Lab Guide

Hardware Tests - Python

Content Description

The following document describes the hardware tests for the QCar 2 in python or MATLAB software environments.

Content Description 1
Initial Hardware Test Procedure 1
MATLAB 2
  Hardware Tests 2
Python 4
  Hardware Tests 4

Initial Hardware Test

This section provides the steps for conducting the initial hardware test to ensure all system components are functioning as expected. It is critical that this real-time (RT) model be deployed and verified before proceeding with any other tests.

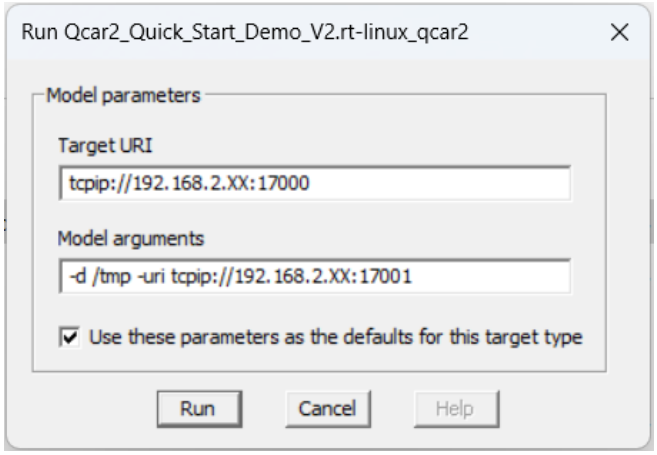
1. **Power the QCar 2**
   Use a fully charged battery to power the QCar 2.
   Note: Refer to the User Manual – Power section for detailed instructions.

2. **Verify Connectivity**
   Perform a quick ping test to confirm that your system is connected to the QCar 2.
   Note: Refer to the User Manual – Connectivity section for detailed instructions on setting up a remote Wi-Fi connection with QCar 2.

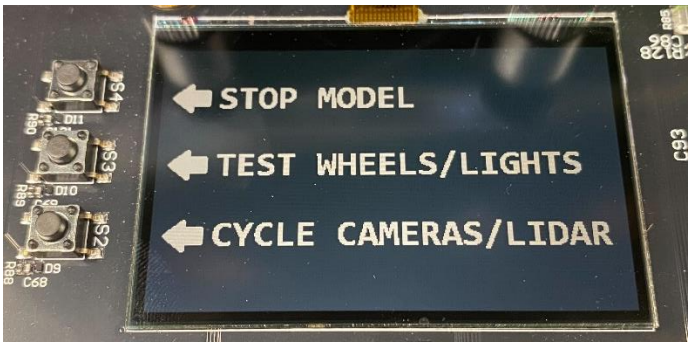
3. **Deploy the Real-Time Model**
   Deploy the 'Qcar2_harware_initial_test.rt-linux_qcar2' file to the QCar 2:
   - Right-click the RT model file
   - Select 'Show more options'
   - Choose 'Run on Target' to begin deployment.

4. **Update Model Parameters**
   Ensure the Target URI and Model arguments are correctly configured:
   - Change the Target URI to 'tcpip://192.168.2.xxx:17000'
   - Update the Model arguments to '-d /tmp -uri tcpip://192.168.2.xxx:17001'

   Replace xxx with the IP address displayed on the LCD screen of the QCar 2.



5. **Run the model**
   After properly configuring the model parameters and clicking the 'Run' button, the LCD screen will display the initial hardware test interface as shown below. Use the left three buttons to navigate the model and follow the instructions on screen to complete the tests.



MATLAB

Hardware Tests

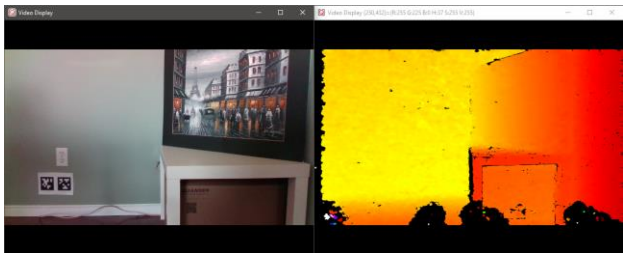
1. Use a charged battery to power the QCar 2.
   Note: Check User Manual – Power for information on this.

2. Use a quick ping test to ensure that you are connected to the QCar 2.
   Note: Check User Manual – Connectivity for setting up a remote Wi-Fi based connection with the QCar 2.

3. Run the individual hardware test models one at a time. Ensure that the IP address in the **MEX-file arguments** of the **Code Generation > Interface** tab of the **Model Configuration Settings** matches that of your platform.
   Note: Check User Manual – Software Simulink for information on how to configure and deploy Simulink models to the QCar 2 target.

4. The expected behaviour is as follows:

   a. Intel_Realsense.slx
   This model should launch an RGB and Depth video display on your screen. An example output is shown here (RGB on left, Depth on right).

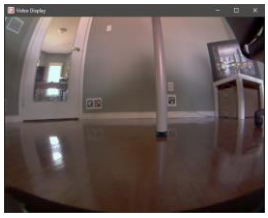


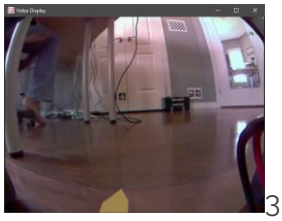
   If your depth display looks monotonous, try reducing the **Maximum pixel value** parameter in the **Image Transform** block inside the **depthDisplay** subsystem.

   b. CSI_Cameras.slx
   This model should display one CSI image in a video display on your screen depending on the camera ID constant (0 to 3) in the model's root level. A sample output is shown below (in order, 0 - right, 1 - rear, 2 - front and 3 - left camera).

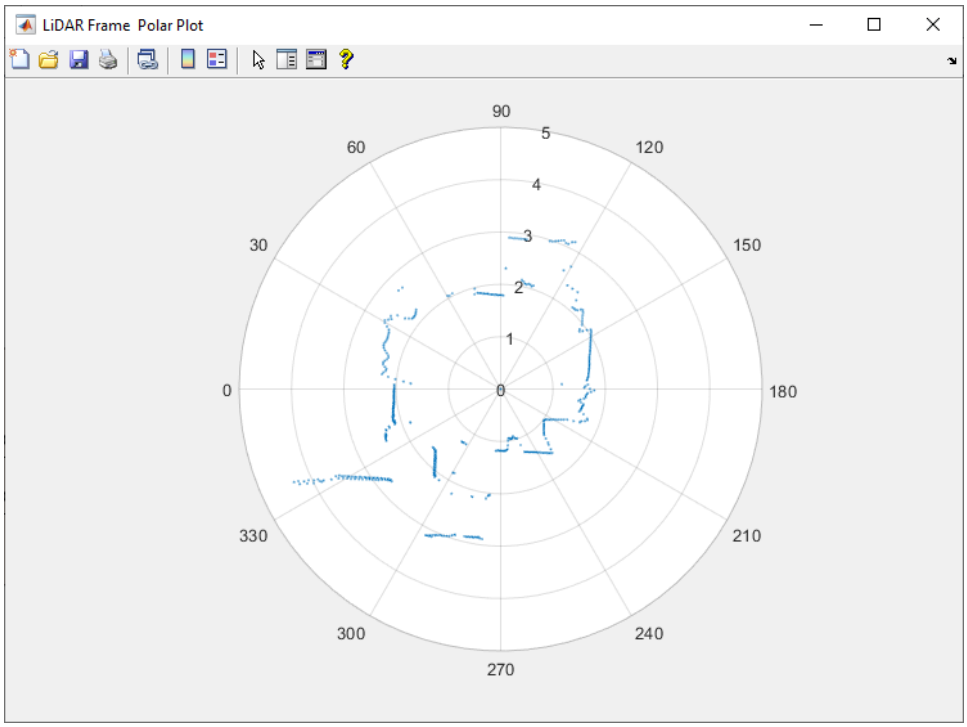

0


1


2


3

   c. RP_LIDAR_A2.slx
   This model should display a polar plot of the LIDAR scans. A sample output is shown here. Note that the 0 degree mark corresponds to the front of the vehicle, and the data is scanned in a counterclockwise positive direction.



d. Basic_IO.slx
This model should let you command the throttle and steering motors and write high (**1**) or low (**0**) values to the 8 LEDs. In addition, you can also read the motor current, battery voltage and motor speed. You should see the wheels spin forwards for a positive throttle, and the front wheels steer towards the left for a positive steering (resulting in counterclockwise rotation).

Python

Hardware Tests

1. Use a charged battery to power the QCar 2.

   Note: Check User Manual – Power for information on this.

2. Use a quick ping test to ensure that you are connected to the QCar 2.

   Note: Check User Manual – Connectivity for setting up a remote Wi-Fi based connection with the QCar 2.

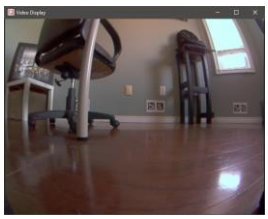
3. Deploy the hardware test scripts on the QCar 2 one at a time.

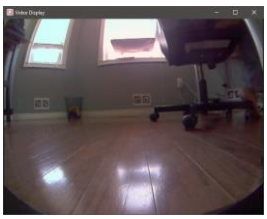
   Note: Check User Manual – Software Python for information on how to deploy python applications to the QCar 2 target.

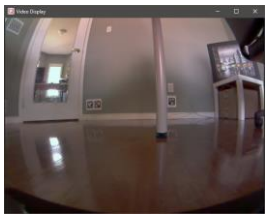
4. The expected behaviour is as follows:

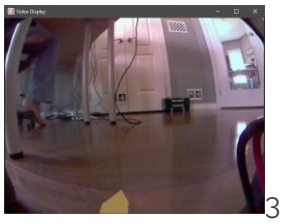
a. hardware_tests_intelrealsense.py
This script should launch an RGB and a Depth window on your screen. An example output is shown here (RGB on left, Depth on right).

b. hardware_tests_csi_cameras_probe.py & observer.py
CSI camera remote streaming using Windows Remote Desktop or X11 forwarding is currently not supported. To see the CSI feeds remotely, first open probe.py and change the variable *ipHost* in the "Initial Setup" section to the IP of the **local machine** before deploying to the QCar 2. Then, run probe.py on the QCar 2 via PuTTY and observer.py on the local machine. Images from all CSI cameras will be displayed in multiple windows on your screen. A sample output is shown below (in order, 0 - right, 1 - rear, 2 - front and 3 - left camera).

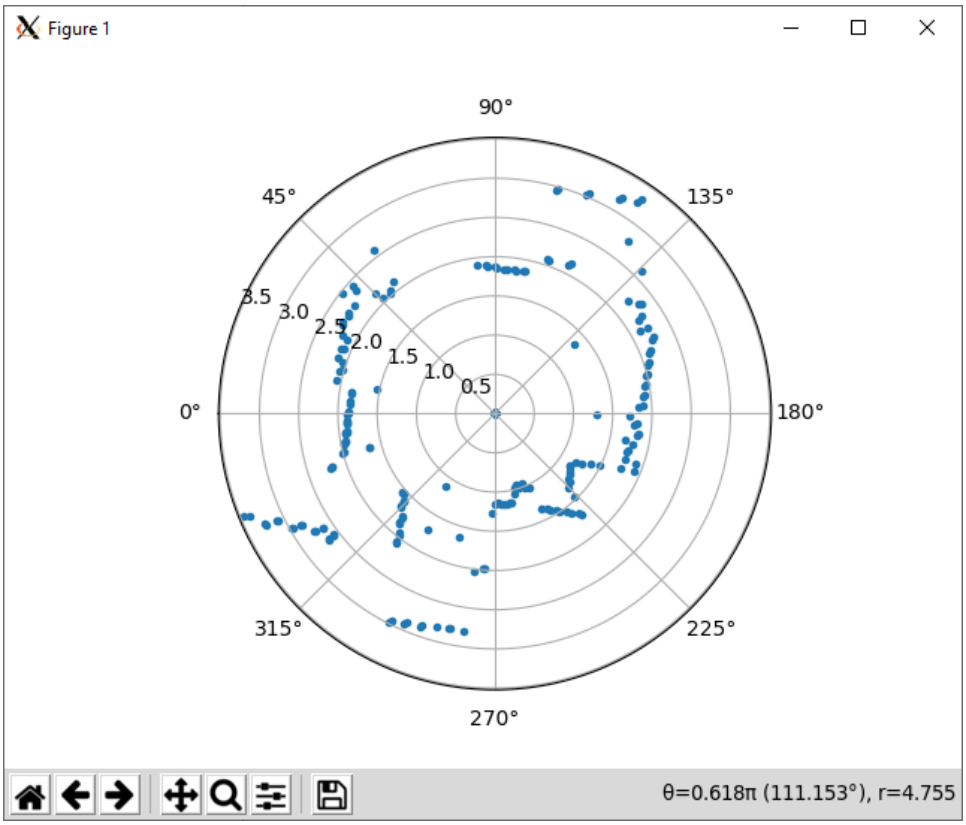

0


1


2


3

c. hardware_test_rp_lidar_a2.py
This application should display a polar plot of the LIDAR scans. A sample output is shown here. Note that the 0-degree mark corresponds to the front of the vehicle, and the data is scanned in a counterclockwise positive direction.



d. hardware_test_basic_io.py
This script should automatically drive a sinusoidal throttle and steering command to the wheels. As the steering changes left and right, the corresponding LED indicators should light up. As the wheels spin forward or backwards, the corresponding headlamps or rear lamps/reverse indicators should light up.

e. hardware_test_gamepad.py
This script initializes and reads the Joystick - **Logitech Gamepad F710**. Plug the gamepad's USB dongle into the USB ports on the QCar 2. As you operate the joystick, the status of corresponding buttons will be printed in the terminal.