# Installing RPLiDAR and Hector\_slam package on the Raspberry Pi

## Reference:

First of all, I want to thank user Tiziano Fiorenzani on YouTube, the creators of Robopeak on Github, the Slamtech team on the ROS wiki and StefanFabian on Github. These persons gave me the information to make this tutorial and I could not have made it without there information. This guide follows for the most part what is said in the YouTube video of mister Fiorenzani, but to been altered as some libraries were missing. At the end of the guide you will find links to the different sites that were used to make this guide.

## Instructions:

### Installing and using RPLiDAR package

First let’s connect the RPLiDAR to the Raspberry Pi.

Now we are going to check what the authority of the RPLiDAR’s serial-port is: [1] [2]

$ ls -l /dev |grep ttyUSB

Let’s add the authority that it can write: (such as /dev/ttyUSB0) [1] [2]

$ sudo chmod 666 /dev/ttyUSB0

Now we are going to clone the RPLiDAR package into our work space. [2]

$ cd ~/catkin\_ws/scr

$ git clone <https://github.com/Slamtec/rplidar_ros.git>

Now that that’s done we are going to make or package. [2]

$ cd ..

$ catkin\_make

If everything goes as planned you should be able to use the RPLiDAR nodes and launch files. Before that let’s first source our setup.bash file. [2]

**Note:** if you have not sourced the setup file of the work space in the .bashrc file, then you have to source the setup file manually. This must be done every time again with the following command.

$ source devel/setup.bash

**Attention:** If you make a new work space you can best use the following command after you have built the workspace.

echo "source $HOME/catkin\_ws/devel/setup.bash" >> ~/.bashrc

this command ensures that the setup .bash file is sourced when you open a new terminal and you do not have to do it yourself every time you open a new terminal. If you have already done this you do not have to do it again. You only have to do it once.

Now let’s try to use the RPLiDAR with the Raspberry Pi.

You first need to find which RPLiDAR you are working with. This will influence which launch files you will be using.

For the RPLiDAR A1 and A2 series use: [1] [2]

$ roslaunch rplidar\_ros view\_rplidar.launch

For the RPLiDAR A3 series use:

$ roslaunch rplidar\_ros view\_rplidar\_a3.launch

This should open a new window named RViz.

On the grey plain you should see red dots. These display the data point acquired by the RPLiDAR.

### Installing and using Hector\_slam

Before we are going to install Hector\_slam we are first going to install Qt 4 [3]

$ sudo apt install qt4-default

**Note:** When I installed Hector\_slam there was an error that pointed out that it needed Qt 4.

Now let’s install the Hector\_slam package. [2]

$ cd ~/catkin\_ws/scr

$ git clone https://github.com/tu-darmstadt-ros-pkg/hector\_slam.git

Now we will need to edit some files. This is because the Hector\_slam is made to work with a robot car.

In the following link you will find the changes that gave to be done but I will also put them here to make the guide more complete.

**Note:** it is a good idea to just comment out the code that you are replacing as you may want to change it in the future.

The first file to edit is mapping\_default.launch. This file is located at: [2] [4]

catkin\_ws/src/rplidar\_hector\_slam/hector\_slam/hector\_mapping/launch/mapping\_default.launch

In this file replace the second last line with [2] [4]

<node pkg="tf" type="static\_transform\_publisher" name="base\_to\_laser\_broadcaster" args="0 0 0 0 0 0 base\_link laser 100" />

The third line with [2] [4]

<arg name="base\_frame" default="base\_link"/>

And the fourth line with [2] [4]

<arg name="odom\_frame" default="base\_link"/>

After that save the file. [2] [4]

Now we are going to the tutorial.launch file. This file is located at: [2] [4]

catkin\_ws/src/rplidar\_hector\_slam/hector\_slam/hector\_slam\_launch/launch/tutorial.launch

Here we are going to replace the third line with [2] [4]

<param name="/use\_sim\_time" value="false"/>

Now let’s try Hector\_slam.

For this we will need to start 2 launch files

The first program is from the RPLiDAR\_ros package. This program will collect the raw data of the RPLiDAR and convert it to readable data and publish it on the /scan topic.

For this part we again need to check which RPLiDAR we are working with. This will influence which launch files we will be using.

For the RPLiDAR A1 and A2 series use: [2]

$ roslaunch rplidar\_ros rplidar.launch

For the RPLiDAR A3 series use:

$ roslaunch rplidar\_ros rplidar\_a3.launch

The second program is from the Hector\_slam\_launch package. This program will collect the data from the /scan topic and implement simultaneous localization and mapping using the data it receives. [2]

$ roslaunch hector\_slam\_launch tutorial.launch

This should open a new window named RViz.

## Reference list

|  |  |
| --- | --- |
| [1] | Robopeak, “How to build rplidar ros package,” RoboPeak Public Repos, 17 January 2017. [Online]. Available: https://github.com/robopeak/rplidar\_ros/wiki. [Accessed 2020 March 13]. |
| [2] | T. Fiorenzani, “RPLidar and Hector SLAM for Beginners | ROS Tutorial #8,” Youtube, 3 July 2019. [Online]. Available: https://www.youtube.com/watch?v=Qrtz0a7HaQ4. [Accessed 13 March 2020]. |
| [3] | GabrielHuettenberger, “Qt version issue? #59,” GitHub, 19 March 2019. [Online]. Available: https://github.com/tu-darmstadt-ros-pkg/hector\_slam/issues/59. [Accessed 13 March 2020]. |
| [4] | StefanFabian, “tu-darmstadt-ros-pkg,” Github, 4 December 2019. [Online]. Available: https://github.com/tu-darmstadt-ros-pkg/hector\_slam. [Accessed 13 March 2020]. |

## Links

<https://www.youtube.com/watch?v=Qrtz0a7HaQ4>

<https://github.com/tu-darmstadt-ros-pkg/hector_slam>

<http://wiki.ros.org/hector_slam/Tutorials/SettingUpForYourRobot>

<http://wiki.ros.org/hector_slam>

<https://github.com/NickL77/RPLidar_Hector_SLAM>