

## EXPERIMENT 10

### Basics of ROS

#### Aim:

To install the ROS Noetic version and learn to create a workspace in ROS and to run simple program.

#### Software/ Package Used:

- Ubuntu 18.04
- ROS - Noetic

#### Programs:

**1. Write a ROS program to configure a node and send a message and configure two different users to receive the same message.**

#### PUBLISHER CODE:

```
#!/usr/bin/env python3
import rospy
from std_msgs.msg import String
def talker():
    pub = rospy.Publisher('chatter', String, queue_size=10)
    rospy.init_node('talker', anonymous=True)
    rate = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        hello_str = "hello world %s" % rospy.get_time()
        rospy.loginfo(hello_str)
        pub.publish(hello_str)
        rate.sleep()
if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException:
        pass
```

#### SUBSCRIBER 1 CODE:

```
#!/usr/bin/env python3
import rospy
from std_msgs.msg import String
def callback(data):
    rospy.loginfo(rospy.get_caller_id() + 'I heard %s', data.data)
def listener():
```

```

rospy.init_node('listener', anonymous=True)
rospy.Subscriber('chatter', String, callback)
# spin() simply keeps python from exiting until this node is stopped
rospy.spin()

if __name__ == '__main__':
    listener()

```

## SUBSCRIBER 2 CODE:

```

#!/usr/bin/env python3
import rospy
from std_msgs.msg import String
def callback(data):
    rospy.loginfo(rospy.get_caller_id() + 'I heard as well %s', data.data)
def listener():
    rospy.init_node('listener', anonymous=True)
    rospy.Subscriber('chatter', String, callback)
    rospy.spin()
if __name__ == '__main__':
    listener()

```

## OUTPUT:

```

roscat http://raec40:11311/
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://raec40:46143/
ros_comm version 1.16.0

SUMMARY
*****
PARAMETERS
 * /roslaunch: noetic
 * /rosversion: 1.16.0

NODES
auto-starting new master
process[master]: started with pid [7954]
ROS_MASTER_URI=http://raec40:11311/

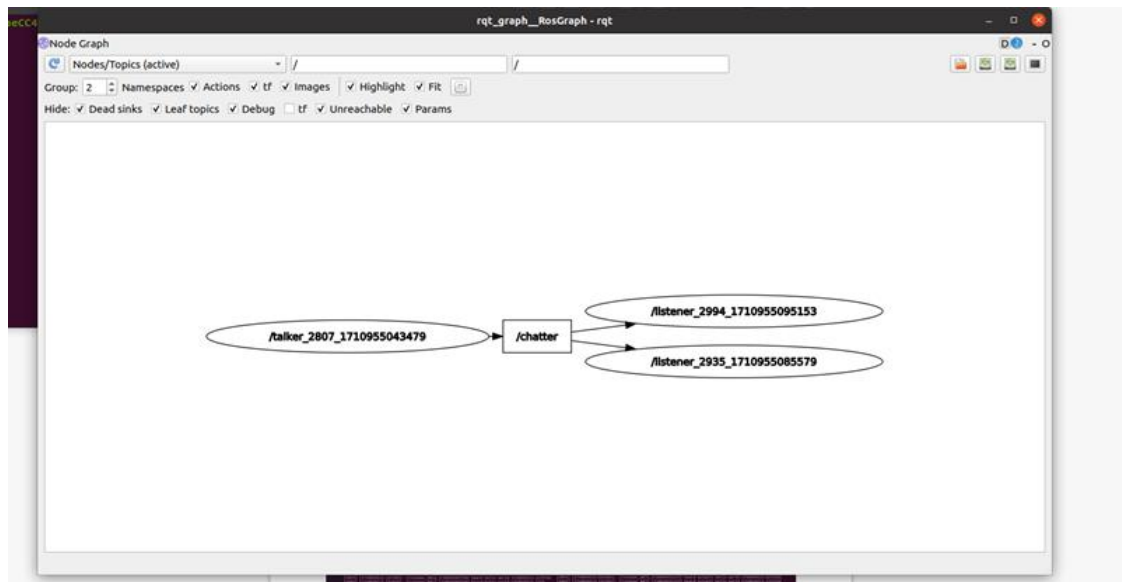
setting /run_id to 4d93872e-e6a1-11ee-8413-594366903e84
process[roscout-1]: started with pid [7964]
started core service [/roscout]

[roscout-1] [1710929512.727602]: hello world 1710929512.727303
[roscout-1] [1710929512.727603]: hello world 1710929512.727305
[roscout-1] [1710929512.727603]: hello world 1710929512.727307
[roscout-1] [1710929513.027491]: hello world 1710929513.027226
[roscout-1] [1710929513.127569]: hello world 1710929513.127226
[roscout-1] [1710929513.227595]: hello world 1710929513.2273
[roscout-1] [1710929513.327599]: hello world 1710929513.327284
[roscout-1] [1710929513.427604]: hello world 1710929513.427316
[roscout-1] [1710929513.527586]: hello world 1710929513.527293
[roscout-1] [1710929513.627591]: hello world 1710929513.627284
[roscout-1] [1710929513.727565]: hello world 1710929513.727268
[roscout-1] [1710929513.827602]: hello world 1710929513.827307
[roscout-1] [1710929513.927578]: hello world 1710929513.927294
[roscout-1] [1710929514.027544]: hello world 1710929514.027278
[roscout-1] [1710929514.127503]: hello world 1710929514.127254
[roscout-1] [1710929514.227596]: hello world 1710929514.227297
[roscout-1] [1710929514.327609]: hello world 1710929514.327306
[roscout-1] [1710929514.427607]: hello world 1710929514.427136
[roscout-1] [1710929514.527543]: hello world 1710929514.527211
[roscout-1] [1710929514.627612]: hello world 1710929514.627317
[roscout-1] [1710929514.727586]: hello world 1710929514.727254
[roscout-1] [1710929514.827575]: hello world 1710929514.827262
[roscout-1] [1710929514.927602]: hello world 1710929514.927301

[roscout-1] [1710929512.831701]: /listener_8812_17109293072501 heard hello world 1710929512.8273025
[roscout-1] [1710929512.931700]: /listener_8812_17109293072501 heard hello world 1710929512.9273007
[roscout-1] [1710929513.031601]: /listener_8812_17109293072501 heard hello world 1710929513.0272226
[roscout-1] [1710929513.131704]: /listener_8812_17109293072501 heard hello world 1710929513.1272726
[roscout-1] [1710929513.231657]: /listener_8812_17109293072501 heard hello world 1710929513.2273
[roscout-1] [1710929513.331698]: /listener_8812_17109293072501 heard hello world 1710929513.3272804
[roscout-1] [1710929513.431744]: /listener_8812_17109293072501 heard hello world 1710929513.4273126
[roscout-1] [1710929513.531109]: /listener_8812_17109293072501 heard hello world 1710929513.5272593
[roscout-1] [1710929513.631764]: /listener_8812_17109293072501 heard hello world 1710929513.6272864
[roscout-1] [1710929513.731703]: /listener_8812_17109293072501 heard hello world 1710929513.7272868
[roscout-1] [1710929513.831681]: /listener_8812_17109293072501 heard hello world 1710929513.8273027
[roscout-1] [1710929513.931729]: /listener_8812_17109293072501 heard hello world 1710929513.9272954
[roscout-1] [1710929514.031565]: /listener_8812_17109293072501 heard hello world 1710929514.0272708
[roscout-1] [1710929514.131084]: /listener_8812_17109293072501 heard hello world 1710929514.1272504
[roscout-1] [1710929514.231790]: /listener_8812_17109293072501 heard hello world 1710929514.2272997
[roscout-1] [1710929514.331786]: /listener_8812_17109293072501 heard hello world 1710929514.3273056
[roscout-1] [1710929514.430351]: /listener_8812_17109293072501 heard hello world 1710929514.4271736
[roscout-1] [1710929514.531732]: /listener_8812_17109293072501 heard hello world 1710929514.527211
[roscout-1] [1710929514.631723]: /listener_8812_17109293072501 heard hello world 1710929514.6273117
[roscout-1] [1710929514.731699]: /listener_8812_17109293072501 heard hello world 1710929514.7272854
[roscout-1] [1710929514.831692]: /listener_8812_17109293072501 heard hello world 1710929514.827262
[roscout-1] [1710929514.931703]: /listener_8812_17109293072501 heard hello world 1710929514.927301

[roscout-1] [1710929512.731871]: /listener_8872_17109293311571 heard as well hello world 1710929512.727303
[roscout-1] [1710929512.831819]: /listener_8872_17109293311571 heard as well hello world 1710929512.8273025
[roscout-1] [1710929512.931792]: /listener_8872_17109293311571 heard as well hello world 1710929512.9273007
[roscout-1] [1710929513.031384]: /listener_8872_17109293311571 heard as well hello world 1710929513.0272226
[roscout-1] [1710929513.131793]: /listener_8872_17109293311571 heard as well hello world 1710929513.1272726
[roscout-1] [1710929513.231764]: /listener_8872_17109293311571 heard as well hello world 1710929513.2273
[roscout-1] [1710929513.331779]: /listener_8872_17109293311571 heard as well hello world 1710929513.3272804
[roscout-1] [1710929513.431830]: /listener_8872_17109293311571 heard as well hello world 1710929513.4273126
[roscout-1] [1710929513.531150]: /listener_8872_17109293311571 heard as well hello world 1710929513.5272593
[roscout-1] [1710929513.631855]: /listener_8872_17109293311571 heard as well hello world 1710929513.6272864
[roscout-1] [1710929513.731802]: /listener_8872_17109293311571 heard as well hello world 1710929513.7272868
[roscout-1] [1710929513.831765]: /listener_8872_17109293311571 heard as well hello world 1710929513.8273027
[roscout-1] [1710929513.931816]: /listener_8872_17109293311571 heard as well hello world 1710929513.9272954
[roscout-1] [1710929514.031569]: /listener_8872_17109293311571 heard as well hello world 1710929514.0272708
[roscout-1] [1710929514.131152]: /listener_8872_17109293311571 heard as well hello world 1710929514.1272504
[roscout-1] [1710929514.231888]: /listener_8872_17109293311571 heard as well hello world 1710929514.2272997
[roscout-1] [1710929514.331800]: /listener_8872_17109293311571 heard as well hello world 1710929514.3273056
[roscout-1] [1710929514.430361]: /listener_8872_17109293311571 heard as well hello world 1710929514.4271736
[roscout-1] [1710929514.531822]: /listener_8872_17109293311571 heard as well hello world 1710929514.527211
[roscout-1] [1710929514.631804]: /listener_8872_17109293311571 heard as well hello world 1710929514.6273117
[roscout-1] [1710929514.731792]: /listener_8872_17109293311571 heard as well hello world 1710929514.7272854
[roscout-1] [1710929514.831743]: /listener_8872_17109293311571 heard as well hello world 1710929514.827262
[roscout-1] [1710929514.931806]: /listener_8872_17109293311571 heard as well hello world 1710929514.927301

```

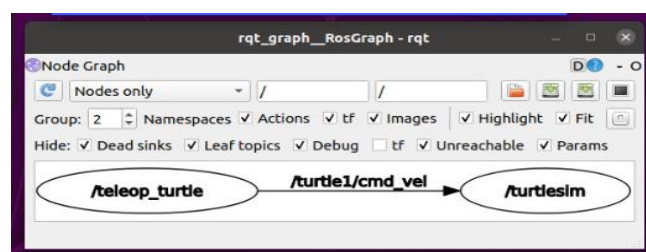
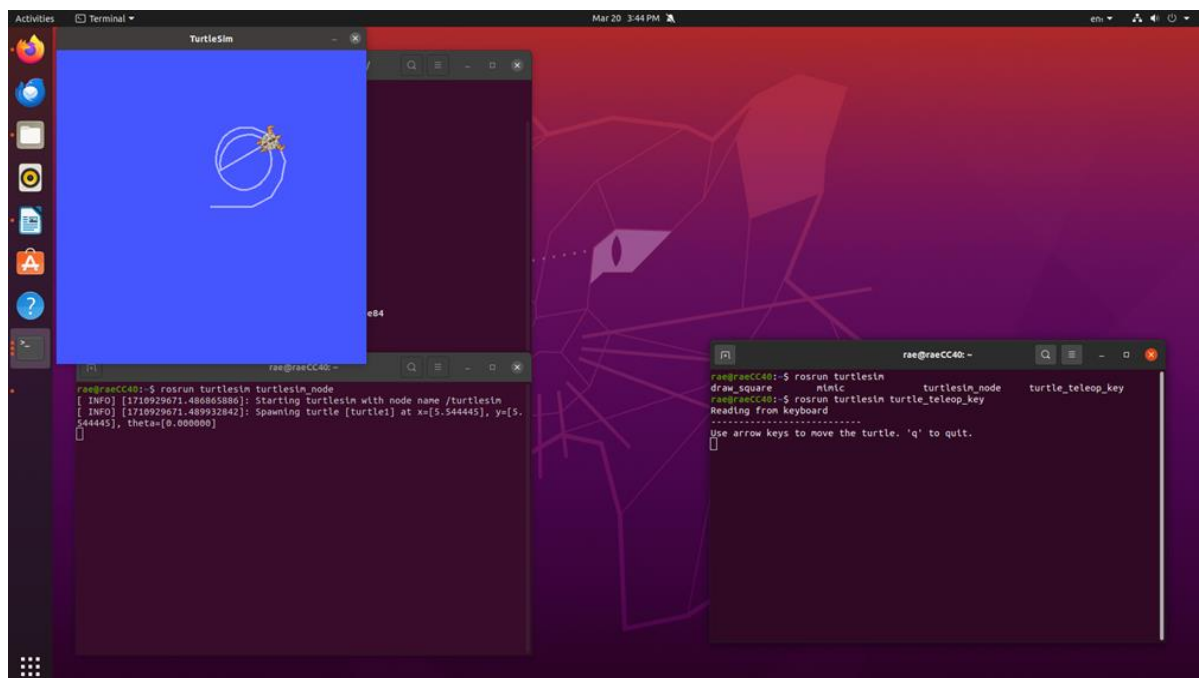


## 2. Run turtlesim

### COMMANDS:

- `roslaunch turtlesim turtlesim_node`
- `roslaunch turtlesim turtle_teleop_key`
- `rqt_graph`

### OUTPUT:

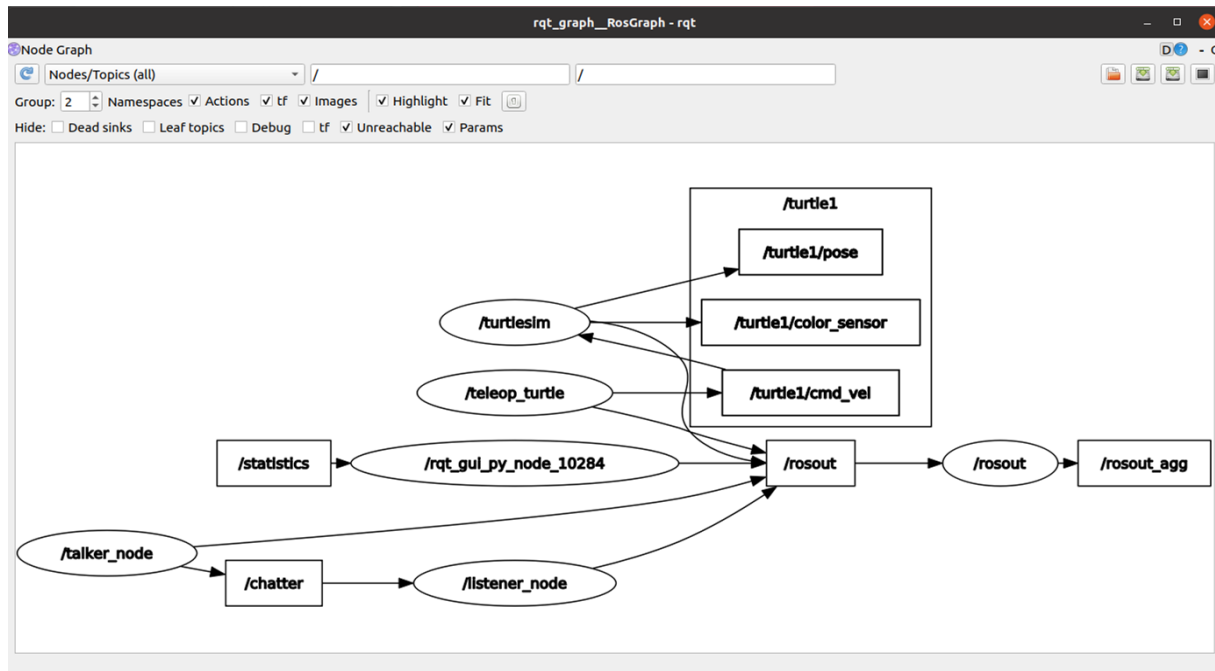


### 3. RQT Graph.

#### COMMANDS:

- rqt\_graph

#### OUTPUT:



### 4. Bridge ROS with openCV. Read an image into ROS and rotate the image.

#### CODE:

```
#!/usr/bin/env python3
import rospy # Python library for ROS
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
def callback(data):
    br = CvBridge()
    rospy.loginfo("receiving video frame")
    current_frame = br.imgmsg_to_cv2(data)
    current_frame=cv2.circle(current_frame,(60,60),10,(0,255,255),-1)
    current_frame=cv2.rotate(current_frame,cv2.ROTATE_90_CLOCKWISE)
    cv2.imshow("camera", current_frame)
    cv2.waitKey(0)
def receive_message():
    rospy.init_node('video_sub_py', anonymous=True)
```

```

rospy.Subscriber('video_frames', Image, callback)
rospy.spin()
cv2.destroyAllWindows()
if __name__ == '__main__':
    receive_message()

```

## OUTPUT:



## 5. Read an image into ROS and perform color conversions on an image.

### CODE:

```

#!/usr/bin/env python3
import rospy # Python library for ROS
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
def publish_message():
    pub = rospy.Publisher('video_frames', Image, queue_size=10)
    rospy.init_node('video_pub_py', anonymous=True)
    rate = rospy.Rate(10) # 10hz
    cap = cv2.imread('/home/rae/Downloads/test.png',0)
    br = CvBridge()
    while not rospy.is_shutdown():

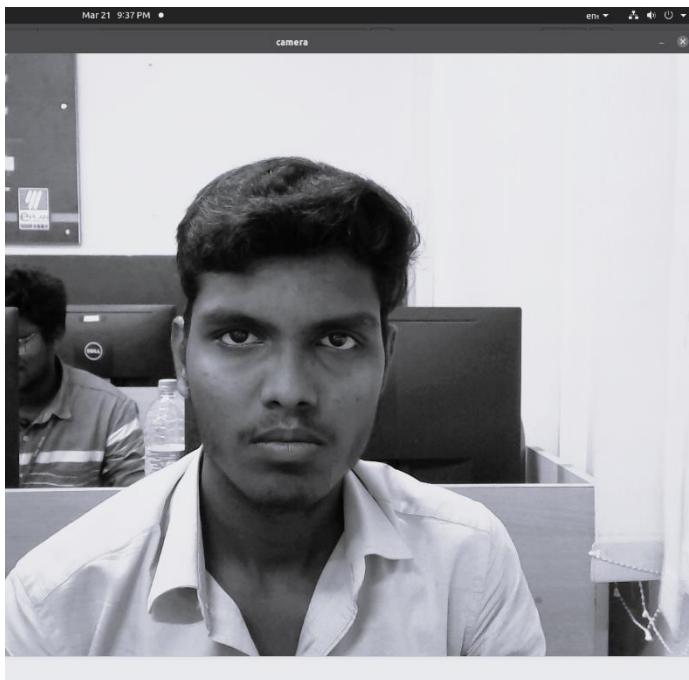
```

```

    if True:
        rospy.loginfo('publishing video frame')
        pub.publish(br.cv2_to_imgmsg(cap))
        rate.sleep()
if __name__ == '__main__':
    try:
        publish_message()
    except rospy.ROSInterruptException:
        pass

```

## OUTPUT:



## 6. Write AI and vision on an image.

## CODE:

```

#!/usr/bin/env python3
import rospy # Python library for ROS
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
def publish_message():
    pub = rospy.Publisher('video_frames', Image, queue_size=10)
    rospy.init_node('video_pub_py', anonymous=True)
    rate = rospy.Rate(10) # 10hz
    cap = cv2.imread('/home/rae/Downloads/black screen.png')
    current_frame=cv2.putText(current_frame,"AI &
VISION",(50,50),cv2.FONT_HERSHEY_SIMPLEX,1,(255,0,255),2,cv2.LINE_AA)

```

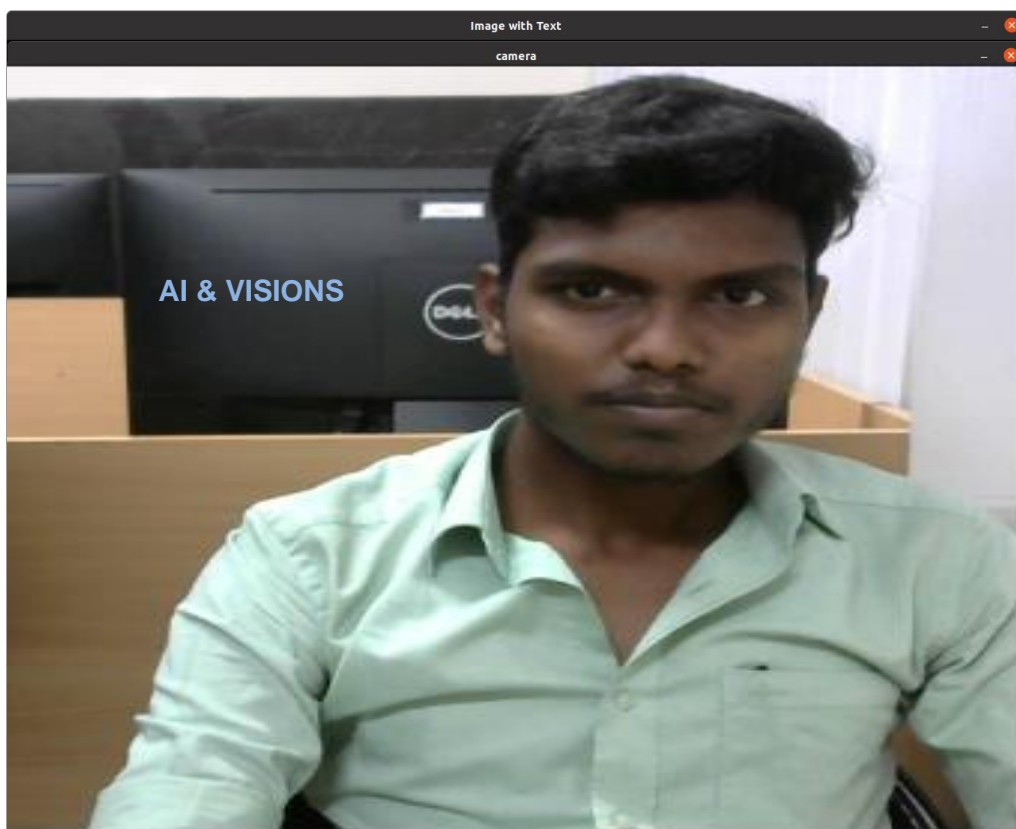


```

br = CvBridge()
while not rospy.is_shutdown():
    if True:
        rospy.loginfo('publishing video frame')
        pub.publish(br.cv2_to_imgmsg(cap))
        rate.sleep()
if __name__ == '__main__':
    try:
        publish_message()
    except rospy.ROSInterruptException:
        pass

```

### OUTPUT:



### 7. Find the difference between the two images.

#### CODE:

```

import rospy # Python library for ROS
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
def publish_message():
    pub = rospy.Publisher('video_frames', Image, queue_size=10)

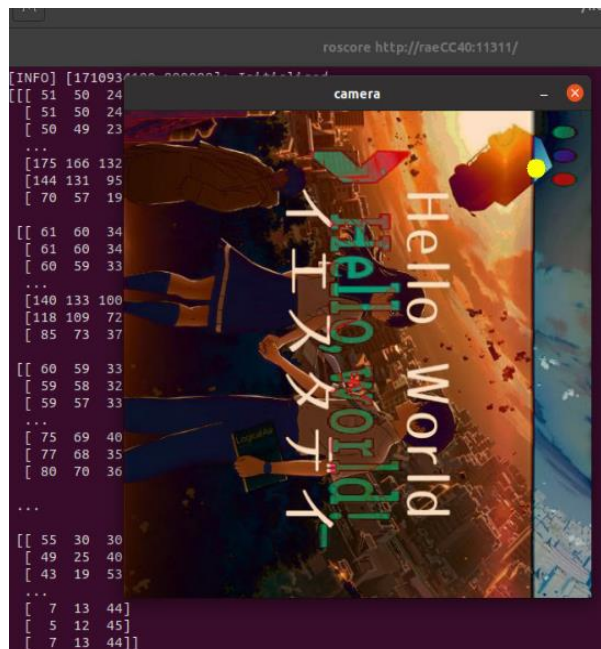
```

```

rospy.init_node('video_pub_py', anonymous=True)
rate = rospy.Rate(10) # 10hz
cap = cv2.imread('/home/rae/Downloads/test.png',0)
br = CvBridge()
while not rospy.is_shutdown():
    rospy.loginfo('publishing video frame')
    pub.publish(br.cv2_to_imgmsg(cap))
    rate.sleep()
if __name__ == '__main__':
    try:
        publish_message()
    except rospy.ROSInterruptException:
        pass

```

## OUTPUT:



## 8. Write a python program in ROS to sort a given set of numbers.

### CODE:

```

import rospy
from std_msgs.msg import Int32MultiArray
def sort_numbers(numbers):
    sorted_numbers = sorted(numbers.data)
    rospy.loginfo("Sorted Numbers:
    {}".format(sorted_numbers)) return
    sorted_numbers
def numbers_callback(data):

```



```

rospy.loginfo("Received Numbers:
{}".format(data.data)) sorted_numbers =
sort_numbers(data)
rospy.signal_shutdown("Numbers sorted successfully. Shutting down...") #
Shutdown after sorting
def numbers_sorter():
    rospy.init_node('numbers_sorter',
    anonymous=True)
    numbers = Int32MultiArray(data=[5, 2, 8, 1, 9]) # Predefined numbers to
    sort sorted_numbers = sort_numbers(numbers)
    rospy.loginfo("Sorted Numbers:
{}".format(sorted_numbers)) rospy.spin()

if __name__ == '__main__':
    try:
        numbers_sorter()
    except rospy.ROSInterruptException:
        pass

```

## OUTPUT:

```

Mar 24 21:01
/home/harsha/catkin_ws/src/number_sorter/number_sorter.launch http://localhost:11311
roscore http://harsha-HP-Pavilion-Gaming-Laptop-15-ec2xxx:11311/
/home/harsha/catkin_ws/src/number_sorter/number_sorter.launch http://localhost:11311
harsha@harsha-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~$ cd ~/catkin_ws
harsha@harsha-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/catkin_ws$ catkin_make
Base path: /home/harsha/catkin_ws
Source space: /home/harsha/catkin_ws/src
Build space: /home/harsha/catkin_ws/build
Devel space: /home/harsha/catkin_ws/devel
Install space: /home/harsha/catkin_ws/install
#### Running command: "make cmake_check_build_system" in "/home/harsha/catkin_ws/build"
####
#### Running command: "make -j12 -l12" in "/home/harsha/catkin_ws/build"
####
harsha@harsha-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/catkin_ws$ source devel/setup.bash
harsha@harsha-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/catkin_ws$ roslaunch number_sorter number_sorter.launch
... logging to /home/harsha/.ros/log/82b9c544-e9f3-11ee-bf06-5903bb6e615e/roslaunch-harsha-HP-Pavilion-Gaming-Laptop-15-ec2xxx-98648.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://harsha-HP-Pavilion-Gaming-Laptop-15-ec2xxx:42557/
SUMMARY
=====
PARAMETERS
 * /roslistro: noetic
 * /rosversion: 1.16.0
NODES
 /
  numbers_sorter (number_sorter/sort_numbers.py)
ROS_MASTER_URI=http://localhost:11311
process[numbers_sorter-1]: started with pid [98654]
[INFO] [1711294278.158760]: Sorted Numbers: [1, 2, 5, 8, 9]
[INFO] [1711294278.159367]: Sorted Numbers: [1, 2, 5, 8, 9]

```

## 9. Stream the video from USB camera in RoS and write your name on the Stream.

### CODE:

```

#!/usr/bin/env python3
import rospy # Python library for ROS
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

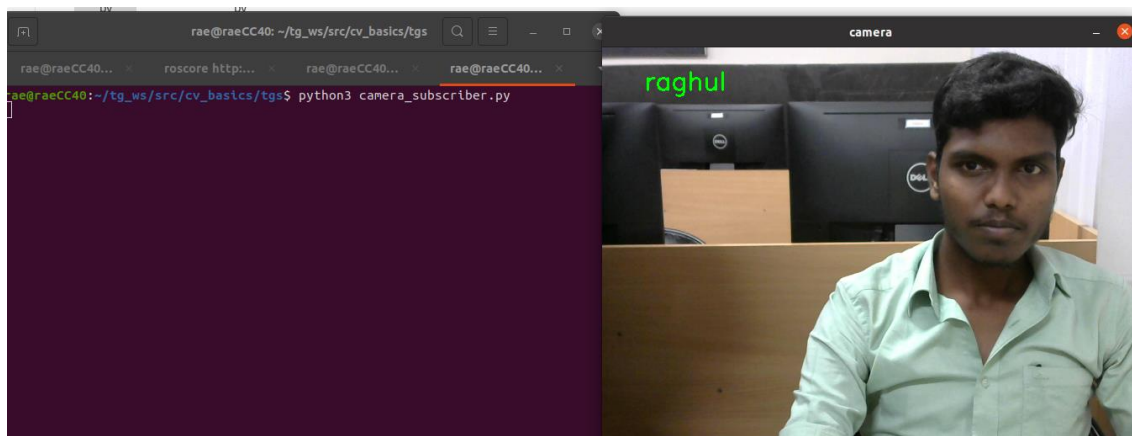
```

```

def callback(data):
    br = CvBridge()
    rospy.loginfo("receiving video frame")
    current_frame = br.imgmsg_to_cv2(data)
    current_frame=cv2.putText(current_frame,"hiiii", (50,50),cv2.FONT_HERSHEY_SIMPLEX,1,(255,0,255),2,cv2.LINE_AA)
    current_frame=cv2.circle(current_frame,(60,60),10,(0,255,255),-1)
    cv2.imshow("camera", current_frame)
    cv2.waitKey(1)
def receive_message():
    rospy.init_node('video_sub_py', anonymous=True)
    rospy.Subscriber('video_frames', Image, callback)
    rospy.spin()
    cv2.destroyAllWindows()
if __name__ == '__main__':
    receive_message()

```

## OUTPUT:



**10. Simulate a world of your own in Gazebo and Rviz and spawn a turtlebot on it. Environment with turtlebot 3 has been setup:**

## SETUP:

Download link:

[https://github.com/SakshayMahna/Robotics-Playground/tree/main/turtlebot3\\_ws](https://github.com/SakshayMahna/Robotics-Playground/tree/main/turtlebot3_ws)

Unzip into home dir.

Open terminal

```
roscore
```

Open another terminal

```
sudo apt-get install ros-noetic-navigation
```

Open another terminal

```
cd turtlebot3_ws/
catkin_make
```

```
source devel/setup.bash
```

## TO RUN TURTLEBOT3:

Terminal 1

```
roscore
```

Terminal 2

```
cd turtlebot3_ws/  
catkin_make  
source devel/setup.bash  
roslaunch ros_world turtlebot3_world.launch
```

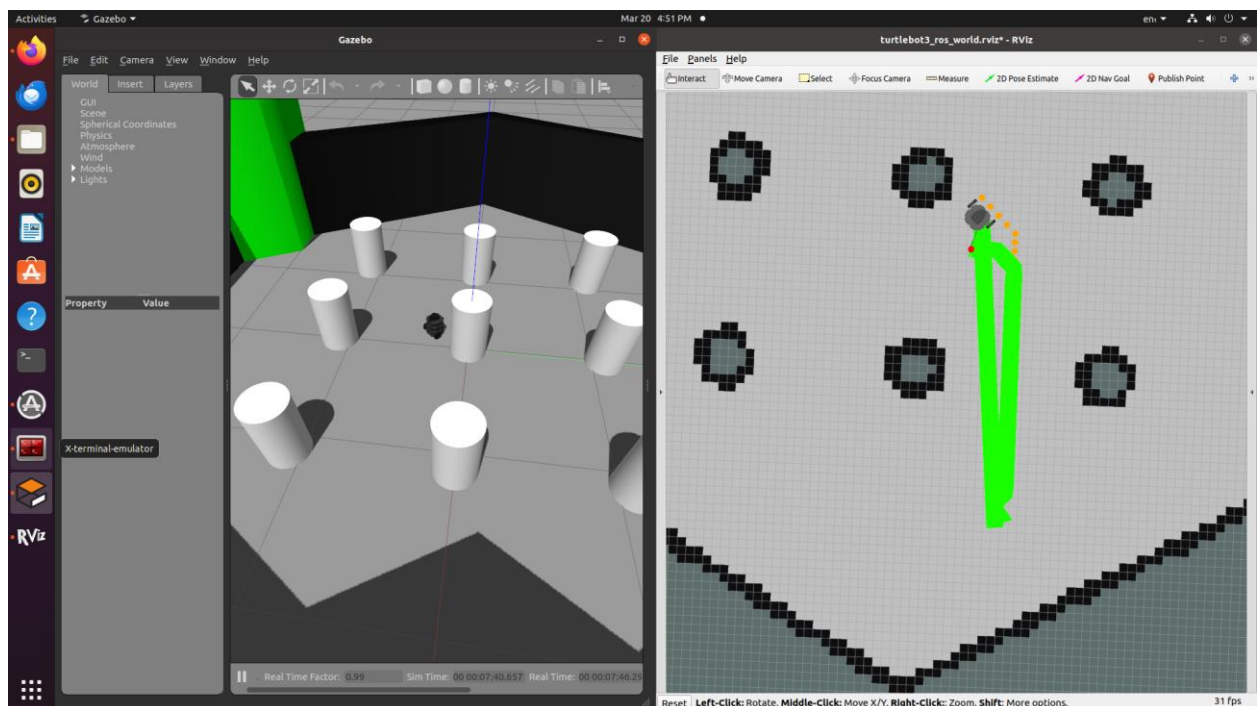
Terminal 3

```
cd turtlebot3_ws/  
catkin_make  
source devel/setup.bash  
roslaunch global_path_planning turtlebot3_ros_world.launch
```

Terminal 4

```
cd turtlebot3_ws/  
catkin_make  
source devel/setup.bash  
roslaunch global_path_planning path_planning_server.py
```

## OUTPUT:



Department of RAE			
Criteria	Excellent (75% - 100%)	Good (50 - 75%)	Poor (<50%)
Preparation (30)			
Performance (30)			
Evaluation (20)			
Report (20)			
Sign:	Total (100)		

**Result:**

Thus, the ROS basics has been successfully implemented.