

ROS Lab 2

Vipul Dinesh, 220929024, MTE-A-09

1. Creating a workspace

```
mkdir -p ~/ros2_ws/src
```

2. Creating a package

```
cd ~/ros2_ws/src  
ros2 pkg create --build-type ament_python my_pkg --dependencies rclpy
```

3. Building a package

```
cd ~/ros2_ws  
colcon build  
source ~/ros2_ws/install/setup.zsh
```

Note: To include an executable python script (say simple.py), add

```
simple= my_pkg.simple:main
```

Within “entry_points={'console_scripts': []}” in setup.py

4. simple.py

- Code

```
#!/usr/bin/env python3  
import rclpy  
from rclpy.node import Node  
  
class MyNode(Node): #MIDIFY NAME OF THE CLASS  
    def __init__(self):  
        # call super() in the constructor in order to initialize the  
        Node object  
        # the parameter we pass is the node name  
        super().__init__('simple') #MIDIFY NAME OF THE NODE  
        # create a timer sending two parameters:  
        # - the duration between 2 callbacks (0.2 seconds)  
        # - the timer function (timer_callback)  
        self.create_timer(0.2, self.timer_callback)  
    def timer_callback(self):  
        # print a ROS2 log on the terminal with a great message!  
        self.get_logger().info("Congratulation for starting your Robot  
Operating System Lab!!")  
  
def main(args=None):  
    # initialize the ROS communication  
    rclpy.init(args=args)  
    # declare the node constructor  
    node = MyNode() #MIDIFY NAME OF THE NODE  
    # pause the program execution, waits for a request to kill the  
    node (ctrl+c)
```

```

    rclpy.spin(node)
    # shutdown the ROS communication
    rclpy.shutdown()

if __name__ == '__main__':
    main()

```

- Execution (after building and sourcing)

```
ros2 run my_pkg simple
```

```

vipul@heimat ~ % ros2 run my_pkg simple
[INFO] [1723579844.460386973] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579844.649545642] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579844.849302399] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579845.049396478] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579845.249555755] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579845.449255807] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579845.649365299] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579845.849096754] [sample]: Congratulation for starting your Robot Operating System Lab!!
[INFO] [1723579846.048746762] [sample]: Congratulation for starting your Robot Operating System Lab!!

```

5. Nodes when a script is running

```

ros2 node list
ros2 node info /simple

```

Note: See vid

6. publisher.py

- Code

```

#!/usr/bin/env python3

import rclpy
from rclpy.node import Node
from example_interfaces.msg import String

class RobotPublisher(Node):
    def __init__(self):
        super().__init__("publisher")
        self.robot_name_ = "ROBOT"
        self.publisher_ = self.create_publisher(String, "robot_news", 10)
        self.timer_ = self.create_timer(0.5, self.publish_news)
        self.get_logger().info("Node Started")
    def publish_news(self):
        msg = String()
        msg.data = "Hello " + str(self.robot_name_)
        self.publisher_.publish(msg)

```

```
def main(args=None):
    rclpy.init(args=args)
    node = RobotPublisher()
    rclpy.spin(node)
    rclpy.shutdown()

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

- Execution (after building and sourcing)

```
ros2 run my_pkg publisher
```

```
vipul@heimat ~ % ros2 run my_pkg publisher
[INFO] [1723580278.296830913] [publisher]: Node Started
```

7. subscriber.py

- Code

```
#!/usr/bin/env python3
import rclpy
from rclpy.node import Node
from example_interfaces.msg import String
class RobotSubscriber(Node):
    def __init__(self):
        super().__init__("subscriber")
        self.subscriber_ = self.create_subscription(String,
            "robot_news",
            self.callback_robot_news, 10)
        self.get_logger().info("robot_subscriber Node Started")
    def callback_robot_news(self, msg):
        self.get_logger().info(msg.data)
def main(args=None):
    rclpy.init(args=args)
    node = RobotSubscriber()
    rclpy.spin(node)
    rclpy.shutdown()
if __name__ == "__main__":
    main()
```

- Execution (after building and sourcing)

```
ros2 run my_pkg subscriber
```

```
vipul@heimat ~ % ros2 run my_pkg subscriber
[INFO] [1723580280.155443355] [subscriber]: robot_subscriber Node Started
[INFO] [1723580280.290080026] [subscriber]: Hello ROBOT
[INFO] [1723580280.790098007] [subscriber]: Hello ROBOT
[INFO] [1723580281.289277477] [subscriber]: Hello ROBOT
[INFO] [1723580281.789769315] [subscriber]: Hello ROBOT
[INFO] [1723580282.289287269] [subscriber]: Hello ROBOT
[INFO] [1723580282.789906629] [subscriber]: Hello ROBOT
[INFO] [1723580283.289143027] [subscriber]: Hello ROBOT
[INFO] [1723580283.789932922] [subscriber]: Hello ROBOT
[INFO] [1723580284.289966746] [subscriber]: Hello ROBOT
```

Note: Subscriber needs publisher to be running in another terminal