**ROS**

source /opt/ros/iron/setup.bash

ros2 <command>

**Some useful commands:**

1. action:

1.1. info: shows info about an action

Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 action info send\_goal  
Action: send\_goal  
Action clients: 0

Action servers: 0

1.2. list: output a list of action names

1.3. send\_goal: send an action goal

1. list : shows all the available packages
2. executables [package name] : shows all the available executables

Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 pkg executables turtlesim  
turtlesim draw\_square  
turtlesim mimic  
turtlesim turtle\_teleop\_key  
turtlesim turtlesim\_node

1. run <package> <executable>

Eg: ros2 run turtlesim turtlesim\_node: will open the turtlesim gui

1. node
   1. list: shows all the nodes that are currently there
   2. info: shows subscribers, publishers, service servers, service clients, action servers, and action clients

Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 node info /turtlesim –

/turtlesim  
  Subscribers:  
    /parameter\_events: rcl\_interfaces/msg/ParameterEvent  
    /turtle1/cmd\_vel: geometry\_msgs/msg/Twist  
  Publishers:  
    /parameter\_events: rcl\_interfaces/msg/ParameterEvent  
    /rosout: rcl\_interfaces/msg/Log  
    /turtle1/color\_sensor: turtlesim/msg/Color  
    /turtle1/pose: turtlesim/msg/Pose  
  Service Servers:  
    /clear: std\_srvs/srv/Empty  
    /kill: turtlesim/srv/Kill  
    /reset: std\_srvs/srv/Empty  
    /spawn: turtlesim/srv/Spawn  
    …   
    /turtlesim/set\_parameters\_atomically: rcl\_interfaces/srv/SetParametersAtomically  
  Service Clients:  
  
  Action Servers:  
    /turtle1/rotate\_absolute: turtlesim/action/RotateAbsolute  
  Action Clients:

1. interface proto <type>: shows prototype

Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 interface proto turtlesim/srv/Spawn  
"x: 0.0  
y: 0.0  
theta: 0.0  
name: ''  
"

1. topic pub [-t times] <topic> <message>

Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 topic pub -1 /turtle1/cmd\_vel geometry\_msgs/msg/Twist "linear:  
  x: 2.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 2.0  
"

publisher: beginning loop  
publishing #1: geometry\_msgs.msg.Twist(linear=geometry\_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry\_msgs.msg.Vector3(x=0.0, y=0.0, z=2.0))

1. service

8.1. list -t: shows the available services and their type

Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 service list -t  
/clear [std\_srvs/srv/Empty]  
/kill [turtlesim/srv/Kill]  
/reset [std\_srvs/srv/Empty]  
/spawn [turtlesim/srv/Spawn]  
…

/turtlesim/set\_parameters\_atomically [rcl\_interfaces/srv/SetParametersAtomically]

8.2. call <service name> <service type> [values]

Eg: cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 call /spawn turtlesim/srv/Spawn  
“x: 0.0  
y: 0.0  
theta: 0.0  
name: “turty”

“

1. param: parameters for nodes
2. launch <package> [launch file]

**Publisher and Subscriber using C++:**

cubesat@cubesat-OptiPlex-7460-AIO:~$ cd ros2\_ws/src/  
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws/src$ ros2 pkg create --build-type ament\_cmake cpp\_topic

cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws/src$ cd..  
cd..: command not found  
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws/src$ cd ..  
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws$ colcon build

cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws$ ros2 run cpp\_topic cpp\_topic\_publisher\_spiral  
[INFO] [1720506984.563733873] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '2.500000'  
[INFO] [1720506985.063744280] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '2.600000'  
[INFO] [1720506985.563968460] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '2.700000'  
[INFO] [1720506986.063875405] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '2.800000'  
[INFO] [1720506986.563874948] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '2.900000'  
[INFO] [1720506987.063751099] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '3.000000'  
[INFO] [1720506987.563706104] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '3.100000'  
[INFO] [1720506988.063900609] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '3.200000'  
[INFO] [1720506988.563896863] [cpp\_topic\_publisher\_spiral]: Sending - Linear Velocity : '4.000000', Angular Velocity : '3.300000'

cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws$ ros2 run cpp\_topic cpp\_topic\_subscriber\_spiral  
[INFO] [1720506986.564407289] [cpp\_topic\_subscriber\_spiral]: Recieved - Linear Velocity : '4.000000', Angular Velocity : '2.900000'  
[INFO] [1720506987.064263391] [cpp\_topic\_subscriber\_spiral]: Recieved - Linear Velocity : '4.000000', Angular Velocity : '3.000000'  
[INFO] [1720506987.564207517] [cpp\_topic\_subscriber\_spiral]: Recieved - Linear Velocity : '4.000000', Angular Velocity : '3.100000'  
[INFO] [1720506988.064409149] [cpp\_topic\_subscriber\_spiral]: Recieved - Linear Velocity : '4.000000', Angular Velocity : '3.200000'  
[INFO] [1720506988.564411002] [cpp\_topic\_subscriber\_spiral]: Recieved - Linear Velocity : '4.000000', Angular Velocity : '3.300000'

**Service Server and Client using Python:**

cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws/src$ ros2 pkg create --build-type ament\_python py\_service  
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws/src$ cd ..  
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws$ colcon build

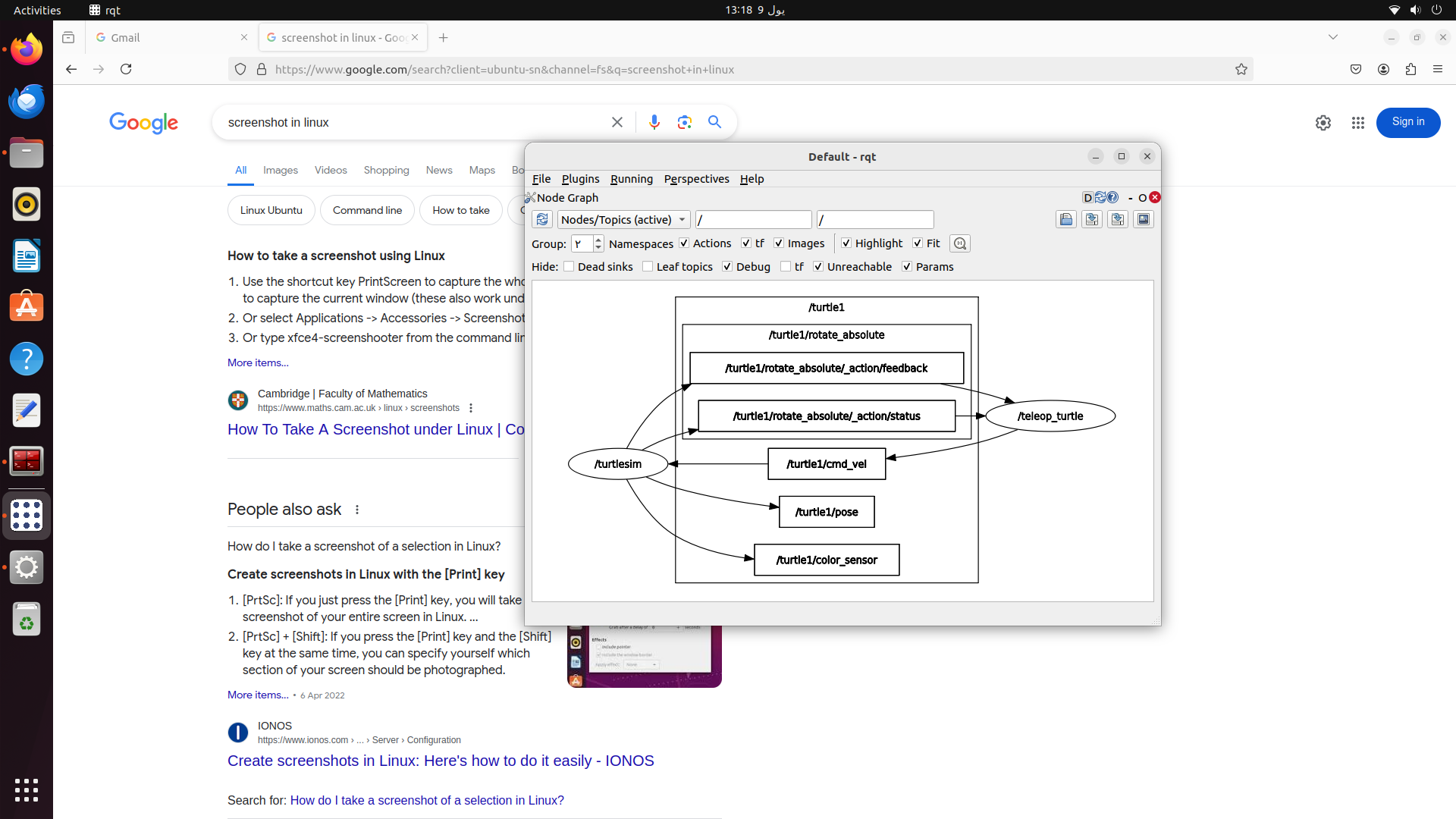
cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws$ ros2 run py\_service py\_service\_server\_polar  
[INFO] [1720507959.416145756] [py\_service\_server\_polar]: Response - Polar Radial Coordinate : 7.071068, Polar Angular Coordinate : 45.000000

cubesat@cubesat-OptiPlex-7460-AIO:~/ros2\_ws$ ros2 run py\_service py\_service\_client\_polar  
[INFO] [1720507959.404140904] [py\_service\_client\_polar]: Sending - X Coordinate : 5.000000, Y Coordinate : 5.000000  
[INFO] [1720507959.416642845] [py\_service\_client\_polar]: Success

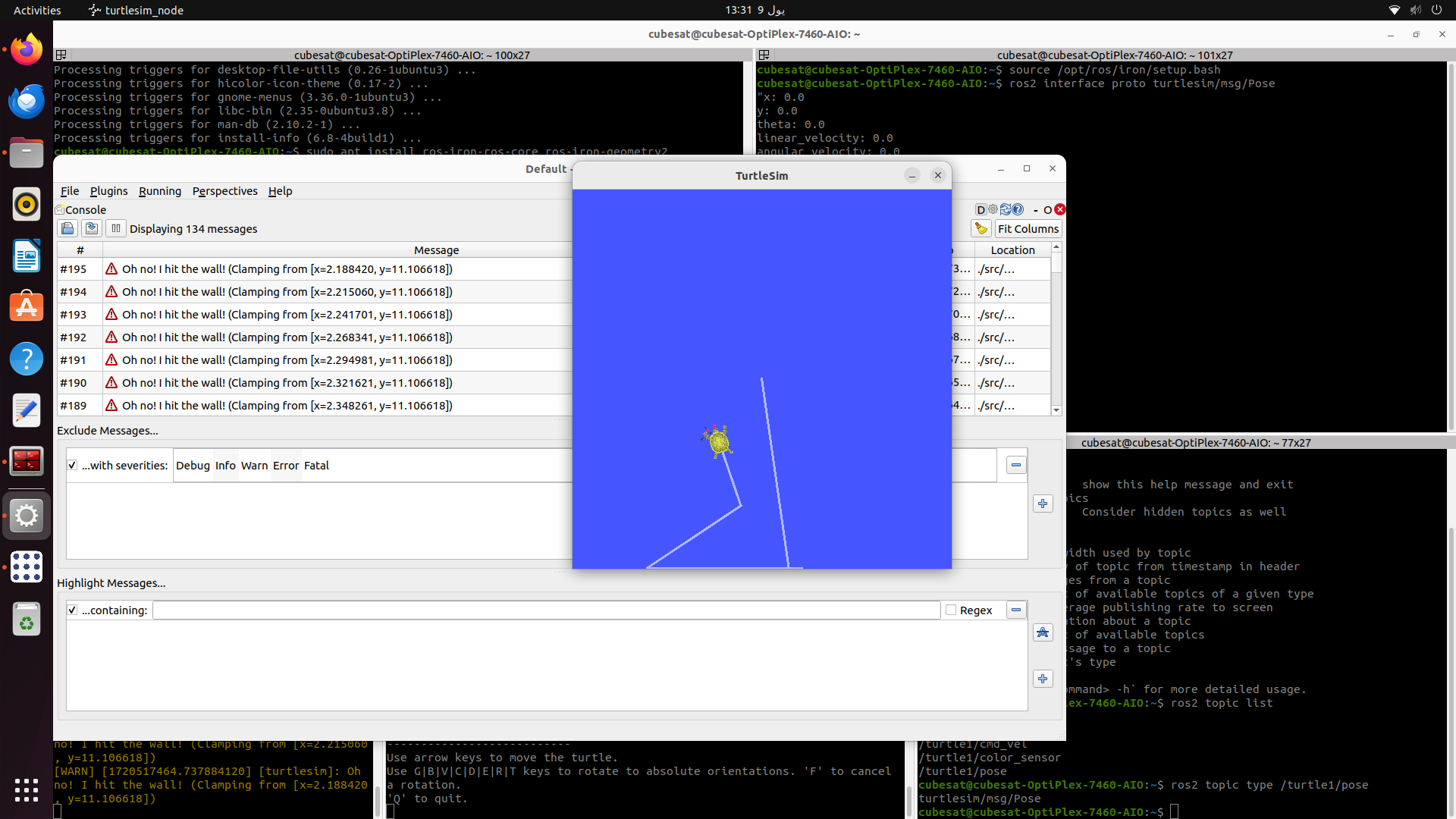
**RQt**

Plugins:

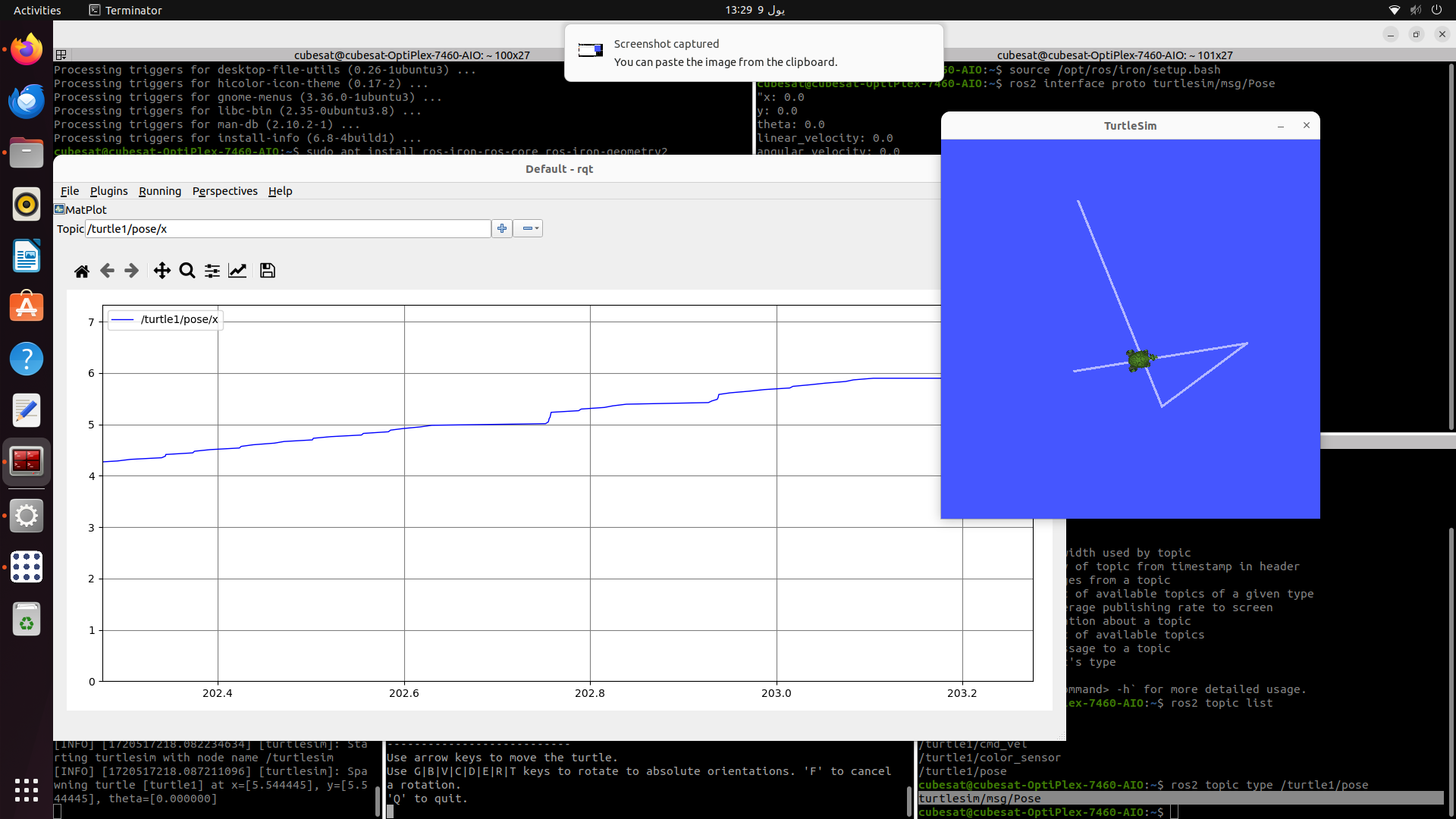
1. Introspection -> Node Graph



1. Logging -> Console



1. Plots



**Bag**

Records data published on a topic and saves it. The data can be used to reproduce the result of our testcases.

cubesat@cubesat-OptiPlex-7460-AIO:~$ ros2 bag record /turtle1/cmd\_vel

cubesat@cubesat-OptiPlex-7460-AIO:~/rosbag2\_2024\_07\_09-13\_38\_45$ ros2 bag info rosbag2\_2024\_07\_09-13\_38\_45\_0.mcap  
  
Files:             rosbag2\_2024\_07\_09-13\_38\_45\_0.mcap  
Bag size:          8.3 KiB  
Storage id:        mcap  
Duration:          9.537s  
Start:             Jul  9 2024 13:38:56.747 (1720517936.747)  
End:               Jul  9 2024 13:39:06.284 (1720517946.284)  
Messages:          61  
Topic information: Topic: /turtle1/cmd\_vel | Type: geometry\_msgs/msg/Twist | Count: 61 | Serialization Format: cdr

cubesat@cubesat-OptiPlex-7460-AIO:~/rosbag2\_2024\_07\_09-13\_38\_45$ ros2 bag play rosbag2\_2024\_07\_09-13\_38\_45\_0.mcap

The turtle will move according to the data stored.

**Gazebo**

Sudo apt install ros-iron-gazebo-ros-pkgs

Running a demo simulation:

cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo\_plugins/worlds$ gazebo --verbose /opt/ros/iron/share/gazebo\_plugins/worlds/gazebo\_ros\_diff\_drive\_demo.world

To move the robot forward:

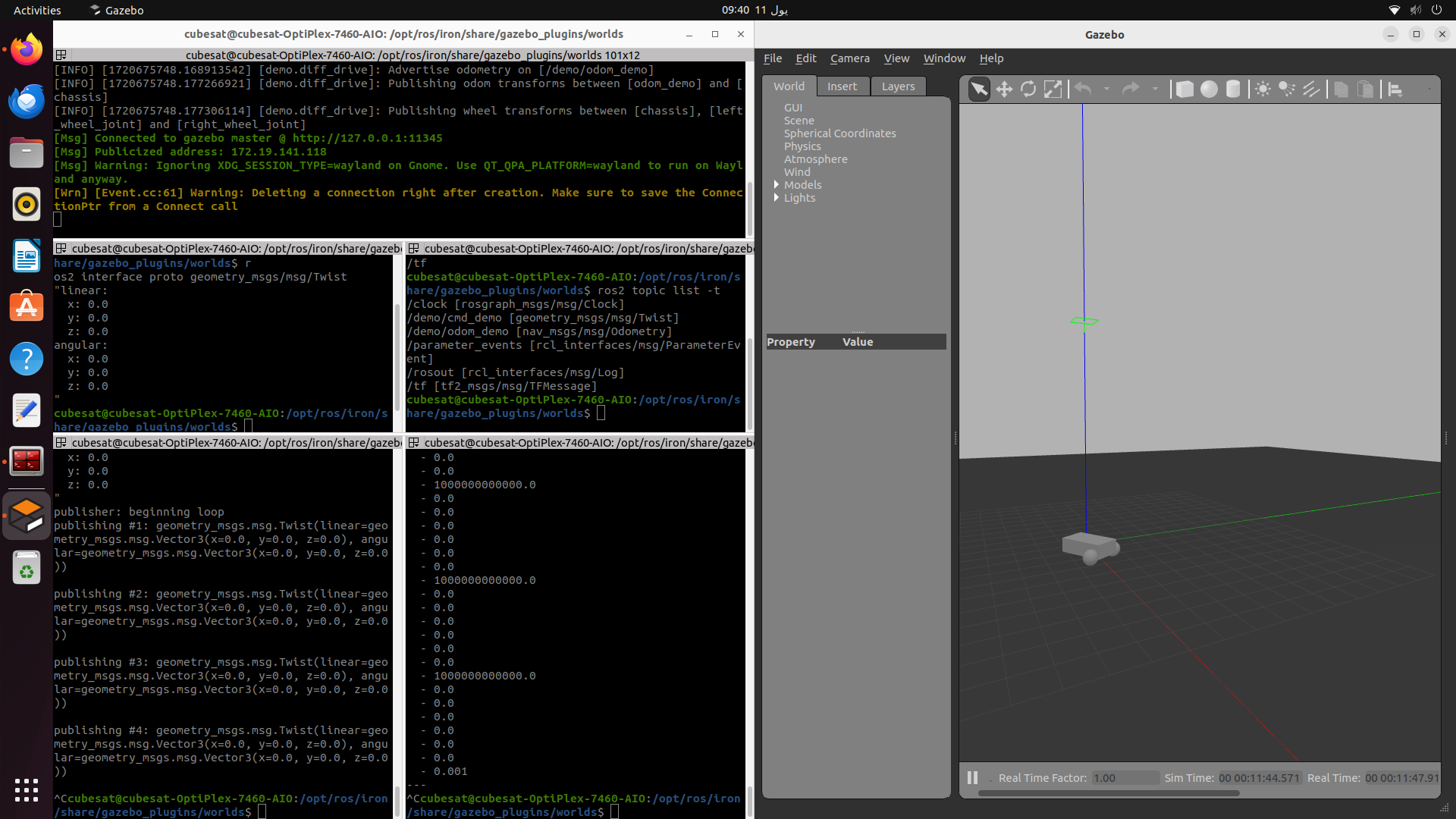
cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo\_plugins/worlds$ ros2 topic pub /demo/cmd\_demo geometry\_msgs/msg/Twist "linear:  
  x: 1.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
"

To stop the movement:

cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo\_plugins/worlds$ ros2 topic pub /demo/cmd\_demo geometry\_msgs/msg/Twist "linear:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 0.0  
"

To see odometry:

cubesat@cubesat-OptiPlex-7460-AIO:/opt/ros/iron/share/gazebo\_plugins/worlds$ ros2 topic echo /demo/odom\_demo



Making a robot and an environment:

1. Make a package ws\_gazebo

cubesat@cubesat-OptiPlex-7460-AIO:/$ mkdir -p ~/ws\_gazebo/src

cubesat@cubesat-OptiPlex-7460-AIO:/$ cd ~/ws\_gazebo/  
cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo$ colcon build

cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo$ cd src/  
cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo/src$ ros2 pkg create --build-type ament\_cmake gazebo\_test

cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo/src$ cd gazebo\_test/

cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo/src/gazebo\_test$ ls -l  
total 16  
-rw-rw-r-- 1 cubesat cubesat  904 يول 11 09:53 CMakeLists.txt  
drwxrwxr-x 3 cubesat cubesat 4096 يول 11 09:53 include  
-rw-rw-r-- 1 cubesat cubesat  601 يول 11 09:53 package.xml  
drwxrwxr-x 2 cubesat cubesat 4096 يول 11 09:53 src  
cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo/src/gazebo\_test$ mkdir launch model  
cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo/src/gazebo\_test$ ls -l  
total 24  
-rw-rw-r-- 1 cubesat cubesat  904 يول 11 09:53 CMakeLists.txt  
drwxrwxr-x 3 cubesat cubesat 4096 يول 11 09:53 include  
drwxrwxr-x 2 cubesat cubesat 4096 يول 11 09:54 launch  
drwxrwxr-x 2 cubesat cubesat 4096 يول 11 09:54 model  
-rw-rw-r-- 1 cubesat cubesat  601 يول 11 09:53 package.xml  
drwxrwxr-x 2 cubesat cubesat 4096 يول 11 09:53 src

cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo/src/gazebo\_test$ cd ~/ws\_gazebo/  
cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo$ colcon build

1. Make xacro and gazebo files for the robot

robot.xacro

robot.gazebo

1. Make the file for the world

empty\_world.world

1. Make the launch file

gazebo\_model.launch.py

1. Edit package.xml and CMakeLists.txt to include all dependencies
2. Run on ros

cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo$ colcon build  
cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo$ source ~/ws\_gazebo/install/setup.bash  
cubesat@cubesat-OptiPlex-7460-AIO:~/ws\_gazebo$ ros2 launch gazebo\_test gazebo\_model.launch.py