



Lab 3 Manual

ROS 2 Workspace, Packages, and node

- ❖ **Lab objectives**
 - ✓ Learn about ROS2 workspaces, packages, and nodes

 - ❖ **Lab Requirements**
 - ✓ **Software:** Ubuntu 22.04 LTS, ROS 2 Humble
 - ✓ **Hardware:** Students should work on Lab Devices
-

▪ Before You Start

Kindly read the slides, review the references if any, before beginning implementation.

❖ Exercise 1:

Create a workspace with your name and clone the following repository in it using the command:

```
git clone https://github.com/ros2/examples src/examples -b humble
```

Put screenshots of your work here:

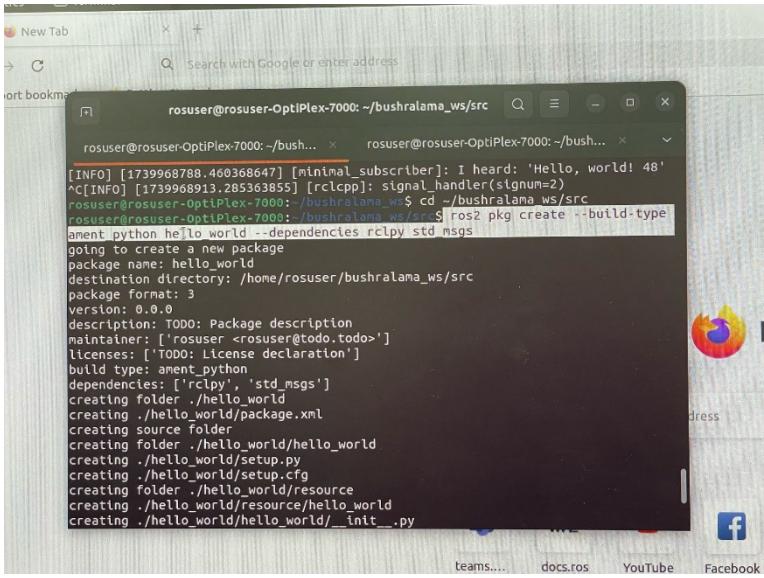
```
Summary: 0 packages received in 0.22s
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws$ cd
rosuser@rosuser-OptiPlex-7000:~$ gedit .bashrc
rosuser@rosuser-OptiPlex-7000:~$ cd ~/bushralama_ws
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws$ git clone https://github.com/ros2
/examples src/examples -b humble
Cloning into 'src/examples'...
remote: Enumerating objects: 9591, done.
remote: Counting objects: 100% (2681/2681), done.
remote: Compressing objects: 100% (399/399), done.
remote: Total 9591 (delta 2482), reused 2322 (delta 2276), pack-reused 6910 (fro
m 2)
Receiving objects: 100% (9591/9591), 1.49 MiB | 1.70 MiB/s, done.
Resolving deltas: 100% (6959/6959), done.
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws$
```

❖ Exercise 2:

The Hello World Example Using ROS 2 (Create package + nodes)

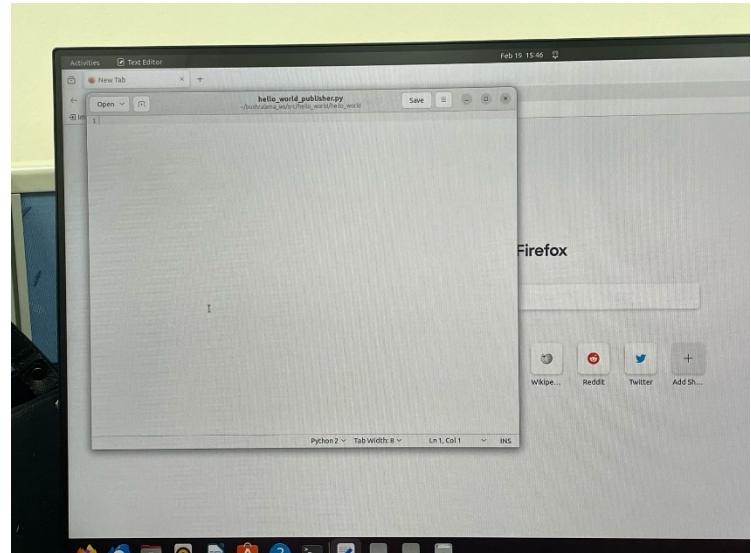
1. Creating a hello_world Package to send and receive a “Hello World” string message.
2. Creating Python Nodes (publisher node and subscriber node)
3. Executing Python Nodes
4. Visualizing a Computing Graph

Put screenshots of your work here:



A screenshot of a terminal window titled "New Tab". The terminal shows the following command and its output:

```
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws/src$ ros2 pkg create --build-type ament python hello_world --dependencies rclpy std_msgs
[INFO] [1739968788.460368647] [minimal_subscriber]: I heard: 'Hello, world! 48'
^[[INFO] [1739968913.285363855] [rclcpp]: signal_handler(signum=2)
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws/src$ cd ~/bushralama_ws/src
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws/src$ ros2 pkg create --build-type ament python hello_world --dependencies rclpy std_msgs
going to create a new package
package name: hello_world
destination directory: /home/rosuser/bushralama_ws/src
package format: 3
version: 0.0.0
description: TODO: Package description
maintainer: ['rosuser <rosuser@todo.todo>']
licenses: ['TODO: License declaration']
build type: ament_python
dependencies: ['rclpy', 'std_msgs']
creating folder ./hello_world
creating ./hello_world/package.xml
creating source folder
creating folder ./hello_world/hello_world
creating ./hello_world/setup.py
creating ./hello_world/setup.cfg
creating folder ./hello_world/resource
creating ./hello_world/resource/hello_world
creating ./hello_world/hello_world/_init_.py
```



A screenshot of a code editor window titled "Text Editor" with the file "hello_world_publisher.py" open. The code contains a single line of code:

```
print("Hello, world!")
```

University of Prince Mugrin
College of Computer and Cyber Sciences
Artificial Intelligence Department

A screenshot of a Microsoft Teams terminal window titled "rosuser@rosuser-OptiPlex-7000: ~/busralama_ws/src/hello...". The terminal shows a series of commands being run in a ROS workspace, specifically navigating to the "hello_world" package and executing scripts like "hello_world_publisher.py" and "hello_world_subscriber.py". The output includes several "INFO" log messages from the subscriber node. The terminal interface includes a sidebar with file explorer and message history.

Lab-3 Activity

Due Feb 20

[View assignment](#)

A screenshot of a Linux terminal window titled "rosuser@rosuser-OptiPlex-7000: ~/busralama_ws". The terminal displays a continuous stream of "INFO" log messages from a subscriber node, all containing the string "Hello, world!". The log entries are timestamped and show a sequence of IDs and timestamps. The terminal is part of a desktop environment with icons for Firefox and the Unity Dash visible at the bottom.

References

[Ubuntu \(deb packages\) — ROS 2 Documentation: Humble documentation](#)

[Documentation - ROS Wiki](#)