# ROS Interface for PiRobot

Swagat Kumar

# Objectives

- To be able to tele-operate the RPi Robot using a keyboard on a remote system.

- To stream live video from the picamera onboard robot to the remote system.

- To control robot motion through hand gestures.

- In the process, we will understand how to develop a distributed system for robotic applications using ROS.

# Requirements

- Raspberry Pi 3/4  - Two numbers (one for the robot & one for remote host)
- Pi Robot Kit assembled
- Raspbian Buster OS
- ROS Noetic
- Python 3.x
- Opencv 4.x
- Picamera V2

# Steps for both Robot & Remote Host

These steps are to be executed **on both Robot and Remote Host:**

1. Install ROS Noetic on Raspbian Buster
   https://varhowto.com/install-ros-noetic-raspberry-pi-4/

2. Clone this github repo in your home directory:
   ```
   $ git clone https://github.com/swagatk/Raspi_codes.git
   ```
   It will create a folder called 'Raspi_codes' in your home directory.

3. Create catkin workspace in your home folder:
   ```
   $ mkdir -p ~/catkin_ws/src
   $ cd ~/catkin_ws/
   $ catkin_make
   $ source devel/setup.bash
   ```

4. Create a catkin package 'pirobot' inside your catkin_ws folder:
   ```
   $ cd catkin_make/src/
   $ catkin_create_pkg pirobot rospy roscpp std_msgs sensor_msgs
   $ cd ~/catkin_ws/
   $ catkin_make
   $ source devel/setup.bash
   ```

5. Copy the files from GIT repo to the pirobot package:
   ```
   $ roscd pirobot
   $ cp -r ~/Raspi_codes/ROS/pirobot/scripts/ ./
   $ cp -r ~/Raspi_codes/ROS/pirobot/launch/ ./
   $ cd ~/catkin_ws/
   $ catkin_make
   $ source ./devel/setup.bash
   ```
   This will make sure that the new files are accessible through ROS commands.

# Setup on Remote Host

1. On the Remote host, enter the following lines inside the ~/.bashrc file

   ```
   export ROS_MASTER_URI=http://192.168.1.144:11311
   export ROS_IP=192.168.1.144
   ```

   Replace the IP with your own IP address. Run the command ‘`ifconfig`’ to find the IP address of your raspberry pi.
2. Open a new terminal and run the following commands:

   ```
   $ bash
   $ roscore
   ```

   Check that the ROS_MASTER_URI address should as set above.

# Setup on Remote Host contd ...

3. Open another terminal and run the following command:

```
$ source ~/catkin_ws/devel/setup.bash
$ roslaunch pirobot remote.launch
```

It is possible to run individual nodes by running following commands separately one on each terminal:

```
$ rosrun pirobot kb_teleop.py __name:=’kb_teleop’
```

It will run a publisher to catch the key press on the remote host. You can use the arrow keys to move the robot left, right, forward and backward.

# Setup on Robot

1. On the robot system, please enter the following lines to `~/.bashrc` file:

   ```
   export ROS_MASTER_URI=http://<previous_master_ip>:11311
   export ROS_IP=192.168.1.145 (its own ip)
   ```

2. Open a terminal and run the following commands:

   ```
   $ bash
   $ source ~/catkin_ws/devel/setup.bash
   $ rosrun pirobot robot_motion.py __name:=’robot_motion’
   ```

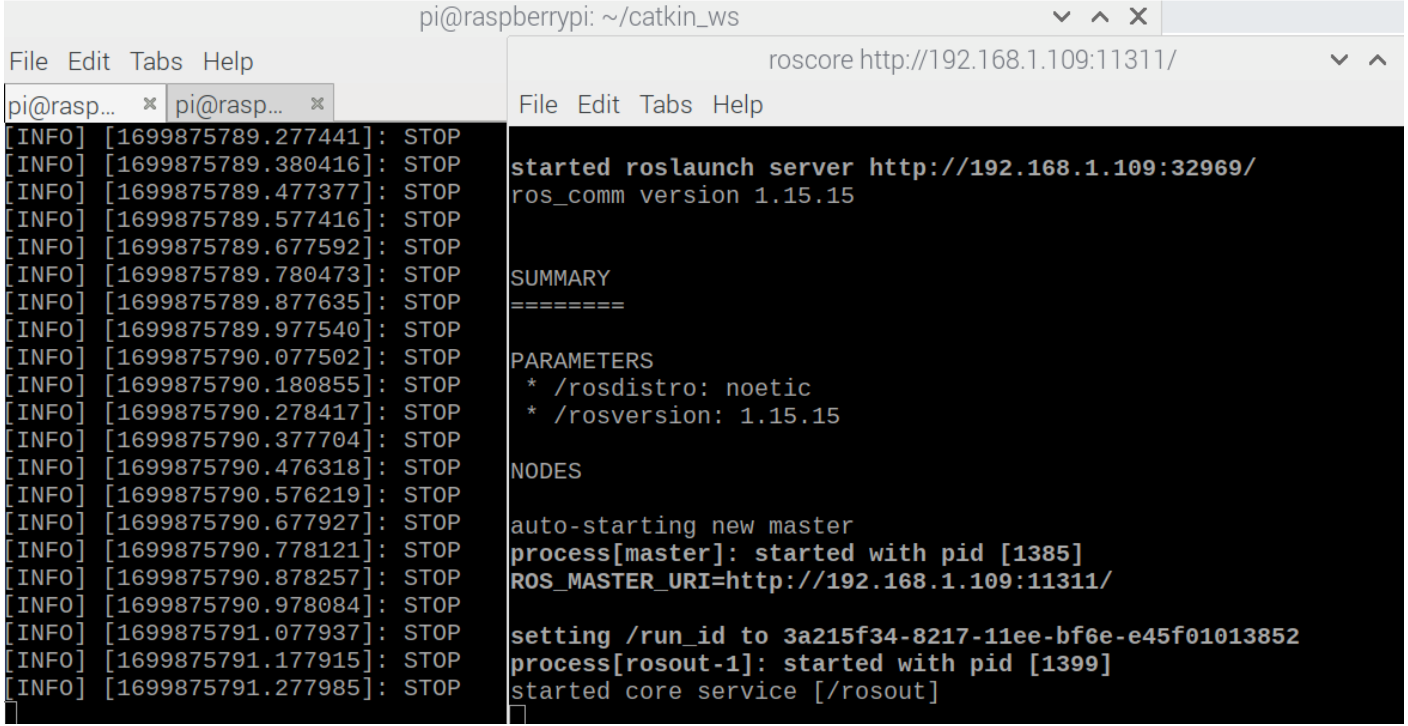
   It runs a subscriber to communicate with the publisher running on the remote host.

3. Now you can press the arrow keys on remote host to control the motion of the pi robot. Make sure that both robot and remote host are connected to the same network. Also make sure that you don’t get any error message while executing above steps.

4. It is also possible to execute the node by running the following command:

   ```
   $ roslaunch pirobot pirobot.launch
   ```

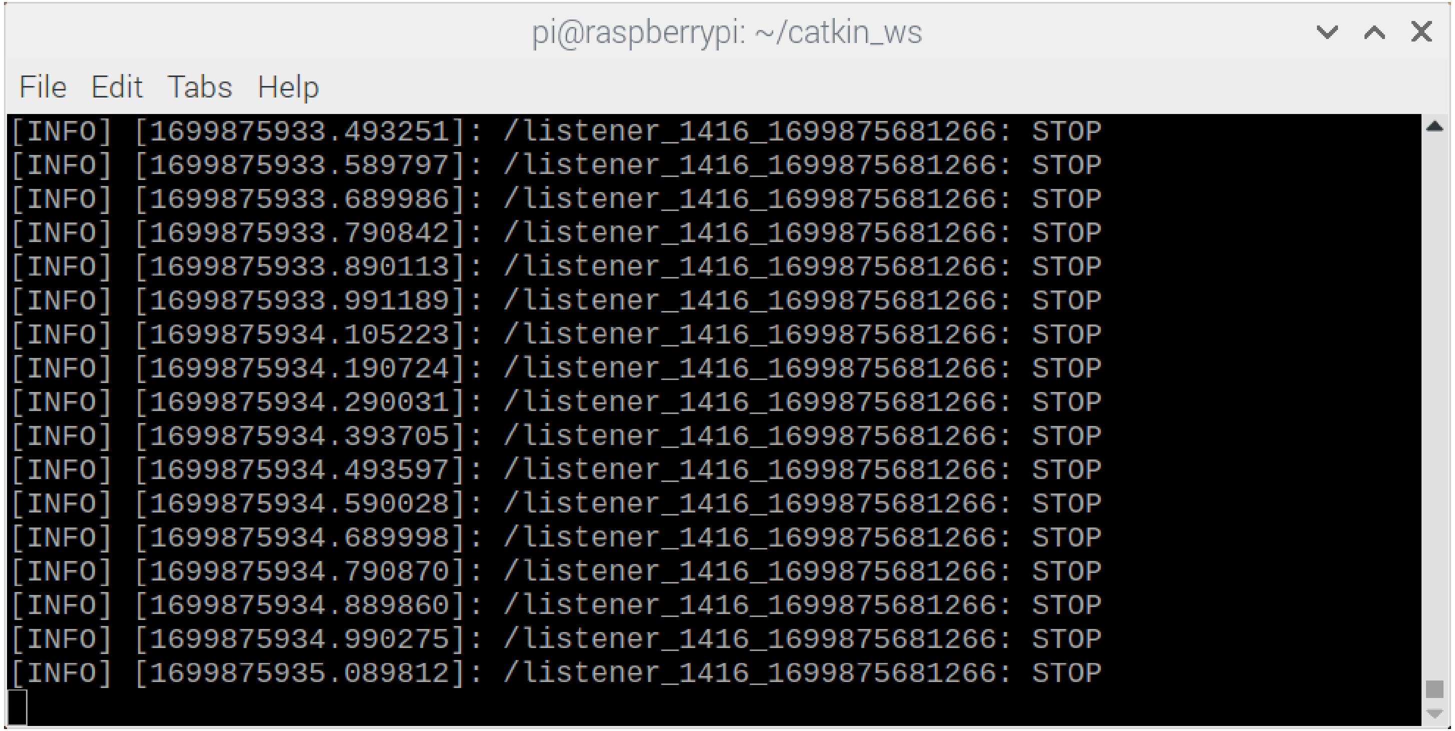
# Remote System



```
[INFO] [1699875789.277441]: STOP
[INFO] [1699875789.380416]: STOP
[INFO] [1699875789.477377]: STOP
[INFO] [1699875789.577416]: STOP
[INFO] [1699875789.677592]: STOP
[INFO] [1699875789.780473]: STOP
[INFO] [1699875789.877635]: STOP
[INFO] [1699875789.977540]: STOP
[INFO] [1699875790.077502]: STOP
[INFO] [1699875790.180855]: STOP
[INFO] [1699875790.278417]: STOP
[INFO] [1699875790.377704]: STOP
[INFO] [1699875790.476318]: STOP
[INFO] [1699875790.576219]: STOP
[INFO] [1699875790.677927]: STOP
[INFO] [1699875790.778121]: STOP
[INFO] [1699875790.878257]: STOP
[INFO] [1699875790.978084]: STOP
[INFO] [1699875791.077937]: STOP
[INFO] [1699875791.177915]: STOP
[INFO] [1699875791.277985]: STOP
```

```

started roslaunch server http://192.168.1.109:32969/
ros_comm version 1.15.15


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [1385]
ROS_MASTER_URI=http://192.168.1.109:11311/

setting /run_id to 3a215f34-8217-11ee-bf6e-e45f01013852
process[rosout-1]: started with pid [1399]
started core service [/rosout]
```

# PiRobot Subscriber



```
[INFO] [1699875933.493251]: /listener_1416_1699875681266: STOP
[INFO] [1699875933.589797]: /listener_1416_1699875681266: STOP
[INFO] [1699875933.689986]: /listener_1416_1699875681266: STOP
[INFO] [1699875933.790842]: /listener_1416_1699875681266: STOP
[INFO] [1699875933.890113]: /listener_1416_1699875681266: STOP
[INFO] [1699875933.991189]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.105223]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.190724]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.290031]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.393705]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.493597]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.590028]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.689998]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.790870]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.889860]: /listener_1416_1699875681266: STOP
[INFO] [1699875934.990275]: /listener_1416_1699875681266: STOP
[INFO] [1699875935.089812]: /listener_1416_1699875681266: STOP
```

# ROS Image Publisher & Subscriber with Picamera

# Install Opencv 4.5.x on Raspberry Pi

- **For RPi 4**, follow instruction available at this site: https://qengineering.eu/install-opencv-4.5-on-raspberry-pi-4.html

- **For RPI 3**, we need to install the available debian package (this also works for Pi 4)

```
$ sudo apt-get install python-opencv python3-opencv
```

- Make sure that the cv2 is accessible from python console

```
$ python3
Python 3.7.3 (default, Oct 31 2022, 14:04:00)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.5.5'
>>>
```

# Install ‘sensor_msg’s into the existing ROS installation

This step is required if only ‘ros_comm’ package is installed previously. ‘`sensor_msgs`’ are required for broadcasting image type messages over the ROS network. If ROS-Desktop version is installed, this step can be skipped.

```
$ cd ~/ros_catkin_ws/

$ rosinstall_generator  sensor_msgs --rosdistro noetic --deps --wet-only --tar >
noetic-sensor_msgs-wet.rosinstall

$ wstool merge noetic-sensor_msgs-wet.rosinstall -t src

$ wstool update -t src

$ rosdep install -y --from-paths src --ignore-src --rosdistro noetic -r --
os=debian:buster

$ sudo src/catkin/bin/catkin_make_isolated --install -DCMAKE_BUILD_TYPE=Release
--install-space /opt/ros/noetic -j1 -DPYTHON_EXECUTABLE=/usr/bin/python3
```

# Enable/Install PiCamera

- Follow the instructions for **Raspbian Buster** available the following link: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera

- Write a simple python script to test if the camera module is working:

```python
from picamera import PiCamera
from time import sleep

camera = PiCamera()
camera.start_preview()
sleep(5)
camera.stop_preview()
```

# Run publisher & Subscriber

- Execute the following command in one terminal for publisher:

```
$ roslaunch pirobot pirobot.launch
```

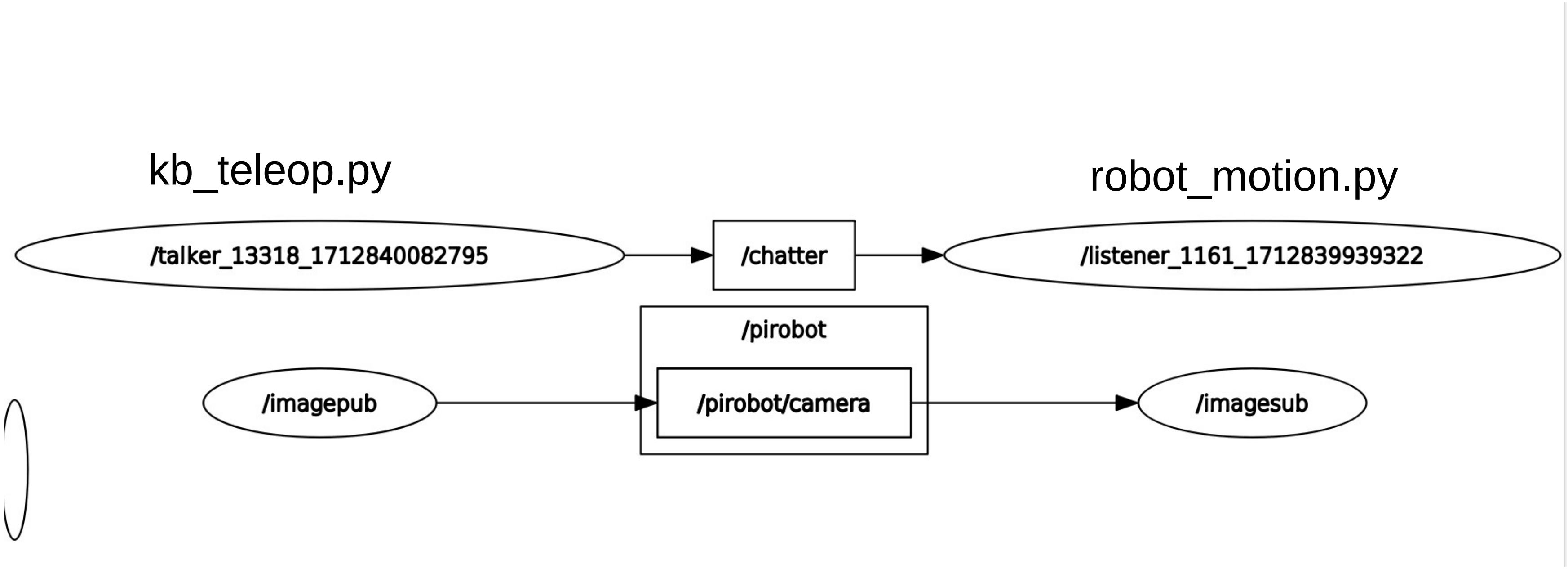
- Execute the following command in the second terminal for subscriber:

```
$ rosrun pirobot image_subscriber.py
```

- Explore rosnodes & rostopics:

```
$ rosnode list
/imagepub
/imagesub
/rosout

$ rostopic list
/picam/camera
/rosout
/rosout_agg
```

# ROS Graph



kb_teleop.py

robot_motion.py

# Creating Obstacle Avoidance node

- Relevant files:
  - `~/catkin_ws/pirobot/scripts/ao_node.py`
  - `~/catkin_ws/pirobot/scripts/avoid_obstacle.py`
  - `~/catkin_ws/pirobot/scripts/motor_control.py`

On the robot system, first compile the `pirobot` package and then run the following commands to run the obstacle avoidance node:

```
$ cd ~/catkin_ws
$ source devel/setup.bash
$ rosrun pirobot ao_node.py
```

# Controlling Robot through Hand Gestures

- Task is to control the pirobot through hand pose recognition (not really gestures!!)
- The user makes gestures in front of a camera mounted on the remote system (such as laptop).
- The hand pose is recognized and relevant commands are published on the ‘chatter’ topic.
- The robot_motion node running on RPi robot subscribes to this ‘chatter’ topic receive the respective commands which are then translated to the actual motor actuation.
- The code for gesture control was contributed by Tomasz Szmidt (EHU batch 2024)

# Gesture Control

Relevant files of interest are as follows:

  - `~/catkin_ws/src/pirobot/scripts/hand_teleop.py`
  - `~/catkin_ws/src/pirobot/scripts/recognise_direction_of_finger.py`
  - `~/catkin_ws/src/pirobot/scripts/module.py`
  - `~/catkin_ws/src/pirobot/scripts/robot_motion.py`

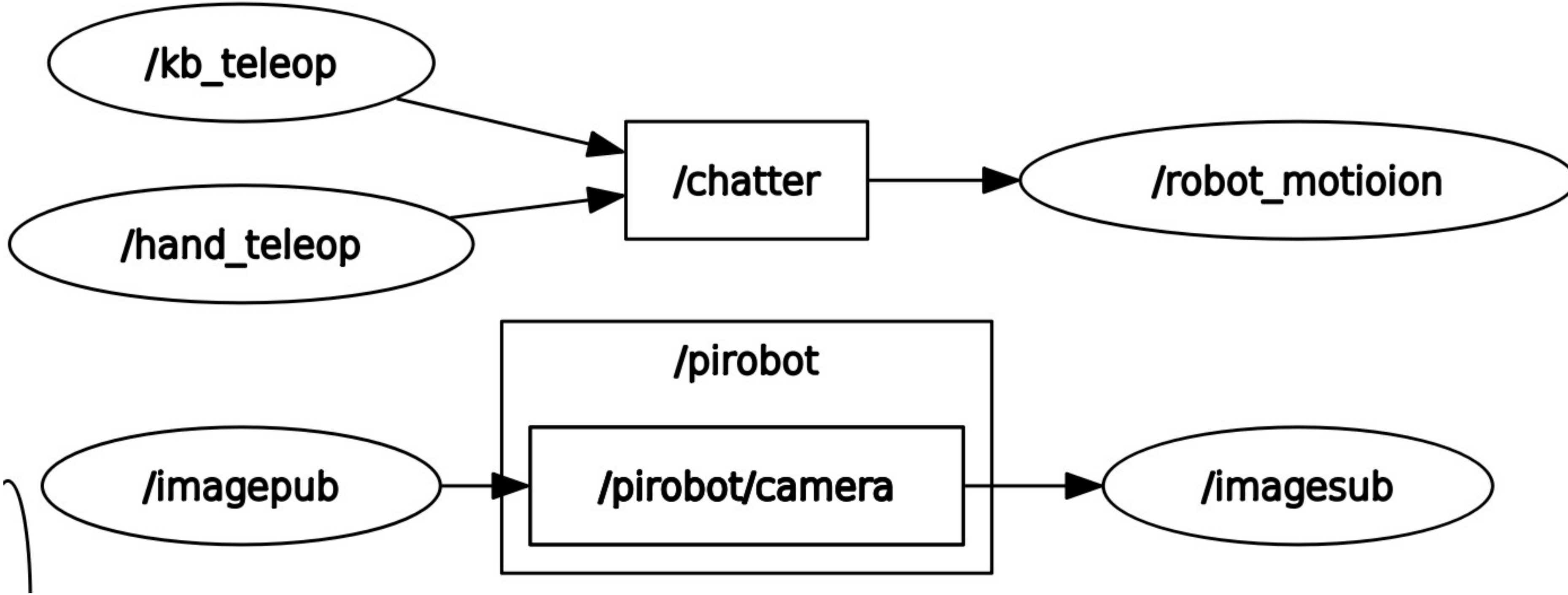
- On the robot system, run the ‘robot_motion’ node:

  ```
  $ rosrun pirobot robot_motion.py
  ```

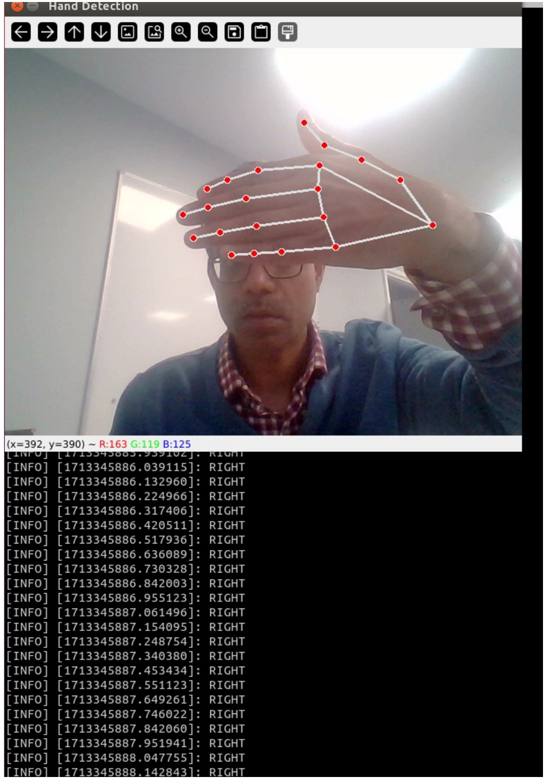
- On the remote system with on-board camera, run the following code:

  ```
  $ rosrun pirobot hand_teleop.py
  ```

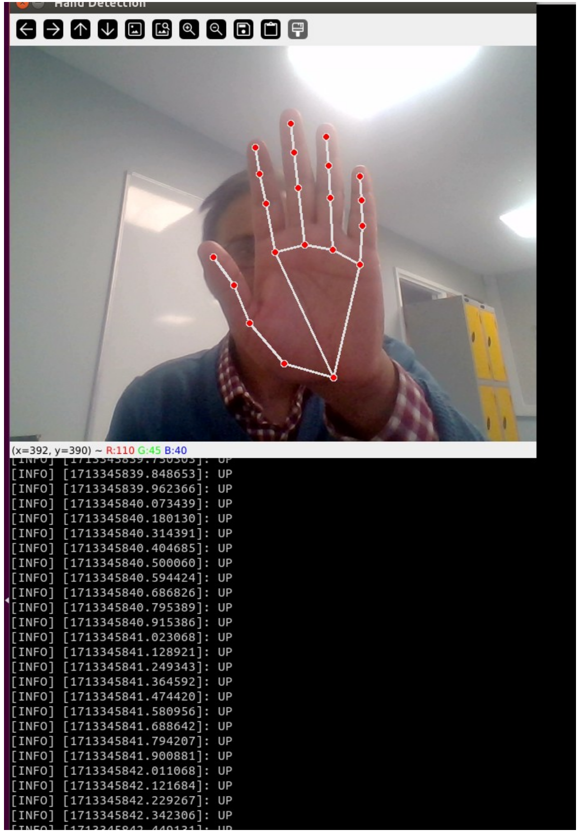
# ROS Graph


/kb_teleop
/hand_teleop
/chatter
/robot_motioion
/pirobot
/imagepub
/pirobot/camera
/imagesub

# Gesture Control Output



```
[INFO] [1713345885.939102]: RIGHT
[INFO] [1713345886.039115]: RIGHT
[INFO] [1713345886.132960]: RIGHT
[INFO] [1713345886.224966]: RIGHT
[INFO] [1713345886.317406]: RIGHT
[INFO] [1713345886.420511]: RIGHT
[INFO] [1713345886.517936]: RIGHT
[INFO] [1713345886.636089]: RIGHT
[INFO] [1713345886.730328]: RIGHT
[INFO] [1713345886.842003]: RIGHT
[INFO] [1713345886.955123]: RIGHT
[INFO] [1713345887.061496]: RIGHT
[INFO] [1713345887.154095]: RIGHT
[INFO] [1713345887.248754]: RIGHT
[INFO] [1713345887.340380]: RIGHT
[INFO] [1713345887.453434]: RIGHT
[INFO] [1713345887.551123]: RIGHT
[INFO] [1713345887.649261]: RIGHT
[INFO] [1713345887.746022]: RIGHT
[INFO] [1713345887.842060]: RIGHT
[INFO] [1713345887.951941]: RIGHT
[INFO] [1713345888.047755]: RIGHT
[INFO] [1713345888.142843]: RIGHT
```



```
[INFO] [1713345839.730303]: UP
[INFO] [1713345839.848653]: UP
[INFO] [1713345839.962366]: UP
[INFO] [1713345840.073439]: UP
[INFO] [1713345840.180130]: UP
[INFO] [1713345840.314391]: UP
[INFO] [1713345840.404685]: UP
[INFO] [1713345840.500060]: UP
[INFO] [1713345840.594424]: UP
[INFO] [1713345840.686826]: UP
[INFO] [1713345840.795389]: UP
[INFO] [1713345840.915386]: UP
[INFO] [1713345841.023068]: UP
[INFO] [1713345841.128921]: UP
[INFO] [1713345841.249343]: UP
[INFO] [1713345841.364592]: UP
[INFO] [1713345841.474420]: UP
[INFO] [1713345841.580956]: UP
[INFO] [1713345841.688642]: UP
[INFO] [1713345841.794207]: UP
[INFO] [1713345841.900881]: UP
[INFO] [1713345842.011068]: UP
[INFO] [1713345842.121684]: UP
[INFO] [1713345842.229267]: UP
[INFO] [1713345842.342306]: UP
[INFO] [1713345842.449131]: UP
```

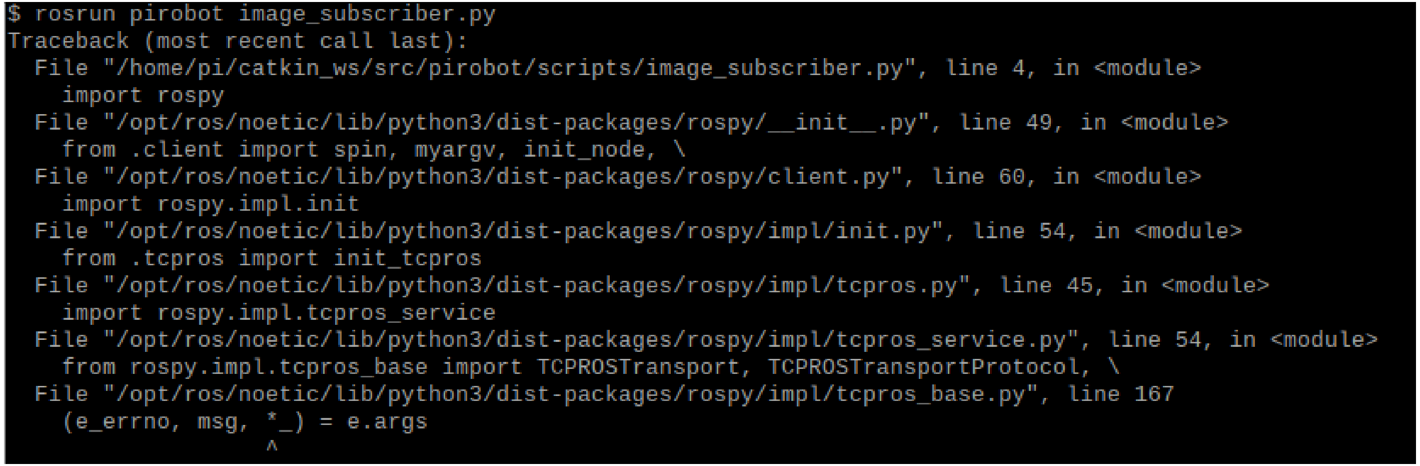
# Errors Debugging

If you get error with OpenCV 4.5.5 and Python3, you may edit the corresponding file and removing the last argument in that line as well.

```
$ sudoedit /opt/ros/noetic/lib/python3/dist-packages/rospy/impl/tcpros_base.py
```

Change the line no. 167 to the following:

```
(e_errorno, msg) = e.args
```



```
$ rosrun pirobot image_subscriber.py
Traceback (most recent call last):
  File "/home/pi/catkin_ws/src/pirobot/scripts/image_subscriber.py", line 4, in <module>
    import rospy
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/__init__.py", line 49, in <module>
    from .client import spin, myargv, init_node, \
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/client.py", line 60, in <module>
    import rospy.impl.init
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/impl/init.py", line 54, in <module>
    from .tcpros import init_tcpros
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/impl/tcpros.py", line 45, in <module>
    import rospy.impl.tcpros_service
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/impl/tcpros_service.py", line 54, in <module>
    from rospy.impl.tcpros_base import TCPROSTransport, TCPROSTransportProtocol, \
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/impl/tcpros_base.py", line 167
    (e_errno, msg, *_) = e.args
                   ^
```

# Installing Noetic-Desktop on RPi 4

- Instruction is available here: https://www.hackster.io/shahizat/lidar-integration-with-ros-noetic-on-raspberry-pi-os-8ea140
- This source is also useful: https://wiki.ros.org/noetic/Installation/Source