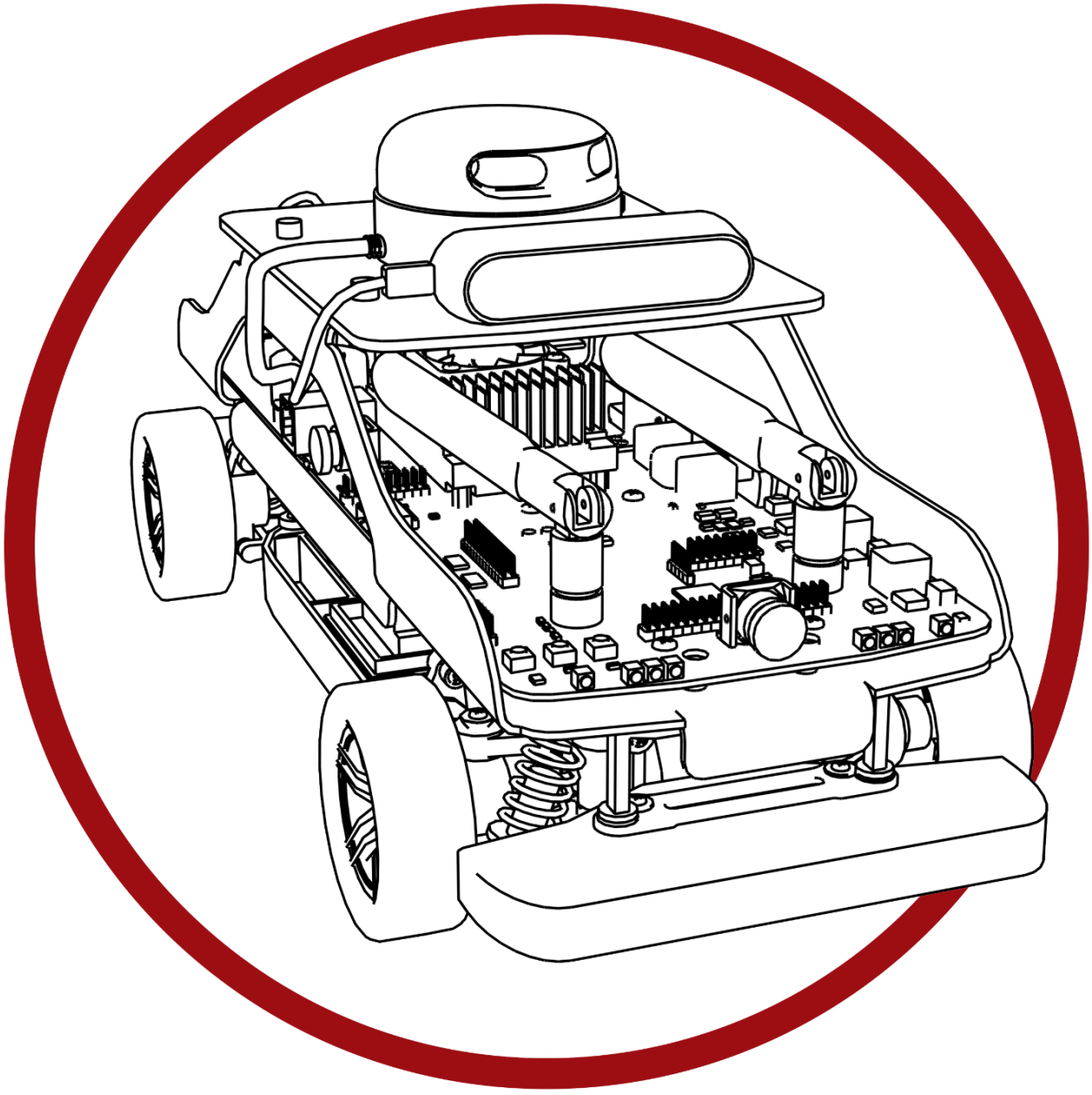
QUANSER
INNOVATE·EDUCATE

Self-Driving Car Research Studio



LIDAR Point Cloud - Python

V 1.1 (November 2020)

Table of Contents

I. System Description

In this example, we will capture LIDAR data from the RP LIDAR A2 on the QCar platform, and generate a point cloud map. The process is shown in Figure 1.

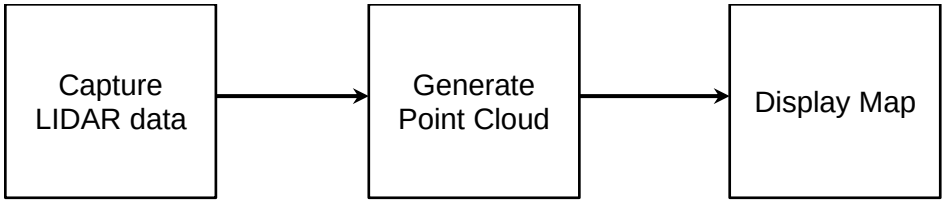


Figure 1. Component diagram

II. Running the example

Check the user guide **V - Software - Python** for details on deploying python scripts to the QCar platform. As your room size may vary, change the parameters **dim** and **gain** as you see fit. Figure 2 shows the typical output expected when running this example (via XLaunch).

Figure 2. Point cloud map generated in a room.

III. Details

1. Capturing LIDAR data

   The data available using the LIDAR class is already adjusted to be presented counterclockwise starting at the positive X axis of the lidar frame (right direction). The LIDAR class object has two attributes, distances and angles, that correspond to the data. The first line below shows the initialization step, and the second shows how to read the data.

```
myLidar = LIDAR(num_measurements=720)

    myLidar.read()
```

2. Converting distances/angles to x y

   After heading angles are converted from lidar frame to QCar body frame, the **distance/heading** data pairs are converted to **x y** pairs (in meters) using the lines below, and then to **pX pY** pairs (in pixels) for the image.

```
x = myLidar.distances[idx]*np.cos(angles_in_body_frame[idx])
y = myLidar.distances[idx]*np.sin(angles_in_body_frame[idx])

pX = (dim/2 - x*gain).astype(np.uint16)
pY = (dim/2 - y*gain).astype(np.uint16)
```

3. Generating the point cloud

   Note that the **map** is set to zeros at the beginning.

```
map = np.zeros((dim, dim), dtype=np.float32)
```

   It is then decayed slowly using the **decay** parameter at the start of the loop.

```
map = decay*map
```

   A line below updates the **map** at the locations **pX pY** near the end of the loop.

```
map[pX, pY] = 1
```

4. Performance considerations

   To improve performance, we only create a blank map when initializing the code. Within the main loop, older map data is slowly decayed. The module **opencv** provides the **waitKey()** method for pausing in this case. See the user guide **V - Software - Python** for more information on timing.