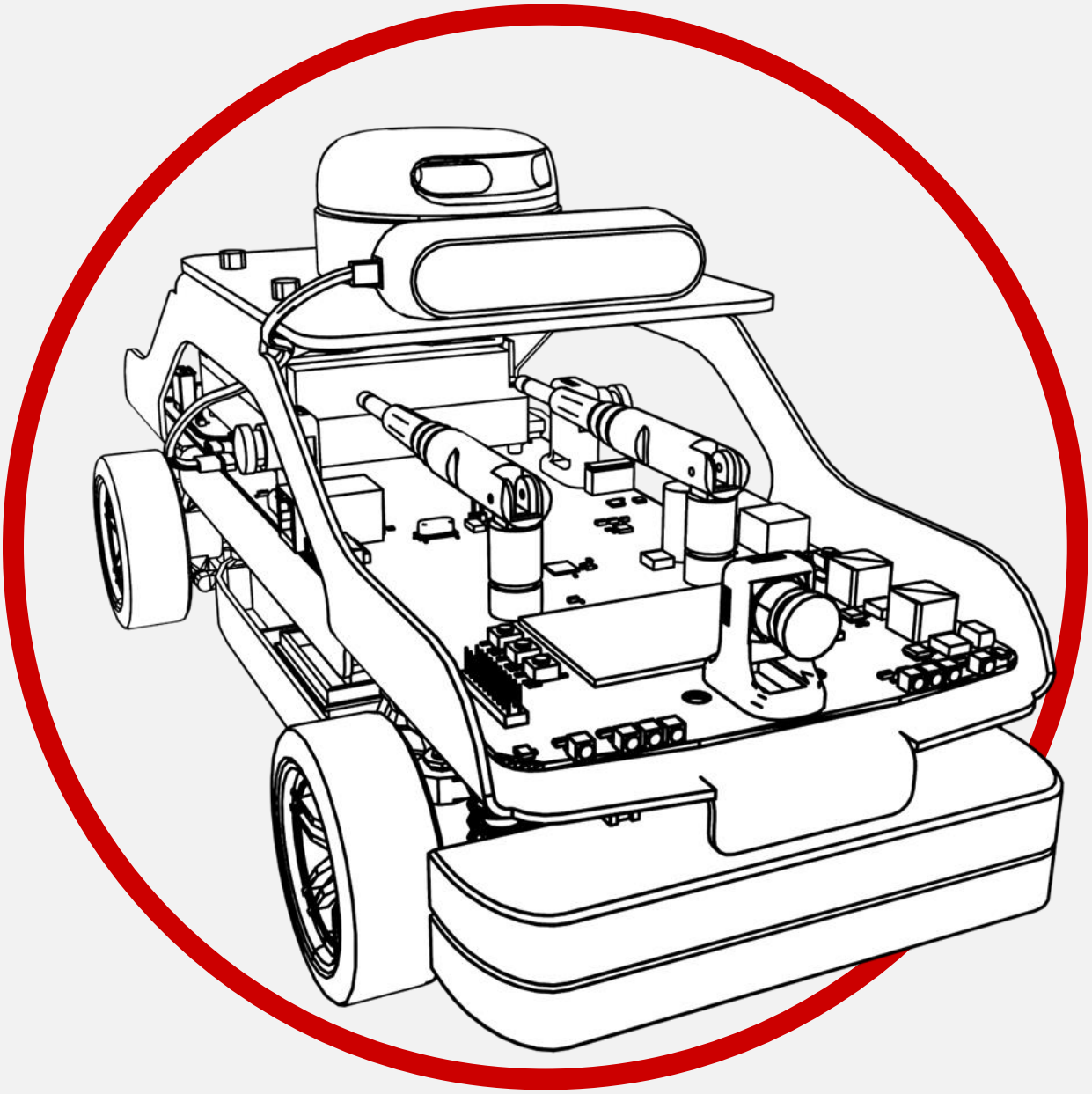
QUANSER
INNOVATE·EDUCATE



QCar 2

User Manual – Software: Simulink

v 1.0 — 1st Oct 2024

For more information on the solutions Quanser offers,
please visit the web site at: http://www.quanser.com

Quanser Consulting Inc.
119 Spy Court
Markham, Ontario
L3R 5H6, Canada

info@quanser.com
Phone : 19059403575
Fax : 19059403576
printed in Markham, Ontario.

Caution

This equipment is designed to be used for educational and research purposes and is not intended for use by the public. The user is responsible for ensuring that the equipment will be used by technically qualified personnel only. **NOTE:** While the GPIO, ethernet and USB ports provides connections for external user devices, users are responsible for certifying any modifications or additions they make to the default configuration.

Caution

The Intel RealSense D435 RGB-D camera is classified as a Class 1 Laser Product under the IEC 60825-1, Edition 3 (2014) internationally and EN 60825-1:2014+A11:2021 in Europe. The camera complies with FDA performance standards for laser products except for conformance with IEC 60825-1 Ed. 3 as described in Laser Notice No. 56, dated May 8, 2019. The RPLIDAR A2M12 reaches Class I laser safety standard and complies with 21 CFR 1040.10 and 1040.11 except for deviations pursuant to Laser Notice No. 50, dated June 24, 2007.
Do not power on the product if any external damage is observed. Do not open or modify any portion of any laser product as it may cause the emissions to exceed Class 1. Invisible laser radiation when opened. Do not look directly at the transmitting laser through optical instruments such as a magnifying glass or microscope. Do not update laser product firmware unless instructed by Quanser.

Table of Contents

A. Overview

Quanser's QCar 2 supports application design in multiple programming languages. This guide describes the process of designing applications using MATLAB/Simulink.

Simulink is used as an open canvas to design your applications as you see fit. To deploy the code, you first build a run-time application (C code) from the Simulink diagram, and then deploy it to your QCar using QUARC. Simulink then connects to the application and displays any output connected to Simulink sinks, such as scope or display blocks, to the screen of the host machine that is running Simulink.

Figure 1 below represents the workflow for developing Simulink applications. The next sections describe each step in more detail.

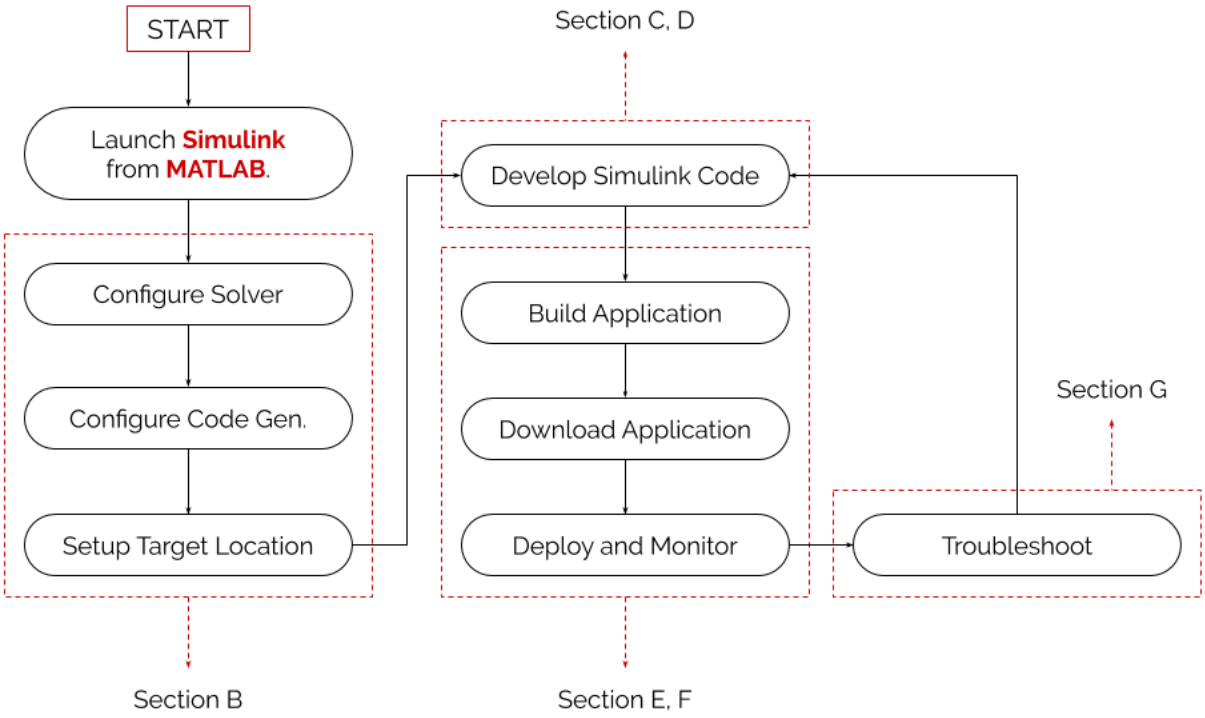


Figure 1. Process diagram for Simulink code deployment

B. Model Configuration Settings

Examples provided for QCar 2 have most settings defined already (except the IP of the car), however, going through these steps will help understand important settings that need to be defined in the model.

1. Open a **Blank QUARC Model** or the example being used. Using the Hardware ribbon in Simulink, open the **Hardware Settings** using the icon  or pressing **`Ctrl+E`**.

2. Under **Solver,** locate **Solver selection** and set the **Type** to **`Fixed-step`** and set the **Solver** to the desired one. If you are unsure, select **`ode1 (Euler)`**.

3. Under **Solver,** locate **Solver details**, set the **Fixed-step size (fundamental sample time)** to the desired time in seconds. For example, for 500 Hz, set the value to **`1/500`** or **`0.002`**.

4. Under **Code Generation,** locate **Target Selection** at the top and set **System target file** based on the application you are creating or deploying, for example:
    a. For applications running on the development machine, use **`quarc_win64.tlc`** for a windows-based system (e.g., the provided Ground Control Station). This may be used for real-time simulations, robust communications proxies to QCars, or infrastructure servers to coordinate the actions of multiple QCars.
    b. For applications to be deployed to a QCar 2 target, select the **`quarc_linux_qcar2.tlc`** for the QCar 2. This will be the typical target for most of the Simulink examples.

5. If the target is the QCar 2, you must also specify the location of the QCar 2 on the network so that the application can be downloaded to the QCar 2 automatically by QUARC. To do this, navigate to **Code Generation > Interface**, locate **External mode configuration** and edit the **MEX-file arguments** by adding the following code:

    ```
    '-w -d /tmp -uri %u', 'tcpip://IP_ADDRESS:PORT'
    ```

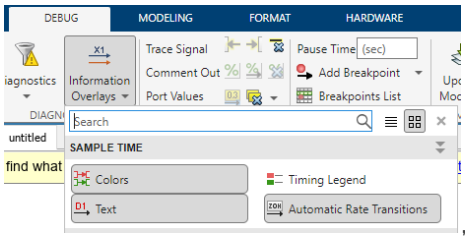
    Do not exclude the comma or the single quotations. Here, **`IP_ADDRESS`** refers to the IPv4 address of the QCar 2, shown on the car's LCD screen and **`PORT`** refers to a network port. Use numbers between 17001 and 17999. Simulink will use the **`PORT`** to communicate with the deployed application and display any data connected to Simulink sinks in your code on your local machine itself. For example, if the IP_ADDRESS of the QCar 2 is **`192.168.2.12`** and we are targeting the model to port **`17001`**, then the MEX-file arguments string will be:

    ```
    '-w -d /tmp -uri %u', 'tcpip://192.168.2.12:17001'
    ```

    Typically, port 17001 will be used if a single model is running on the device, but if you are running multiple Simulink models on the QCar 2 simultaneously, then you will need to specify a different port for each simultaneous model you run (17001 - 17999).

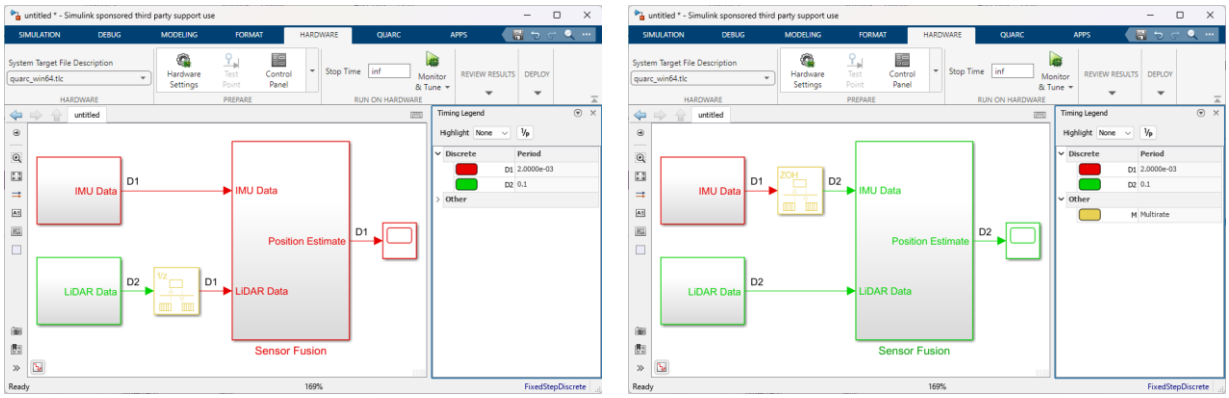
C. Multiple Sampling Rates

The **Fixed-step size (fundamental sample time)** parameter in the **Model Configuration Settings** corresponds to the fastest sample time in your model. To view the timing and different rates in your model, under the Debug Ribbon, go to Information Overlays and under Sample Time, turn on Colors and Text.



The fastest sample time in your model will be denoted as D1 as shown in Figure 2. This also corresponds to the **continuous** sample time. The current value of this parameter can be accessed in the model by using the constant **`qc_get_step_size`.**

Most Simulink/QUARC blocks have a **Sample Time** parameter. When set to -1, this makes the block automatically acquire its sample time based on other blocks that are connected to it. However, it is good practice to explicitly specify the sample time, which makes it easier to read the diagram for future developers and debugging. Integer multiples of the time step constant **`qc_get_step_size`** can be used to specify other sample rates.



a. Sensor Fusion at 500Hz

b. Sensor Fusion at 10Hz

Figure 2. Using Rate Transition to combine signals at various rates

Consider the example shown in Figure 2 where the **LiDAR Data** subsystem provides data at 10Hz and the **IMU Acquisition** subsystem provides data at 500Hz. The subsystem labelled **Sensor Fusion** needs to use both the signals to estimate position. To do so without Simulink errors, you must use a **Rate Transition** block on the signal depending on the rate at which you expect sensor fusion to run. For example, if **Sensor Fusion** must be executed at 500Hz, insert a **Rate Transition** on the LiDAR signal line with sample time set to -1 as in Figure 2a. To run **Sensor Fusion** at 10Hz, insert a **Rate Transition** block on the IMU signal line, with sample time set to -1 as shown in Figure 2b. This ensures that the data transfer between different blocks is handled correctly.

D. Supplied Content

Quanser provides different resources to get started with the QCar 2. If you are using a PC not supplied by Quanser, download the research or SDCS resources from our website. The downloaded resources will have a setup.bat script that will install the resources on your PC. The supplied Ground Control Station (GCS) or the PC with the installed resources will have examples and hardware tests to help start development.

The GSC or the PC with installed resources should have the folder under `Documents/Quanser`.

Under `Documents/Quanser/examples/sdcs/qcar2/hardware` there are two folders, **hardware_tests** and **applications**.

The **hardware_tests** folder includes examples in both Python and Simulink for testing the IO of the car as well as LiDAR and cameras of the system. To run them, refer to section E. Code generation, deployment and monitoring.

The **applications** folder includes different completed examples to start using QCar 2. These examples include lane following, manual driving as well as applications for LiDAR or Cameras.

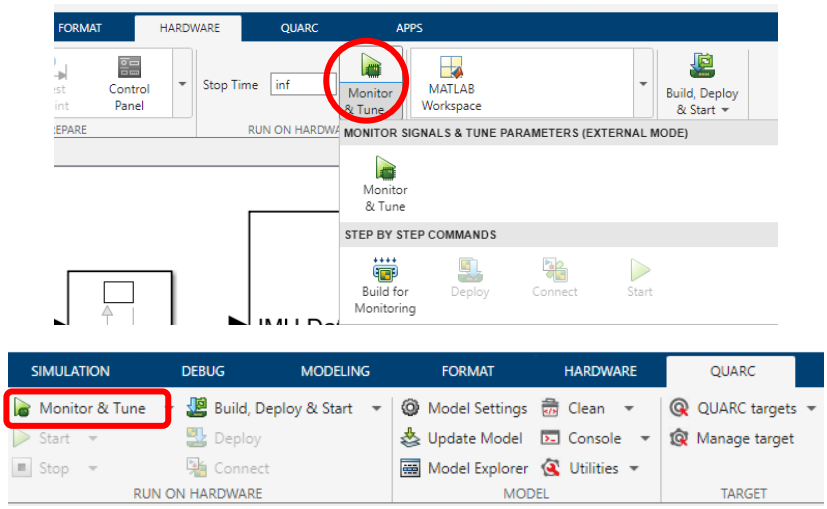
If using the teaching resources, under `Documents/Quanser/SDCS/skills_activities` you will find 5 skills activities to learn different aspects of Self-Driving and walk students through creating a Self-Driving stack. Note that however, these skills activities are only developed for Python. See the User Manual – Software Python to get more information.

E. Code Generation, Deployment and Monitoring

Once your code development is done, you must build the application so it compiles, and you can deploy to the target. Follow these steps to deploy your code:

For MATLAB r2023a and newer:

1. In the Simulink ribbon under the **HARDWARE** tab, click the **Monitor and Tune** button to start the compilation, download and model execution process. You can also use the **Monitor and Tune** button under the **QUARC** tab.



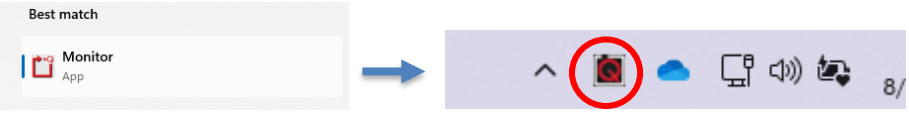
2. Note that when clicking Monitor and Tune, in the background, Simulink is running the 4 steps shown under Step-by-Step Commands shown in the picture above. You could use those one by one.

3. **Stop your model**: When ready to stop execution, click on the stop icon  that will appear instead of the **Monitor and Tune** button or press **`Ctrl+Shift+W`**.

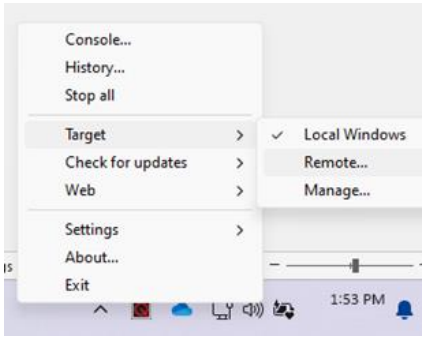
F. QUARC Monitor and Console

The QUARC Monitor is a small application that appears as an icon in the system tray. It provides a convenient mechanism for monitoring whether models are loaded on a local or remote target. For more information about the QUARC Monitor, refer to its documentation.

From the **Start** menu, find and launch the **Monitor** app. It should load a small Q icon in the system tray.



Click this icon, and on **Target,** choose **Remote** if you will monitor apps running on the QCar 2 or **Local Windows** for applications running on the Windows PC. Note that when applications are being run for QCar 2, although you see them in Simulink in your windows PC, the application is actually running in the car, and therefore, to monitor it, you will have to choose remote.



For the **Remote** case, set the **Target URI** to the following,

```
tcpip://IP_ADDRESS:17000
```

where IP_ADDRESS corresponds to the IPv4 address of the QCar (e.g., 192.168.2.11). When the monitor searches for the target, a yellow warning symbol is also displayed as shown in Figure 4b. Once connected, the warning sign disappears as in Figure 4c. There is no need to add any URI when connecting to the local machine.

a. QUARC Monitor tray icon b. connecting to remote target c. connected to remote target

Figure 4. QUARC Monitor tray icon and connecting to target

If any model is running or loaded on the target, the line around the Q (white or green), will be turning. To monitor what is currently running on the system, click the icon, and go to Target>Manage… . This will open a new window. It will show the models currently loaded on the target, the ones configured to run at boot and the ones downloaded on the target.

To open the device's QUARC console to monitor the system: From the **Target Manager** you could click **Console…** at the bottom or click the **Monitor tray icon** and click on **Console…** at the top. This opens a

QUARC console that shows additional information on the system. If you open it before you run the model, you can use it to validate the model being downloaded, connected to and started (Figure 5).

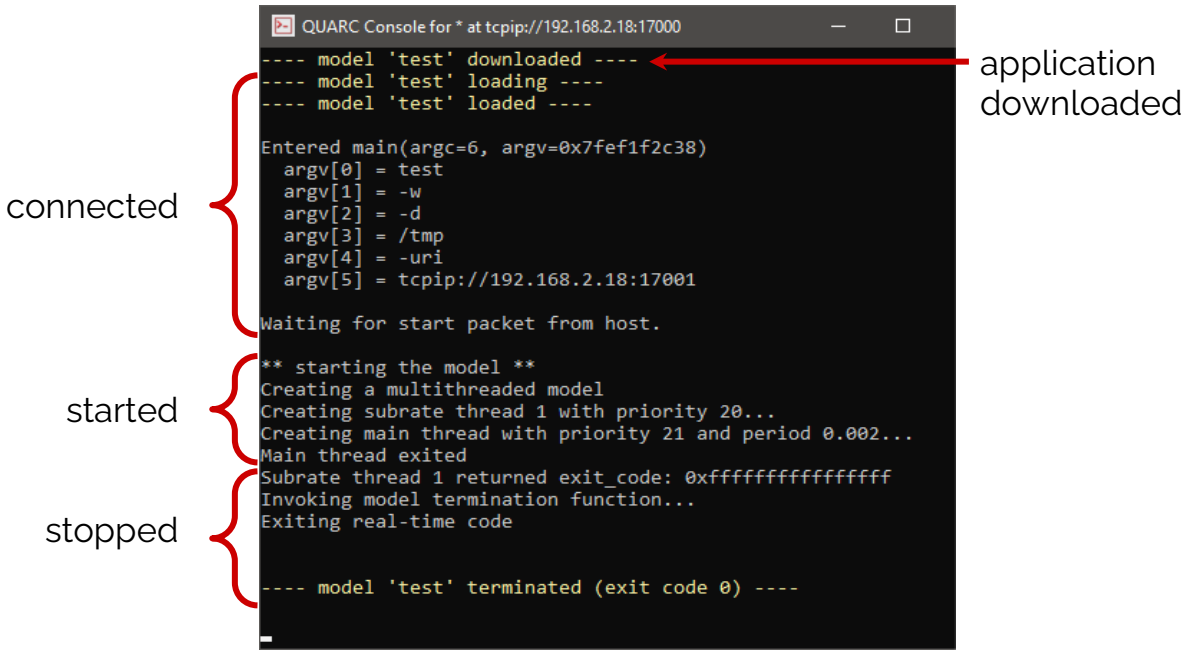


Figure 5. QUARC Console for additional debugging information

G. Upgrading QUARC

Quanser regularly updates QUARC with new features. Many of these features only require that you update your development PC with the latest version of QUARC. The target device (the QCar 2) can typically continue to operate with the original version of QUARC, however, if a QUARC update is required in the QCar 2, it will be noted in the change log of this content package. To upgrade QUARC on the QCar 2, see the User Manual - Customizing QCar 2.

H. Basic Troubleshooting

It is always recommended to ensure that your GCS is connected to the same network as the car, use **`ipconfig`** (in a Windows command prompt) or **`ifconfig`** (in Ubuntu terminal) to check your current IPv4 address. You can also use the **`ping`** command with the IP of the QCar 2 in both Windows and Ubuntu to check your connection to the car:

```
>> ping 192.168.2.115 -t
```

This non-exhaustive list consists of common errors when building code, or connecting to and starting applications.

Note: If your software error is not present here, please contact tech@quanser.com.

1. **The 'Build' step completes successfully but the MATLAB Command Window displays**

   ```
   '??? Model ***model_name*** cannot be downloaded to target
   'tcpip://IP_ADDRESS:17000?keep_alive=1'. It was not possible to connect to the specified URI.'
   ```

   This indicates that even though the build succeeded, and the application is ready, the 'download' step has failed due to QUARC not being able to find the target.

   a. Check your connection to the target - both your local machine and the target should be on the same network. For more details, see the User Manual - Connectivity.
   b. Check the **IP_ADDRESS** of your target in the **MEX-file arguments**
   c. Check the Windows Firewall setting to ensure the requested port numbers in the model URI and in the QUARC Stream API blocks such as Stream Server and Stream Client (e.g., 17000-17020 and 18000-18999) are allowed to pass through (for both incoming and outgoing TCP/IP and UDP traffic).

   Once your connection to the target resumes, download the application already built by clicking on **Deploy** in the **QUARC** tab in Simulink.

2. **The 'Connect' step returns the following error in the Diagnostic Viewer**

   ```
   'Error occurred while executing External Mode MEX-file 'quarc_comm':
   Unable to establish connection with QUARC Target manager for external mode communications. It was
   not possible to connect to the specified URI. Verify that the target is serving on URI
   tcpip://IP_ADDRESS:17000?keep_alive=1'. Use the QUARC Console for debugging.'
   ```

   This mirrors issue (1) described above. Communication with the **Quarc Target Manager** application could not be established because the target could not be found. Follow the instructions given in the troubleshooting steps for issue (1) to establish the connection.

3. **The 'Connect' step returns the following error in the Diagnostic Viewer**

   ```
   'Error occurred while executing External Mode MEX-file 'quarc_comm':
   Unable to establish connection with real-time model for external mode communications. The remote
   peer refused the connection, most likely because no server application was listening for
   connections. Verify that your real-time model is serving on URI 'tcpip://IP_ADDRESS:PORT'.
   ```

The connection between your development machine and the target was successful, however, the built application was not downloaded successfully and hence cannot be found. Click on **Download** in the **QUARC** menu to download the built application to your target and **Connect** again.

Note: It is common to get this error after the error in 1.b. If the connection to the target wasn't established when you used Build to build your application, it will fail to download the model to the target. After resuming connection, trying to Connect directly will also fail, resulting in this error. Try downloading the model again should resolve this issue.

4. **Error occurred while executing External Mode MEX-file 'quarc_comm': The card specific option specified is not recognized.**

   This sometimes occurs when models are moved between MATLAB versions. Find the HIL Initialize block in the model and double click on it to open the dialog. Confirm that QCar 2 is selected then click on the Defaults button at the bottom of the dialog to restore the default board specific options.

Solutions for teaching and research. Made in Canada.