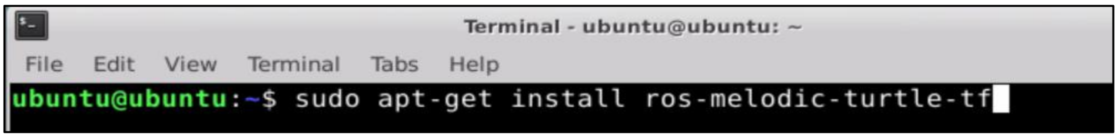
Lesson 13 The Programming Realization of TF Coordinates Broadcasting and Listening

1. Coordinates Transformation

Before programming, the coordinates transformation of robot needs to be learned about first. Here takes running TurtleSim project as an example and the operation steps are as follow:

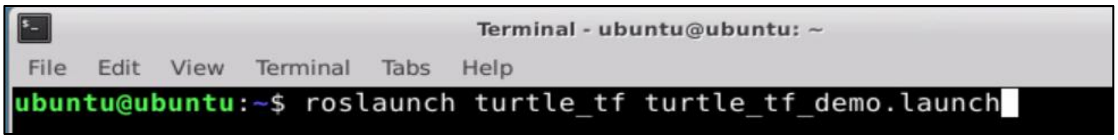
1) Enter "sudo apt-get install ros-melodic-turtle-tf" command to install the package.



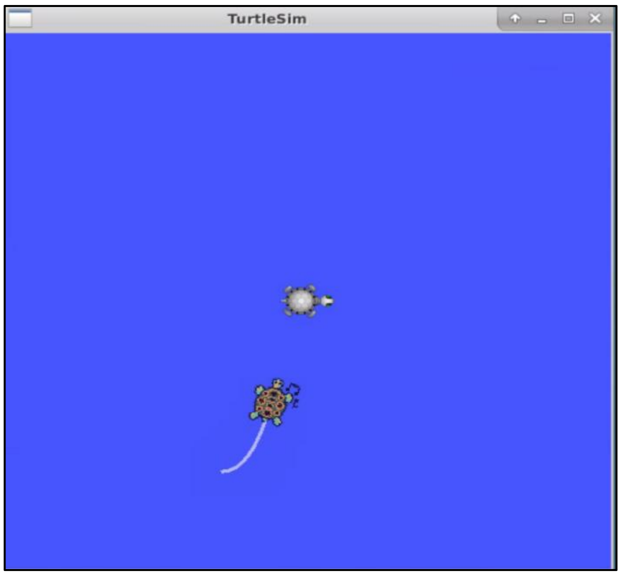
```
ubuntu@ubuntu:~$ sudo apt-get install ros-melodic-turtle-tf
```

Among them, "melodic" corresponds to ROS version.

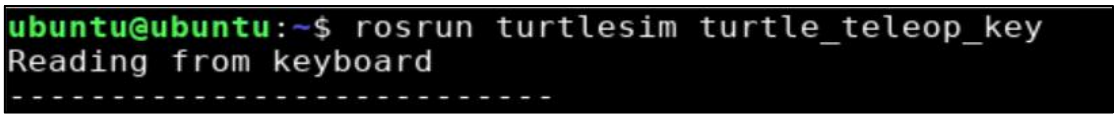
2) Enter "roslaunch turtle_tf turtle_tf_demo.launch" command to run launch file.



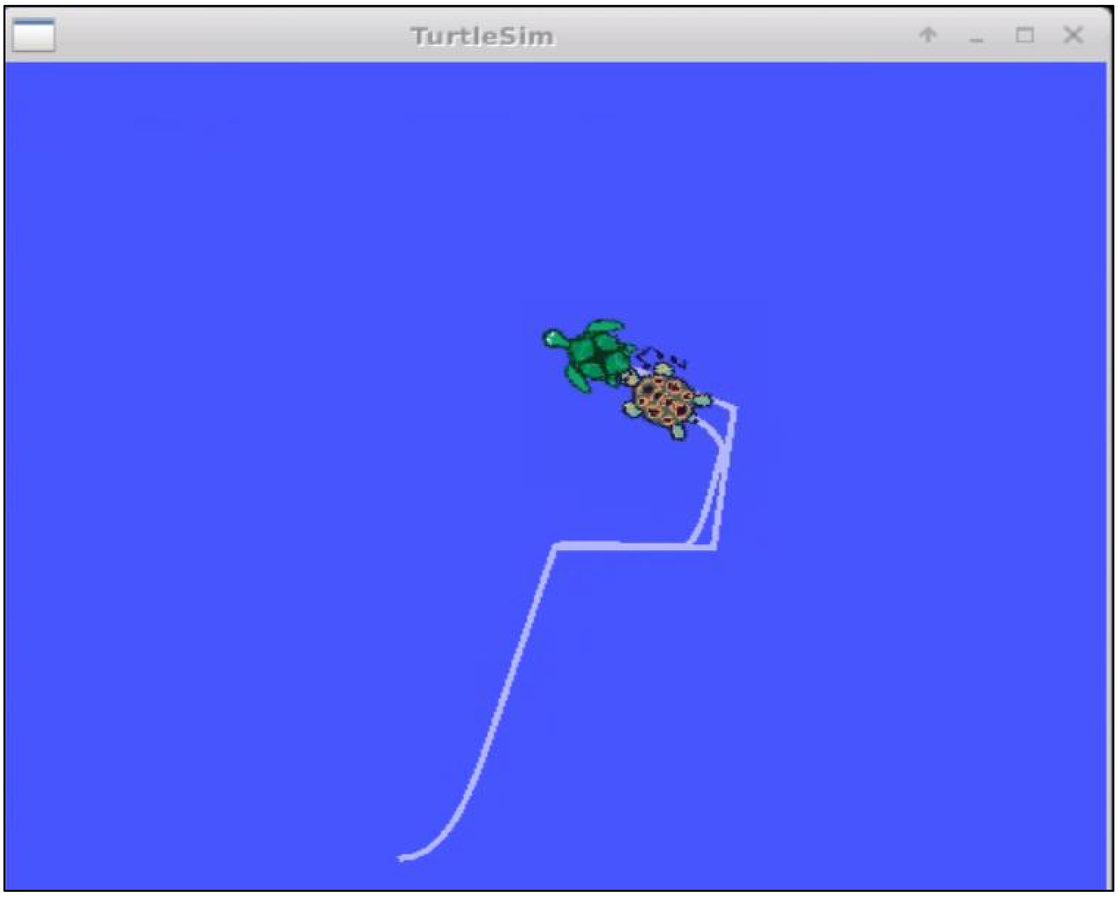
```
ubuntu@ubuntu:~$ roslaunch turtle_tf turtle_tf_demo.launch
```



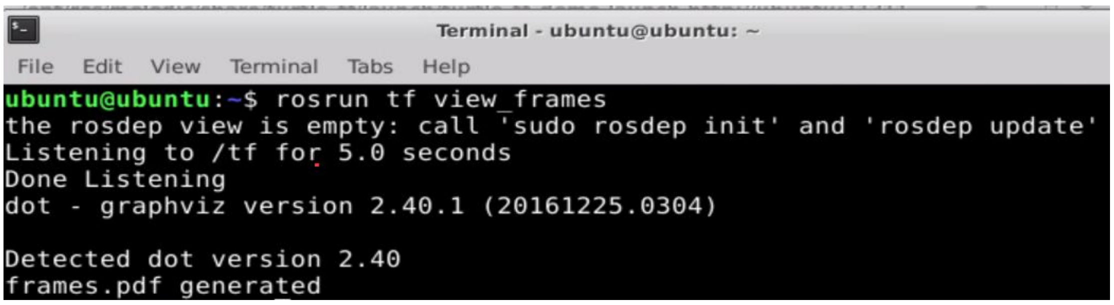
3) Enter “rosrun turtlesim turtle_teleop_key” command to run turtle keyboard control node.



```
ubuntu@ubuntu:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
---------------------------
```

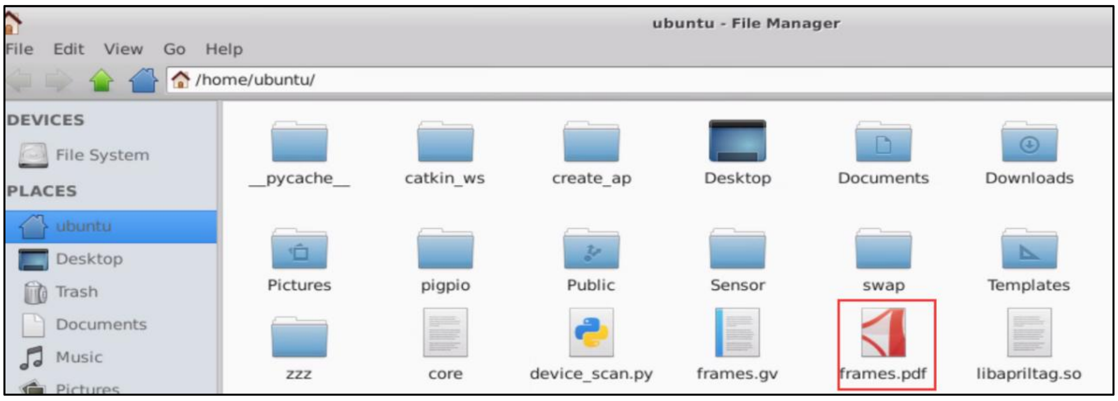


4) Enter “rosrun tf view_frames” command to view the frame.

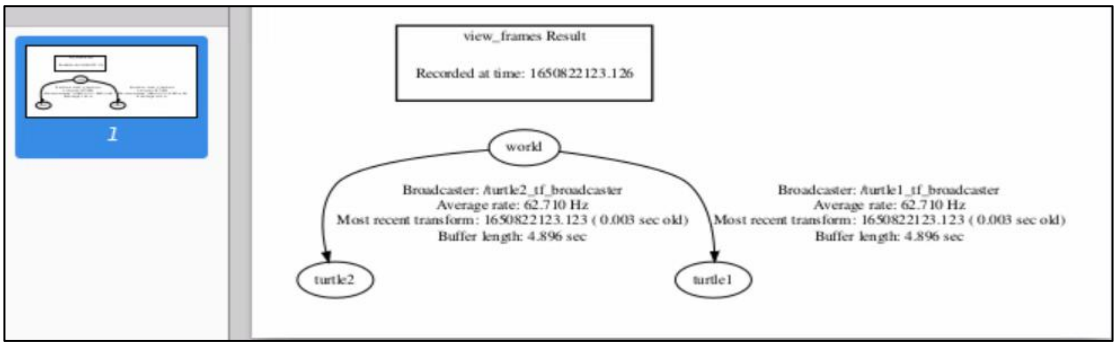

Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~$ rosrun tf view_frames
the rosdep view is empty: call 'sudo rosdep init' and 'rosdep update'
Listening to /tf for 5.0 seconds
Done Listening
dot - graphviz version 2.40.1 (20161225.0304)

Detected dot version 2.40
frames.pdf generated
```

5) Find the file “frames.pdf” under the main directory as the figure shown below:



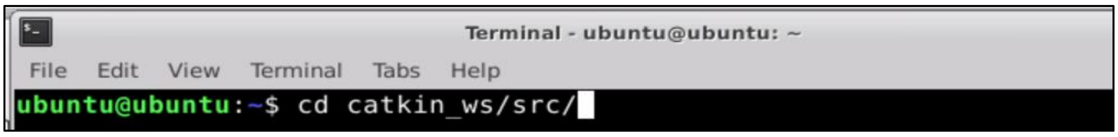
6) Open the file "frames.pdf", and then the position relationship between TF coordinates in current system can be viewed, as the figure shown below:


view_frames Result
Recorded at time: 1650822123.126
world
Broadcaster: /turtle2_tf_broadcaster
Average rate: 62.710 Hz
Most recent transform: 1650822123.123 (0.003 sec old)
Buffer length: 4.896 sec
Broadcaster: /turtle1_tf_broadcaster
Average rate: 62.710 Hz
Most recent transform: 1650822123.123 (0.003 sec old)
Buffer length: 4.896 sec
turtle2
turtle1
1

2. Create Package

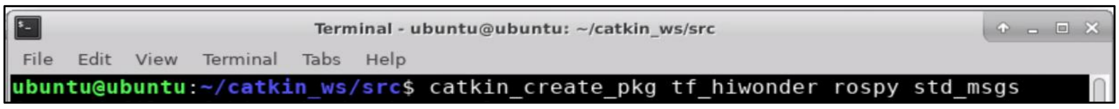
The following operation are going to create the package:

1) Enter "cd catkin_ws/src/" command and press "Eenter" to come to the workspace.


Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~$ cd catkin_ws/src/
```

2) Enter "catkin_create_pkg tf_hiwonder rospy std_msgs" command and press "Enter" to create package.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src
File Edit View Terminal Tabs Help

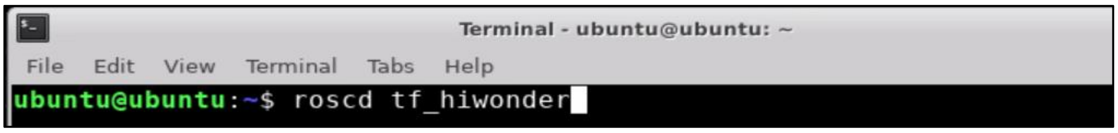
```
ubuntu@ubuntu:~/catkin_ws/src$ catkin_create_pkg tf_hiwonder rospy std_msgs
```

3. Programming Method

3.1 Write Broadcasting and Listening Programs

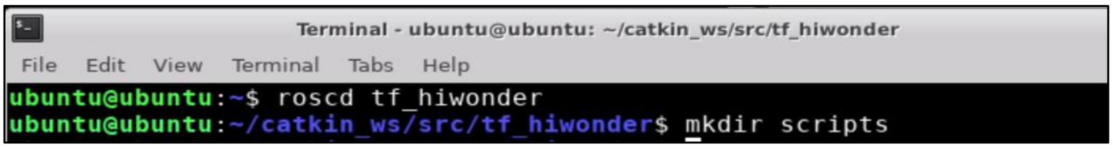
1) Open the terminal.

2) Enter "roscd tf_hiwonder" command to enter the package directory and press "Enter".


Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

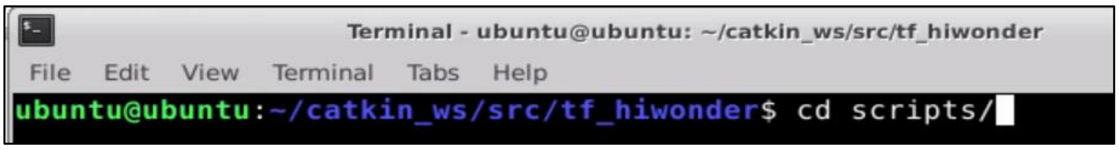
```
ubuntu@ubuntu:~$ roscd tf_hiwonder
```

3) Enter "mkdir scripts" command and press "Enter" to create a new folder "scripts" where Python scripts are stored.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/tf_hiwonder
File Edit View Terminal Tabs Help

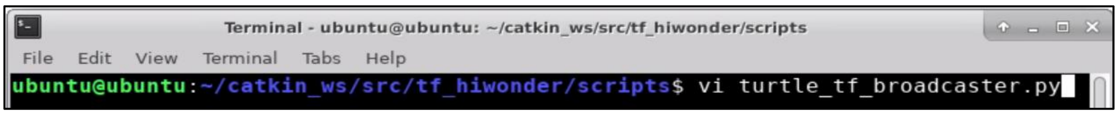
```
ubuntu@ubuntu:~$ roscd tf_hiwonder
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder$ mkdir scripts
```

4) Enter "cd scripts/" command and press "Enter" to enter the folder "scripts" where Python scripts are stored.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/tf_hiwonder
File Edit View Terminal Tabs Help

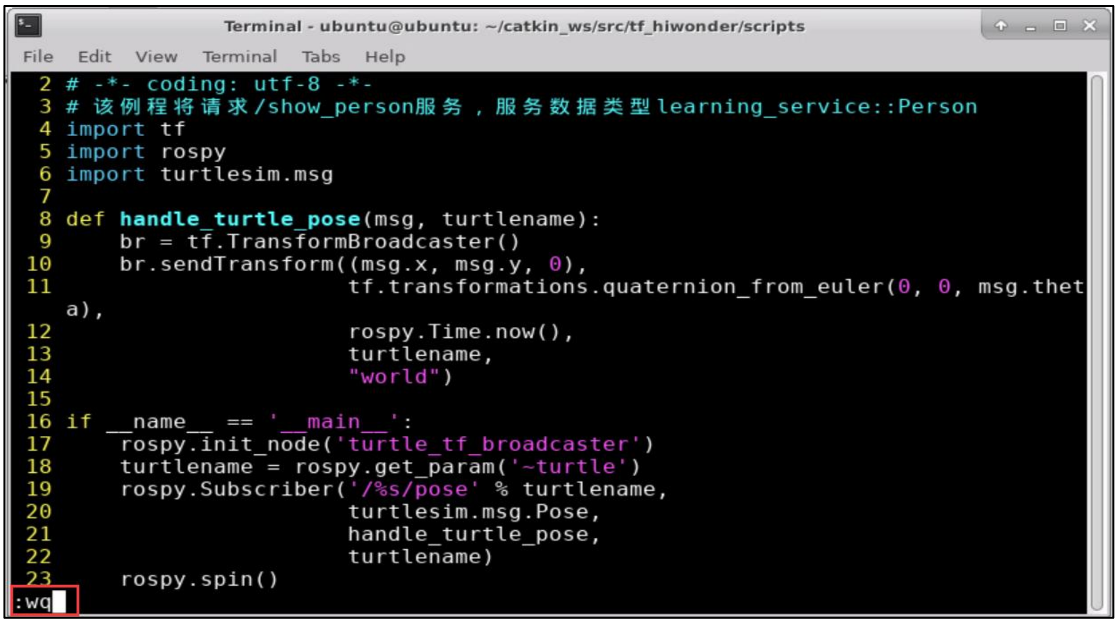
```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder$ cd scripts/
```

5) Enter "vi turtle_tf_broadcaster.py" command to edit program and copy the following program. If want to modify, you can press "i". After modifying, press "Esc" and enter ":wq" to save and exit.


Terminal - ubuntu@ubuntu: ~/catkin_ws/src/tf_hiwonder/scripts
File Edit View Terminal Tabs Help

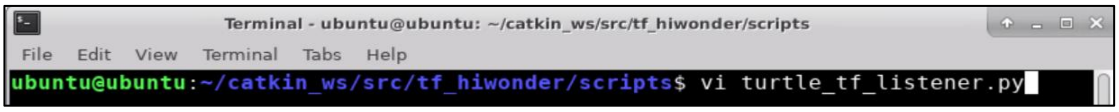
```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder/scripts$ vi turtle_tf_broadcaster.py
```

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# This routine will request /show_person service and the service data type is learning_service::Person
import tf
import rospy
import turtlesim.msg


def handle_turtle_pose(msg, turtlename):
    br = tf.TransformBroadcaster()
    br.sendTransform((msg.x, msg.y, 0),
                     tf.transformations.quaternion_from_euler(0, 0, msg.theta),
                     rospy.Time.now(),
                     turtlename,
                     "world")


if __name__ == '__main__':
    rospy.init_node('turtle_tf_broadcaster')
    turtlename = rospy.get_param('~turtle')
    rospy.Subscriber('/%s/pose' % turtlename,
                     turtlesim.msg.Pose,
                     handle_turtle_pose,
                    turtlename)

        rospy.spin()
```



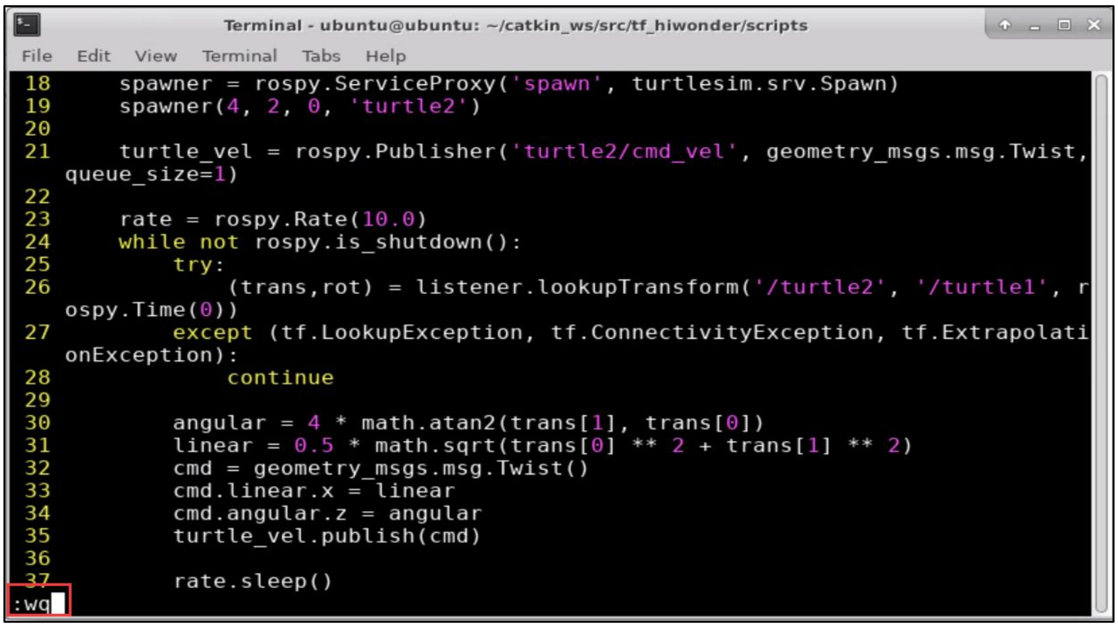
```
# -*- coding: utf-8 -*-
# 该例程将请求/show_person服务，服务数据类型learning_service::Person
import tf
import rospy
import turtlesim.msg

def handle_turtle_pose(msg, turtlename):
    br = tf.TransformBroadcaster()
    br.sendTransform((msg.x, msg.y, 0),
                     tf.transformations.quaternion_from_euler(0, 0, msg.theta),
                     rospy.Time.now(),
                     turtlename,
                     "world")

if __name__ == '__main__':
    rospy.init_node('turtle_tf_broadcaster')
    turtlename = rospy.get_param('~turtle')
    rospy.Subscriber('/%s/pose' % turtlename,
                     turtlesim.msg.Pose,
                     handle_turtle_pose,
                     turtlename)
    rospy.spin()
:wq
```

7) Enter "vi turtle_tf_listener.py" command to edit program and copy the following program. If need to modify, you can press "i". After modifying, press "i" and enter ":wq" to save and exit.



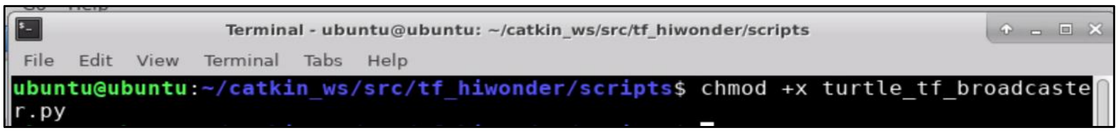
```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder/scripts$ vi turtle_tf_listener.py
```

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-

# This routine will request /show_person service and the service data type is learning_service::Person

import roslib

import rospy

import math

import tf
import geometry_msgs.msg

import turtlesim.srv



if __name__ == '__main__':

    rospy.init_node('turtle_tf_listener')



    listener = tf.TransformListener()



    rospy.wait_for_service('spawn')

    spawner = rospy.ServiceProxy('spawn', turtlesim.srv.Spawn)

    spawner(4, 2, 0, 'turtle2')



    turtle_vel = rospy.Publisher('turtle2/cmd_vel', geometry_msgs.msg.Twist,queue_size=1)



    rate = rospy.Rate(10.0)

    while not rospy.is_shutdown():

        try:

            (trans,rot) = listener.lookupTransform('/turtle2', '/turtle1', rospy.Time(0))

        except (tf.LookupException, tf.ConnectivityException, tf.ExtrapolationException):

            continue



        angular = 4 * math.atan2(trans[1], trans[0])
        linear = 0.5 * math.sqrt(trans[0] ** 2 + trans[1] ** 2)

        cmd = geometry_msgs.msg.Twist()

        cmd.linear.x = linear

        cmd.angular.z = angular

        turtle_vel.publish(cmd)



        rate.sleep()
```

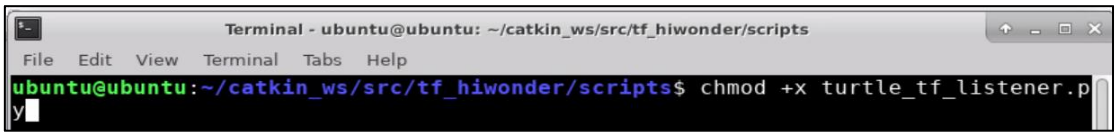


```
    spawner = rospy.ServiceProxy('spawn', turtlesim.srv.Spawn)
    spawner(4, 2, 0, 'turtle2')

    turtle_vel = rospy.Publisher('turtle2/cmd_vel', geometry_msgs.msg.Twist,
queue_size=1)

    rate = rospy.Rate(10.0)
    while not rospy.is_shutdown():
        try:
            (trans,rot) = listener.lookupTransform('/turtle2', '/turtle1', r
ospy.Time(0))
        except (tf.LookupException, tf.ConnectivityException, tf.Extrapolati
onException):
            continue

        angular = 4 * math.atan2(trans[1], trans[0])
        linear = 0.5 * math.sqrt(trans[0] ** 2 + trans[1] ** 2)
        cmd = geometry_msgs.msg.Twist()
        cmd.linear.x = linear
        cmd.angular.z = angular
        turtle_vel.publish(cmd)

        rate.sleep()
:wq
```

8) Enter "chmod +x turtle_tf_broadcaster.py" and "chmod +x turtle_tf_listener.py" command, and then press "Enter" to give the executable permission to the files.



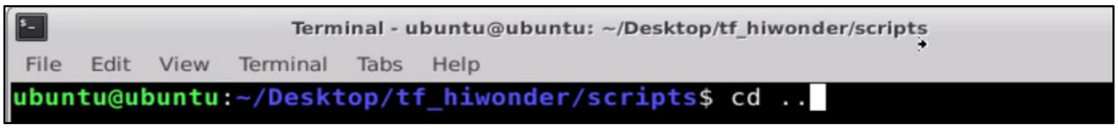
```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder/scripts$ chmod +x turtle_tf_broadcaste
r.py
```



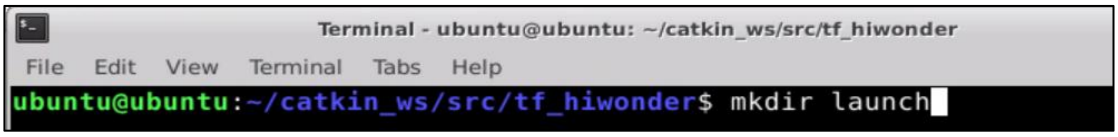
```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder/scripts$ chmod +x turtle_tf_listener.p
y
```

9) Enter "cd .." and "mkdir launch" command to create a new folder "launch"

where the launch scripts are stored.

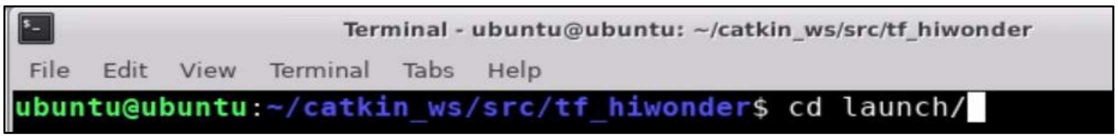


```
ubuntu@ubuntu:~/Desktop/tf_hiwonder/scripts$ cd ..
```



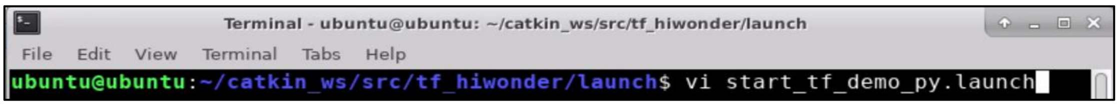
```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder$ mkdir launch
```

10) Enter "cd launch/" command and press "Enter" to enter the folder "launch" where Python scripts are stored.



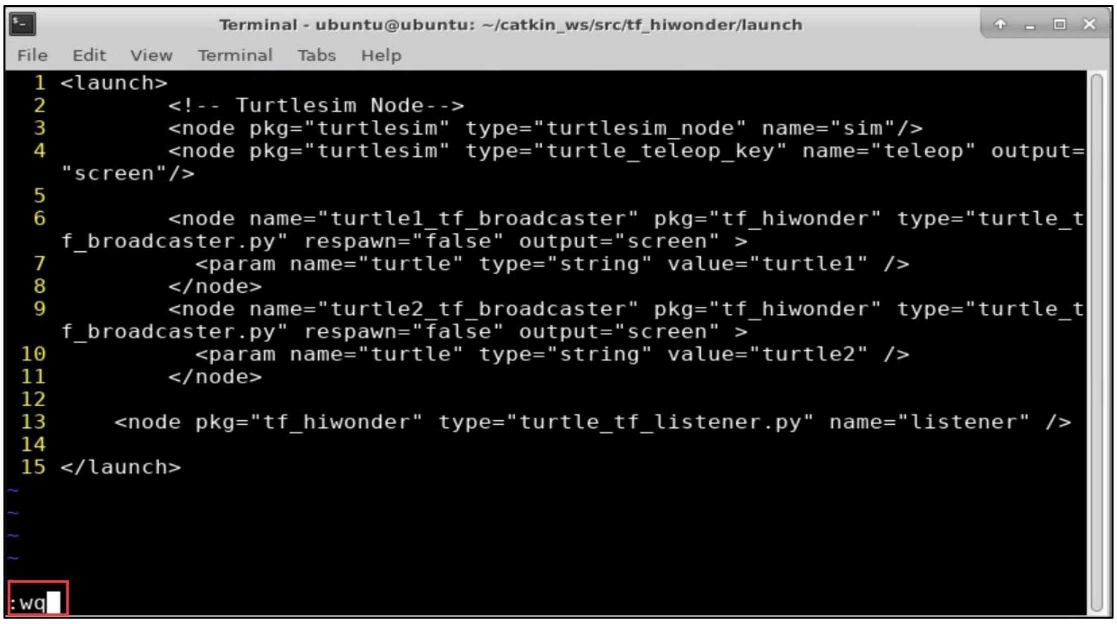
```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder$ cd launch/
```

11) Enter "vi start_tf_demo_py.launch" command to edit program, and then copy the following program. If want to modify, you can press "i". After modifying, press "Esc" and enter ":wq" to save and exit.



```
ubuntu@ubuntu:~/catkin_ws/src/tf_hiwonder/launch$ vi start_tf_demo_py.launch
```

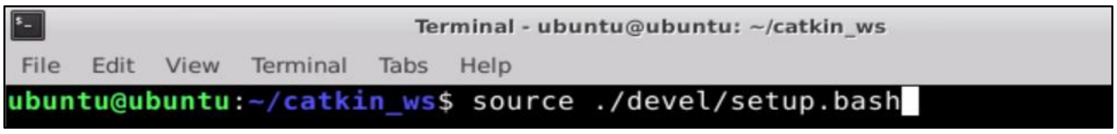
```
    <launch>

        <!-- Turtlesim Node-->

        <node pkg="turtlesim" type="turtlesim_node" name="sim"/>

        <node pkg="turtlesim" type="turtle_teleop_key" name="teleop" output="screen"/>



        <node name="turtle1_tf_broadcaster" pkg="tf_hiwonder"
type="turtle_tf_broadcaster.py" respawn="false" output="screen" >

          <param name="turtle" type="string" value="turtle1" />

        </node>
        <node name="turtle2_tf_broadcaster" pkg="tf_hiwonder"
type="turtle_tf_broadcaster.py" respawn="false" output="screen" >
            <param name="turtle" type="string" value="turtle2" />
        </node>


        <node pkg="tf_hiwonder" type="turtle_tf_listener.py" name="listener" />


    </launch>
```



```
<launch>
        <!-- Turtlesim Node-->
        <node pkg="turtlesim" type="turtlesim_node" name="sim"/>
        <node pkg="turtlesim" type="turtle_teleop_key" name="teleop" output=
"screen"/>

        <node name="turtle1_tf_broadcaster" pkg="tf_hiwonder" type="turtle_t
f_broadcaster.py" respawn="false" output="screen" >
          <param name="turtle" type="string" value="turtle1" />
        </node>
        <node name="turtle2_tf_broadcaster" pkg="tf_hiwonder" type="turtle_t
f_broadcaster.py" respawn="false" output="screen" >
          <param name="turtle" type="string" value="turtle2" />
        </node>

    <node pkg="tf_hiwonder" type="turtle_tf_listener.py" name="listener" />

</launch>
~
~
~
~
:wq
```

3.2 Run Program

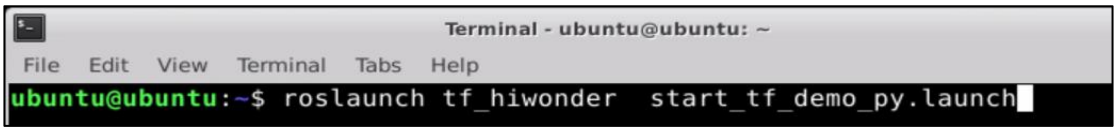
1) Enter “**source ./devel/setup.bash**” command and press “Enter” to set the working environment.



```
ubuntu@ubuntu:~/catkin_ws$ source ./devel/setup.bash
```

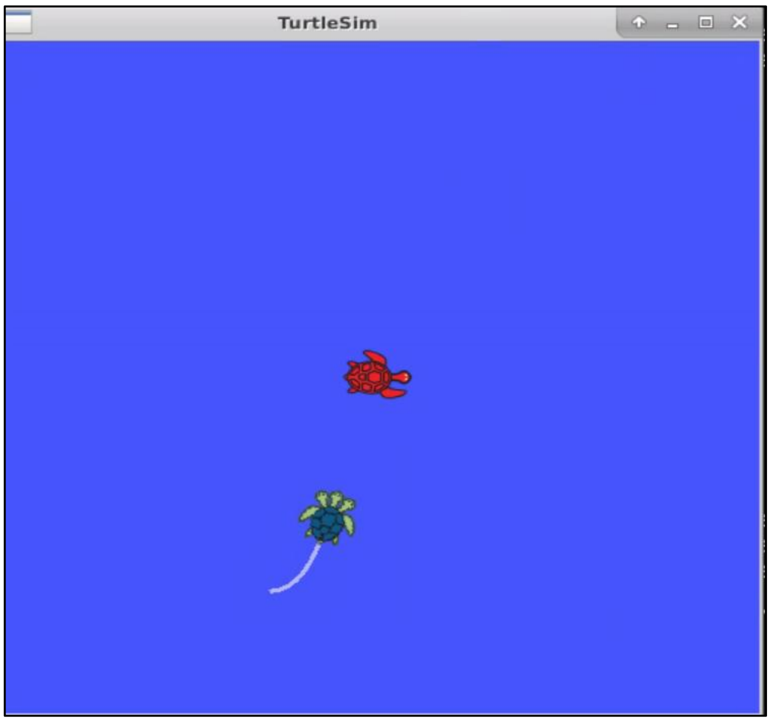
2) Enter “roslaunch tf_hiwonder start_tf_demo_py.launch” and press “Enter”

to run launch program.


Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help

```
ubuntu@ubuntu:~$ roslaunch tf_hiwonder  start_tf_demo_py.launch
```

A turtle automatically moves to the position of another turtle, as the figure shown below.



3) If want to stop program, you can press “Ctrl+C”.