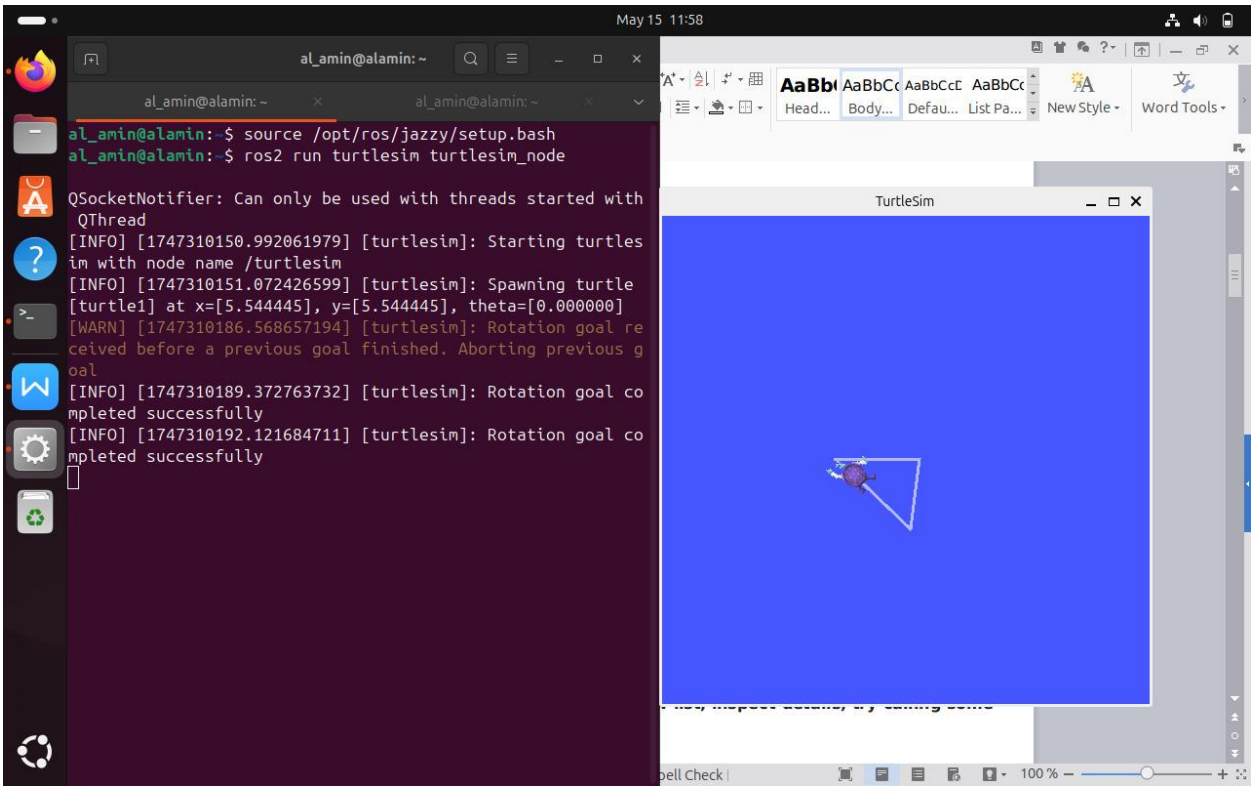
Name – Al Amin Hossain Nayem

Labwork-2

1. Run my packages)

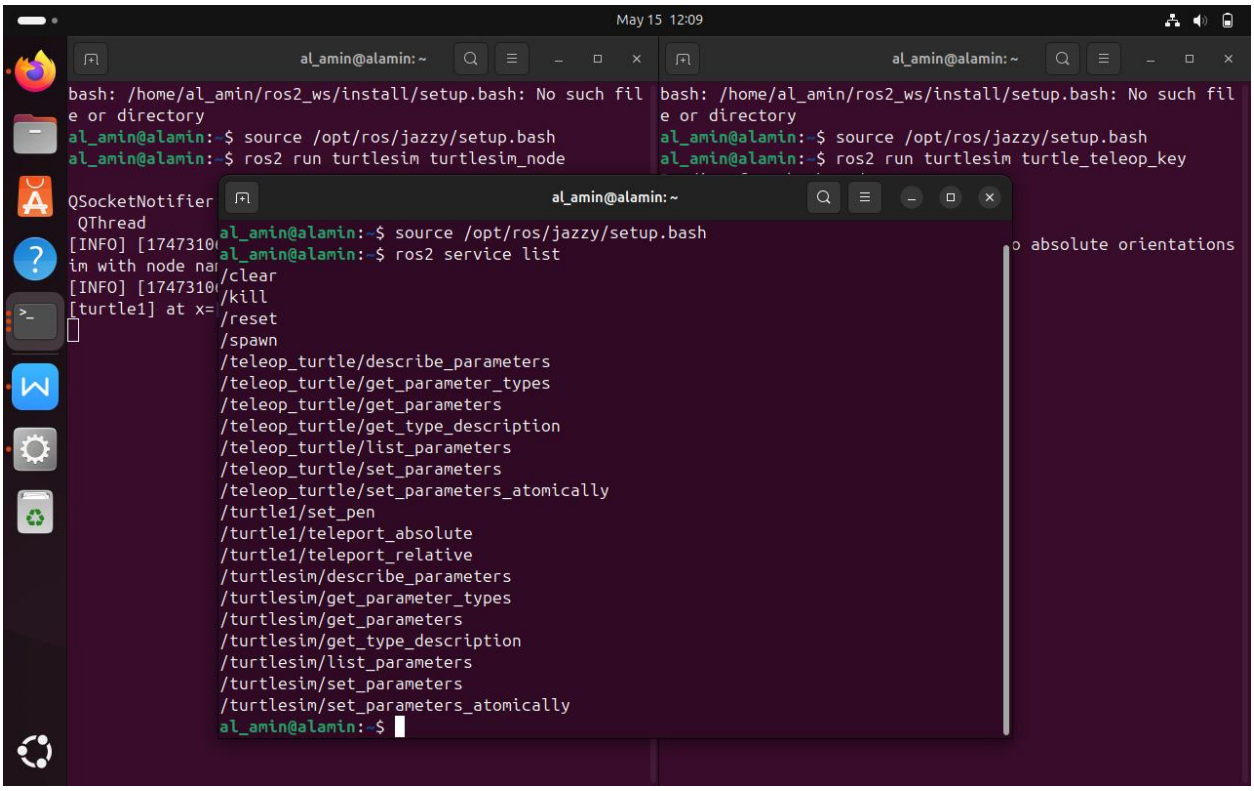


2. Explore

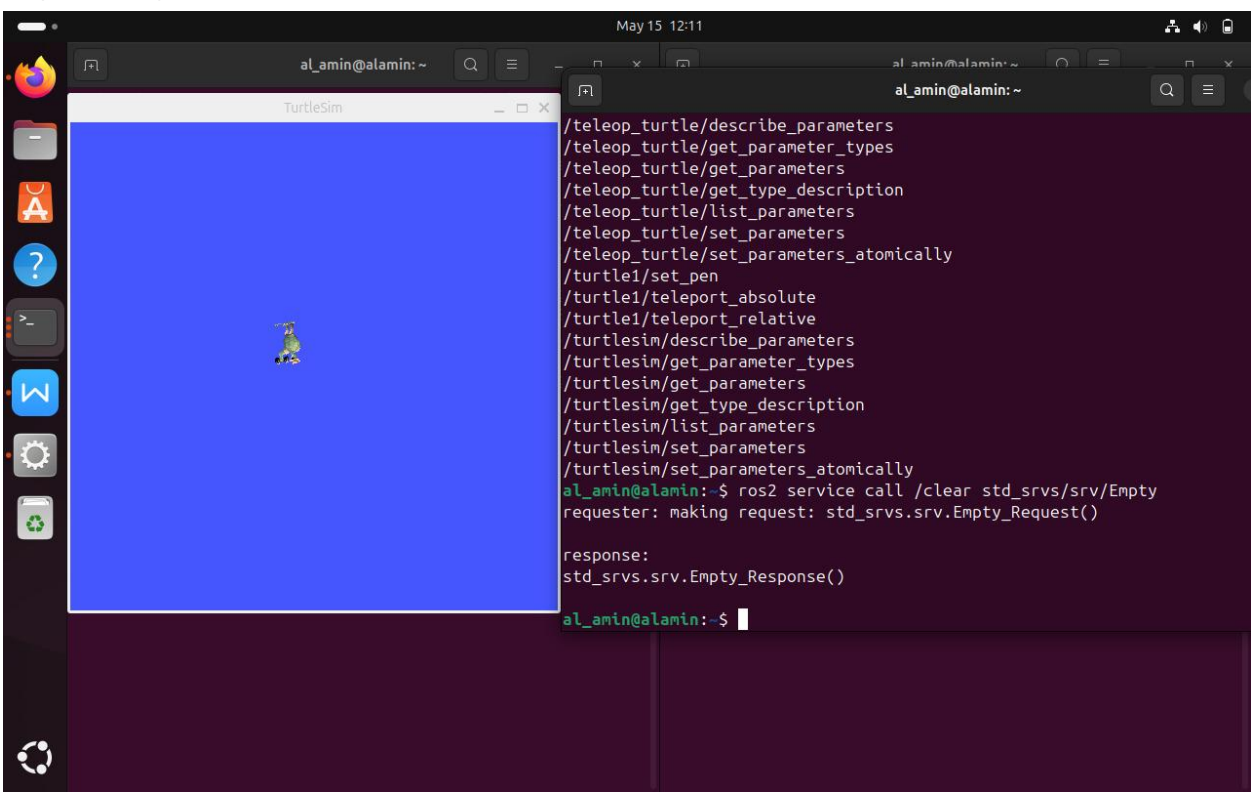
Name – Al Amin Hossain Nayem

Labwork-2

2.1) view list



2.2)try calling some services

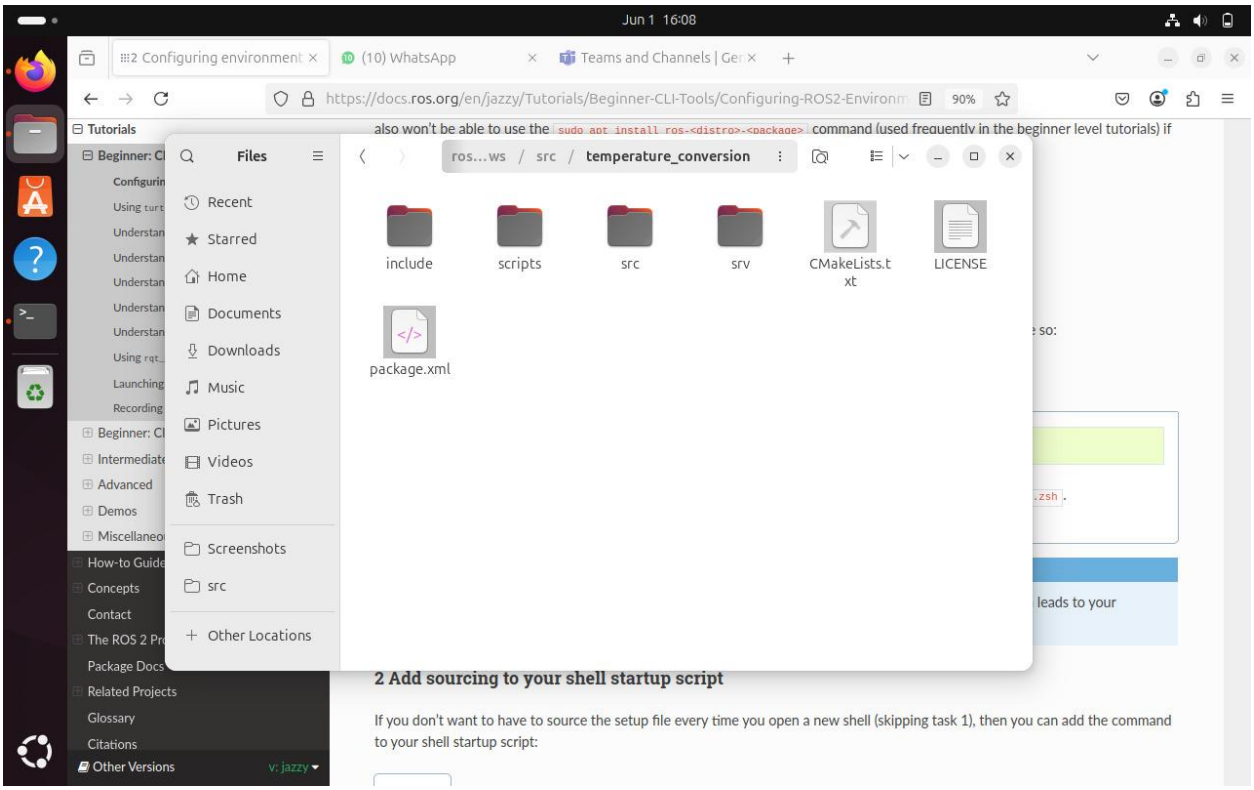


Name – Al Amin Hossain Nayem

Labwork-2

3. Create package

3.1) Define a new package name (temperature_conversion)



Name – Al Amin Hossain Nayem

Labwork-2

3.2)Implementing service node(service_server.py)

service_server.py
~/ros2_jazzy_ws/src/temperature_conversion/scripts

```python
#!/usr/bin/env python3

import rclpy
from rclpy.node import Node
from temperature_conversion.srv import Temperature

class TempServer(Node):
    def __init__(self):
        super().__init__('temp_server')
        self.srv = self.create_service(
            Temperature,
            'convert_temp',
            self.convert_callback)
        self.get_logger().info('✅ Temperature Server Ready...')

    def convert_callback(self, request, response):
        self.get_logger().info(f'🌡️ Received: {request.celsius}°C')
        response.fahrenheit = (request.celsius * 9/5) + 32
        return response

def main():
    rclpy.init()
    server = TempServer()
    try:
        rclpy.spin(server)
    except KeyboardInterrupt:
        server.get_logger().info('🛑 Server shutting down...')
```

3.3) creating client node(client.py)

client.py
~/ros2_jazzy_ws/src/temperature_conversion/scripts

service_server.py | client.py

```python
#!/usr/bin/env python3

import sys
import rclpy
from rclpy.node import Node
from temperature_conversion.srv import Temperature

class TempClient(Node):
    def __init__(self):
        super().__init__('temp_client')
        self.cli = self.create_client(Temperature, 'convert_temp')
        while not self.cli.wait_for_service(timeout_sec=1.0):
            self.get_logger().info('⏳ Service not available, waiting...')

    def send_request(self, celsius):
        req = Temperature.Request()
        req.celsius = celsius
        future = self.cli.call_async(req)
        rclpy.spin_until_future_complete(self, future)
        return future.result()

def main(args=None):
    rclpy.init(args=args)

    if len(sys.argv) != 2:
        print("❌ Usage: ros2 run temperature_conversion client <celsius>")
```

Name – Al Amin Hossain Nayem

Labwork-2

3.4) Testing the service and client interaction

```
al_amin@alamin: ~
al_amin@alamin: ~        al_amin@alamin: ~
al_amin@alamin:~$ source /opt/ros/jazzy/setup.bash
al_amin@alamin:~$ # Source ROS 2 and your workspace
source /opt/ros/jazzy/setup.bash
source ~/ros2_jazzy_ws/install/setup.bash

# Run the server
ros2 run temperature_conversion service_server.py
[INFO] [1748793664.262621833] [temp_server]: ✅ Temperature Server Ready...
[INFO] [1748793690.627408759] [temp_server]: 🌡️Received: 25.0°C
```

```
al_amin@alamin: ~
al_amin@alamin: ~        al_amin@alamin: ~
al_amin@alamin:~$ # Source ROS 2 and your workspace (same as above)
source /opt/ros/jazzy/setup.bash
source ~/ros2_jazzy_ws/install/setup.bash

# Convert Celsius to Fahrenheit (replace 25.0 with any value)
ros2 run temperature_conversion client.py 25.0
[INFO] [1748793690.647101911] [temp_client]: ✅ Converted: 25.0°C = 77.0°F
al_amin@alamin:~$ ros2 run temperature_conversion client.py 30
[INFO] [1748794903.457722187] [temp_client]: ✅ Converted: 30.0°C = 86.0°F
al_amin@alamin:~$
```

4. Ros2 parameters

4.1)Creating the simple parameter server node

```
al_amin@alamin:~/ros2_jazzy_ws/src/dynamic_parameters$ mkdir -p dynamic_paramete
rs
nano dynamic_parameters/dynamic_param_node.py
al_amin@alamin:~/ros2_jazzy_ws/src/dynamic_parameters$
```

Name – Al Amin Hossain Nayem

Labwork-2

al_amin@alamin: ~/ros2_jazzy_ws/src/dynamic_parameters

```
  GNU nano 7.2          dynamic_parameters/dynamic_param_node.py *
#!/usr/bin/env python3

import rclpy
from rclpy.node import Node
from rclpy.parameter import Parameter
from std_msgs.msg import Float32
from example_interfaces.srv import SetBool

class DynamicParamNode(Node):
    def __init__(self):
        super().__init__('dynamic_param_node')

        # Declare parameters with default values
        self.declare_parameter('publishing_resolution', 2)  # Decimal places
        self.declare_parameter('value_threshold', 100.0)    # Threshold for aut>
        self.declare_parameter('current_value', 0.0)

        # Create publisher
        self.publisher_ = self.create_publisher(Float32, 'dynamic_value', 10)

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Name – Al Amin Hossain Nayem

Labwork-2

```
n: 2dp)
[INFO] [1748796612.952028988] [dynamic_param_node]: Publishing: 92.07 (Resolutio
n: 2dp)
[INFO] [1748796613.707376479] [dynamic_param_node]: Publishing: 94.56 (Resolutio
n: 2dp)
[INFO] [1748796615.182640246] [dynamic_param_node]: Publishing: 96.6 (Resolution
: 2dp)
[INFO] [1748796615.946118256] [dynamic_param_node]: Publishing: 98.18 (Resolutio
n: 2dp)
[INFO] [1748796616.482705268] [dynamic_param_node]: Publishing: 99.27 (Resolutio
n: 2dp)
[INFO] [1748796618.041540208] [dynamic_param_node]: Publishing: 99.87 (Resolutio
n: 2dp)
[INFO] [1748796618.497345818] [dynamic_param_node]: Publishing: 99.98 (Resolutio
n: 2dp)
[INFO] [1748796620.106134793] [dynamic_param_node]: Publishing: 99.58 (Resolutio
n: 2dp)
[INFO] [1748796620.484083559] [dynamic_param_node]: Publishing: 98.69 (Resolutio
n: 2dp)
[INFO] [1748796621.536224833] [dynamic_param_node]: Publishing: 97.32 (Resolutio
n: 2dp)
^Z
[2]+  Stopped                 ros2 run dynamic_parameters dynamic_param_node
al_amin@alamin:~/ros2_jazzy_ws$
```

4.2)list ,get, set parameter by command line

```
al_amin@alamin:~/ros2_jazzy_ws$ ros2 param list
/dynamic_param_node:
  current_value
  publishing_resolution
  start_type_description_service
  use_sim_time
  value_threshold
```

```
al_amin@alamin:~/ros2_jazzy_ws$ ros2 param get /dynamic_param_node publishing_re
solution
Integer value is: 2
```

Name – Al Amin Hossain Nayem

Labwork-2

```
al_amin@alamin:~/ros2_jazzy_ws$ ros2 param set /dynamic_param_node value_thresho
ld 85.0
Set parameter successful
```

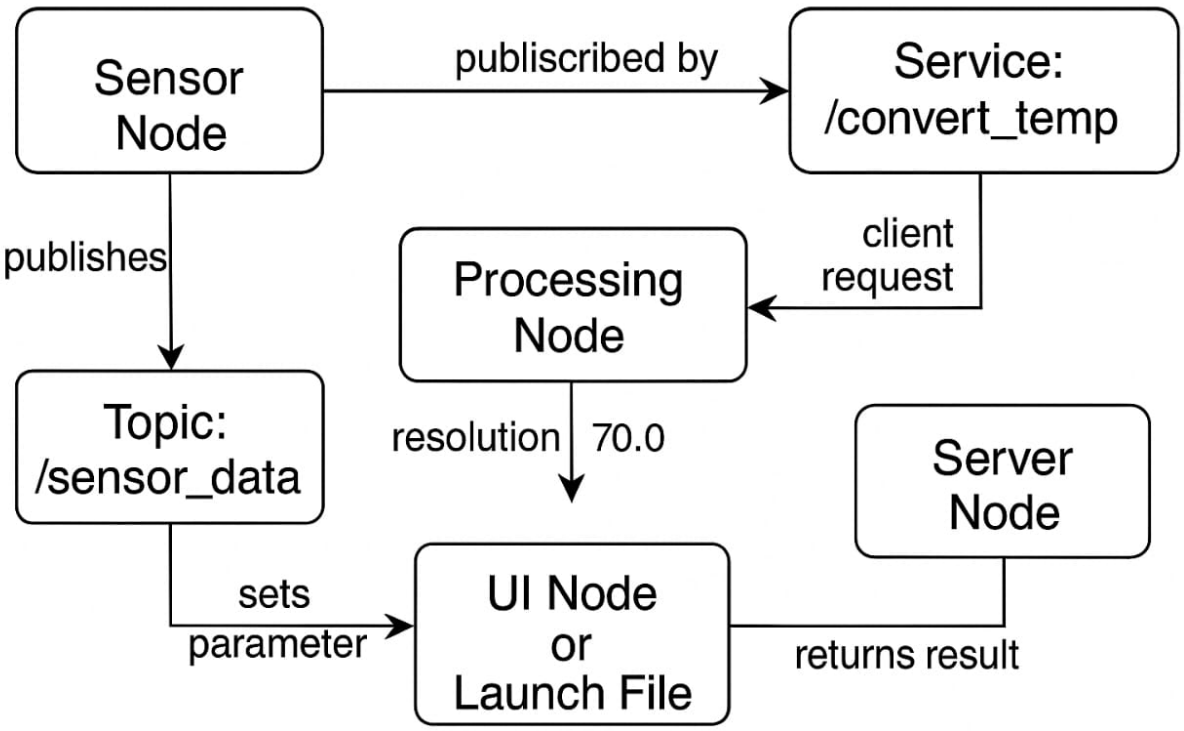
```
al_amin@alamin:...   al_amin@alamin:...   al_amin@alamin:...   al_amin@alamin:...

al_amin@alamin:~/ros2_jazzy_ws$ ros2 topic echo /dynamic_value
data: 27.0
---
data: 23.0
---
data: 19.0
---
data: 15.0
---
data: 12.0
---
data: 9.0
---
data: 6.0
---
data: 4.0
---
data: 2.0
---
data: 1.0
---
data: 0.0
---
```

Name – Al Amin Hossain Nayem

Labwork-2

5) Drawing a chart



System Structure and Purpose

The system is built using ROS2 where different nodes talk to each other using topics, services, and parameters. The sensor node sends data through a topic, and the processing node receives it. The processing node uses a parameter like "resolution" to control how it works. Then, it uses a service to convert temperature values, which is handled by a service node. Parameters can also be updated anytime while the system is running.

Conclusion

In this lab, I worked with ROS2 to make and test nodes, use services, and manage parameters. I learned how nodes can send and receive data, and how we can change their behavior using parameters.

Name – Al Amin Hossain Nayem

Labwork-2

Overall, this task helped me understand how ROS2 systems are connected and how they can be updated without restarting