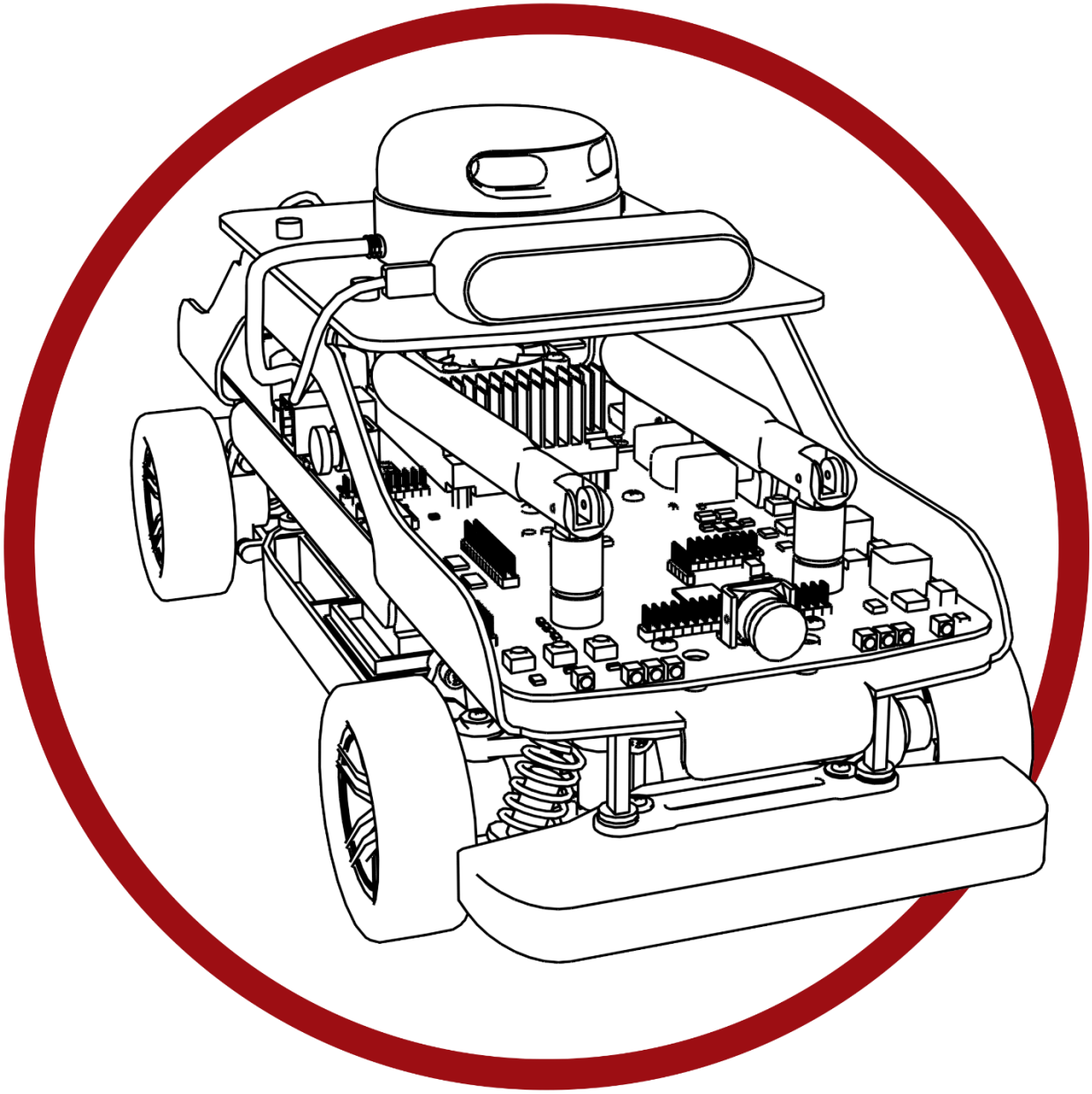
QUANSER
INNOVATE·EDUCATE

Self-Driving Car Research Studio



Localization via LiDAR - ROS

V 2.1 (November 2020)

Table of Contents

I. System 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..

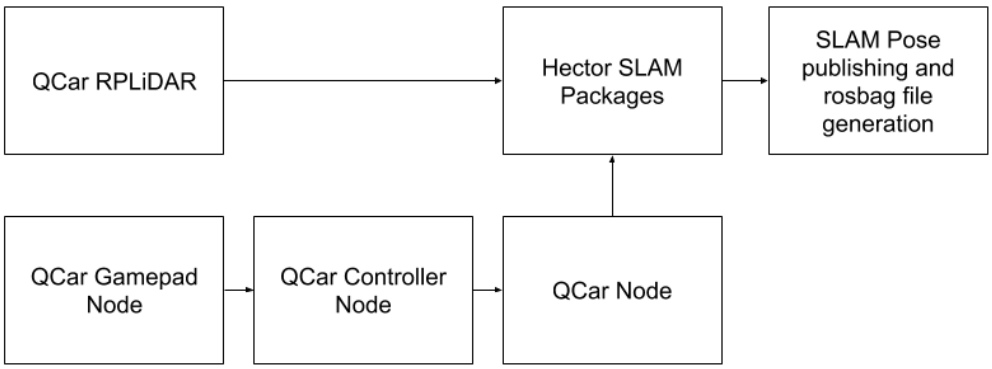


Figure 1. Component diagram

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 III - Connectivity User Guide 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.

II. Running the example

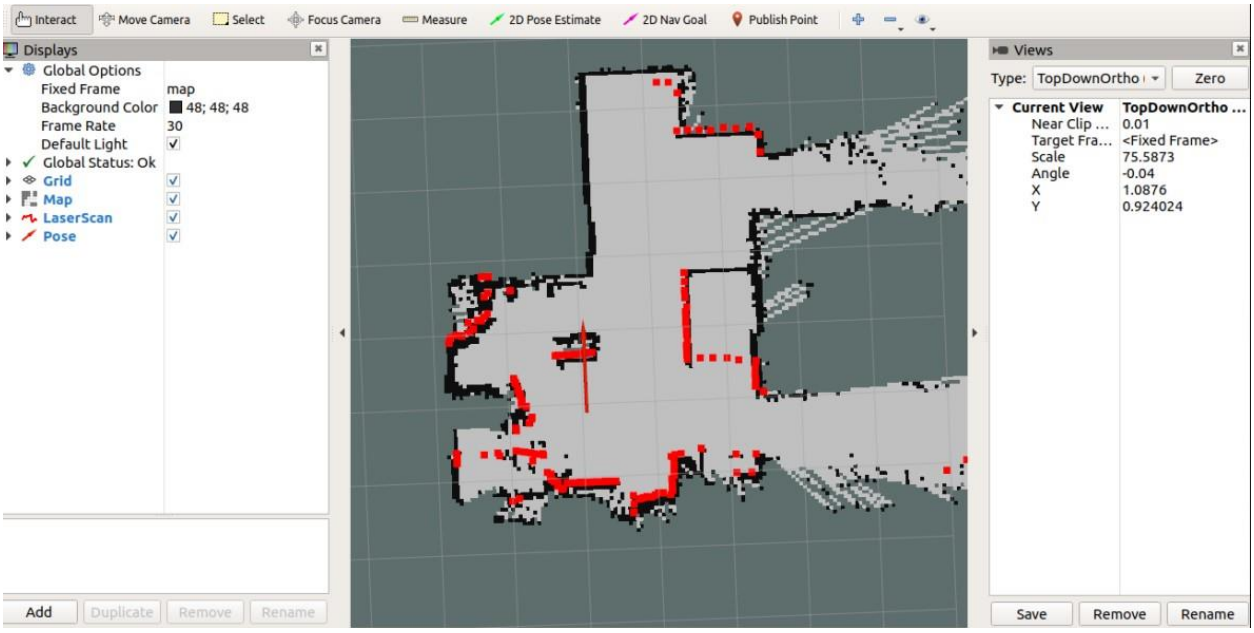


Figure 2. Example occupancy grid generated via Hector SLAM packages..

File Transfer:

Running the Hector SLAM example requires you to copy the file called slam.sh into the **ros1** directory found inside the QCar. For information on how to transfer files to and from the QCar please review the user guide **III - Connectivity User Guide** document.

Prior to running this application make sure you copy the content of **Core Modules/ROS-Python** to the ~**/ros1/src/** directory of the QCar. Then copy the content of **Applications/ROS-Python/Localization via LIDAR SLAM/software** and merge with the ~**/ros1** directory of the QCar. Copy the **HardwareStop.py** script from **Applications/ROS-Python/Localization via LIDAR SLAM/software** to ***~/Documents/Python***.

Running the Hector SLAM example:

In this document we will focus on running the example using a session of PuTTY. Within PuTTY you will need to configure the Logitech gamepad. Please refer to the **II - Hardware Tests - Python** for the steps necessary to identify the device ID for the logitech gamepad. Once you have the gamepad device ID please go to the following directory: **~/ros1/src/qcar/src** and **modify** the file **commandnode.py**. You will need to edit the following line:

```
self.gpad = gamepadViaTarget(<LOGITECH_GAMEPAD_ID>).
```

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

```
sudo ./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 using super user authority. The saved rosbag can be found in the **~/ros1/src/qcar/bagfiles** directory.

III. Details

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
- hector_mapping

Please do not upgrade these packages as changes to these files can cause the example to run incorrectly. For the **rplidar_ros** package, a modification to the default communication port was changed from `/tty/USB0` to `/tty/THS2`. Within the **Hector mapping** package the default mapping launch file was modified to include different variables for the frame names.

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:

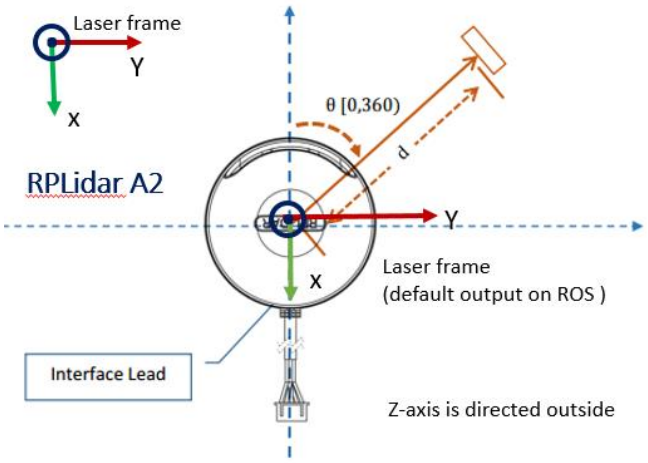


Figure3: RPLiDAR axis convention.

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 being 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 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.