ROS2 Python programming

RLmodel

ybbaek

Contents

프로그램 실습

ROS2 Jupyter notebook

Jupyter notebook

설치 방법:

>sudo apt install python3-pip

>pip3 install —upgrade pip

>pip3 install jupyter ipywidgets pyyaml bqplot

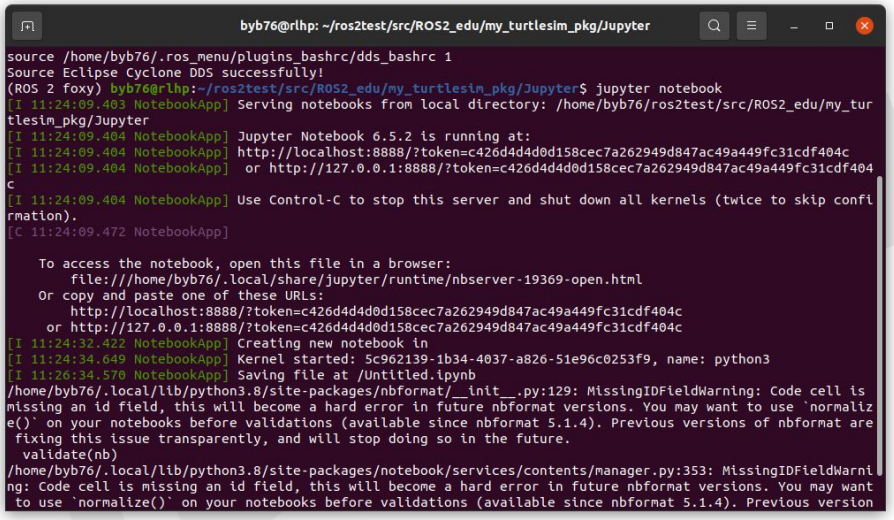
>sudo apt install jupyter-core

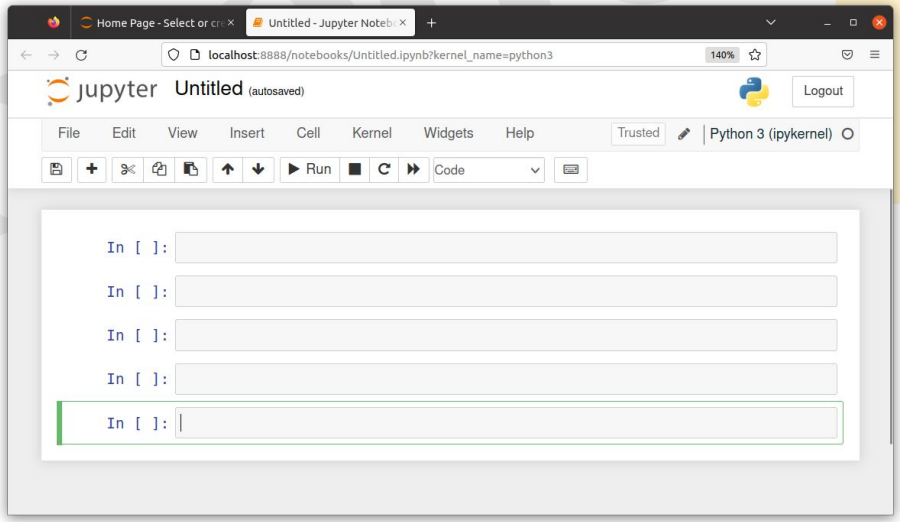
ROS2 Python 실습

Jupyter notebook 실행

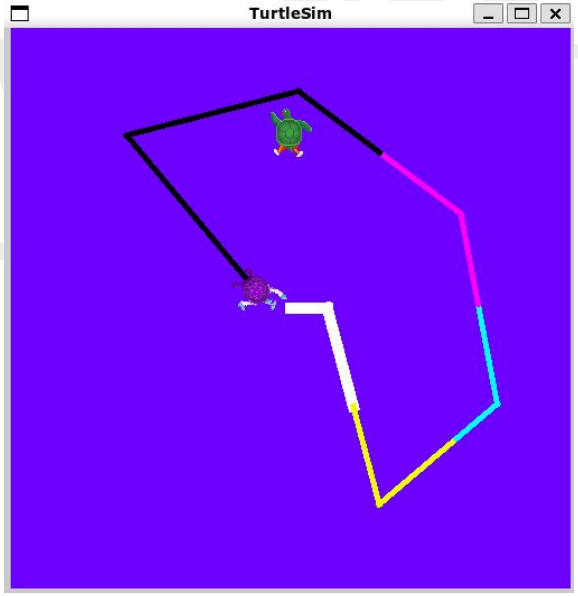
>jupyter notebook

>mark down 언어지원





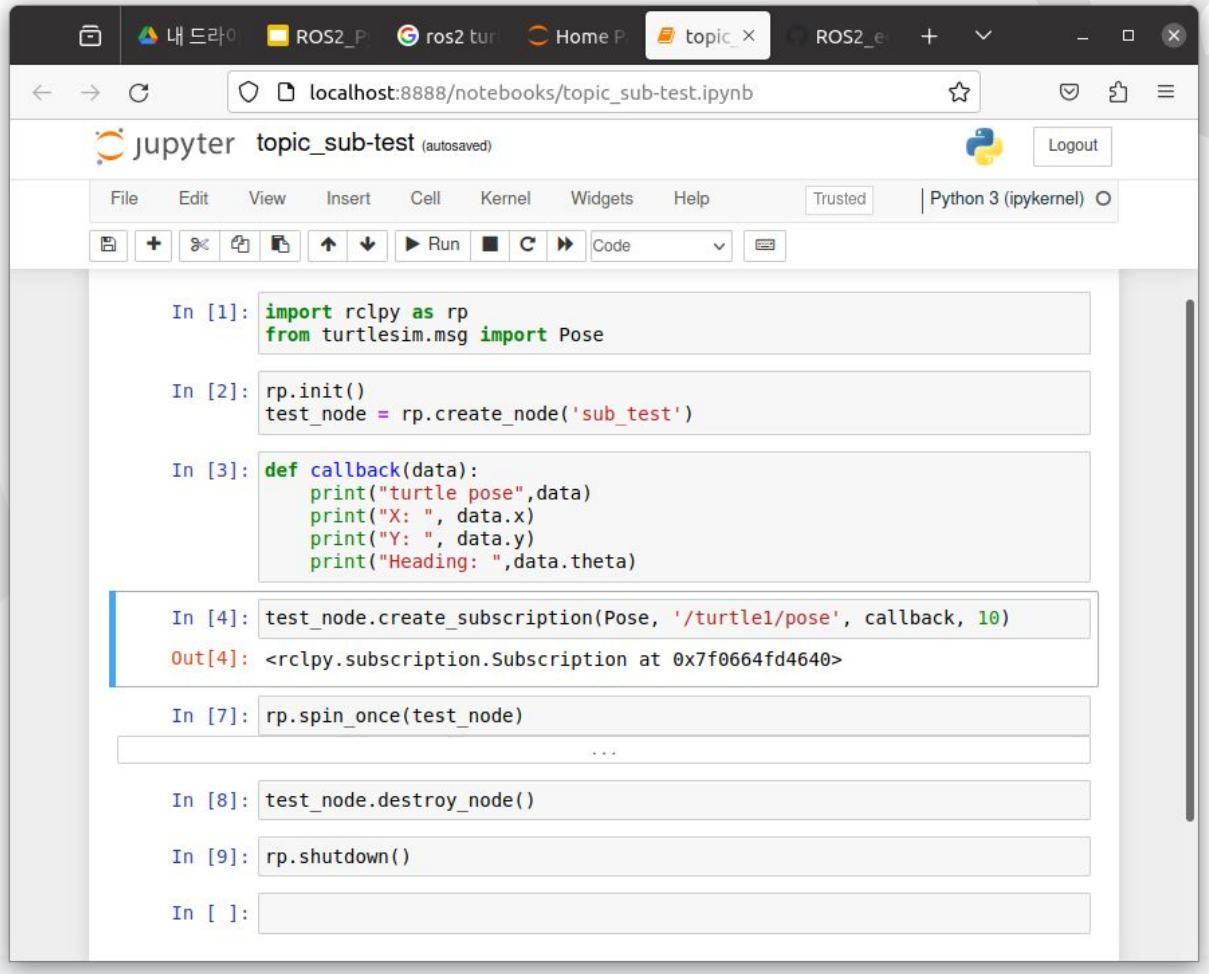
Example1
turtlesim pose subscribe



ROS2 Python 실습

topic 수신

상세: turtlesim pose 토픽 수신



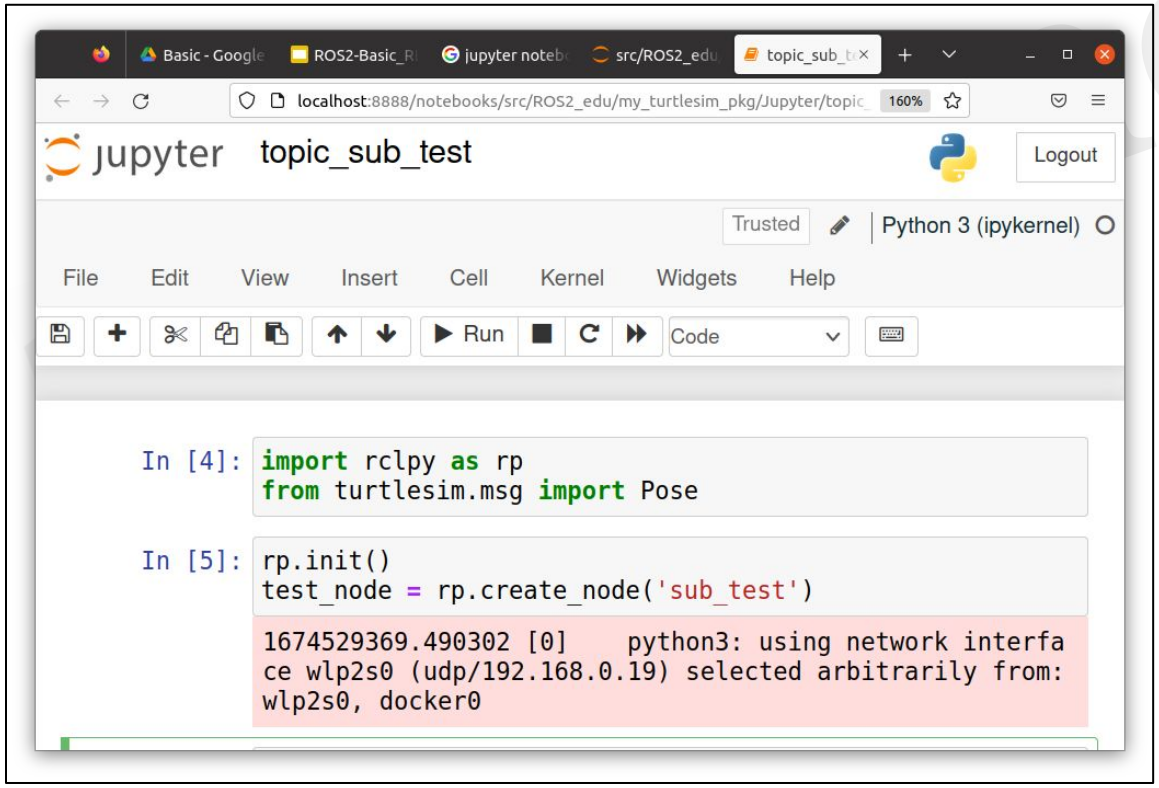
ROS2 Python 실습

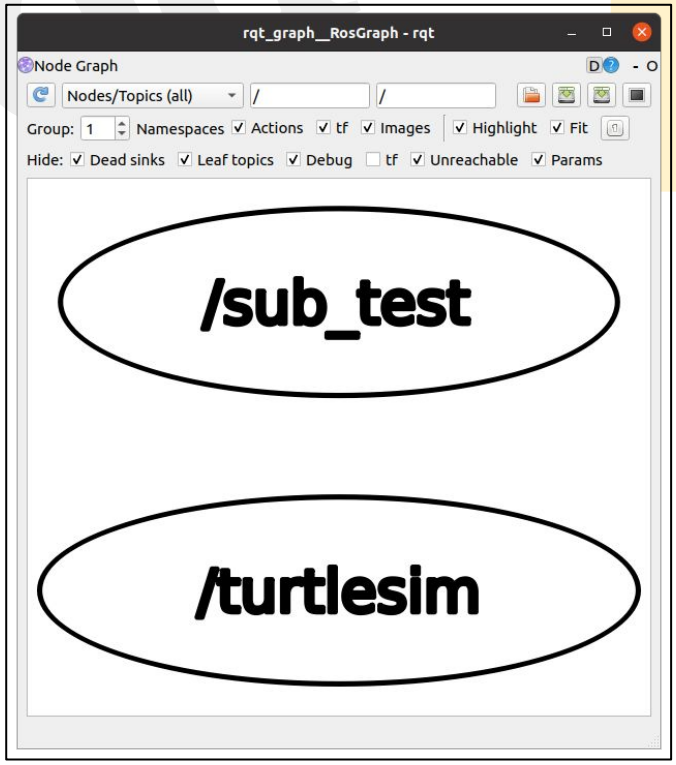
topic 수신

준비: $Source /opt/ros/foxy/setup.bash

순서: python code 작성 > 노드 생성

확인: $ros2 node list



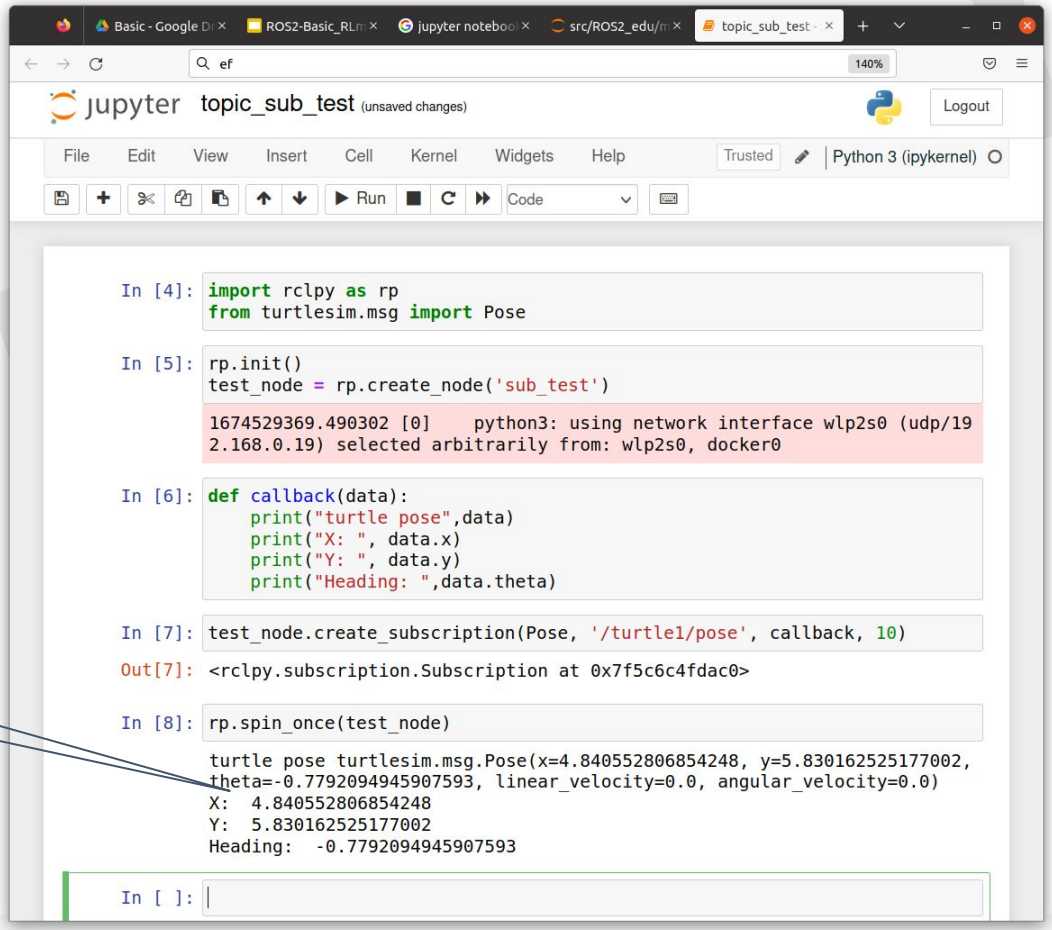


ROS2 Python 실습

topic 코드

>토픽 수신 (subscription)

rp.spin_once(sub_node)



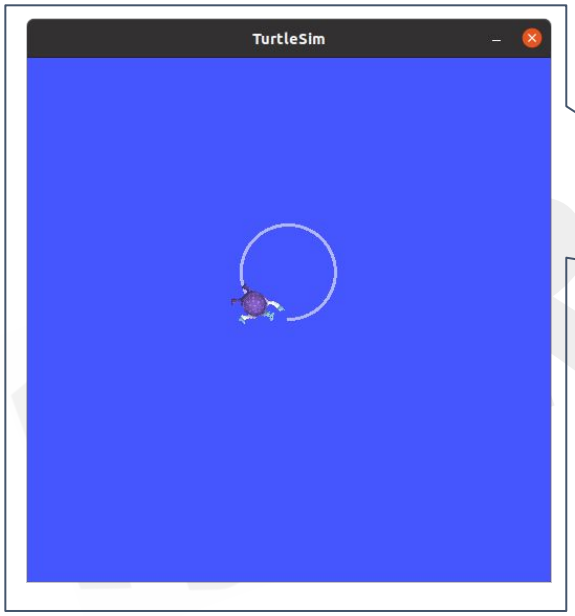
수신된 토픽 내용 확인

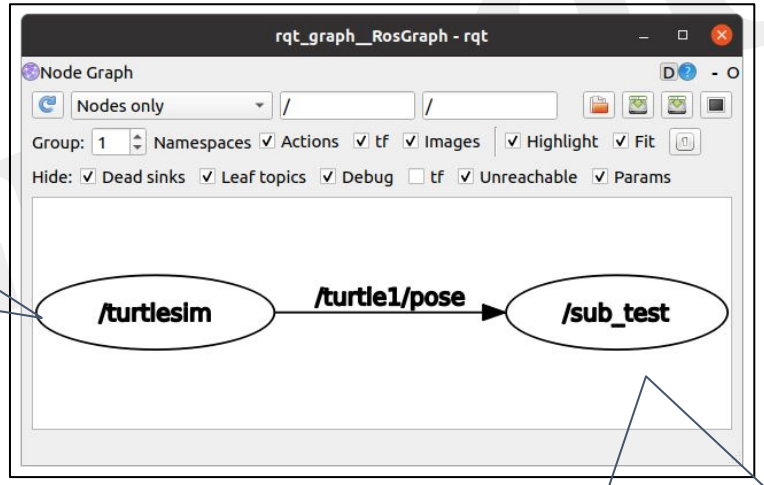
ROS2 Python 실습

topic 코드

>토픽 구독

rp.spin_once(sub_node)



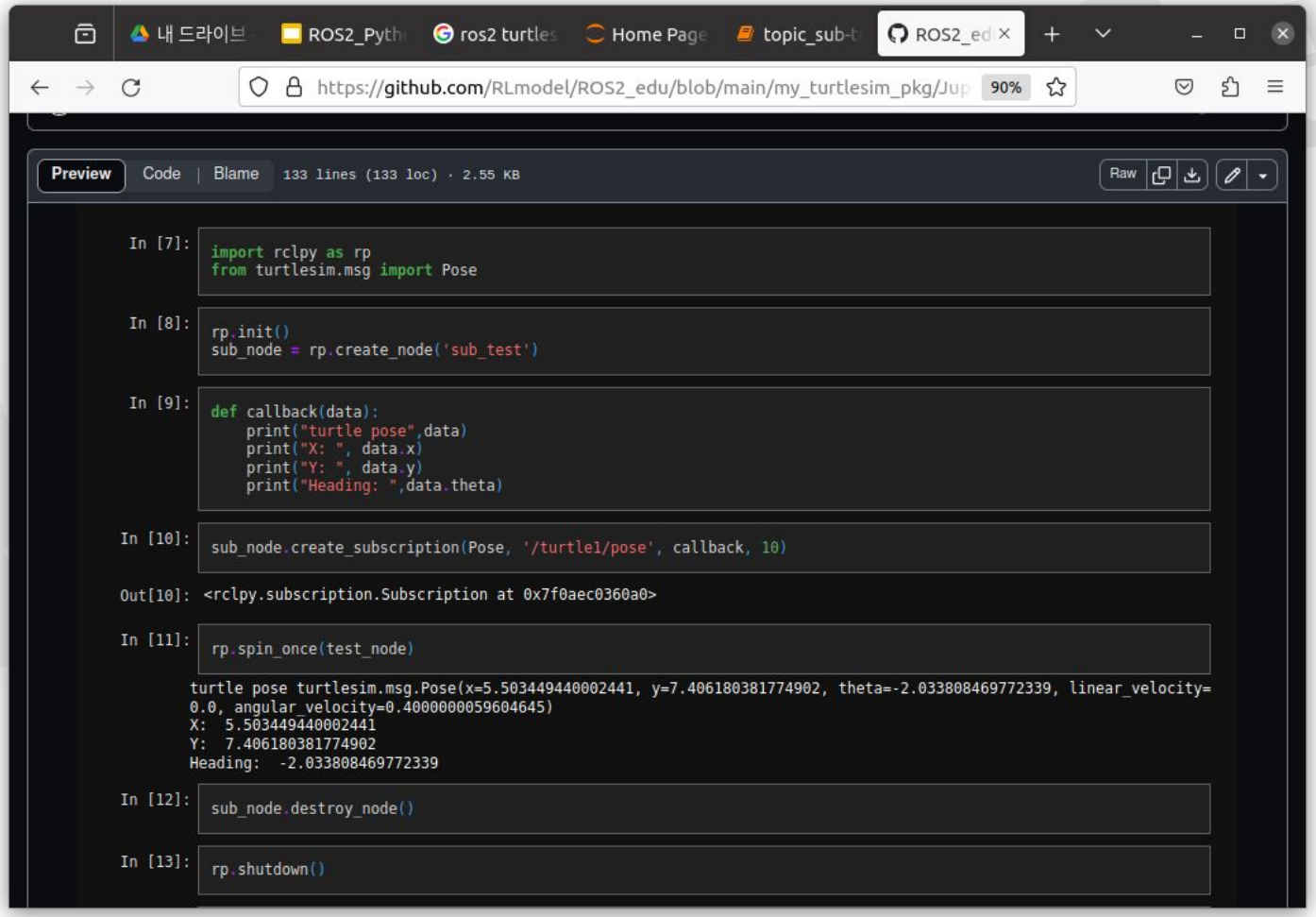



```
In [8]: rp.spin_once(test_node)

        turtle pose turtlesim.msg.Pose(x=4.840552806854248, y=5.830162525177002,
        theta=-0.7792094945907593, linear_velocity=0.0, angular_velocity=0.0)
        X:  4.840552806854248
        Y:  5.830162525177002
        Heading:  -0.7792094945907593
```

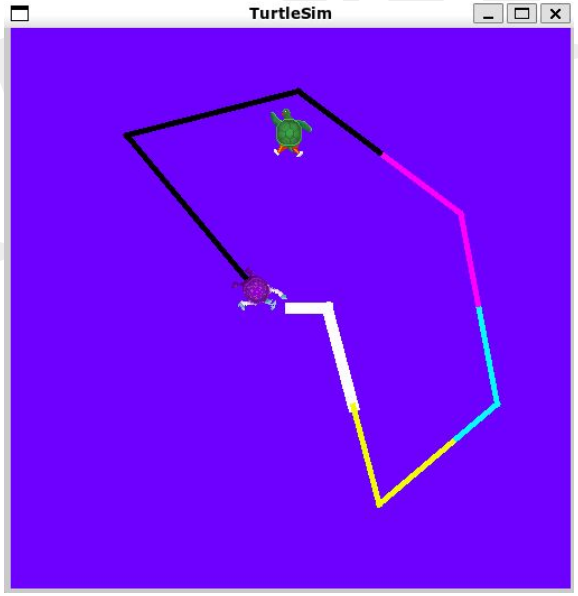
ROS2 Python 실습

topic 코드

>링크: https://github.com/RLmodel/ROS2_edu/blob/main/my_turtlesim_pkg/Jupyter/py_sub-pose.ipynb



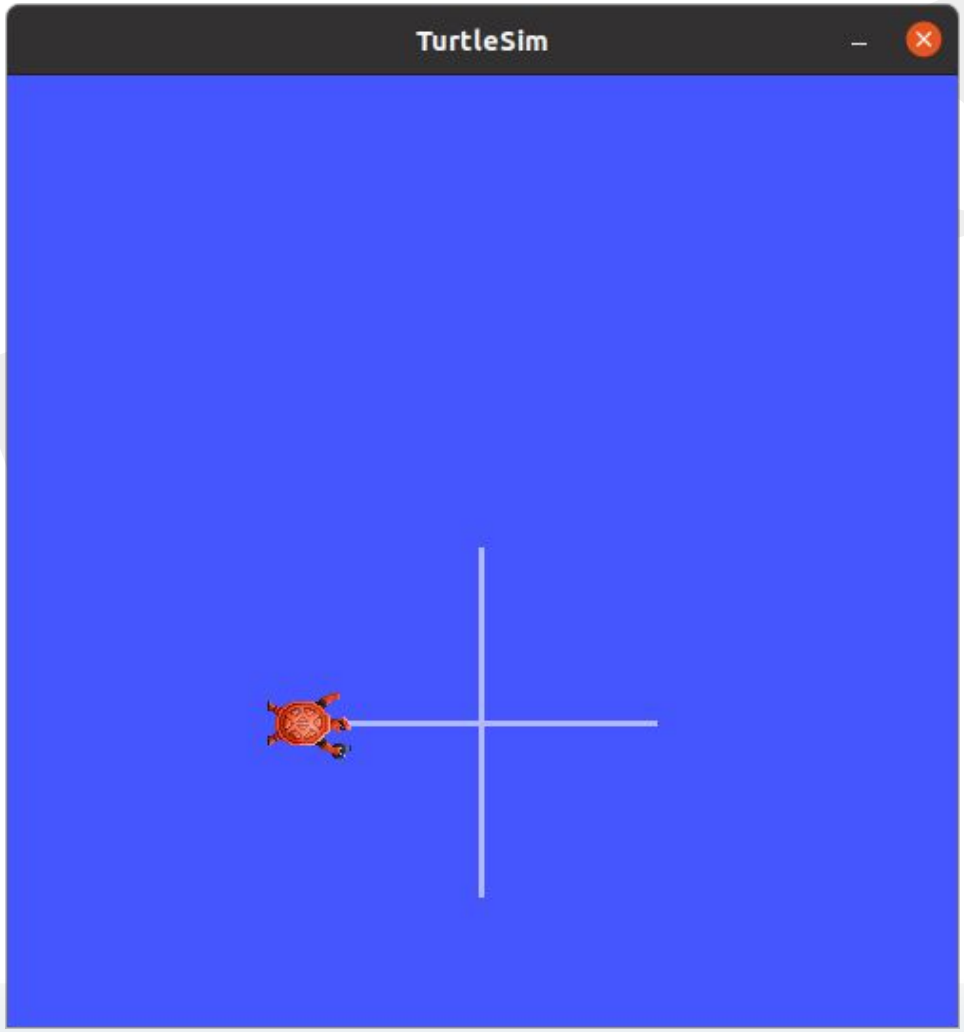
Example2
turtlesim Twist msg publish



ROS2 Python 실습

Topic 생성 코드 / turtlesim

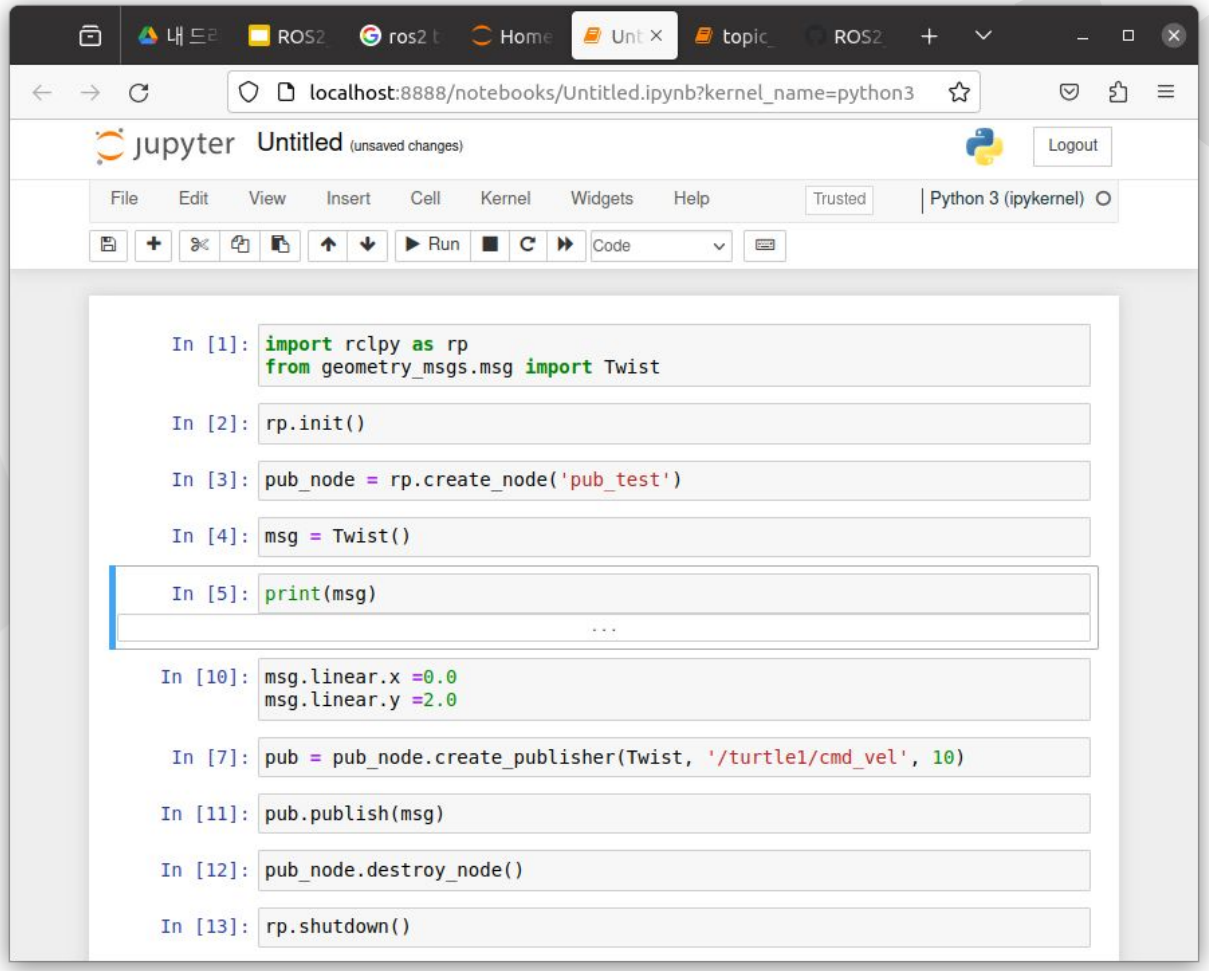
기능: Twist msg (cmd_vel) publish



ROS2 Python 실습

Topic 생성 코드

기능: Twist msg (cmd_vel) publish

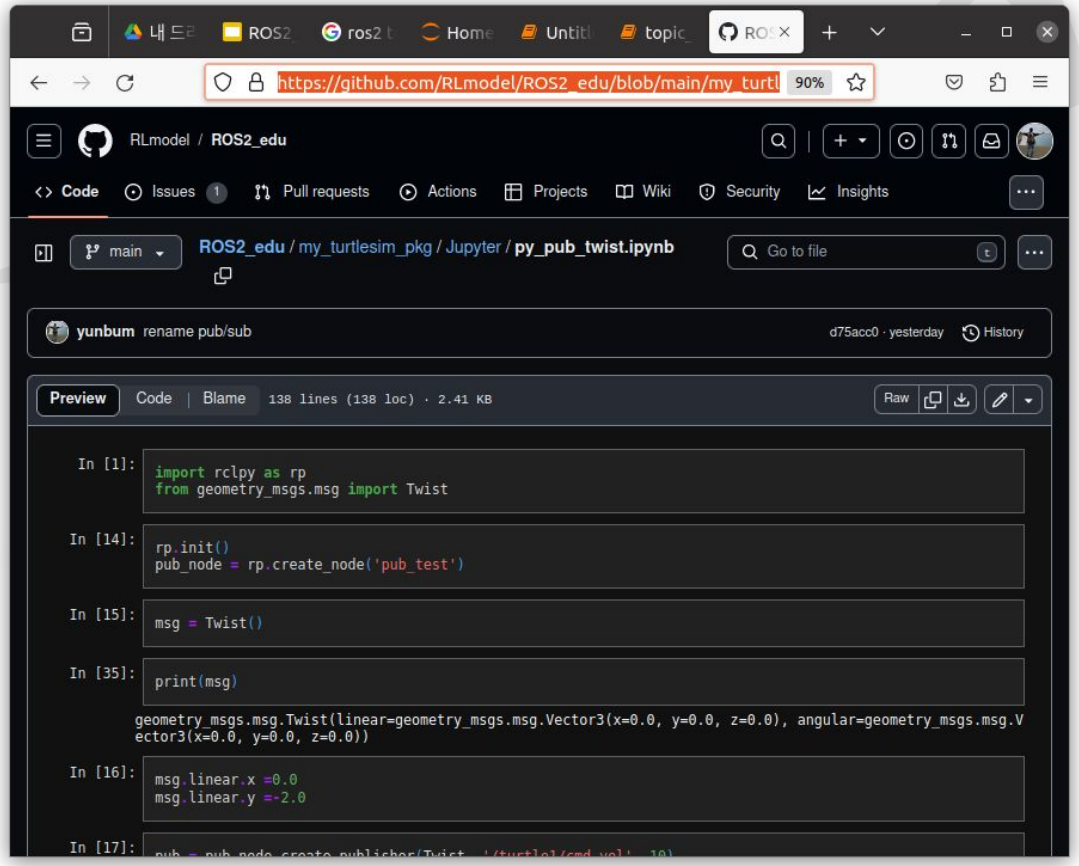


ROS2 Python 실습

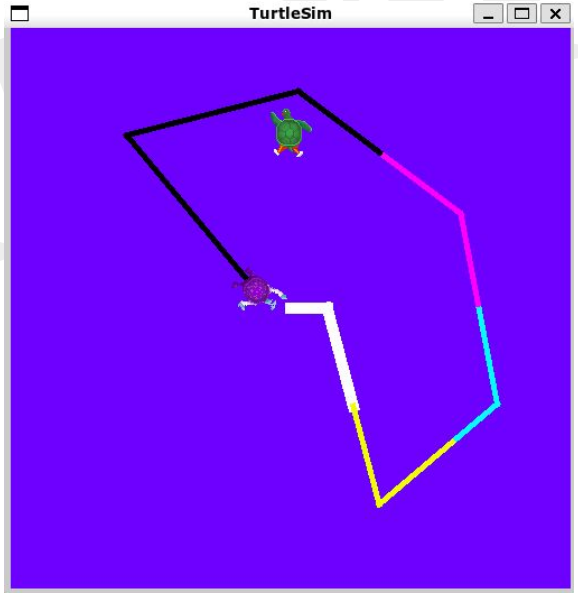
Topic 생성 코드

기능: Twist msg (cmd_vel) publish

Link: https://github.com/RLmodel/ROS2_edu/blob/main/my_turtlesim_pkg/Jupyter/py_pub_twist.ipynb



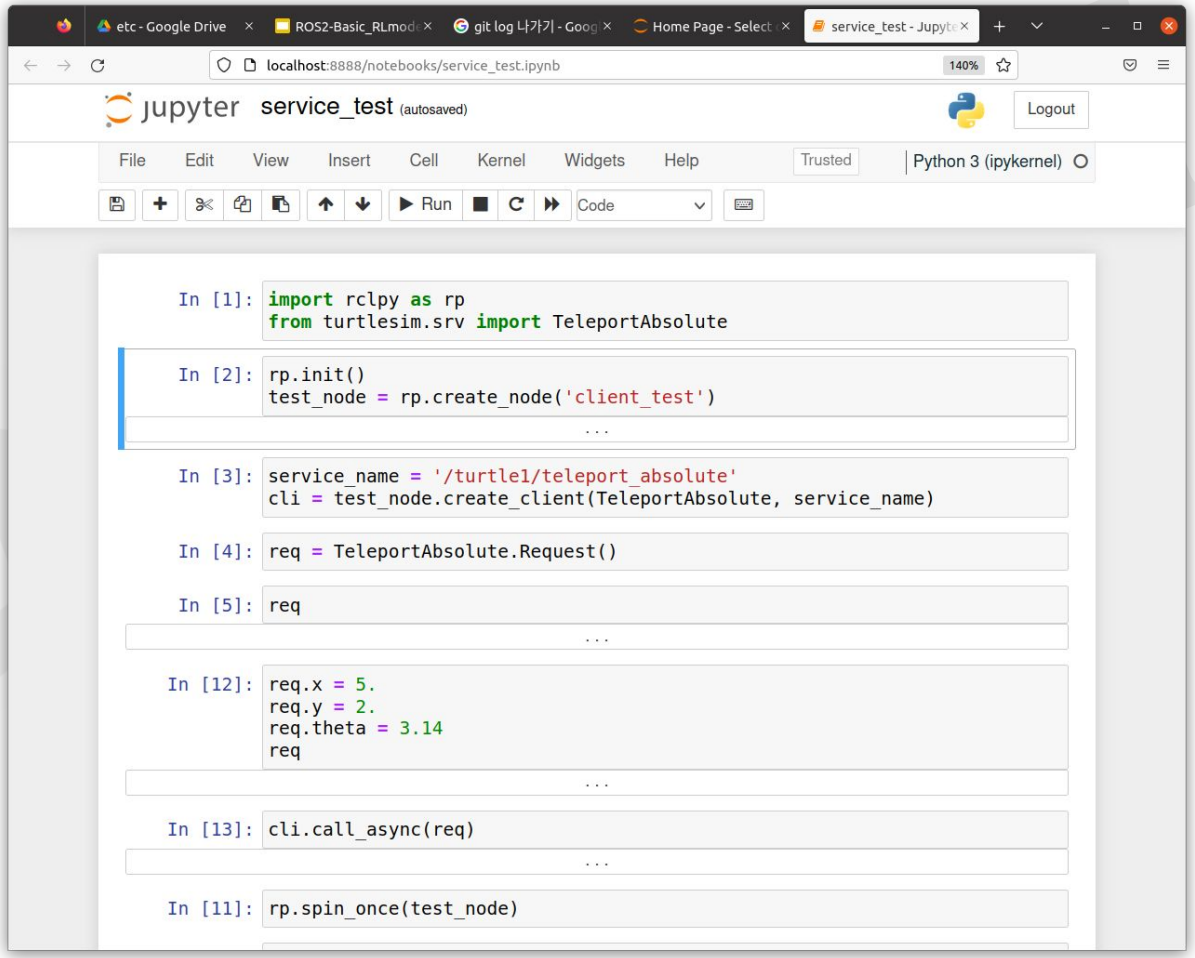
Example3
turtlesim service call



ROS2 Python 실습

service 코드

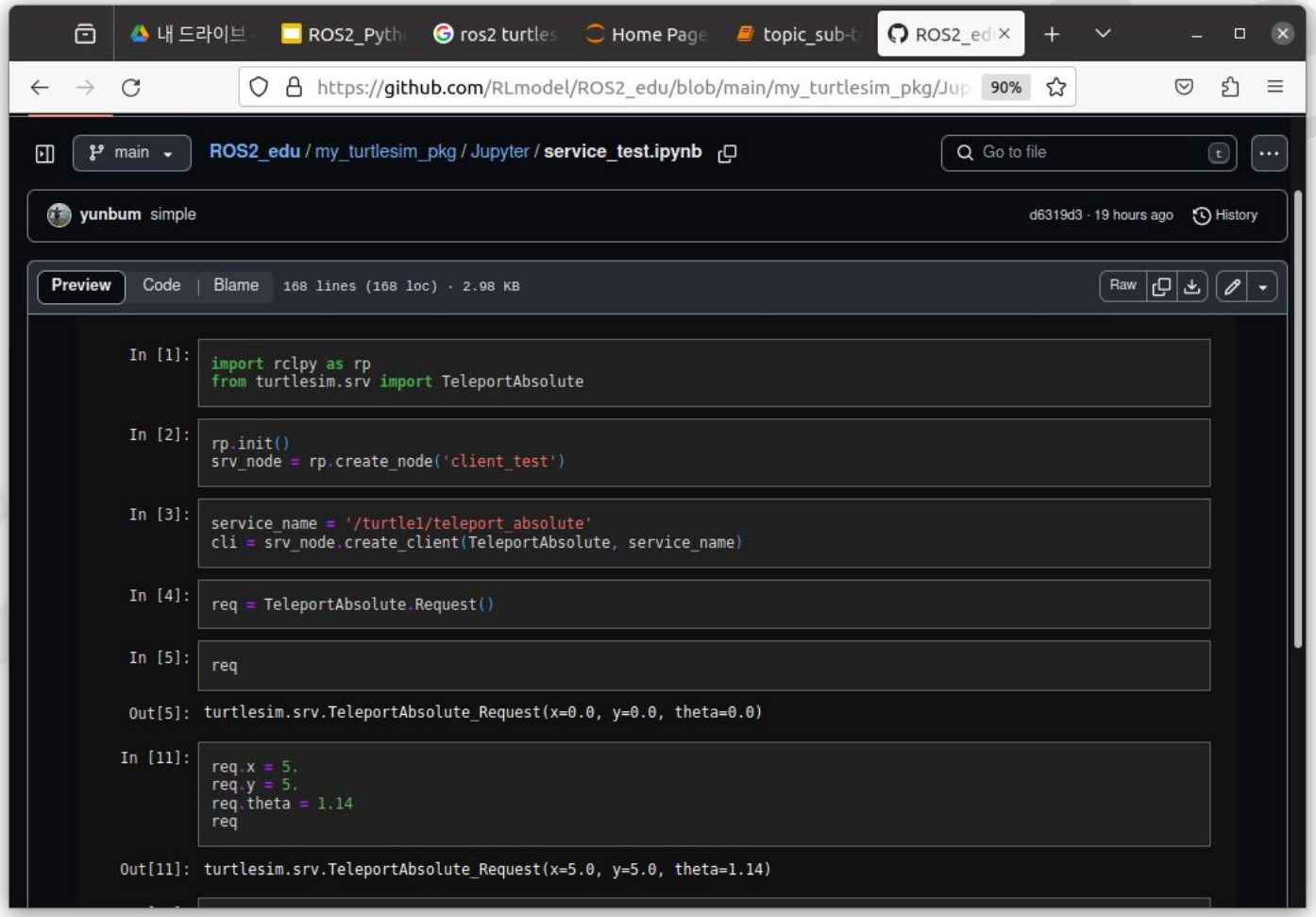
순서: node 생성 > client 생성 > client call >



ROS2 Python 실습

service 코드

link: https://github.com/RLmodel/ROS2_edu/blob/main/my_turtlesim_pkg/Jupyter/service_test.ipynb



프로그램 실습

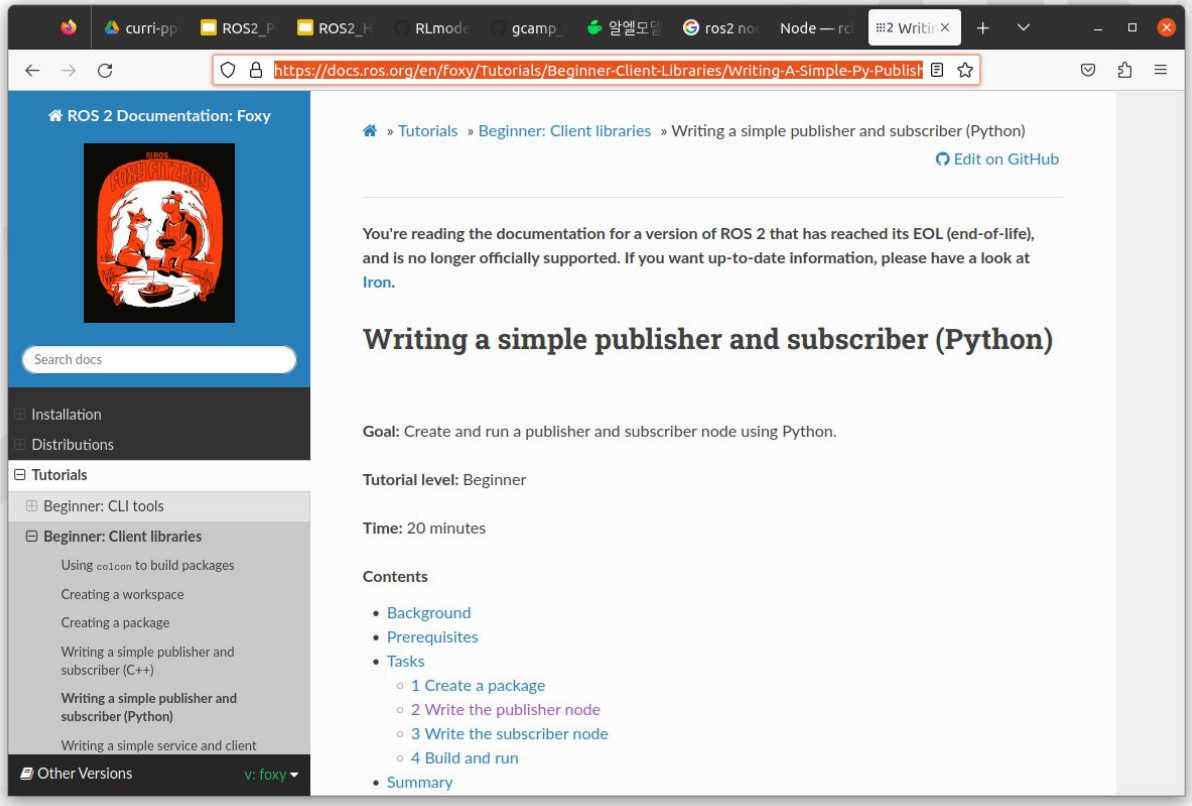
ROS2 Python

ROS2 실습

Topic publish

<String msg>

link:
https://docs.ros.org/en/foxy/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Py-Publisher-And-Subscriber.html#write-the-publisher-node



ROS2 실습

Topic publish

<String msg>

link: ttps://docs.ros.org/en/foxy/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Py-Publisher-And-Subscriber.html#write-the-publisher-node

```
import rclpy
from rclpy.node import Node

from std_msgs.msg import String


class MinimalPublisher(Node):

        def __init__(self):
        super().__init__('minimal_publisher')
        self.publisher_ = self.create_publisher(String, 'topic', 10)
        timer_period = 0.5  # seconds
        self.timer = self.create_timer(timer_period, self.timer_callback)
        self.i = 0

        def timer_callback(self):
        msg = String()
        msg.data = 'Hello World: %d' % self.i
        self.publisher_.publish(msg)
        self.get_logger().info('Publishing: "%s"' % msg.data)
        self.i += 1


def main(args=None):
        rclpy.init(args=args)

        minimal_publisher = MinimalPublisher()

        rclpy.spin(minimal_publisher)

        # Destroy the node explicitly
        # (optional - otherwise it will be done automatically
        # when the garbage collector destroys the node object)
        minimal_publisher.destroy_node()
        rclpy.shutdown()


if __name__ == '__main__':
        main()
```

ROS2 실습

workspace 생성

참고링크: https://docs.ros.org/en/foxy/Tutorials/Beginner-Client-Libraries/Creating-A-Workspace/Creating-A-Workspace.html

<순서>

1 Source ROS 2 environment
2 Create a new directory
3 Clone a sample repo
4 Resolve dependencies
5 Build the workspace with colcon
6 Source the overlay
7 Modify the overlay

underlay : ROS 2 설치의 기본 패키지와 라이브러리를 포함하는 디렉토리
overlay : 개발자가 추가로 설치한 패키지와 라이브러리를 포함하는 디렉토리
동작: ROS2는 먼저 overlay 디렉터리에서 찾아보고, 찾을 수 없으면 underlay 디렉토리에서 패키지를 찾습니다.

ROS2 실습

package 생성

참고링크: https://docs.ros.org/en/foxy/Tutorials/Beginner-Client-Libraries/Creating-A-Workspace/Creating-A-Workspace.html

<순서>

```
$ mkdir -p ~/ros2_ws/src
$ cd ~/ros2_ws/src
$ git clone xxx
$ cd..
$ sudo apt install python3-rosdep
$ sudo rosdep init
$ rosdep update
$ rosdep install -i --from-path src --rosdistro foxy -y
$ colcon build
$ . install/local_setup.bash
```

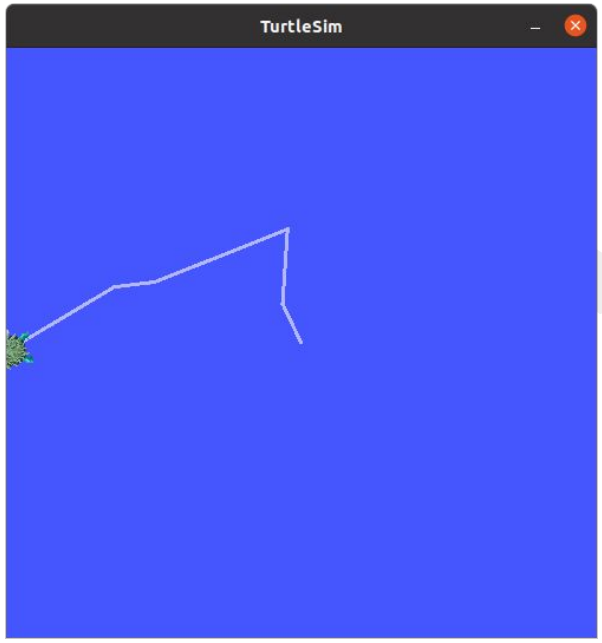
ROS2 실습

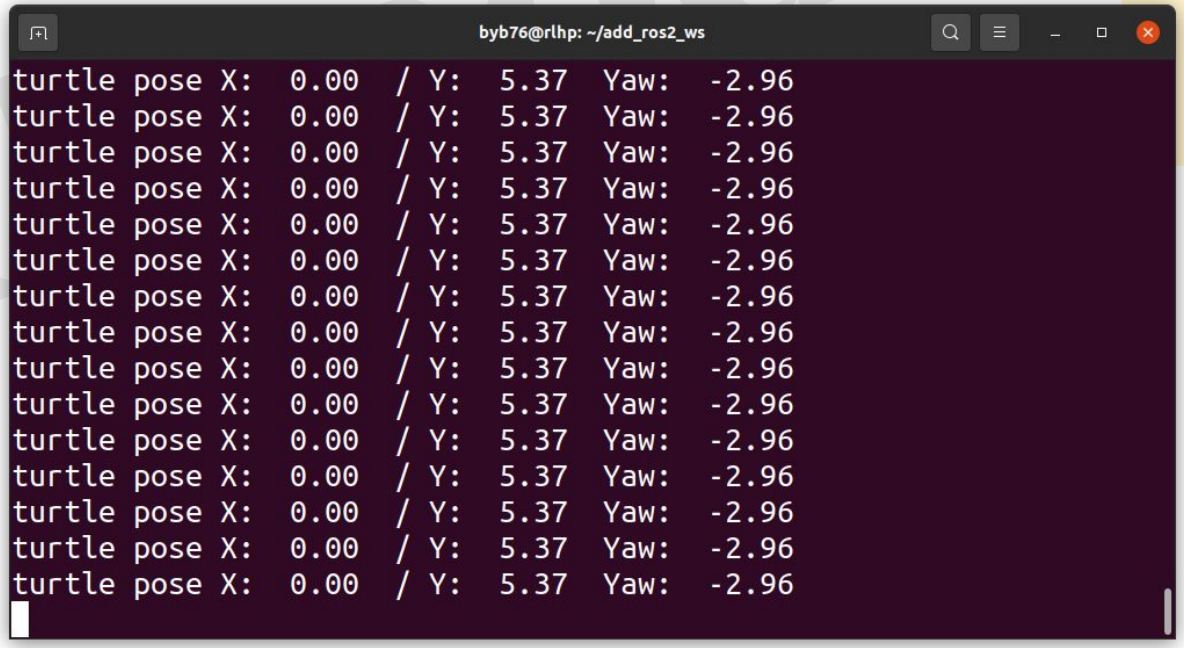
workspace 생성

생성: mkdir -p ~/ros2_ws/src

워크스페이스 : ros2_ws

link: https://github.com/yunbum/ROS2_edu.git



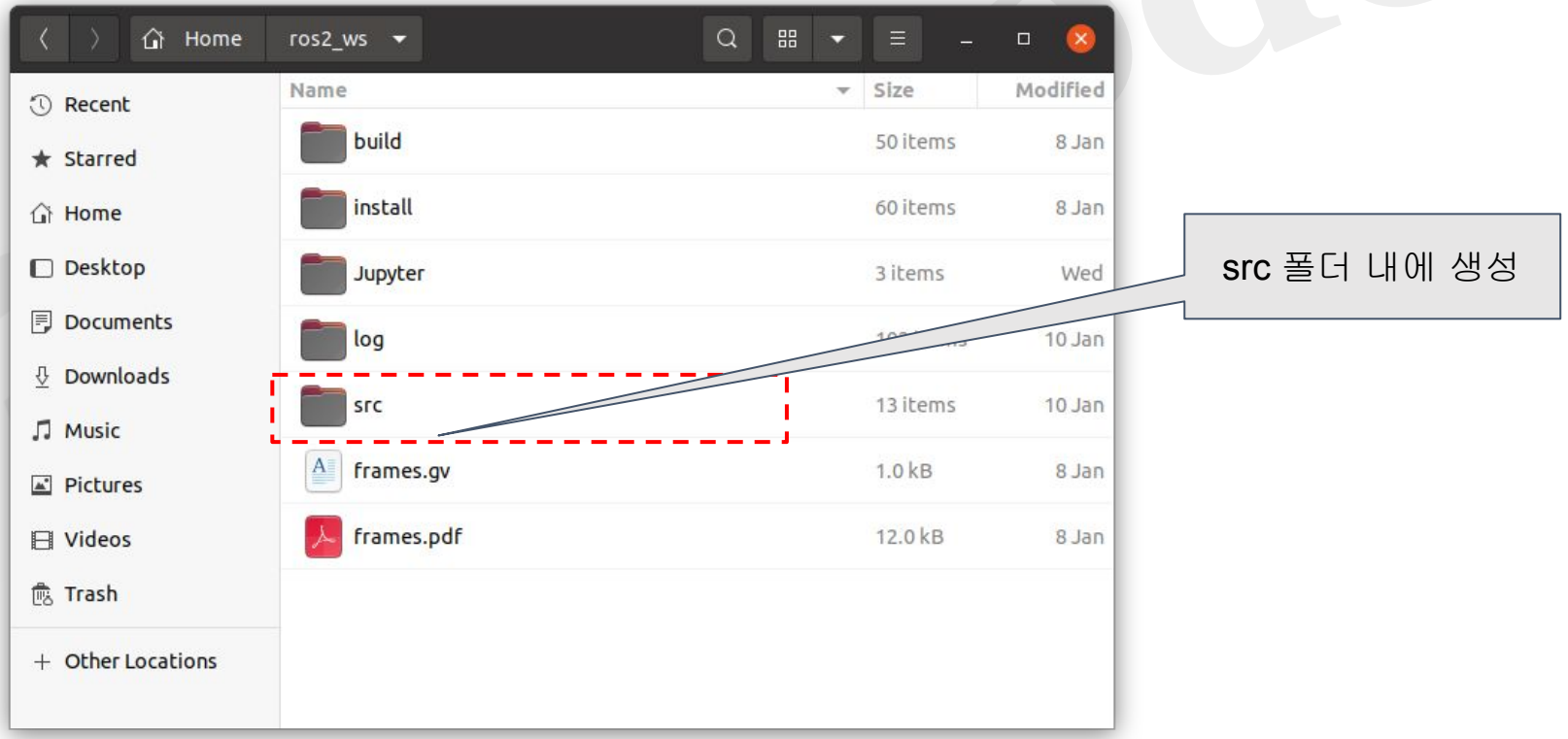


ROS2 실습

Package 생성

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

```
$ cd ~/ros2_ws/src
$ ros2 pkg create --build-type ament_python pkg_name
```

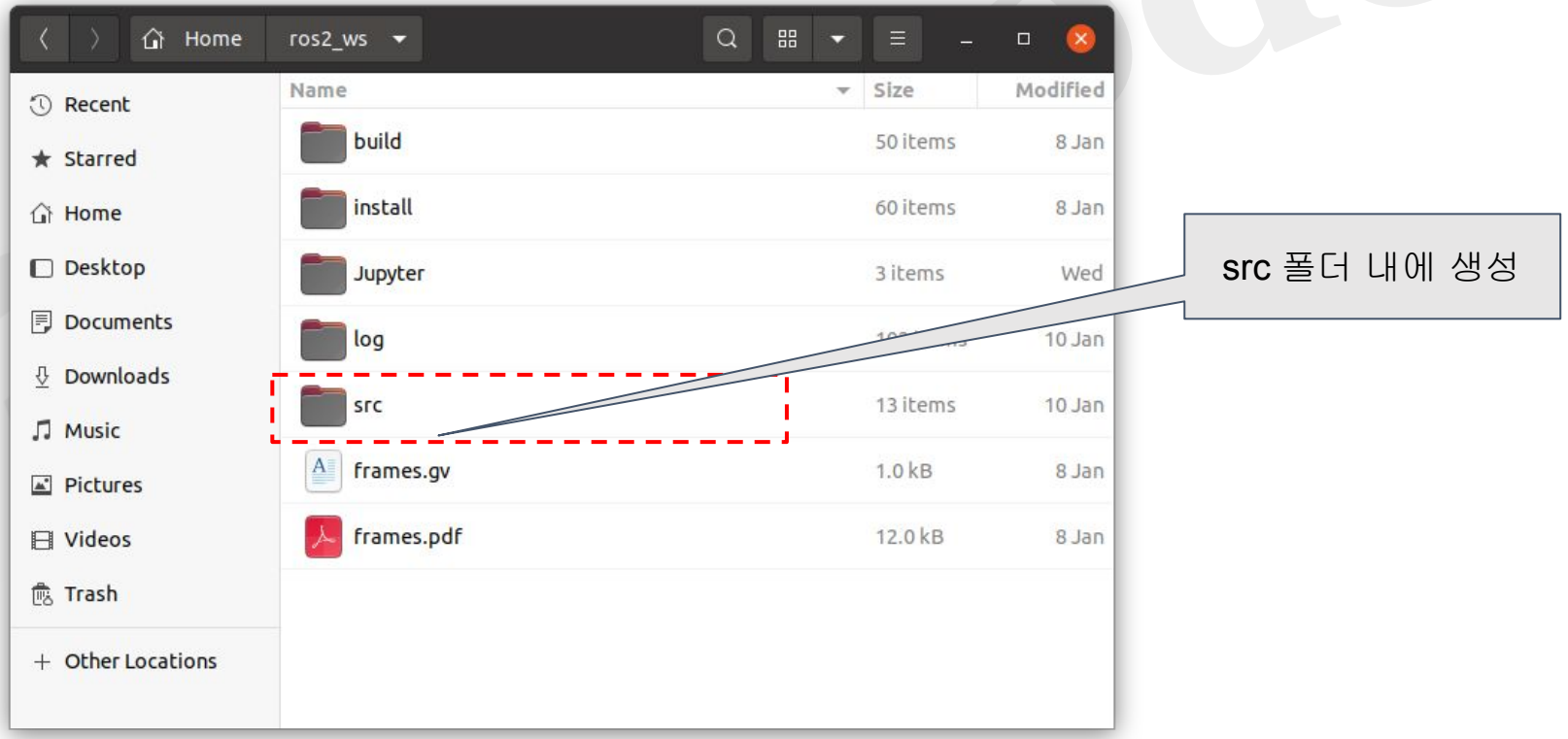


ROS2 실습

Topic publish

<twist msg cmd_vel>

```
$ cd ~/ros2_ws/src
$ ros2 pkg create --build-type ament_python py_twistpub
```



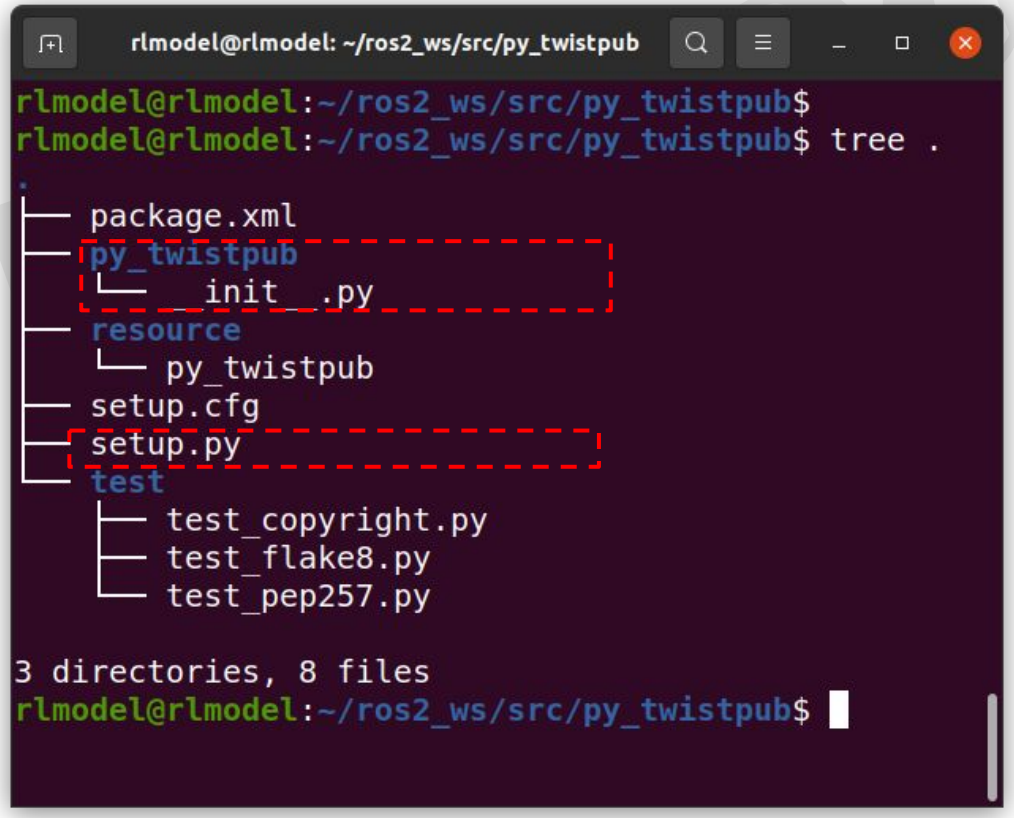
ROS2 실습

Topic publish

<twist msg cmd_vel>

$ tree .

> sudo apt-get install tree

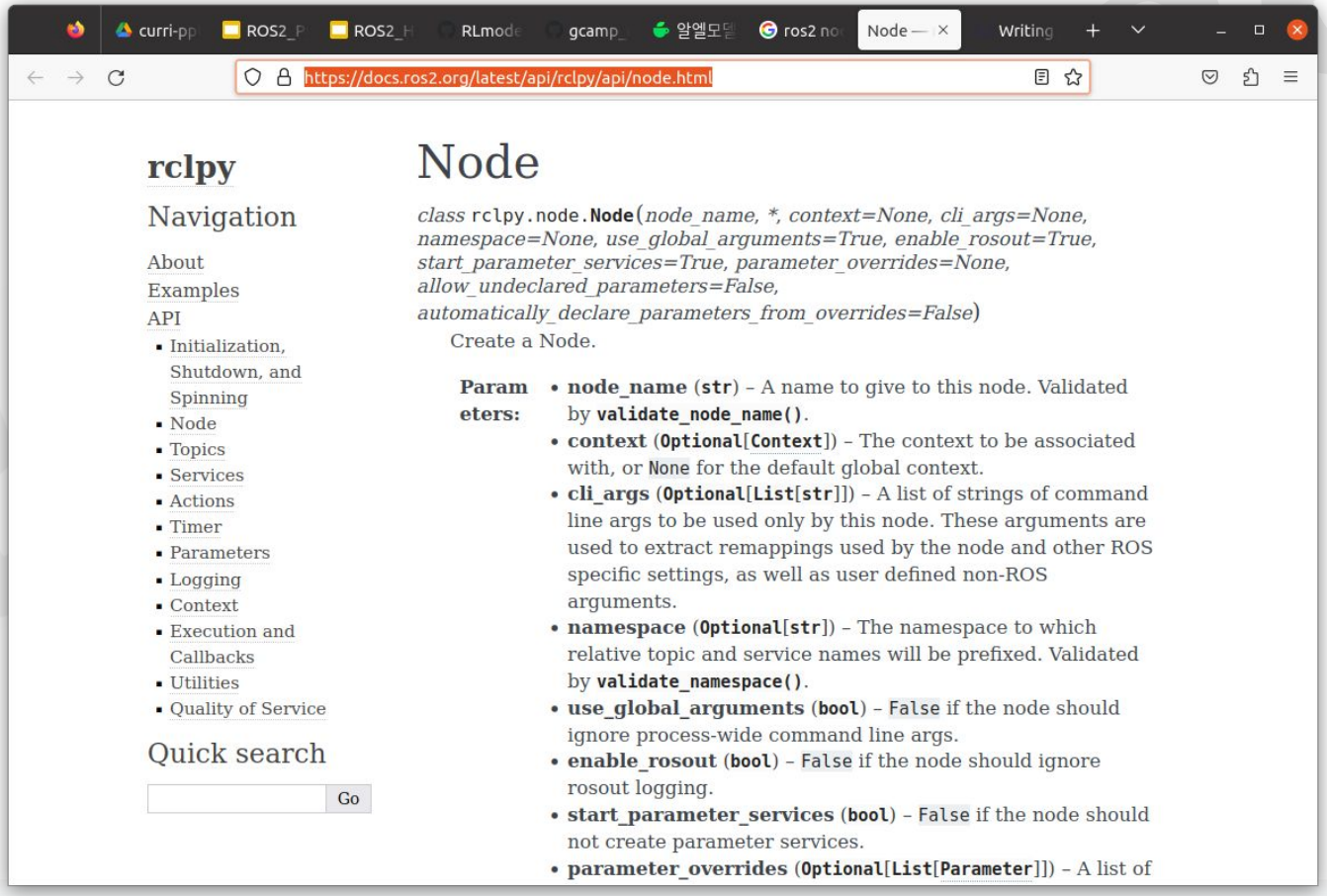


ROS2 실습

Topic publish

<twist msg cmd_vel>

link: https://docs.ros2.org/latest/api/rclpy/api/node.html



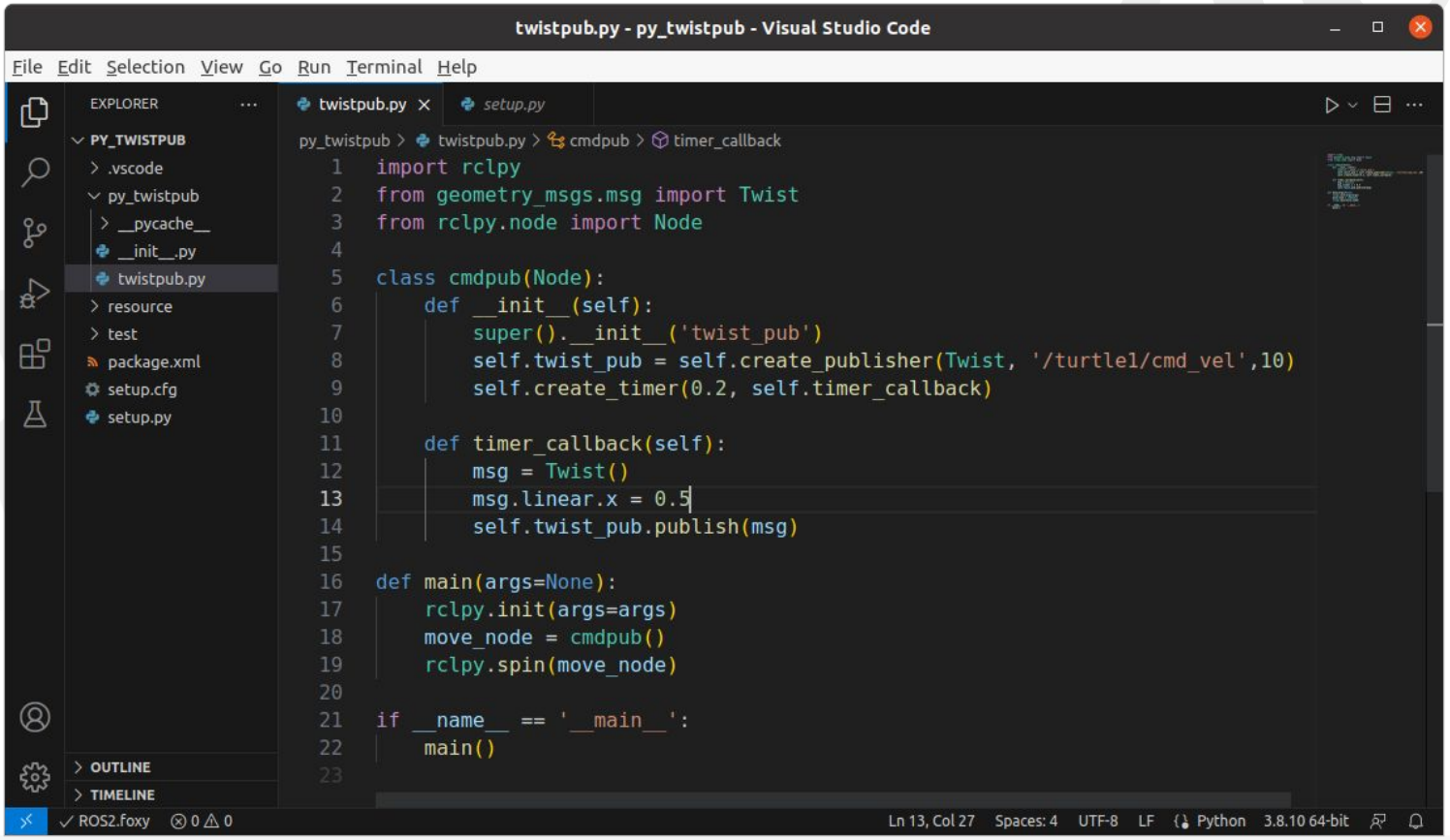
ROS2 실습

Topic publish

<twist msg cmd_vel>

$ code .

>twistpub.py



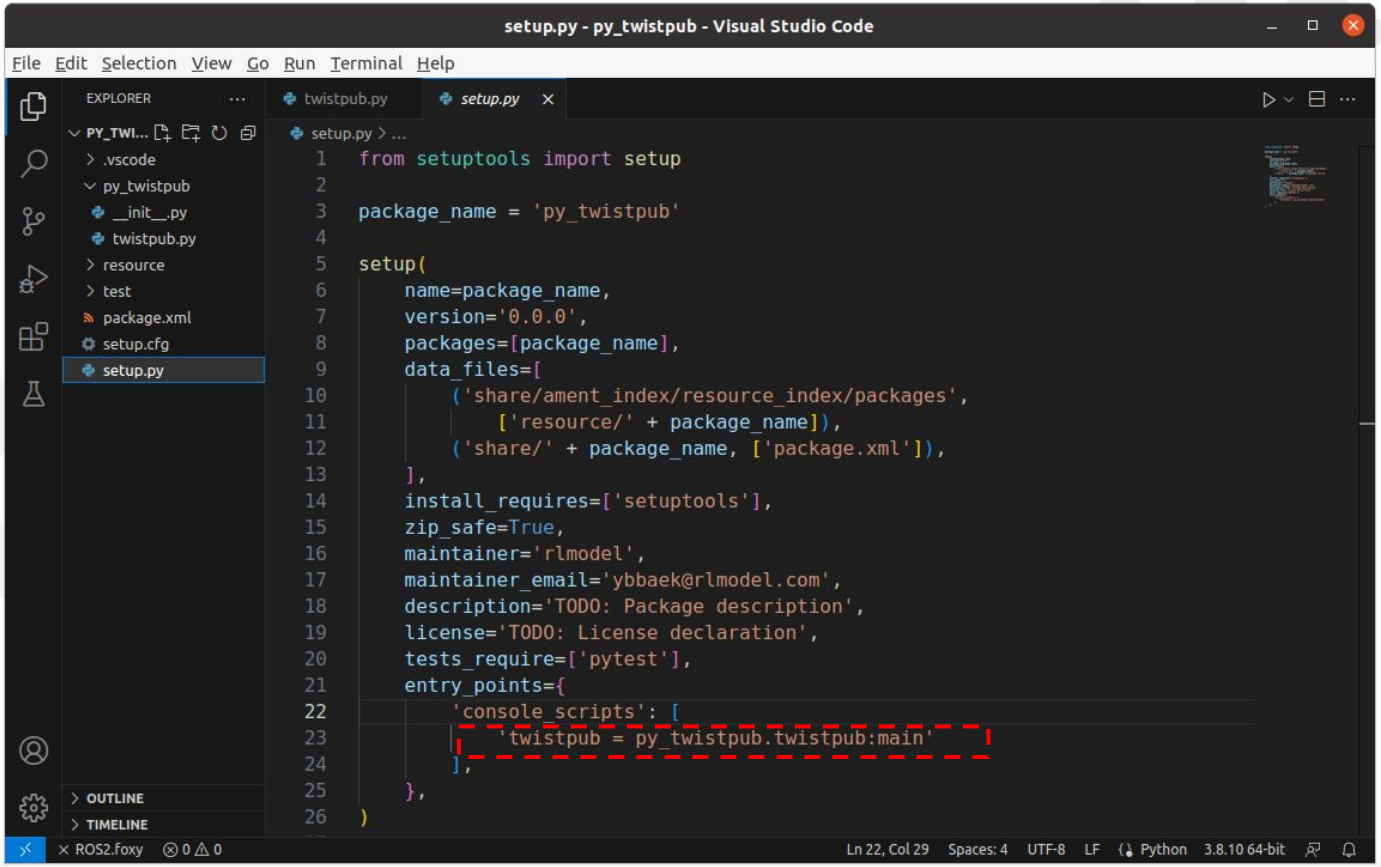
ROS2 실습

Topic publish

<twist msg cmd_vel>

$ code .

>setup.py



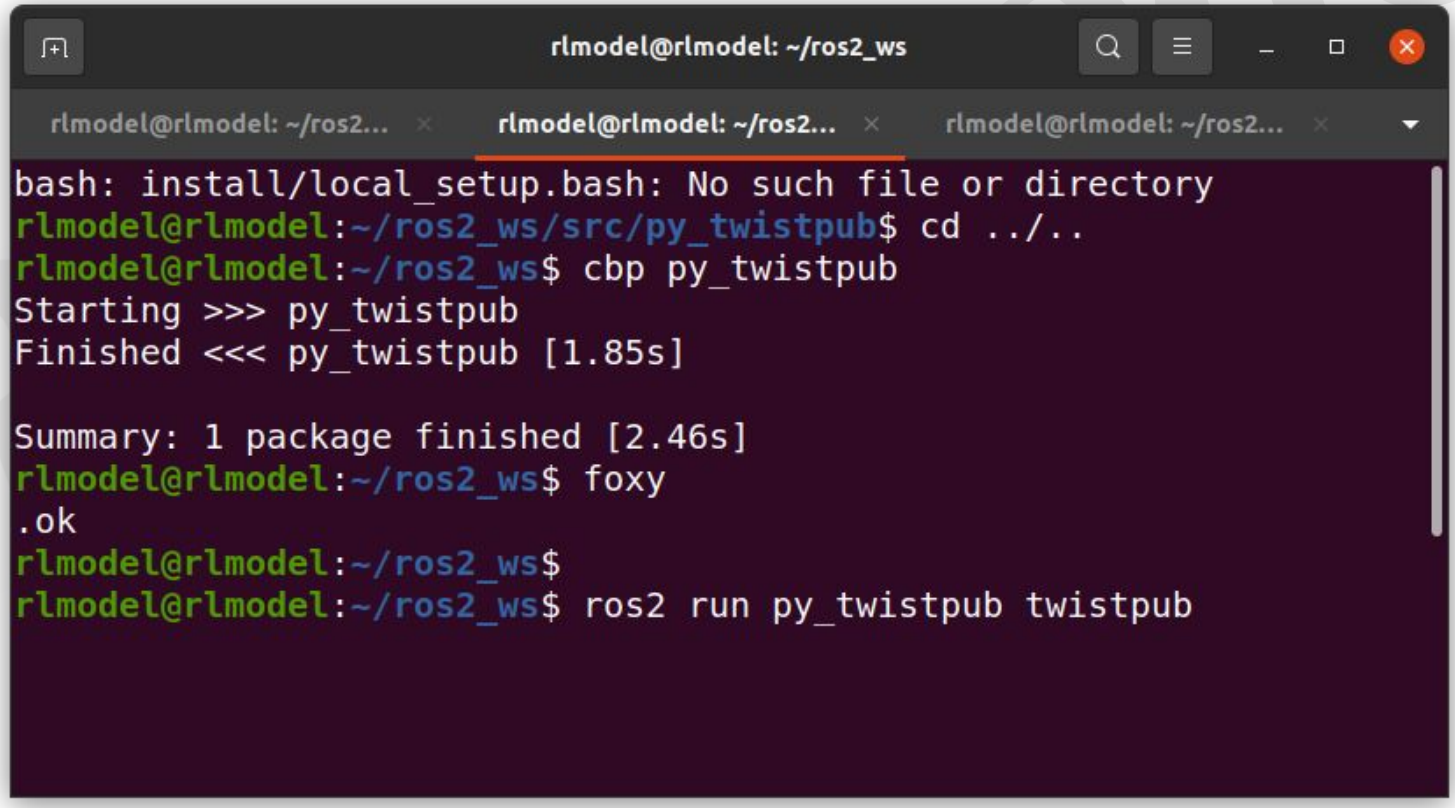
ROS2 실습

Topic publish

<twist msg cmd_vel>

$ colcon build --packages-select py_twistpub

$ ros2 run py_twistpub twistpub

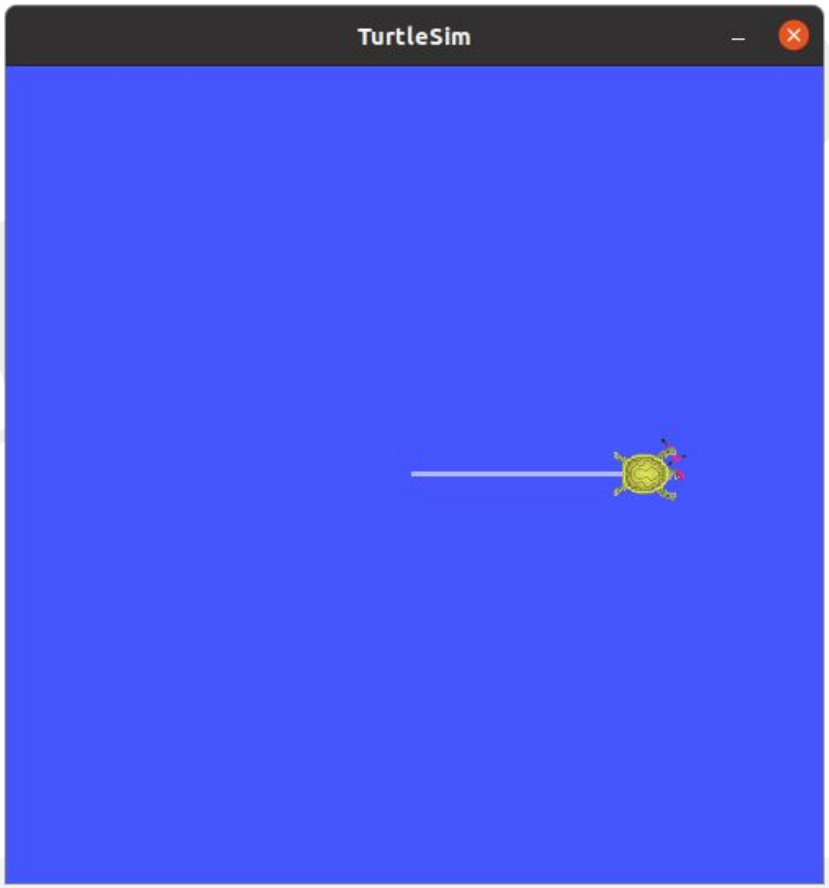


ROS2 실습

Topic publish

<twist msg cmd_vel>

```
$ colcon build --packages-select py_twistpub
$ ros2 run py_twistpub twistpub
```



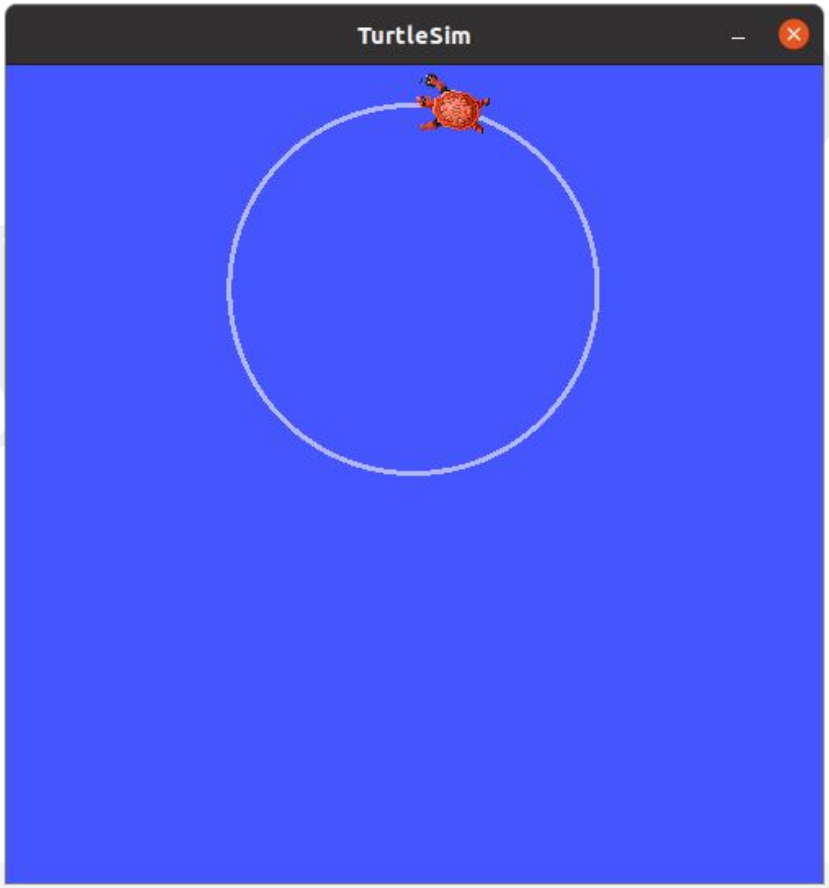
ROS2 실습

Topic publish

<twist msg cmd_vel>

$ colcon build --packages-select py_twistpub

$ ros2 run py_twistpub twistpub



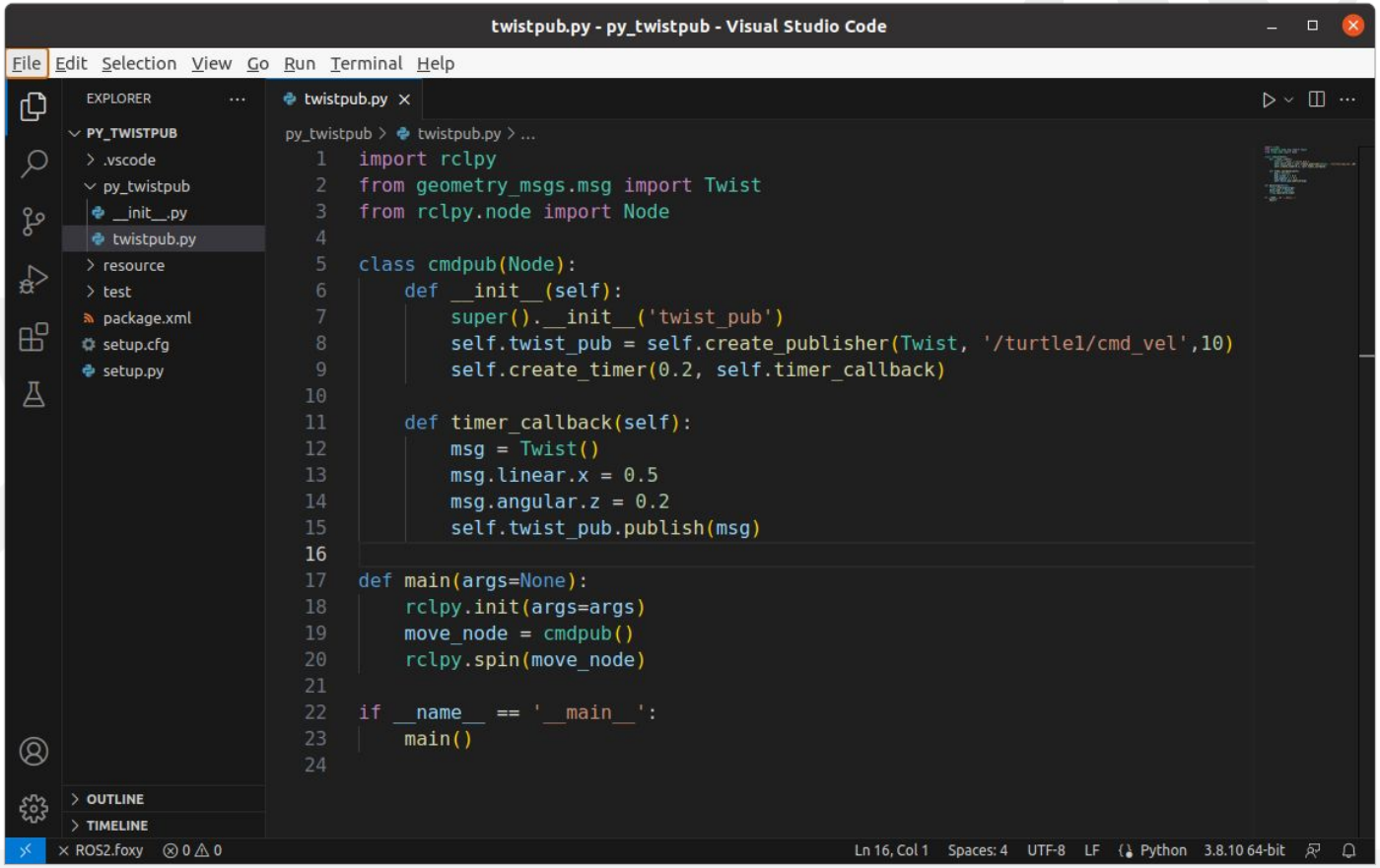
ROS2 실습

Topic publish

<twist msg cmd_vel>

$ code .

>twistpub.py

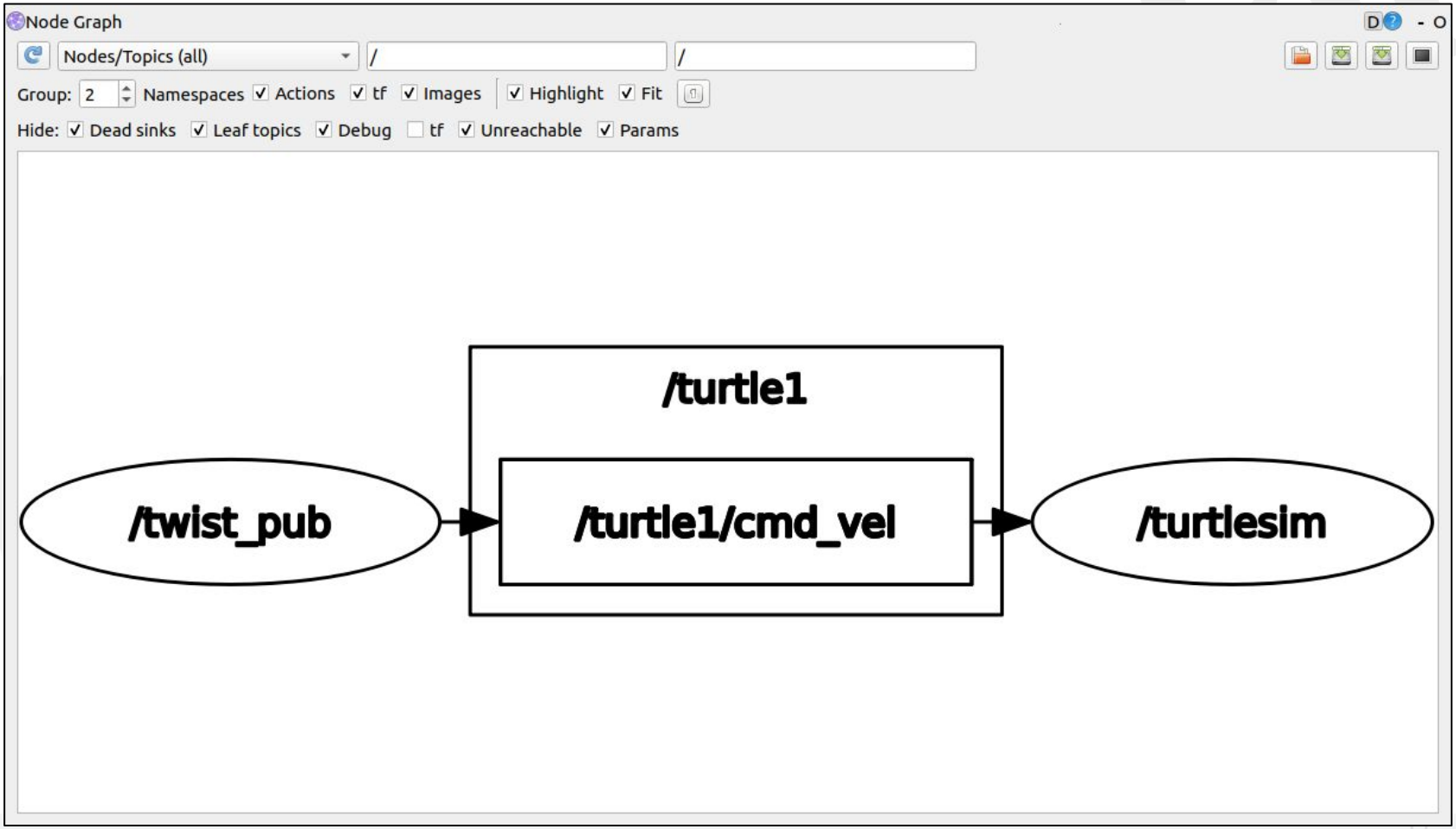


ROS2 실습

Topic publish

<twist msg cmd_vel>

$ rqt_graph

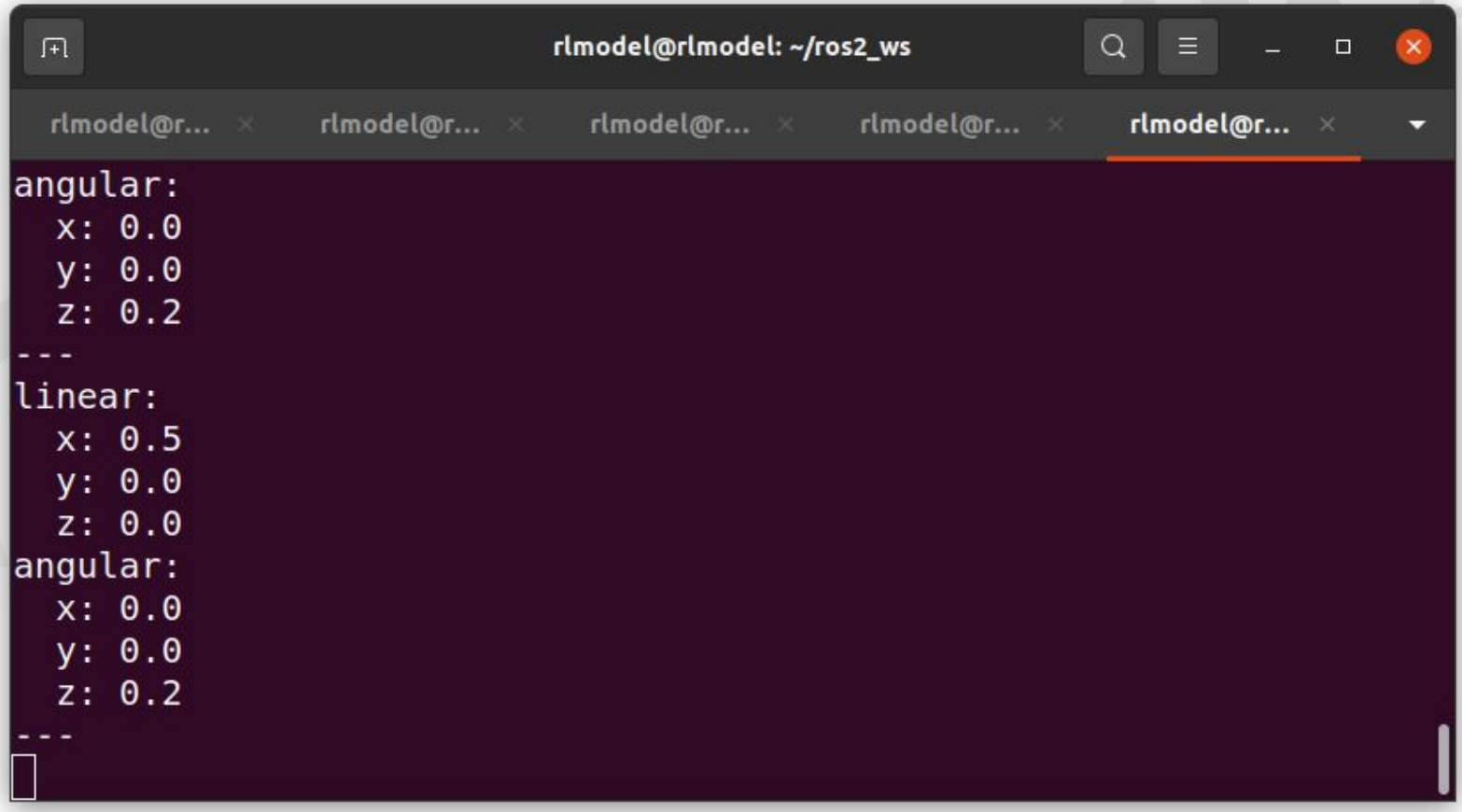


ROS2 실습

Topic publish

<twist msg cmd_vel>

$ ros2 topic echo /turtle1/cmd_vel

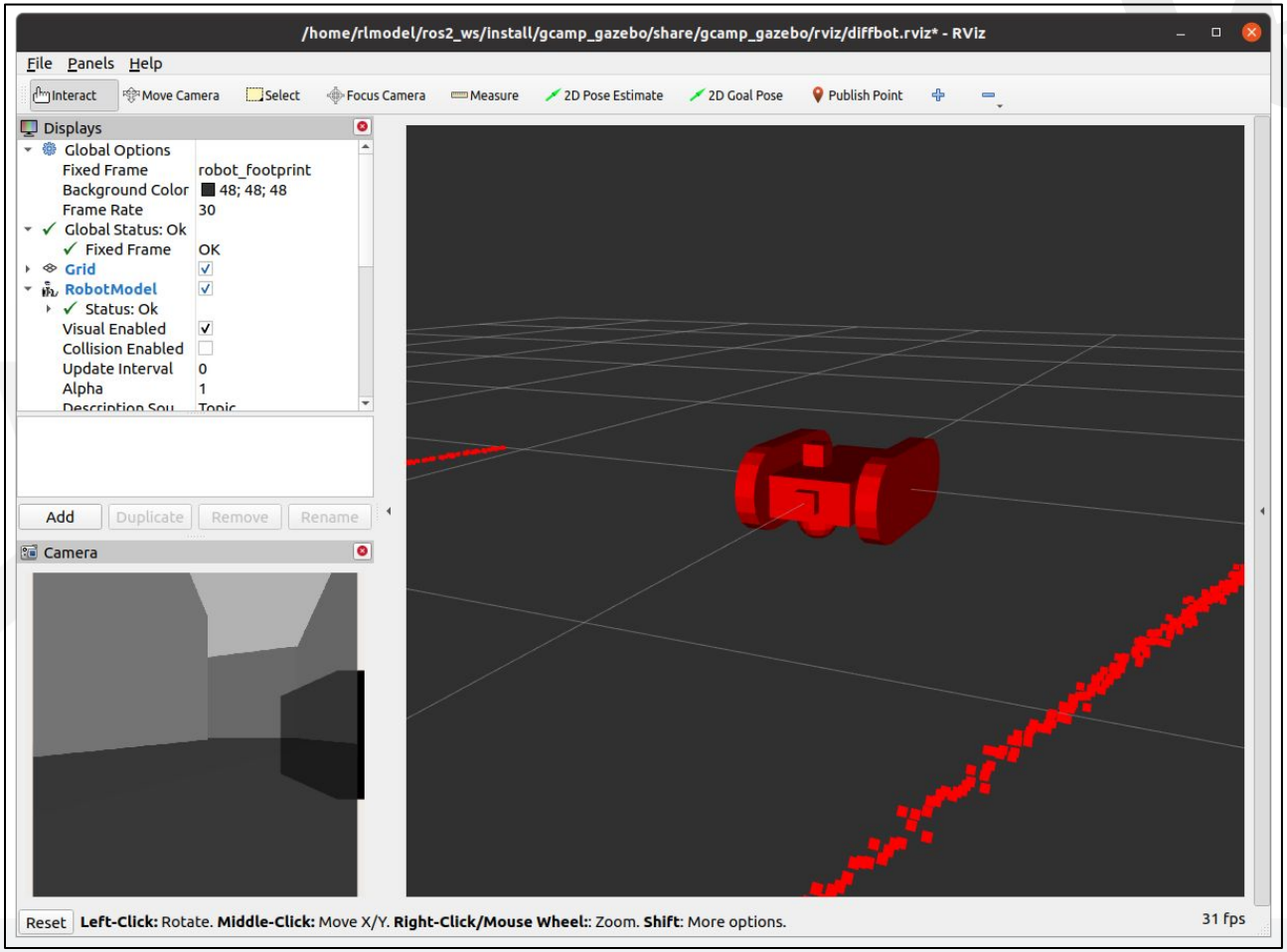


ROS2 실습

Topic publish

<twist msg cmd_vel>

example > diffbot turning



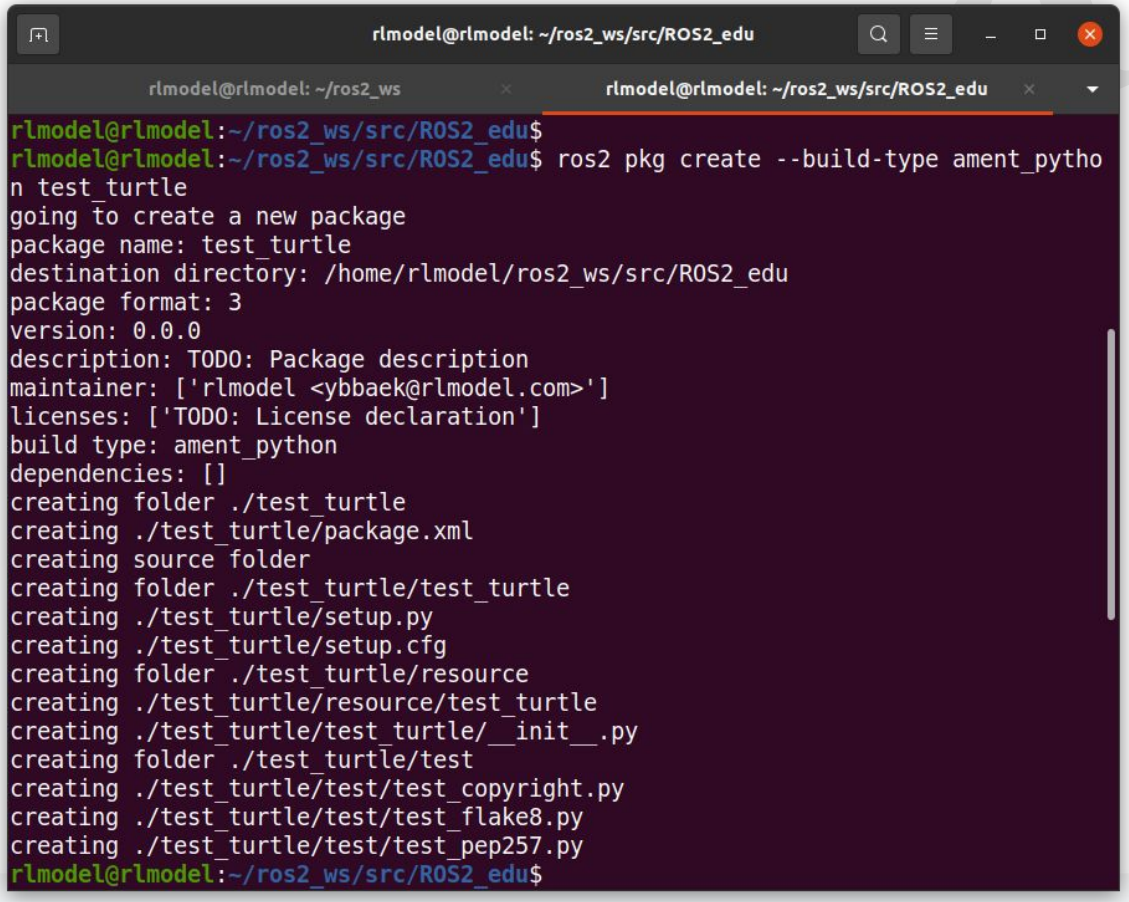
Break Time

ROS2 실습

Package 생성

<turtlesim_subscriber>

>new pkg - test_turtle

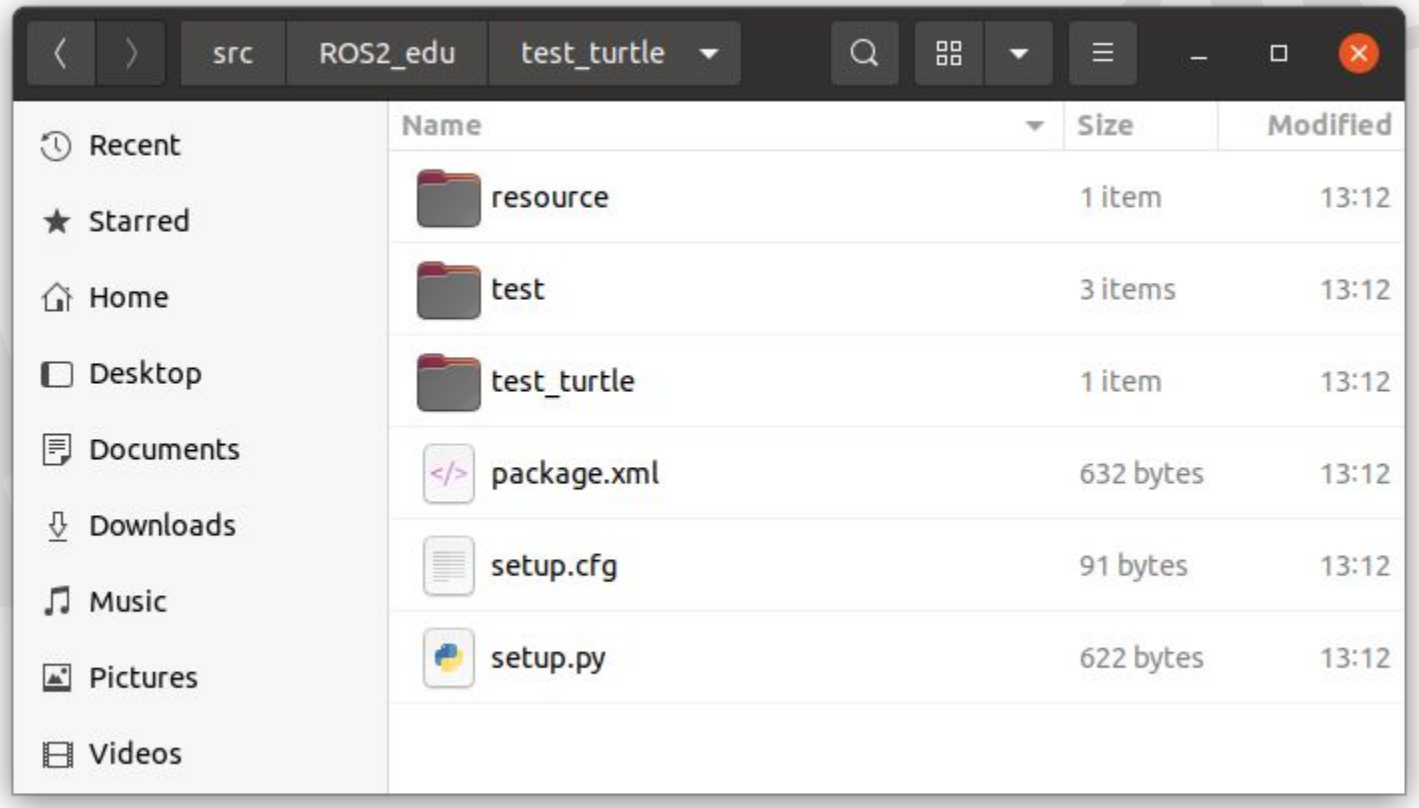


ROS2 실습

Package 생성

<turtlesim_subscriber>

>check folder & files



ROS2 실습

Package 생성

<turtlesim_subscriber>

>check folder & files

Open | package.xml ~/ros2_ws/src/ROS2_edu/test_turtle | Save

```
<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd"
schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>test_turtle</name>
  <version>0.0.0</version>
  <description>TODO: Package description</description>
  <maintainer email="ybbaek@rlmodel.com">rlmodel</maintainer>
  <license>TODO: License declaration</license>

  <test_depend>ament_copyright</test_depend>
  <test_depend>ament_flake8</test_depend>
  <test_depend>ament_pep257</test_depend>
  <test_depend>python3-pytest</test_depend>

  <export>
    <build_type>ament_python</build_type>
  </export>
</package>
```

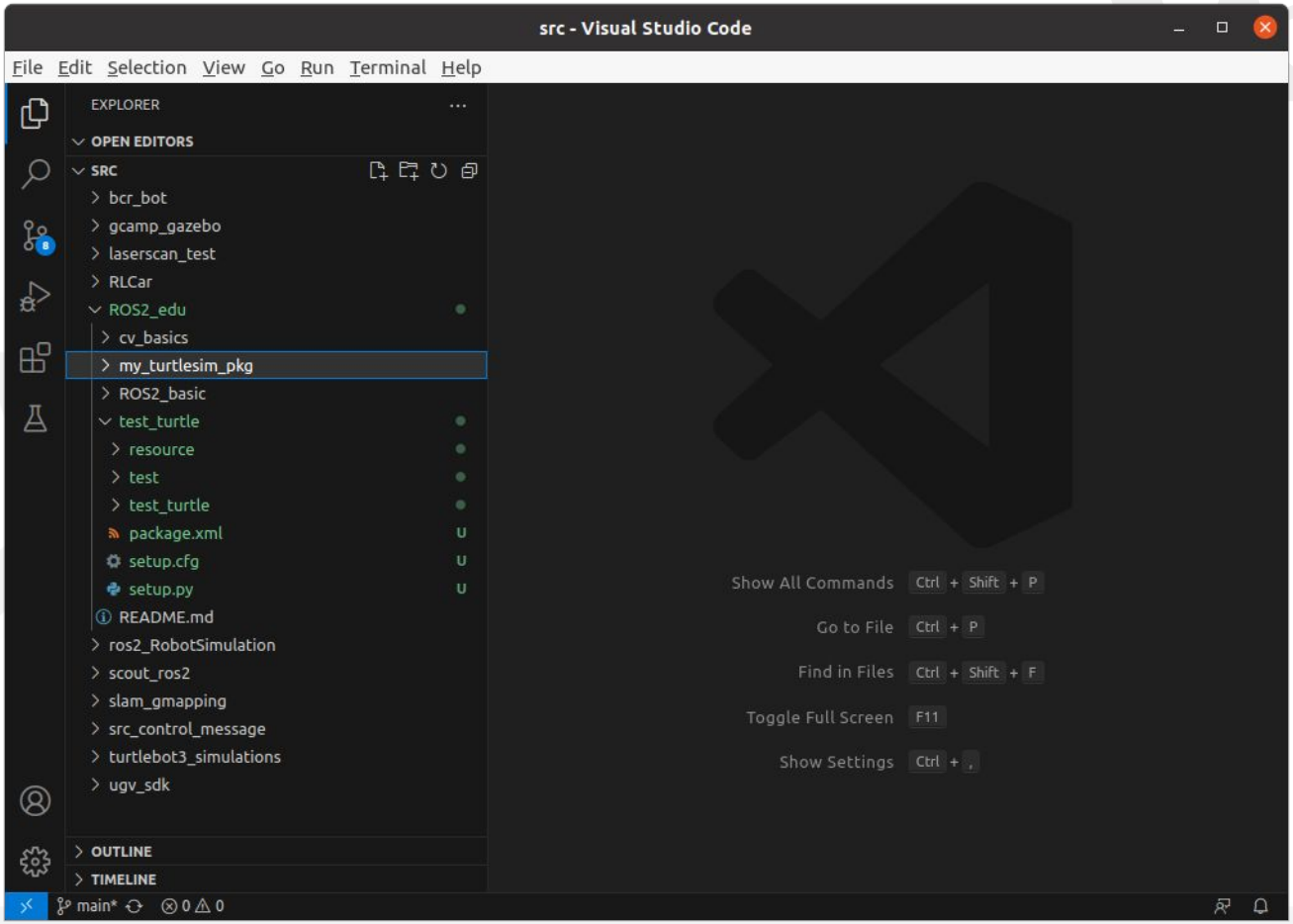
XML | Tab Width: 8 | Ln 1, Col 1 | INS

ROS2 실습

Package 생성

<turtlesim_subscriber>

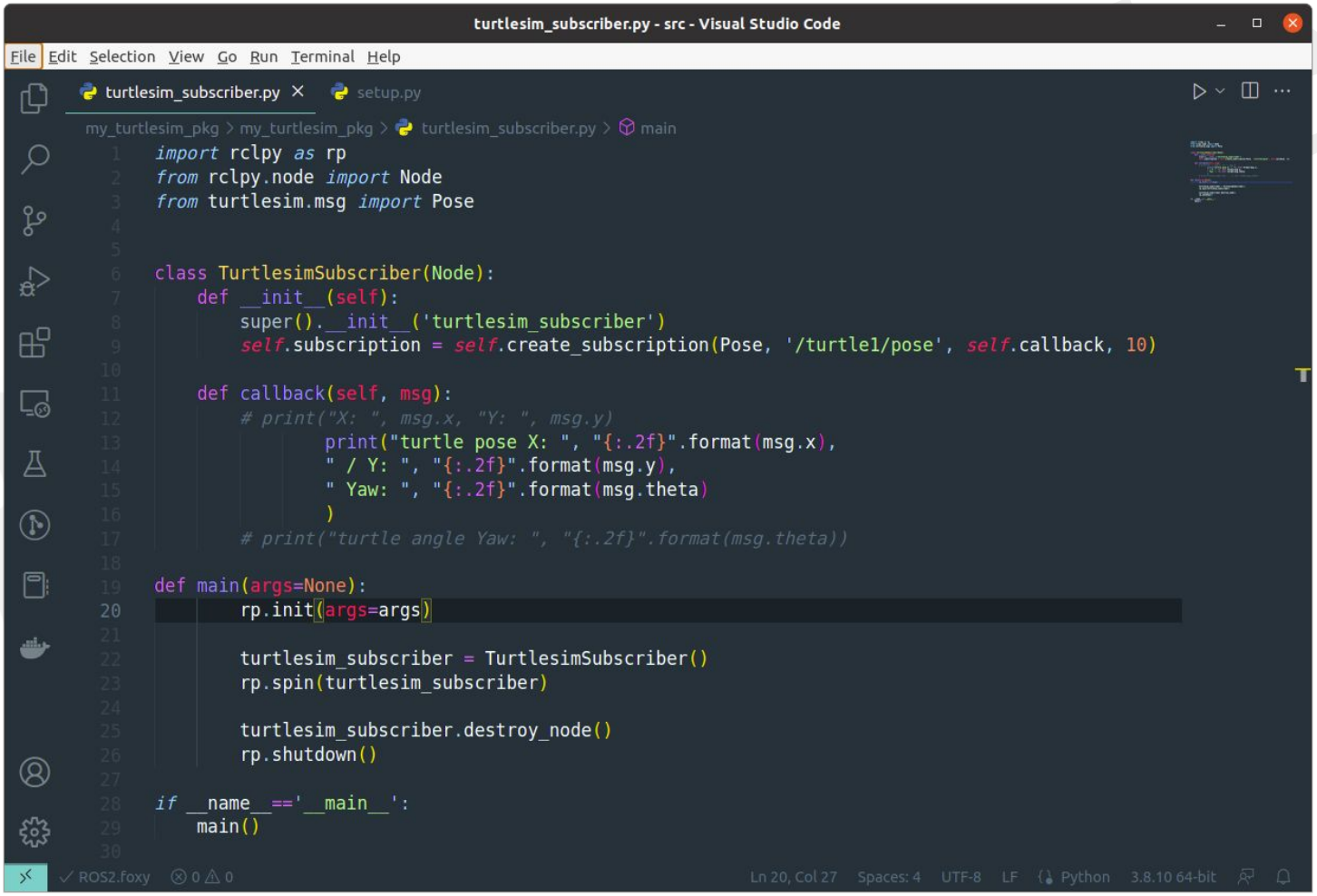
$ code .



ROS2 실습

Python code 생성

- turtlesim_subscriber.py



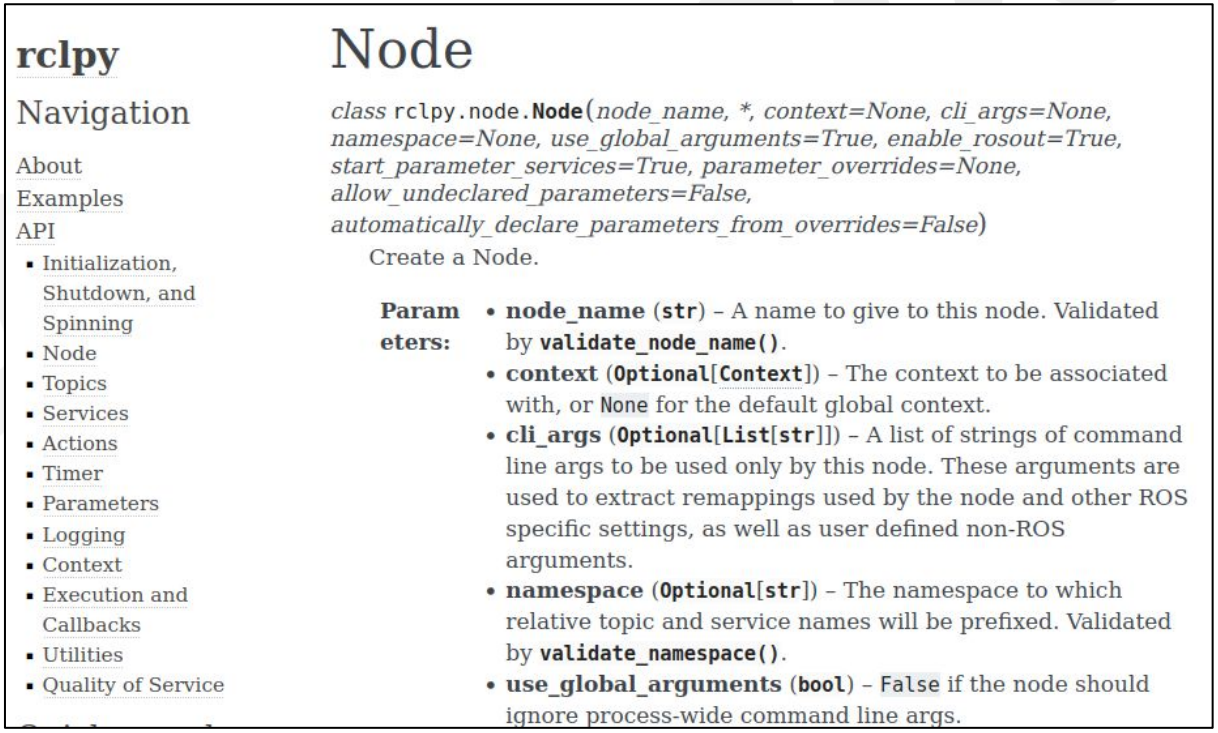
ROS2 실습

Python code 생성

- turtlesim_subscriber.py

Node class 정보

link: https://docs.ros2.org/latest/api/rclpy/api/node.html



rclpy

Navigation

About
Examples
API
- Initialization, Shutdown, and Spinning
- Node
- Topics
- Services
- Actions
- Timer
- Parameters
- Logging
- Context
- Execution and Callbacks
- Utilities
- Quality of Service

Node

class rclpy.node.**Node**(*node_name, *, context=None, cli_args=None, namespace=None, use_global_arguments=True, enable_rosout=True, start_parameter_services=True, parameter_overrides=None, allow_undeclared_parameters=False, automatically_declare_parameters_from_overrides=False*)

Create a Node.

Parameters:
- **node_name** (**str**) – A name to give to this node. Validated by **validate_node_name()**.
- **context** (**Optional**[**Context**]) – The context to be associated with, or None for the default global context.
- **cli_args** (**Optional**[**List**[**str**]]) – A list of strings of command line args to be used only by this node. These arguments are used to extract remappings used by the node and other ROS specific settings, as well as user defined non-ROS arguments.
- **namespace** (**Optional**[**str**]) – The namespace to which relative topic and service names will be prefixed. Validated by **validate_namespace()**.
- **use_global_arguments** (**bool**) – False if the node should ignore process-wide command line args.

ROS2 실습

dependency 확인

```
# cd if you're still in the ``src`` directory with the ``ros_tutorials`` clone
```

- cd ..
- rosdep install -i --from-path src --rosdistro foxy -y

colcon build 실행

- colcon build

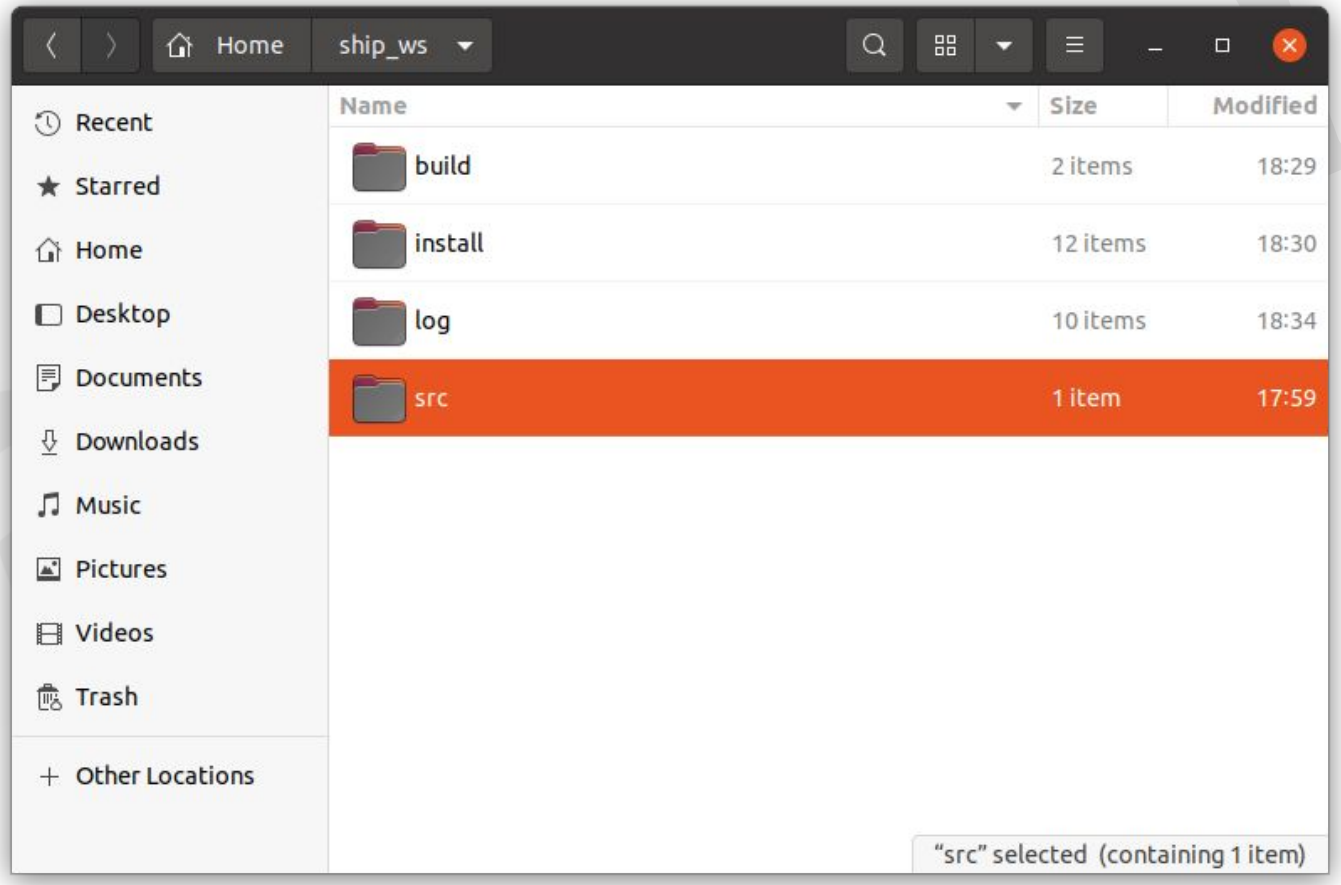
<결과>

- build, install, log 폴더 생성

ROS2 실습

build 결과확인

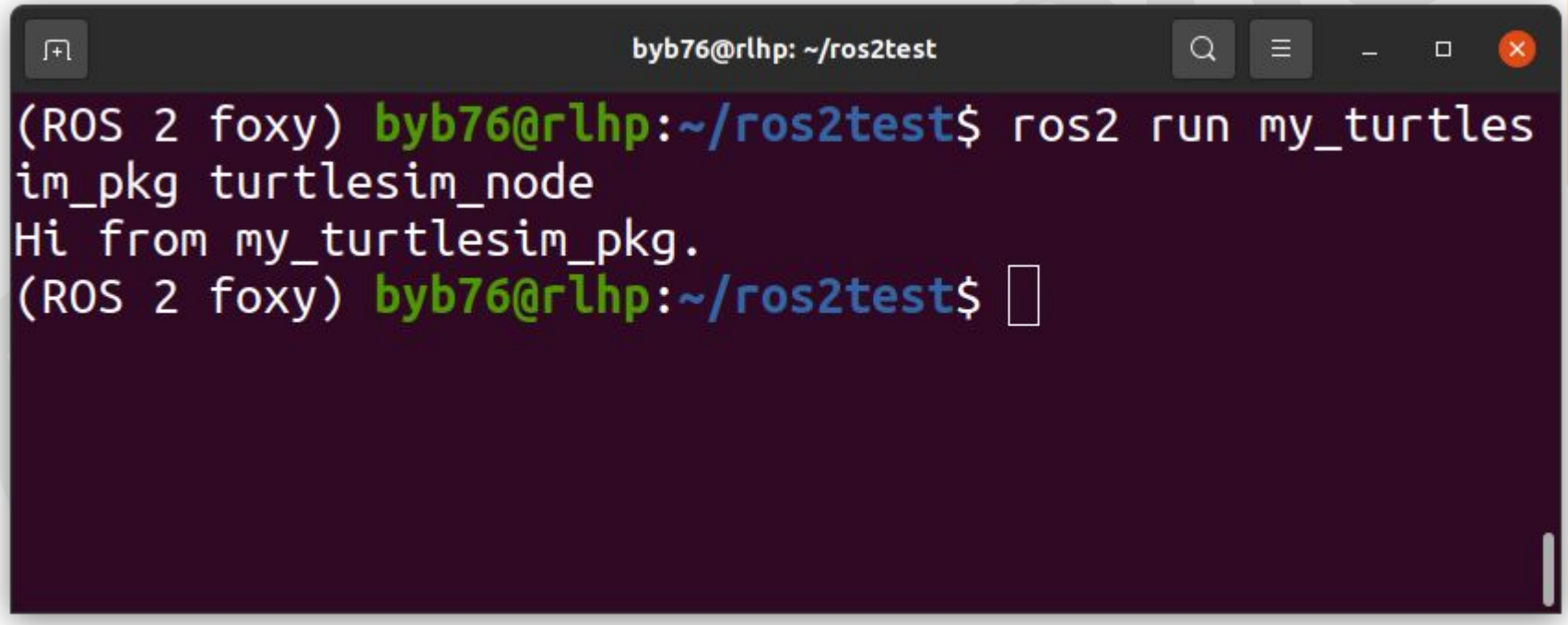
- build, install, log 파일생성



ROS2 실습

package 실행

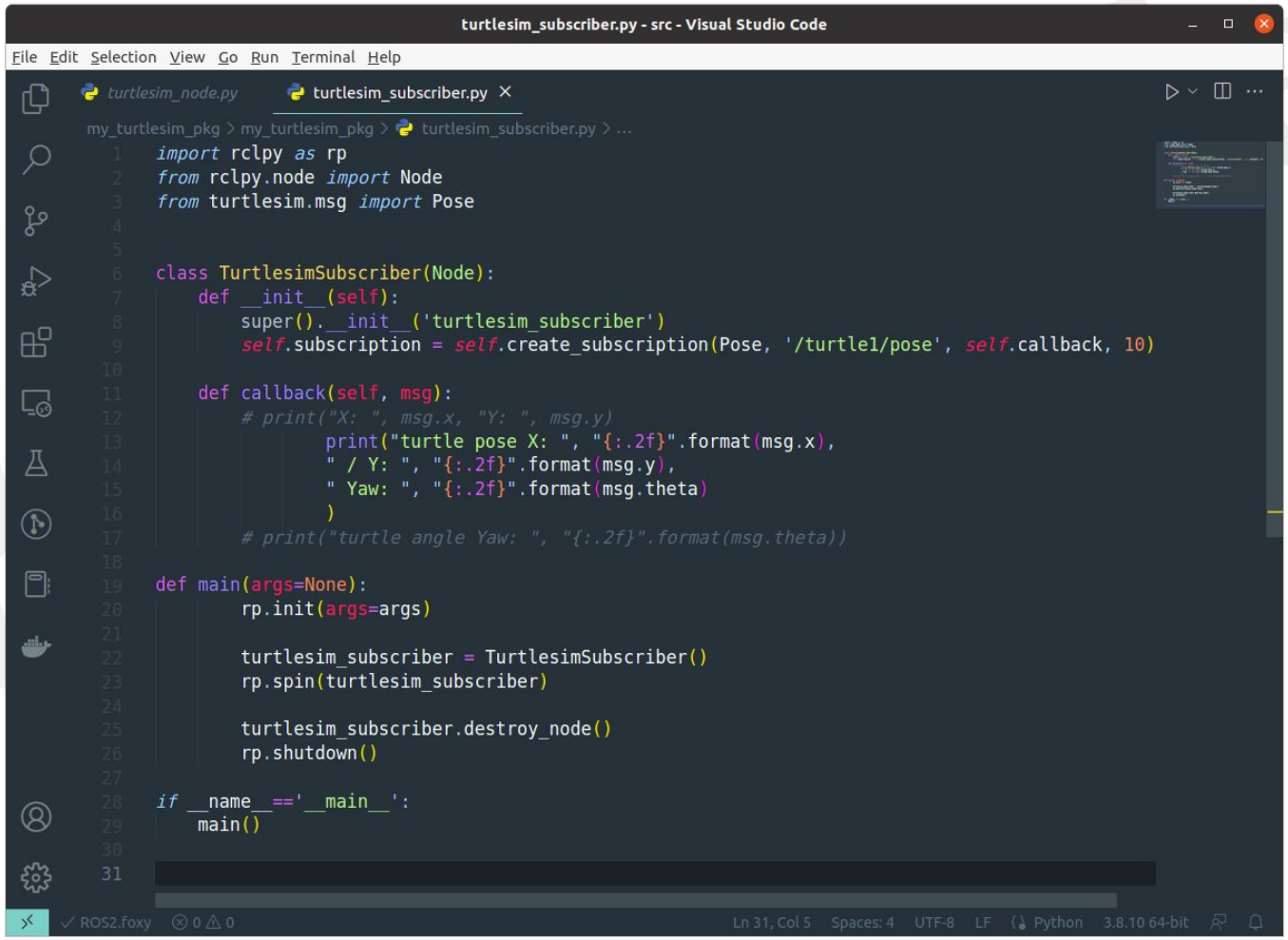
- . install/local_setup.bash
- ros2 run my_turtlesim_pkg turtle_subscriber



ROS2 실습

turtlesim topic subscriber 노드 생성

파이선 파일 생성: turtlesim_subscriber.py



ROS2 실습

turtlesim topic subscriber 노드 생성

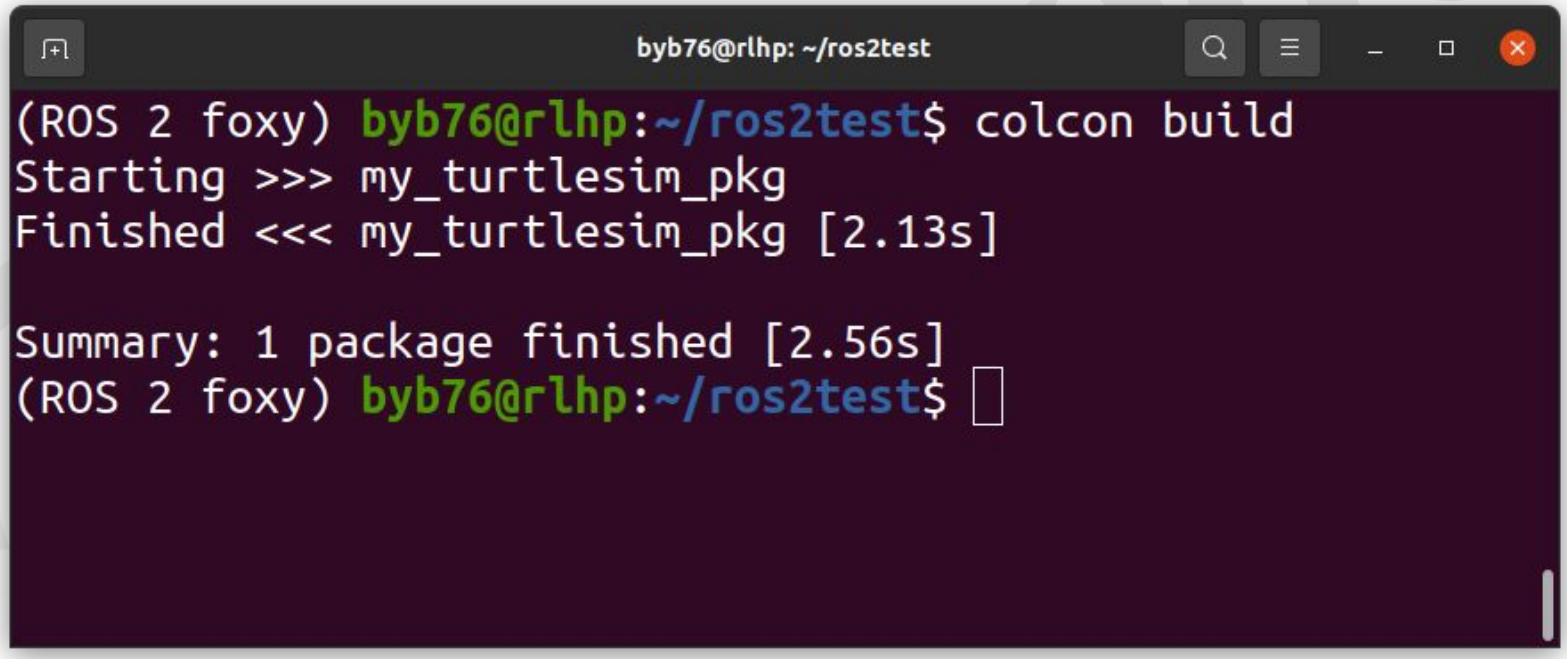
setup.py 업데이트: entrypoint 추가 (, 주의)

ROS2 실습

package build

$colcon build

$. install/localsetup.bash



ROS2 실습

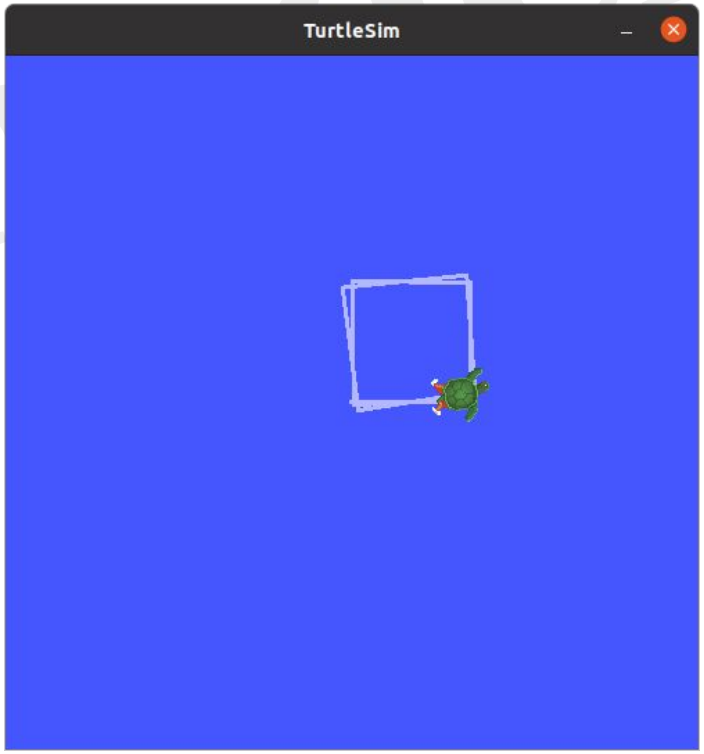
package test

$colcon build

$. install/localsetup.bash

$ ros2 run turtlesim turtlesim_node

$ ros2 run turtlesim draw_square



ROS2 실습

package 실행

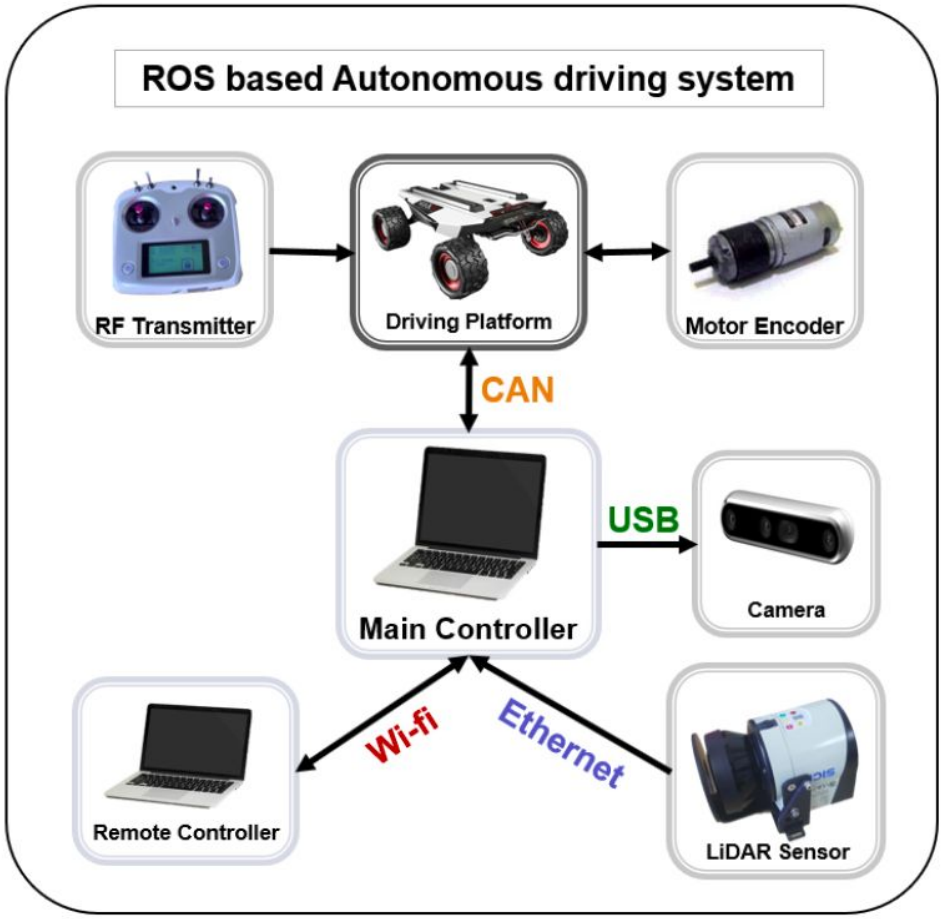
$ros2 run my_turtlesim_pkg turtle_subscriber

>위 이름은 setup.py entrypoint 이름과 일치

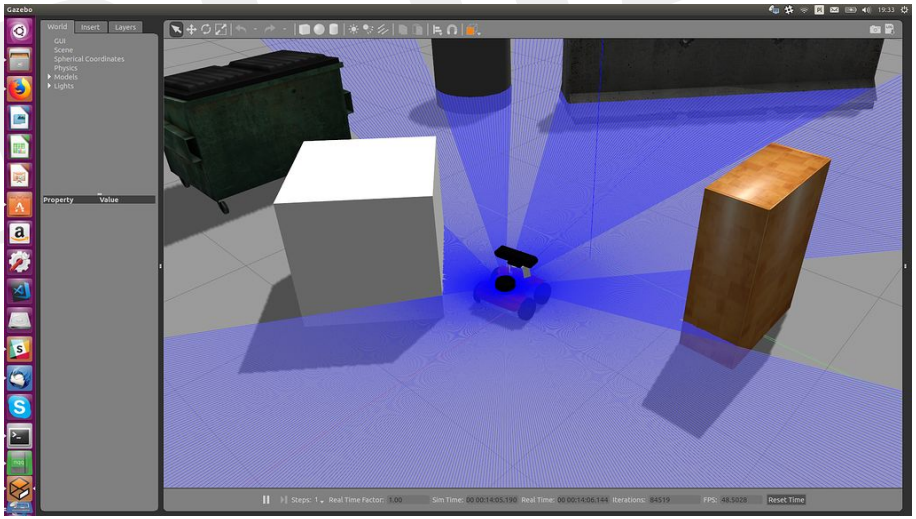
byb76@rlhp: ~/ros2test

```
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
turtle pose X:  0.00  / Y:  5.37  Yaw:  -2.96
```

ROS2 laser scan



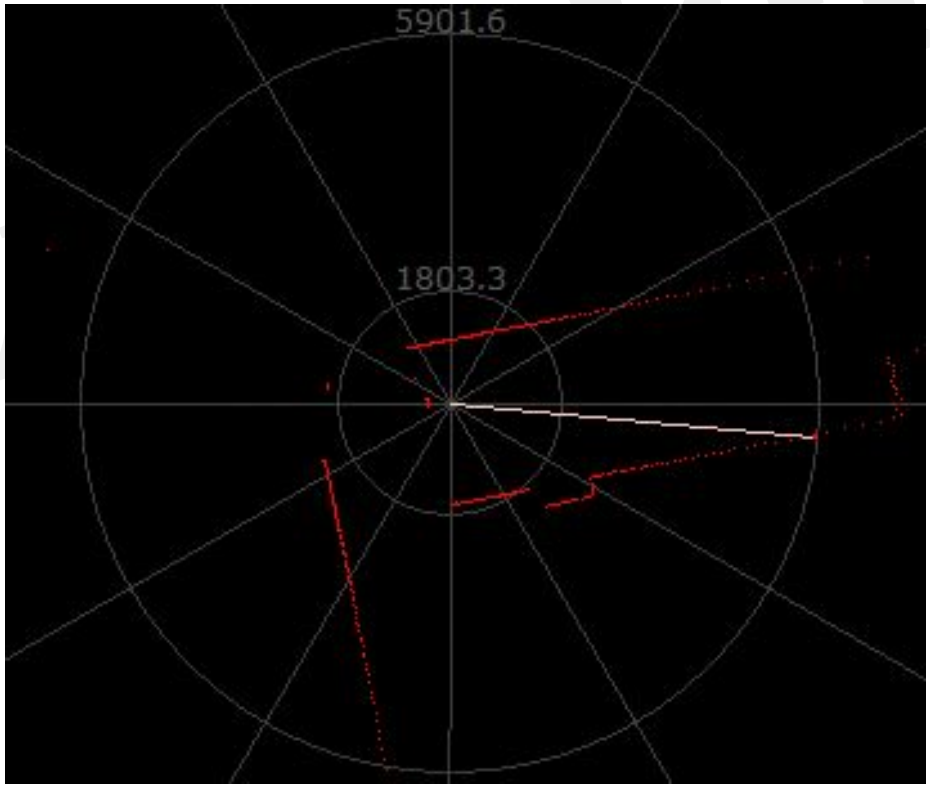
Lidar sensor
실습



ROS2 HW

Lidar

demo - rplidar a2

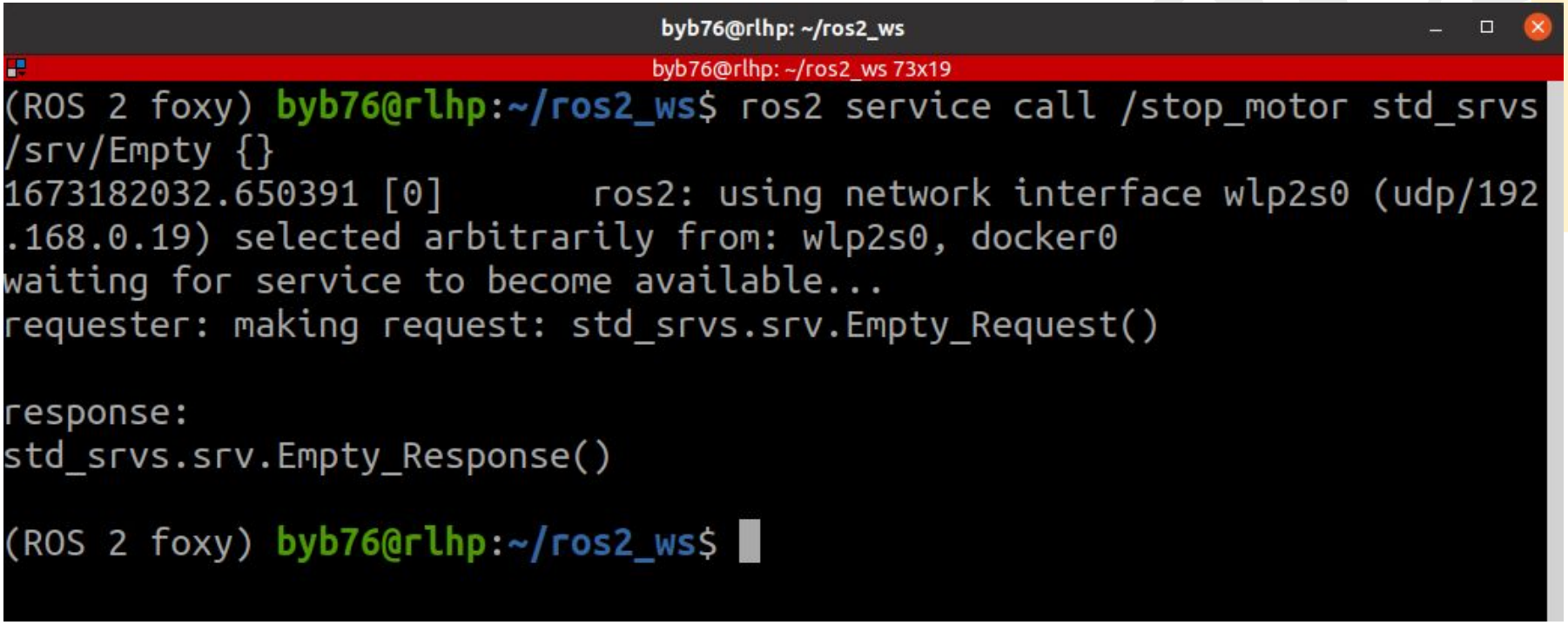


ROS2 구성

Service

demo - rplidar a2

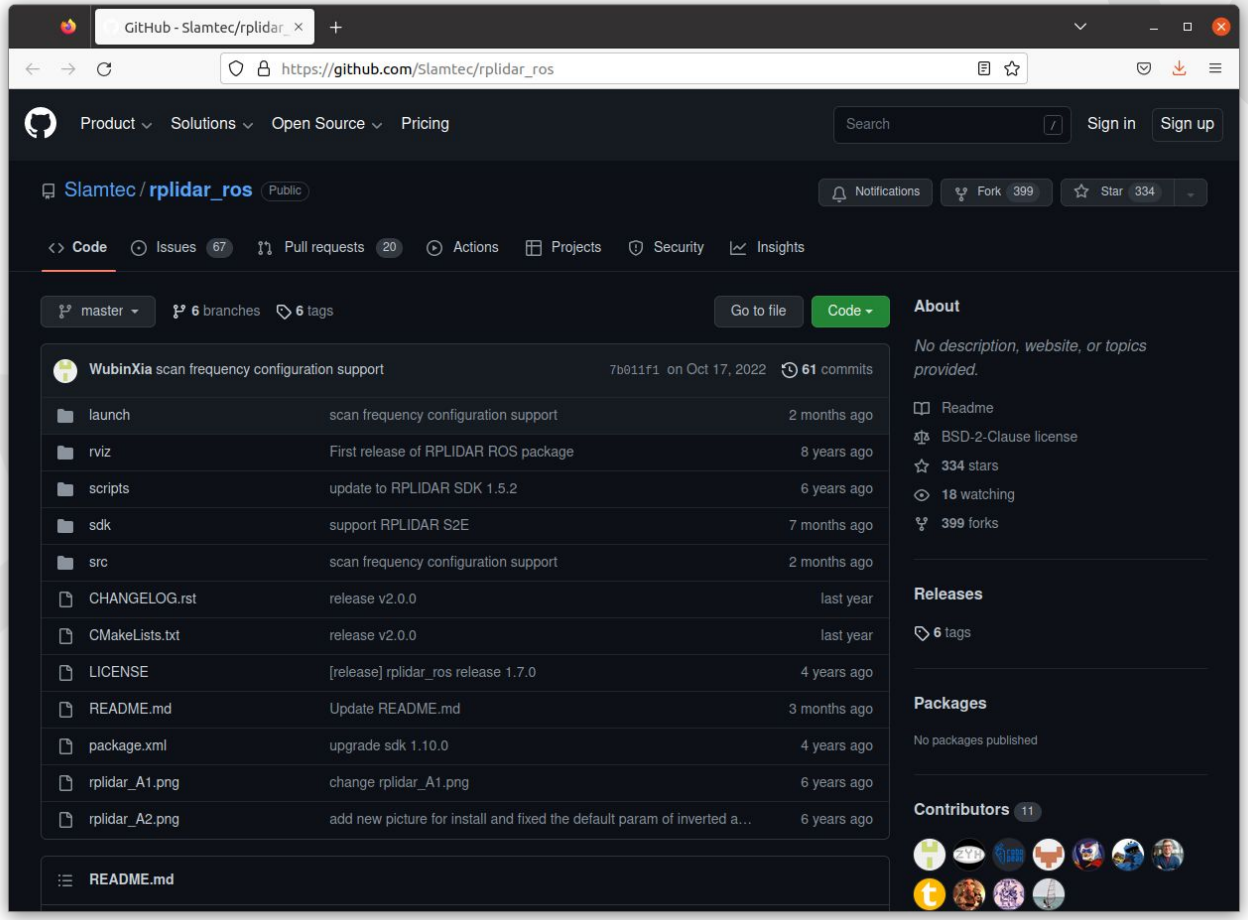
$ ros2 service call /stop_motor std_srvs/srv/Empty {}



ROS2 HW

RPLidar

link: https://github.com/Slamtec/rplidar_ros



ROS2 HW

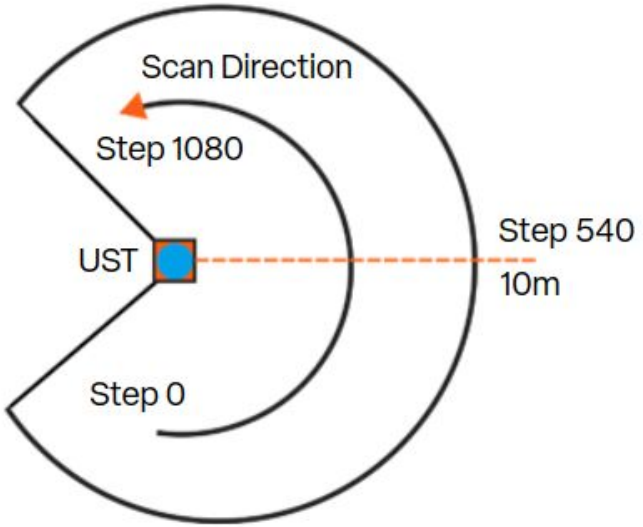
dummy_laser

Hukuyo 2D lidar

Measuring Steps 1081

Detection Angle 270°

Angular Resolution 0.25°



ROS2 HW

dummy_laser

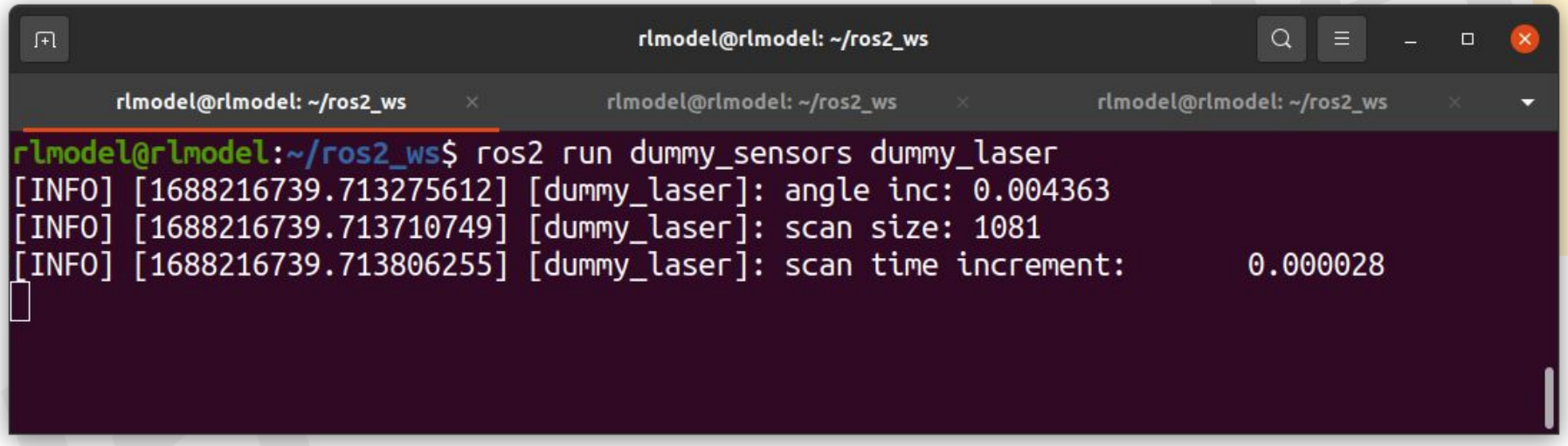
Hukuyo 2D lidar

Product Name	Scanning Laser Range Finder
Model	UST-10LX
Supply voltage	12VDC/24VDC (Operation range 10 to 30V ripple within 10%)
Supply current	150mA or less (When using DC24V) (during start up 450mA is necessary.)
Light source	Laser semiconductor (905nm) Laser class 1 (IEC60825-1:2007)
Detection range	0.06m to 10m (white Kent sheet) 0.06m to 4m (diffuse reflectance 10%) Max. detection distance : 30m
Accuracy	±40mm (*1)
Repeated accuracy	σ< 30mm (*1)
Scan angle	270°
Scan speed	25ms (Motor speed 2400rpm)
Angular resolution	0.25°
Start up time	Within 10 sec (start up time differs if malfunction is detected during start up)
Input	IP reset input, photo-coupler input (current 4mA at ON)
Output	Synchronous Output, photo coupler open collector output 30VDC 50mA MAX.
Interface	Ethernet 100BASE-TX
LED display	Power supply LED display (Blue): Blinks during start up and malfunction state.
Surrounding intensity	Less than 80,000lx Note : Avoid direct sunlight or other illumination sources as it may cause sensor malfunction
Ambient temperature and humidity	-10°C to +50°C, below 85%RH (without dew, frost)
Storage temperature and humidity	-30°C to +70°C, below 85%RH (without dew, frost)
Vibration resistance	10 to 55Hz double amplitude of 1.5mm for 2hrs in each X, Y, and Z direction 55 to 200Hz 98m / s2 sweep of 2min for 1hr in each X,Y and Z direction
Vibration resistance (Operating)	55 to 150Hz 19.6m / s2 sweep of 2min for 30min in each X,Y and Z direction
Shock resistance	196m/s2 (20G) X,Y and Z direction 10 times.

ROS2 HW

dummy_laser

$ ros2 run dummy_sensors dummy_laser

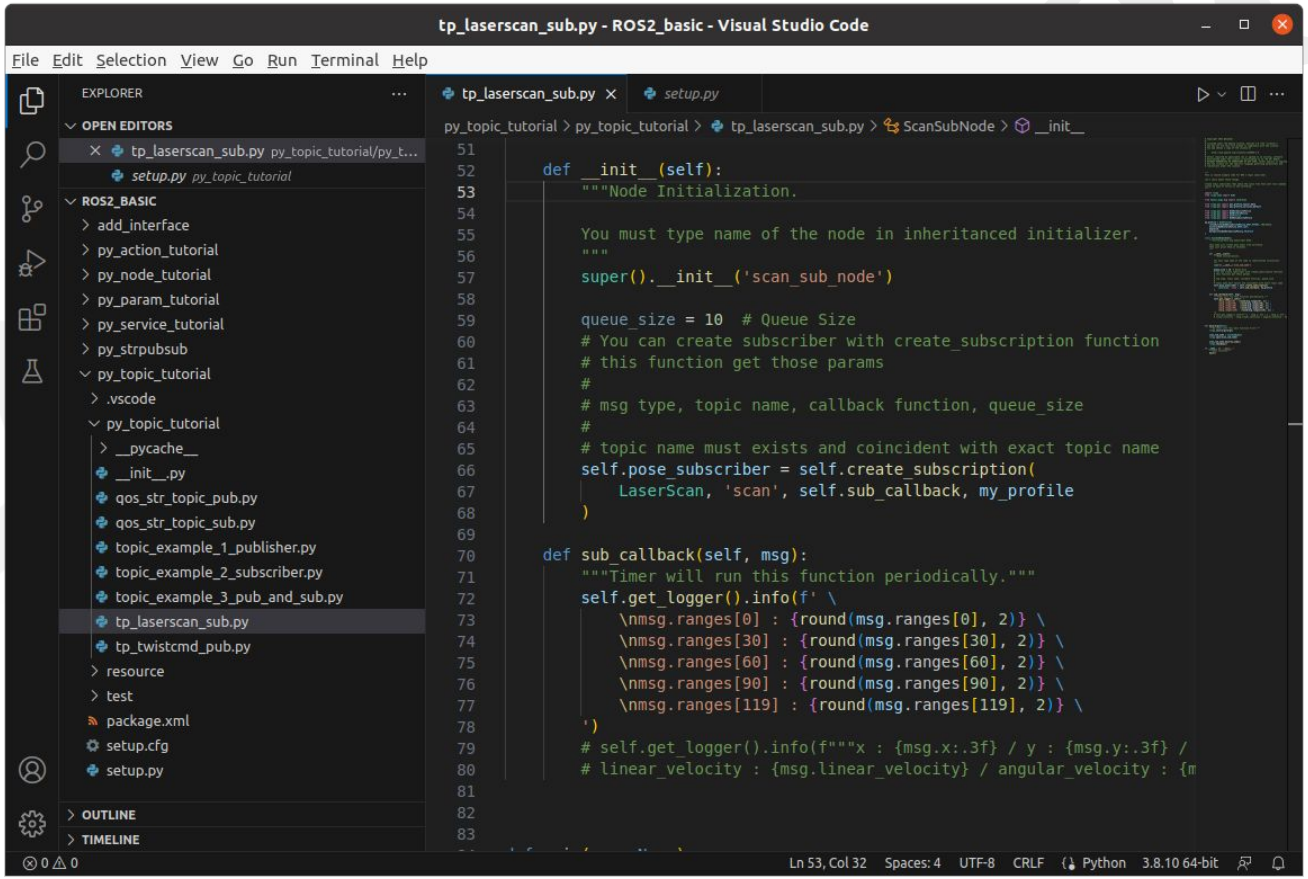


ROS2 HW

dummy_laser

>new python file - publisher

ROS2_basic/py_topic_tutorial/py_topic_tutorial/tp_laserscan_sub.py



ROS2 HW

LaserScan msg

>link:http://docs.ros.org/en/melodic/api/sensor_msgs/html/msg/LaserScan.html

sensor_msgs/LaserScan Message

File: sensor_msgs/LaserScan.msg

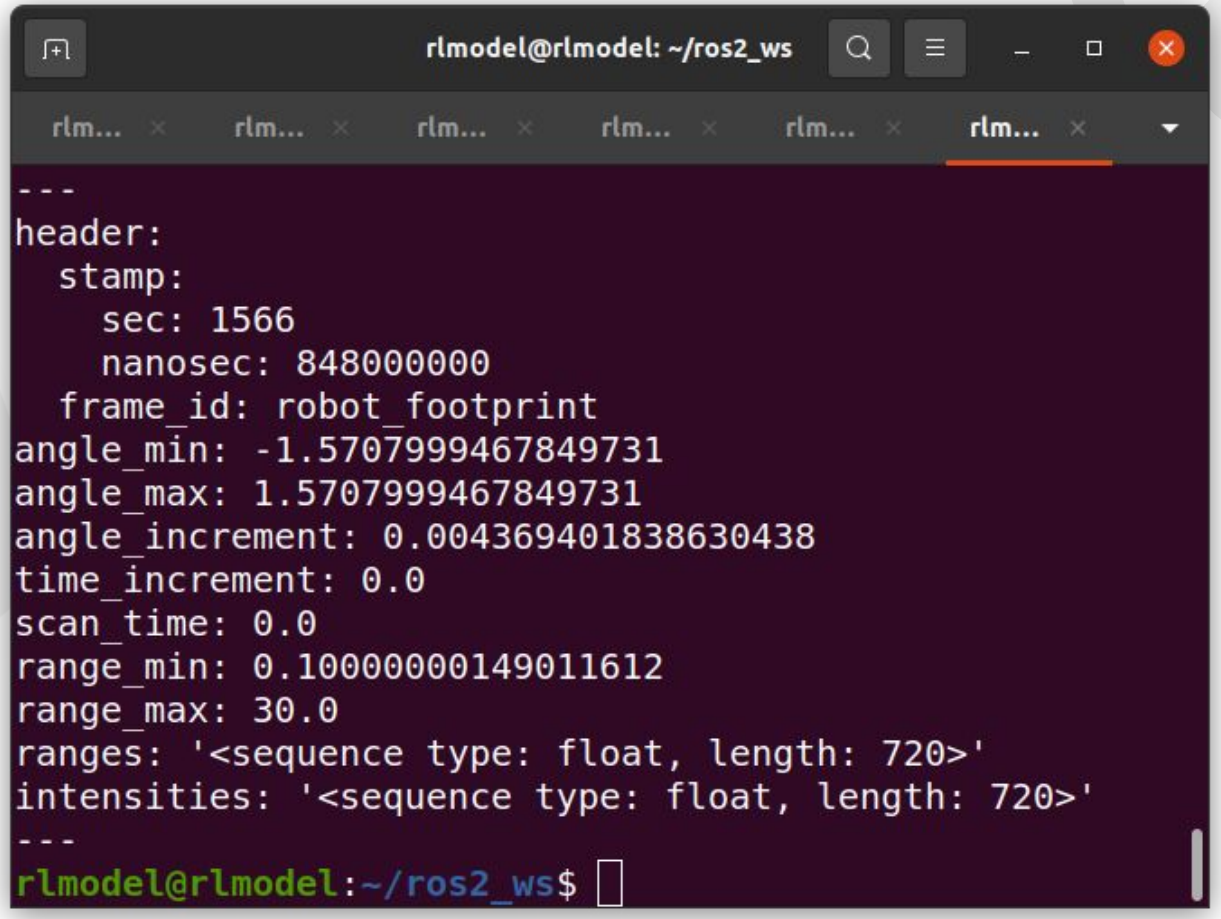
Raw Message Definition

```
# Single scan from a planar laser range-finder
#
# If you have another ranging device with different behavior (e.g. a sonar
# array), please find or create a different message, since applications
# will make fairly laser-specific assumptions about this data

Header header            # timestamp in the header is the acquisition time of
                         # the first ray in the scan.
                         #
                         # in frame frame_id, angles are measured around
                         # the positive Z axis (counterclockwise, if Z is up)
                         # with zero angle being forward along the x axis

float32 angle_min        # start angle of the scan [rad]
float32 angle_max        # end angle of the scan [rad]
float32 angle_increment  # angular distance between measurements [rad]

float32 time_increment   # time between measurements [seconds] - if your scanner
                         # is moving, this will be used in interpolating position
                         # of 3d points
float32 scan_time        # time between scans [seconds]

float32 range_min        # minimum range value [m]
float32 range_max        # maximum range value [m]

float32[] ranges         # range data [m] (Note: values < range_min or > range_max should be discarded)
float32[] intensities    # intensity data [device-specific units].  If your
                         # device does not provide intensities, please leave
                         # the array empty.
```

ROS2 HW

/diffbot/scan

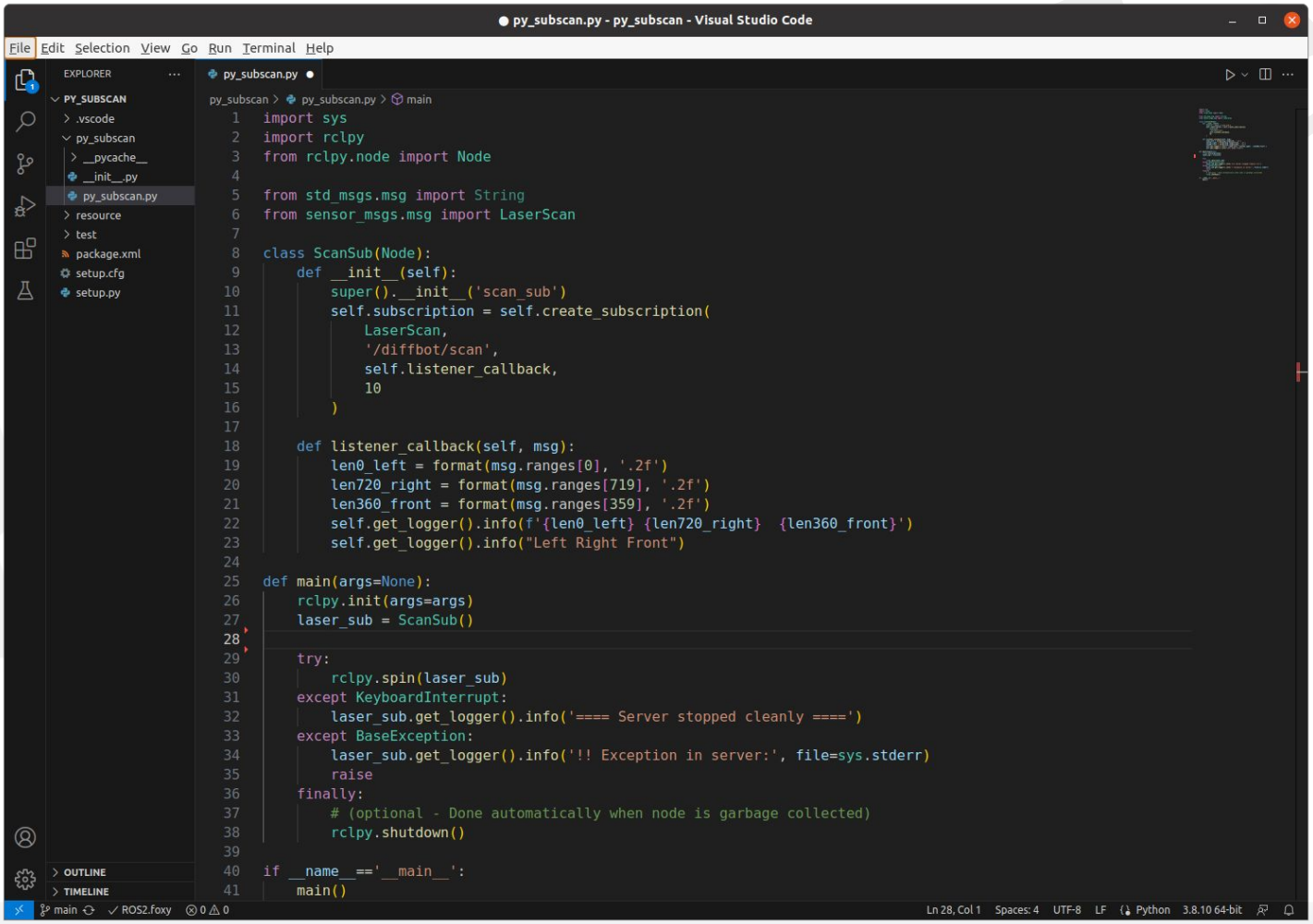
$ ros2 topic echo /scan --no-arr



ROS2 HW

/diffbot/scan

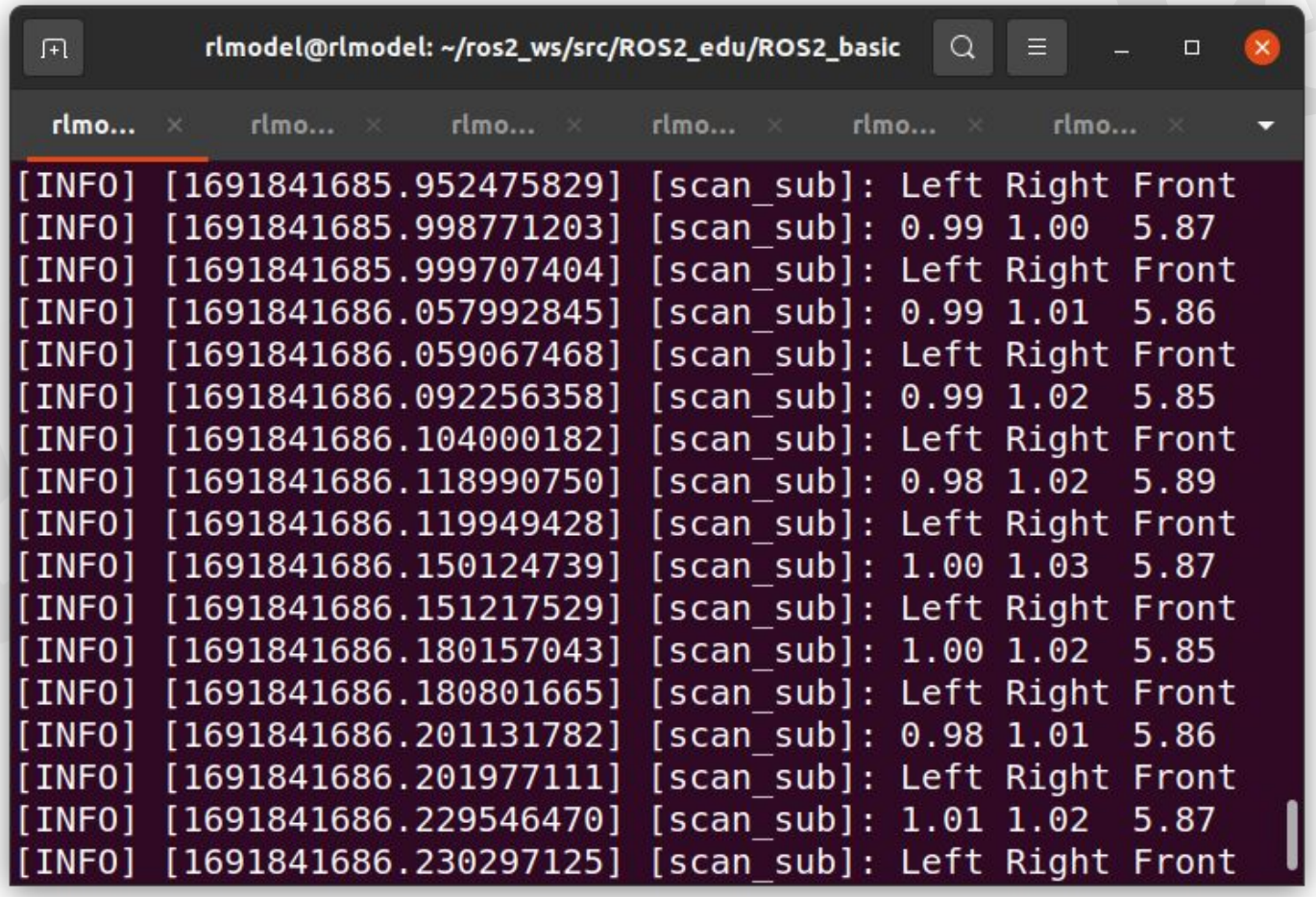
create pkg



ROS2 HW

/diffbot/scan

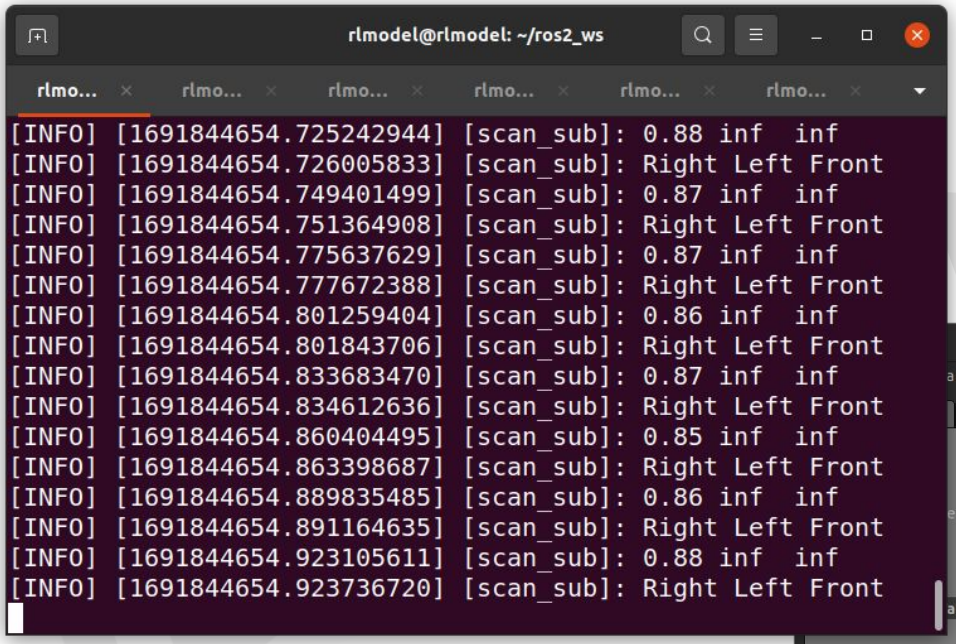
create pkg / compile build / run

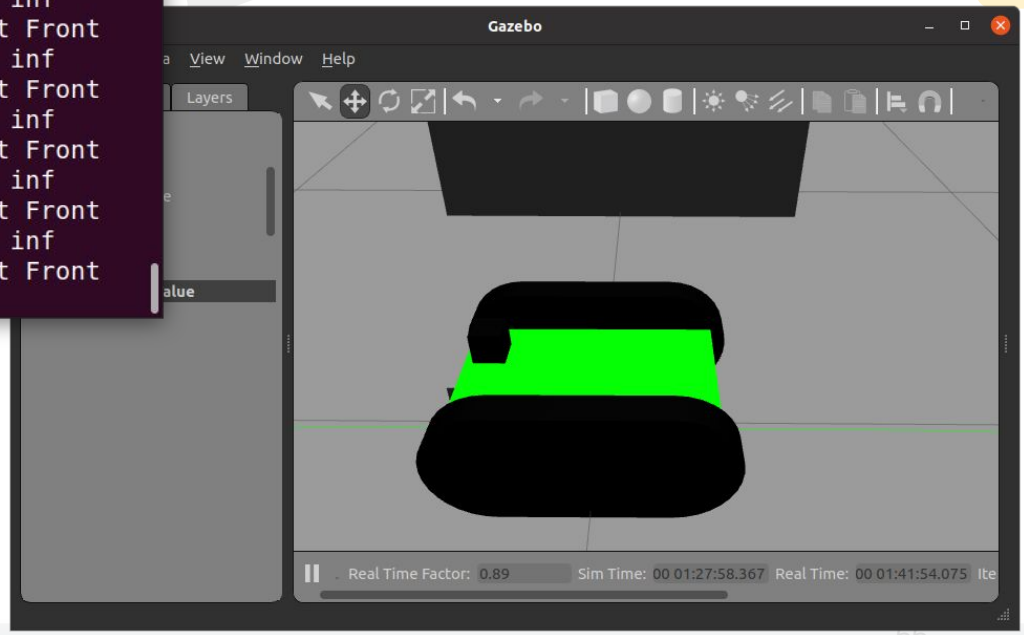


ROS2 HW

/diffbot/scan

create pkg -> compile build -> run -> check scan data



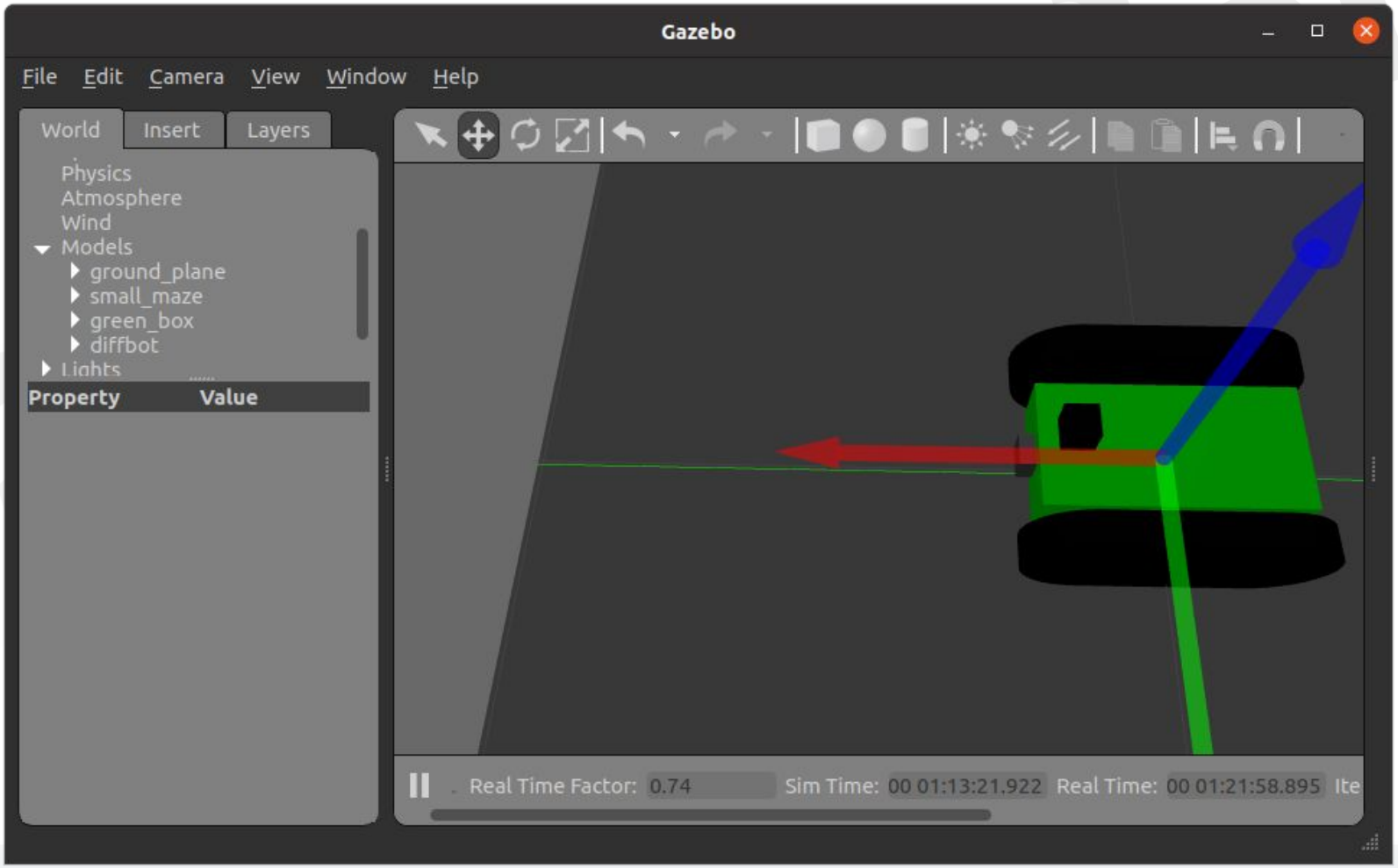


ROS2 HW

/diffbot/scan

challenge: parking.

Home work

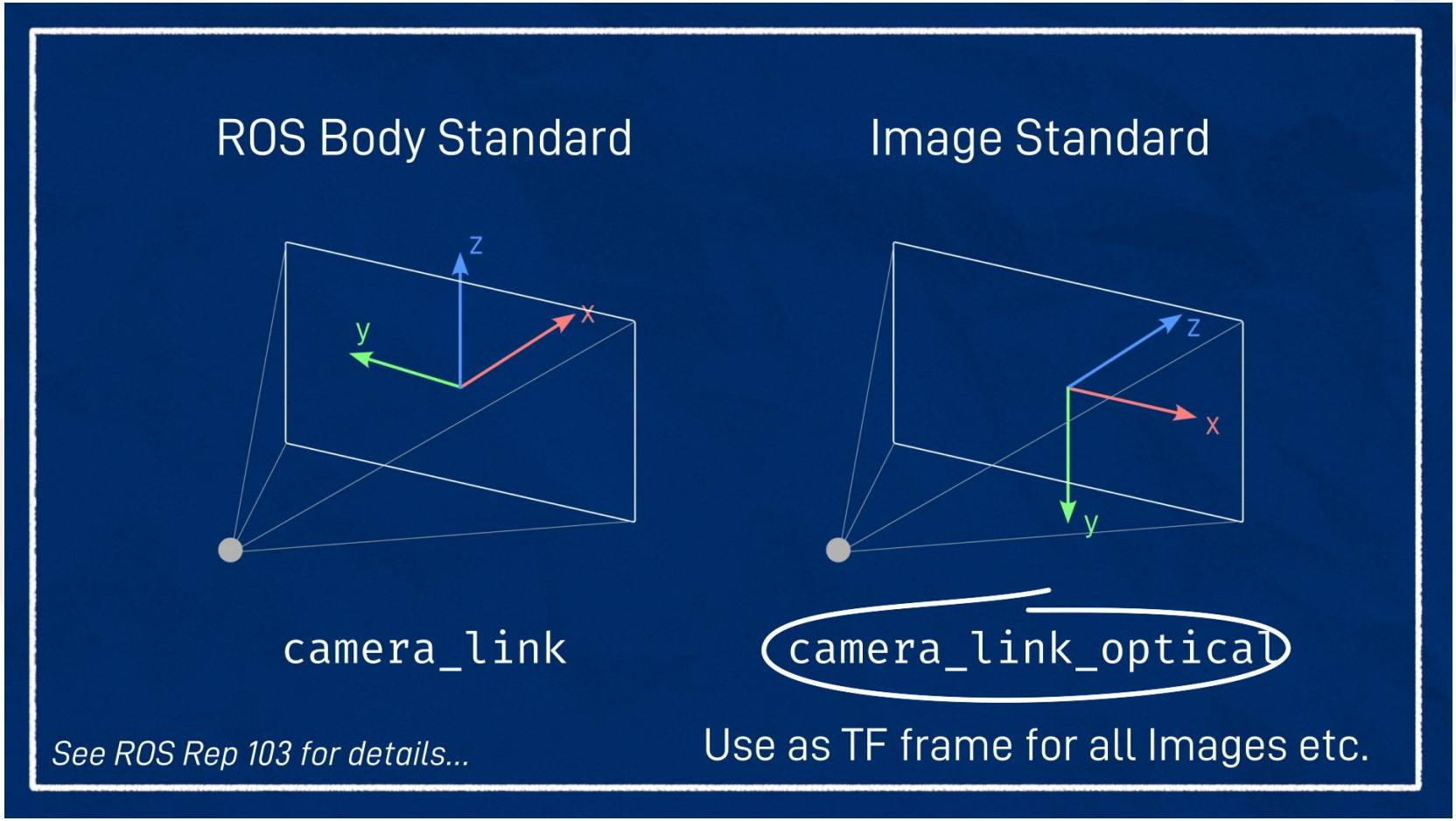


Camera & Vision

ROS2 HW

CAMERA

demo - webcam mage

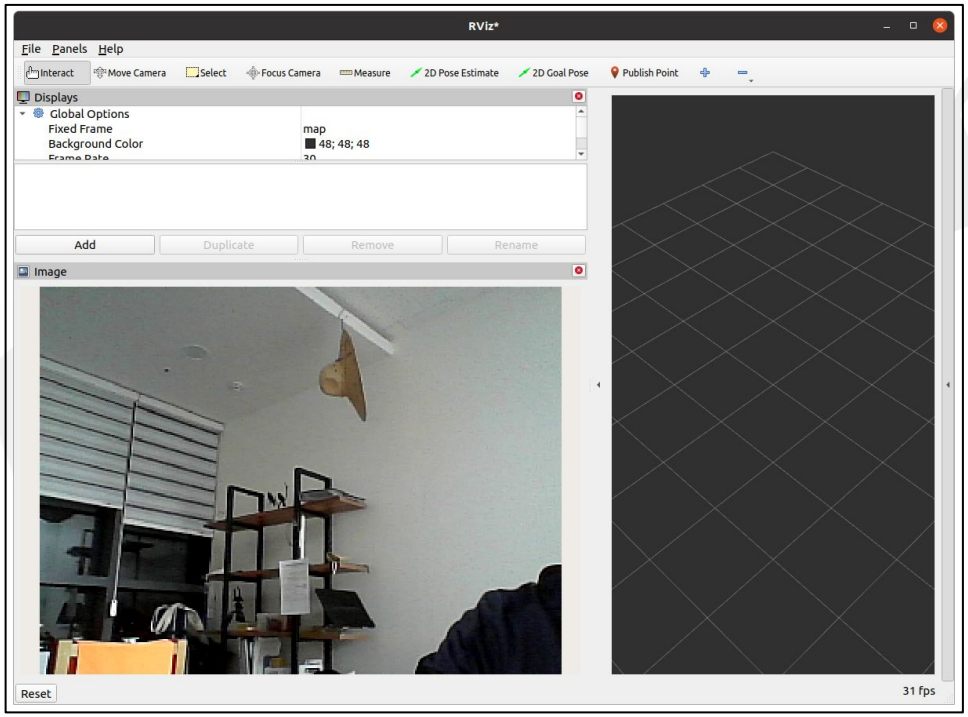


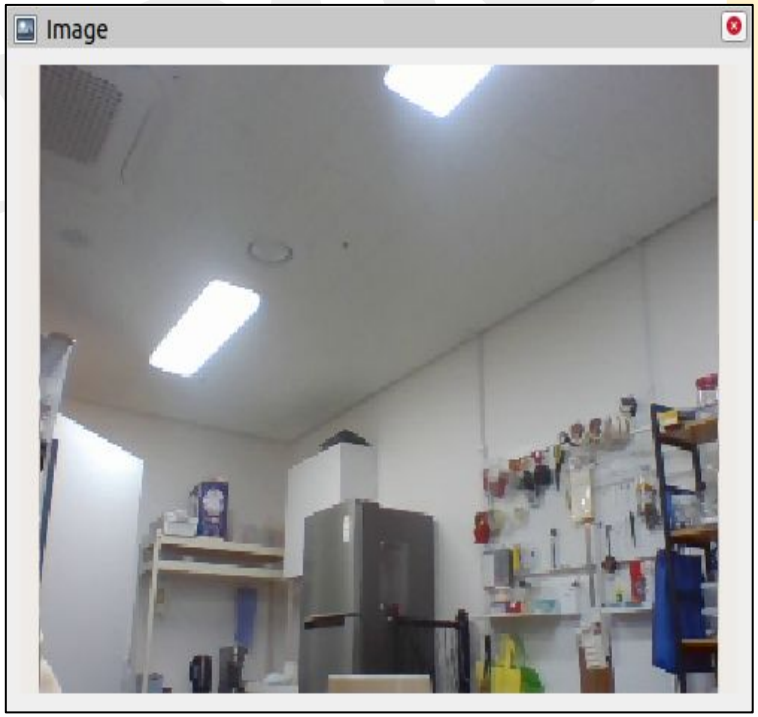
ROS2 HW

CAMERA

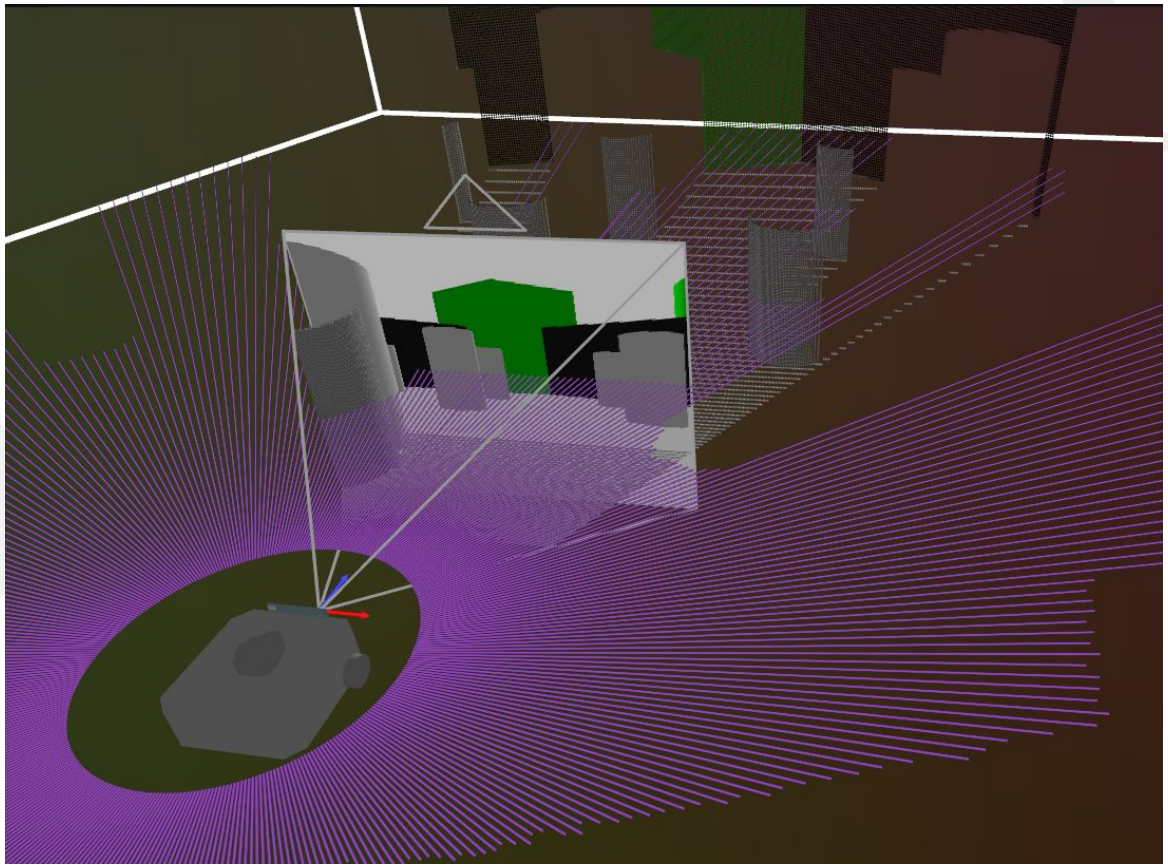
명령어: $ros2 run image_tools cam2image

확인: rviz2 -> add -> By topic -> /image/Image





cv_bridge

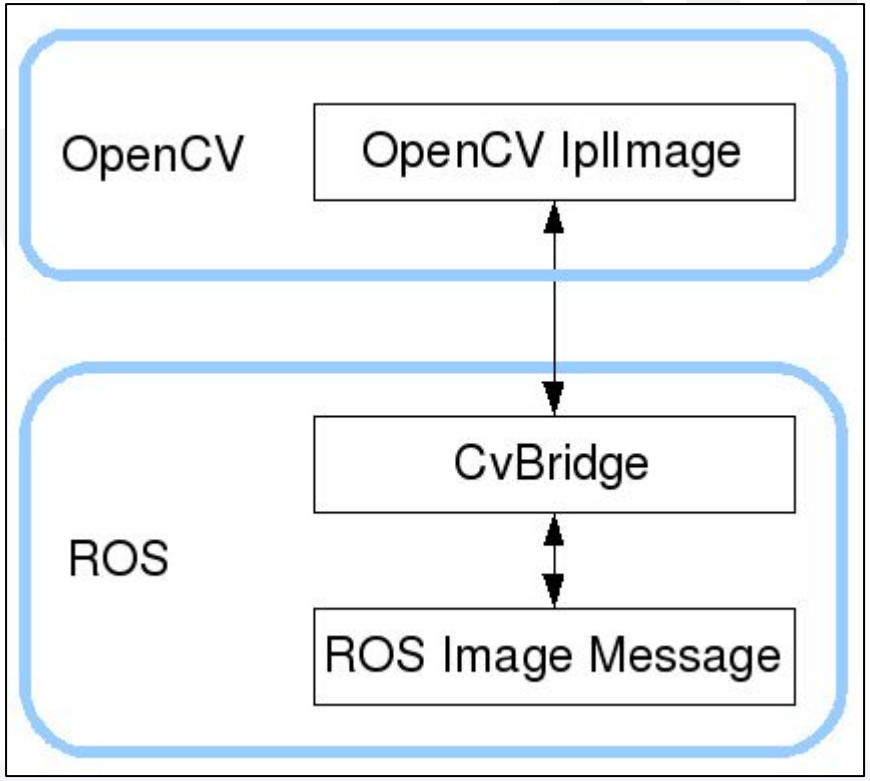


cv_bridge

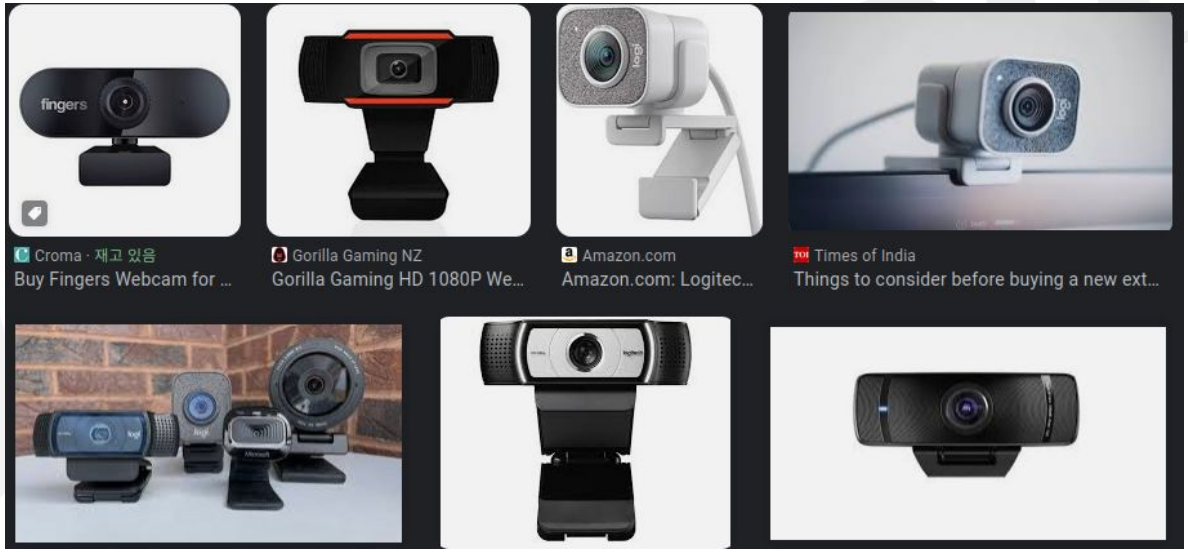
기본설명

cv_bridge converts between ROS 2 image messages and OpenCV image representation for perception applications. As follows:

https://index.ros.org/p/cv_bridge/



Camera test


fingers
Croma · 재고 있음
Buy Fingers Webcam for ...
Gorilla Gaming NZ
Gorilla Gaming HD 1080P We...
Amazon.com
Amazon.com: Logitec...
Times of India
Things to consider before buying a new ext...

image_tools

image_tools package

HW확인: $ lsusb

ref: genicam (Appendix)

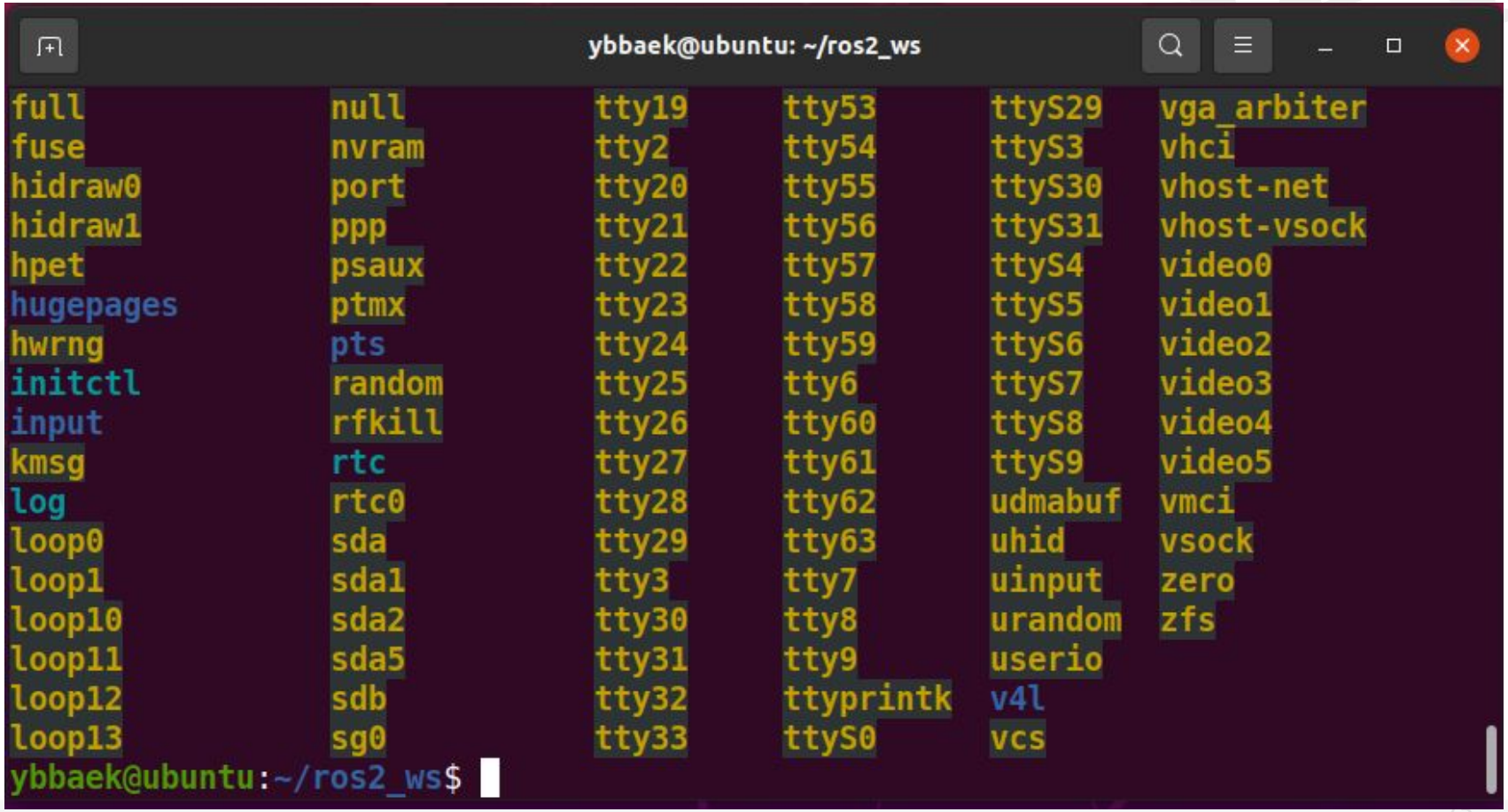
```
ybbaek@ubuntu: ~/ros2_ws
ybbaek@ubuntu:~/ros2_ws$ lsusb
Bus 001 Device 002: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 013: ID 0bda:58fe Realtek Semiconductor Corp. Integrated_Webcam
_HD
Bus 001 Device 011: ID 2109:8817 VIA Labs, Inc. USB Billboard Device
Bus 001 Device 010: ID 2109:8824 VIA Labs, Inc. USB Billboard Device
Bus 001 Device 009: ID 0bda:9210 Realtek Semiconductor Corp. RTL9210B-CG
Bus 001 Device 015: ID 1902:8301 Generic HD camera
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 008: ID 08bb:2707 Texas Instruments PCM2707 stereo audio DAC
Bus 002 Device 007: ID 0e0f:0008 VMware, Inc. VMware Virtual USB Mouse
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
ybbaek@ubuntu:~/ros2_ws$
```

image_tools

image_tools package

영상장치 확인: $ ls /dev

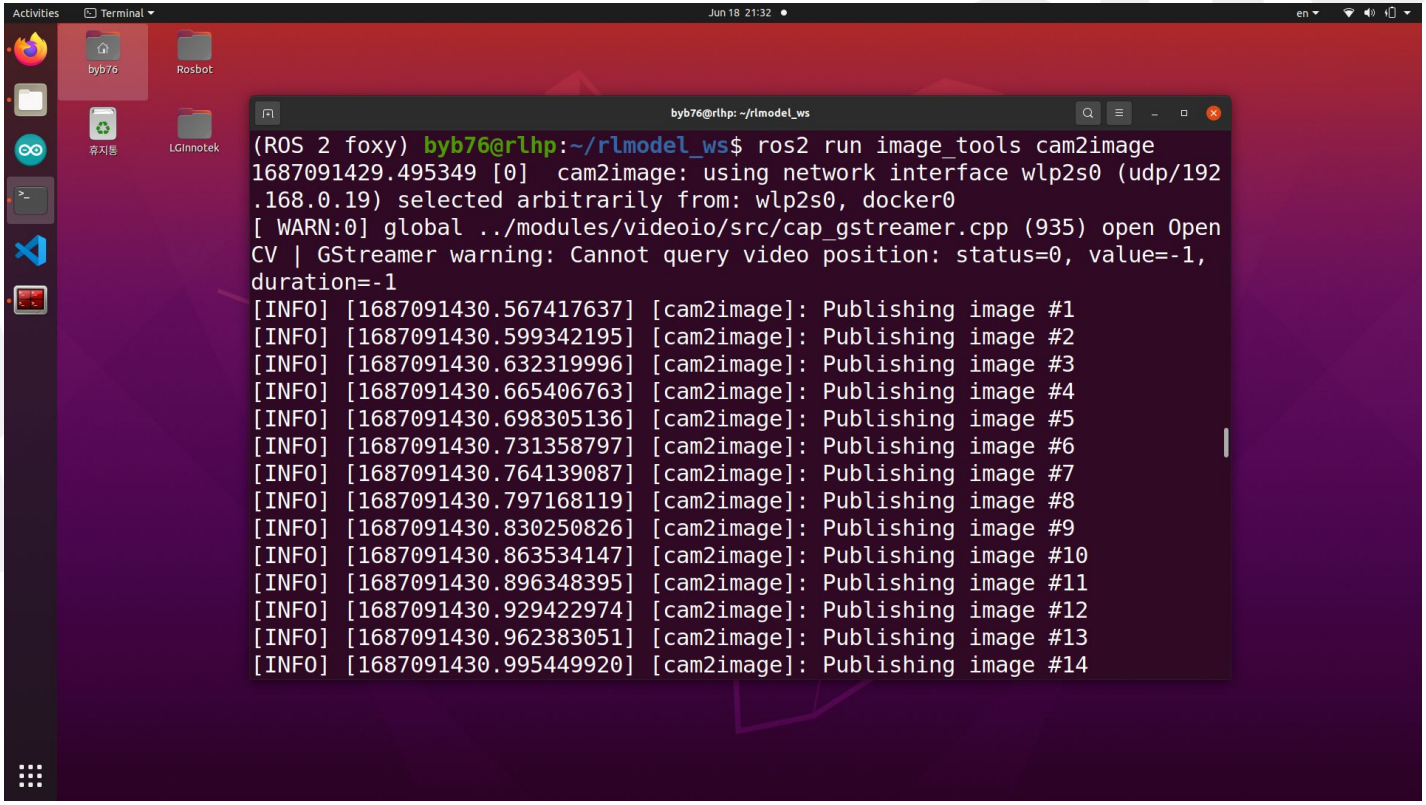
>video0, video1 항목 확인



image_tools

image_tools package

```
$ ros2 run image_tools cam2image
$ ros2 run image_tools cam2image video0:=video2
$ ros2 run image_tools cam2image --ros-args --remap video0:=video1
```



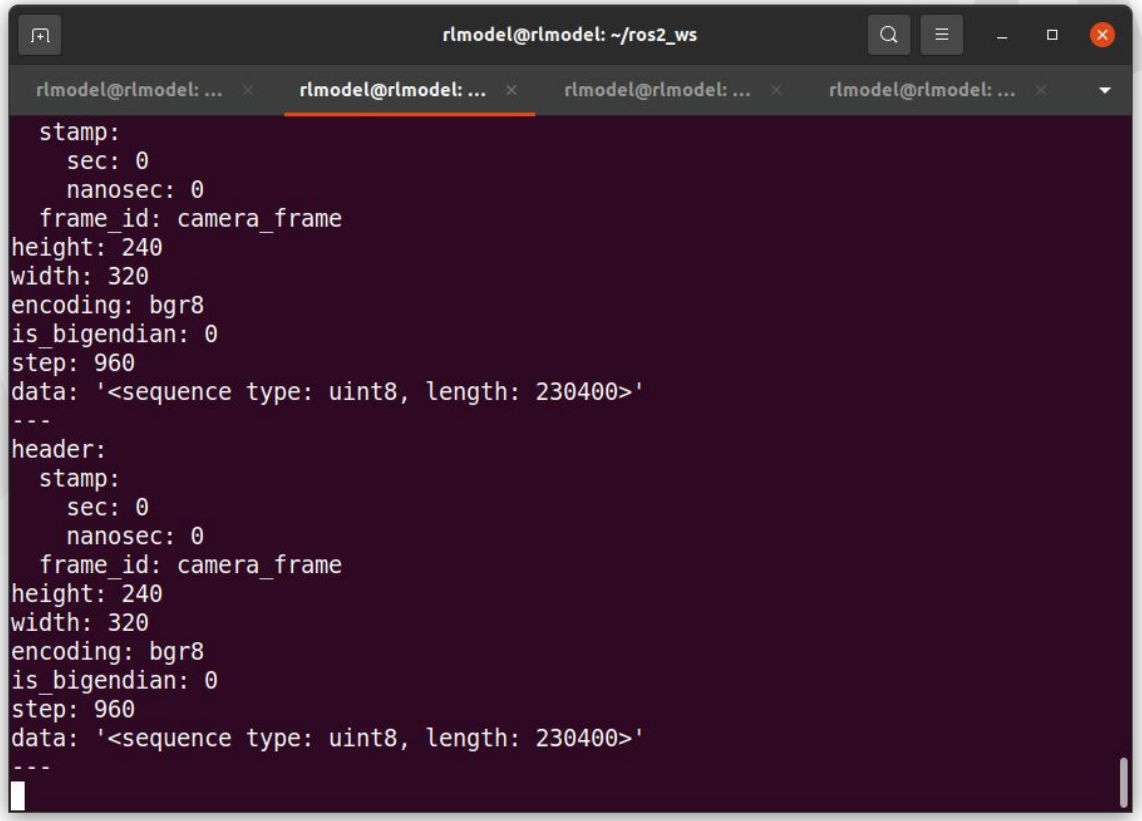
image_tools

image_tools package

image topic check:

$ ros2 run image_tools cam2image

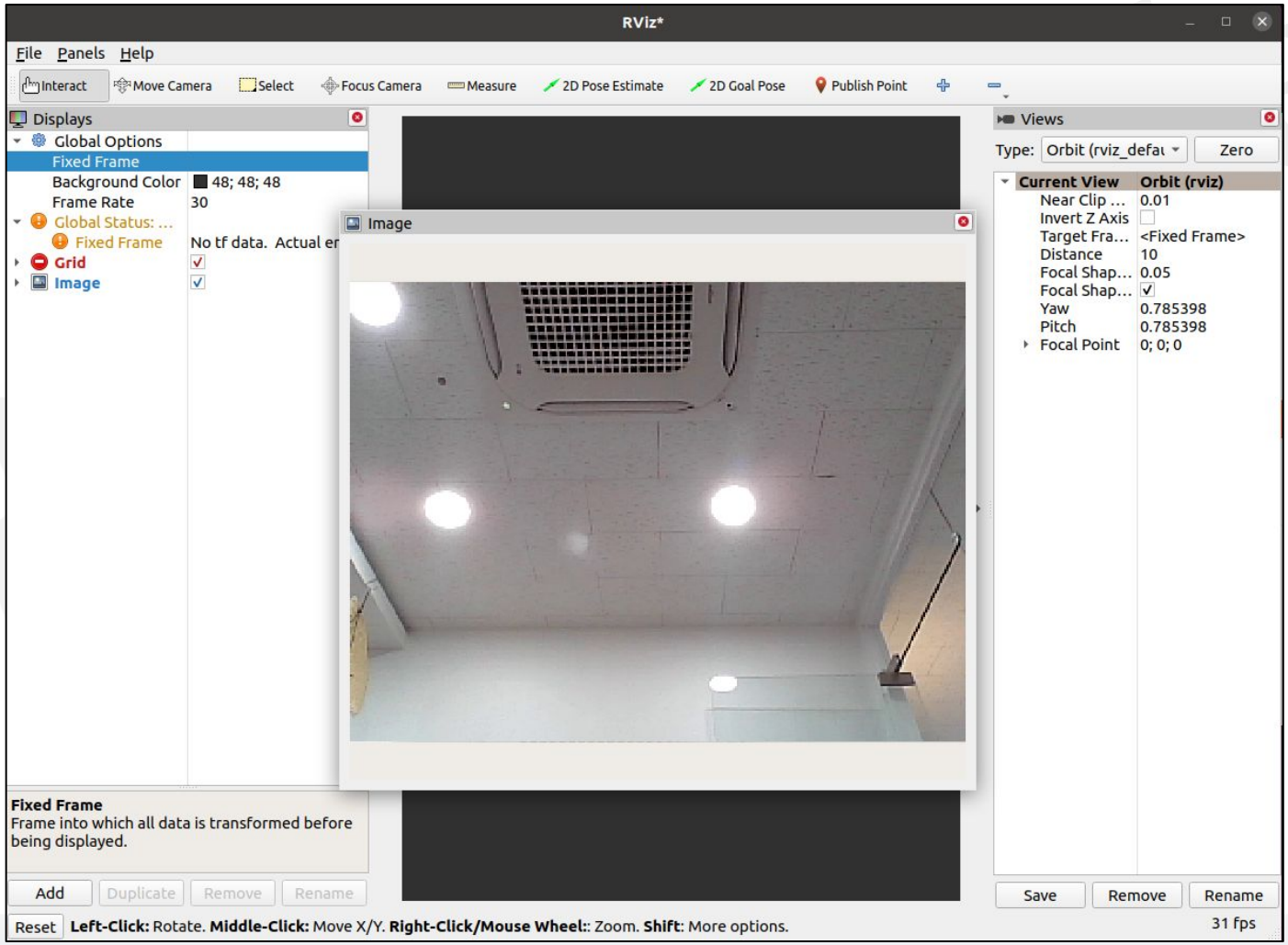
$ ros2 topic echo --no-arr /image_raw



image_tools

image_tools package

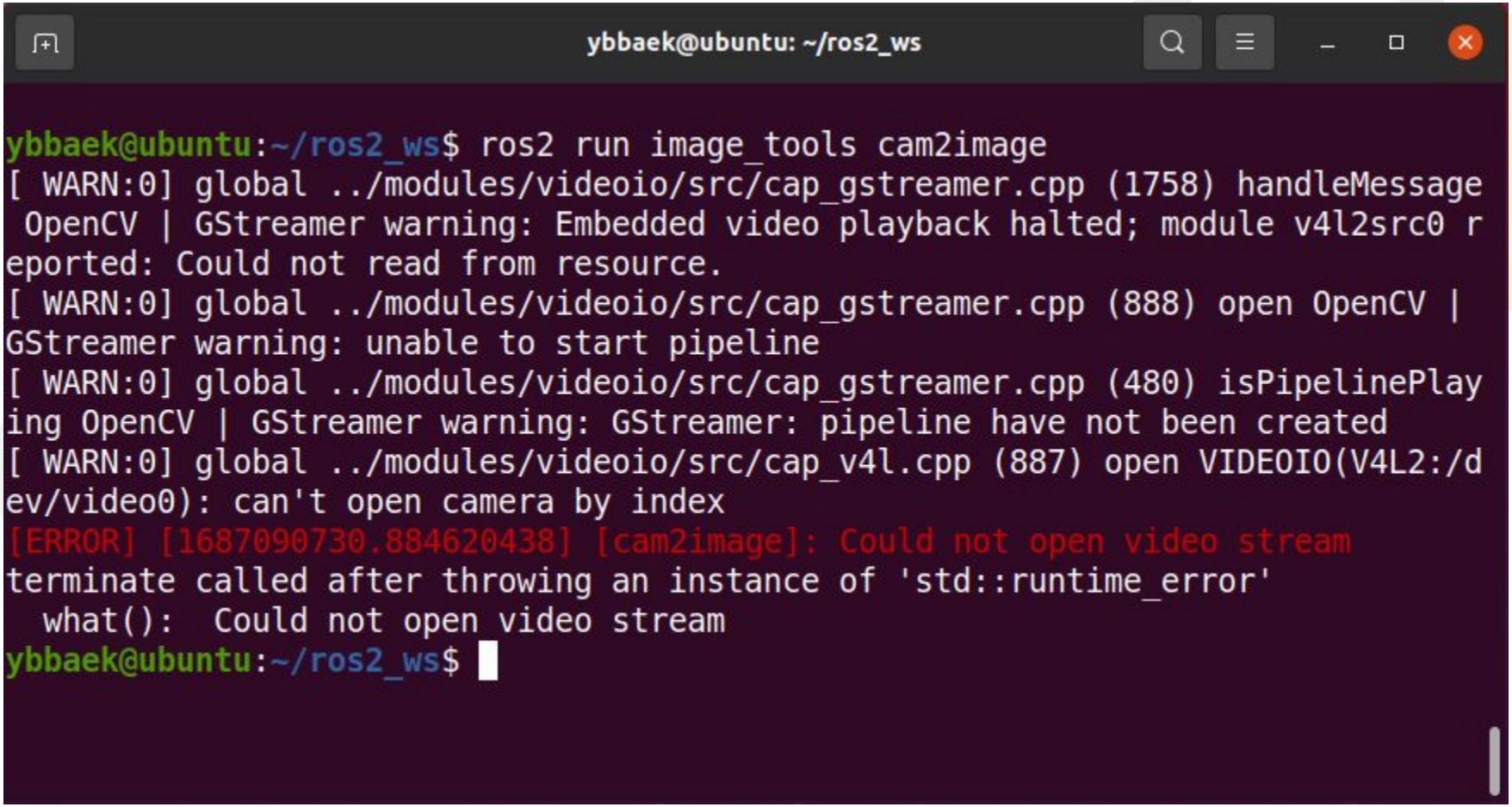
확인: rviz2



image_tools

image_tools package

에러 케이스

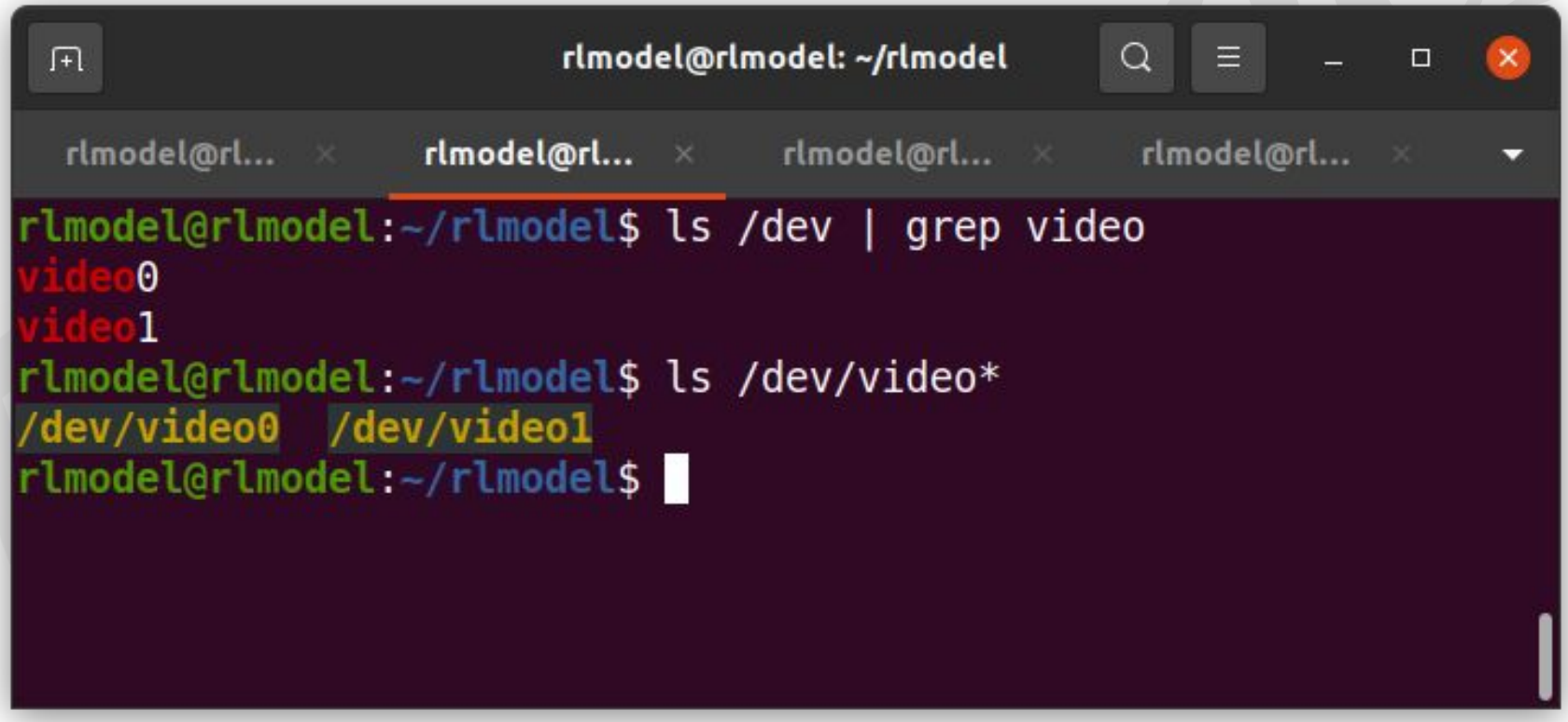


image_tools

image_tools package

$ ls /dev | grep video

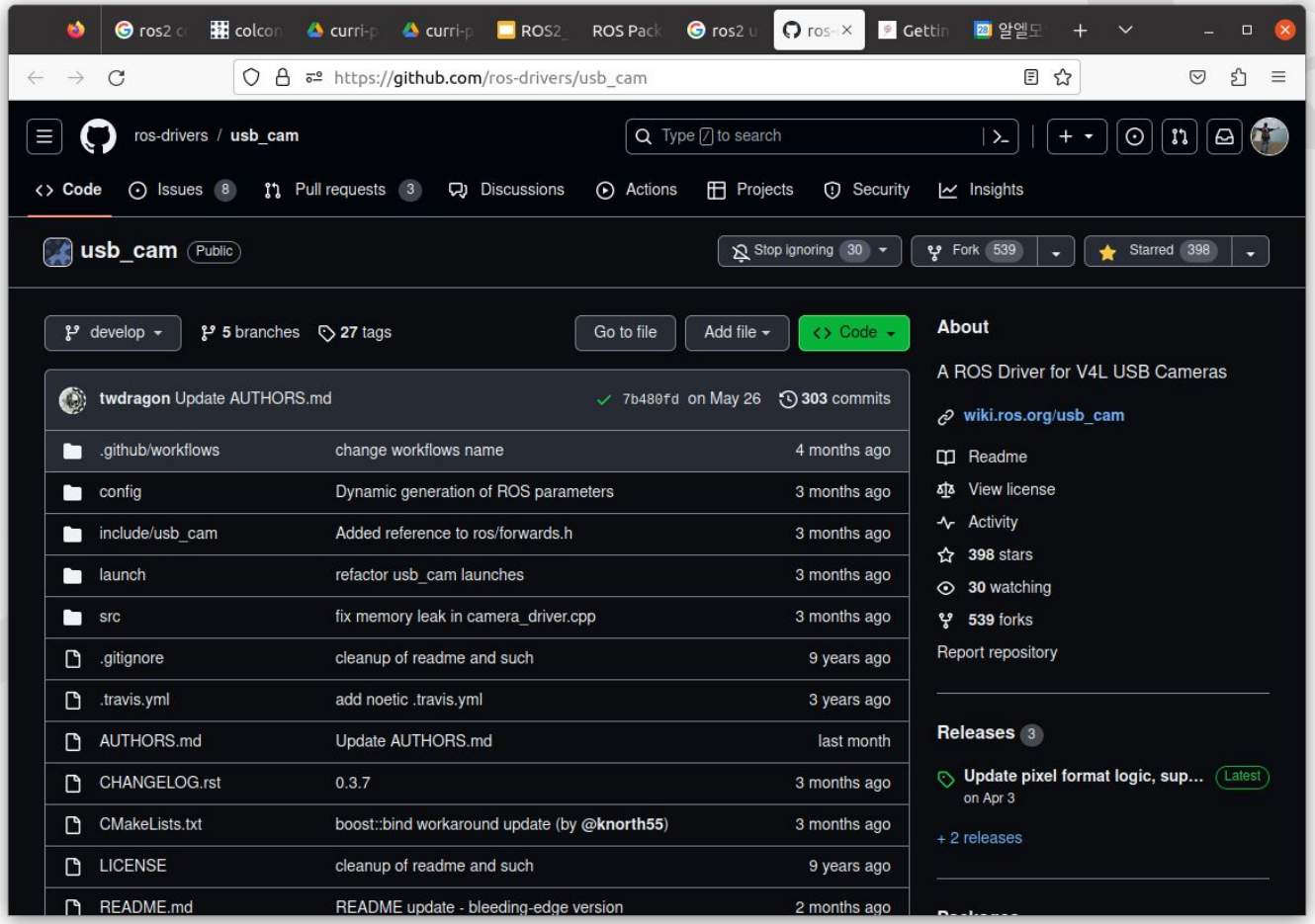
$ ls /dev/video*



usb_cam package

usb_cam (ros2 driver)

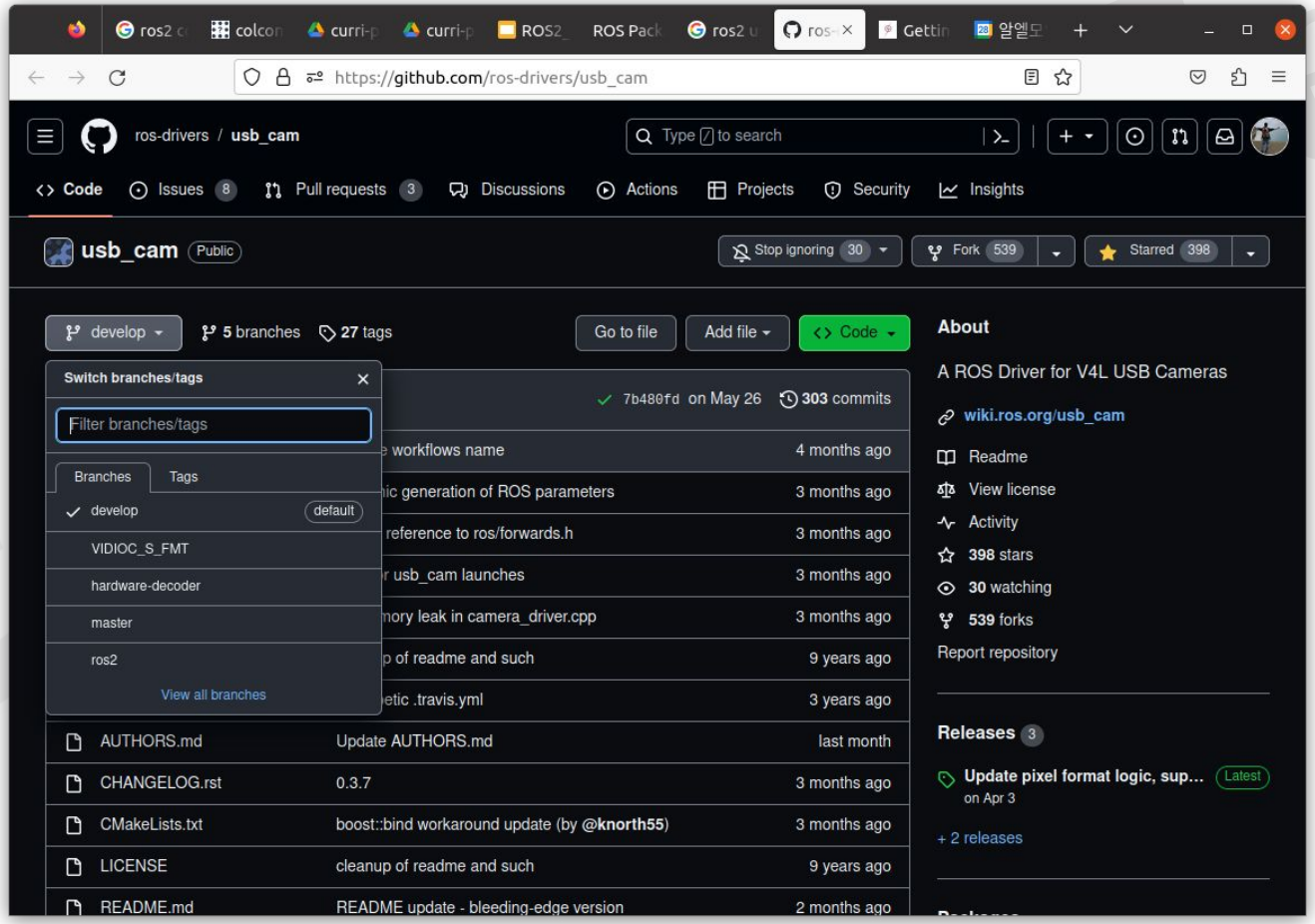
Link: https://github.com/ros-drivers/usb_cam



usb_cam package

usb_cam (ros2 driver)

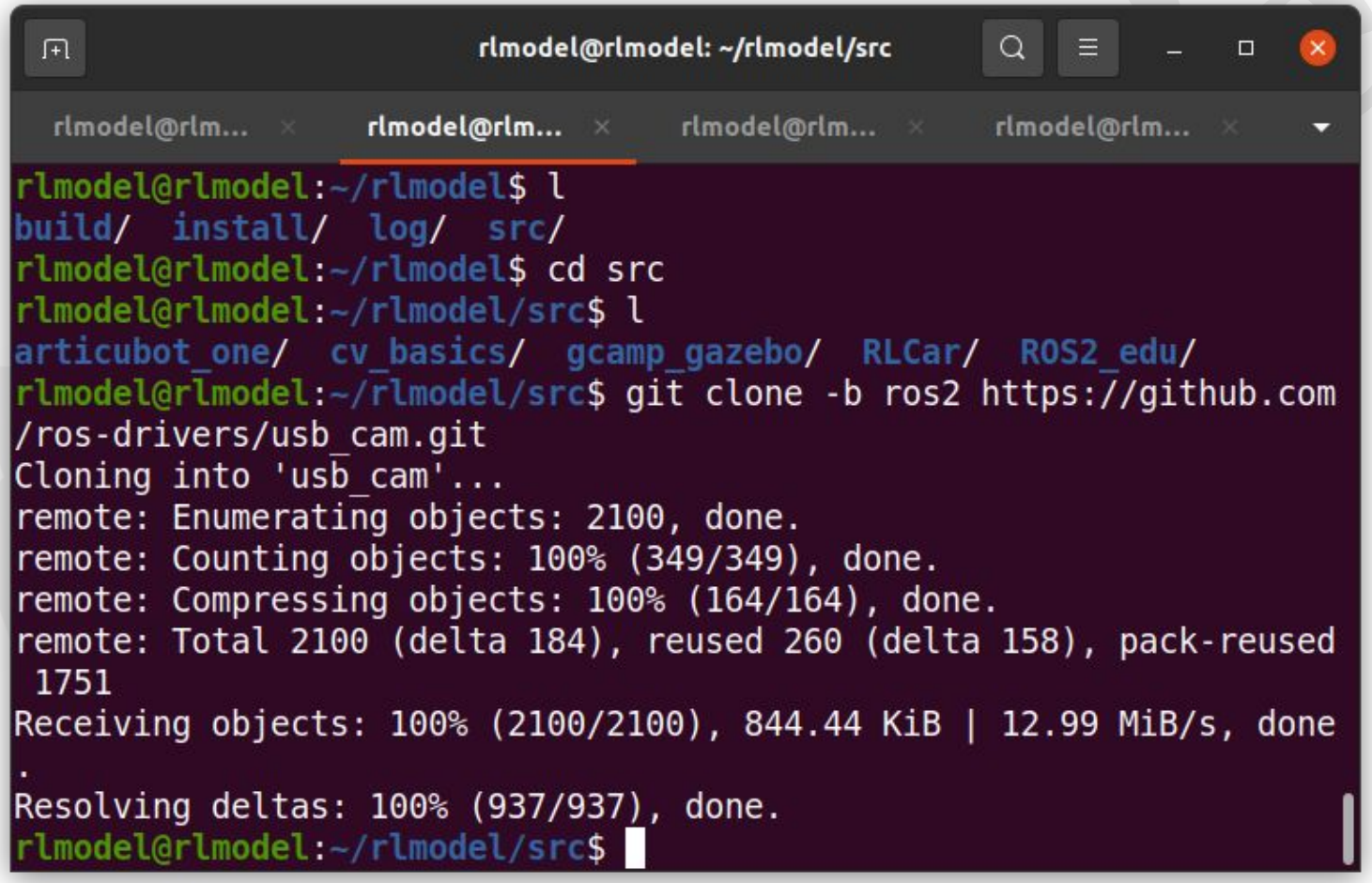
Link: branch ros2



usb_cam package

usb_cam (ros2 driver)

$ git clone -b ros2 https://github.com/ros-drivers/usb_cam.git

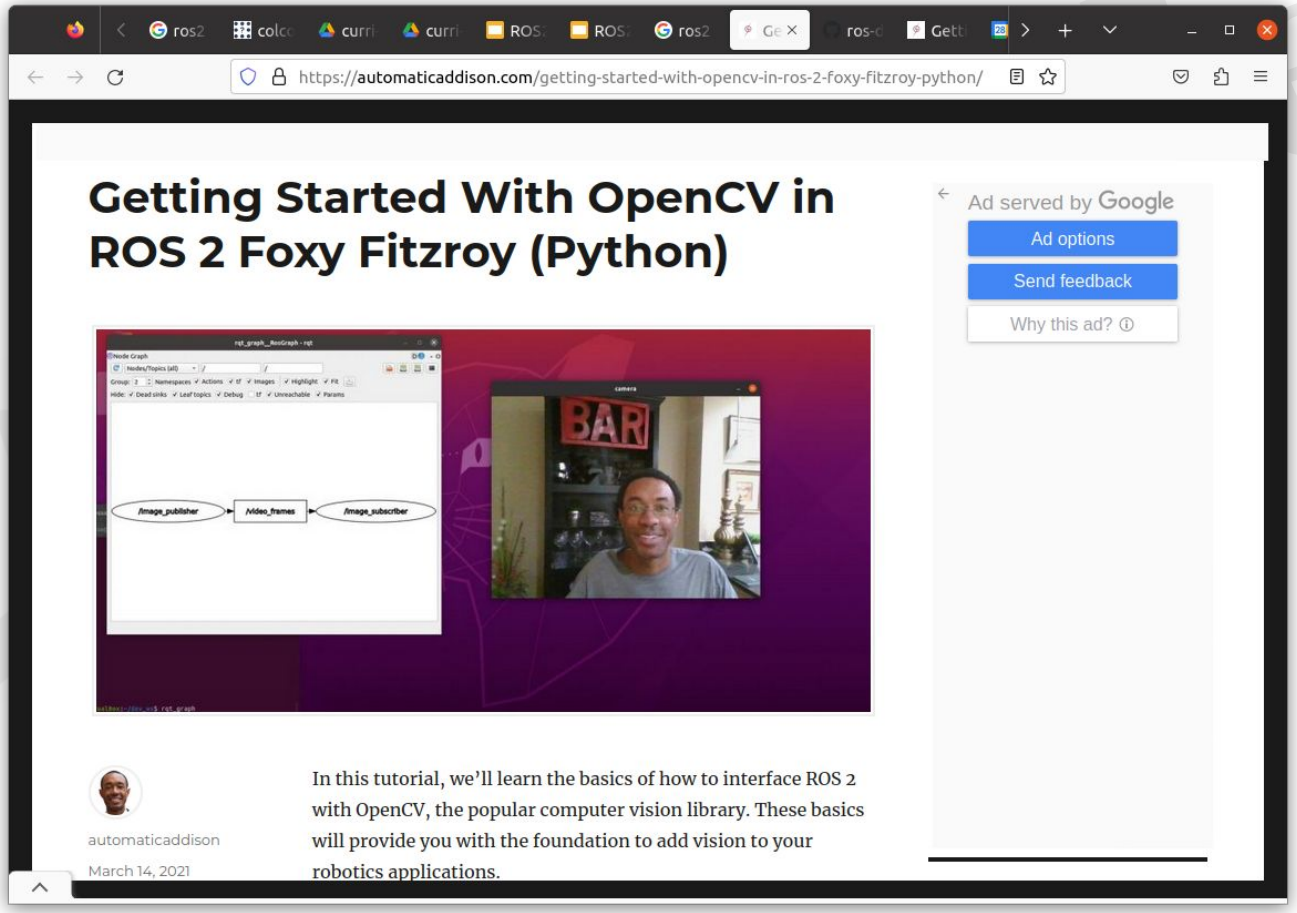


cv_bridge Package example

cv package

ref

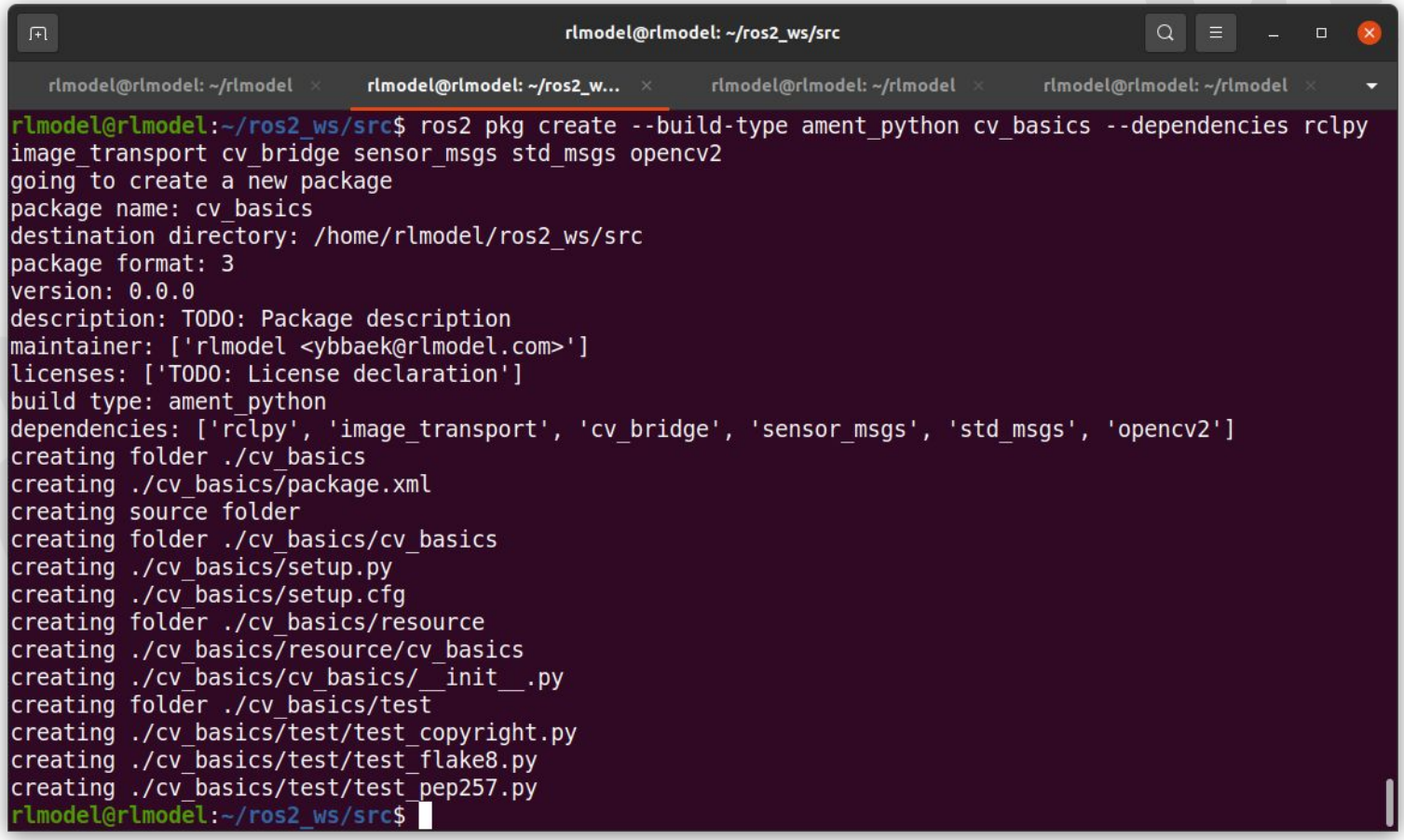
https://automaticaddison.com/getting-started-with-opencv-in-ros-2-foxy-fitzroy-python/



cv package

Package creation

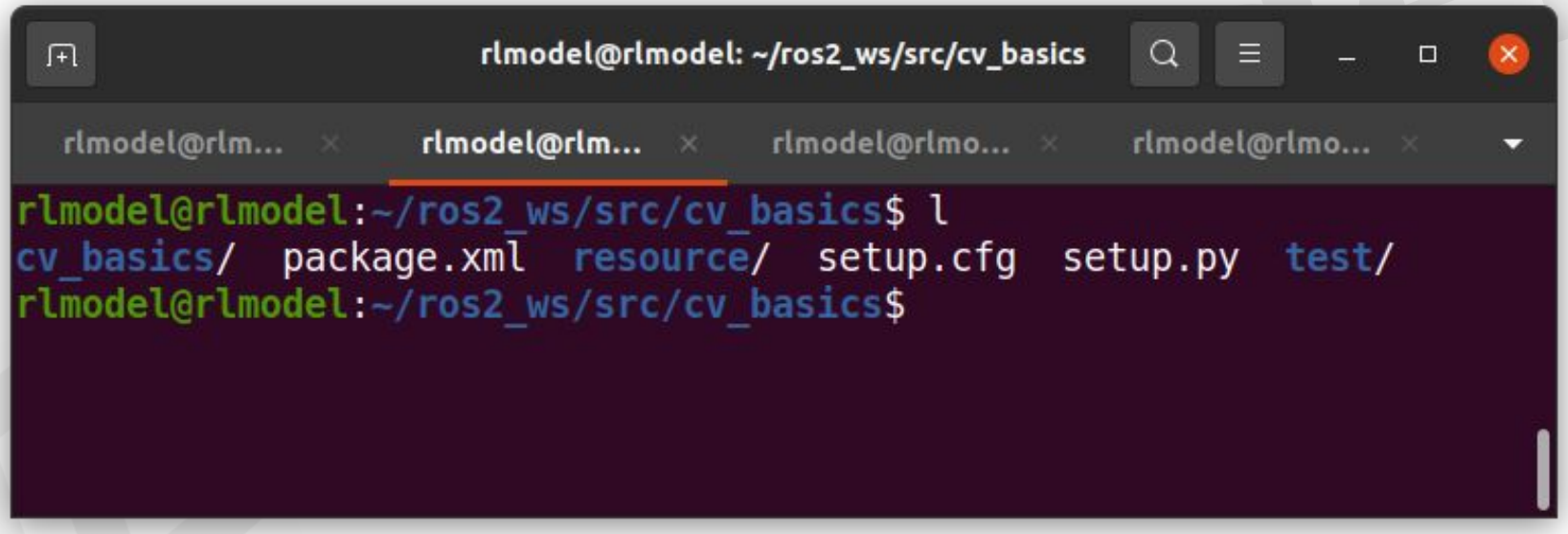
ros2 pkg create --build-type ament_python cv_basics --dependencies rclpy image_transport cv_bridge sensor_msgs std_msgs opencv2



cv package

Package check

ros2_ws/src/cv_basics

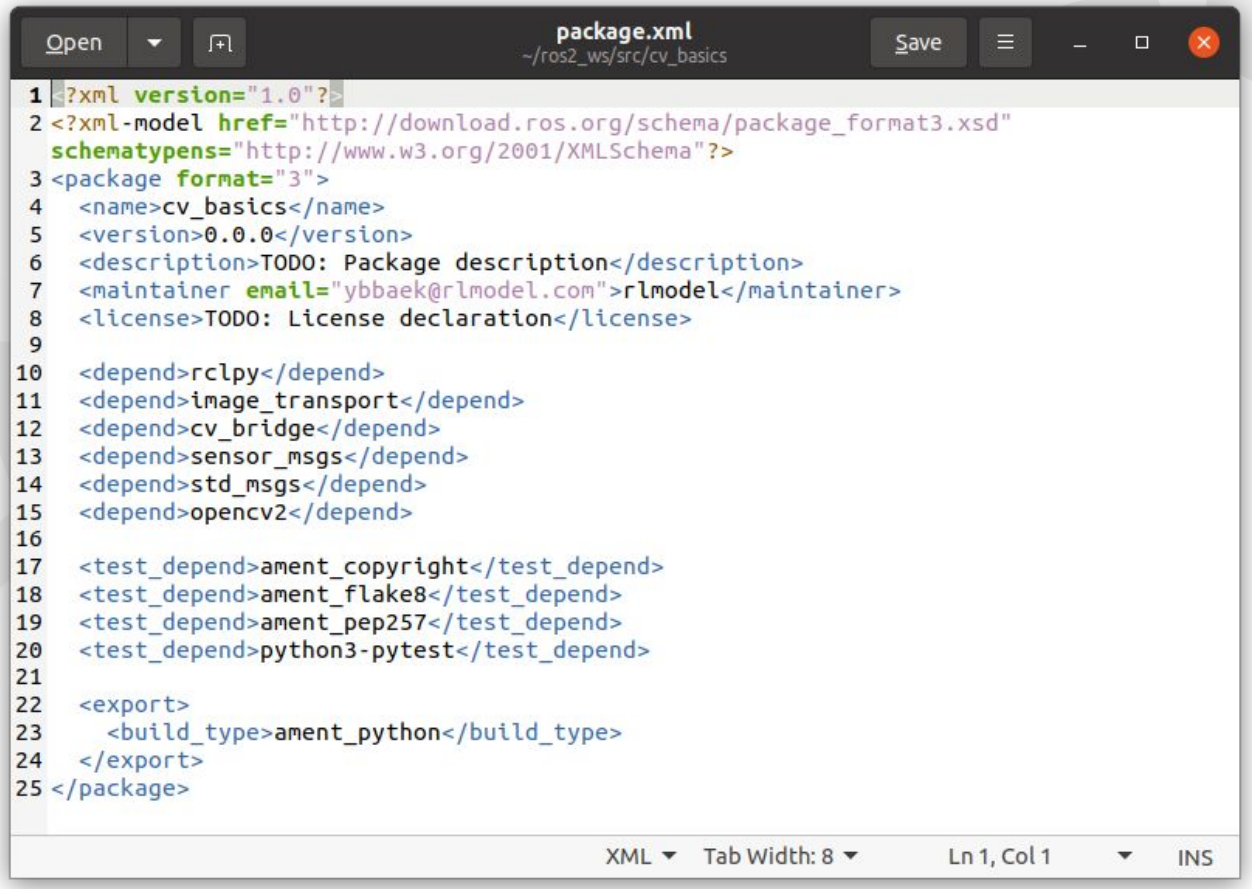


cv package

Package check

location: ros2_ws/src/cv_basics

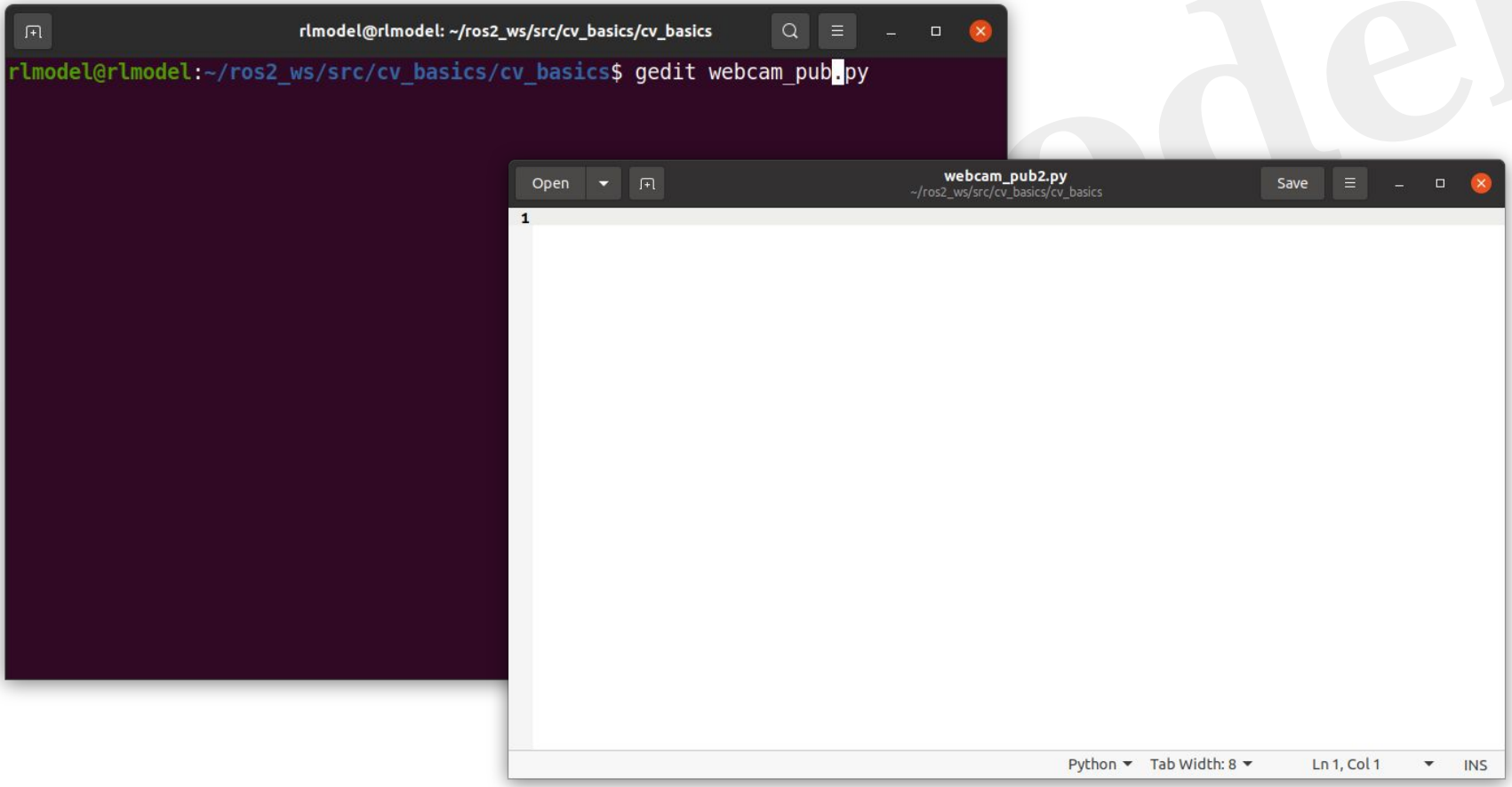
check: package.xml



cv package

New python file

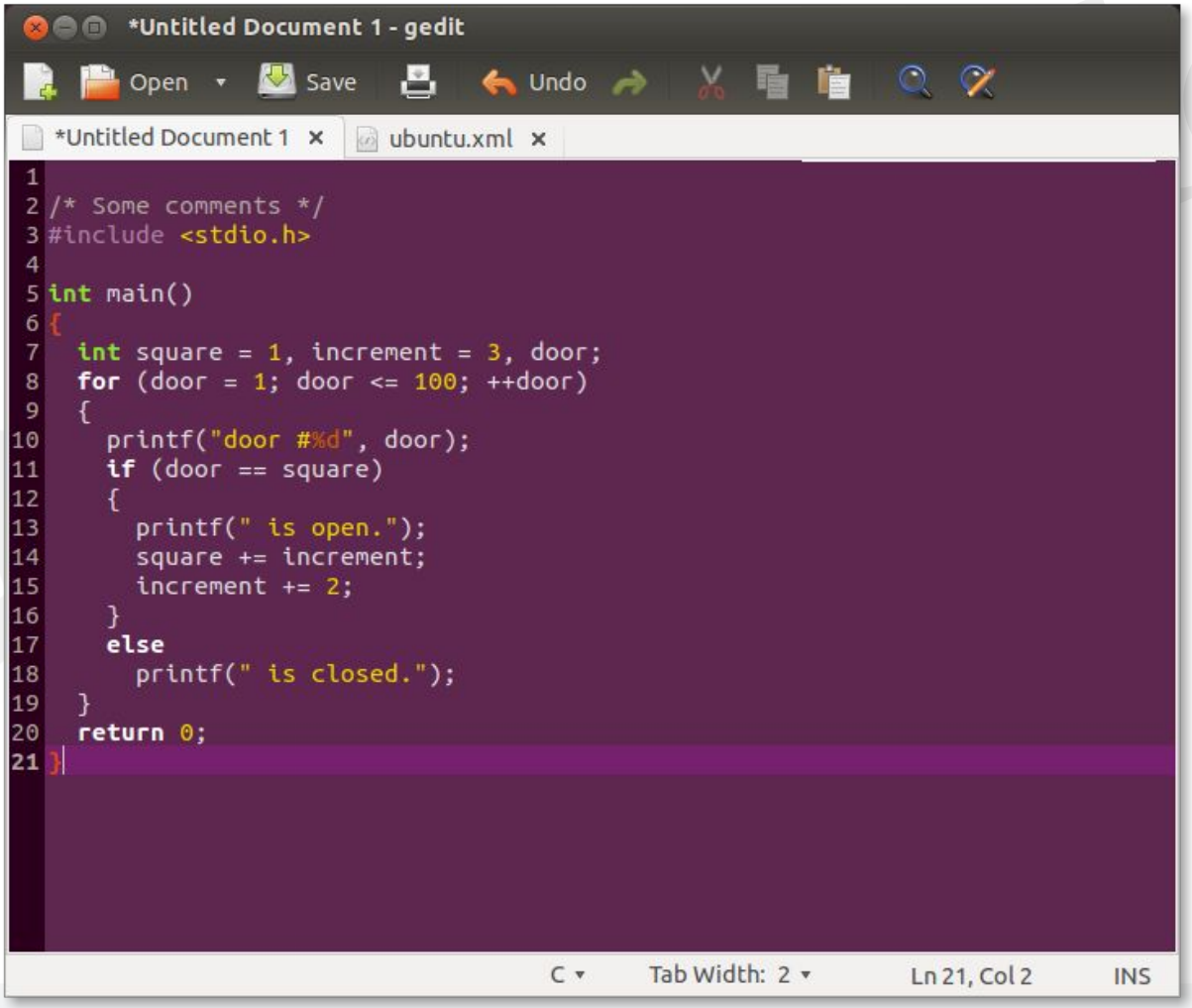
new file: gedit webcam_pub.py



cv package

geditor

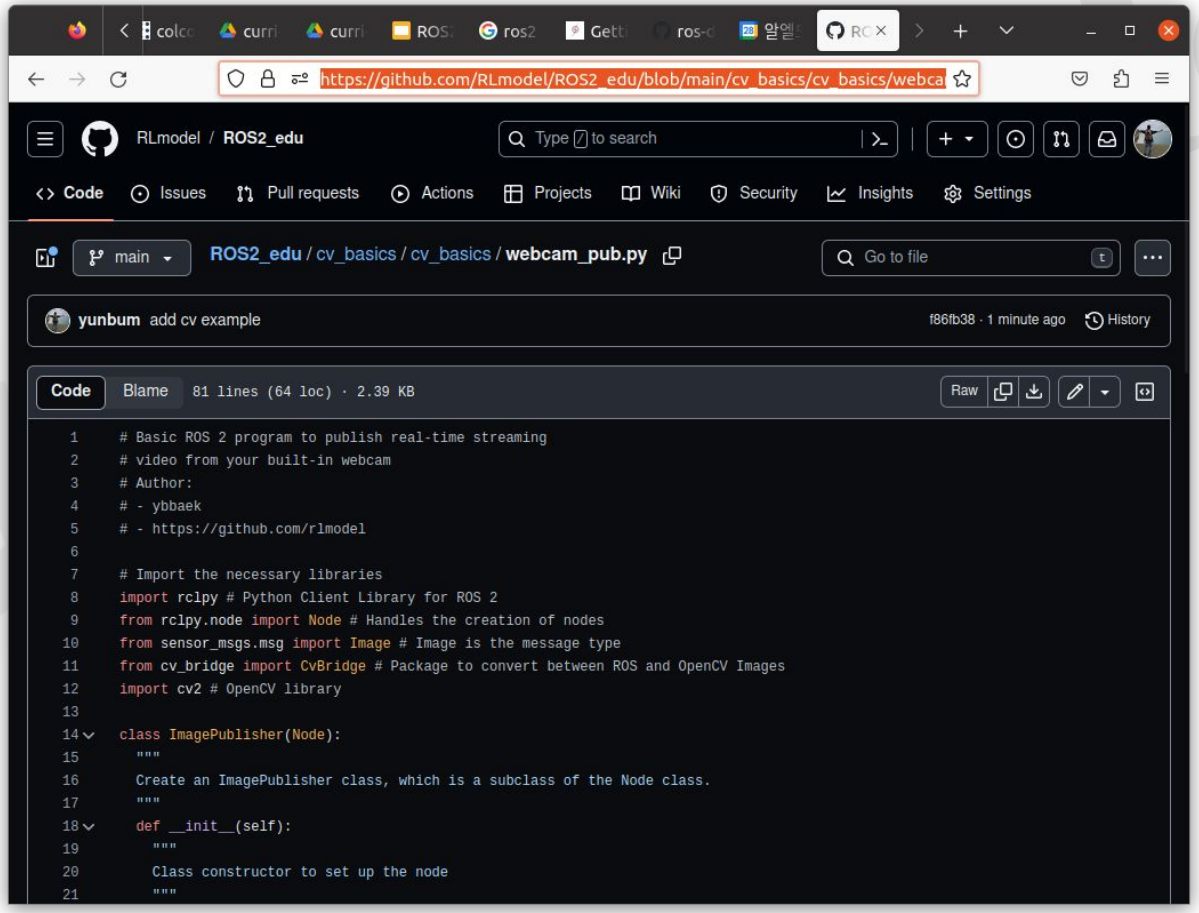
install: sudo apt-get install gedit (if error)



cv package

geditor

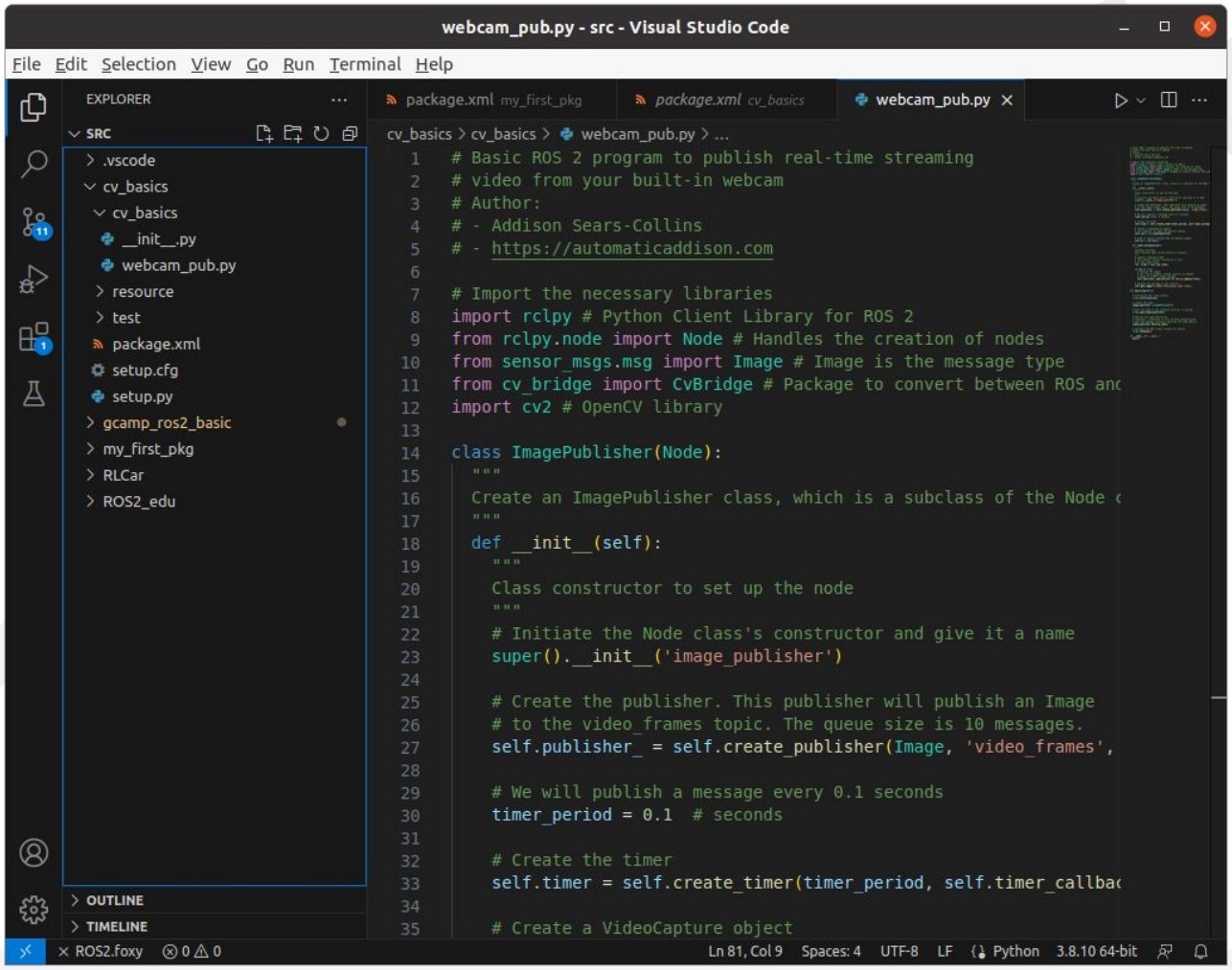
source:
https://github.com/RLmodel/ROS2_edu/blob/main/cv_basics/cv_basics/webcam_pub.py



cv package

cv_bridge coding

code review: class, node, image handling



cv package

cv_bridge coding

code review: class, node, image handling

Open ▾ | webcam_pub.py ~/ros2_ws/src/cv_basics/cv_basics | Save

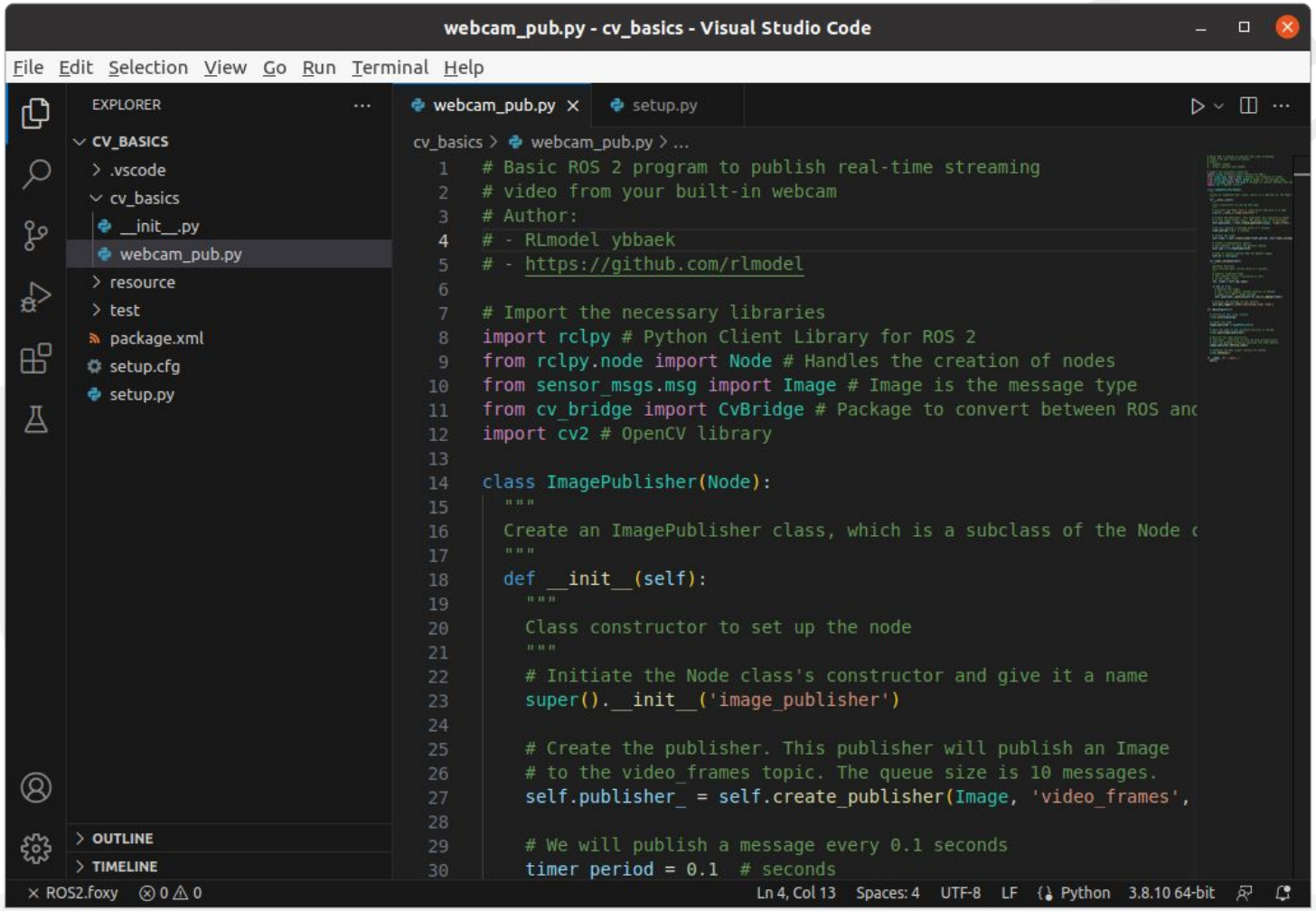
```python
# Basic ROS 2 program to publish real-time streaming
# video from your built-in webcam
# Author:
# - Addison Sears-Collins
# - https://automaticaddison.com

# Import the necessary libraries
import rclpy # Python Client Library for ROS 2
from rclpy.node import Node # Handles the creation of nodes
from sensor_msgs.msg import Image # Image is the message type
from cv_bridge import CvBridge # Package to convert between ROS and OpenCV Images
import cv2 # OpenCV library

class ImagePublisher(Node):
  """
  Create an ImagePublisher class, which is a subclass of the Node class.
  """
  def __init__(self):
    """
    Class constructor to set up the node
    """
    # Initiate the Node class's constructor and give it a name
    super().__init__('image_publisher')

    # Create the publisher. This publisher will publish an Image
    # to the video_frames topic. The queue size is 10 messages.
    self.publisher_ = self.create_publisher(Image, 'video_frames', 10)

    # We will publish a message every 0.1 seconds
```

Python ▾ | Tab Width: 8 ▾ | Ln 1, Col 1 | INS

cv package

cv_bridge coding

code review: class, node, image handling

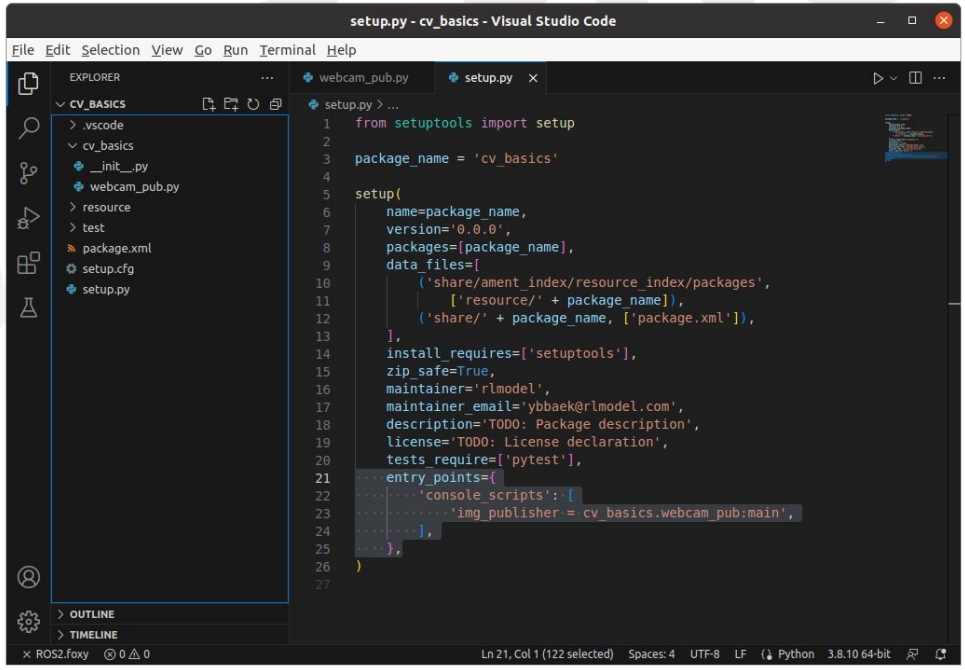


cv package

cv_bridge coding

entry_point fks: node python connection

```
entry_points={
    'console_scripts': [
        'img_publisher = cv_basics.webcam_pub:main',
    ],
},
```



cv package

cv_bridge coding

compile: colcon build --packages-select cv_basic

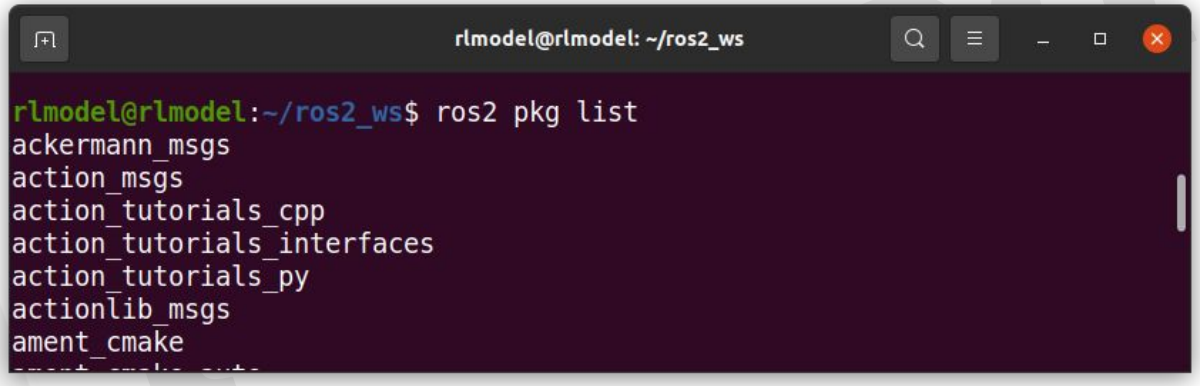
cv package

cv_bridge coding

sourcing: foxy (.bashrc alias)

>. /opt/ros/foxy/setup.bash

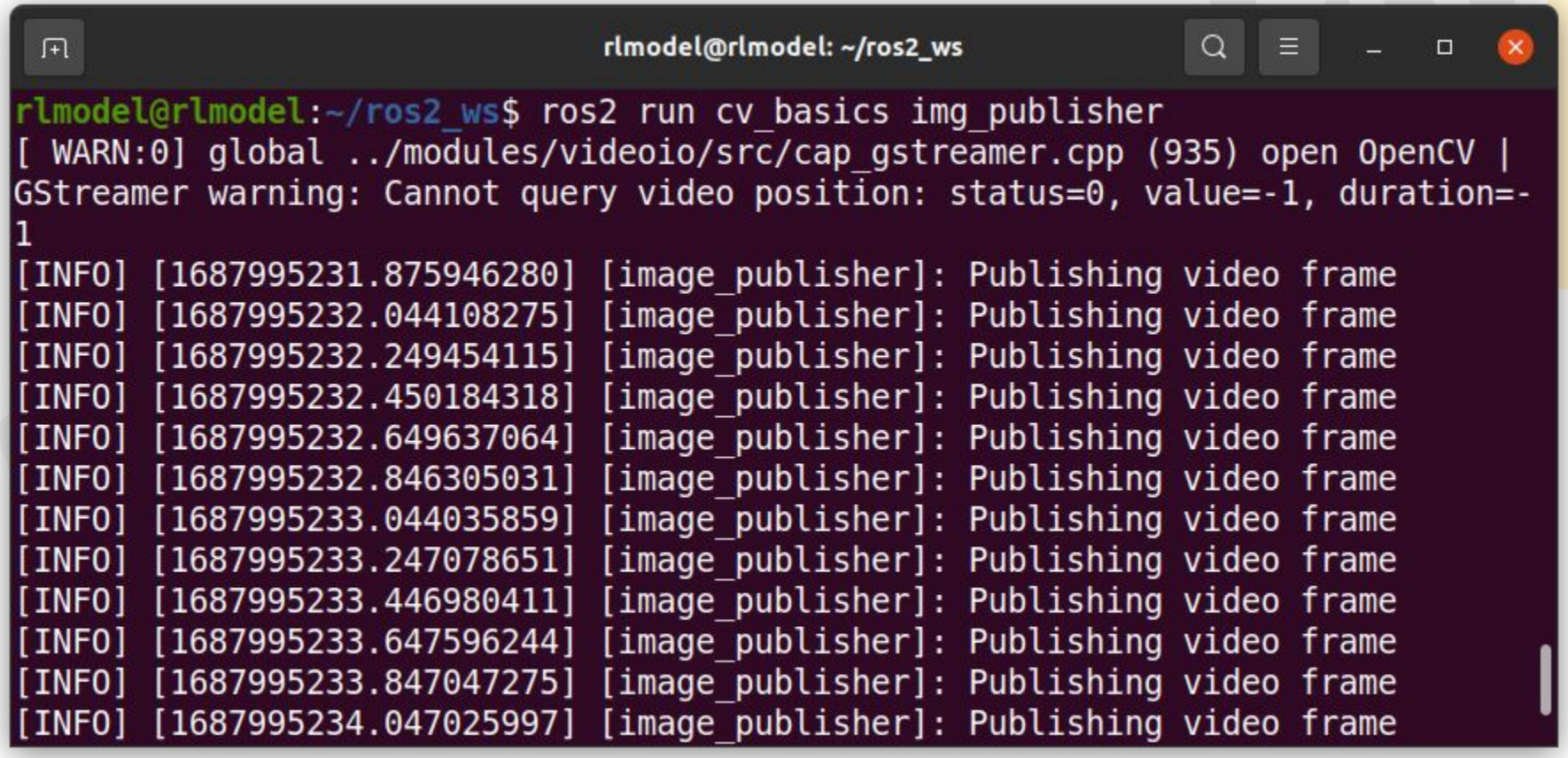
>. install/setup.bash



cv package

cv_bridge coding

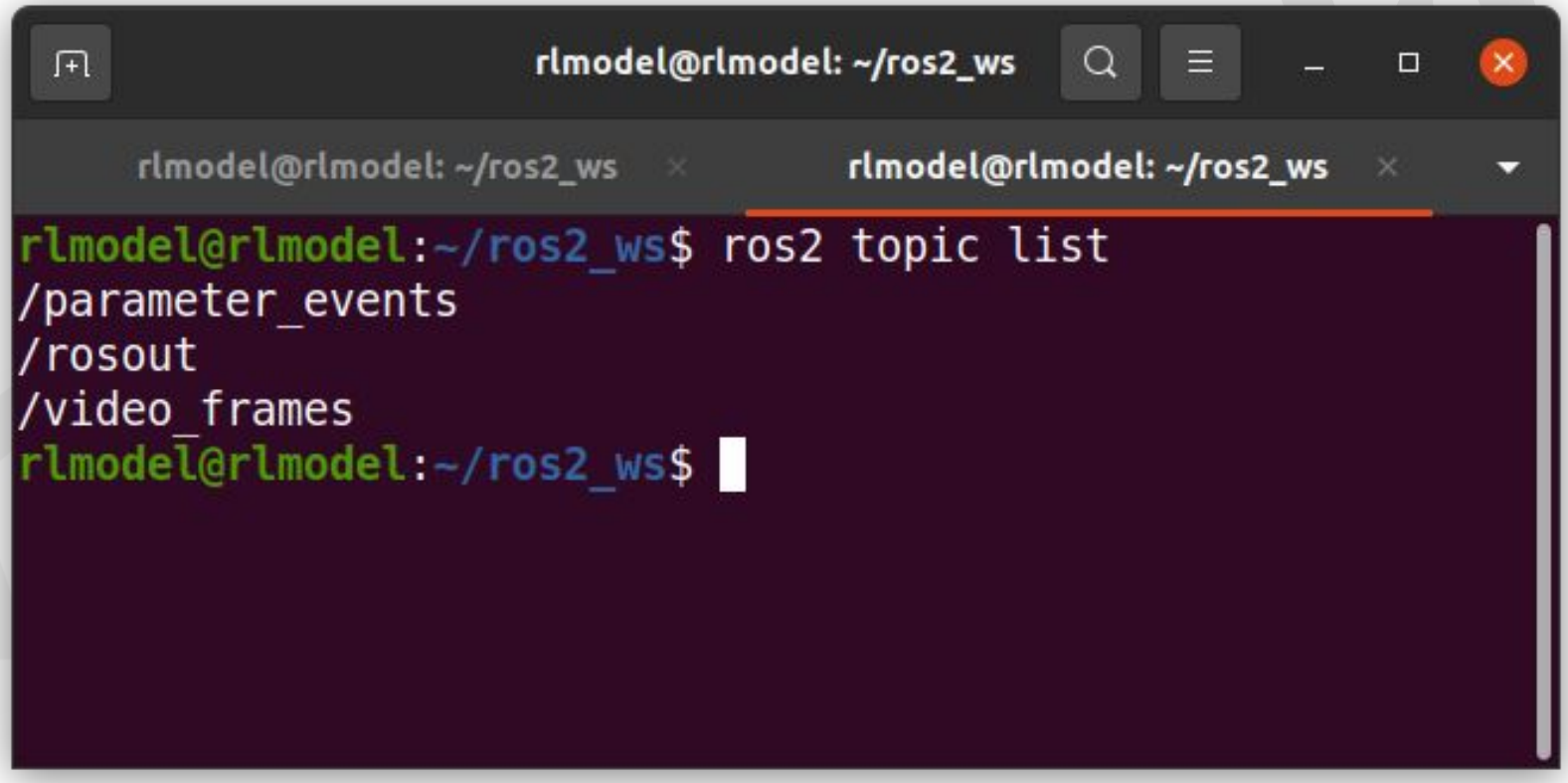
execute: ros2 run cv_basic img_publisher



cv package

cv_bridge coding

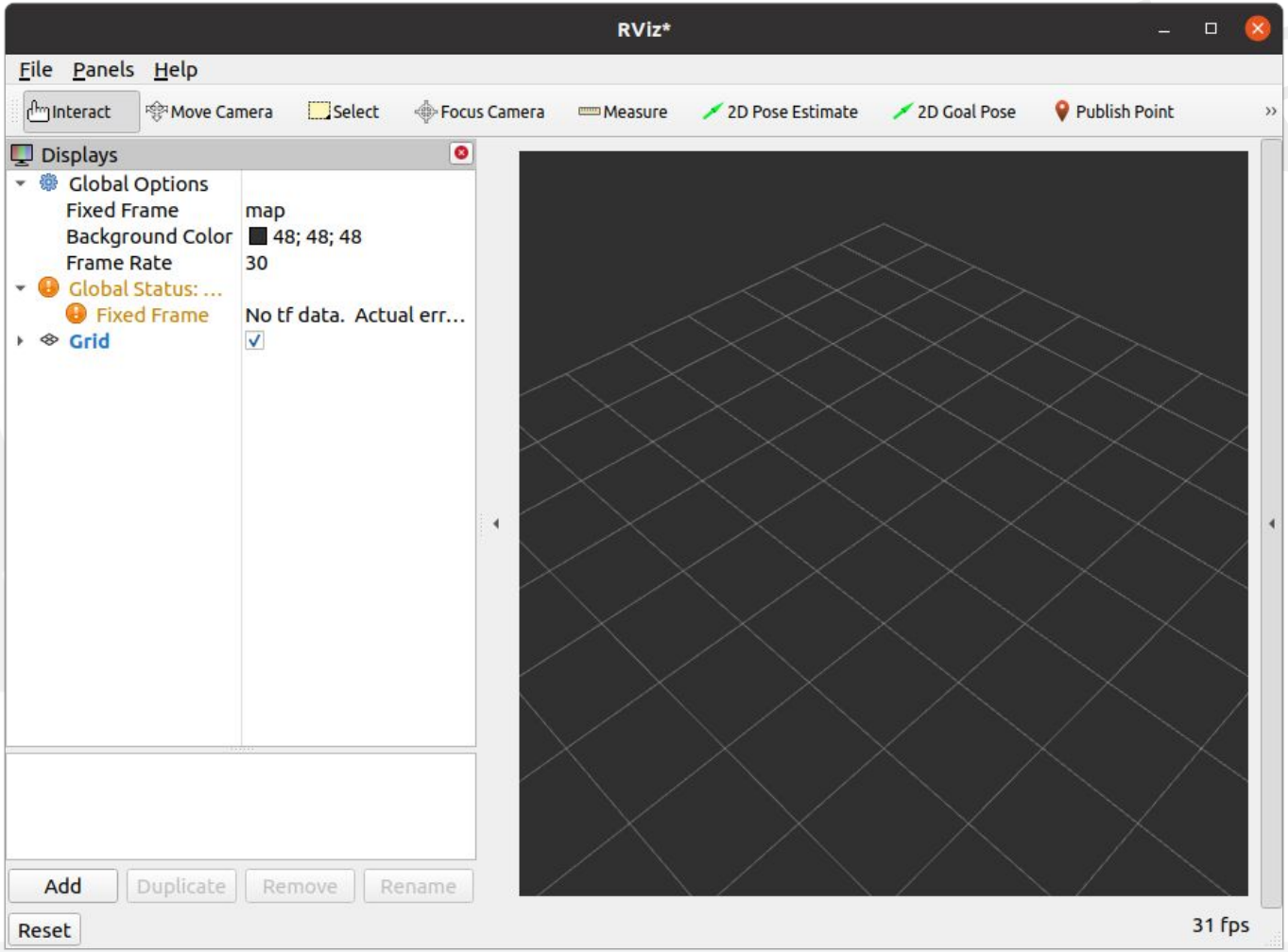
check topic: ros2 topic list



cv package

cv_bridge coding

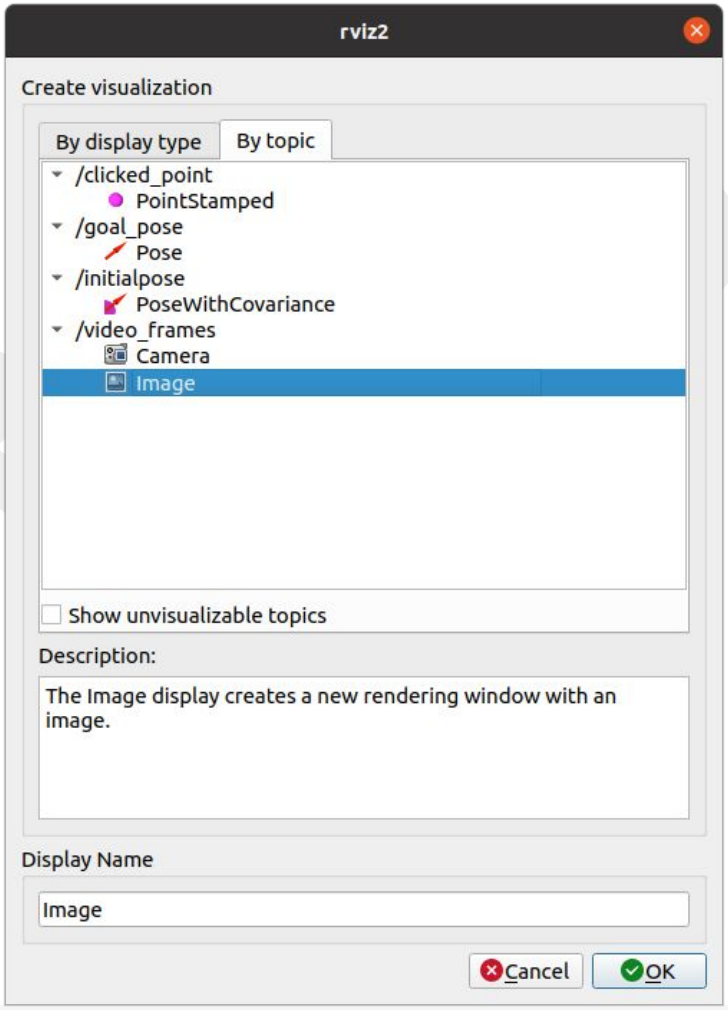
check topic: rviz



cv package

cv_bridge coding

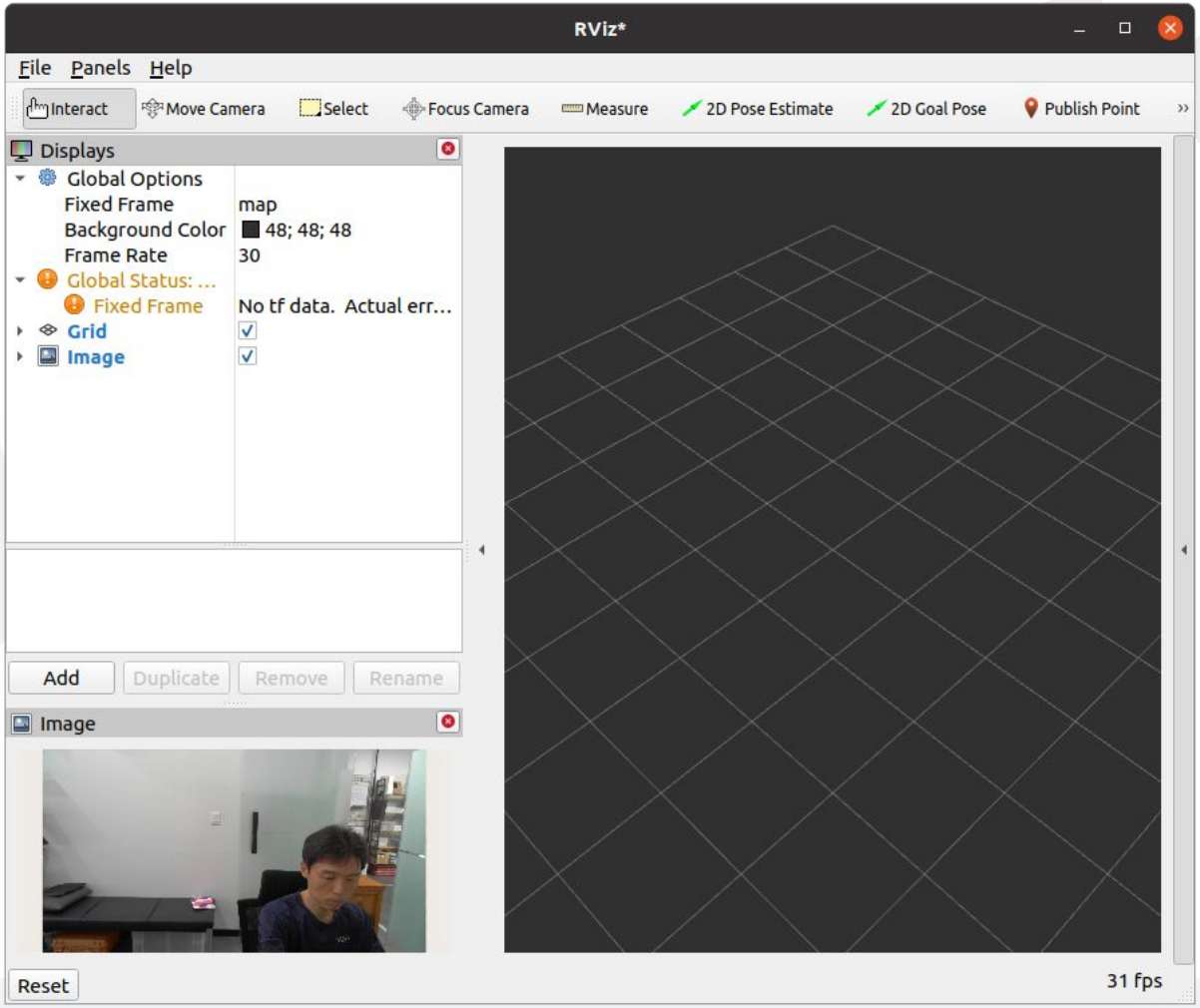
check topic: rviz



cv package

cv_bridge coding

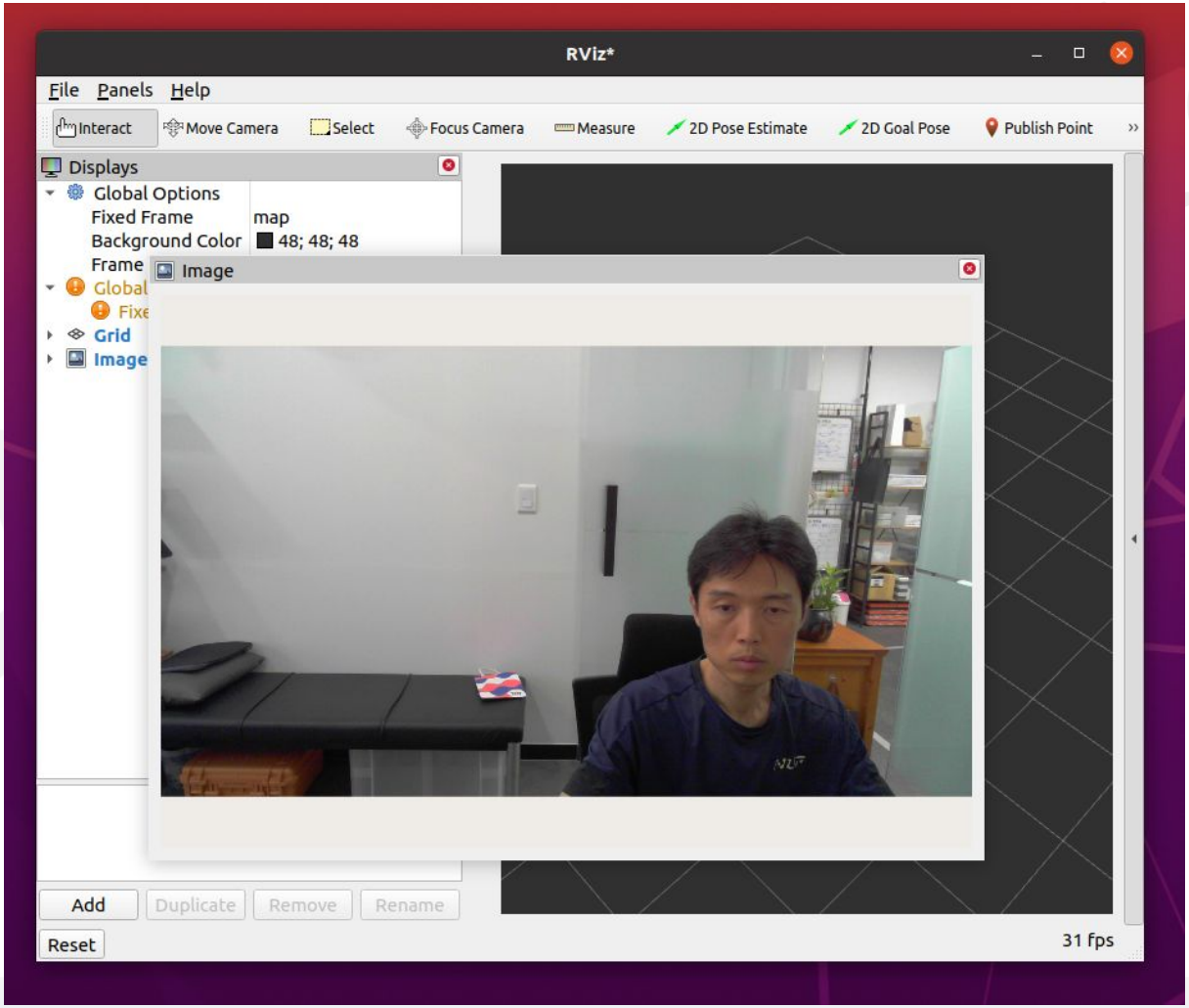
check topic: rviz



cv package

cv_bridge coding

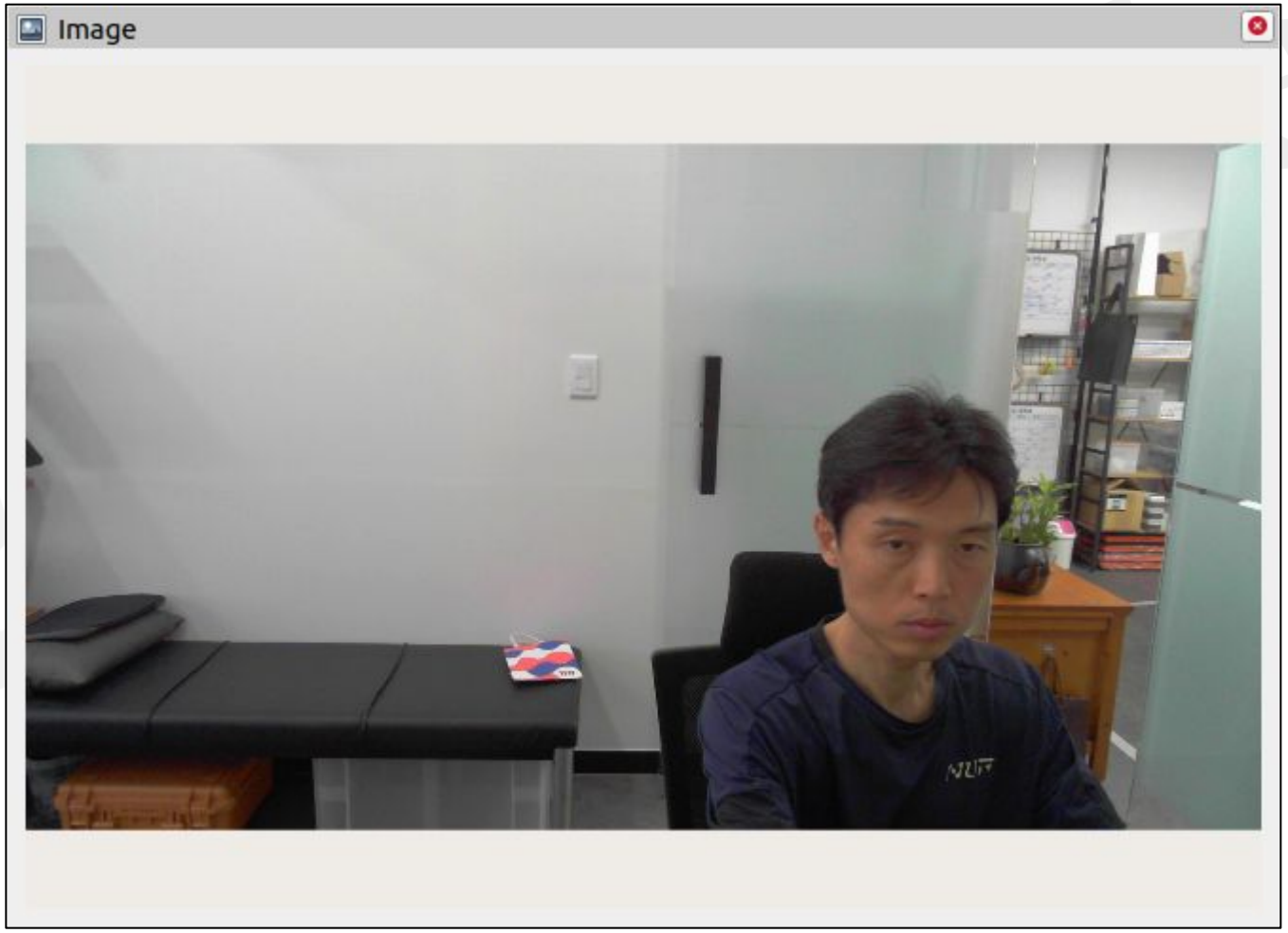
check topic: rviz



cv package

cv_bridge coding

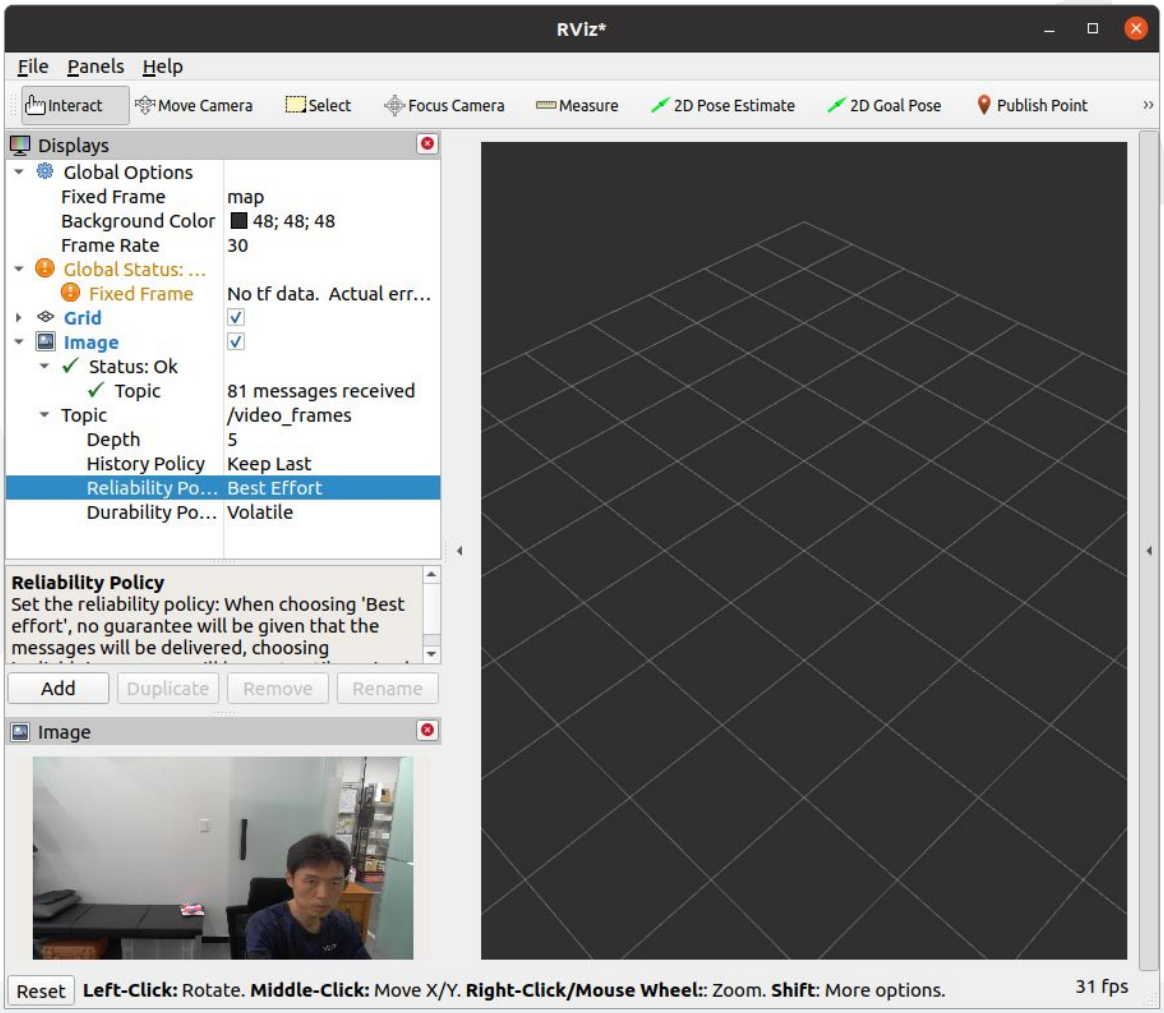
check topic: rviz



cv package

cv_bridge coding

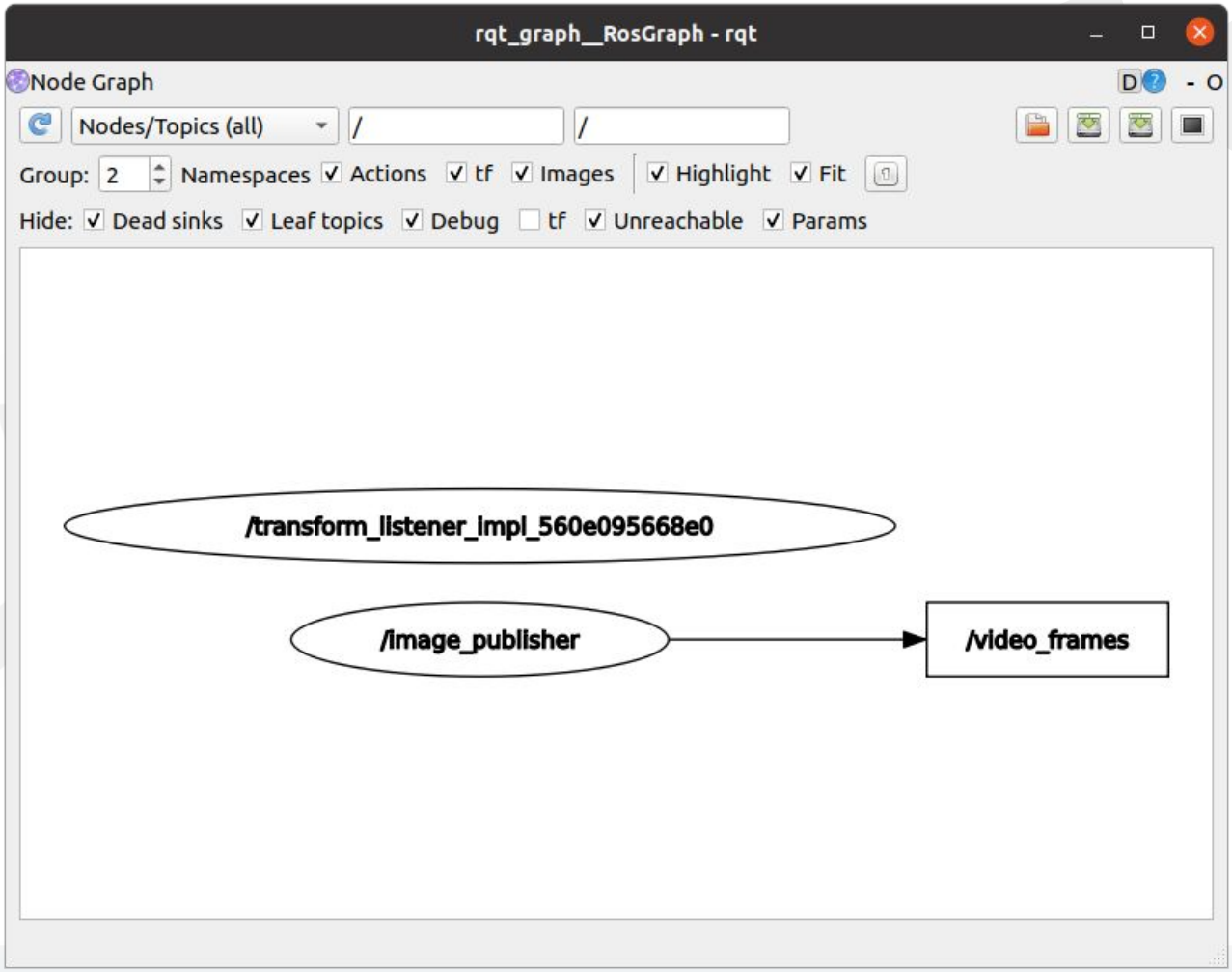
check DDS: Best effort, Reliable



cv package

cv_bridge coding

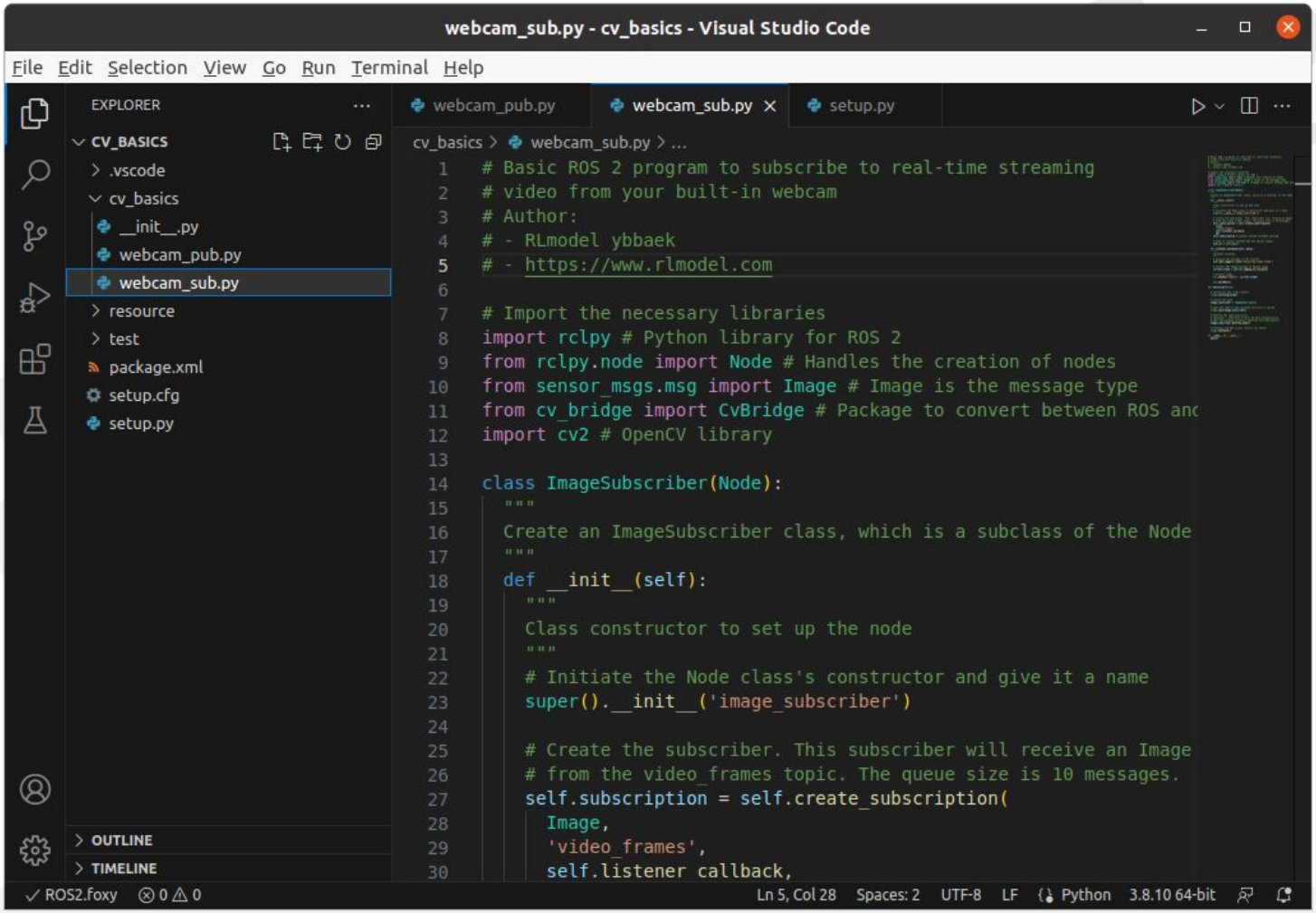
check rqt_graph: ros2 run rqt_graph rqt_graph



cv package

cv_bridge coding

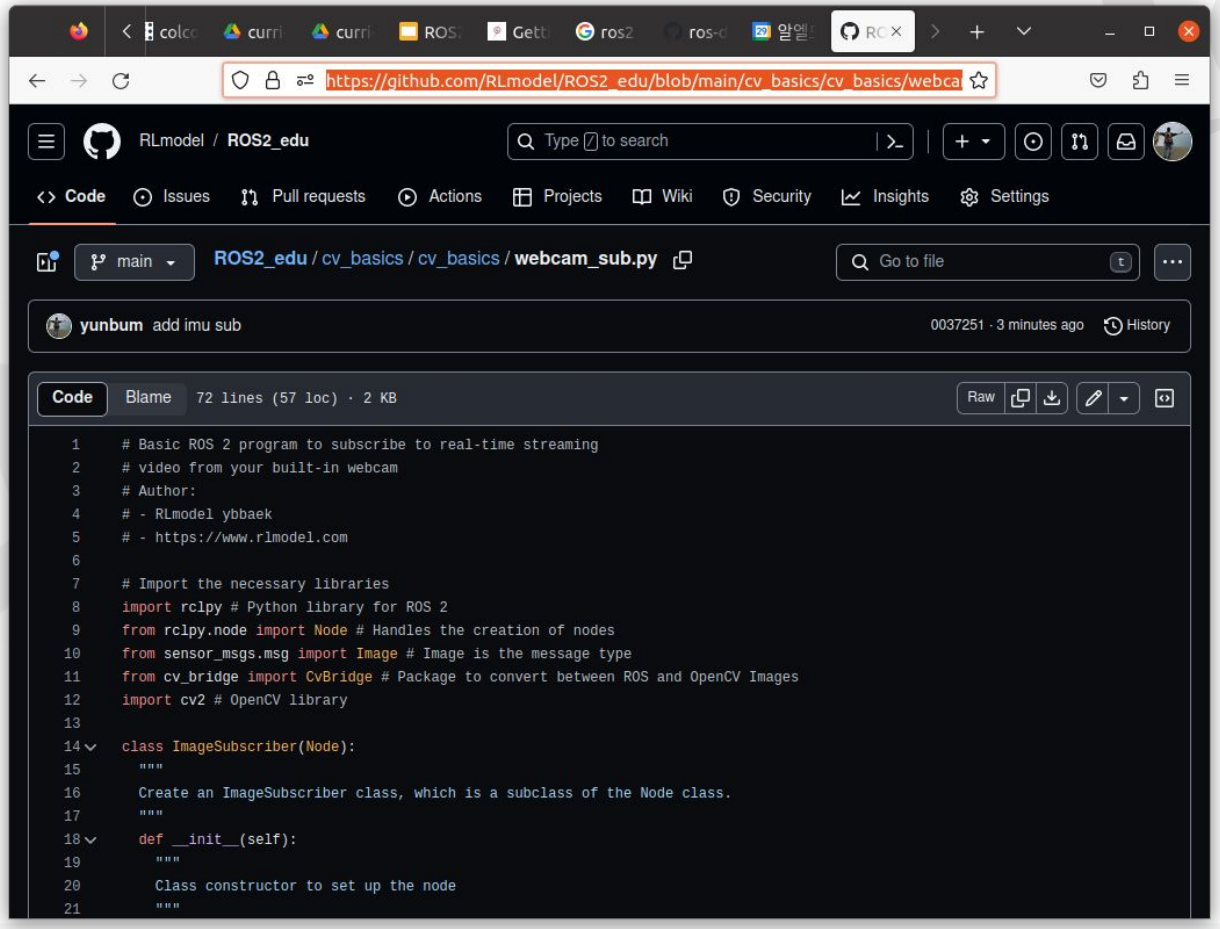
image topic subscribe: new webam_sub.py python file



cv package

cv_bridge coding

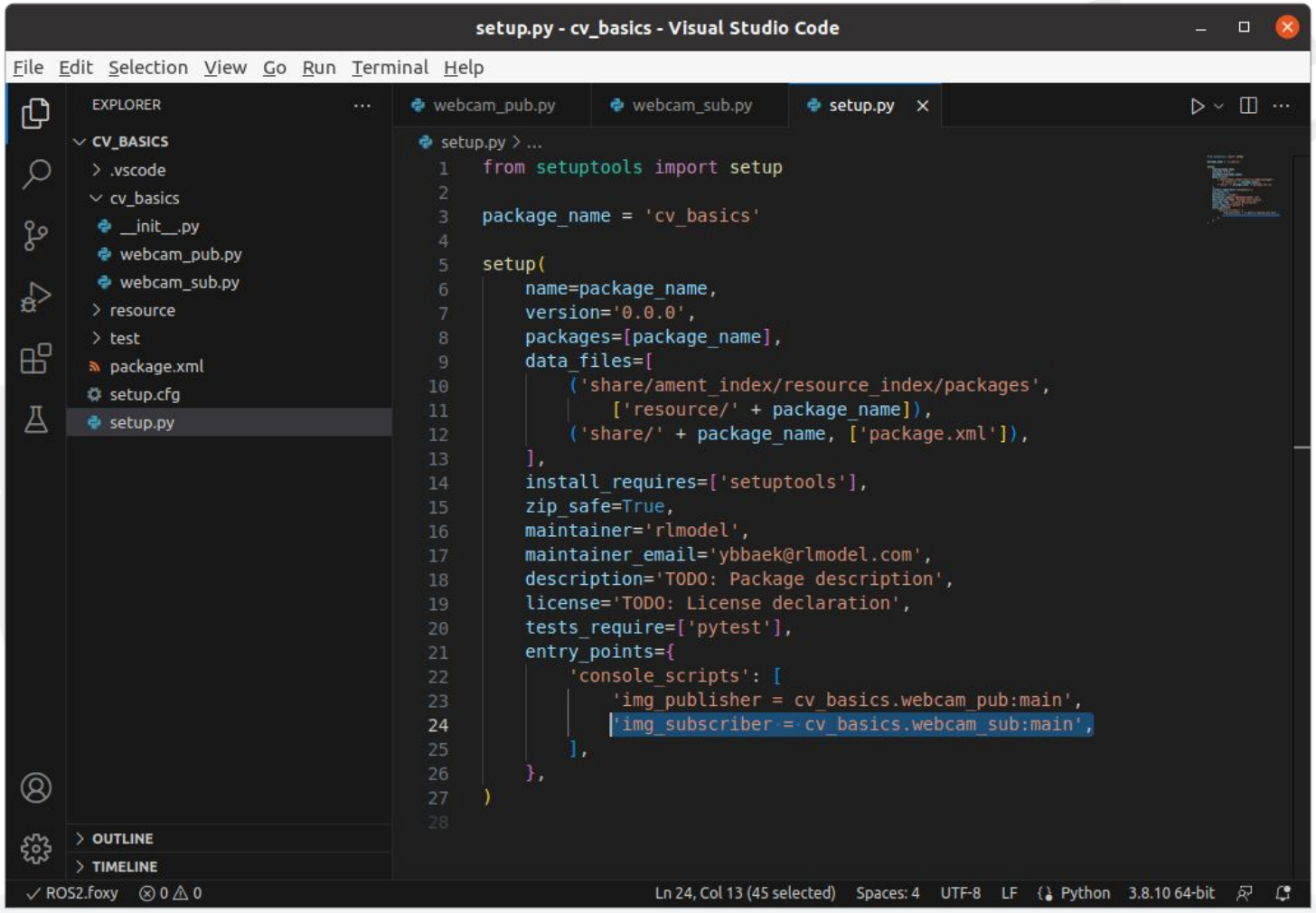
source link:
https://github.com/RLmodel/ROS2_edu/blob/main/cv_basics/cv_basics/webcam_sub.py



cv package

cv_bridge coding

update entry_points: 'img_subscriber = cv_basics.webcam_sub:main',



cv package

cv_bridge coding

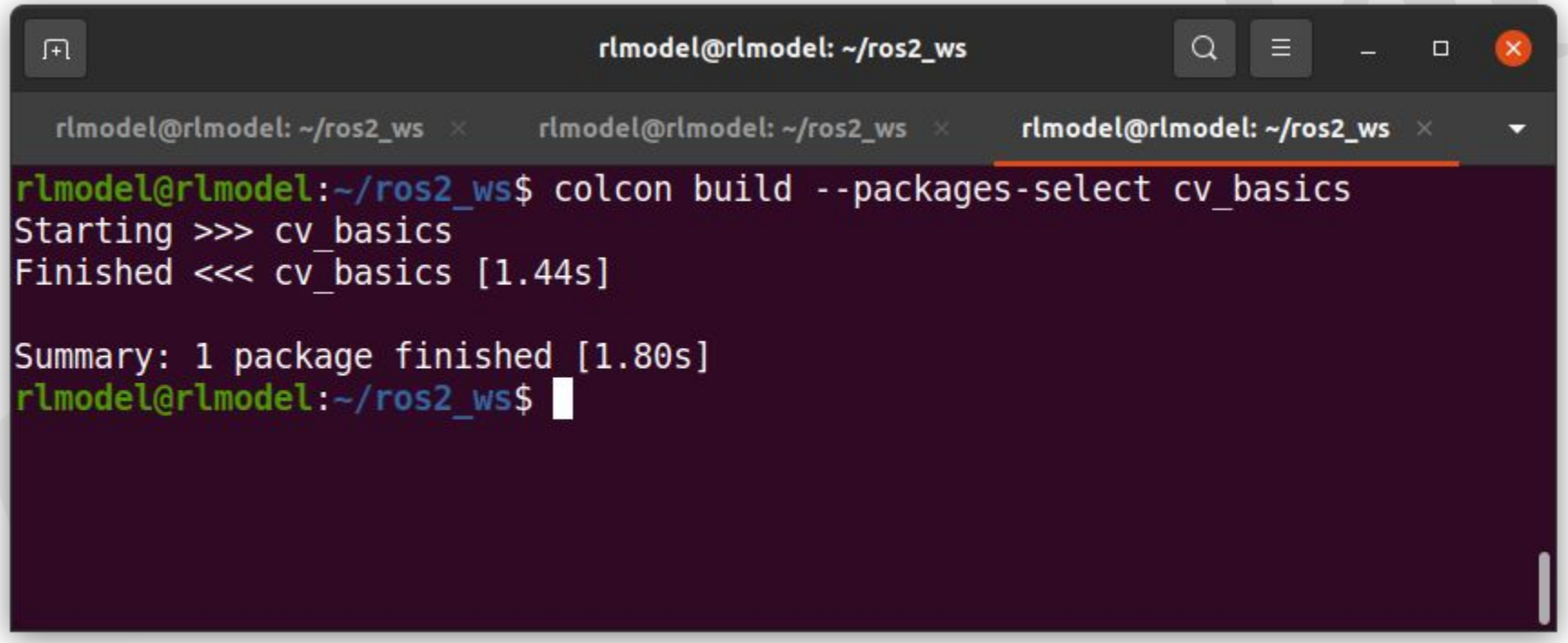
update entry_points: setup.py

```
    entry_points={
        'console_scripts': [
            'img_publisher = cv_basics.webcam_pub:main',
            'img_subscriber = cv_basics.webcam_sub:main',
        ],
    },
)
```

cv package

cv_bridge coding

build: colcon build --packages-select cv_basics

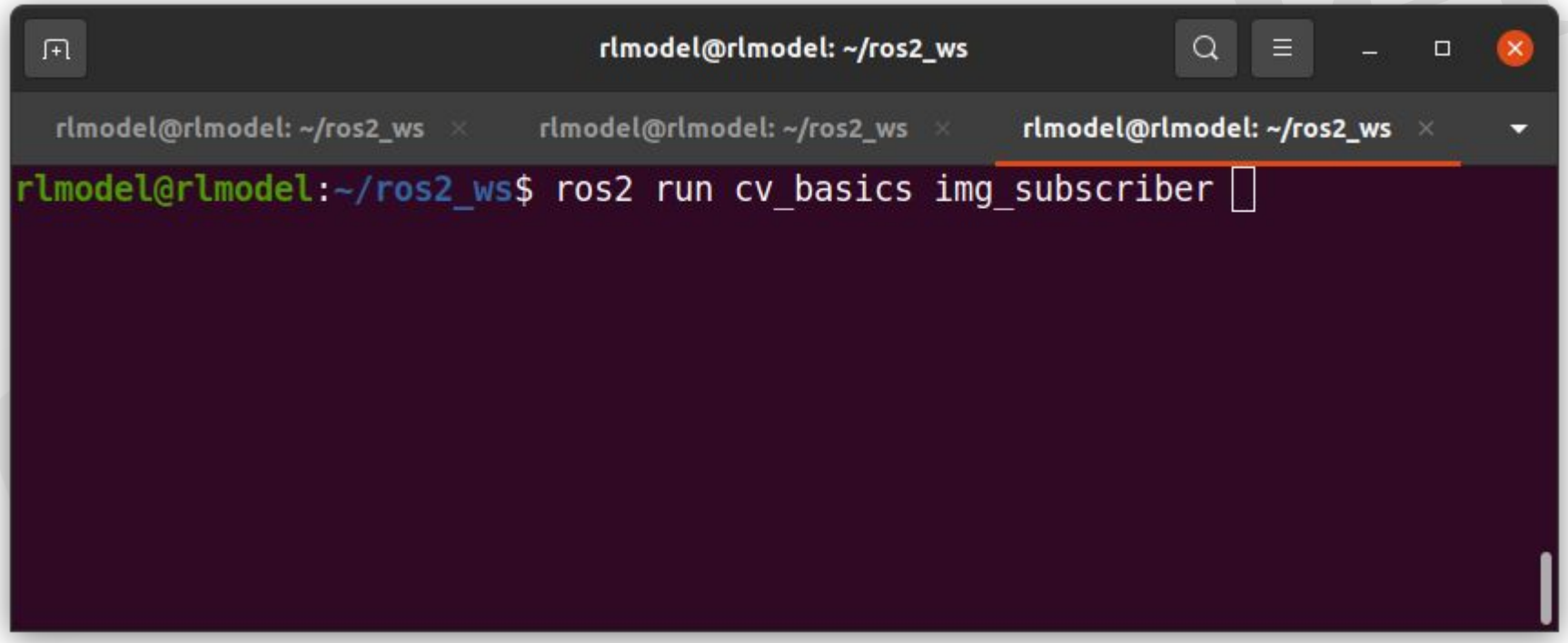


cv package

cv_bridge coding

sourcing: foxy

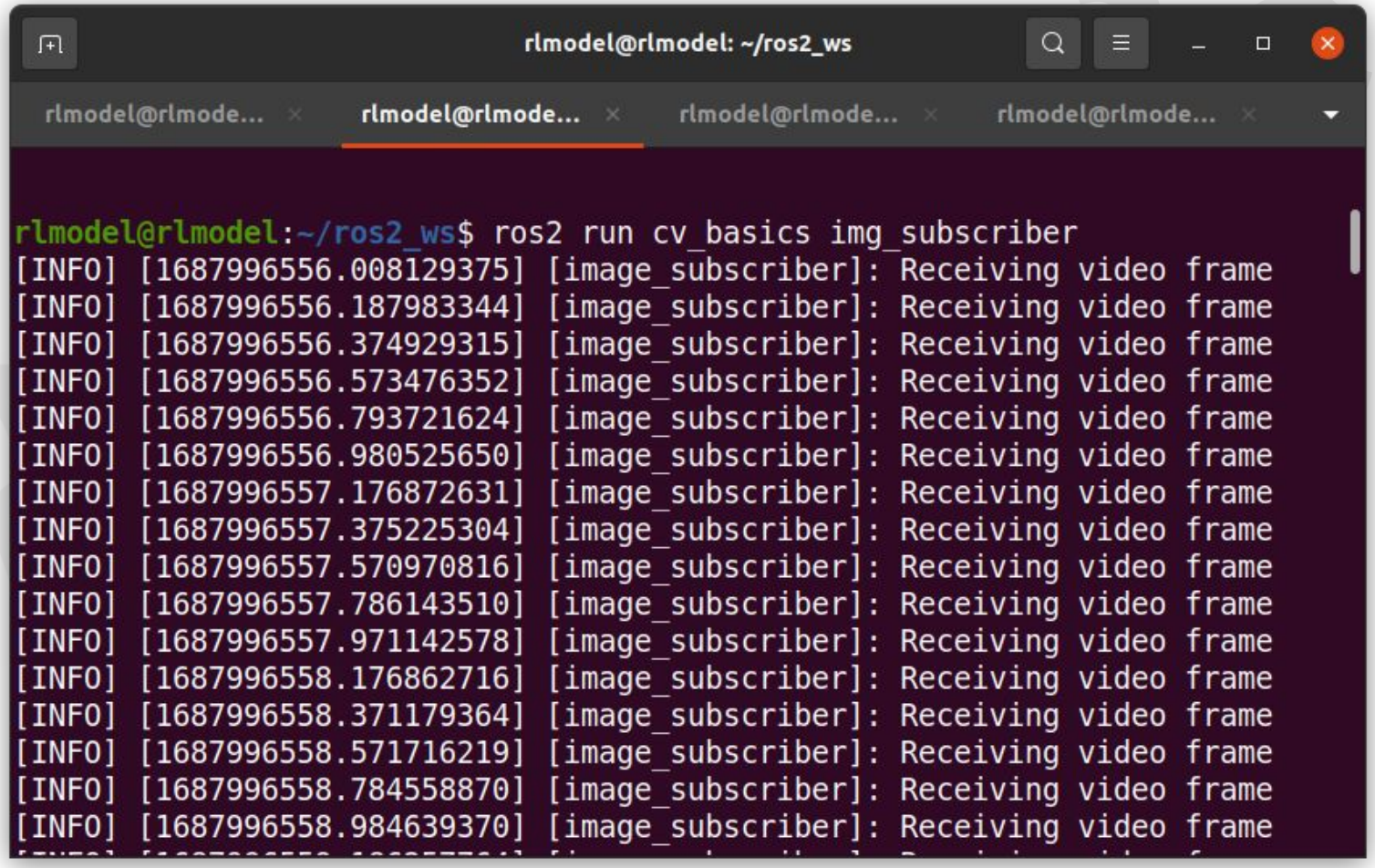
>check auto completion



cv package

cv_bridge coding

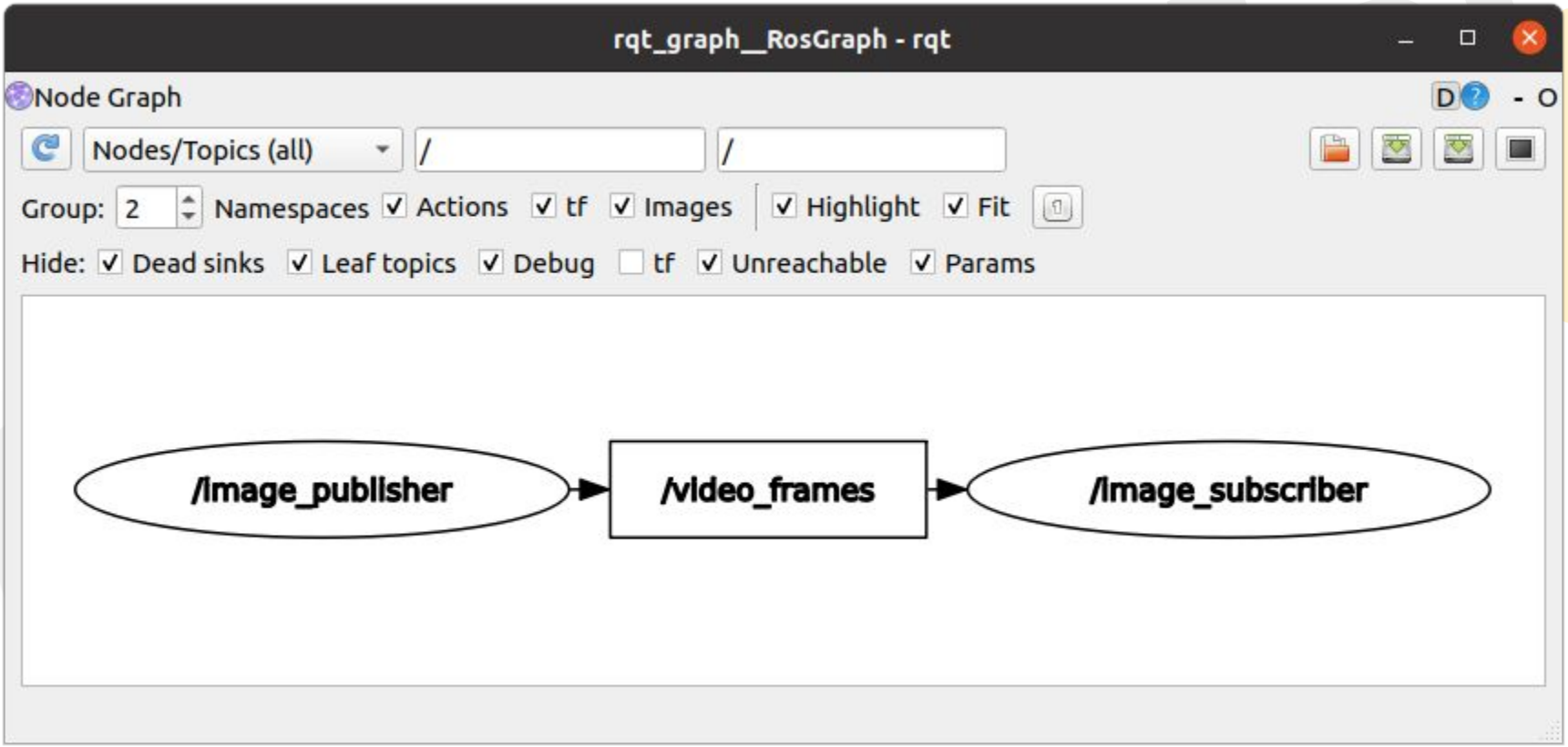
execute: img_subscriber



cv package

cv_bridge coding

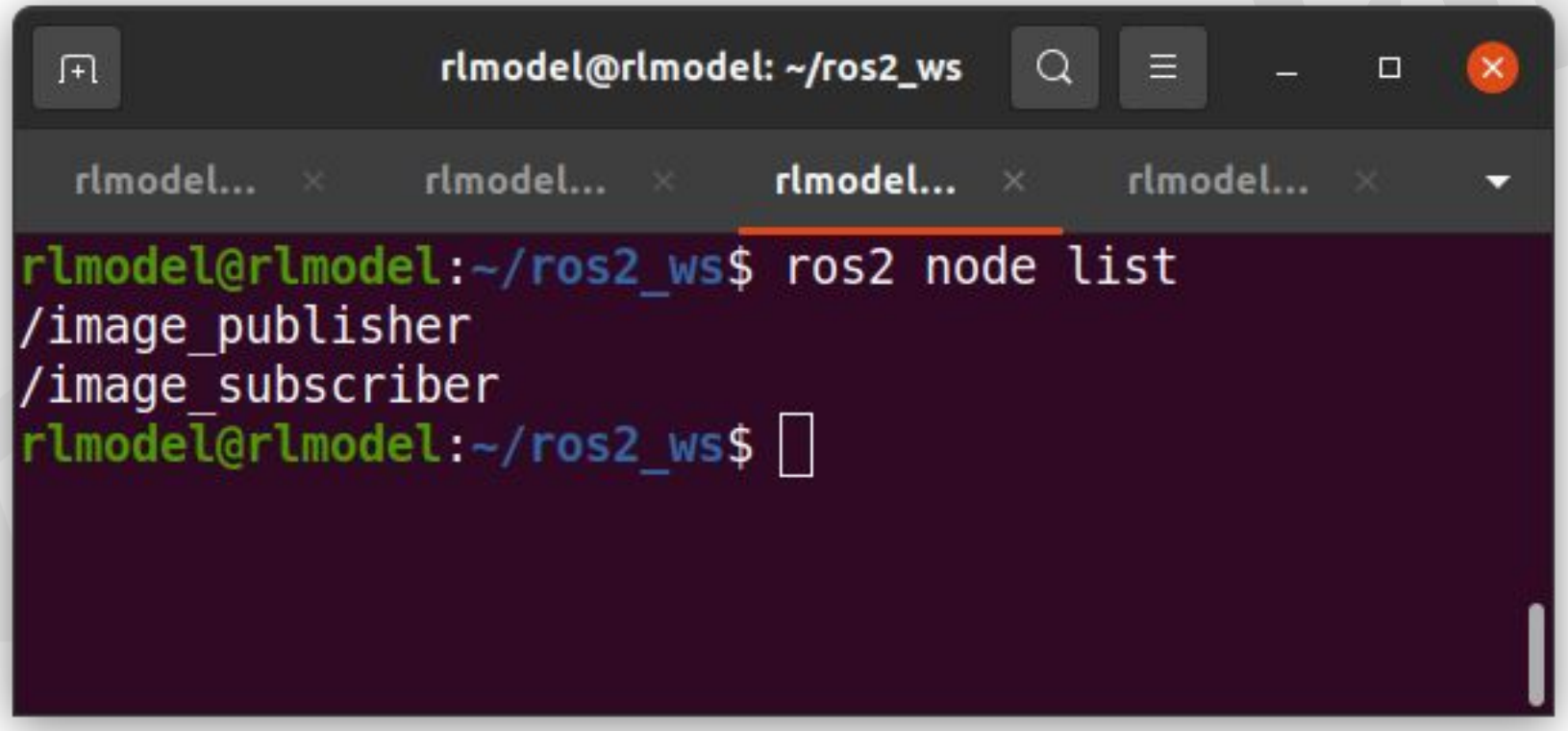
check node: rqt_graph



cv package

cv_bridge coding

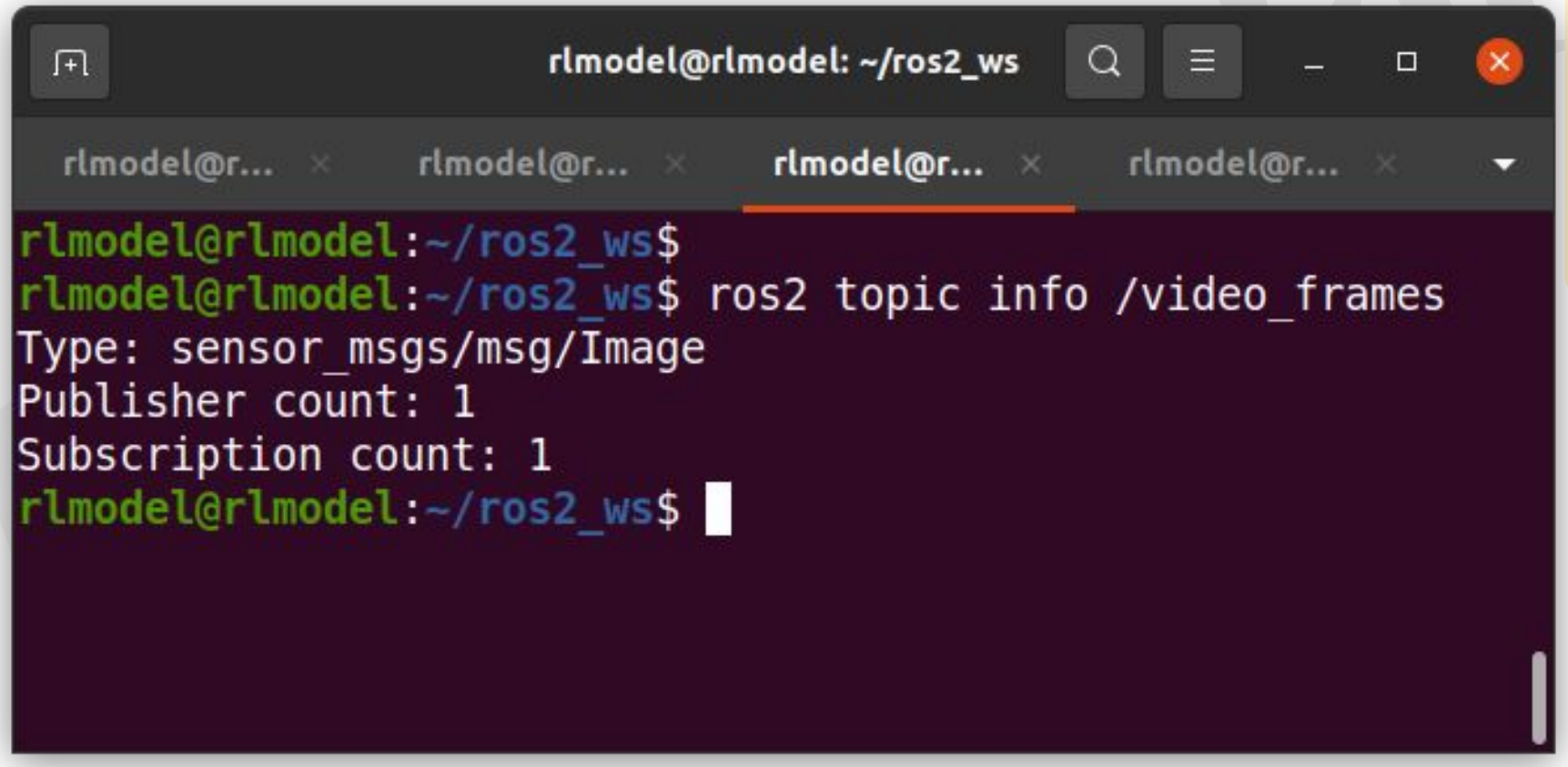
check node: ros2 node list



cv package

cv_bridge coding

check topic usage: ros2 topic info /video_frames



cv package

cv_bridge coding

sourcing: foxy

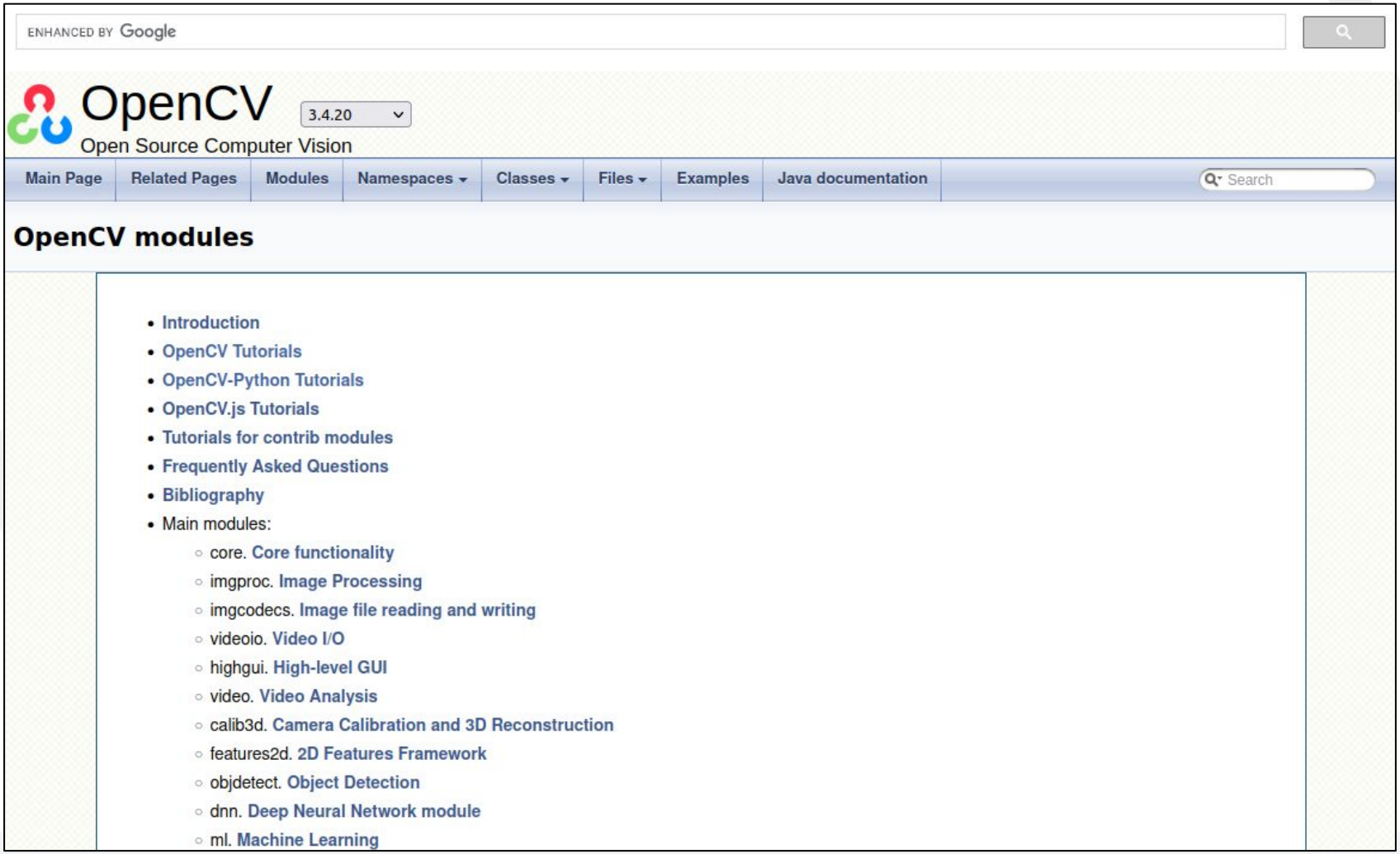
>check auto completion

Appendix

opencv

Open Source Computer Vision

link: https://docs.opencv.org/3.4/index.html



Genicam

intro

GenICam은 머신 비전 카메라용 일반 프로그래밍 인터페이스입니다.

이 표준의 목표는 사용자 애플리케이션 프로그래밍 인터페이스에서 산업용 카메라 인터페이스 기술을 분리하는 것입니다.

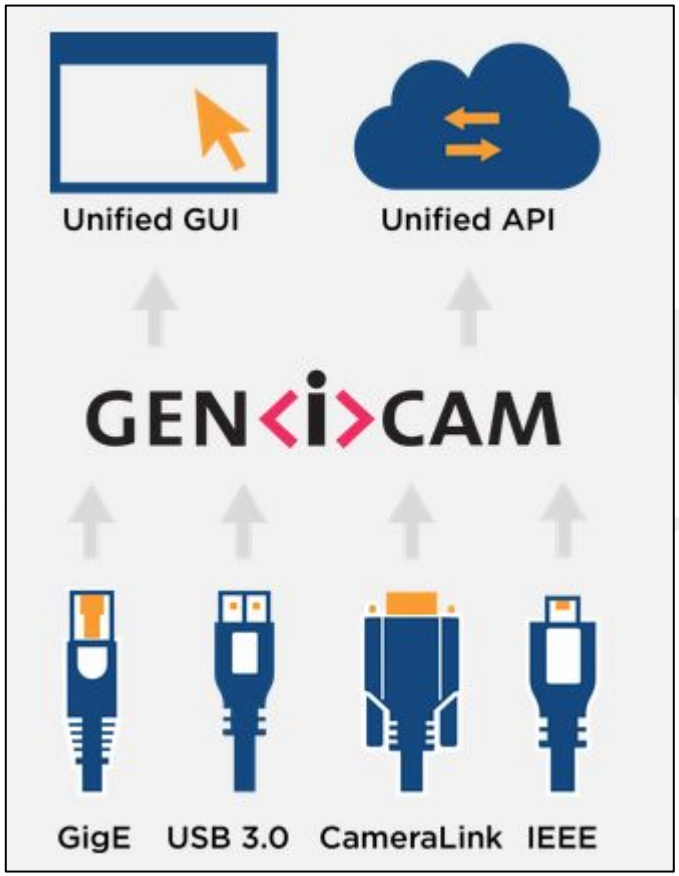
> GenICam은 유럽 머신 비전 협회에서 관리

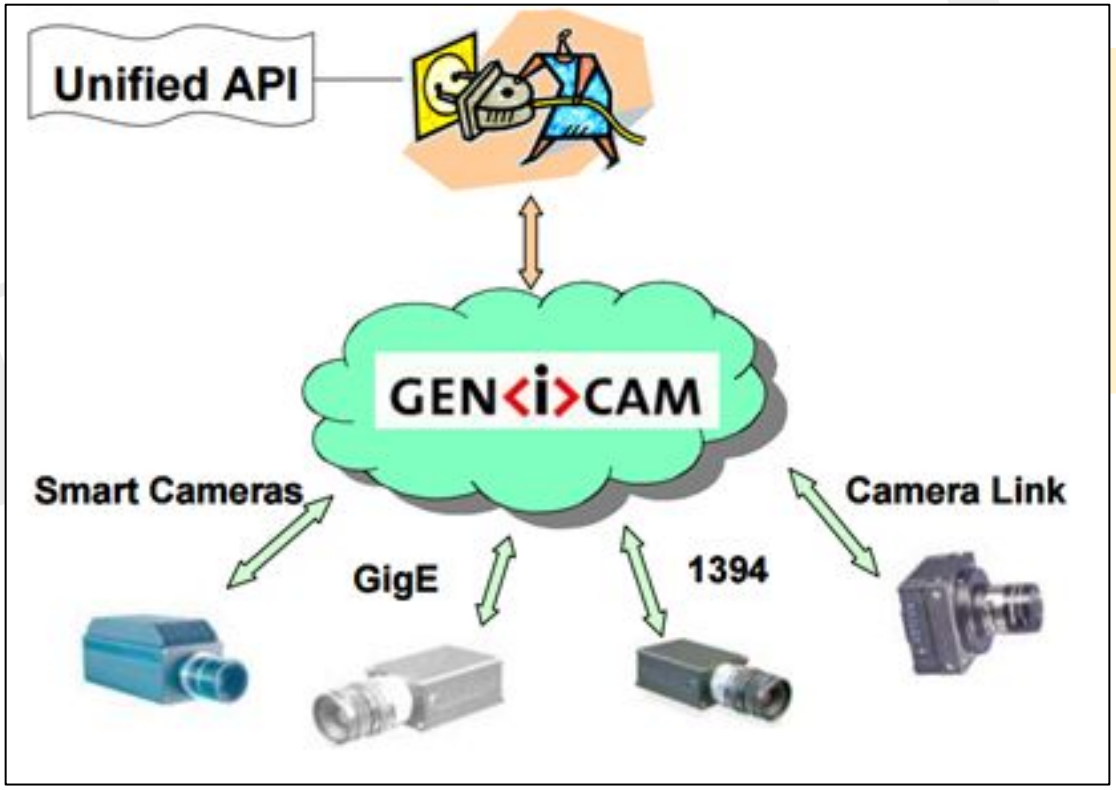
GenICam을 사용해야 하는 이유는?

GenICam은 범용 소프트웨어 인터페이스를 제공합니다. 이것은 GigE Vision, USB3 Vision, Camera Link® 및 IEEE 1394를 포함한 광범위한 표준 인터페이스를 위한 종단간 구성 인터페이스를 제공해 모든 카메라 종류 및 이미지 형식을 커버합니다. 이 방식은 카메라별 구성 필요 없이 GenICam 표준과 호환되는 카메라를 쉽게 연결할 수 있게 만들어줍니다.

Genicam

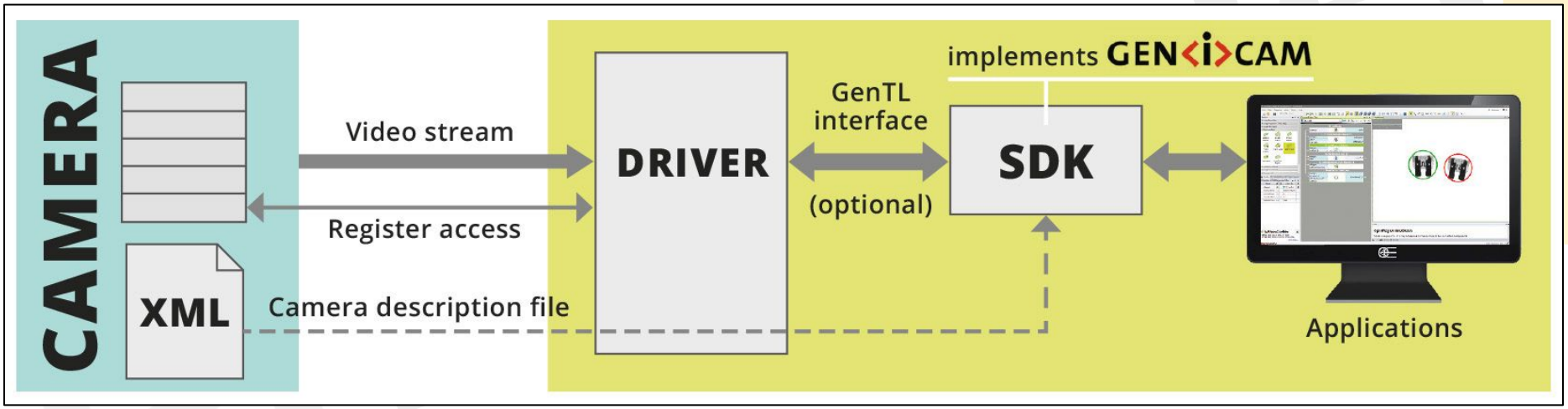
Configuration





Genicam

Block diagram



video viewer

guvcview

명령어: $ sudo apt-get install guvcview