TÉCNICO LISBOA

Institute for Systems and Robotics | LISBOA

ROS 2 Workshop

Marcelo Jacinto
(mjacinto@isr.tecnico.ulisboa.pt)

LARSyS
Laboratory of Robotics and Engineering Systems

Motivation

ROS 2 is useful if you ever dream of ruling the world with an army of drones...

Why ROS 2

2 > 1

What is ROS 2

Not an operating system! (they use the name as clickbait)

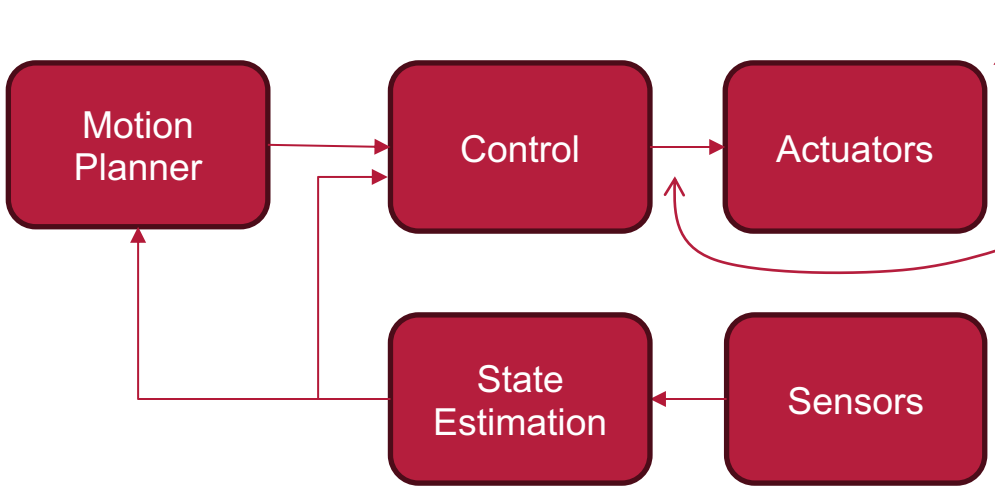
"The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications."

ROS 2 provides:

- a set of useful programs that run in your computer
- a communication system for programs to share data with each other

Communication Structure

Problem Formulation/Motivation: You want to design a Guidance, Control and Navigation (GNC) system for a drone, such that each block of the system is implemented as an individual program (written in Python or C++).

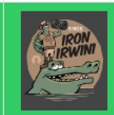


Each program is called a Node

Nodes send information to each other through topics

Information is sent/published through topics using messages

Installing ROS 2

Distro	Release date	Logo	EOL date
Iron Irwini	May 23rd, 2023		November 2024
Humble Hawksbill	May 23rd, 2022		May 2027
Galactic Geochelone	May 23rd, 2021		December 9th, 2022
Foxy Fitzroy	June 5th, 2020		June 20th, 2023

Follow the steps on the links bellow:

Ubuntu 22.04LTS

Official Documentation:
https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

Ubuntu 20.04LTS

Official Documentation:
https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html

Installation tutorials on YouTube: https://youtu.be/0aPbWsyENA8?t=122&si=CbwtHPBpd0aOQdI-

After Installing ROS 2

Run the following line on the terminal

```
source /opt/ros/humble/setup.bash >> ~/.bashrc
```

foxy (if using Ubuntu 20.04LTS)

Tell your terminal where ROS 2 is installed

Also install colcon build tools

```
sudo apt install python3-colcon-common-extensions
```

```
echo "source /usr/share/colcon_argcomplete/hook/colcon-argcomplete.bash" >> ~/.bashrc
echo "source /usr/share/colcon_cd/function/colcon_cd.sh" >> ~/.bashrc
echo "export _colcon_cd_root=/opt/ros/humble/" >> ~/.bashrc
```

foxy (if using Ubuntu 20.04LTS)

Convenience tools such as auto-complete in the terminal

Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Colcon-Tutorial.html

Getting Started

Create a folder/workspace where the code will live.

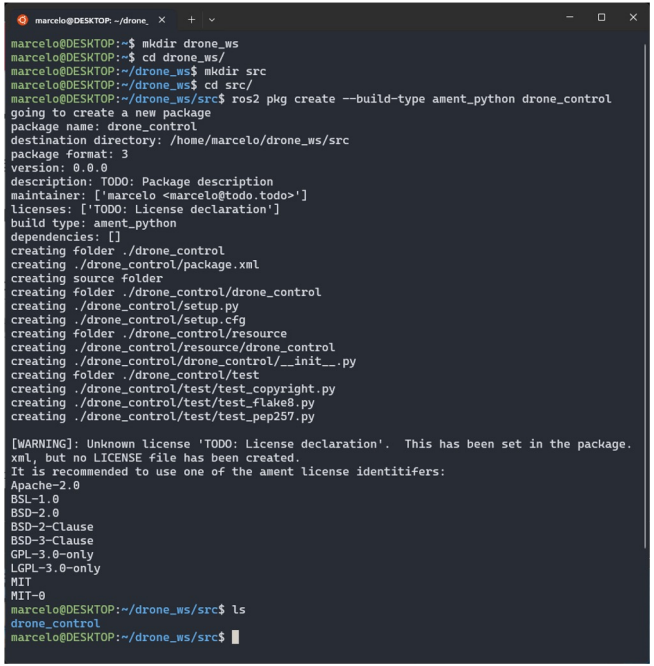
```
mkdir drone_ws
```

Inside the workspace create a src folder/directory

```
cd drone_ws
mkdir src
```

Inside the src directory create a Python code package

```
cd src
ros2 pkg create --build-type ament_python drone_control
```

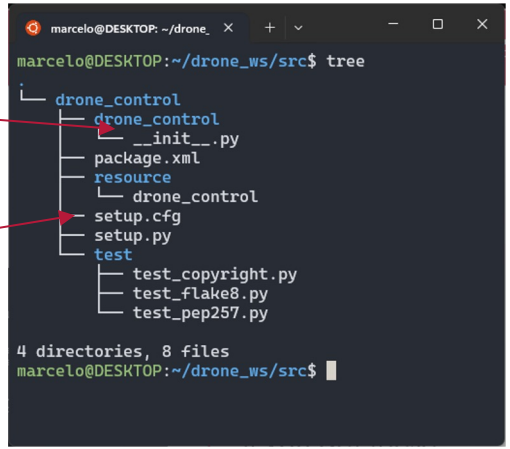


Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html

Folder/Package Structure (Python)

Write your code here

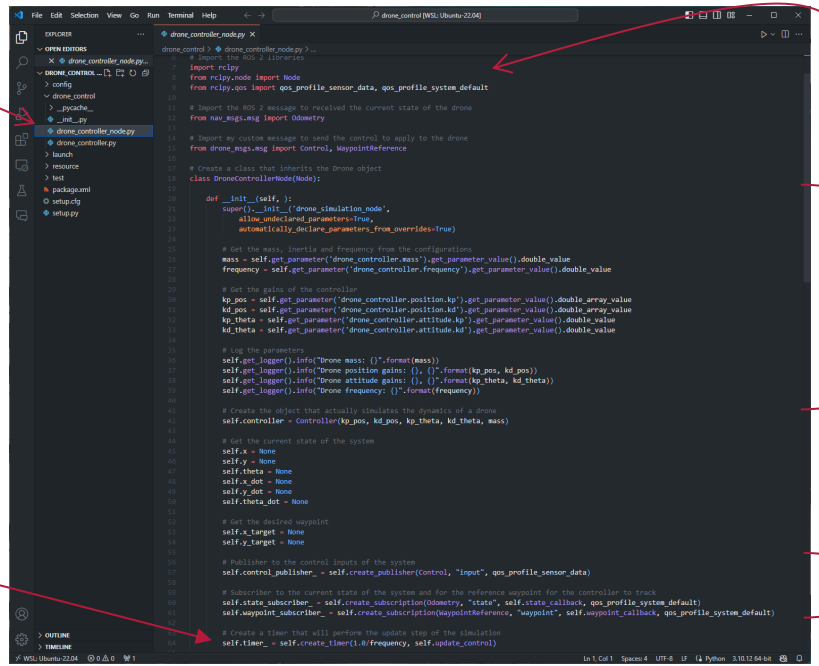
Inform ROS 2 where your “main” is in the setup.py file



https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html

Creating a Node

Python file for node + Python file for logic



ROS 2 library imports

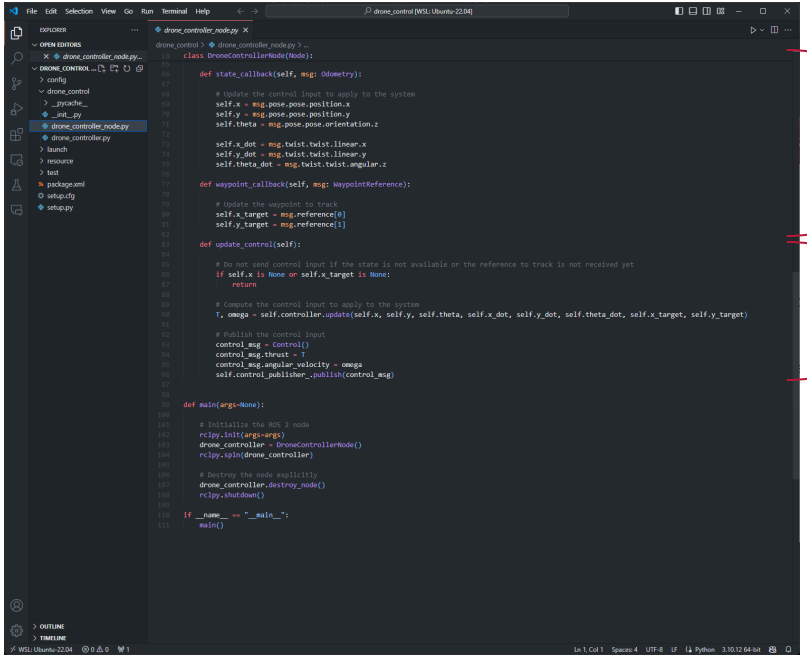
Reading parameters dynamically

Timer to periodically compute the control signal

Creating the publishers/subscribers

Institute for Systems and Robotics | LISBOA

Creating a Node (2)

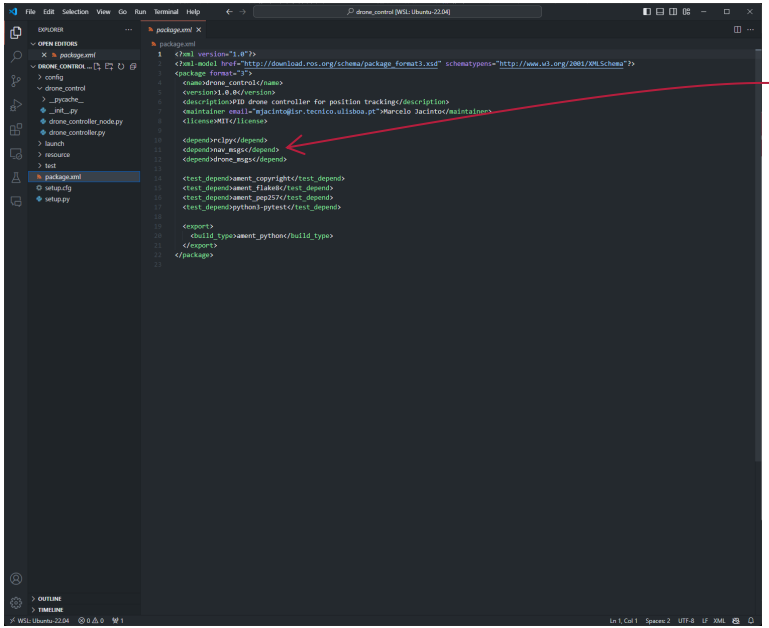


Subscriber callbacks are called when a new message arrives

Function called by the timer to compute and publish the control signal

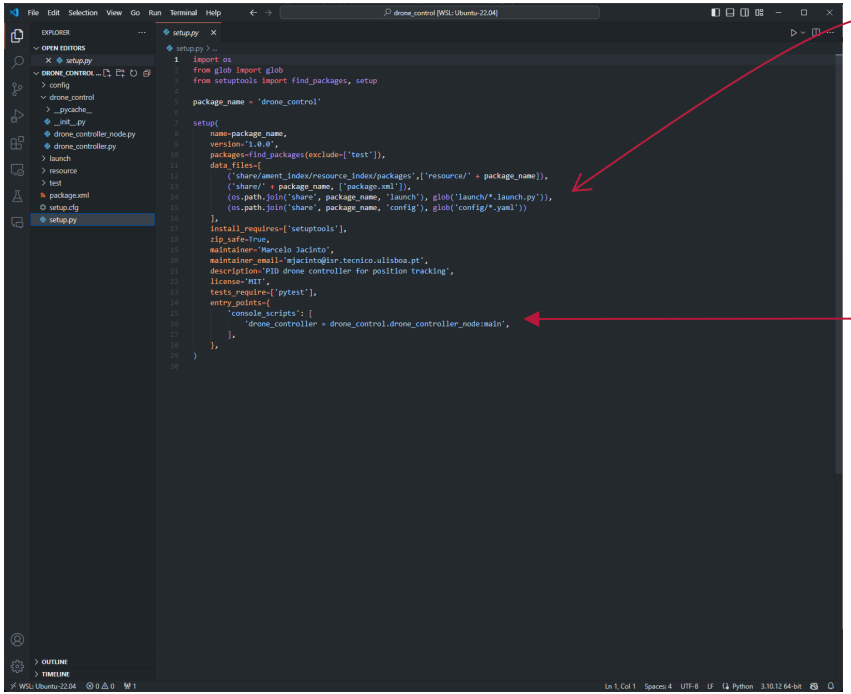
TÉCNICO LISBOA

Creating a Node (3)



Add the used packages to the package.xml file

Creating a Node (4)



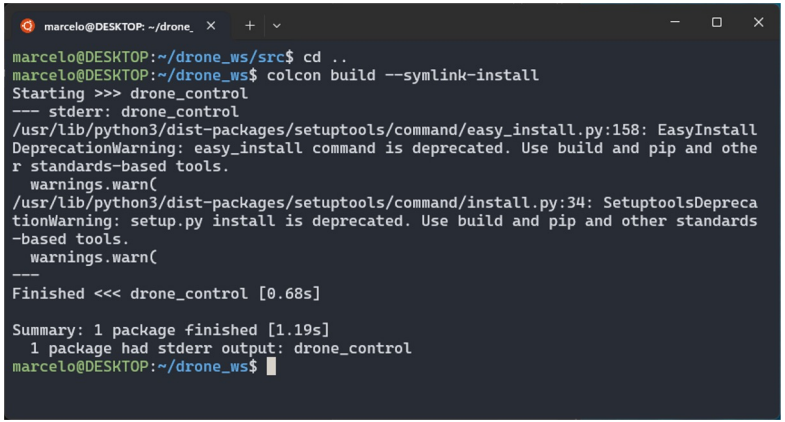
Index configuration and launch directories here

Setup your python executable nodes here

Compiling/Indexing the Package

To compile all the packages, run (at the workspace level):

```
colcon build --symlink-install
```



If you get these warnings, just run the following lines

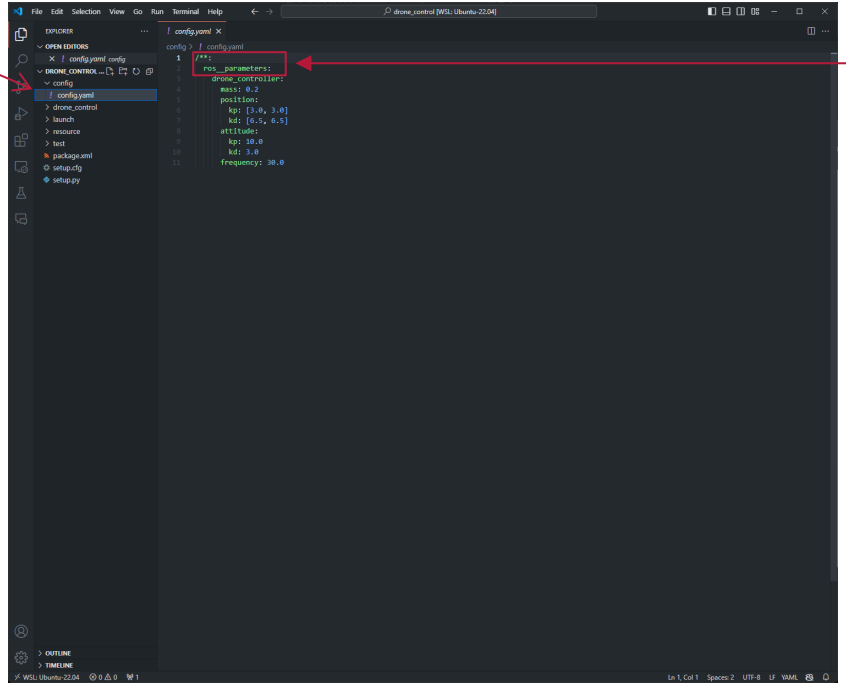
```
PYTHONWARNINGS="ignore:setup.py install is deprecated::setuptools.command.install" >> ~/.bashrc
export PYTHONWARNINGS >> ~/.bashrc
```


Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html

Institute for Systems and Robotics | LISBOA

Dynamic Parameters

Create a configuration directory with a YAML file



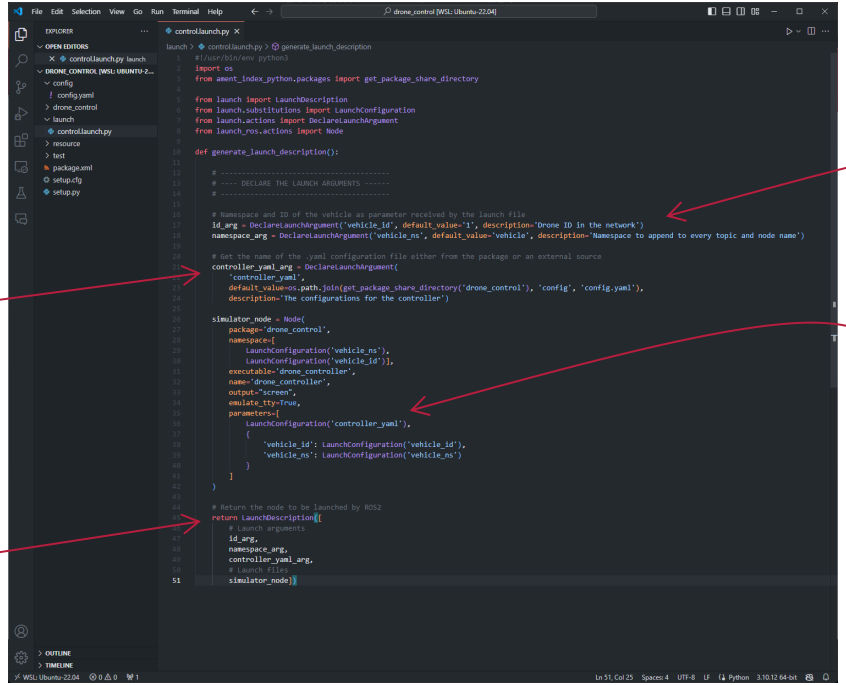
Use /** such that all the parameters are available to all nodes

Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Parameters/Understanding-ROS2-Parameters.html

Launch Files

Specify the yaml file that contains parameters

Return the node and arguments to be evaluated at runtime



Define some parameters directly in the launch file

Inject the parameters from the yaml file in the node

Compiling/Indexing the Package (2)

To compile all the packages, run (at the workspace level):

```
colcon build --symlink-install
```

Important! Otherwise, every time a launch file or configuration files is changed, you must re-compile the code!

To compile a specific package, run (at the workspace level):

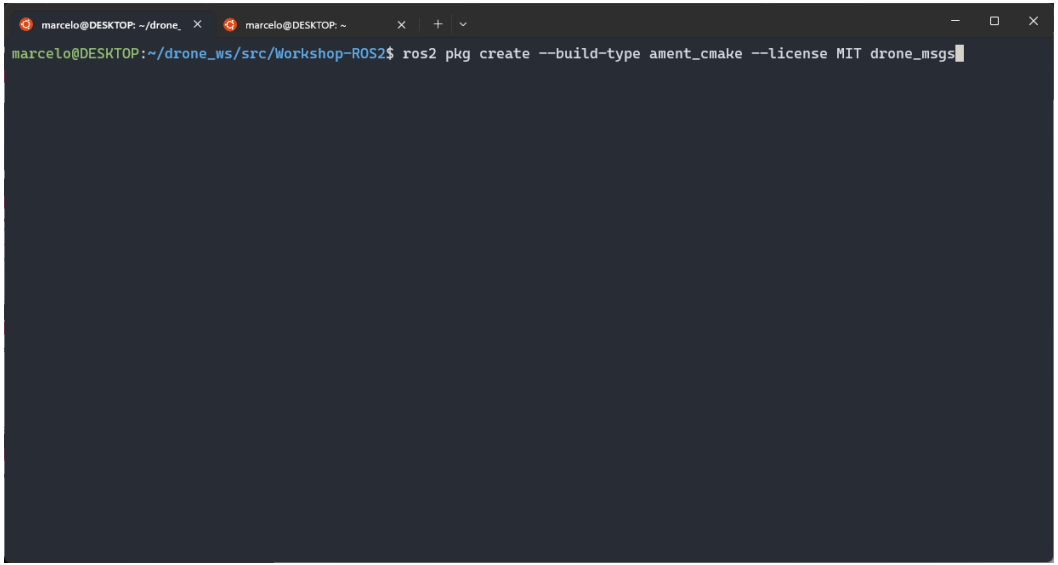
```
colcon build --symlink-install --packages-select <my_package_name>
```

To clear all the compiled packages, run (at the workspace level):

```
rm -rf install build install
```


Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html

Creating a Messages Package

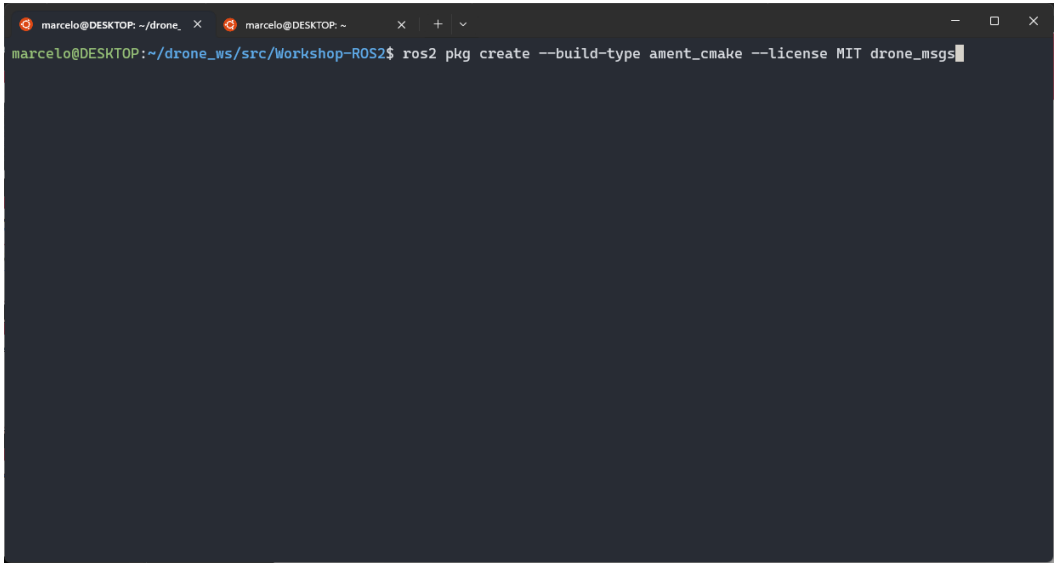


ros2 pkg create –build-type ament_cmake --license MIT drone_msgs

Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Custom-ROS2-Interfaces.html

Institute for Systems and Robotics | LISBOA

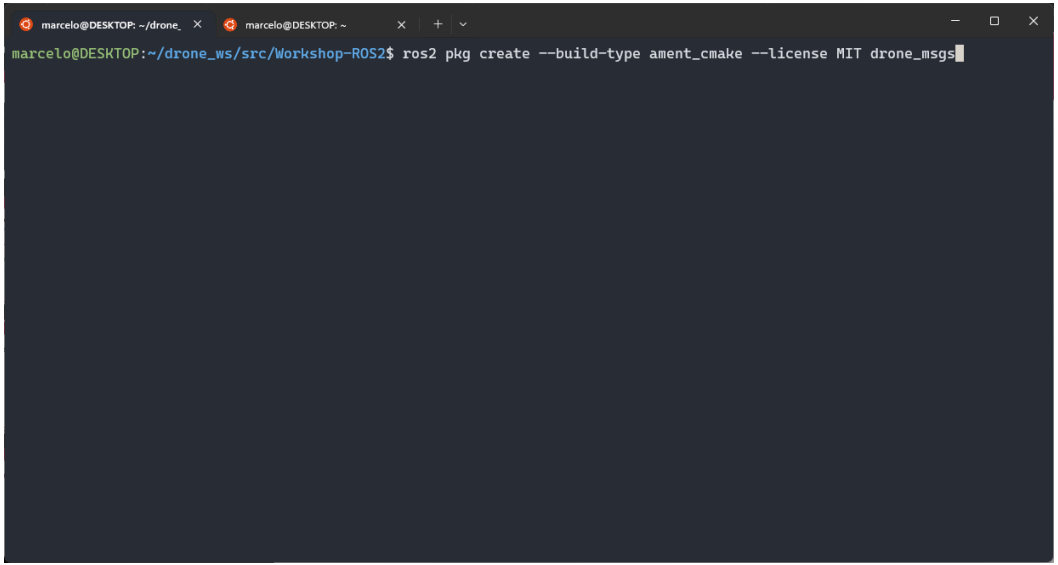
Creating a Messages Package



Ros2 pkg create –build-type ament_cmake --license MIT drone_msgs

Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Custom-ROS2-Interfaces.html

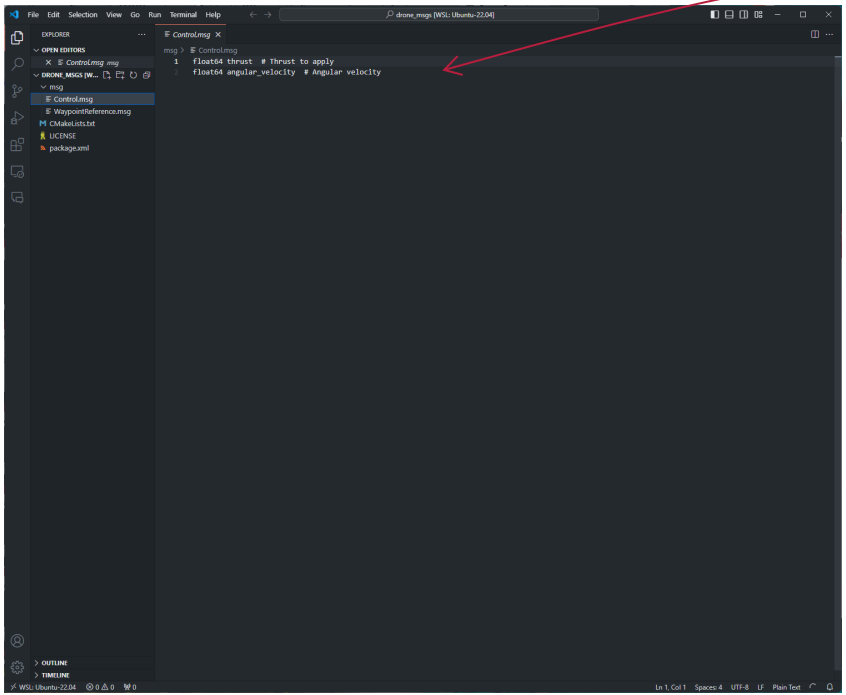
Creating a Messages Package



Ros2 pkg create –build-type ament_cmake --license MIT drone_msgs

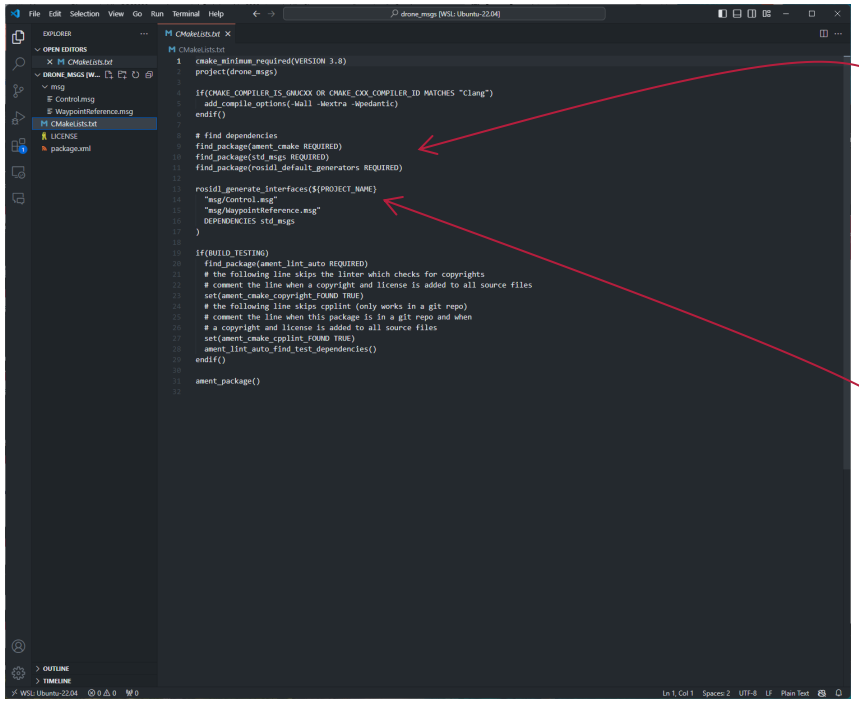
Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Custom-ROS2-Interfaces.html

Messages Provided by ROS 2



Message definition

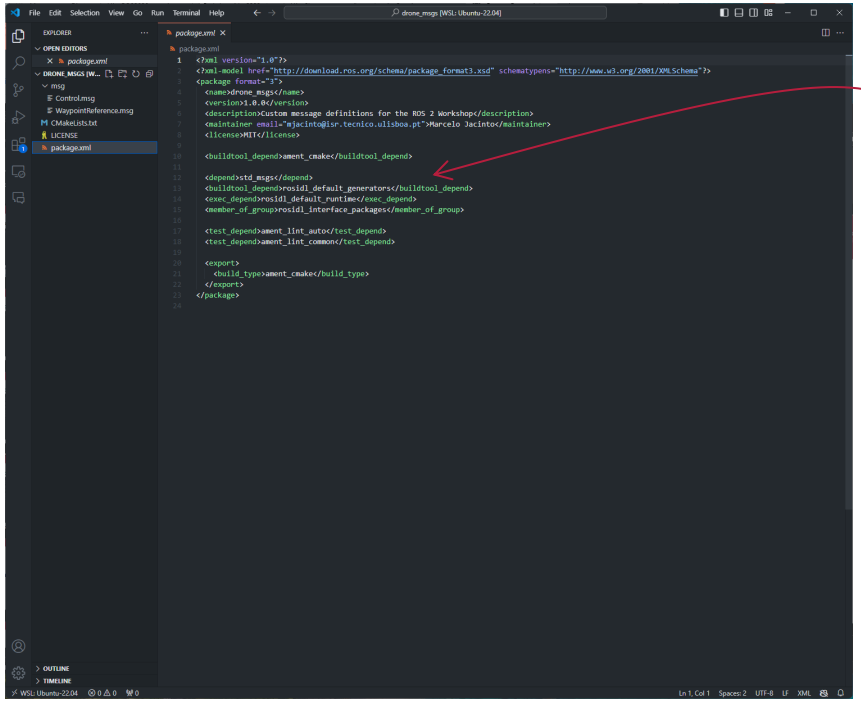
Messages Provided by ROS 2



Import the base packages that the messages will use

List the messages to be compiled

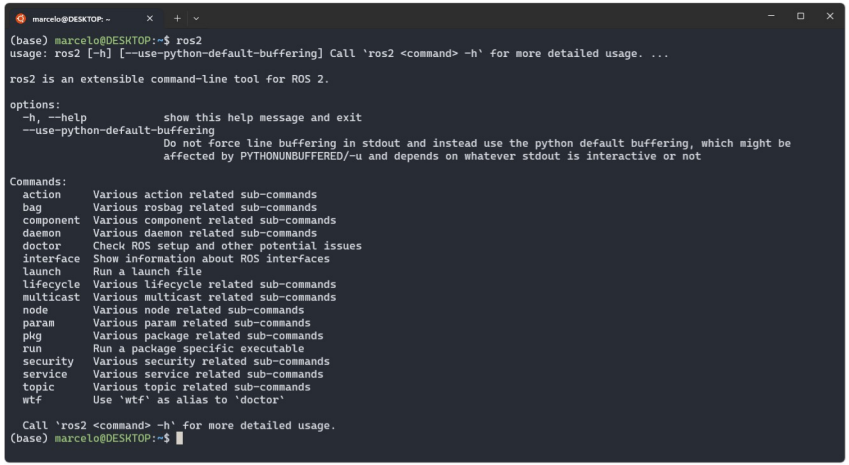
Messages Provided by ROS 2



Import the base packages that the messages will use

Terminal Tools

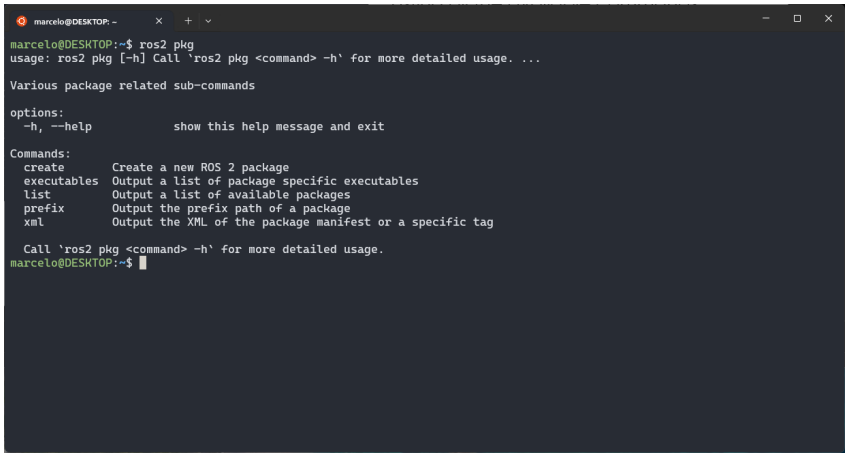
Terminal Tools



Listing all the available commands:

ros2

Terminal Tools

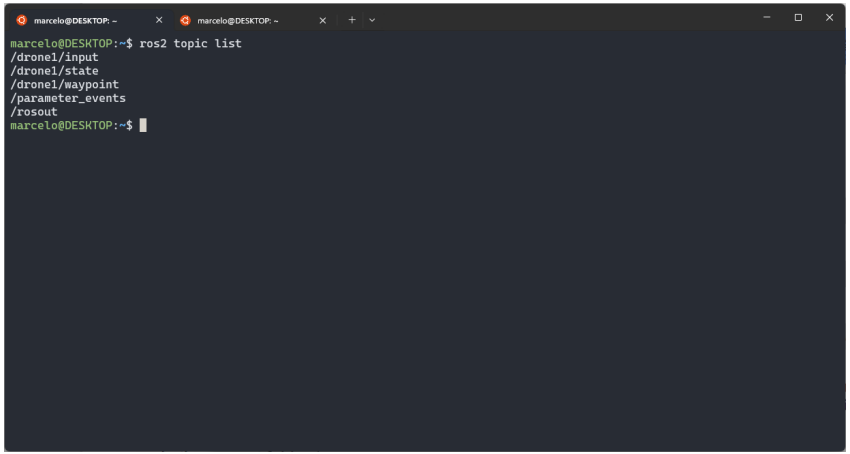


Each command can have multiple options. For example:

`ros2 pkg`

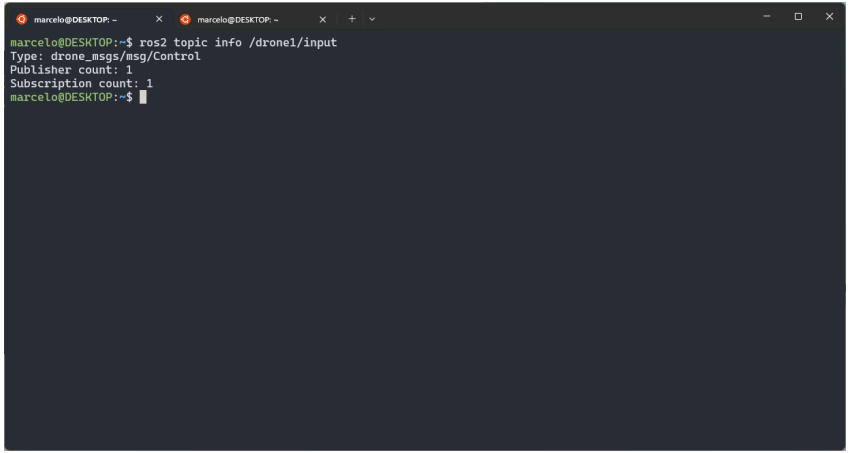
Institute for Systems and Robotics | LISBOA

List all Topics Published/Subscribed



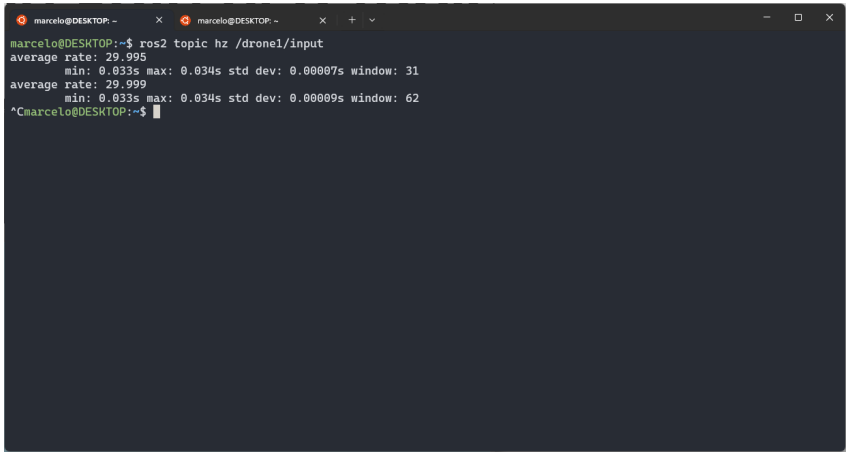
ros2 topic list

Get Information About a Topic



ros2 topic info /<name_of_topic>

Get the Rate of a Topic

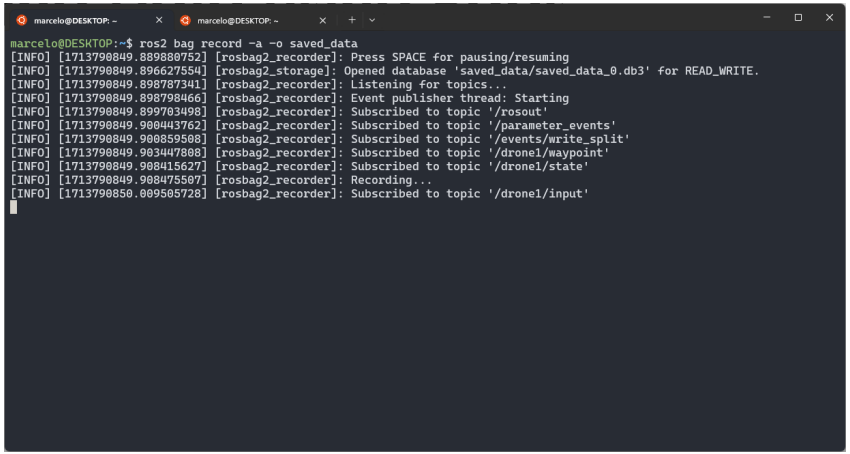


ros2 topic hz /<name_of_topic>

Get all the Nodes Running

ros2 node list

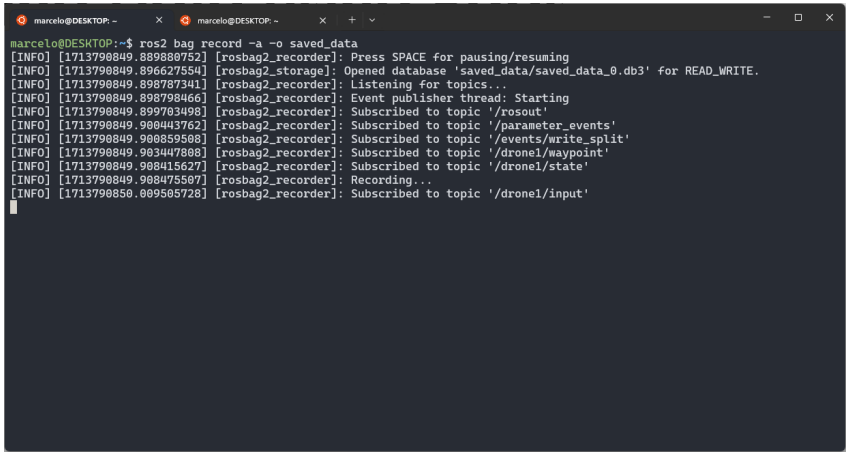
Saving Data using Bags



ros2 bag record -a –o <bag_name>

Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Recording-And-Playing-Back-Data/Recording-And-Playing-Back-Data.html

Playing Back the Bag Data

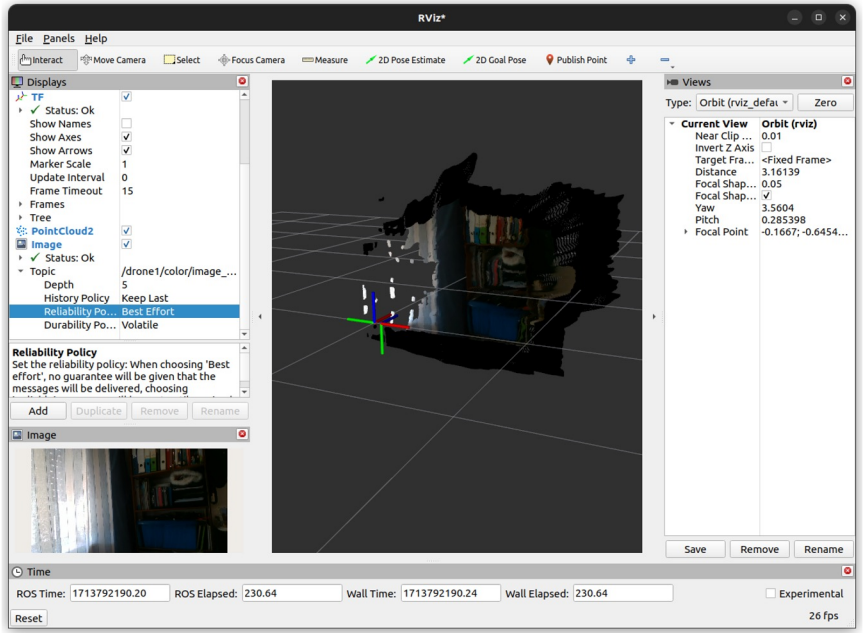


ros2 bag play <bag_name>

Official Documentation: https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Recording-And-Playing-Back-Data/Recording-And-Playing-Back-Data.html

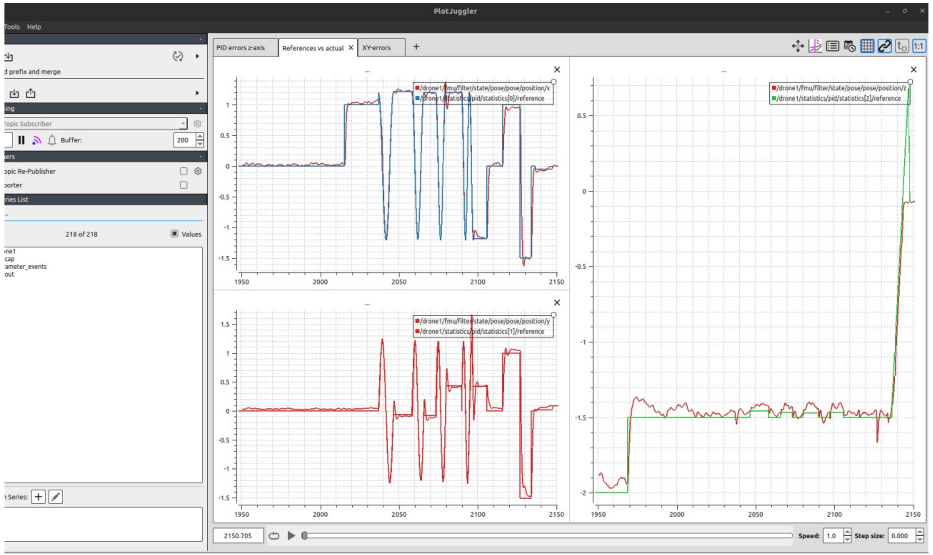
Visualization Tools

RVIZ 2



Executing: `rviz2`

Plotjuggler



Installing:

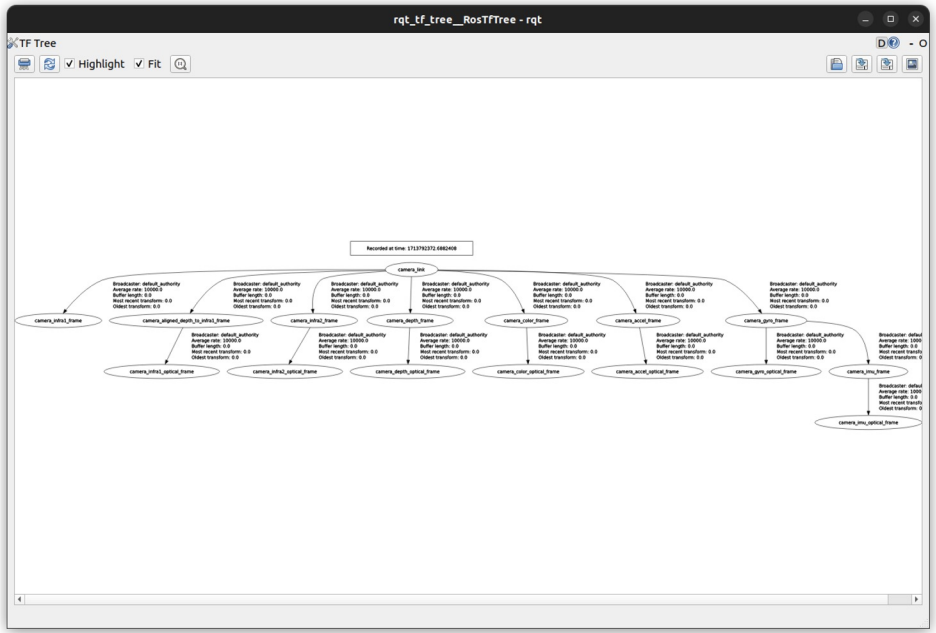
```
sudo apt install ros-humble-plotjuggler
```

Executing:

```
ros2 run plojuggler plotjuggler
```

Official Documentation: https://plotjuggler.io

Transforms (TF2)

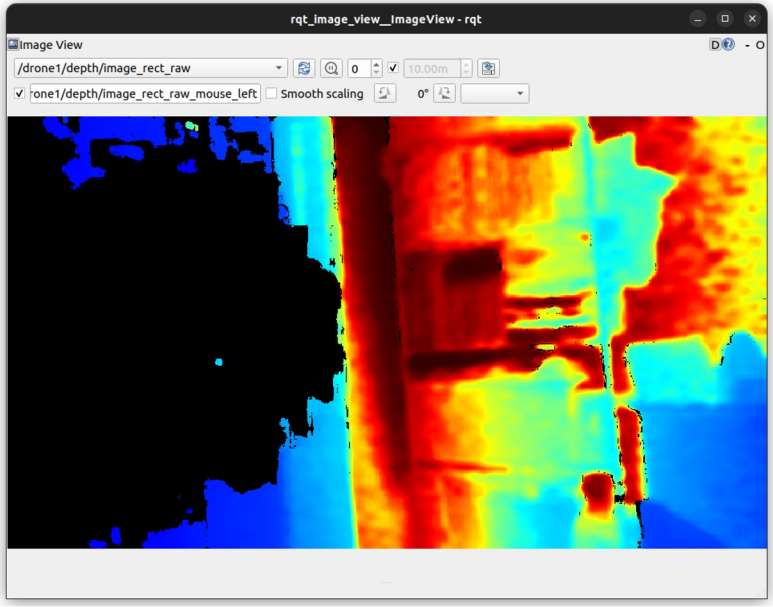


Executing:

```
ros2 run rqt_tf_tree rqt_tf_tree
```


Official Documentation: https://docs.ros.org/en/humble/Concepts/About-RQt.html

RQT Image View

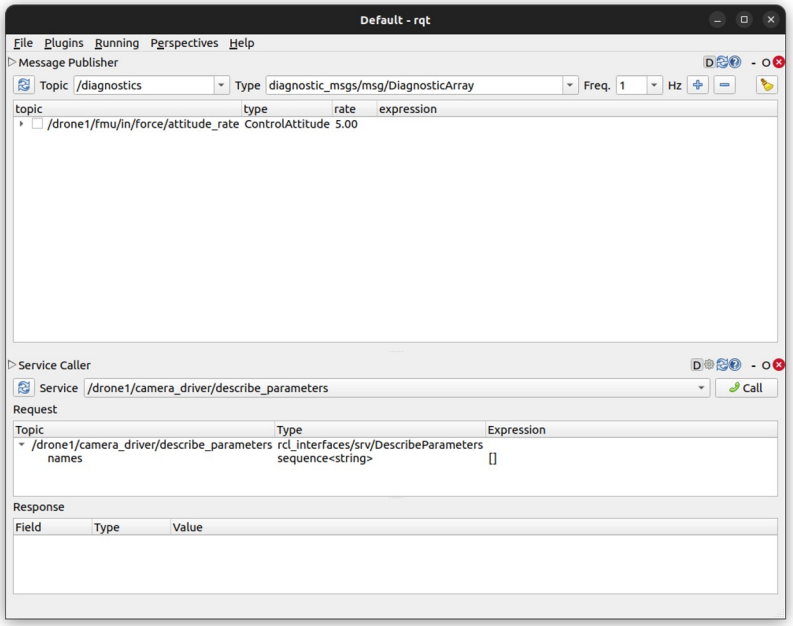


Executing:

```
ros2 run rqt_image_view rqt_image_view
```


Official Documentation: https://docs.ros.org/en/humble/Concepts/About-RQt.html

RQT GUI



Executing:

```
ros2 run rqt_gui rqt_gui
```


Official Documentation: https://docs.ros.org/en/humble/Concepts/About-RQt.html

ROS ID

- By default, all the ROS 2 nodes run with domain ID 0
- To avoid interference between different groups of computers running ROS 2 on the same network, a different domain ID should be set for each group

```
export ROS_DOMAIN_ID=0
```

- By default, all the ROS 2 nodes broadcast to the local network

```
export ROS_LOCALHOST_ONLY=0
```


Official Documentation: https://docs.ros.org/en/humble/Concepts/Intermediate/About-Domain-ID.html

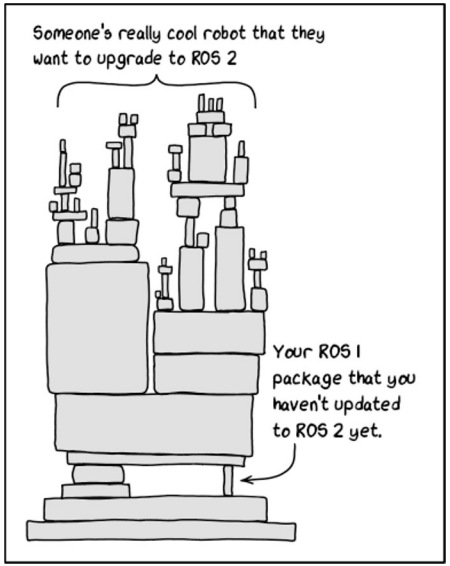
Installing External Libraries

Installing packages follows this standard:

`sudo apt install ros-<version>-<name-of-package>`

```
marcelo@DESKTOP:~$ sudo apt install ros-humble-vision-msgs
[sudo] password for marcelo:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cuda-12-0 cuda-cccl-12-0 cuda-command-line-tools-12-0 cuda-compiler-12-0 cuda-cudart-12-0 cuda-cudart-dev-12-0
  cuda-cuobjdump-12-0 cuda-cupti-12-0 cuda-cupti-dev-12-0 cuda-cuxxfilt-12-0 cuda-demo-suite-12-0
  cuda-documentation-12-0 cuda-driver-dev-12-0 cuda-gdb-12-0 cuda-libraries-12-0 cuda-libraries-dev-12-0
  cuda-nsight-12-0 cuda-nsight-compute-12-0 cuda-nsight-systems-12-0 cuda-nvcc-12-0 cuda-nvdisasm-12-0
  cuda-nvml-dev-12-0 cuda-nvprof-12-0 cuda-nvprune-12-0 cuda-nvrtc-12-0 cuda-nvrtc-dev-12-0 cuda-nvtx-12-0
  cuda-nvvp-12-0 cuda-opencl-12-0 cuda-opencl-dev-12-0 cuda-profiler-api-12-0 cuda-runtime-12-0 cuda-sanitizer-12-0
  cuda-toolkit-12-0 cuda-toolkit-12-0-config-common cuda-tools-12-0 cuda-visual-tools-12-0 gds-tools-12-0
  libcublas-12-0 libcublas-dev-12-0 libcufft-12-0 libcufft-dev-12-0 libcufile-12-0 libcufile-dev-12-0 libcurand-12-0
  libcurand-dev-12-0 libcusolver-12-0 libcusolver-dev-12-0 libcusparse-12-0 libcusparse-dev-12-0 libllvm13 libnpp-12-0
  libnpp-dev-12-0 libnvjitlink-12-0 libnvjitlink-dev-12-0 libnvjpeg-12-0 libnvjpeg-dev-12-0 libnvvm-samples-12-0
  nsight-compute-2022.4.0 nsight-compute-2022.4.1 nsight-systems-2022.4.2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  ros-humble-vision-msgs
0 upgraded, 1 newly installed, 0 to remove and 9 not upgraded.
Need to get 150 kB of archives.
After this operation, 2,105 kB of additional disk space will be used.
0% [Connecting to packages.ros.org]
```

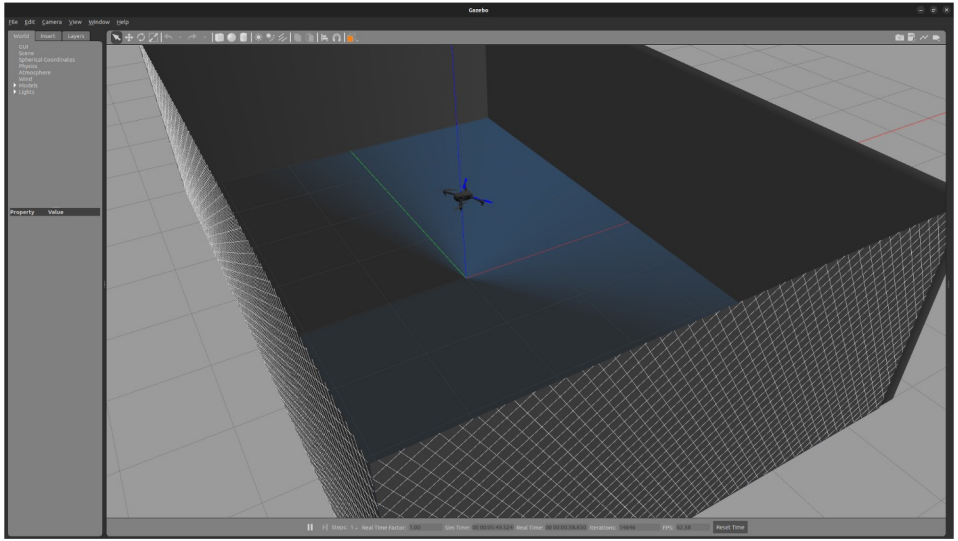
Example for the vision_msgs package



Official list of packages: https://index.ros.org/packages/

3D Simulators

Simulators (Gazebo Classic)

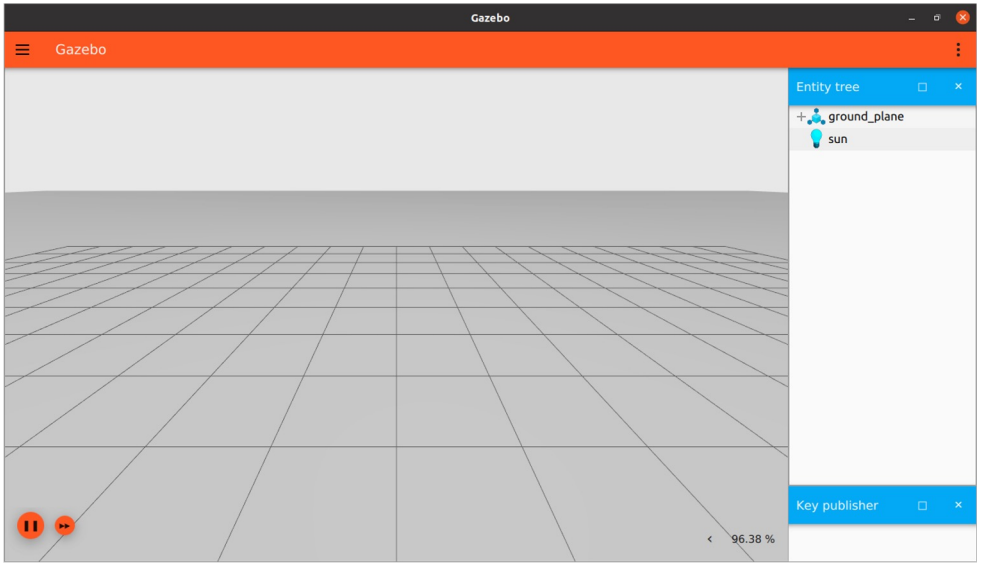


Installing

```
sudo apt install ros-humble-gazebo-ros-pkgs
```


Official Documentation: https://classic.gazebosim.org/tutorials?tut=ros2_installing#TestingGazeboandROS2integration

Simulators (Gazebo Garden)

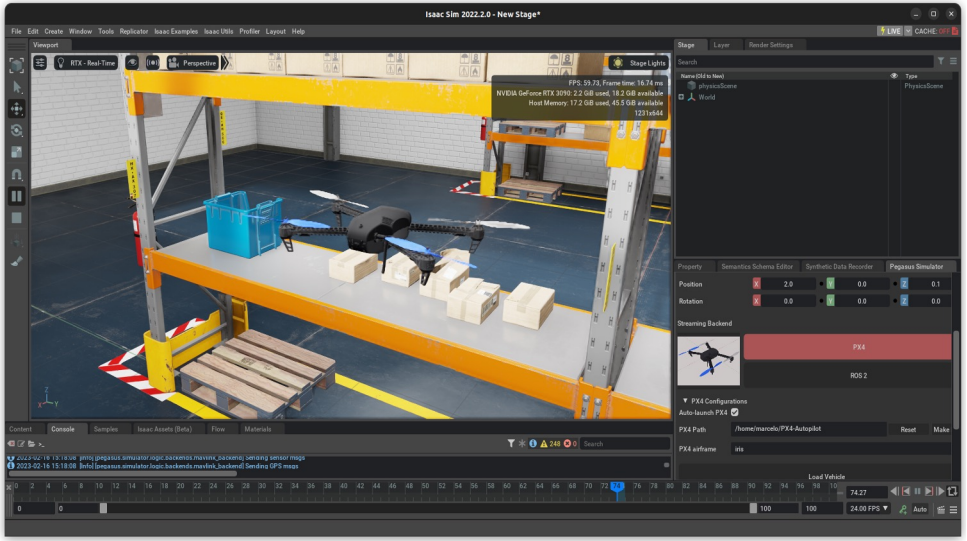


Installing

```
sudo apt-get install ros-humble-ros-gz
```


Official Documentation: https://gazebosim.org/docs/garden/ros_installation

Simulators (Pegasus Simulator)



Official Documentation: https://pegasussimulator.github.io/PegasusSimulator/

Institute for Systems
and Robotics | LISBOA

TÉCNICO
LISBOA