**Instructions:**

Copy the code in red. Make sure every line gets copies over.

**Step 1: Install ROS2 Foxy**

1. **Set up your locale:**

Copy code

```
sudo locale-gen en_US en_US.UTF-8

sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8

export LANG=en_US.UTF-8
```

2. **Add the ROS2 apt repository:**

Copy code

```
sudo apt update && sudo apt install -y curl gnupg lsb-release

sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add
-

sudo sh -c 'echo "deb [arch=amd64,arm64,armhf] http://packages.ros.org/ros2/ubuntu
$(lsb_release -cs) main" > /etc/apt/sources.list.d/ros2-latest.list'
```

3. **Install ROS2 Foxy:**

Copy code

```
sudo apt update

sudo apt install ros-foxy-desktop
```

4. **Install development tools and dependencies:**

Copy code

```
sudo apt install -y python3-colcon-common-extensions python3-rosdep git
```

5. **Initialize rosdep:**

Copy code

```
sudo rosdep init

rosdep update
```

6. **Source ROS2 Foxy:**

Copy code

source /opt/ros/foxy/setup.bash

**Step 2: Install and Set Up rplidar_ros**

1. **Create a ROS2 workspace:**

Copy code

mkdir -p ~/ros2_ws/src

cd ~/ros2_ws/src

2. **Clone the rplidar_ros package:**

Copy code

git clone -b ros2 https://github.com/Slamtec/rplidar_ros.git

3. **Build the workspace:**

Copy code

cd ~/ros2_ws

colcon build --packages-select rplidar_ros

4. **Source the workspace:**

Copy code

source ~/ros2_ws/install/setup.bash

**Step 3: Add Your Custom obstacle_avoidance Package**

1. **Create a new ROS2 Python package for obstacle avoidance:**

Copy code

cd ~/ros2_ws/src

ros2 pkg create --build-type ament_python obstacle_avoidance --dependencies rclpy sensor_msgs

2. **Create the obstacle_avoidance_node.py script:**

Copy code

mkdir ~/ros2_ws/src/obstacle_avoidance/obstacle_avoidance

touch ~/ros2_ws/src/obstacle_avoidance/obstacle_avoidance/obstacle_avoidance_node.py

3. **Add the obstacle avoidance script to the Python file:**

Open the obstacle_avoidance_node.py file:

Copy code

nano ~/ros2_ws/src/obstacle_avoidance/obstacle_avoidance/obstacle_avoidance_node.py

Then paste your obstacle avoidance script:

python

Copy code

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

import rclpy
from rclpy.node import Node
from sensor_msgs.msg import LaserScan

class ObstacleAvoidanceNode(Node):
    def __init__(self):
        super().__init__('obstacle_avoidance_node')

        # Create a subscriber to the LaserScan topic
        self.scan_subscriber = self.create_subscription(
            LaserScan,
            '/scan',  # This is the topic RPLIDAR publishes to
            self.scan_callback,
            10
        )

        # Set your threshold for obstacle avoidance in meters
        self.distance_threshold = 1.0  # Example: 1 meter
        self.get_logger().info('Obstacle Avoidance Node Started')

    def scan_callback(self, msg):
        # Find the minimum distance from the laser scan data
        min_distance = min(msg.ranges)
```

```
        # Print the minimum distance
        self.get_logger().info(f'Minimum distance: {min_distance:.2f} meters')

        # Check if there is an obstacle within the threshold
        if min_distance < self.distance_threshold:
            self.get_logger().warn('Obstacle detected! Turning around...')
            # Insert logic here to turn the robot around
            # For example, send commands to the robot's motors

def main(args=None):
    rclpy.init(args=args)
    node = ObstacleAvoidanceNode()
    rclpy.spin(node)

    # Shutdown
    node.destroy_node()
    rclpy.shutdown()

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

4. **Make the Python script executable:**

Copy code

```
chmod +x ~/ros2_ws/src/obstacle_avoidance/obstacle_avoidance/obstacle_avoidance_node.py
```

5. **Create the setup.py file for your Python package:**

Open the setup.py file:

Copy code

```
nano ~/ros2_ws/src/obstacle_avoidance/setup.py
```

Add the following content:

python

Copy code

```
from setuptools import setup

package_name = 'obstacle_avoidance'

setup(
    name=package_name,
    version='0.0.0',
    packages=[package_name],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='your_name',
    maintainer_email='your_email@todo.todo',
    description='A simple obstacle avoidance node for RPLIDAR',
    license='Apache License 2.0',
    entry_points={
        'console_scripts': [
            'obstacle_avoidance_node = obstacle_avoidance.obstacle_avoidance_node:main',
        ],
    },
)
```

6. **Build the package:**

Copy code

```
cd ~/ros2_ws
colcon build
```

7. **Source the workspace:**

Copy code

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

**Step 4: Launch the RPLIDAR and Run the Obstacle Avoidance Node**

1. **Check for the LIDAR device (should show /dev/ttyUSB0):**

Copy code

```
ls /dev/ttyUSB*
```

2. **Set permissions for the LIDAR device:**

Copy code

```
sudo chmod 666 /dev/ttyUSB0
```

3. **Launch the RPLIDAR node:**

Copy code

```
ros2 launch rplidar_ros rplidar_launch.py
```

4. **Run the obstacle avoidance node:**

Copy code

```
ros2 run obstacle_avoidance obstacle_avoidance_node
```

**In another terminal:**

**Steps to Source the Workspace and Launch the Node:**

1. **Source ROS2 Foxy Environment**: Before doing anything, you need to source your ROS2 Foxy installation. Run this command in your terminal:

Copy code

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

2. **Source Your Workspace**: After building your workspace, you need to source it so the system knows where your packages and launch files are. Run the following command:

Copy code

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

3. **Check if the Launch File Exists**: Verify whether the launch file exists in the rplidar_ros package. You can list the contents of the launch folder:

Copy code

```
ls ~/ros2_ws/install/rplidar_ros/share/rplidar_ros/launch
```

If the rplidar_launch.py file is missing, you may need to create it.

4. **Create the rplidar_launch.py File** (if missing):

If the launch file does not exist, you can manually create it:

Copy code

```
nano ~/ros2_ws/src/rplidar_ros/launch/rplidar_launch.py
```

Then, paste the following content into the file:

python

Copy code

```
from launch import LaunchDescription
from launch_ros.actions import Node

def generate_launch_description():
    return LaunchDescription([
        Node(
            package='rplidar_ros',
            executable='rplidar_node',
            name='rplidar_node',
            output='screen',
            parameters=[{
                'serial_port': '/dev/ttyUSB0',
                'serial_baudrate': 115200,  # Change if your device requires a different baudrate
                'frame_id': 'laser_frame',
                'inverted': False,
                'angle_compensate': True
            }],
        )
    ])
```

5. **Rebuild the Workspace**:

After making any changes (like adding a launch file), you need to rebuild your workspace:

Copy code

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

6. **Source the Workspace Again**:

Every time you rebuild the workspace, remember to source it again:

Copy code

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

7. **Launch the RPLIDAR Node**:

Once everything is sourced and built correctly, you should be able to launch the node with the following command:

Copy code

```
ros2 launch rplidar_ros rplidar_launch.py
```