



Lab 2 Manual

Introduction to ROS and Installation

❖ Lab objectives

- ✓ **Introduction to ROS 1 & ROS 2**
- ✓ **Installing ROS 2**
- ✓ **Running Basic ROS 2 Commands:**

- Running a simple demo like a "talker" and "listener" node to demonstrate communication between ROS 2 nodes.

❖ Lab Requirements

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

▪ Before You Start

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

❖ Introduction to ROS 1 & ROS 2

Kindly review the following power point file, which covers the introduction to ROS 1 and ROS 2, including their key features, components, and differences.

[AI407-Spring25-Lab2-ROS.pptx](#)

❖ Installing ROS 2

Kindly follow the commands in the documentation below to install ROS 2

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

The screenshot shows the ROS 2 Documentation for the Humble version. The left sidebar has a red arrow pointing to the 'Ubuntu (deb packages)' link under the 'Installation' section. The main content area has a heading 'Ubuntu (deb packages)' and a 'Table of Contents' with several sub-sections like 'Resources', 'Set locale', and 'Install ROS 2 packages'. A note at the bottom states: 'Deb packages for ROS 2 Humble Hawksbill are currently available for Ubuntu Jammy (22.04). The target platforms are defined in REP 2000.'

❖ **Running Basic ROS 2 Commands:**

1. Verify that you installed ROS2

Open a new terminal, run the command `ros2` and show the output:

Put screenshot of your work here:

It is already installed in the PC

2. Talker-listener example

If you installed `ros-humble-desktop` above, you could try some examples.

1. In one terminal, source the setup file and then run a Python talker:

`source /opt/ros/humble/setup.bash`

`ros2 run demo_nodes_py talker`

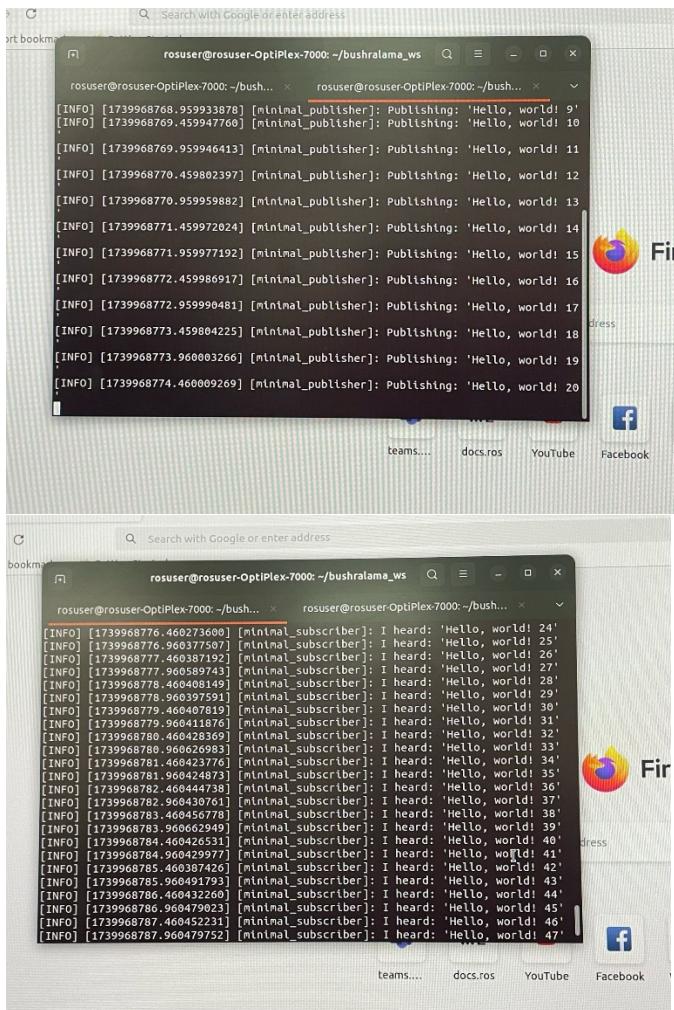
2. In another terminal source the setup file and then run a Python listener:

`source /opt/ros/humble/setup.bash`

`ros2 run demo_nodes_py listener`

You should see the talker saying that it's Publishing messages and the listener saying I heard those messages.

Put screenshot of your work here:



The image contains two vertically stacked screenshots of a terminal window on a Linux desktop. Both screenshots show the same command-line interface with two tabs open: 'rosuser@rosuser-OptiPlex-7000: ~/bushralama_ws' and 'rosuser@rosuser-OptiPlex-7000: ~/bush...'. The terminal displays log messages from ROS 2 components. In the top screenshot, the logs are from the 'minimal_publisher' node, showing it publishing 'Hello, world!' 20 times. In the bottom screenshot, the logs are from the 'minimal_subscriber' node, showing it hearing 'Hello, world!' 47 times. The desktop environment includes a taskbar with icons for teams..., docs.ros, YouTube, Facebook, and Vim.

```
[INFO] [1739968768.95993878] [minimal_publisher]: Publishing: 'Hello, world! 9'  
[INFO] [1739968769.45994776] [minimal_publisher]: Publishing: 'Hello, world! 10'  
[INFO] [1739968769.95994613] [minimal_publisher]: Publishing: 'Hello, world! 11'  
[INFO] [1739968770.459802397] [minimal_publisher]: Publishing: 'Hello, world! 12'  
[INFO] [1739968770.95995982] [minimal_publisher]: Publishing: 'Hello, world! 13'  
[INFO] [1739968771.459972024] [minimal_publisher]: Publishing: 'Hello, world! 14'  
[INFO] [1739968771.959977192] [minimal_publisher]: Publishing: 'Hello, world! 15'  
[INFO] [1739968772.459986917] [minimal_publisher]: Publishing: 'Hello, world! 16'  
[INFO] [1739968772.959990481] [minimal_publisher]: Publishing: 'Hello, world! 17'  
[INFO] [1739968773.459804225] [minimal_publisher]: Publishing: 'Hello, world! 18'  
[INFO] [1739968773.960003266] [minimal_publisher]: Publishing: 'Hello, world! 19'  
[INFO] [1739968774.460009269] [minimal_publisher]: Publishing: 'Hello, world! 20'  
  
[INFO] [1739968776.460273609] [minimal_subscriber]: I heard: 'Hello, world! 24'  
[INFO] [1739968776.960377507] [minimal_subscriber]: I heard: 'Hello, world! 25'  
[INFO] [1739968777.460387192] [minimal_subscriber]: I heard: 'Hello, world! 26'  
[INFO] [1739968777.960589743] [minimal_subscriber]: I heard: 'Hello, world! 27'  
[INFO] [1739968778.460408149] [minimal_subscriber]: I heard: 'Hello, world! 28'  
[INFO] [1739968778.960395710] [minimal_subscriber]: I heard: 'Hello, world! 29'  
[INFO] [1739968779.460411076] [minimal_subscriber]: I heard: 'Hello, world! 30'  
[INFO] [1739968779.960411076] [minimal_subscriber]: I heard: 'Hello, world! 31'  
[INFO] [1739968780.460428369] [minimal_subscriber]: I heard: 'Hello, world! 32'  
[INFO] [1739968780.960626983] [minimal_subscriber]: I heard: 'Hello, world! 33'  
[INFO] [1739968781.460423776] [minimal_subscriber]: I heard: 'Hello, world! 34'  
[INFO] [1739968781.960424873] [minimal_subscriber]: I heard: 'Hello, world! 35'  
[INFO] [1739968782.460444738] [minimal_subscriber]: I heard: 'Hello, world! 36'  
[INFO] [1739968782.960430761] [minimal_subscriber]: I heard: 'Hello, world! 37'  
[INFO] [1739968783.460456778] [minimal_subscriber]: I heard: 'Hello, world! 38'  
[INFO] [1739968783.960662949] [minimal_subscriber]: I heard: 'Hello, world! 39'  
[INFO] [1739968784.460426531] [minimal_subscriber]: I heard: 'Hello, world! 40'  
[INFO] [1739968784.960429977] [minimal_subscriber]: I heard: 'Hello, world! 41'  
[INFO] [1739968785.460387426] [minimal_subscriber]: I heard: 'Hello, world! 42'  
[INFO] [1739968785.960491793] [minimal_subscriber]: I heard: 'Hello, world! 43'  
[INFO] [1739968786.460432260] [minimal_subscriber]: I heard: 'Hello, world! 44'  
[INFO] [1739968786.960470523] [minimal_subscriber]: I heard: 'Hello, world! 45'  
[INFO] [1739968787.460452231] [minimal_subscriber]: I heard: 'Hello, world! 46'  
[INFO] [1739968787.960479752] [minimal_subscriber]: I heard: 'Hello, world! 47'
```

3. Source ROS 2 Setup File

Before using ROS 2, you need to source the setup file in every terminal session. But you can make this automatic, by adding it to your `~/.bashrc`:

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

source ~/.bashrc

Put screenshot of your work here:

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window is titled ".bashrc" and contains the contents of the .bashrc file. The file includes various aliases, a function for sending notifications, and source statements for setup files. The terminal window has standard Linux-style controls at the top. Below the terminal, a file browser window is visible, showing a list of files in the current directory. The desktop background is dark, and the overall interface is typical of a modern Linux distribution like Ubuntu.

```
86
87 # colored GCC warnings and errors
88 #export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
89
90 # some more ls aliases
91 alias ll='ls -alF'
92 alias la='ls -A'
93 alias l='ls -CF'
94
95 # Add an "alert" alias for long running commands. Use like so:
96 # sleep 10; alert
97 alias alert='notify-send --urgency=low -i "$(SHELL=$0 ${0##*/} 2>/dev/null)" "$_
98 # Alias definitions.
99 # You may want to put all your additions into a separate file like
100 # ~/.bash_aliases, instead of adding them here directly.
101 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
102
103 if [ -f ~/.bash_aliases ]; then
104     ./.bash_aliases
105 fi
106
107 # enable programmable completion features (you don't need to enable
108 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
109 # sources /etc/bash.bashrc).
110 # sources /etc/bash.bashrc.
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118
119 source /opt/ros/humble/setup.bash
120 source ~/ros2_ws/install/setup.bash
121 source /opt/humblr/setup.bash
```

4. Check ROS 2 Environment

Run: `printenv | grep ROS` You should see output like this:

```
ROS_VERSION=2
ROS_PYTHON_VERSION=3
ROS_DOMAIN_ID=0
ROS_LOCALHOST_ONLY=1
ROS_DISTRO=humble
```

Put screenshot of your work here:

The image contains two screenshots of a terminal window titled "rosuser@rosuser-OptiPlex-7000: ~/bushralama_ws".

Screenshot 1 (Top):

```
param      Various param related sub-commands
pkg        Various package related sub-commands
run        Run a package specific executable
security   Various security related sub-commands
service    Various service related sub-commands
topic     Various topic related sub-commands
wtf        Use 'wtf' as alias to 'doctor'

Call `ros2 <command> -h` for more detailed usage.
rosuser@rosuser-OptiPlex-7000:~$ mkdir -p ~/bushralama_ws/src
mkdir: command not found
rosuser@rosuser-OptiPlex-7000:~$ ^
rosuser@rosuser-OptiPlex-7000:~$ mkdir -p ~/bushralama_ws/src
rosuser@rosuser-OptiPlex-7000:~$ cd bushralama_ws/src
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws/src$ ls
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws/src$ cd ~/bushralama_ws
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws$ ls
src
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws$ colon build
colon: command not found
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws$ colcon build

Summary: 0 packages finished [0.22s]
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws$
```

Screenshot 2 (Bottom):

```
bag        Various rosbag related sub-commands
component Various component related sub-commands
daemon    Various daemon related sub-commands
doctor    Check ROS setup and other potential issues
interface Show information about ROS interfaces
launch    Run a launch file
lifecycle Various lifecycle related sub-commands
multicast Various multicast related sub-commands
node      Various node related sub-commands
param     Various param related sub-commands
pkg       Various package related sub-commands
run       Run a package specific executable
security  Various security related sub-commands
service   Various service related sub--commands
topic    Various topic related sub-commands
wtf      Use 'wtf' as alias to 'doctor'

Call `ros2 <command> -h` for more detailed usage.
rosuser@rosuser-OptiPlex-7000:~$ mkdir -p ~/bushralama_ws/src
mkdir: command not found
rosuser@rosuser-OptiPlex-7000:~$ ^
rosuser@rosuser-OptiPlex-7000:~$ mkdir -p ~/bushralama_ws/src
rosuser@rosuser-OptiPlex-7000:~$ cd bushralama_ws/src
rosuser@rosuser-OptiPlex-7000:~/bushralama_ws/src$
```

References

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

[Documentation - ROS Wiki](#)