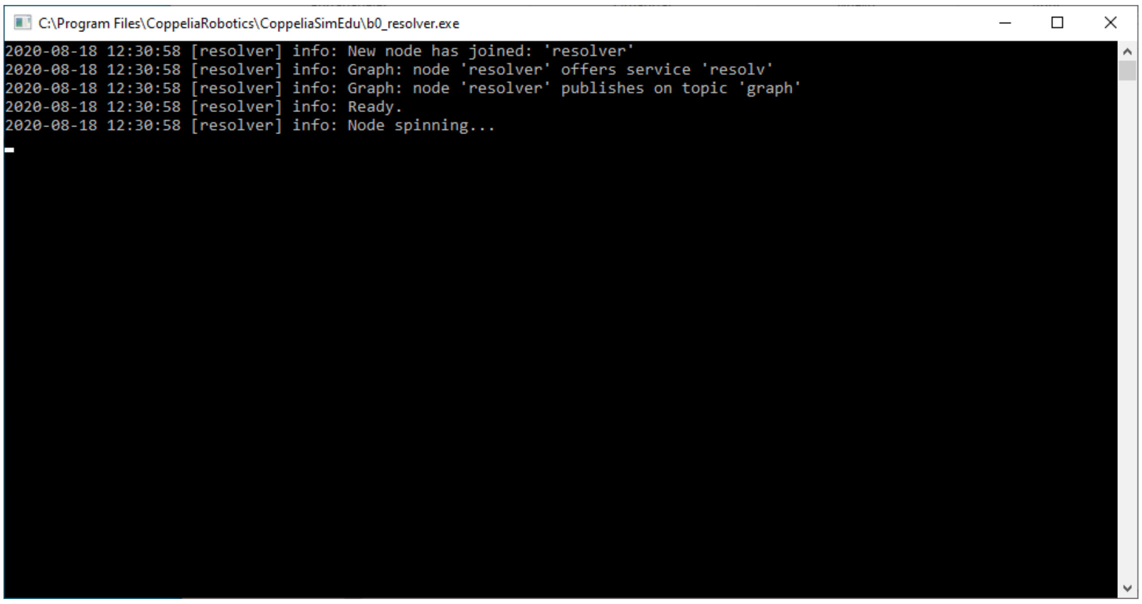
B0RemoteApi test

Follow the next procedure for Windows 10:

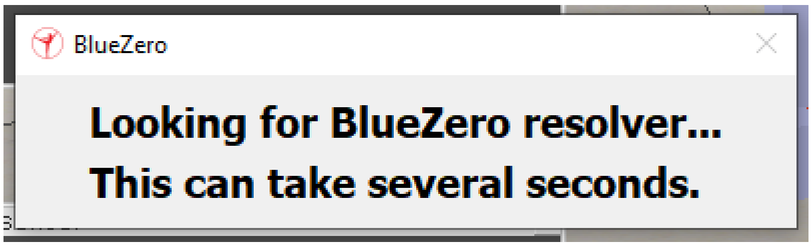
1-Run *B0_resolver.exe*.


C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\b0_resolver.exe

```
2020-08-18 12:30:58 [resolver] info: New node has joined: 'resolver'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' offers service 'resolv'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' publishes on topic 'graph'
2020-08-18 12:30:58 [resolver] info: Ready.
2020-08-18 12:30:58 [resolver] info: Node spinning...
```

Figure 1: b0_resolver running

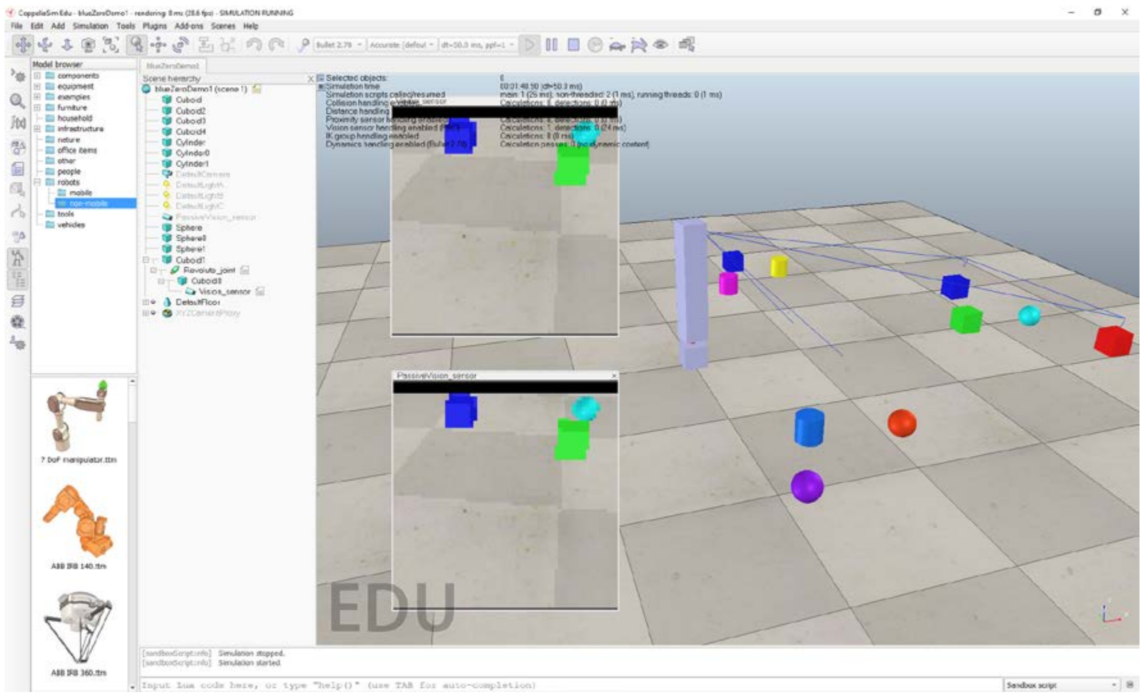
This pass is irrelevant because if you forgot start B0_resolver the simulator tries to start it. You will see a window at the center of the screen:



Looking for BlueZero resolver...
This can take several seconds.

Figure 2: Looking for BlueZero resolver

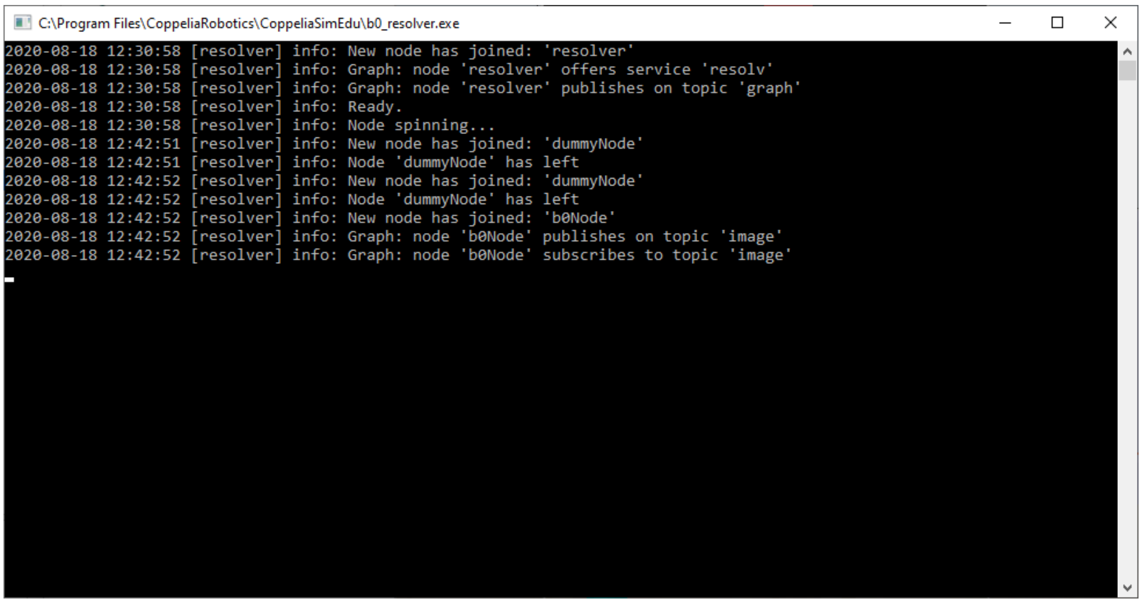
2- Open *blueZeroDemo1.ttt* and Start simulation



```
[sandboxScript:info]  Simulation stopped.
[sandboxScript:info]  Simulation started.
```

Figure 3: blueZeroDermo1 running with simulation started

3- Take a look to *B0_resolver* window:



C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\b0_resolver.exe

```
2020-08-18 12:30:58 [resolver] info: New node has joined: 'resolver'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' offers service 'resolv'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' publishes on topic 'graph'
2020-08-18 12:30:58 [resolver] info: Ready.
2020-08-18 12:30:58 [resolver] info: Node spinning...
2020-08-18 12:42:51 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 12:42:51 [resolver] info: Node 'dummyNode' has left
2020-08-18 12:42:52 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 12:42:52 [resolver] info: Node 'dummyNode' has left
2020-08-18 12:42:52 [resolver] info: New node has joined: 'b0Node'
2020-08-18 12:42:52 [resolver] info: Graph: node 'b0Node' publishes on topic 'image'
2020-08-18 12:42:52 [resolver] info: Graph: node 'b0Node' subscribes to topic 'image'
```

Figure 4: b0_resolver window after blueZeroDemo1 running with simulation started

You can see after dummyNode the node b0Node create at Vision_sensor script (lines 21-31):

```
    if simB0.pingResolver() then
        b0Node=simB0.nodeCreate('b0Node')

        -- Enable an image publisher and subscriber:
        pub=simB0.publisherCreate(b0Node,'image')
        sub=simB0.subscriberCreate(b0Node,'image','imageMessage_callback')

        simB0.nodeInit(b0Node)
     else
        sim.addLog(sim.verbosity_scripterrors,'B0 resolver could not be launched.')
     end
```

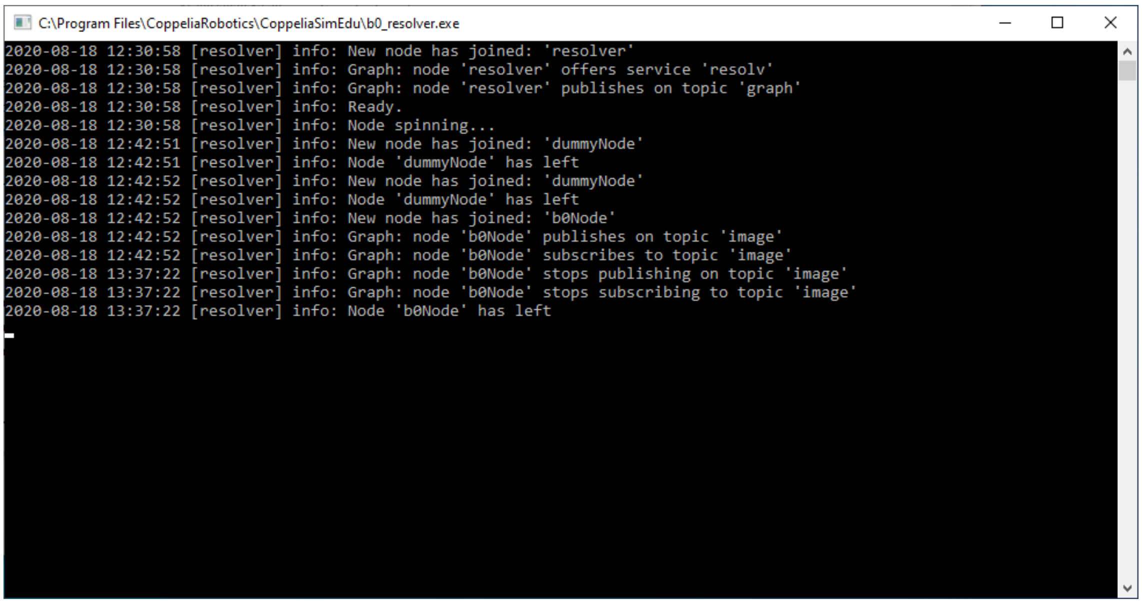
The Vision_sensor script tries to pass image from Vision_sensor (activeVisionSensor handle) to Floating view PassiveVisionSensor (passiveVisionSensor handle) via publisher:

```
function imageMessage_callback(msg)
    -- Apply the received image to the passive vision sensor that acts as an image
container
    sim.setVisionSensorCharImage(passiveVisionSensor,msg)
end


function sysCall_sensing()
    -- Publish the image of the active vision sensor:
    if b0Node then
        local msg=sim.getVisionSensorCharImage(activeVisionSensor)
        simB0.publisherPublish(pub,msg)
        simB0.nodeSpinOnce(b0Node) -- required for the subscriber
    end
end
```

Note: simB0 is the plugin

4- Stop simulation.

5- Take a look again to B0_resolver window:


C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\b0_resolver.exe

```
2020-08-18 12:30:58 [resolver] info: New node has joined: 'resolver'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' offers service 'resolv'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' publishes on topic 'graph'
2020-08-18 12:30:58 [resolver] info: Ready.
2020-08-18 12:30:58 [resolver] info: Node spinning...
2020-08-18 12:42:51 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 12:42:51 [resolver] info: Node 'dummyNode' has left
2020-08-18 12:42:52 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 12:42:52 [resolver] info: Node 'dummyNode' has left
2020-08-18 12:42:52 [resolver] info: New node has joined: 'b0Node'
2020-08-18 12:42:52 [resolver] info: Graph: node 'b0Node' publishes on topic 'image'
2020-08-18 12:42:52 [resolver] info: Graph: node 'b0Node' subscribes to topic 'image'
2020-08-18 13:37:22 [resolver] info: Graph: node 'b0Node' stops publishing on topic 'image'
2020-08-18 13:37:22 [resolver] info: Graph: node 'b0Node' stops subscribing to topic 'image'
2020-08-18 13:37:22 [resolver] info: Node 'b0Node' has left
```

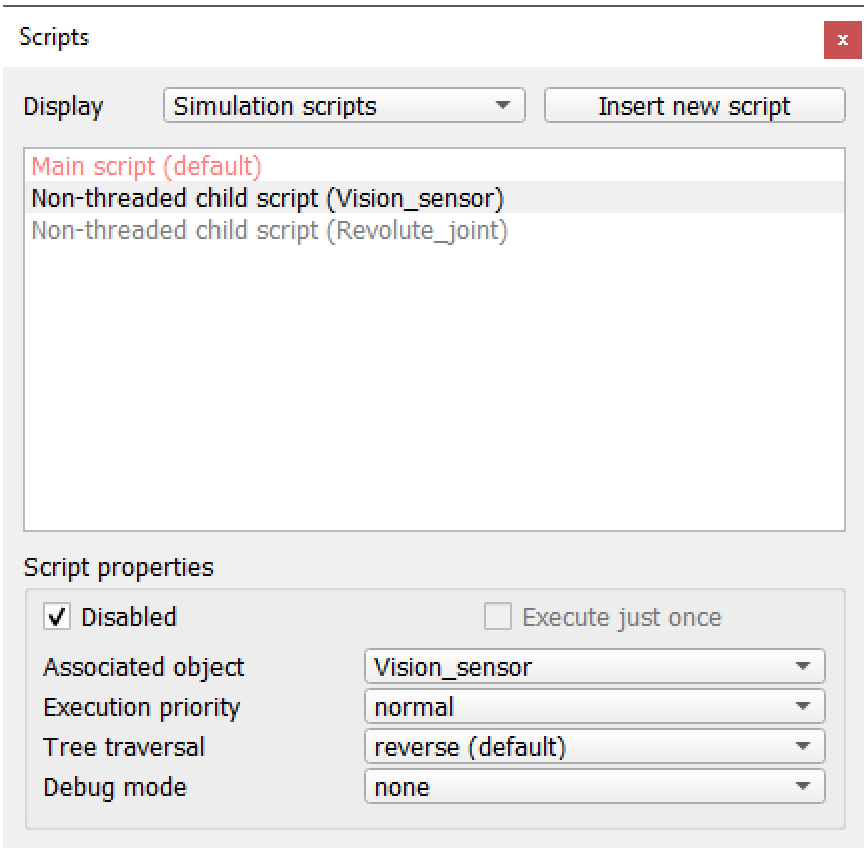
Figure 5:b0_resolver window after blueZeroDemo1 running with simulation stopped

The node ‘b0Node has left as a result of the following lua code:

```
function sysCall_cleanup()
    -- Shut down publisher and subscriber.
    if b0Node then
        simB0.nodeCleanup(b0Node)
        simB0.publisherDestroy(pub)
        simB0.subscriberDestroy(sub)
        simB0.nodeDestroy(b0Node)
    end
end
```

But the true utility of the B0RemoteApi is work from remote. For this we go to make the same from a Python script. The first thing is to disable the Vision Sensor script.

6- Disabling *Vision_sensor* script:



Scripts

Display: Simulation scripts | Insert new script

Main script (default)
Non-threaded child script (Vision_sensor)
Non-threaded child script (Revolute_joint)

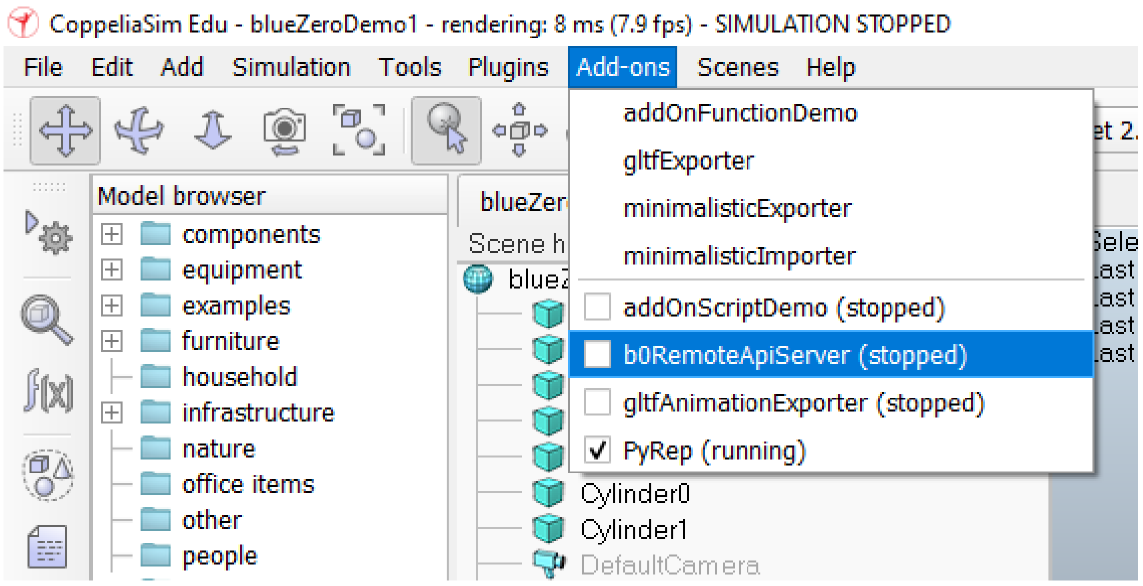
Script properties

☑ Disabled ☐ Execute just once

Associated object	Vision_sensor
Execution priority	normal
Tree traversal	reverse (default)
Debug mode	none

Figure 6: Vision_sensor script disabled

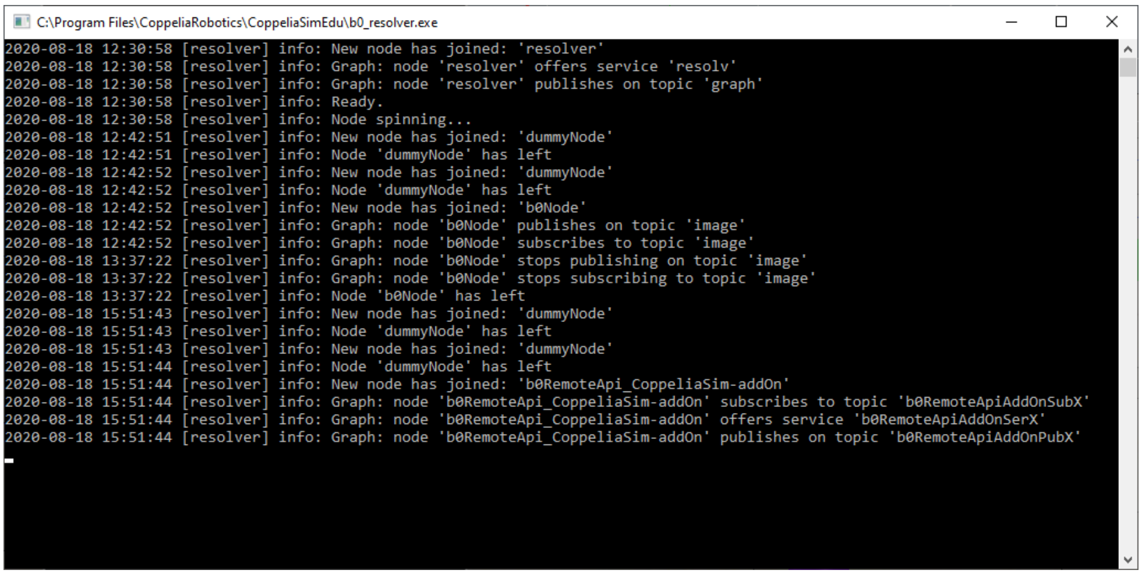
7- Enabling *b0RemoteApiServer*:



CoppeliaSim Edu - blueZeroDemo1 - rendering: 8 ms (7.9 fps) - SIMULATION STOPPED

Figure 7: Enabling boRemoteApiServer

We can see what *B0_resolver window* show now:



C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\b0_resolver.exe

```
2020-08-18 12:30:58 [resolver] info: New node has joined: 'resolver'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' offers service 'resolv'
2020-08-18 12:30:58 [resolver] info: Graph: node 'resolver' publishes on topic 'graph'
2020-08-18 12:30:58 [resolver] info: Ready.
2020-08-18 12:30:58 [resolver] info: Node spinning...
2020-08-18 12:42:51 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 12:42:51 [resolver] info: Node 'dummyNode' has left
2020-08-18 12:42:52 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 12:42:52 [resolver] info: Node 'dummyNode' has left
2020-08-18 12:42:52 [resolver] info: New node has joined: 'b0Node'
2020-08-18 12:42:52 [resolver] info: Graph: node 'b0Node' publishes on topic 'image'
2020-08-18 12:42:52 [resolver] info: Graph: node 'b0Node' subscribes to topic 'image'
2020-08-18 13:37:22 [resolver] info: Graph: node 'b0Node' stops publishing on topic 'image'
2020-08-18 13:37:22 [resolver] info: Graph: node 'b0Node' stops subscribing to topic 'image'
2020-08-18 13:37:22 [resolver] info: Node 'b0Node' has left
2020-08-18 15:51:43 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 15:51:43 [resolver] info: Node 'dummyNode' has left
2020-08-18 15:51:43 [resolver] info: New node has joined: 'dummyNode'
2020-08-18 15:51:44 [resolver] info: Node 'dummyNode' has left
2020-08-18 15:51:44 [resolver] info: New node has joined: 'b0RemoteApi_CoppeliaSim-addOn'
2020-08-18 15:51:44 [resolver] info: Graph: node 'b0RemoteApi_CoppeliaSim-addOn' subscribes to topic 'b0RemoteApiAddOnSubX'
2020-08-18 15:51:44 [resolver] info: Graph: node 'b0RemoteApi_CoppeliaSim-addOn' offers service 'b0RemoteApiAddOnSerX'
2020-08-18 15:51:44 [resolver] info: Graph: node 'b0RemoteApi_CoppeliaSim-addOn' publishes on topic 'b0RemoteApiAddOnPubX'
```

Figure 8: b0_resolver window after add-on script b0RemoteAipiServer started

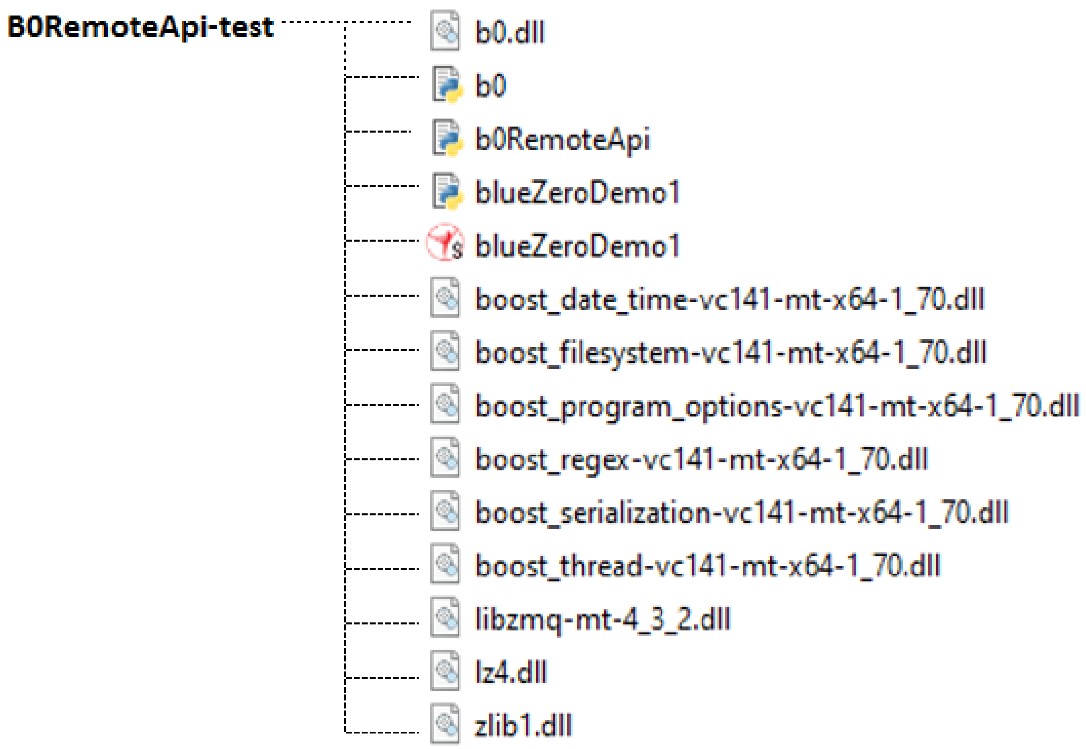
At message bar you can see too:

[b0RemoteApiServer@addOnScript:info]
[15:51:43] creating BlueZero node 'b0RemoteApi_CoppeliaSim-addOn'
and associated publisher, subscriber and service server
(on channel 'b0RemoteApiAddOn')
[CoppeliaSim:info] Started add-on script b0RemoteApiServer

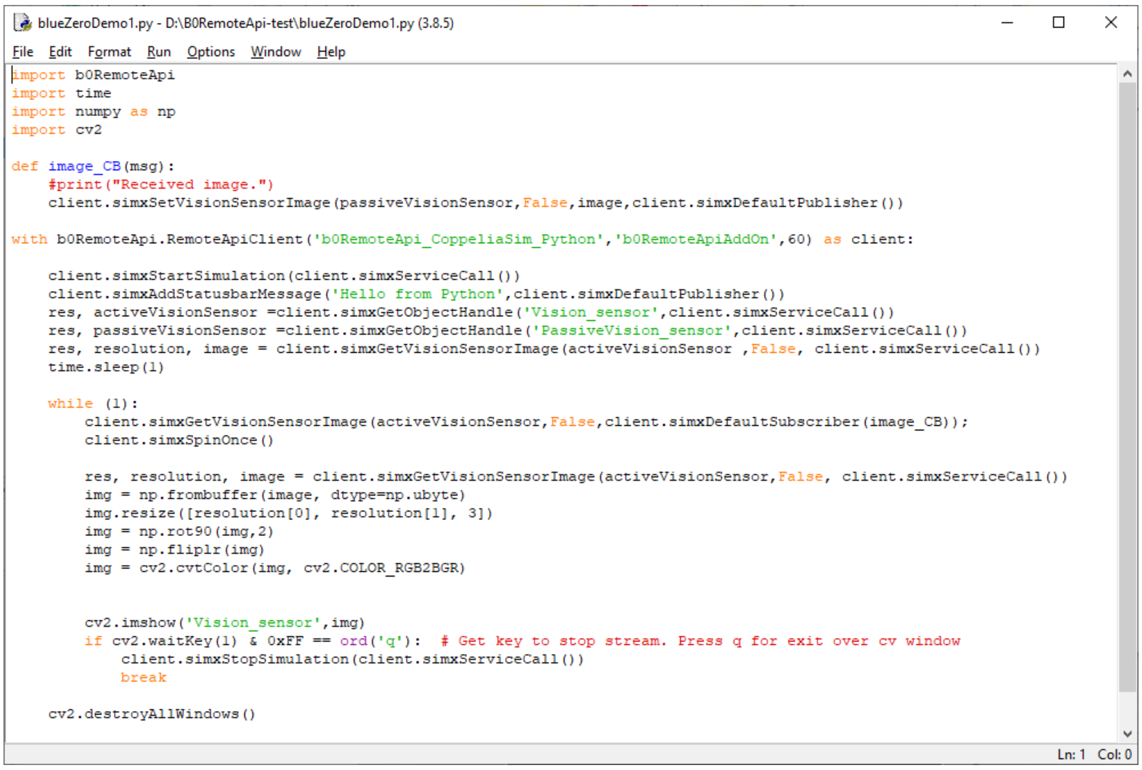
From here we have one node 'b0RemoteApi_CoppeliaSim-addOn' with associated **publisher**, **subscriber** and **service server** on channel 'b0RemoteApiAddOn'. This channel name is very important for create the client.

8- Client from Python

I you want, you can put in a directory, by instance, **B0RemoteApi-test**, the following files:

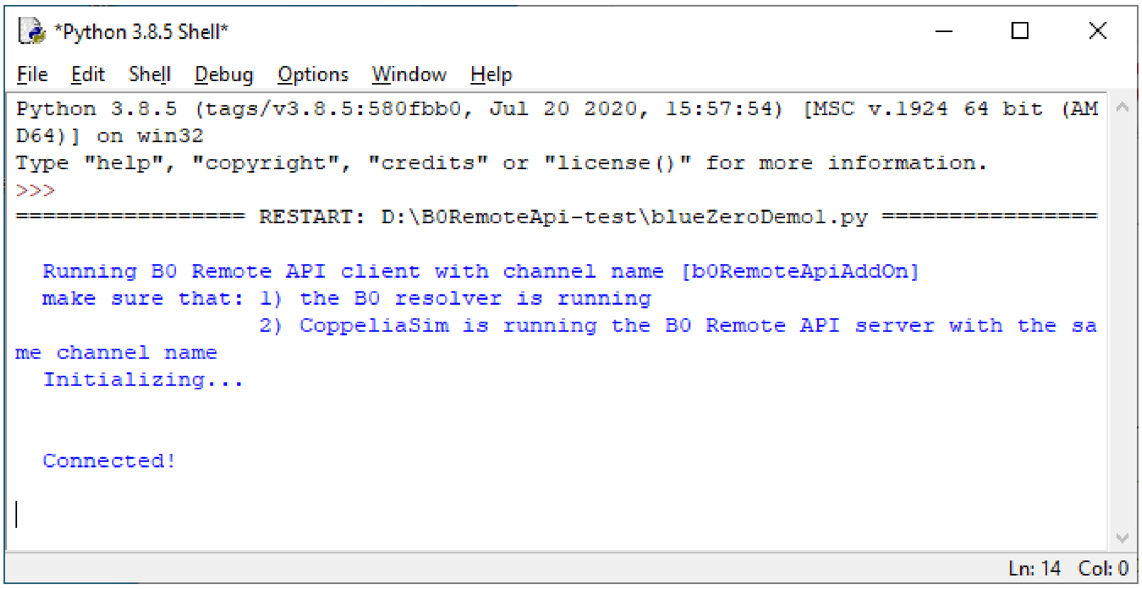

B0RemoteApi-test
b0.dll
b0
b0RemoteApi
blueZeroDemo1
blueZeroDemo1
boost_date_time-vc141-mt-x64-1_70.dll
boost_filesystem-vc141-mt-x64-1_70.dll
boost_program_options-vc141-mt-x64-1_70.dll
boost_regex-vc141-mt-x64-1_70.dll
boost_serialization-vc141-mt-x64-1_70.dll
boost_thread-vc141-mt-x64-1_70.dll
libzmq-mt-4_3_2.dll
lz4.dll
zlib1.dll

The Python code blueZeroDemo1.py inspired in simpleTest.cpp:


blueZeroDemo1.py - D:\B0RemoteApi-test\blueZeroDemo1.py (3.8.5)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```python
import b0RemoteApi
import time
import numpy as np
import cv2

def image_CB(msg):
    #print("Received image.")
    client.simxSetVisionSensorImage(passiveVisionSensor,False,image,client.simxDefaultPublisher())

with b0RemoteApi.RemoteApiClient('b0RemoteApi_CoppeliaSim_Python','b0RemoteApiAddOn',60) as client:

    client.simxStartSimulation(client.simxServiceCall())
    client.simxAddStatusbarMessage('Hello from Python',client.simxDefaultPublisher())
    res, activeVisionSensor =client.simxGetObjectHandle('Vision_sensor',client.simxServiceCall())
    res, passiveVisionSensor =client.simxGetObjectHandle('PassiveVision_sensor',client.simxServiceCall())
    res, resolution, image = client.simxGetVisionSensorImage(activeVisionSensor ,False, client.simxServiceCall())
    time.sleep(1)

    while (1):
        client.simxGetVisionSensorImage(activeVisionSensor,False,client.simxDefaultSubscriber(image_CB));
        client.simxSpinOnce()

        res, resolution, image = client.simxGetVisionSensorImage(activeVisionSensor,False, client.simxServiceCall())
        img = np.frombuffer(image, dtype=np.ubyte)
        img.resize([resolution[0], resolution[1], 3])
        img = np.rot90(img,2)
        img = np.fliplr(img)
        img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)


        cv2.imshow('Vision_sensor',img)
        if cv2.waitKey(1) & 0xFF == ord('q'):  # Get key to stop stream. Press q for exit over cv window
            client.simxStopSimulation(client.simxServiceCall())
            break

    cv2.destroyAllWindows()
```

Now Run module blueZeroDemo1.py at Python shell:



```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: D:\B0RemoteApi-test\blueZeroDemo1.py ================

  Running B0 Remote API client with channel name [b0RemoteApiAddOn]
  make sure that: 1) the B0 resolver is running
                  2) CoppeliaSim is running the B0 Remote API server with the sa
me channel name
  Initializing...


  Connected!

```

Figure 9: Python shell window with messages from b0RemoteApi.py

At the CoppeliaSim will be show the same as before in step 2 plus a little window generate by OpenCv:

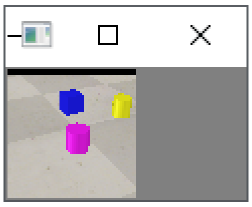


Figure 10: Window from OpenCv code

If you make click over this window and press the 'q' key, the window will be closed and the simulation stopped.