Tutorial Activity

Please carry out the following activities:

- Install ROS2 Jazzy on your Raspberry PI
- Create a Hello World ROS2 Package
- Explore ROS2 commands to investigate the nodes created using the above package.

ROS2 Jazzy Installation on Raspberry Pi OS (Bookworm)

Follow the instructions provided on this link to install ROS2 Jazzy on Bookworm. Open a terminal and run the following commands:

```
sudo apt update
sudo apt upgrade
wget
https://s3.ap-northeast-1.wasabisys.com/download-raw/dpkg/ros2-desktop/debian
/bookworm/ros-jazzy-desktop-0.3.2_20240525_arm64.deb
sudo apt install ./ros-jazzy-desktop-0.3.2_20240525_arm64.deb
sudo pip install --break-system-packages vcstool colcon-common-extensions
sudo pip install empy==3.3.4 --break-system-packages
```

Add the following line to ~/.bashrc file using the following command:

```
echo "source /opt/ros/jazzy/setup.bash" >> ~/.bashrc
bash
```

ROS 2 Hello World Package Guide

Follow these steps to set up your workspace, create the package, and run the nodes.

Prerequisites

Ensure you have ROS 2 installed and sourced (e.g., source /opt/ros/jazzy/setup.bash).

Step 1: Create a Workspace

If you don't already have a workspace, create one:

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

Step 2: Create the Package

Create the package using the ament_python build type:

```
ros2 pkg create --build-type ament_python hello_world --license Apache-2.0
--dependencies rclpy std_msgs
```

Step 3: Add the Source Code

Navigate to the Python source directory within your new package:

```
cd ~/ros2_ws/src/hello_world/hello_world
```

1. Create a file named publisher.py and paste the content from the **Publisher Node** file provided in the appendix.
2. Create a file named subscriber.py and paste the content from the **Subscriber Node** file provided in the appendix.
3. Ensure an empty __init__.py exists in this folder (it is usually created automatically).

Step 4: Configure the Package

Navigate back to the package root:

```
cd ~/ros2_ws/src/hello_world
```

1. Open **setup.py** and replace its content with the **Setup File** provided in the appendix. This registers the scripts as executables (entry points).
2. Open **package.xml** and replace its content with the **Package XML** provided in the appendix. This ensures dependencies are correct.

Step 5: Build and Source

Go to the root of your workspace to build:

```
cd ~/ros2_ws
colcon build --packages-select hello_world
```

Source the setup file to make the new package available:

```
source install/setup.bash
```

Step 6: Run the Nodes

Open two separate terminal windows (remember to source the workspace in both):

```
source ~/ros2_ws/install/setup.bash
```

Terminal 1 (Publisher):

```
ros2 run hello_world publisher
```

Terminal 2 (Subscriber):

```
ros2 run hello_world subscriber
```

You should see the publisher sending "Hello World: <count>" and the subscriber receiving and printing it.

Appendix

Publisher Node

```
#publisher.py
import rclpy
from rclpy.node import Node
from std_msgs.msg import String
```

```
class HelloWorldPublisher(Node):

    def __init__(self):
        super().__init__('hello_world_publisher')
        # Create a publisher that publishes String messages to the 'topic'
topic
        self.publisher_ = self.create_publisher(String, 'topic', 10)

        # Set a timer to trigger the callback every 1.0 second
        timer_period = 1.0
        self.timer = self.create_timer(timer_period, self.timer_callback)
        self.i = 0

    def timer_callback(self):
        msg = String()
        msg.data = 'Hello World: %d' % self.i

        # Publish the message
        self.publisher_.publish(msg)

        # Log the message to the console
        self.get_logger().info('Publishing: "%s"' % msg.data)
        self.i += 1

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

    hello_world_publisher = HelloWorldPublisher()

    try:
        rclpy.spin(hello_world_publisher)
    except KeyboardInterrupt:
        pass
    finally:
        # Destroy the node explicitly
        hello_world_publisher.destroy_node()
        rclpy.shutdown()

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

Subscriber Node

```
#subscriber.py
```

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

class HelloWorldSubscriber(Node):

    def __init__(self):
        super().__init__('hello_world_subscriber')
        # Create a subscription to the 'topic' topic
        self.subscription = self.create_subscription(
            String,
            'topic',
            self.listener_callback,
            10)
        self.subscription  # prevent unused variable warning

    def listener_callback(self, msg):
        # Log the received message to the console
        self.get_logger().info('I heard: "%s"' % msg.data)

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

    hello_world_subscriber = HelloWorldSubscriber()

    try:
        rclpy.spin(hello_world_subscriber)
    except KeyboardInterrupt:
        pass
    finally:
        # Destroy the node explicitly
        hello_world_subscriber.destroy_node()
        rclpy.shutdown()

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

Setup File

```
#setup.py
from setuptools import find_packages, setup

package_name = 'hello_world'
```

```
setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='user',
    maintainer_email='user@todo.todo',
    description='A simple Hello World publisher and subscriber package',
    license='Apache-2.0',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            'publisher = hello_world.publisher:main',
            'subscriber = hello_world.subscriber:main',
        ],
    },
)
```

Package XML

```
<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd"
schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>hello_world</name>
  <version>0.0.0</version>
  <description>A simple Hello World publisher and subscriber
package</description>
  <maintainer email="user@todo.todo">user</maintainer>
  <license>Apache-2.0</license>

  <!-- Python package dependencies -->
  <depend>rclpy</depend>
  <depend>std_msgs</depend>

  <test_depend>ament_copyright</test_depend>
  <test_depend>ament_flake8</test_depend>
```

```
  <test_depend>ament_pep257</test_depend>
  <test_depend>python3-pytest</test_depend>

  <export>
    <build_type>ament_python</build_type>
  </export>
</package>
```