleep(10)
    robot.pho_request_get_available_solution()

    robot.close_connection()  #communication needs to be closed
    time.sleep(2)


def test_bps():
```

```
OBJECT: [849.646,718.137,-1816.77,0.006,0.719,-0.694,-0.0]
INFO: [320]
INFO: [316]
INFO: [1606]
OBJECT: [519.745,703.015,-1816.813,-0.007,1.0,0.002,0.005]
INFO: [326]
INFO: [313]
INFO: [4]
OBJECT: [510.535,389.685,-1823.433,0.0,0.999,0.004,0.036]
INFO: [319]
INFO: [315]
INFO: [7]
OBJECT: [506.825,70.747,-1823.873,0.013,1.0,0.003,-0.024]
INFO: [318]
INFO: [315]
```

Action Request Client: DISCONNECTED

Sensor: 1

ID: 252, Example solution - AI Localization (imported 2024-06-28_10-41-06)

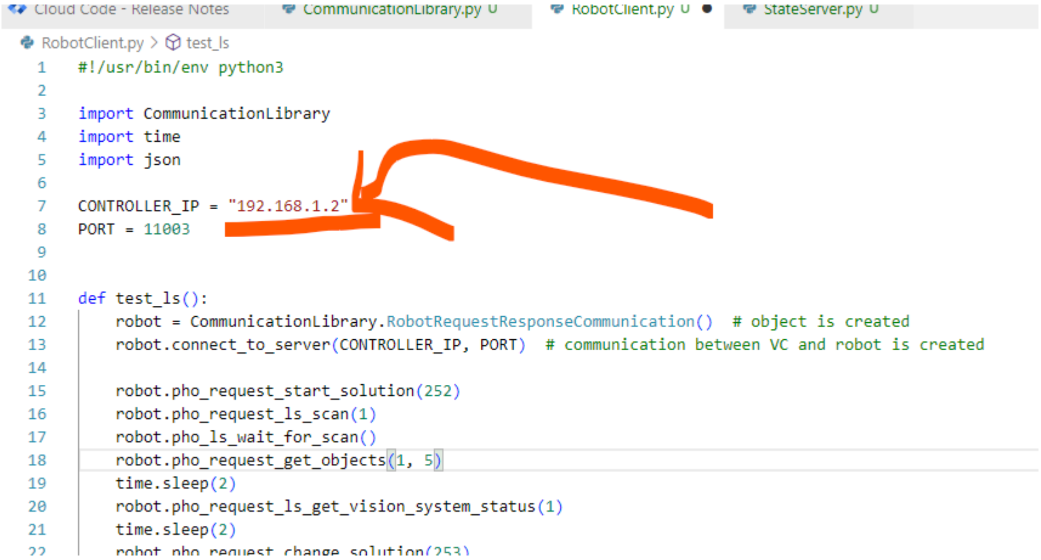
```
11:03:15 - [INFO] Waiting for localized objects [Vision System 1]
11:03:15 - [INFO] Request received - Get 5 objects [Vision System 1]
11:03:16 - [INFO] Localization found the first object [Vision System 1]
11:03:16 - [INFO] Localization finished [Vision System 1]
11:03:16 - [INFO] Localization found 10 objects [Vision System 1]
11:03:16 - [INFO] 5 object poses sent to the robotic controller [Vision System 1]
11:03:46 - [WARNING] The Action Request Client disconnected from the Action Request Server.
```

Object ID: 20 Status: Sent Angle: 0.013 Vision System: 1 Score: 1.00 Angle units: ○ Deg ● Rad

Dimensions: [320, 316, 1.61] ← (X, Y, Rot Z)

Position [m] - x: 0.85 y: 0.718 z: -1.817

Rotation - x: 0.719 y: -0.694 z: 0 w: 0.006



```python
#!/usr/bin/env python3

import CommunicationLibrary
import time
import json

CONTROLLER_IP = "192.168.1.2"
PORT = 11003


def test_ls():
    robot = CommunicationLibrary.RobotRequestResponseCommunication()  # object is created
    robot.connect_to_server(CONTROLLER_IP, PORT)  # communication between VC and robot is created

    robot.pho_request_start_solution(252)
    robot.pho_request_ls_scan(1)
    robot.pho_ls_wait_for_scan()
    robot.pho_request_get_objects(1, 5)
    time.sleep(2)
    robot.pho_request_ls_get_vision_system_status(1)
    time.sleep(2)
    robot.pho_request_change_solution(253)
```