**How to connect ROS with the Unity Simulation**

**Previous notes:**

If you are using Windows, you need to set up a virtual machine to use the ROS environment for that. We recommend using VM VirtualMachine and installing Ubuntu 20.04. If your main OS is already Ubuntu 20.04 you don't need to use it.

If you have already installed Ubuntu check your version:

| lsb\_release -a |
| --- |

Depending on your version you need to install the correct [distribution](http://wiki.ros.org/ROS/Installation) of ROS, in our case with Ubuntu 20.04 we installed ROS Noetic.

Once you install Ubuntu 20.04 , [ROS Noetic](http://wiki.ros.org/noetic/Installation/Ubuntu) and [ROS environment](http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment)go to the settings of your virtual machine > Network > Attached to: Bridged Adapter.

In the folder you have an .ova with all installed, you can use it on VirtualBox if you want.

**Step 1 Cloning the Endpoint repository**

For connecting ROS with Unity the ROS-TCP-Endpoint has to be running on your computer, so in your catkin\_ws/src clone the next repository:

<https://github.com/Unity-Technologies/ROS-TCP-Endpoint/tree/v0.7.0>

Once cloned, use catkin\_make for building the new package, then don’t forget to source your workspace **source** ~/catkin\_ws/devel/setup.bash

Go to /catkin\_ws/src/ROS-TCP-Endpoint-0.7.0/src/ros\_tcp\_endpoint and edit the first line of default\_server\_endpoint.py substituting python for python3.

**Step 2 Downloading the Unity Simulator**

Download the simulator file, there is compressed and uncompressed versions.

**Step 3 Setting up the environment variable**

Use the next command to find your IP and remember it because you will use it in the next steps:

| hostname -I |
| --- |

In this step you need to modify the ./bashrc file, for that use the next commands:

| **cd** gedit ./.bashrc |
| --- |

The end of the file should look like this:

| #export ROS\_MASTER\_URI=http://YOUR\_IP:11311 - example: **export** ROS\_MASTER\_URI=http://192.168.1.100:11311 #export ROS\_IP=/YOUR\_IP - example: **export** ROS\_IP=192.168.1.100  **source** /opt/ros/noetic/setup.bash #source ~/YOUR\_CATKIN\_ROUTE/devel/setup.bash - example: **source** ~/catkin\_ws/devel/setup.bash |
| --- |

To apply the changes utilize the next command in each terminal you are using or close it and open it again.

| **source** ./.bashrc |
| --- |

**Step 4 Launching the Endpoint file**

To launch the End point use the next command:

| roslaunch ros\_tcp\_endpoint endpoint.launch |
| --- |

Or you can also drive through the folders and launch the file from the launch folder.

| **cd** catkin\_ws/src/ROS-TCP-Endpoint/launch roslaunch endpoint.launch |
| --- |

**Step 5 Modifying the Json file with your IP your PORT**

On the root directory of the simulator go to the folder mir\_competition\_simulation\_linux\_Data on Linux or mir\_competition\_sim\_Data on Windows, next to the folder StreamingAssets and edit Configuration\_Parameters.json.

Change the ROS\_IP\_ADDRESS to your own IP.

Use the next command to find your IP:

| hostname -I |
| --- |

**Step 6 Run the Unity simulation**

On Windows run mir\_competition\_sim.exe.

On Linux right click mir\_competition\_sim\_.x86\_64 > Properties > Permissions and check the box ‘Allow executing file as program’ , then run mir\_competition\_sim\_.x86\_64.

.

**Step 7 How to move the view of the simulation**

It is possible to move the camera and rotate it in the X axis. You can also switch the camera between the camera and the BlueRov camera. If you want to close the simulation , just press the key **Esc**.

Movement Controllers:

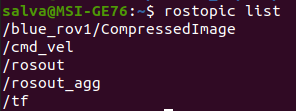
* **W**: Forward
* **S**: Back
* **A**: Left
* **D**: Right
* **Q**: Up
* **E**: Down
* **Space:** Rotate the camera 180º

Switch camera:

* **Tab**: Switch camera, you can still move the camera.

**Step 8 ROS topics**

Once you launch the end point and run the unity simulator you can use the rostopic list command for identifying the topics.First you have topic call CompressedImage, with this one you can use python3 and OpenCV or other library for visualizing the BlueRov camera, finally you have the cmd\_vel topic, where you can publish the linear and angular velocity of the robot after the calculations.

****

*Image 1 -Rostopic list*