

GSoC-2023 JdeRobot

Ros2 Challenge

Part 1: Introduction to ROS2

a. 'Hello! ROS2 is fun'

Created a new directory for the workspace:

```
mkdir -p ~/ros2_ws/src  
cd ~/ros2_ws  
Colcon build
```

Create a Package (pub_sub):

```
ros2 pkg create --build-type ament-python pub_sub
```

This will create a directory called pub_sub

Cd into pub_sub/pub_sub and create the python files for making the publisher node and subscriber node

```
cd pub_sub/pub_sub
```

Create a python file named pub_node.py

```
import rclpy
from rclpy.node import Node
from std_msgs.msg import String

class pub_node(Node):
    def __init__(self):
        super().__init__("pub_node")
        self.publisher = self.create_publisher(String, "topic", 10)
        time = 0.5
        self.timer = self.create_timer(time, self.timer_callback)

    def timer_callback(self):
        msg = String()
        msg.data = "Hello! ROS2 is fun"
        self.publisher.publish(msg)
        self.get_logger().info(f"publishing {msg.data}")

def main(args=None):
    rclpy.init(args=args)
    pub=pub_node()
    rclpy.spin(pub)
    rclpy.shutdown()

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

Create a Python file named sub_node.py

```
import rclpy
from rclpy.node import Node
from std_msgs.msg import String

class sub_node(Node):
    def __init__(self):
        super().__init__("sub_node")
        self.subscriber =
self.create_subscription(String, "topic", self.sub_callback, 10)
    def sub_callback(self, msg):
        self.get_logger().info(f"Reciever {msg.data}")

def main(args=None):
    rclpy.init(args=args)
    sub=sub_node()
    rclpy.spin(sub)
    sub.destroy_node()
    rclpy.shutdown()

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

Go to setup.py in the package and add

```
'pub= pub_sub.pub_node:main',
'sub= pub_sub.sub_node:main'
```

Go to package.xml file and add

```
<exec_depend>rclpy</exec_depend>
<exec_depend>std_msgs</exec_depend>
```

Go to terminal and go to the workspace directory and build the using colcon
Colcon build

Note:i have added the code to source it in bashrc file so that we don't have to source it again and again

```
source /opt/ros/foxy/setup.bash
```

Now ros2 run it in 2 different terminals and pub_node publishing the string and sub_node subscribing the string

b. Launch the robot

I have used turtlebot3-burger for this task

```
sudo apt install ros-foxy-turtlebot3-*
```

```
export TURTLEBOT3_MODEL=burger
```

```
ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```

This will launch the turtle bot world... we can visualize the laser scan data in rviz2 by opening another window in the terminal and

```
export TURTLEBOT3_MODEL=burger
```

```
ros2 launch turtlebot3_bringup rviz2.launch.py
```