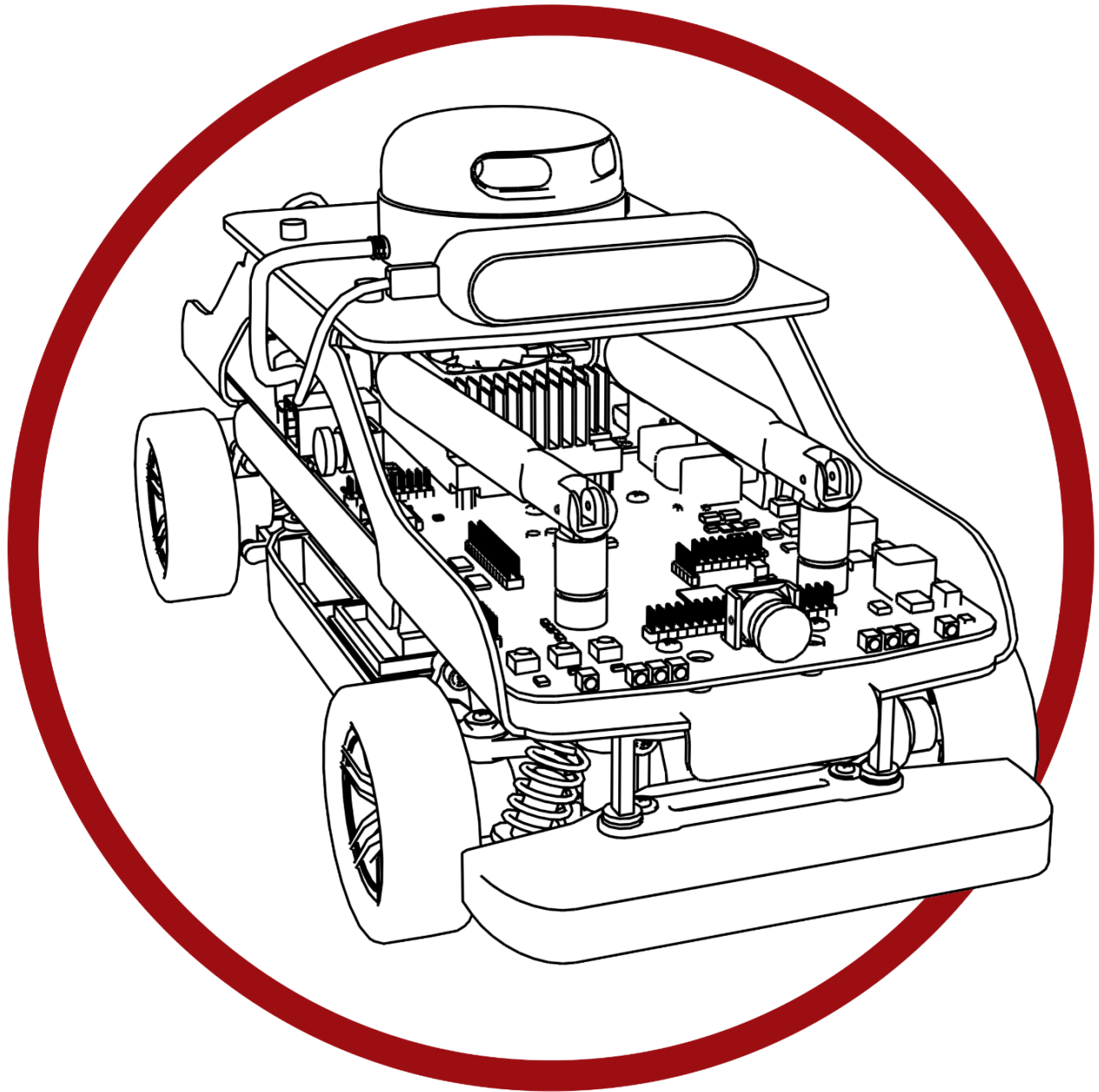QUANSER
INNOVATE · EDUCATE

Self-Driving Car Research Studio

User Guide – Software – ROS (Melodic)

V 1.1 (November 2020)

⚠️ **Caution** **This equipment is designed to be used for educational and research purposes and is not intended for use by the general public.** The user is responsible to ensure that the equipment will be used by technically qualified personnel only.

Table of Contents

A. Overview

Prior to running sample ROS applications please read through the Python user guide **V - Software - Python** before continuing with this document. The ROS code examples and this user guide are using Python and our Quanser Python Library. C++ functions are available for **advanced users**. The overall process is described in Figure 1 below. Design your application as you see fit for **ROS (Melodic)**. You can then transfer the application to the embedded target or run it on your local development machine.

Note: This user guide assumes users have the fundamental knowledge of ROS. Please check http://wiki.ros.org/ROS/Tutorials for more information.
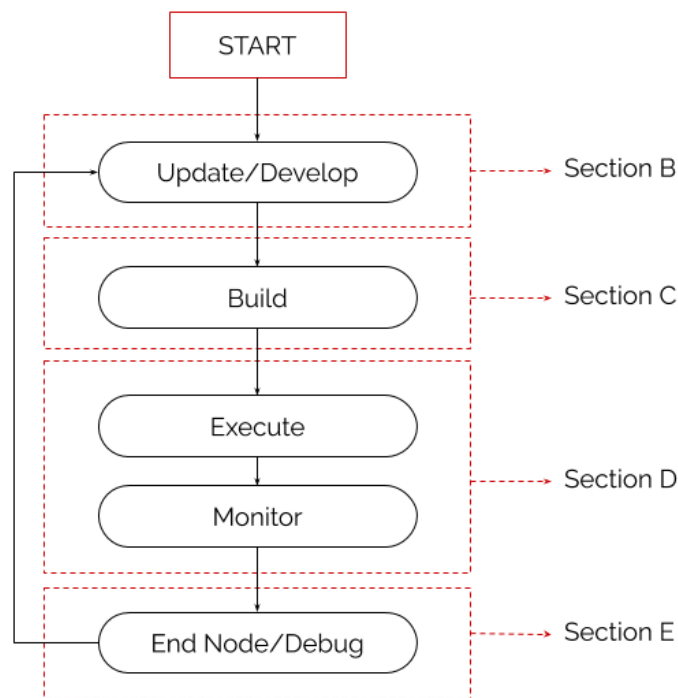

Figure 1. Process diagram for ROS code deployment

B. Update/Development Details

Please refer to the **Quick Start Guide** for links regarding the latest ROS software packages for the QCar.

Please check **Section B** in **V - Software - Python** to make sure all the python modules and Quanser Core (python library) are installed. To check the list of installed ROS packages please run the following command:

```
nvidia@qcar-*****:~$ rospack list-names
```

When developing ROS nodes, please use the **python3** environment for better usage of our python library.

```
#!/usr/bin/env python3
```

Any nodes that use our python library should be placed under **~/ros1/src/qcar/src** directory. Unlike in general Python development, we have also set our python library as a ROS package. Therefore, please **do not** change the name of its folder as it will fail to integrate into the ROS environment. When calling our python library in ROS codes, please use **qcar.** prefix instead of **Quanser.** in Python

```
from qcar.q_essential import Camera2D
```

C. Build

When building the package for the first time, please make sure **catkin_make** uses python3:

```
nvidia@qcar-*****:~$ catkin_make –DPYTHON_EXECUTABLE=/usr/bin/python3
```

Later builds can run without specifying your python version as it has set python3 as default python environment. If you only want to build one single ROS package you can add **--pkg package_name** after **catkin_make**. After finishing building, please make sure to source your devel/setup.bash file.

Note: **Do not run catkin_make while in sudo authority. The best solution we have is for you to make a new ros1 workspace..**

D. Executing and Monitoring Nodes

1. When executing a ROS node, please run with sudo authority if necessary. Do not forget to run **roscore** if you are running a ROS node directly using **rosrun.**.

2. If you are viewing your ROS nodes from a Ground Control Station, please make sure to have **XLaunch** set up properly. See the **III - Connectivity User Guide** for more information on this.

3. Use standard command **rostopic list/echo/hz/info** to monitor the topics that each node is sending/receiving.

    Hint: **Use image_view to view camera streams**.

E. Stop Node & Troubleshooting

Press **Ctrl + C** to stop any ROS node/launch files. If the terminal hangs after pressing **Ctrl + C**, please press **Ctrl + Z** to refresh the terminal. When stopping a ROS node that involves motor and/or Lidar, please run the python script with **sudo** authority called **HardwareStop.py.** Please put the file under *~/Documents/Python*.

For general troubleshooting, users can still use the **try/except/finally** structure in their ROS codes to get error messages. Please view **Section E** in **V - Software - Python** for more information on this structure.

For ROS troubleshooting, users can use either the go-to command **roswtf** or look for more information at http://wiki.ros.org/ROS/Troubleshooting.

QUANSER
INNOVATE·EDUCATE

Solutions for teaching and research. Made in Canada.