# **✅ Unitree L2 LiDAR Setup Guide for UDP over Ethernet**

This guide outlines the **correct procedure to configure, compile, and test** Unitree L2 LiDAR in **UDP mode over Ethernet** using the Unitree SDK and ROS2.

## **🧾 System Setup Summary**

* **LiDAR IP**: 192.168.1.4
* **PC IP**: 192.168.1.3
* **Serial Port**: /dev/ttyACM0
* **Baudrate**: 4000000
* **UDP Mode WorkMode**: 0
* **Serial Mode WorkMode**: 8

## **🧰 Initial One-Time Setup (First-Time Users)**

### **1. Clone and Build the SDK and ROS2 Package**

mkdir -p ~/unilidar\_sdk2 && cd ~/unilidar\_sdk2

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

cd unilidar\_sdk2/unitree\_lidar\_sdk

mkdir build && cd build

cmake ..

make -j4

### **2. Verify and Edit Example Files (in unitree\_lidar\_sdk/examples)**

See section "Edit Required Example Files" below to apply the necessary edits.

### **3. Set LiDAR IP Address over Serial**

cd ~/unilidar\_sdk2/unitree\_lidar\_sdk/build

../bin/set\_ip\_address

Reboot the LiDAR after this step.

### **4. Set the LiDAR to UDP Mode**

../bin/set\_to\_udp\_mode

### **5. Configure PC Ethernet Interface**

sudo ip addr add 192.168.1.3/24 dev enp0s31f6

sudo ip link set dev enp0s31f6 up

//sudo ip addr add 192.168.1.3/24 dev eth0

### **6. Test Ping and UDP Example**

ping 192.168.1.4

../bin/example\_lidar\_udp

### **7. Build ROS2 Package**

cd ~/unilidar\_sdk2/unitree\_lidar\_ros2

colcon build --packages-select unitree\_lidar\_ros2

source install/setup.bash

## **🚀 Daily Use Instructions (After Initial Setup)**

Follow these steps each time you boot the computer and want to use the LiDAR:

# Set PC Ethernet IP (if not persistent)

sudo ip addr add 192.168.1.3/24 dev enp0s31f6

sudo ip link set dev enp0s31f6 up

# Ping to confirm LiDAR connectivity

ping 192.168.1.4

# Source and run ROS2 node

cd ~/unilidar\_sdk2/unitree\_lidar\_ros2

source install/setup.bash

ros2 launch unitree\_lidar\_ros2 launch.py

# In another terminal (optional):

ros2 topic echo /unilidar/cloud

## **🛠️ Edit Required Example Files (in unitree\_lidar\_sdk/examples)**

### **1. set\_to\_udp\_mode.cpp ✅**

No changes needed:

initializeSerial("/dev/ttyACM0", 4000000);

setLidarWorkMode(0); // UDP

### **2. set\_to\_serial\_mode.cpp ❌**

Update IPs:

std::string lidar\_ip = "192.168.1.4";

std::string local\_ip = "192.168.1.3";

### **3. set\_ip\_address.cpp ❌**

* Replace UDP init with Serial:

lreader->initializeSerial("/dev/ttyACM0", 4000000);

* Update IP config:

config.lidar\_ip = {192, 168, 1, 4};

config.user\_ip = {192, 168, 1, 3};

### **4. example\_lidar\_udp.cpp ✅**

Already correct:

lidar\_ip = "192.168.1.4";

local\_ip = "192.168.1.3";

workMode = 0;

### **5. example\_lidar\_serial.cpp ✅**

Already correct:

initializeSerial("/dev/ttyACM0", 4000000);

workMode = 8;

## **🔧 Build Instructions**

cd ~/unilidar\_sdk2/unitree\_lidar\_sdk/build

cmake ..

make -j4

## **🧩 ROS2 Launch Setup**

### **launch.py Parameter Block**

parameters=[

{'initialize\_type': 2},

{'work\_mode': 0},

{'use\_system\_timestamp': True},

{'range\_min': 0.0},

{'range\_max': 100.0},

{'cloud\_scan\_num': 18},

{'lidar\_ip': '192.168.1.4'},

{'local\_ip': '192.168.1.3'},

{'lidar\_port': 6101},

{'local\_port': 6201},

{'cloud\_frame': 'unilidar\_lidar'},

{'cloud\_topic': 'unilidar/cloud'},

{'imu\_frame': 'unilidar\_imu'},

{'imu\_topic': 'unilidar/imu'},

]

❗ Remove serial\_port or baudrate unless using serial mode.

### **✅ Optional Dual-Mode Setup**

use\_udp = True

params = [

{'initialize\_type': 2 if use\_udp else 1},

{'work\_mode': 0 if use\_udp else 8},

{'use\_system\_timestamp': True},

{'range\_min': 0.0},

{'range\_max': 100.0},

{'cloud\_scan\_num': 18},

]

if use\_udp:

params += [

{'lidar\_ip': '192.168.1.4'},

{'local\_ip': '192.168.1.3'},

{'lidar\_port': 6101},

{'local\_port': 6201},

]

else:

params += [

{'serial\_port': '/dev/ttyACM0'},

{'baudrate': 4000000},

]

### **🔧 Fix ROS2 Node Name Bug**

In unitree\_lidar\_ros2.h, update:

: Node("unitree\_lidar\_ros2\_node", options)

### **🔧 Pass NodeOptions in Main**

In unitree\_lidar\_ros2\_node.cpp:

rclcpp::spin(std::make\_shared<UnitreeLidarSDKNode>(options));

## **✅ Confirm Success**

* ping 192.168.1.4 returns packets
* ../bin/example\_lidar\_udp shows live stream
* ros2 topic echo /unilidar/cloud outputs point cloud data
* RViz displays /unilidar/cloud with frame unilidar\_lidar

You’re done! 🎉

Let me know if you want to automate this whole setup with a shell script.

c