# **🛠️ SLAM Setup Tutorial: Unitree L2 LiDAR + RealSense D435i (ROS 2)**

This guide documents **every step** to replicate a working SLAM setup that combines:

* 3D LiDAR data from **Unitree L2**
* IMU + RGBD data from **Intel RealSense D435i**
* Odometry fusion via a custom ROS 2 node

Designed for beginners and advanced users alike.

## **🔧 System Requirements**

* Ubuntu 20.04
* ROS 2 Humble installed
* Internet access for cloning packages
* USB 3.0 port for RealSense
* Ethernet for Unitree L2

## **📁 Workspace Setup**

bash

CopyEdit

mkdir -p ~/combined\_ros2\_ws/src

cd ~/combined\_ros2\_ws/src

## **🔌 Step 1: LiDAR Setup**

### **1. Clone the Unitree L2 ROS2 driver**

bash

CopyEdit

git clone https://github.com/unitreerobotics/unilidar\_sdk2.git

### **2. Configure LiDAR to UDP mode**

bash

CopyEdit

cd /unilidar\_sdk2/unitree\_lidar\_sdk/examples

./set\_to\_udp\_mode

./set\_ip\_address

Make sure your LiDAR IP is set to 192.168.1.4 and your PC to 192.168.1.3.

## **📷 Step 2: RealSense Camera Setup**

bash

CopyEdit

cd ~/combined\_ros2\_ws/src

git clone -b ros2-master https://github.com/IntelRealSense/realsense-ros.git

You will primarily use:

* /camera/camera/imu
* /camera/camera/depth/color/points

To launch:

bash

CopyEdit

source /opt/ros/humble/setup.bash

ros2 launch realsense2\_camera rs\_launch.py pointcloud.enable:=true enable\_gyro:=true enable\_accel:=true unite\_imu\_method:=2

## **🧠 Step 3: Point-LIO SLAM for LiDAR**

bash

CopyEdit

cd ~/combined\_ros2\_ws/src

git clone https://github.com/dfloreaa/point\_lio\_ros2.git

Then:

bash

CopyEdit

cd ~/combined\_ros2\_ws

rosdep install --from-paths src --ignore-src -r -y

colcon build --symlink-install

source install/setup.bash

To launch Point-LIO SLAM:

bash

CopyEdit

ros2 launch point\_lio mapping\_unilidar\_l2.launch.py

## **📦 Step 4: Odometry Fusion Node**

### **1. Create Package**

bash

CopyEdit

cd ~/combined\_ros2\_ws/src

ros2 pkg create --build-type ament\_python odometry\_fusion

### **2. Add Node File**

Create: ~/combined\_ros2\_ws/src/odometry\_fusion/odometry\_fusion/odometry\_fusion\_node.py

python

CopyEdit

import rclpy

from rclpy.node import Node

from nav\_msgs.msg import Odometry

from sensor\_msgs.msg import Imu

class OdometryFusionNode(Node):

def \_\_init\_\_(self):

super().\_\_init\_\_('odometry\_fusion\_node')

self.sub\_odom = self.create\_subscription(Odometry, '/odom\_lidar', self.odom\_callback, 10)

self.sub\_imu = self.create\_subscription(Imu, '/imu\_camera', self.imu\_callback, 10)

self.pub\_fused = self.create\_publisher(Odometry, '/fused\_odometry', 10)

self.latest\_odom = None

self.latest\_imu = None

def odom\_callback(self, msg):

self.latest\_odom = msg

self.publish\_fused()

def imu\_callback(self, msg):

self.latest\_imu = msg

self.publish\_fused()

def publish\_fused(self):

if self.latest\_odom and self.latest\_imu:

fused = self.latest\_odom

fused.header.stamp = self.latest\_imu.header.stamp

fused.twist.twist.angular = self.latest\_imu.angular\_velocity

fused.twist.twist.linear = self.latest\_odom.twist.twist.linear

self.pub\_fused.publish(fused)

def main():

rclpy.init()

node = OdometryFusionNode()

rclpy.spin(node)

node.destroy\_node()

rclpy.shutdown()

if \_\_name\_\_ == '\_\_main\_\_':

main()

### **3. Update setup.py**

python

CopyEdit

entry\_points={

'console\_scripts': [

'odometry\_fusion\_node = odometry\_fusion.odometry\_fusion\_node:main'

],

}

### **4. Add Launch File**

Create: ~/combined\_ros2\_ws/src/odometry\_fusion/launch/odometry\_fusion\_launch.py

python

CopyEdit

from launch import LaunchDescription

from launch\_ros.actions import Node

def generate\_launch\_description():

return LaunchDescription([

Node(

package='odometry\_fusion',

executable='odometry\_fusion\_node',

name='odometry\_fusion\_node',

output='screen',

remappings=[

('/odom\_lidar', '/Odometry'),

('/imu\_camera', '/camera/camera/imu'),

('/fused\_odom', '/fused\_odometry')

]

)

])

### **5. Build the Workspace**

bash

CopyEdit

cd ~/combined\_ros2\_ws

colcon build --symlink-install

source install/setup.bash

## **🧪 Step 5: Run All Nodes in Separate Terminals**

* **Terminal 1**: LiDAR ROS2 driver

bash

CopyEdit

ros2 launch unitree\_lidar\_ros2 launch.py

* **Terminal 2**: RealSense camera

bash

CopyEdit

ros2 launch realsense2\_camera rs\_launch.py pointcloud.enable:=true enable\_gyro:=true enable\_accel:=true unite\_imu\_method:=2

* **Terminal 3**: Point-LIO SLAM

bash

CopyEdit

ros2 launch point\_lio mapping\_unilidar\_l2.launch.py

* **Terminal 4**: Odometry fusion

bash

CopyEdit

ros2 launch odometry\_fusion odometry\_fusion\_launch.py

* **Terminal 5**: RViz2 to visualize /fused\_odometry and /unilidar/cloud

bash

CopyEdit

rviz2

Set:

* Fixed Frame: camera\_link
* Add:  
  + PointCloud2 (topic: /unilidar/cloud)
  + Odometry (topic: /fused\_odometry)

## **✅ Verification**

Run:

bash

CopyEdit

ros2 topic echo /fused\_odometry

ros2 run tf2\_tools view\_frames

You should see the fused odometry stream and a connected TF tree with camera\_link, unilidar\_lidar, and aft\_mapped.

## **📄 Downloadable Version**

A downloadable README.md file is provided below:

**👉 Download README.md**

Let me know if you want this turned into a PDF, hosted on GitHub, or converted into a step-by-step bash script.