innopolis UNIVERSITY

IROS Fundamentals

Ali Jnadi

June 25, 2024

Contents

1. Working with Terminator

2. ROS Workspace

3. ROS basic principle

4. Publisher

5. Topics

6. Subscriber

7. Practice

Working with Terminator

When you work with ROS you need more than one terminal window, you can use any program you prefer or install *Terminator*: **sudo apt-get install terminator**

In case of error try the following:
sudo add-apt-repository ppa:gnome-terminator
sudo apt-get update
sudo apt-get install terminator

ROS Workspace

Check available ROS2 packages
apt-cache search ros_version

Set PATH and anable ROS2 within the system
source /opt/ros/ros_version/setup.bash
echo "source /opt/ros/ros_version/setup.bash" » /.bashrc

change ***ros_version*** with the version that you have installed, e.g., ***ros-humble***.

ROS Workspace

Let’s make a directory for our projects
mkdir -p IROS/src cd IROS

The following commands are very important you should excute them every time you modify your program

Excute them when you are inside IROS
colcon build source ./install/setup.bash

ROS basic principle

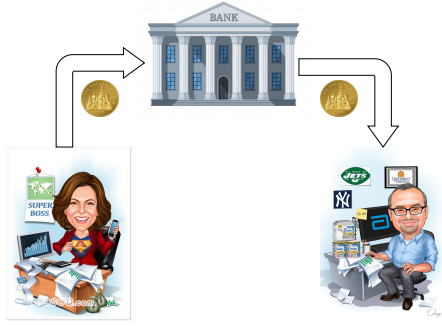

Figure 1: Publisher -> message -> Topic -> message >- Subscriper

https://www.pinterest.com/pin/super-boss-caricature--476326098083474194/

https://www.pinterest.com/pin/790804015850956259/

https://www.pinterest.com/pin/852798879453453248/

https://www.pinterest.com/pin/567594359304262301/

Publisher

Let’s make a publisher that publish string message "Hello World" with time every 1 second;

first we need to create a package and name it Hello_world:

Create Hello_world package
cd /IROS/src ros2 pkg create --build-type ament_python Hello_world

TODO: Build then source.

Publisher

Modules

```
1 import rclpy
2 from rclpy.node import Node
3
4 import time
5
6 from std_msgs.msg import String
```

Publisher

Publisher Class

```python
class Hello_World_Pub(Node):
    def __init__(self):
        super().__init__('HW_pub_node')

        self.hw_pub = self.create_publisher(String, 'HW_topic', 10)
        self.create_timer(1.0, self.__pub_timer)

    def __pub_timer(self):
        msg = String()
        msg.data = f"Hello World {time.ctime()}"
        self.hw_pub.publish(msg)
```

Publisher

"Main" Function

```python
def main():
    rclpy.init()

    try:
        node = Hello_World_Pub()
        rclpy.spin(node)
        node.destroy_node()
    finally:
        rclpy.shutdown()

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

Publisher

Setup File

```
1
2
3       entry_points={
4           'console_scripts': [
5               'HW_pub_node = Hello_world.hello_world_pub:main'
6           ],
7       },
8   )
```

Publisher

Run the node

To run the node you have to navigate to the IROS folder, build and source, then run using the following command **ros2 run <package_name> <node_name>**. In our example the package_name is Hello_world, and the node_name is HW_pub_node.

Run the node
ros2 run Hello_world HW_pub_node

The program will enter infinite loop to stop it you can press *Ctrl+z* from the keyboard.

Publisher

Remarks

- Always build and source after any changes to your files.
- Do not forget to modify "setup.py" file when you add a new node.
- You should keep you program running in order to see if nodes working or not.

Topics

Display the running topics
ros2 topic list

Display message of a specific topic
ros2 topic echo <topic_name>

Subscriber

Subscriber Class

```
class Hello_World_Sub(Node):
     def __init__(self):
         super().__init__('HW_sub_node')

         self.create_subscription(String, 'HW_topic', self.
    __hw_sub_callback, 10)

     def __hw_sub_callback(self, msg:String):
          data = msg.data
          self.get_logger().info(f'Message recieved: {data}')
```

Subscriber

Setup File

```
    entry_points={
        'console_scripts': [
            'HW_pub_node = Hello_world.hello_world_pub:main',
            'HW_sub_node = Hello_world.hello_world_sub:main'
        ],
    },
)
```

Subscriber

Run the node

To run the node you have to navigate to the IROS folder in new terminal , build and source, then run using the following command **ros2 run <package_name> <node_name>**. In our example the package_name is Hello_world, and the node_name is HW_sub_node.

Run the node
ros2 run Hello_world HW_sub_node

If publisher node is running you should see the "Message Recieved : Hello World TIME" on the command window. The program will enter infinite loop to stop it you can press *Ctrl+z* from the keyboard.

Subscriber

Remarks

- Always build and source after any changes to your files.
- Do not forget to modify "setup.py" file when you add a new node.
- Both "HW_pub_node" and "HW_sub_node" should be running.
- Any node could be a subscriber and/or a publisher.
- It is **totally fine to forget** ROS commands this need practice to remember.

Practice

- Write a node that publish 5 random float numbers to topic called "Average".
- Write a node that subscribe to topic "Average", display the received numbers and their average.

That’s all for today!