

# QCar 2

## User Manual – Connectivity

*v 1.0 – 1st Oct 2024*

Quanser Consulting Inc.	info@quanser.com
119 Spy Court	Phone : 19059403575
Markham, Ontario	Fax : 19059403576
L3R 5H6, Canada	printed in Markham, Ontario.

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. Quanser Consulting Inc. ("Quanser") grants the following rights: a) The right to reproduce the work, to incorporate the work into one or more collections, and to reproduce the work as incorporated in the collections, b) to create and reproduce adaptations provided reasonable steps are taken to clearly identify the changes that were made to the original work, c) to distribute and publicly perform the work including as incorporated in collections, and d) to distribute and publicly perform adaptations. The above rights may be exercised in all media and formats whether now known or hereafter devised. These rights are granted subject to and limited by the following restrictions: a) You may not exercise any of the rights granted to You in above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation, and b) You must keep intact all copyright notices for the Work and provide the name Quanser for attribution. These restrictions may not be waived without express prior written permission of Quanser.



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. Network Setup	3
i. Wireless	3
B. User Interface	6
i. Direct	6
ii. Remote	7
File Transfer	7
PuTTY	8
iii. Graphical Remote Connections	8
Remote Desktop	9
VcXsrv/XLaunch	11
VNC Viewer	14

## A. Network Setup

### i. Wireless

The QCar 2 is configured to automatically connect to the router provided with the Self-Driving Car Research Studio. The wireless access point (AP) settings for the network with the provided router are,

SSID	-	<b>Quanser_UVS</b> (2.4GHz) or <b>Quanser_UVS-5G</b> (5 GHz)
Password	-	<b>UVS_wifi</b>

The login information for the QCar is the following:

Username	-	<b>nvidia</b>
Password	-	<b>nvidia</b>

To ensure that your Ground Control Station (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 command window to check your connection to the car:

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

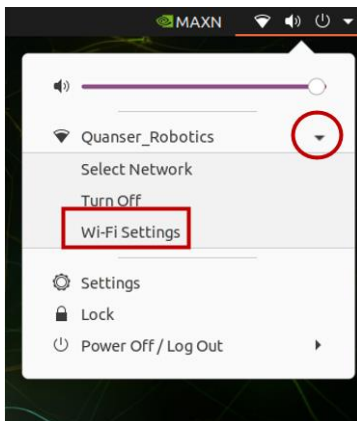
If the QCar 2 was not purchased as part of the studio package or if you choose to set up your own network, you will need to manually configure the Wi-Fi. Please keep the following considerations in mind:

1. The QCar 2 can use either the 2.4GHz (full spectrum) or 5GHz (channels 36 to 64 and 100 to 144) bands. Ensure that your 5GHz network is broadcast over the channels in the range 36 to 144 and not higher. Although it is recommended to use channel 36, 40, 44, and 48.  
**Note:** The 2.4GHz band offers coverage over farther distances but performs at slower speeds. The 5GHz network offers higher bandwidth and data rates over shorter distances.
2. Ensure that your router has the Dynamic Host Configuration Protocol (DHCP) server enabled. This will ensure that the router automatically assigns an IP address to the QCar 2 when it connects.

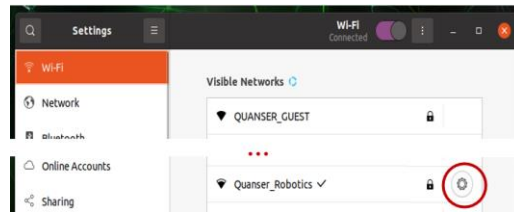
If you want to manually configure the Wi-Fi or change networks, follow the steps below:

3. To have the QCar 2 connected to your network, connect the car directly as shown in section B. i. (using an HDMI cable, keyboard and mouse) and use the Ubuntu Wi-Fi configuration menu as shown in Figure 1a to connect to the desired network of your choice.

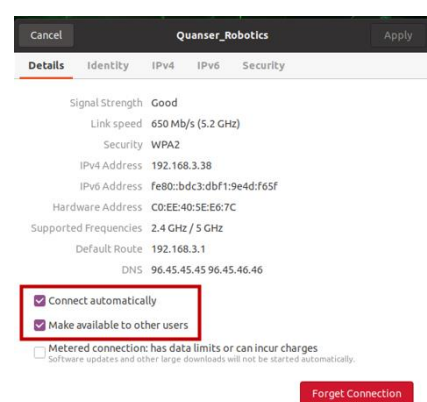
4. After connecting to your network, follow the steps below to ensure that the QCar 2 can connect to the network when a user is not logged in (See Figure 1 for details).
  - a. While the QCar 2 is still connected directly as in section B. i., go to Wi-Fi>Wi-Fi Settings
  - b. Click on the gear icon next to the connected Network (Figure 1b)
  - c. Navigate to the Details Tab and check the box that reads "Make Available to Other Users" (Figure 1c)



a. Edit connections



b. Edit network configuration



c. Allow all users to connect

Figure 1. Allowing the QCar 2 to connect to a network when not logged in

In this case, the LCD will show the network it is connected to, the IPv4 of the QCar 2 assigned by your network, as well as a wireless symbol next to the network strength (%) highlighted in Figure 2.

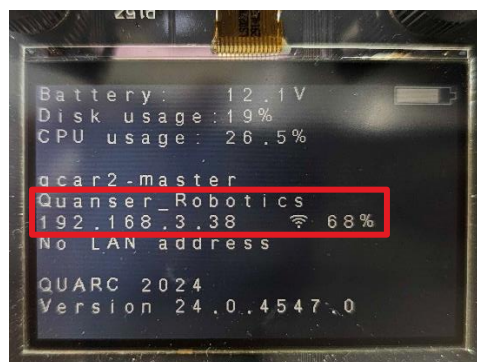
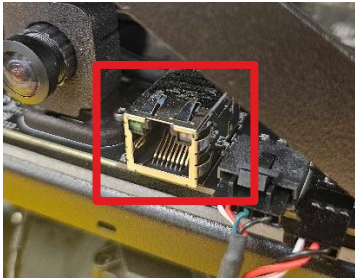


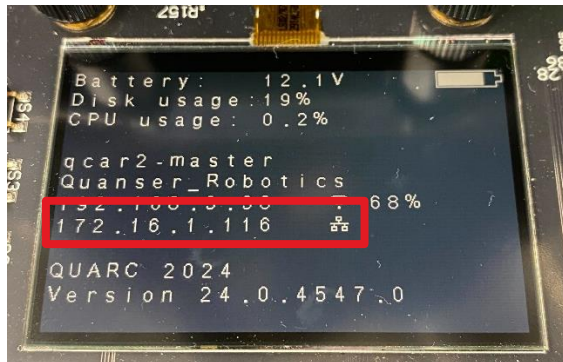
Figure 2. Wi-Fi setup with the QCar 2

## ii. LAN

The QCar 2 may be connected to the supplied router or network of your choice using an ethernet cable connected to the 10/100/1000 Base-T Ethernet jack highlighted in Figure 3a. In this mode, the LCD will show the IPv4 of the QCar 2 assigned dynamically by the DHCP server on the provided router or the network, next to an ethernet symbol highlighted in Figure 3b. When no LAN connection is provided, that line in the display will show "No LAN address" as shown in figure 2.



a. Ethernet jack on QCar 2



b. LCD showing wired connectivity

Figure 3. Ethernet setup with the QCar 2

## B. User Interface

### i. Direct

The QCar 2 can be used directly as a computer, complete with an 8-core CPU, an NVIDIA GPU, a built-in speaker, microphones and an extensive sensor suite. Connect a keyboard/mouse using the provided USB port and connect a monitor using the HDMI port on board. Lift the QCar 2 wheels off the ground using a box if your application or test will require the wheels to move. You can use the external power supply provided with the QCar 2 that will let you run continuously without worry of batteries running low.

An example setup is shown in Figure 4 below. We recommend using LAN when the QCar 2 is connected using the external power supply.



Figure 4. Direct setup with the QCar 2

This setup will typically be used when developing applications using Python/ROS directly on the QCar 2. Examples include viewing collected datasets, training neural networks on-board, post processing collected data during an experiment, or developing image processing algorithms where fluid motion is required by the user.

## ii. Remote

For applications that require the QCar 2 to be moving or require remote access, a direct setup is not feasible. In such cases, a wireless setup may be used, with the provided GCS (if the QCar 2 was purchased as part of the Self-Driving Car Research Studio) or a machine of your choice. The PC must be connected to the same network as the QCar 2 to be able to deploy applications to the QCar. See Section A.i. for login information to the QCar 2,

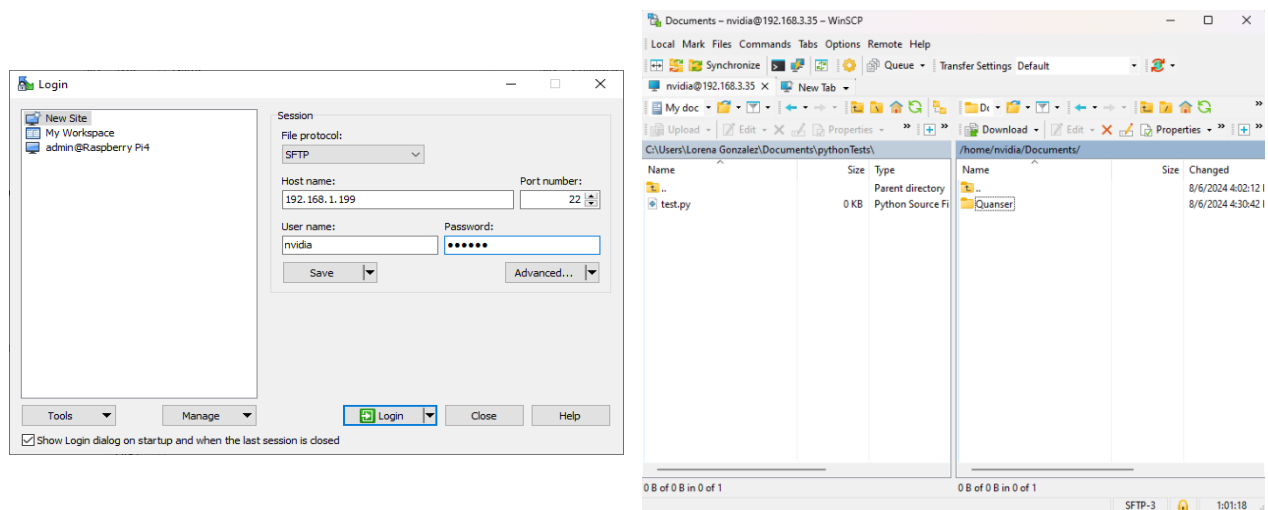
To ensure that your GCS is connected to the same network as the car, use **ipconfig** (in a Windows command prompt) or **ifconfig** (in Ubuntu terminal) to check your current IPv4 address. You can also use the **ping** command with the IP of the QCar 2 in both Windows and Ubuntu to check your connection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

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

### File Transfer

To transfer files between the PC and the QCar, a software tool **WinSCP** can be used. It is installed by default on the provided GCS. Find more information on WinSCP [here](#). This can be used for one-time file transfers or to edit files: navigate to a file onboard the QCar 2, double click on it in WinSCP, and it will open in your default text editor in Windows. When the file is saved, it will automatically be transferred back to the QCar 2.

To use WinSCP, enter the QCar 2's IP address as the host name, **nvidia** as the username, and **nvidia** as the password, as shown in Figure 5a. You can now use the WinSCP browsers to transfer files from the GCS (left panel) to the QCar 2 (right panel), shown in Figure 5b. Dragging a file from one side to the other (PC to Car or Car to PC) will automatically transfer a copy of the file to the other system.



a. WinSCP login

b. WinSCP browser to transfer files

Figure 5. WinSCP usage for file transfer



## PuTTY

Once your files are transferred, you can execute them via remote terminal using **PuTTY**, which is also installed on the provided GCS. Find more information on PuTTY [here](#). If you are familiar with Linux command-line text editors, you can also edit code on the QCar 2 via PuTTY. PuTTY by itself is sufficient to access the file system and execute code that does not require any graphical feedback display.

To use PuTTY, open the application and enter the **IP address of the QCar 2** in the hostname field as shown in Figure 6, leave the default port, and click Open. You will be prompted to login (username **nvidia**, and password **nvidia**). Note that you can open more than one PuTTY terminal to the QCar 2 if you need to.

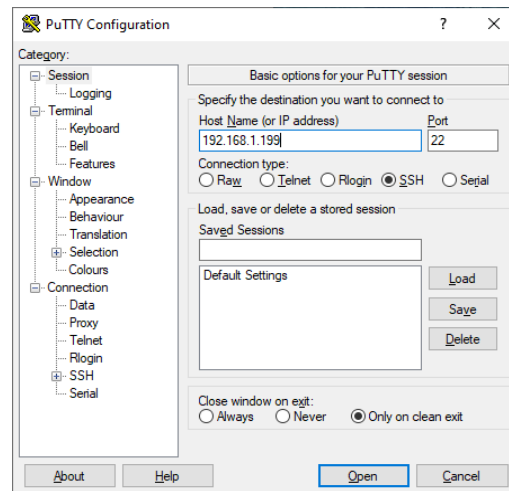


Figure 6. Connecting to the QCar 2 with PuTTY

### iii. Graphical Remote Connections

In this section, we will describe a few ways to connect to the QCar 2 and have a graphical interface of the QCar remotely.

For development directly in the car, we recommend using the **Windows Remote Desktop** method. It is the closest to the direct method described in Section B.i, especially useful when the car needs to be moving, or you want to use your PC alongside the QCar without worrying about having two mice and keyboards.

If you want to not render the desktop, but simply receive the graphical display output from the script, use the **VcXsrv/Xlaunch** method. This is the most performant method for remote monitoring of graphical feedback since only necessary visuals are streamed.

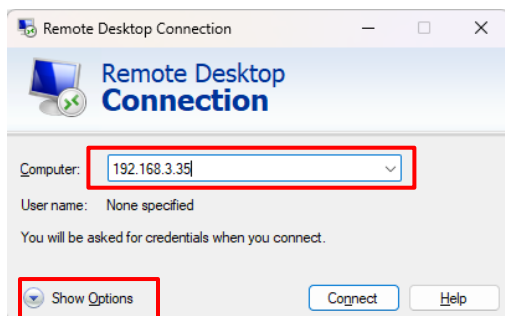
Finally, we describe the method of using **VNC**, with the limitation of being slower and needing an HDMI dongle included in the QCar 2 box, or an HDMI connected to initialize the connection.

**Note:** neither Windows Remote Desktop nor Xlaunch supports the streaming the CSI camera feeds. If needed, use the VNC method described in the following pages. In addition, we also provide examples on using our probe API to remote monitor CSI feeds.

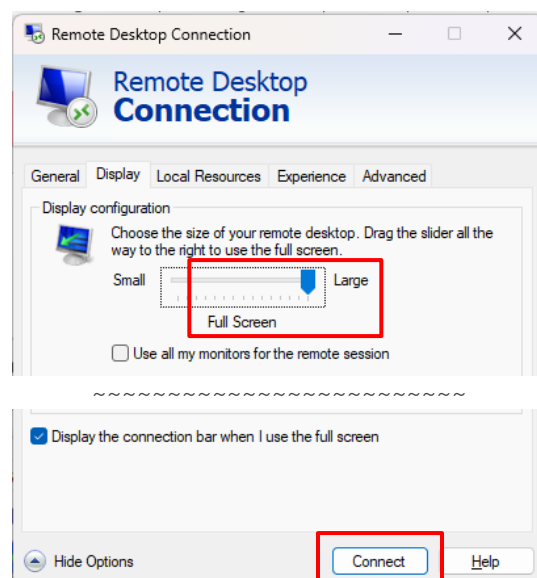
## Windows Remote Desktop

To develop directly on the QCar without the need to connect a separate keyboard and mouse, we suggest using the remote desktop connection. Make sure you have not connected an HDMI dummy dongle or a screen to the QCar 2 or logged in before. It will not open the connection if you have logged into the QCar already.

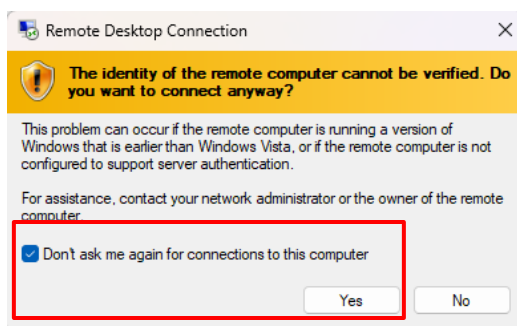
Open **Remote Desktop Connection** application on your windows machine and enter your QCar's IP as the computer (figure 7a). Click show options, go to display and make sure the screen size is set as full screen, you could also restart it and make it smaller if the application starts being too slow (figure 7b). Click connect on that same window. Accept the connection in the follow-up window, you may select "not ask for connections again" to remove this pop up for future connections (figure 7c). And finally, log in with the username and password, both of which are **nvidia** (figure 7d).



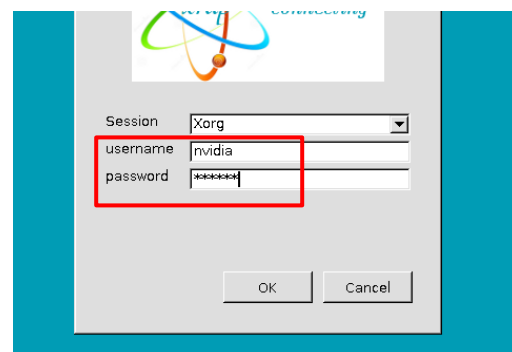
Car IP



Display Options to Full Screen



Accepting the connection



Log in screen

Figure 7. Remote connection to the QCar 2

You should then see a black screen displaying the Quanser Logo in the middle (figure 8a). It can take up to a minute to load and then it will show the QCar 2 desktop as shown in figure 8b. Do all the testing and development needed. You should be able to exit using the top blue bar in the top center of the screen. If it does not show, just hover your mouse around that area and it should come up. Click on x to close the remote connection.



Figure 8a. Quanser Logo while loading

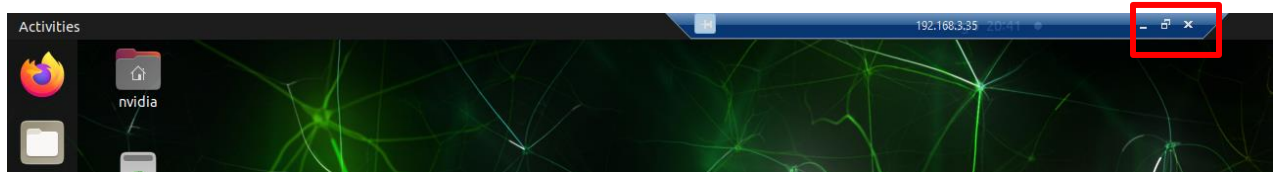


Figure 8b. QCar 2 desktop remote connection

**If stuck at the Quanser screen (figure 8a) for more than two minutes:** if the QCar desktop does not load, that could mean that the QCar desktop is already open in a different session. To avoid this, make sure nothing is connected to the HDMI port of the QCar 2. If you had already connected to it via HDMI, make sure to log out first (figure 9) before trying remote desktop. You could do this through the VNC method or by the direct method (or just restart the QCar without any screen connected to it).

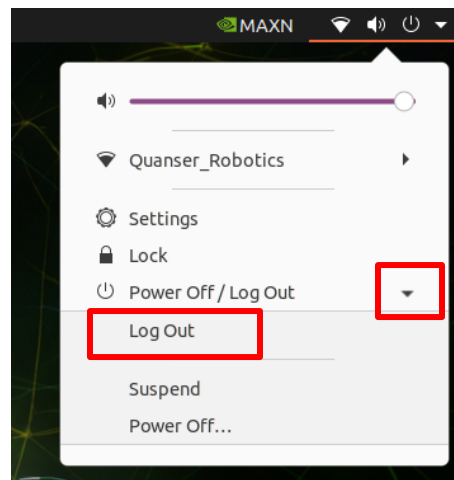


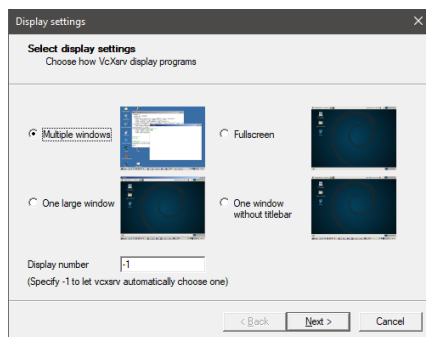
Figure 9. Logging out of the QCar 2

## VcXsrv/XLaunch

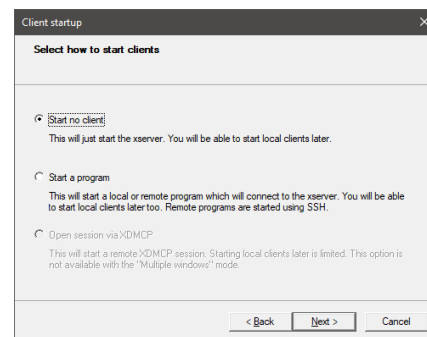
If your code creates any graphical output such as cv2 display windows, an X11 server can be used. The advantage of the X11 interface is that it can be connected to the QCar 2 from a cold boot (see the next section regarding VNC limitations).

A PuTTY terminal can be used to provide X11 forwarding but a display server must be present on the GCS. A tool - **VcXsrv** (under the name **XLaunch** in the start menu), has been installed on the GCS by default for this purpose. Find more information on VcXsrv [here](#). The basics steps are summarized below:

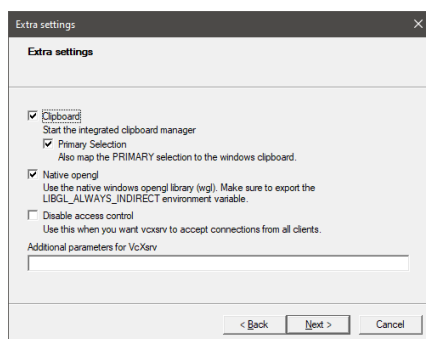
1. Launch **XLaunch** (VcXsrv server application).
2. Pass through all four tabs with default options or set up your own configuration as in Figure 10 (as long as the 'start no client' option is selected as shown in Figure 10b) and click Finish.



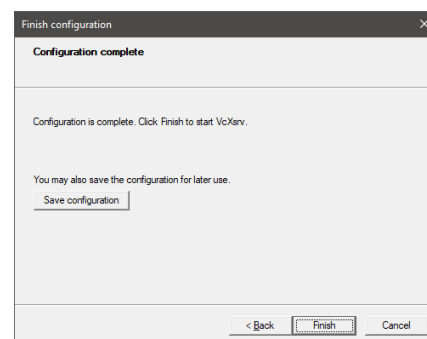
a. Multiple windows default



b. Ensure 'Start no Client' is selected



c. Start the clipboard manager and opengl



d. Save configuration if desired

Figure 10. XLaunch to start a display server on the GCS

3. An XLaunch display server should now be visible in your toolbar as shown in Figure 11.

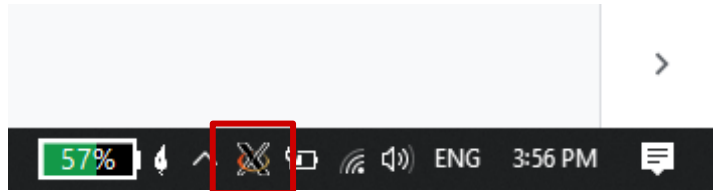
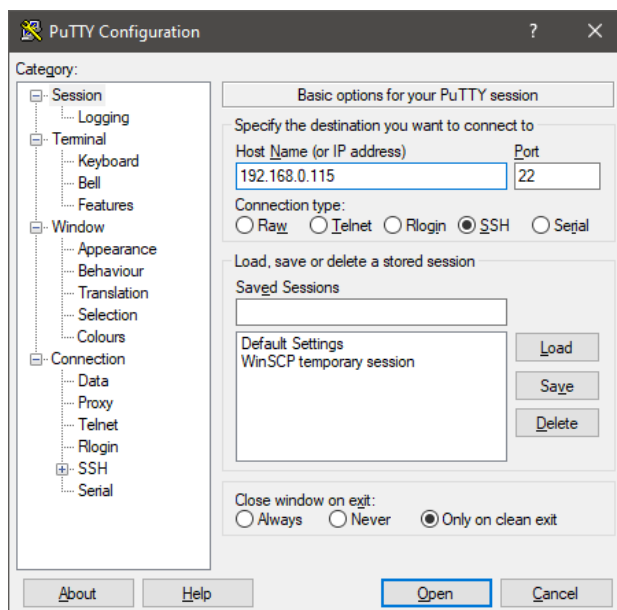
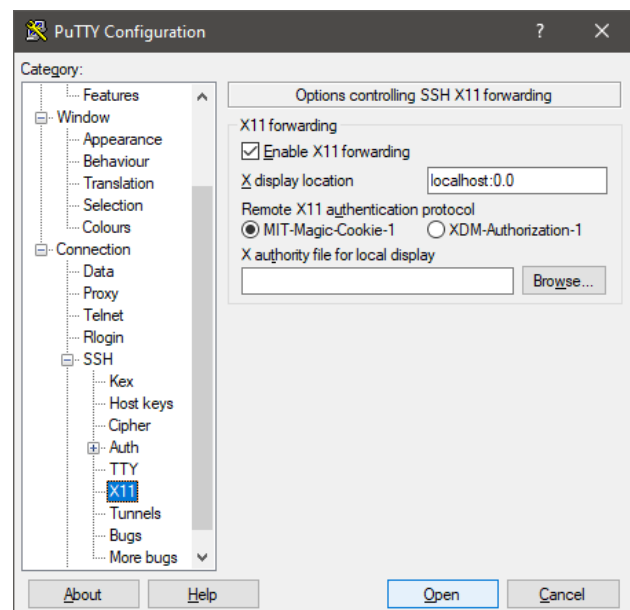


Figure 11. XLaunch display server

4. Launch **PuTTY**.  
**NOTE:** A PuTTY terminal can be launched from WinSCP, but this will not allow you to conveniently set up X11 forwarding. Launch Putty directly for additional set up capabilities.
5. Enter the IP address of the QCar 2 under the Host Name field as shown in Figure 12a. Navigate to the **Connection/SSH/X11** tab as shown in Figure 12b. Check ON the **Enable X11 forwarding** option and enter **localhost:0.0** in the **X display location** field. Click on Open.



a. PuTTY login



b. X11 forward

Figure 12. PuTTY usage to setup a remote terminal

6. Login using the QCar 2 credentials (username **nvidia**, and password **nvidia**).
7. Type the following command to ensure that the display server is set up.  
**>> echo \$DISPLAY**

This should return an output similar to the one shown in Figure 13.

```
quser@qcar-15: ~  
login as: quser  
quser@192.168.0.115's password:  
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.9.140+ aarch64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
* Latest Kubernetes 1.18 beta is now available for your laptop, NUC, cloud  
  instance or Raspberry Pi, with automatic updates to the final GA release.  
  
  sudo snap install microk8s --channel=1.18/beta --classic  
  
* Multipass 1.1 adds proxy support for developers behind enterprise  
  firewalls. Rapid prototyping for cloud operations just got easier.  
  
  https://multipass.run/  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
  
487 packages can be updated.  
40 updates are security updates.  
  
Last login: Fri Feb 21 12:53:02 2020 from 192.168.2.19  
quser@qcar-15:~$ echo $DISPLAY  
localhost:10.0  
quser@qcar-15:~$
```

Figure 13. PuTTY remote terminal connected to a display server

8. Type in a test command,

**>> chromium-browser**

And the chromium application from the QCar 2 should display on your GCS display server in Windows (figure 14). Applications such as python scripts now deployed via the remote PuTTY terminal will forward their display outputs (if any) to the GCS.

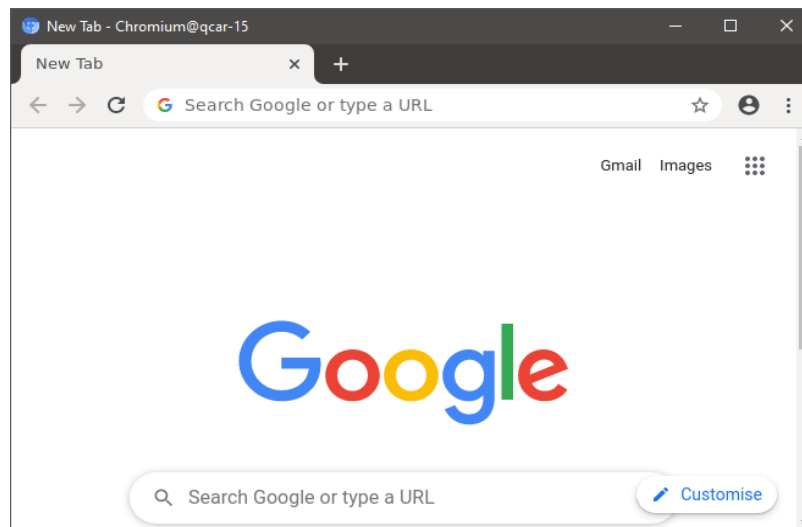


Figure 14. QCar 2's Chromium application launched on GCS

## VNC Viewer

The final method of remote access with the QCar 2 is to use a remote desktop application such as VNC Viewer (if you do not have VNC Viewer installed you may download the application by going [here](#)). This should not be your first method for remotely monitoring image processing outputs but instead use the remote desktop method. However, like the remote desktop method, it also gives you the full graphical desktop which may be easier for users less familiar with Linux commands. The disadvantage of this approach is that the QCar 2 must be first powered up with an HDMI monitor and logged in using a keyboard and mouse before a VNC connection can be established, as well as being the least performant option.

A VNC server has been pre-installed on the QCar 2. To connect from the GCS you will need to perform the following steps:

1. Connect a monitor to the HDMI port on the QCar 2, as well as a keyboard and mouse.
2. Connect a power source to the QCar 2, this can either be the provided LiPo battery or the supplied power supply.
3. Power on the QCar 2.
4. Using the connected screen, log into the QCar using the username and password **nvidia**. Now that you are logged in, you can disconnect the keyboard, mouse and the monitor from the QCar.
5. From the GCS, open VNC viewer.
6. In the VNC connect window enter the IP address of the QCar 2 which can be found on the LCD screen of the QCar and click enter. A connection screen will pop up as shown in figure 15.

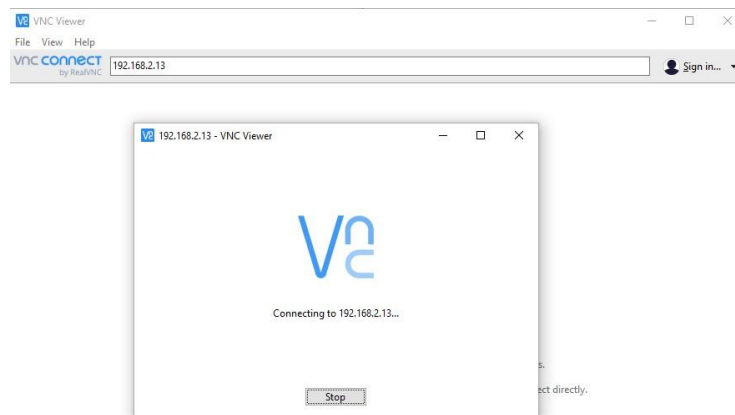


Figure 15. VNC Viewer connection screen.

**Note:** If you want the QCar 2 to automatically log into the desktop when you connect the HDMI dummy dongle, you will need to connect to the QCar, remotely or directly, and enable it under Settings/User/Automatic Login, as shown in figure 16. Doing so will **disable** remote desktop login unless you manually log out of the car every time you want to use remote desktop.

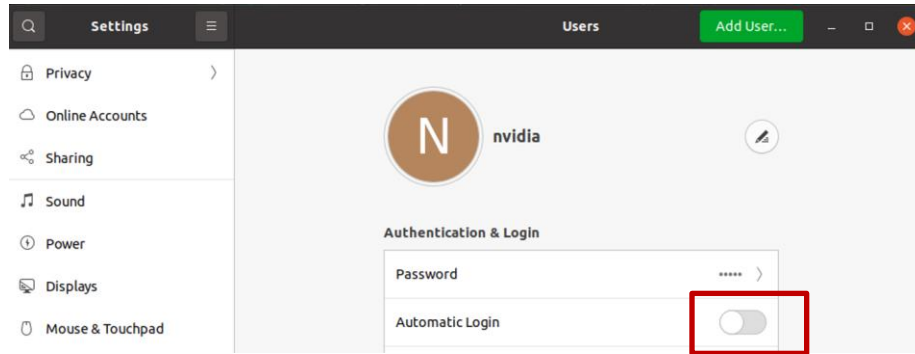


Figure 16. VNC Viewer connection screen.

7. When prompted to allow the remote desktop connection, click on yes.
8. You will be asked for the user password for the QCar 2 to start the remote desktop connection, as shown in figure 17. You may optionally check the box to 'Remember password'.

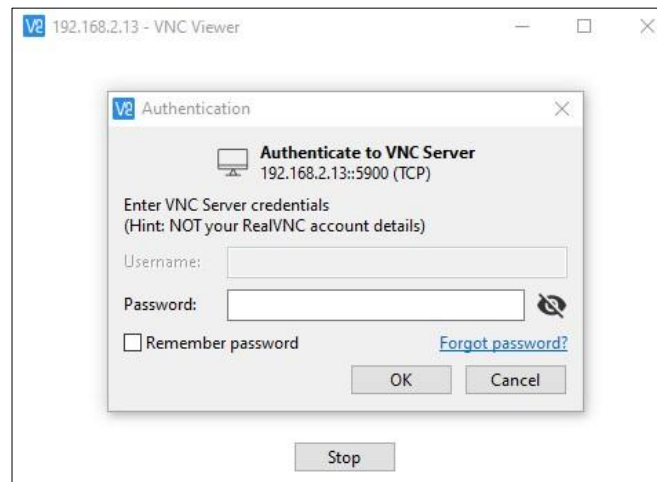


Figure 17. VNC Viewer QCar 2 credential screen.

9. A successful connection will take you to the main user desktop as shown in figure 18.



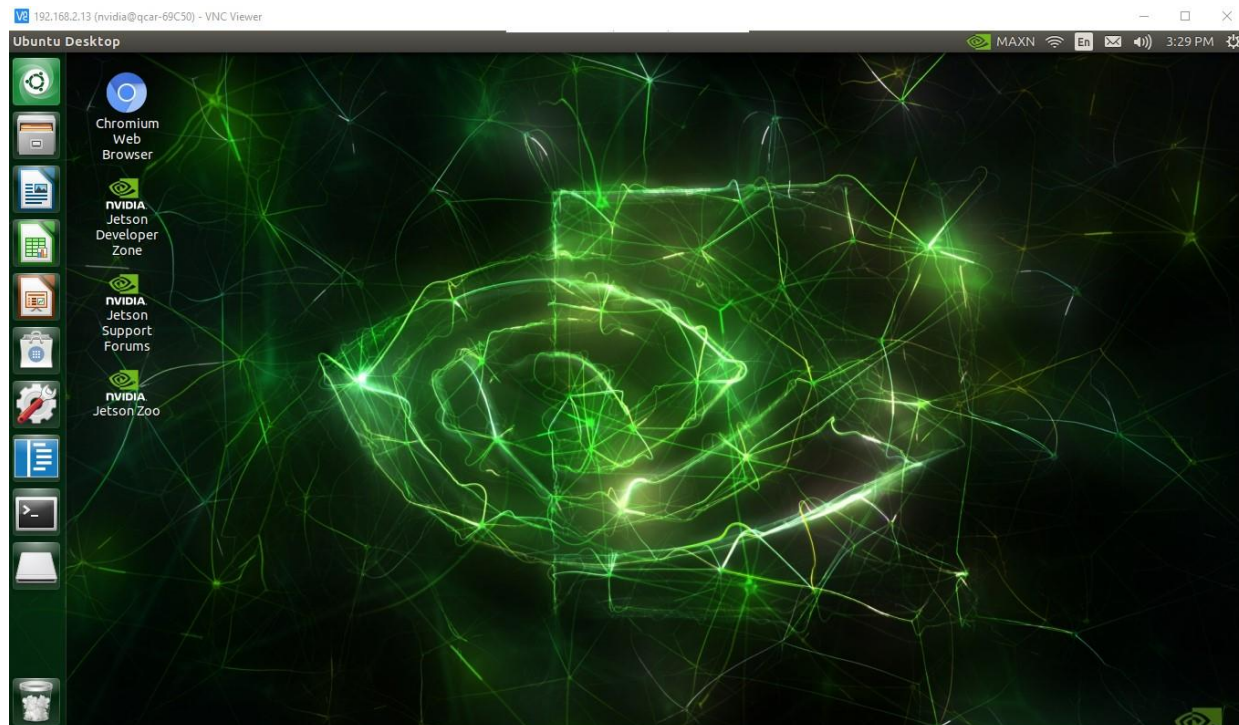


Figure 18. QCar 2 Ubuntu user screen

Note, an Ethernet connection will provide a more responsive interface than Wi-Fi. You can also try reducing the desktop resolution, and Appearance Behavior from High visual effects to Low visual effects to improve response time. You can select which combination of features, desktop space, and interface responsiveness work best for your workflow. A resolution of 800x600 should produce a very responsive display over Wi-Fi. Recommended setting is shown in figure 18.

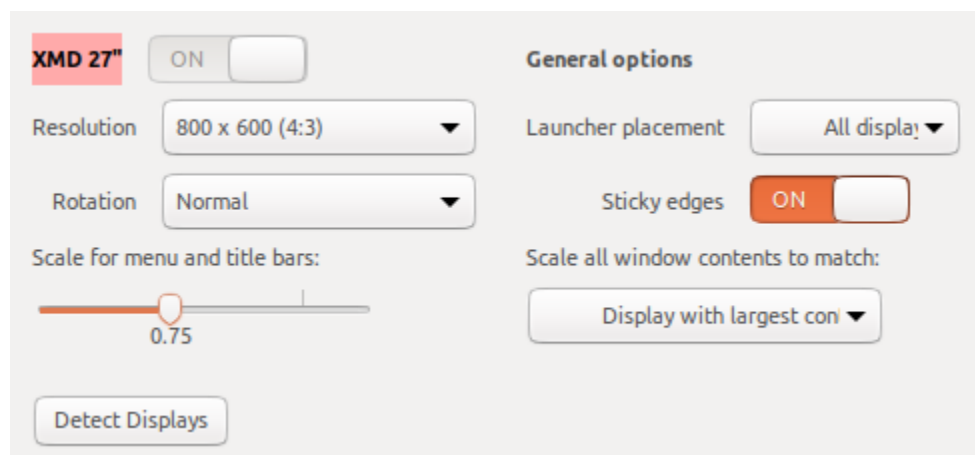


Figure 18. Setting the display settings in Ubuntu to 800 x 600 will produce a responsive display.

© Quanser Inc., All rights reserved.



Solutions for teaching and research. Made in Canada.