**Brief overview**

Light Detection and Ranging (LiDAR) operates in the same way as ultrasonic rangefinders with laser pulse is used instead of sound waves. Yandex, Uber, Waymo and etc. are investing heavily in LiDAR technology for their autonomous car programs. The most critical drawback of LiDAR sensors is their high-cost. However, there are an increasing number of low‐cost options that are already on the market. An example of such is the RPLiDAR A1M8 developed by Slamtec with its 360 degree 2D laser scanner (LIDAR) solution. It can perform 360 degree scan within 12-meter range and take up to 8, 000 samples per second. And it is available for just Rs.8,000

[RPLIDAR](http://www.slamtec.com/en/lidar/a1)is a low-cost LIDAR sensor suitable for indoor robotic SLAM(Simultaneous localization and mapping) application. It can be used in the other applications such as:

* General robot navigation and localization
* Obstacle avoidance
* Environment scanning and 3D modeling

The aim of this tutorial is to use the Robot Operating System (ROS) on a NVIDIA Jetson Nano Developer Kit to test the performance of the low‐cost RPLiDAR A1M8 by Slamtec in the SLAM problem.

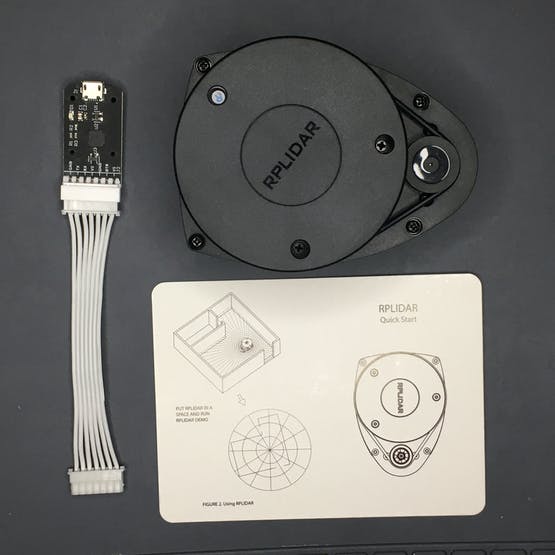
**Unboxing the RPLIDAR A1 Development Kit**



1 / 4

RPLIDAR A1 Development Kit contains:

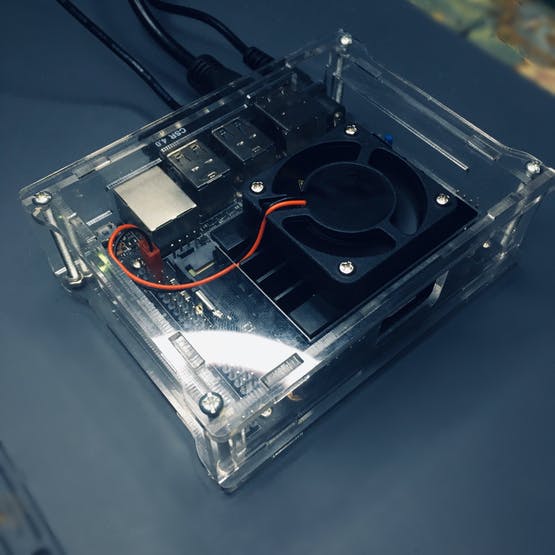
* RPLIDAR A1
* USB Adapter with communication cable
* Documentation



Note: The Micro-USB cable does not included.

**NVIDIA Jetson Nano Developer Kit**

NVIDIA Jetson Nano is a small, powerful and low‐cost single board computer that is capable of almost anything a standalone PC is capable of. It is powered by a 1.4-GHz quad-core ARM A57 CPU, 128-core Nvidia Maxwell GPU and 4 GB of RAM and also has the power to run ROS when running a Linux operating system.



**Preparation**

Make sure you have the latest version of the JetPack. You can download the latest version from the official [website of Nvidia.](https://developer.nvidia.com/embedded/downloads) I’ve already published a quick start [guide](https://www.hackster.io/shahizat005/getting-started-with-the-nvidia-jetson-nano-developer-kit-b350a8)recently. Check it out.

After installing the OS, we will check whether the latest drivers are installed with the following commands.

sudo apt-get update

This command updates the list of available packages and their versions.

sudo apt-get upgrade

Plug in the RPlidar to USB port of your NVIDIA Jetson Nano via USB Adapter with communication cable.

Open your terminal and run the following command.

ls -l /dev | grep ttyUSB

Output of the following command must be:

crw-rw---- 1 root dialout 188, 0 Dec 31 20:33 ttyUSB0

Run below command to change permission:

sudo chmod 666 /dev/ttyUSB0

Now you are able to read and write with this device using the port. Verify it via **ls -l /dev | grep ttyUSB** command.

crw-rw-rw- 1 root dialout 188, 0 Dec 31 20:33 ttyUSB0

**Installation of ROS on Jetson Nano**

Now, we are ready to install the ROS packages on Ubuntu 18.04 LTS based on Jetson Nano. Set up the Jetson Nano to accept software from [packages.ros.org](http://packages.ros.org/) by entering the following command on terminal:

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

Add a new apt key:

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

And you will see the following output:

Executing: /tmp/apt-key-gpghome.kbHNkEyTKo/gpg.1.sh --keyserver hkp://keyserver.ubuntu.com:80 --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654  
gpg: key F42ED6FBAB17C654: public key "Open Robotics <info@osrfoundation.org>" imported  
gpg: Total number processed: 1  
gpg: imported: 1

Update your packages list by the following command:

sudo apt update

Currently, the latest version of ROS is [**Melodic Morenia**](https://www.ros.org/). The command below installs all the software, tools, algorithms, and robot simulators for ROS, including support for **rqt**, **rviz** and other useful robotics packages. After you type the command and press Enter, press**Y** and hit Enter when asked if you want to continue.

sudo apt install ros-melodic-desktop

It lasts for about 15-20 minutes to download and finish executing a command, so feel free to take a break.

Now initialize rosdep.

sudo rosdep init

You will see the following output:

Wrote /etc/ros/rosdep/sources.list.d/20-default.list  
Recommended: please run  
rosdep update

Then run below command

rosdep update

You may see the following error on terminal:

ERROR: error loading sources list:  
<urlopen error <urlopen error ('\_ssl.c:711: The handshake operation timed out',)> (https://raw.githubusercontent.com/ros/rosdistro/master/dashing/distribution.yaml)>

Run again**rosdep update** till the error will disappear. In my case it was done 2 times.

Set up the environment variables

$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc   
$ source ~/.bashrc

Here is the last step of the installation process. Check which version of ROS you have installed. If you see your ROS version as the output, congratulations you have successfully installed ROS.

rosversion -d

In my case it was:

melodic

Now the Jetson Nano is ready to execute ROS packages.

**Configure a catkin workspace**

You must create and configure a **catkin** workspace. A catkin workspace is a directory in which you can create or modify existing catkin packages.

Install the following dependencies:

sudo apt-get install cmake python-catkin-pkg python-empy python-nose python-setuptools libgtest-dev python-rosinstall python-rosinstall-generator python-wstool build-essential git

Create the catkin root and source folders:

mkdir -p ~/catkin\_ws/src

In your terminal, run

cd ~/catkin\_ws/src

Clone the github repository of RPLIDAR ROS package.

git clone https://github.com/robopeak/rplidar\_ros.git

Run

cd ..

Then, run **catkin\_make** to compile your catkin workspace.

catkin\_make

Then run to source the environment with your current terminal.**Don't close the terminal.**

source devel/setup.bash

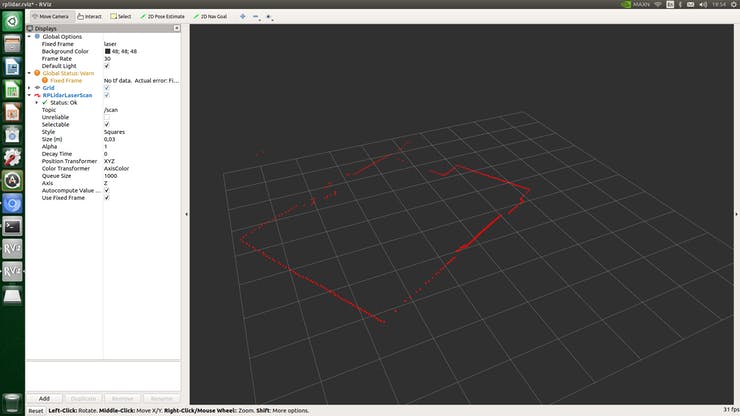
In a new terminal, run the following command

roscore

In the terminal which you sourced the environment, run below command

roslaunch rplidar\_ros view\_rplidar.launch

An instance of **Rviz** will then open with a map of the RPLIDAR’s surroundings.



Scan result in rviz

ROS is a good framework in which we made the map around the RPLIDAR. It is a great tool for build robot software systems which can be useful to a variety of hardware platforms, research settings, and runtime requirements. This work served to prove that the low‐cost RPLiDAR is a suitable solution for implementing SLAM.