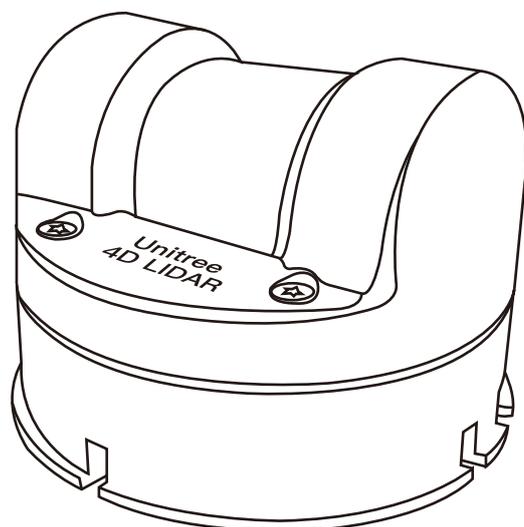


# Unilidar SDK

用户手册 v 1.0

2023.05



Unitree Lidar

# 目录

前言	2
<b>Uniliar SDK</b>	
简介	3
依赖项	3
配置	3
构建	4
运行	4
版本历史	7
<b>Uniliar ROS</b>	
简介	8
依赖项	8
配置	8
构建	9
运行	9
yml配置文件说明	10
版本历史	11
<b>Uniliar ROS2</b>	
简介	12
依赖项	12
配置	12
构建	13
运行	13
launch.py配置文件说明	14
版本历史	15

# 前言

这个代码库是 Unitree L1 LiDAR 的 SDK。

您可以使用这个代码库中提供的接口获取我们的 LiDAR 测量的点云数据和 IMU 数据,以及配置相关参数。

我们为我们的 LiDAR 提供了三个常用的接口:

- 如果您喜欢直接使用原始的 C++ SDK,可以参考 [/unitree\\_lidar\\_sdk/README.md](#);
- 如果您想使用 ROS,可以参考 [/unitree\\_lidar\\_ros/src/unitree\\_lidar\\_ros/README.md](#);
- 如果您正在使用最新的 ROS2 进行开发,可以参考 [unitree\\_lidar\\_ros2/src/unitree\\_lidar\\_ros2/README.md](#)。



# Unilidar SDK

## ·简介

Unilidar SDK 是一个使用 CMake 构建的软件包,适用于 Unitree LiDAR L1。

该软件包可以提供以下功能:

- 解析从激光雷达传输的原始数据,并将其转换为点云和 IMU 数据。
- 获取点云数据。
- 获取 IMU 数据。

默认情况下,输出的点云使用自定义数据类型,以便该 SDK 不过度依赖外部依赖项。在其他情况下,

- 如果您习惯使用 [Point Cloud Library](<https://pointclouds.org/>), 您可以使用头文件 'unitree\_lidar\_sdk\_pcl.h' 将我们的点云转换为 PCL 格式;
- 如果您希望直接使用 [ROS](<https://www.ros.org/>) 软件包, 则也可以使用我们的 ROS 软件包来处理这个激光雷达。

## ·依赖项

我们已经验证该软件包可以在以下环境下成功运行:

- Ubuntu 20.04

这个 SDK 几乎不依赖于任何外部依赖项。

但是,如果您想使用 PCL 点云格式,您需要安装一个 PCL 软件包。

## ·配置

使用 USB 线将您的激光雷达连接到计算机上,然后确认激光雷达的串口名称:

```
...  
  
$ ls /dev/ttyUSB*  
  
/dev/ttyUSB0  
  
...
```

默认的串口名称为 '/dev/ttyUSB0' 。

如果不是默认值,您需要修改 example\_lidar.cpp 中的配置参数。

## · 构建

您可以将此程序作为 cmake 项目构建：

```
...  
  
cd unitree_lidar_sdk  
mkdir build  
cd build  
cmake .. && make -j2  
...
```

其中, mkdir build 用于创建一个名为 build 的文件夹, 用于存放编译生成的文件;

cd build 用于进入 build 文件夹;

cmake .. 用于生成 Makefile 文件;

make -j2 用于编译生成可执行文件。

在编译过程中, -j2 表示开启 2 个线程进行编译, 可以加快编译速度。如果您的计算机有更多的处理器核心, 可以根据实际情况适当调整该参数。

## · 运行

直接运行可执行文件：

```
...  
  
../bin/example_lidar  
...
```

输出结果如下所示：

```
...  
  
$ ../bin/example_lidar  
lidar firmware version = 0.3.2+230511  
lidar sdk version = 1.0.3  
  
Dirty Percentage = 5.145833 %  
Dirty Percentage = 4.166667 %  
Dirty Percentage = 4.166667 %  
Turn on all the LED lights ...  
Turn off all the LED lights ...  
Set LED mode to: FORWARD_SLOW ...  
Set LED mode to: REVERSE_SLOW ...
```

Set LED mode to: SIXSTAGE\_BREATHING ...  
Set Lidar working mode to: NORMAL\_MODE ...

An IMU msg is parsed!

stamp = 1683874160.559222, id = 729  
quaternion (x, y, z, w) = [0.0131, -0.0091, 0.6888, -0.7225]

An IMU msg is parsed!

stamp = 1683874160.564979, id = 121  
quaternion (x, y, z, w) = [0.0102, -0.0093, 0.7099, -0.7018]

An IMU msg is parsed!

stamp = 1683874160.568425, id = 122  
quaternion (x, y, z, w) = [0.0118, -0.0096, 0.7099, -0.7018]

An IMU msg is parsed!

stamp = 1683874160.573472, id = 123  
quaternion (x, y, z, w) = [0.0126, -0.0093, 0.7098, -0.7018]

An IMU msg is parsed!

stamp = 1683874160.577348, id = 124  
quaternion (x, y, z, w) = [0.0128, -0.0093, 0.7099, -0.7018]

A Cloud msg is parsed!

stamp = 1683874145.535888, id = 1  
cloud size = 278

first 10 points (x,y,z,intensity,time,ring) =

(-0.029885, -0.136897, 0.000448, 88.000000, 0.000000, 0)  
(-0.035384, -0.171399, 0.005140, 91.000000, 0.000023, 0)  
(-0.043000, -0.219542, 0.012437, 127.000000, 0.000046, 0)  
(-0.054879, -0.294965, 0.024572, 132.000000, 0.000069, 0)  
(-0.055288, -0.301204, 0.033170, 106.000000, 0.000093, 0)  
(-0.054542, -0.300348, 0.041173, 101.000000, 0.000116, 0)  
(-0.053764, -0.299285, 0.049148, 99.000000, 0.000139, 0)  
(-0.056123, -0.318406, 0.060981, 92.000000, 0.000162, 0)  
(-0.055230, -0.316827, 0.069422, 91.000000, 0.000185, 0)  
(-0.051257, -0.294854, 0.072849, 128.000000, 0.000208, 0)

...  
...

在示例中,我们打印了点云消息的前 10 个点和 IMU 消息的四元数。其中各个变量的含义如下:

- ◆ Dirty Percentage: 激光雷达的脏污百分比,脏污百分比较高时请清洁雷达光学窗口
- ◆ stamp: 数据的时间戳,单位秒
- ◆ id: 数据的编号
- ◆ cloud size: 点云中点的数量
- ◆ x,y,z: 点的 x, y, z 坐标
- ◆ intensity: 点的反射强度
- ◆ time: 该点相对于点云时间戳 stamp 的时间差值
- ◆ ring: 该点所在的激光线圈编号
- ◆ quaternion (x, y, z, w): IMU 的四元数向量 [x, y, z, w]

注意事项:

- ◆ 在 Ubuntu 中,访问串口设备需要适当的权限。如果您的 C++ 程序没有足够的权限访问串口设备,您将会收到一个 "Permission denied" 错误。
- ◆ 要解决这个错误,您可以使用以下命令将当前用户添加到 dialout 组:  
...  

```
sudo usermod -a -G dialout $USER
```

  
...  
◆ 将用户添加到 dialout 组后,您需要注销并重新登录才能使更改生效。

## · 版本历史

v1.0.0(2023.05.04)

- 支持固件版本 0.3.1

v1.0.1(2023.05.05)

- 支持固件版本 0.3.1
- 增加支持设置激光雷达工作模式, 例如正常模式 'NORMAL\_MODE' 和待机模式 'STANDBY\_MODE'。
- 增加支持设置 LED 灯的模式

v1.0.2(2023.05.11)

- 支持固件版本 0.3.2

v1.0.3(2023.05.12)

- 支持固件版本 0.3.2
- 增加支持获取脏污指数

v1.0.4(2023.05.31)

- 支持固件版本 1.0.1

v1.0.5(2023.06.05)

- 支持固件版本: 1.0.1
- 更新默认的 rotate\_yaw\_bias 参数为经过校准的值 -38.5 度。
- 在 README.md 中添加注意事项, 以解决打开串口时出现的 "Permission denied" 错误。

v1.0.6(2023.06.19)

- 支持固件版本: 1.0.1
- 修改 initialize() 函数以检查指定的串口是否存在。
- 如果串口名称不存在, 则初始化失败并且 return -1 而不是意外地抛出错误。
- 将 z 偏移添加到激光雷达基准平面。

# Unilidar ROS

## ·简介

Unilidar ROS 是一个 ROS 的软件包,适用于 Unitree LiDAR L1。  
该软件包可以提供以下功能:

- 解析从激光雷达传输的原始数据,并将其转换为点云和 IMU 数据
- 获取点云数据
- 获取 IMU 数据
- 此外,我们提供了一个 yaml 文件,用于配置激光雷达的相关参数

## ·依赖项

依赖项包括 'PCL' 和 'ROS' 。

我们已经验证了该软件包可以在以下环境下成功运行:

- Ubuntu 20.04
- ROS noetic
- PCL-1.10
- unitree\_lidar\_sdk

建议您配置类似的环境来运行此软件包。

## ·配置

将激光雷达通过 USB 线连接到计算机上,然后确认激光雷达的串口名称:

```
...  
$ ls /dev/ttyUSB*  
/dev/ttyUSB0  
...
```

默认的串口名称是 '/dev/ttyUSB0'。如果不是默认的名称,您需要修改 unitree\_lidar\_ros/config/config.yaml 文件中的配置,并将 port 名称更改为您的名称。例如:

```
...  
# Serial Port  
port: "/dev/ttyYourUSBPortName"  
...
```

您可以保留配置文件中的其他参数,它们将使用其默认值。

如果您有特殊需求,例如更改点云 topic 名称或 IMU topic 名称,也可以在配置文件中配置。

默认的点云话题及其坐标系名称为:

- topic: "unilidar/cloud"
- frame: "unilidar\_lidar"

默认的 IMU 话题及其坐标系名称为：

- topic: "unilidar/imu"
- frame: "unilidar\_imu"

## • 构建

你可以按以下步骤构建这个 ROS 软件包：

- 克隆代码库

...

```
git clone https://github.com/unitreerobotics/unilidar_sdk.git
```

...

- 编译

...

```
cd unilidar_sdk/unitree_lidar_ros
catkin_make
```

...

编译后的内容将保存在 unilidar\_sdk/unitree\_lidar\_ros/devel 目录中。

## • 运行

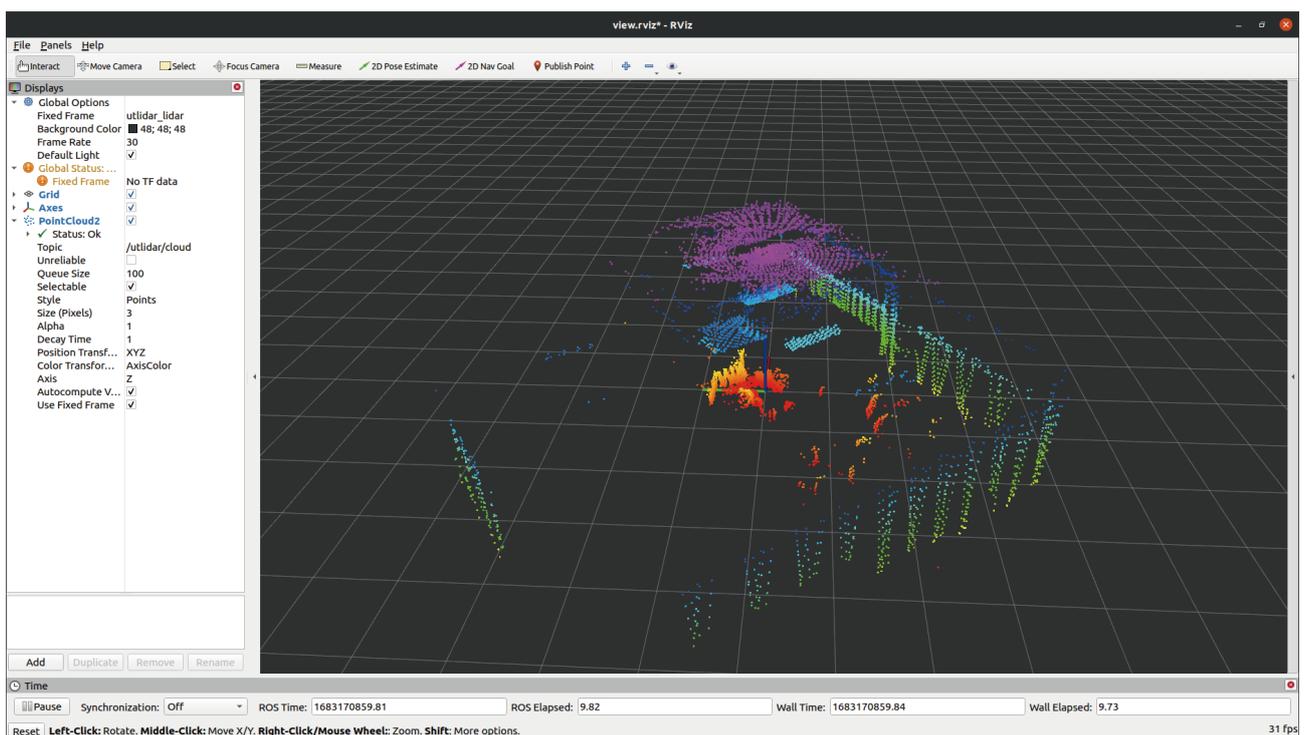
然后，您需要启动 ROS 的环境，并直接运行启动文件：

...

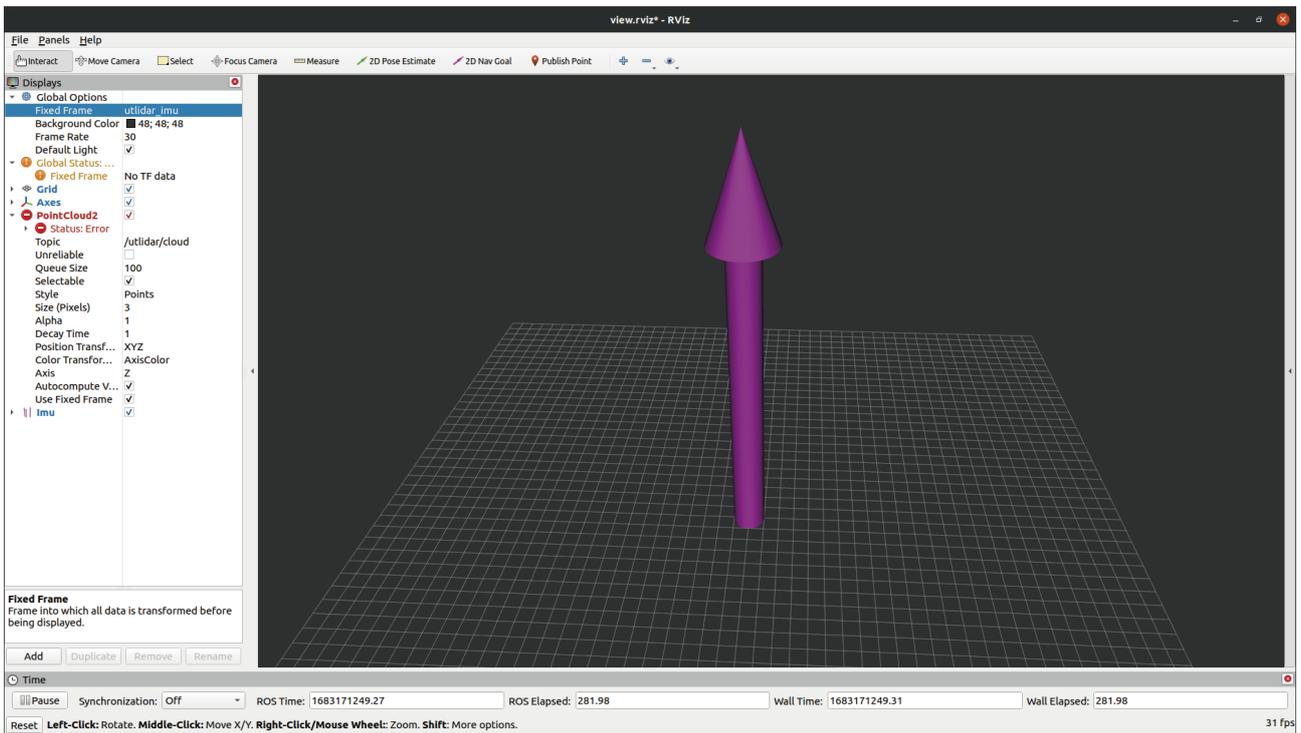
```
source devel/setup.bash
ros launch unitree_lidar_ros run.launch
```

...

在 Rviz 窗口中，您将看到我们的激光雷达点云如下所示：



您可以将 'Fixed Frame' 更改为 IMU 坐标系 'unilidar\_imu'，这样您就可以查看 IMU 四元数向量：



## · yml配置文件说明

文件位置：unitree\_lidar\_ros\config\config.yaml

- ◆ port: 激光雷达连接到电脑的串口号, 默认为 /dev/ttyUSB0。
- ◆ rotate\_yaw\_bias: 激光雷达在安装时可能会存在的固定偏移角度, 单位为度。该参数用于校准点云数据的方向, 默认为 0。
- ◆ range\_scale: 激光雷达的测距比例系数, 用来校准激光雷达的测距精度, 默认为 0.001。
- ◆ range\_bias: 激光雷达的测距偏移量, 用来校准激光雷达的测距精度, 默认为 0。
- ◆ range\_max: 激光雷达的最大测距范围, 单位为米, 默认为 50。
- ◆ range\_min: 激光雷达的最小测距范围, 单位为米, 默认为 0。
- ◆ cloud\_frame: 点云数据的坐标系名称, 默认为 unilidar\_lidar。
- ◆ cloud\_topic: 点云数据的 ROS topic 名称, 默认为 unilidar/cloud。
- ◆ cloud\_scan\_num: 点云数据的激光线圈编号数量, 默认为 18。
- ◆ imu\_frame: IMU 数据的坐标系名称, 默认为 unilidar\_imu。
- ◆ imu\_topic: IMU 数据的 ROS topic 名称, 默认为 unilidar/imu。
- ◆

## · 版本历史

v1.0.0 (2023.05.04)

- ◆ 支持 Unilidar\_sdk 版本: v1.0.0

v1.0.1 (2023.05.05)

- ◆ 支持 Unilidar\_sdk 版本: v1.0.1

v1.0.2 (2023.05.12)

- ◆ 支持 Unilidar\_sdk 版本: v1.0.3

v1.0.3 (2023.05.31)

- ◆ 支持 Unilidar\_sdk 版本: v1.0.4

# Unilidar ROS2

## ·简介

Unilidar ROS2 是一个 ROS2 的软件包,适用于 Unitree LiDAR L1。  
该软件包可以提供以下功能:

- 解析从激光雷达传输的原始数据,并将其转换为点云和 IMU 数据
- 将点云数据发布到 ROS2 topic
- 将 IMU 数据发布到 ROS2 topic
- 此外,您可以在 launch.py 中配置 LiDAR 的相关参数。

## ·依赖项

依赖项包括 'PCL' 和 'ROS2' 。

我们已经验证了该软件包可以在以下环境下成功运行:

- Ubuntu 20.04
- ROS2 foxy
- PCL-1.10
- unitree\_lidar\_sdk

建议您配置类似的环境来运行此软件包。

## ·配置

将激光雷达通过 USB 线连接到计算机上,然后确认激光雷达的串口名称:

```
...  
$ ls /dev/ttyUSB*  
/dev/ttyUSB0  
...
```

默认的串口名称是 '/dev/ttyUSB0'。如果不是默认的名称,您需要修改 unitree\_lidar\_ros2/launch/launch.py 文件中的配置,并将 port 名称更改为您的名称。例如:

```
...  
{'port': '/dev/ttyUSB0'},  
...
```

您可以保留配置文件中的其他参数,它们将使用其默认值。

如果您有特殊需求,例如更改点云 topic 名称或 IMU topic 名称,也可以在配置文件中进行配置。

默认的点云话题及其坐标系名称为:

- topic: "unilidar/cloud"
- frame: "unilidar\_lidar"

默认的 IMU 话题及其坐标系名称为：

- topic: "unilidar/imu"
- frame: "unilidar\_imu"

## · 构建

你可以按以下步骤构建这个 ROS2 软件包：

- 克隆代码库

```
...  
git clone https://github.com/unitreerobotics/unilidar_sdk.git  
...
```

- 编译

```
...  
cd unilidar_sdk/unitree_lidar_ros2  
catkin_make  
...
```

编译后的内容将保存在 unilidar\_sdk/unitree\_lidar\_ros2/devel 目录中。

## · 运行

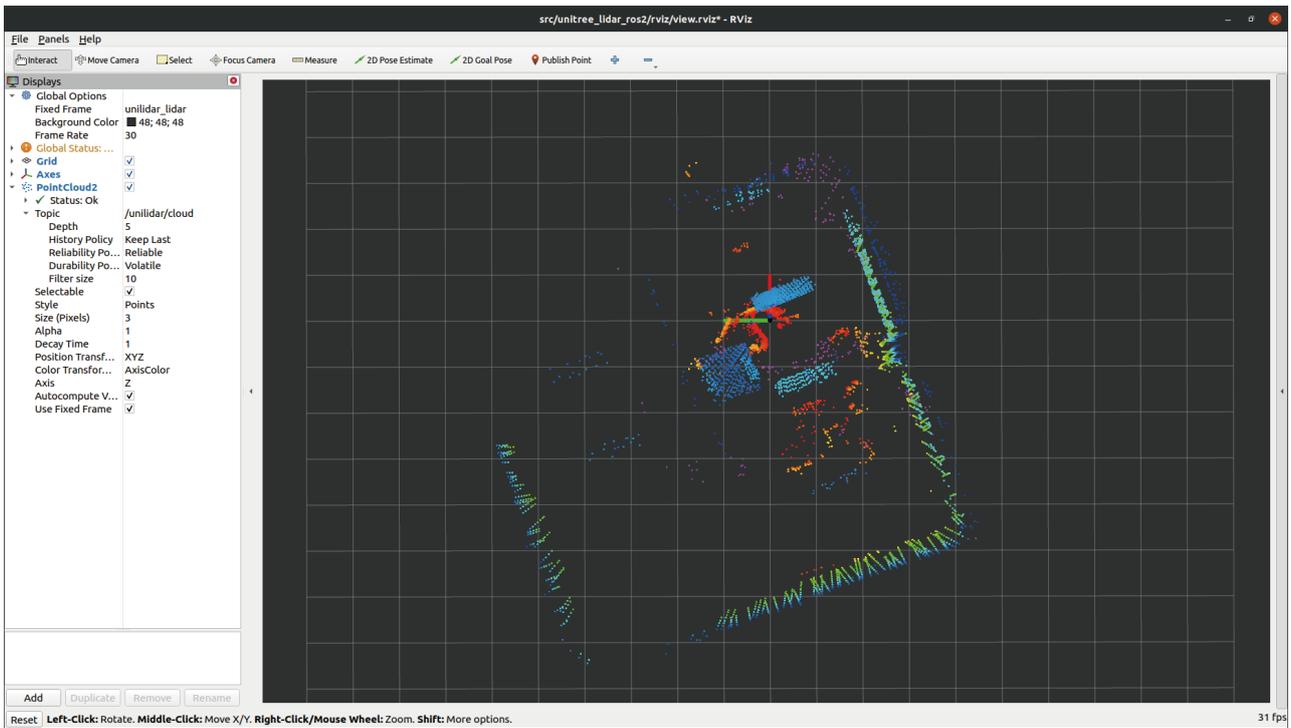
然后，您需要启动 ROS2 的环境，并直接运行启动文件：

```
...  
source install/setup.bash  
ros2 launch unitree_lidar_ros2 launch.py  
...
```

如果您想在 Rviz2 中查看我们的点云数据，需要运行以下命令：

```
...  
rviz2 -d src/unitree_lidar_ros2/rviz/view.rviz  
...
```

在 Rviz 窗口中，您将看到我们的激光雷达点云如下所示：



您可以将 'Fixed Frame' 更改为 IMU 坐标系 'unilidar\_imu'，这样您就可以查看 IMU 四元数向量。

## • launch.py 配置文件说明

文件位置：unitree\_lidar\_ros2\launch\launch.py

- ◆ port: 激光雷达连接到电脑的串口号, 默认为 /dev/ttyUSB0。
- ◆ rotate\_yaw\_bias: 激光雷达在安装时可能会存在的固定偏移角度, 单位为度。该参数用于校准点云数据的方向, 默认为 0。
- ◆ range\_scale: 激光雷达的测距比例系数, 用来校准激光雷达的测距精度, 默认为 0.001。
- range\_bias: 激光雷达的测距偏移量, 用来校准激光雷达的测距精度, 默认为 0。
- ◆ range\_max: 激光雷达的最大测距范围, 单位为米, 默认为 50。
- ◆ range\_min: 激光雷达的最小测距范围, 单位为米, 默认为 0。
- ◆ cloud\_frame: 点云数据的坐标系名称, 默认为 unilidar\_lidar。
- ◆ cloud\_topic: 点云数据的 ROS topic 名称, 默认为 unilidar/cloud。
- ◆ cloud\_scan\_num: 点云数据的激光线圈编号数量, 默认为 18。
- ◆ imu\_frame: IMU 数据的坐标系名称, 默认为 unilidar\_imu。
- ◆ imu\_topic: IMU 数据的 ROS topic 名称, 默认为 unilidar/imu。
- ◆

## · 版本历史

v1.0.0(2023.05.31)

- ◆ 支持 Unilidar\_sdk 版本: v1.0.4

本手册如有更新,恕不另行通知。

您可以再 Unitree 官方网站查询最新版本《Unilidar SDK 用户手册》



<https://www.unitree.com/download>

Unitree 是杭州宇树科技有限公司的商标。