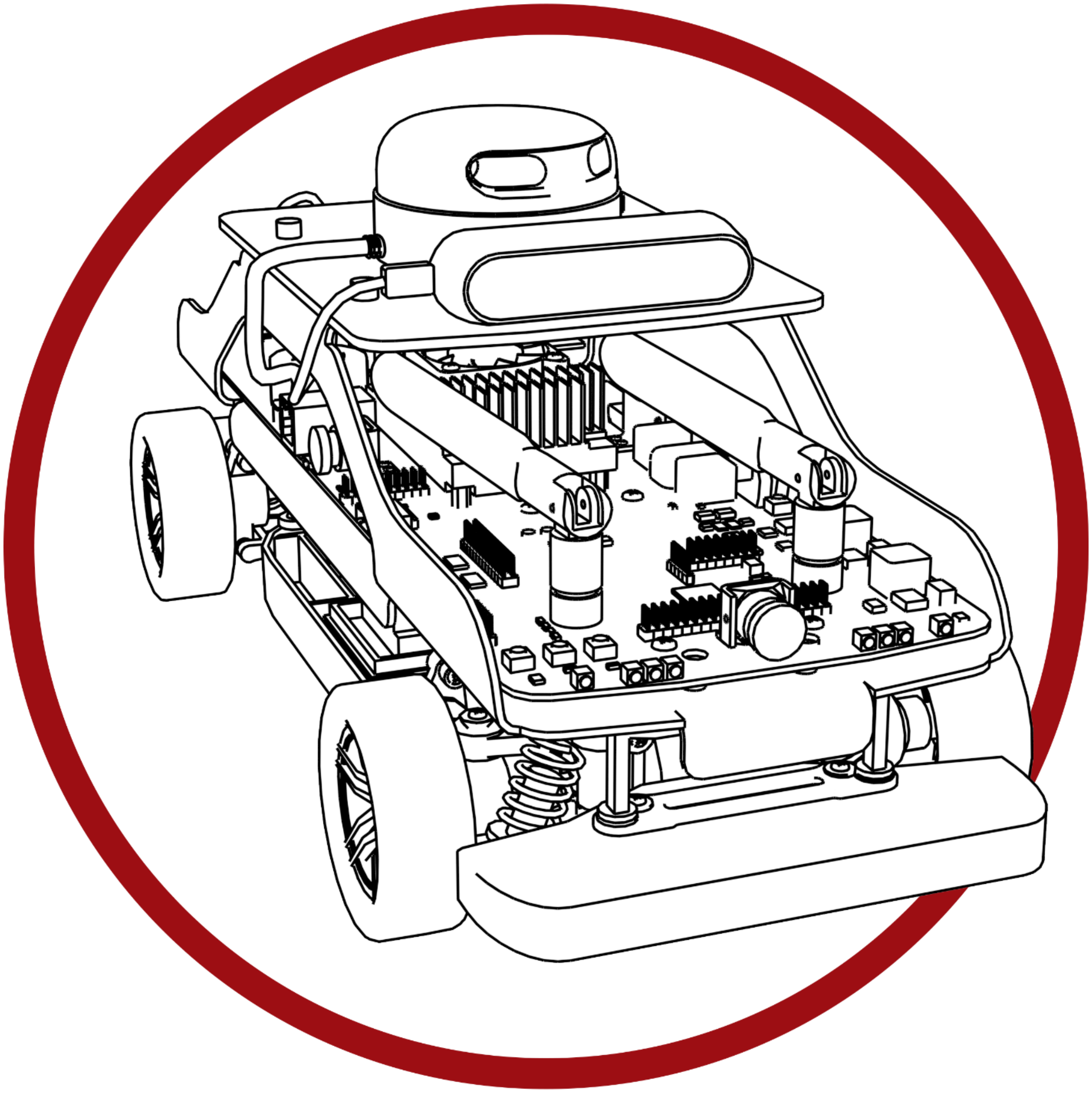
QUANSER
INNOVATE·EDUCATE

Self-Driving Car Research Studio



Quick Start Guide

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.

Latest Documentation and Controllers

For the latest documentation and controllers, please visit Self-Driving Car Research Studio.

Self-Driving Car Research Studio weblink:
https://www.quanser.com/products/self-driving-car-research-studio/

Table of Contents

A. Getting Started

Depending on the configuration purchased for the Self-Driving Car Research Studio Table 1 is a summary of what is available.

	SDRS components
1	QCar
2	Control Station and Infrastructure
3	Reconfigurable Road Panels and Accessories

Table 1. QCar Platform Components

In the following sections we will be going through the steps necessary to set up the Self-Driving Car Research Studio and which documents to follow depending on the development environment you wish to use.

B. Roadmap Setup

If purchased, the Self-Driving Car Research Studio does come with a configurable road layout. You can refer to Figure 1 for possible road layout configurations and the space required to set them up.

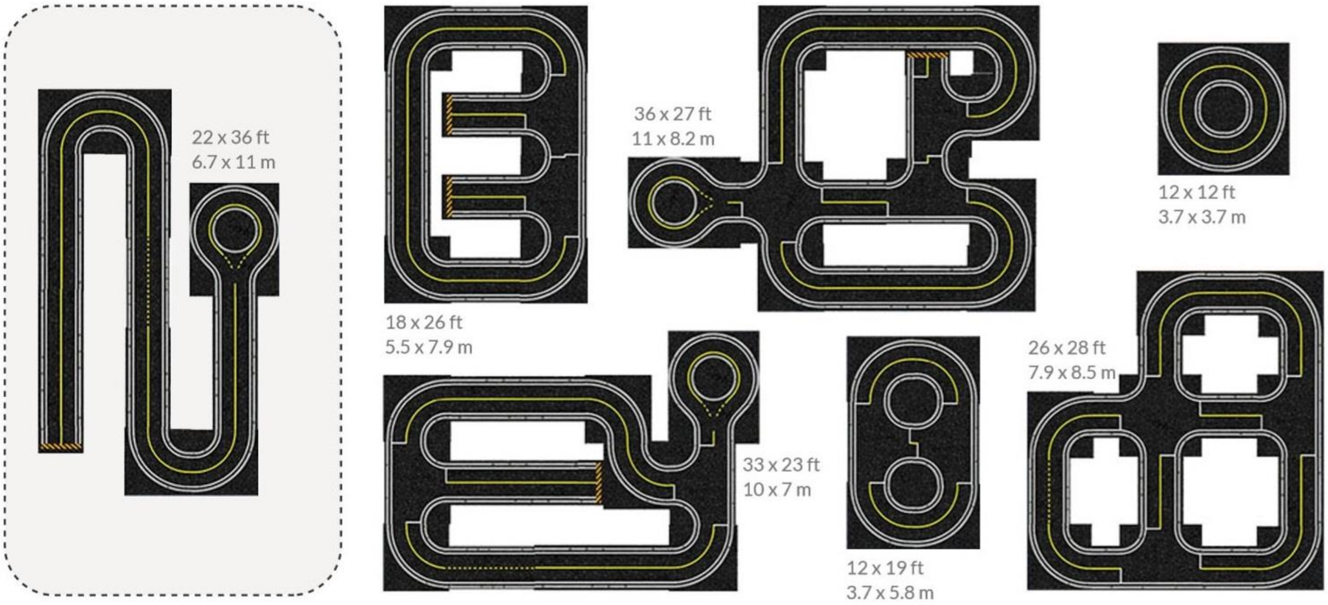


Figure 1: Roadmap layout and dimensions

C. Peripheral Set Up

The standard ground station comes with:

- 3 monitors
- 1 PC w/ keyboard and mouse
- 1 High performance router

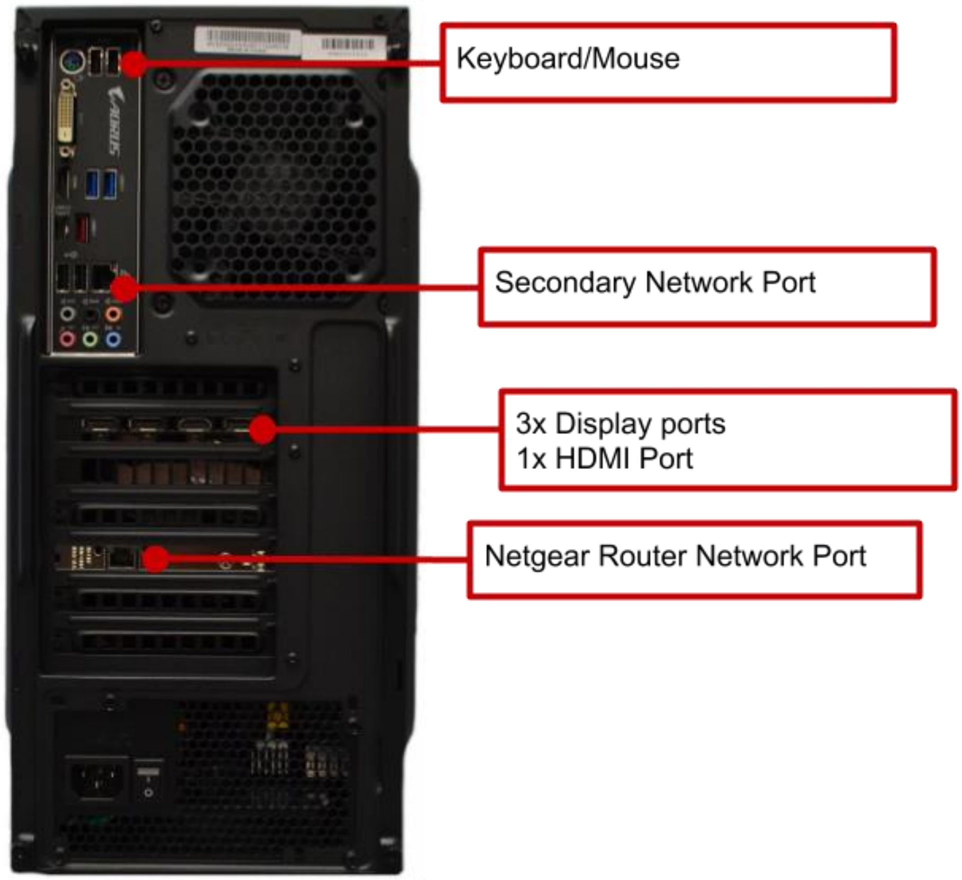


Figure 2. Self-Driving Car Research Studio PC peripheral inputs.

To wire the Self-Driving Research studio in this scenario please use Figure 2, as reference: You may use the **secondary** ethernet port available on the Self-Driving Car Research Studio PC to connect the ground station PC to any additional networks required by your institution. During the set up for the ground station it is always recommended that the user have a full view of the workspace where the QCar will be operating.

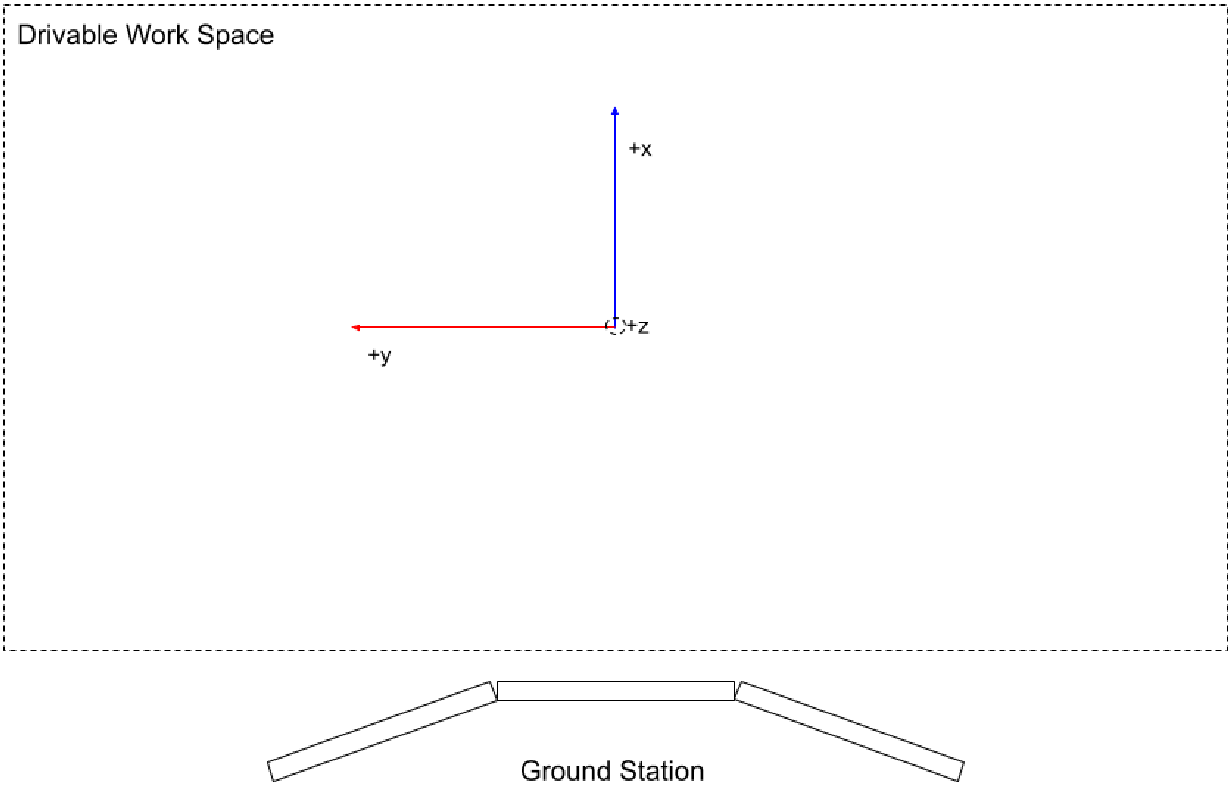


Figure 3. Recommended Self-Driving Car Research Studio layout

The QCar product comes with the following peripherals:

- 1 charger
- 2 LiPo 3300 mhA batteries
- Wall power supply

If you are planning on running the QCar for extended periods of time please make sure the QCar is fully charged to ensure the best performance. For an in-depth explanation on how to use the QCar with either a battery or the provided power supply please review the **I- System Hardware** document found under the **User Guides** for the Self-Driving Car Research Studio.

D. Network Wiring Setup.

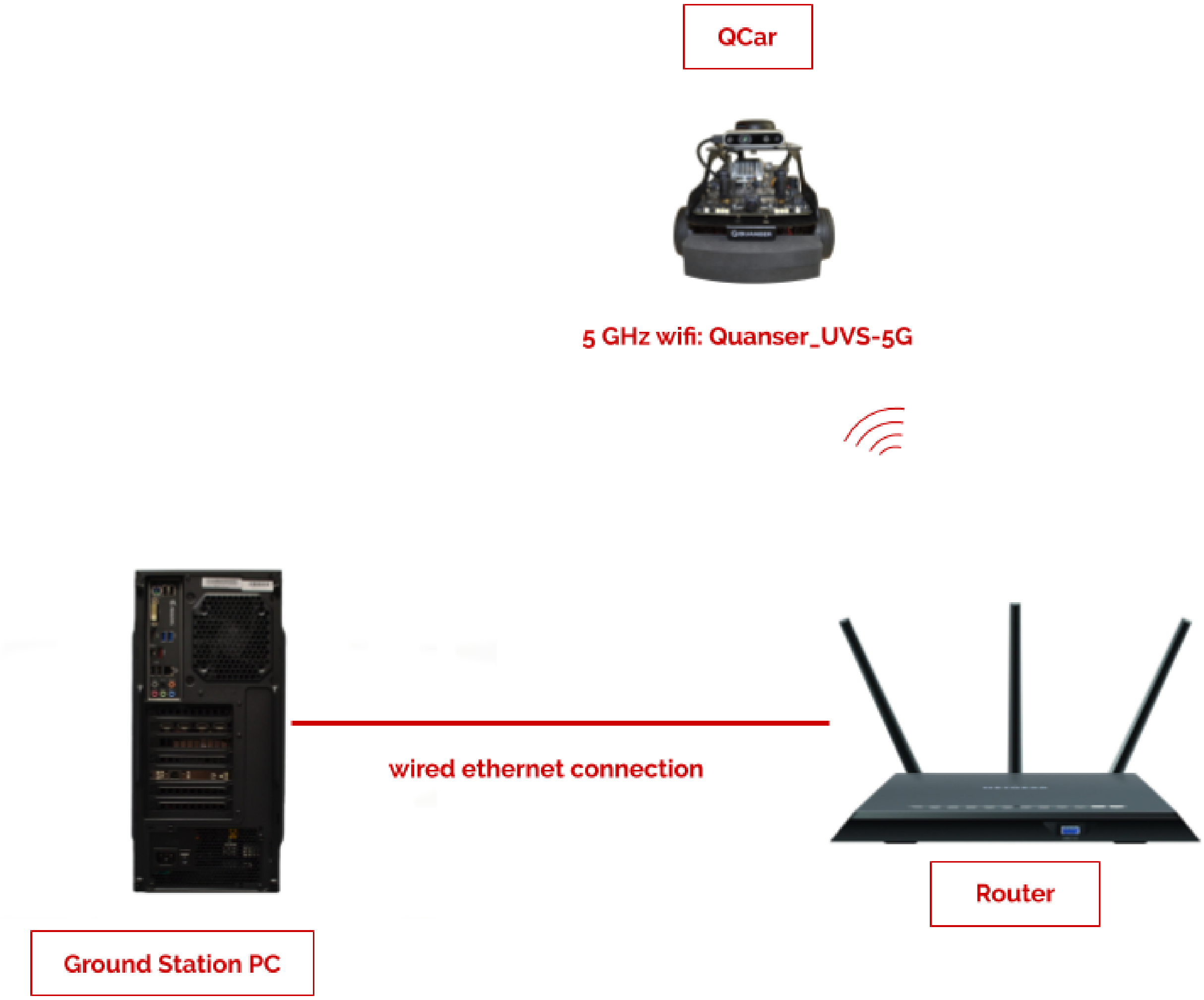
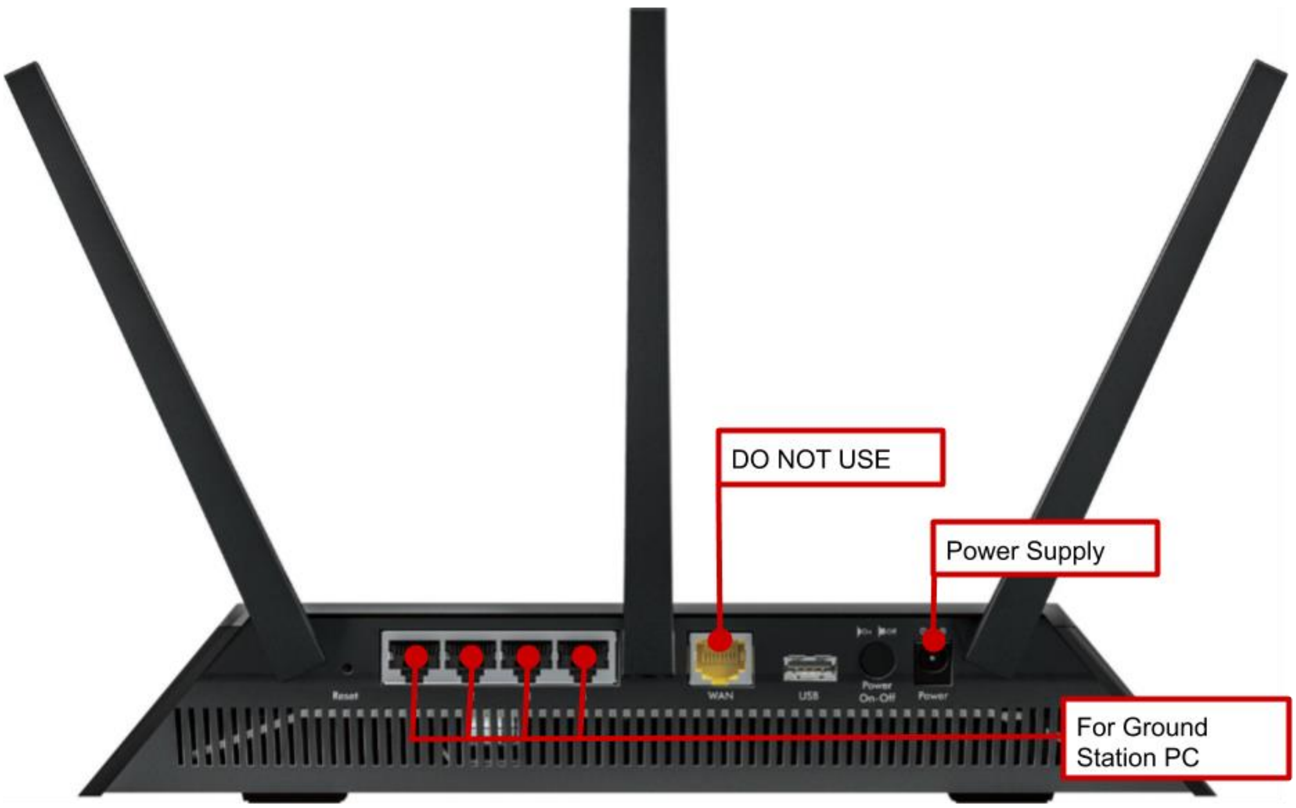


Figure 4. Basic Network Connectivity Map for the Self-Driving Research Studio

Router

The Self-Driving Car Research Studio comes with a NETGEAR R7000 - Nighthawk AC1900 high performance router. It is pre-configured to use both 2.4GHz and 5GHz bands for multiple PCs and autonomous vehicles.

a. Front view of the router b. router power supply c. ethernet cable



d. rear view of the router

Figure 5. NETGEAR R7000 - Nighthawk AC1900 router

Set up sequence:

1. Connect the power supply (Figure 3b) provided with the router to the power port on the back of the router (Figure 5d).
2. Connect the ground control station PC to the router by using the provided ethernet cable (Figure 5c) and one of the four ports on the back of the router labelled 1 to 4 (Figure 5d).

Note: DO NOT use the yellow port labelled WAN to connect to the ground control station PC. This port is used to provide an internet connection to the router, which is not recommended, as the router is configured to optimize local traffic only.

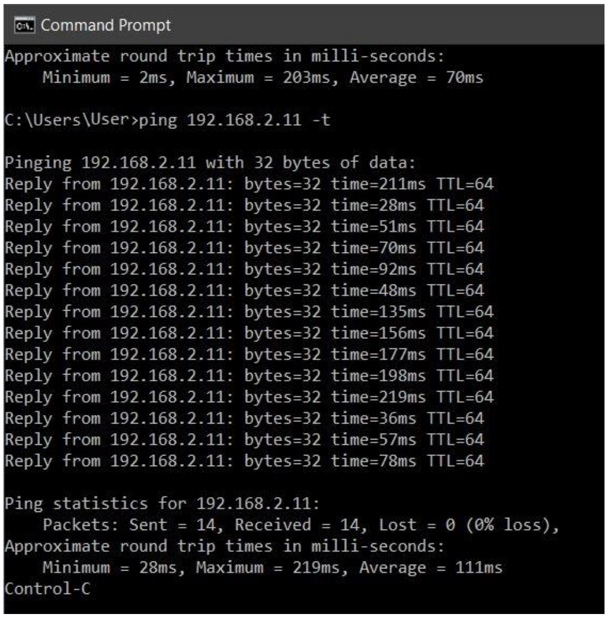
3. Connect the other end of the ethernet cable directly into the ground control station PC using the Ethernet port at the bottom (the PCI-Ethernet adaptor port, see Figure 2).

Note: DO NOT use an ethernet switch or any other device between the router and the ground control station PC

4. Turn on the router. After a few minutes, the lights on the front of the router (Figure 5a) should start flashing with a white light to indicate to the user that the particular ports are active.

E. Connectivity Test

Once the network wiring has been completed you can power on the QCar and use the ping test to confirm connectivity between the QCar and the Ground Station. In Figure 6 the connectivity test was done using the IPv4 address of the QCar which can be found on the LCD display,



```
Command Prompt
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 203ms, Average = 70ms

C:\Users\User>ping 192.168.2.11 -t

Pinging 192.168.2.11 with 32 bytes of data:
Reply from 192.168.2.11: bytes=32 time=211ms TTL=64
Reply from 192.168.2.11: bytes=32 time=28ms TTL=64
Reply from 192.168.2.11: bytes=32 time=51ms TTL=64
Reply from 192.168.2.11: bytes=32 time=70ms TTL=64
Reply from 192.168.2.11: bytes=32 time=92ms TTL=64
Reply from 192.168.2.11: bytes=32 time=48ms TTL=64
Reply from 192.168.2.11: bytes=32 time=135ms TTL=64
Reply from 192.168.2.11: bytes=32 time=156ms TTL=64
Reply from 192.168.2.11: bytes=32 time=177ms TTL=64
Reply from 192.168.2.11: bytes=32 time=198ms TTL=64
Reply from 192.168.2.11: bytes=32 time=219ms TTL=64
Reply from 192.168.2.11: bytes=32 time=36ms TTL=64
Reply from 192.168.2.11: bytes=32 time=57ms TTL=64
Reply from 192.168.2.11: bytes=32 time=78ms TTL=64

Ping statistics for 192.168.2.11:
    Packets: Sent = 14, Received = 14, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 28ms, Maximum = 219ms, Average = 111ms
Control-C
```

Figure 6. Sample ping test between PC and QCar using IPv4 address.

F. Hardware Tests

MATLAB/SIMULINK

To test the SDRS setup within the MATLAB/SIMULINK environment you can run the **BASIC_IO.slx** model found within the **Hardware Tests/Simulink** directory.

To run the model, open the **Hardware Settings/Code Generation and** confirm the **System target file** has been set to **quarc_linux_nvidia.tlc.**

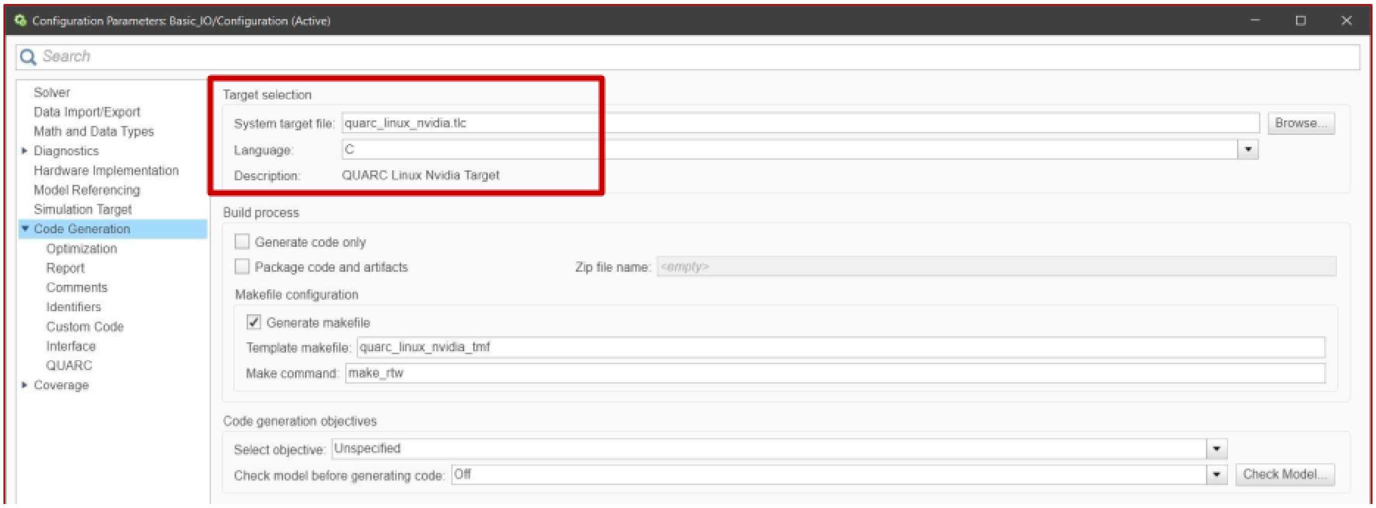


Figure 7. Configuration Parameters target file definition

You will need to **edit** the **MEX-File Argument's** found under **Hardware Settings/Code Generation/Interface** with the following line:

'-w -d /tmp -uri %u', 'tcpip://192.168.2.XX:17001'

Replace the line 192.168.2.XX with the IPv4 Address of the QCar.

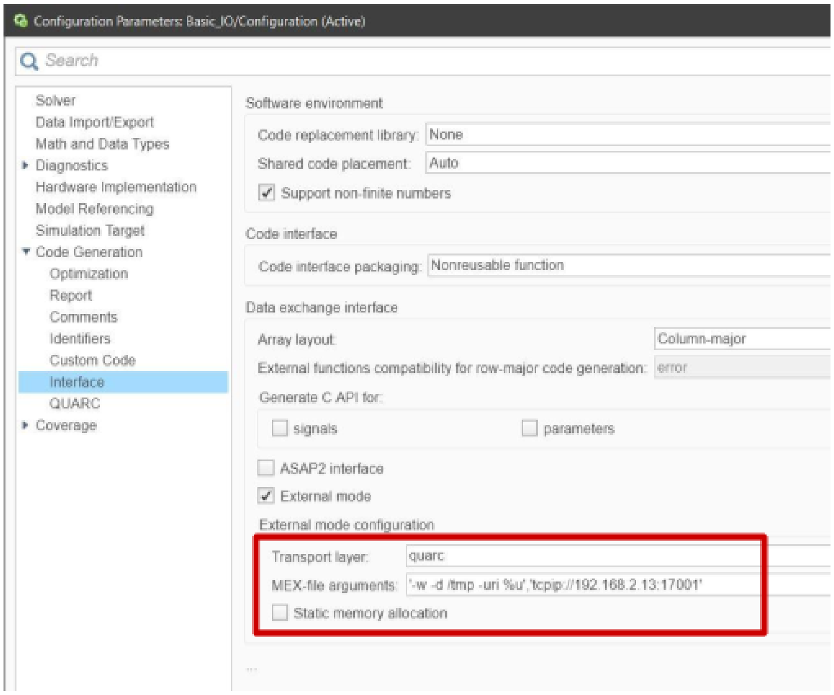


Figure 8. Sample IPv4 Configuration for Mex-File Arguments option in configuration parameters window.

Once these two settings have been checked and configured you can now click on **QUARC/Build.** Once the model has successfully **downloaded** to the QCar click **Connect** and **Run.** You should be able to change the various LED indicators around the QCar and also change the throttle and steering angle.

For a more complete guide of what other IO can be tested for the QCar please go to **Hardware Tests/Simulink** and run the various **.slx** models available for the QCar.

Python

Copy the **Quanser** folder found under the **Core/Python-ROS** directory to the Documents directory of the QCar. For detailed information on how to transfer files to and from the QCar please review the **User Guide/Connectivity** document. To test the basic IO on the QCar using python please copy the python scripts found under the **Hardware Tests/Python** directory to the same directory where you saved the **Quanser** folder (e.g., **/home/nvidia/Documents**). Within the terminal window, **enable superuser authority** and run the **Hardware_Test_Basic_IO.py** script. This script will enable steering, throttle, and LED indicator commands for a preset amount of time. Please ensure the QCar is either **ELEVATED** with the provided stand or in a wide empty room prior -to starting the test.

If the script ran successfully please review the **Hardware Tests** document found within the **Hardware Tests/Python** directory to continue testing the rest of the available sensors on the QCar.

ROS

For testing the QCar using the ROS development environment requires some configuration to be done. Please refer to the **User Guides** for more information regarding the ROS setup. Please review the document **Hardware Tests** found under **Hardware Tests/ROS** for an in-depth explanation of how to test the hardware available on the QCar.

Once you have ensured there is **successful communication** between the **QCar** and the **ground station** you may read the **Support Documentation** for more detail on concepts related to the QCar itself. If you wish to start development of custom applications you may go to the **Application** directory for full examples in either SIMULINK/Python/ROS which use components found under the **Core** directory.

> **Note:** If you run into issues with the example models, code and documents mentioned above please review **User Guides - VII Troubleshooting.**

Solutions for teaching and research. Made in Canada.