

## Lab Guide

# Localization via LiDAR - ROS

## Description

In this application we will use a combination of the Hector SLAM packages, gamepad control node and QCar control node. Occupancy grid map generation can be viewed in RViz by replaying the saved ROS bag..

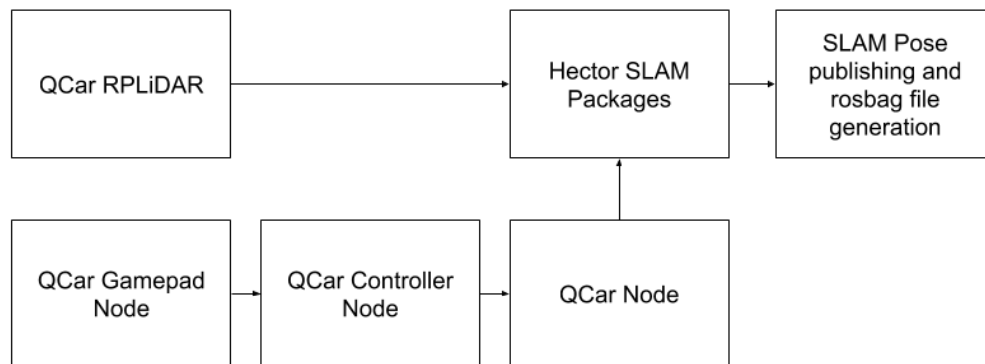


Figure 1. Component diagram

We will use the gamepad to control the speed and orientation of the QCar as it drives around the environment.

**Note:** You can view the map generation in real time by using software like VNC Viewer (Refer to **User Guide – Connectivity** on how to enable remote desktop connection). By default, the map generation is saved in a ROS bag which you may also replay on a separate ROS machine.

Hector SLAM is accomplished by using the following set of standard packages:

- rplidar\_ros
- hector\_mapping
- hector\_geotiff

They are evoked within the slam.launch file located inside of the following directory: **~/ros1/src/qcar/launch**.

As part of this example the following packages have been customized:

- rplidar\_ros

Replace **mapping\_defaults.launch** in **/opt/ros/melodic/share/hector\_mapping/launch** with the custom file provided in with this package..

Please do not upgrade these packages as changes to these files can cause the example to run incorrectly. For the **rpilidar\_ros** package, a modification to the default communication port was changed from `/tty/USB0` to `/tty/THS2`.

The **slam.sh** file launches a simplified set of nodes for the QCar using the launch file `slam.launch`. The following nodes are specific to the QCar:

- `qcarnode.py`
- `commandnode.py`
- `SLAMCorrected.py`

The command node configures the publishing of linear and angular commands generated by the logitech gamepad. The `qcarnode.py` publishes two topics unique to the QCar, the battery level and the QCar estimated velocity. A subscriber is set up to receive linear and angular velocity commands being sent to the QCar.

The `rpilidar_ros` package uses the following axis convention for the RPLiDAR A2:

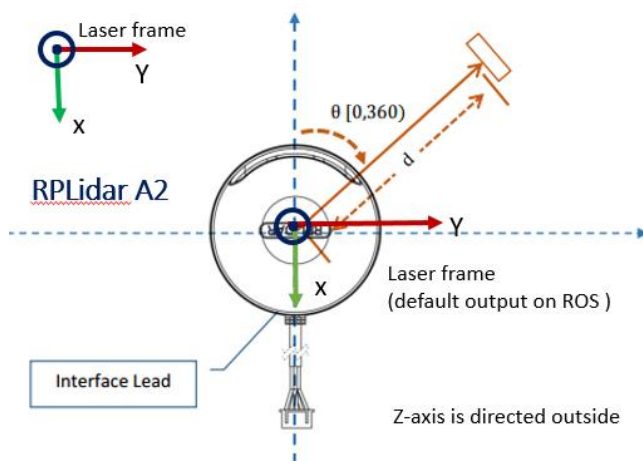


Figure3: RPLiDAR axis convention.

The occupancy grid map generated by the Hector SLAM algorithm also uses the same axis convention as the RPiLiDAR. To keep the axis orientation consistent with the front of the QCar the ros topic `/slam_out_pose` is reoriented via a rotation of 90 degrees and is published via the topic `/slam_pose_corrected`. This corrected heading is displayed in the RViz window shown in Figure 2.

We have also saved the RViz configuration used in Figure 2 which is available in the directory `~/ros1/src/qcar/launch/slam.rviz`. You can use this configuration setting to view the rosbag or if you use a remote desktop connection to run this example on the QCar.

For more information on the topics saved in the **rosbag** please visit the website [http://wiki.ros.org/hector\\_slam/Tutorials/MappingUsingLoggedData](http://wiki.ros.org/hector_slam/Tutorials/MappingUsingLoggedData) for more information. Please remember to use the topic `/slam_pose_corrected` when inside the RViz panel to view the pose of the vehicle with the corrected heading.

## Guide

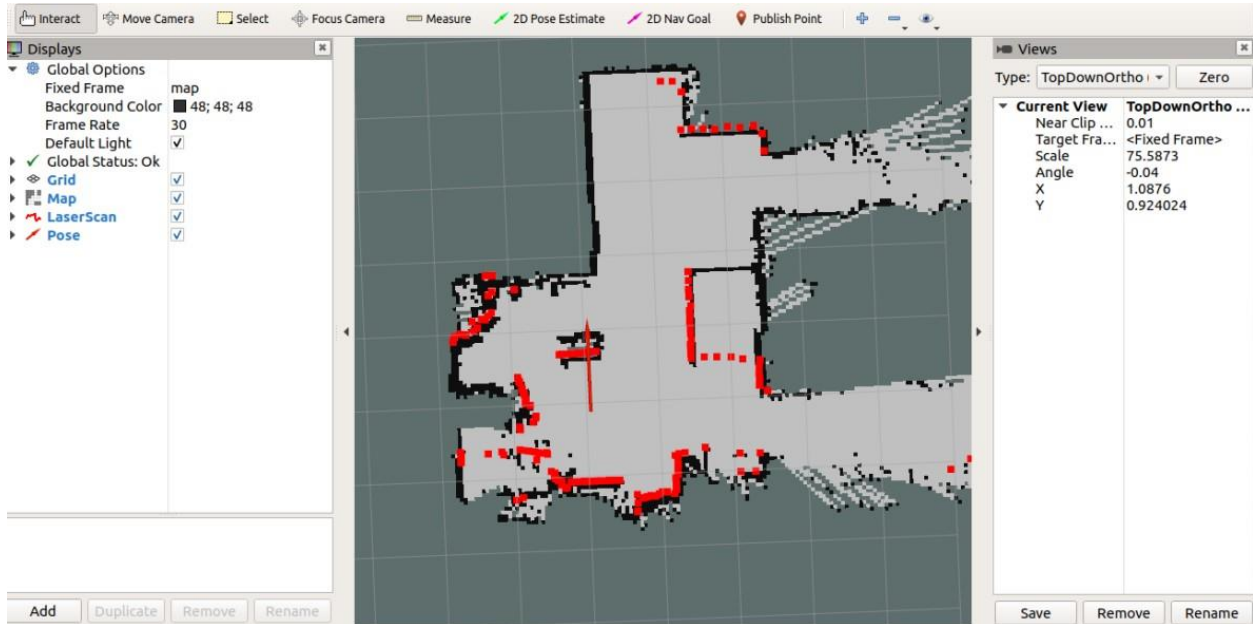


Figure 2. Example occupancy grid generated via Hector SLAM packages..

### 1. File Transfer:

Review [User Guide – Software ROS](#) to confirm how to configure the ros1 package on your QCar. For information on how to transfer files to and from the QCar please review the [User Guide – Connectivity](#).

### 2. Running the Hector SLAM example:

Navigate back to the `~/ros1` directory and with super user authority run the `slam.sh` script using the following line:

```
sudo PYTHONPATH=$PYTHONPATH ./slam.sh
```

Controlling the QCar:

- Use the **LB** button on the Logitech gamepad to **enable** motor commands,
- Use the **RT** to **accelerate** forwards and use the **left joystick** to **steer**.
- To move in **reverse** hold the **LB** and **A** buttons while using the **RT** to control acceleration.

Once you are ready to stop the example you can use the **ctrl+C** keyboard interrupt to terminate the ROS application. To stop the hardware on the QCar run the `HardwareStop.py` script found in `/examples/self_driving_car_studio/hardware_tests` using super user authority. The saved rosbag can be found in the `~/ros1/src/qcar/bagfiles` directory.