# 1 Introduction

The Quanser QBot 3 (Figure 1.1) is an innovative autonomous ground robot system incorporating a robust educational ground vehicle with the Intel RealSense D415 and a QUANSER® embedded target. The QBot 3 comprises of a Yujin Robot Kobuki platform, an Intel RealSense D415 camera and depth sensor, and a wireless embedded computer (also refered to as the target computer). This system mounted on the vehicle uses the Raspberry Pi 4 Model B computer [1] to run QUARC®, Quanser's real-time control software, and interface with the QBot 3 data acquisition card (DAQ).

Figure 1.1: The Quanser QBot 3

The interface to the target computer is MATLAB® SIMULINK® with QUARC. The QBot 3 is accessible through three different block sets: the Quanser Hardware-In-the-Loop (HIL) block set to read from sensors and/or write to outputs, the Quanser Stream API blockset to perform communications over wired and wireless communication channels, and the Quanser Multimedia blockset to read RGB and depth image data from the Intel RealSense D415 sensor. Controllers are developed in Simulink with QUARC on the host computer, and these models are cross-compiled and downloaded to the target (Raspberry Pi 4 Model B[1]) seamlessly. A diagram of this configuration is shown in Figure 1.2.

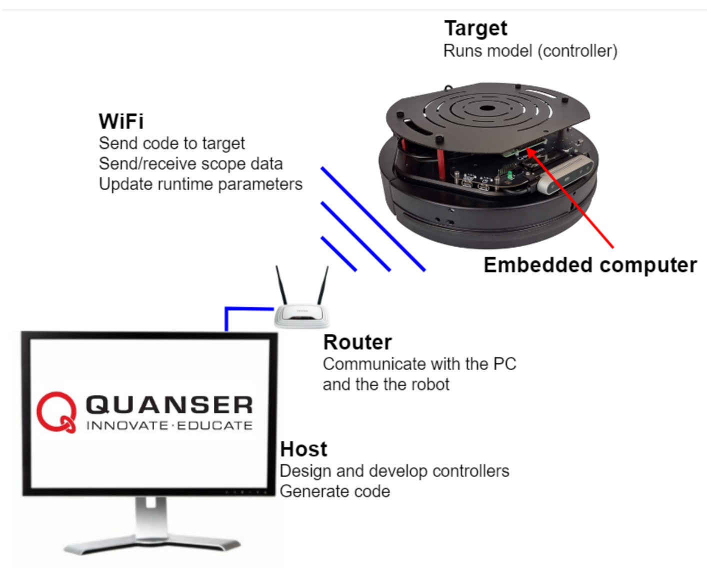



Figure 1.2: Communication Hierarchy

The general system description, component nomenclature, specifications, and model parameters are all given in Section 2. Section 3 goes into detail on how to setup the QBot 3. Lastly, Section 4 contains a troubleshooting guide.

## 1.1 Prerequisites

To successfully operate the QBot 3, the prerequisites are:

1. To be familiar with the components of the QBot 3.
2. To have QUARC 2022 or later installed and properly licensed.
3. To be familiar with using QUARC to control and monitor the vehicle in real-time. See Reference [2] for more details.

## 1.2 References

[1] Raspberry Pi 4 Model B: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/
[2] QUARC User Manual (type `doc quarc` in MATLAB to access)

# 2 Components

The QBot 3 is made up of four main components: the Kobuki robot platform, the QBot 3 PCB, the Raspberry Pi 4 Model B embedded computer, and the Intel RealSense D415 sensor. This section outlines these components in more detail.

## 2.1 The Kobuki Robot Platform

The QBot 3 uses an Kobuki mobile robot platform (Figure 2.1). The QBot 3 follows the Quanser standard for body frame axes, where the x-axis is in the forward direction, the y-axis is to the left, and the z-axis is up. The diameter of the vehicle is $35$ cm, and its height (without attachments) is $10$ cm. The Kobuki is driven by two differential drive wheels with built-in encoders. The Kobuki comes with a bumper sensor as well as a built-in gyroscope and cliff sensors. The embedded computer target can access data from these sensors.

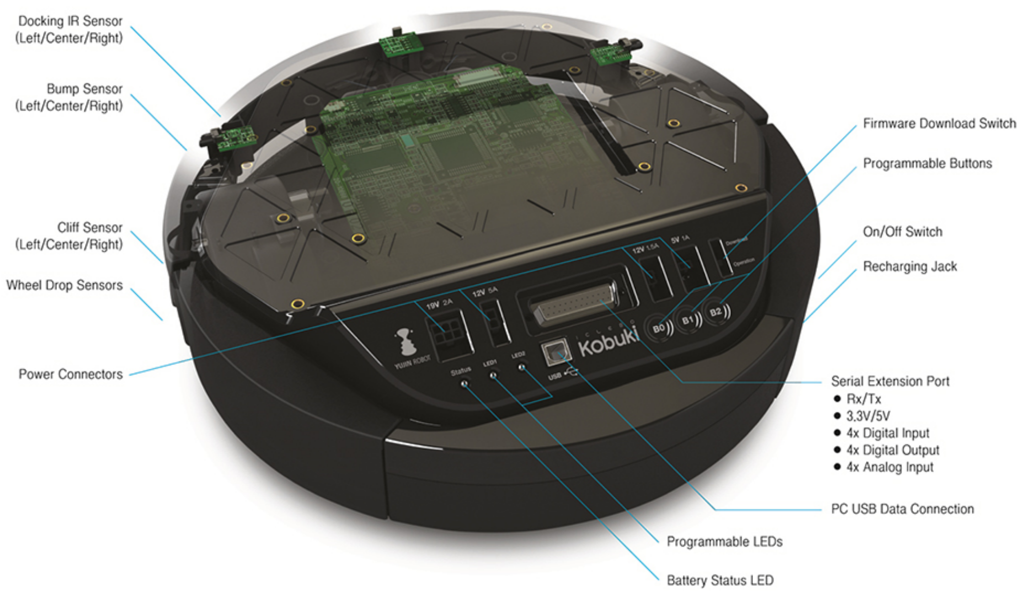
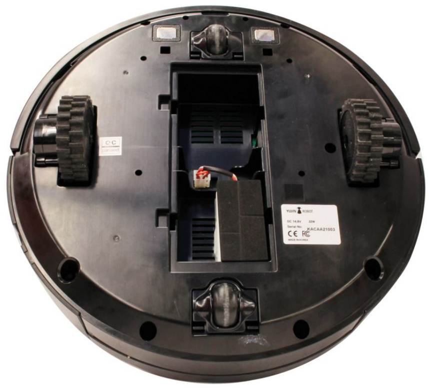



Figure 2.1: The Kobuki mobile robot platform

The QBot 3 is powered by a Lithium ion battery pack (Figure 2.2a) provided by Yujin Robot. The battery fits underneath the QBot 3, and can last continuously for about $3$ hours after a full charge. The battery takes less than $3$ hours to charge. While charging, the power light pulses slowly with an yellow colour, turning to green when fully charged. A battery charger is provided (Figure 2.2b). To recharge the QBot 3, plug in the battery charger and connect it to the charger input port on the Kobuki base next to the *ON/OFF* switch (Figure 2.2c).

**Note:** The QBot 3 must be fully charged before using it for the first time.



(a) The QBot 3 battery

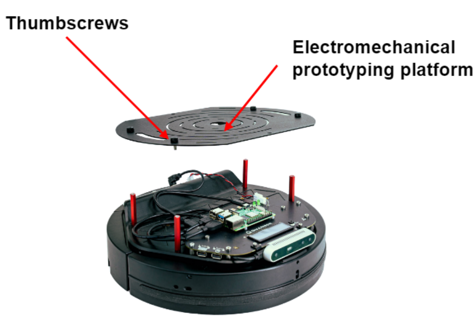
(b) The QBot 3 charger


(c) The QBot 3 charger input

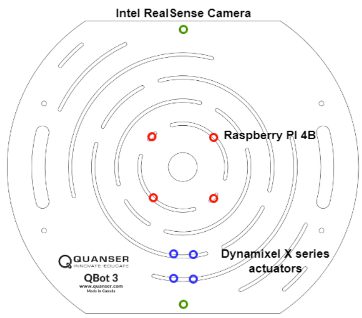
Figure 2.2: The QBot 3 battery and charger input

## 2.2 Electromechanical Prototyping Platform

The QBot 3 includes an electromechanical prototyping platform (Figure 2.3a) that can also be used as a landing pad for UAVs such as the Quanser QDrone. The platform includes hole patterns for numerous standard components including additional Raspberry Pi, Dynamixel X series actuators, Realsense cameras, RP-Lidars, Optitrack markers, etc. (Figure 2.3b). Remove the platform via the 4 thumbscrews to access the embedded computer if needed.




(a) Removing the prototyping platform




(b) Hole Patterns

Figure 2.3: QBot 3's Electromechanical prototyping platform

## 2.3 The Intel RealSense D415 Sensor

The Intel RealSense D415 sensor is an integrated RGB camera and depth sensor used in a variety of experiments (Figure Figure 2.4). It provides RGB image capture at up to 30 fps at 1080p and stereoscopic depth sensing at up to 90 fps at 720p. It's depth sensor utilizes infrared light and has a range of $0.45$ m to $12$ m. Due to the type of infrared sensor, the camera should only be used indoors in locations without direct sunlight for best results.

Figure 2.4: The Intel RealSense D415 sensor

The Intel RealSense D415 is mounted on the QBot 3 to be oriented horizontally. The Intel RealSense D415 can also be relocated to the top of the QBot 3 using holes provided at the front and rear end of the electromechanical prototyping platform (Figure 2.3b).

## 2.4 Embedded Computer and PCB

The QBot 3 uses a Raspberry Pi 4 Model B as a small-scale embedded computer that runs the QUARC runtime. This computer interfaces with the Kobuki as well as the Intel RealSense D415. With QUARC installed, code generated from MATLAB Simulink is cross-compiled, downloaded, and executed directly on the Pi. As shown in Figure 2.5, the Raspberry Pi 4 is mounted on a PCB which itself is mounted on the Kobuki chassis base. The Raspberry Pi 4 also comes with integrated 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN to allow wireless connection between the target Pi and the host computer and/or other vehicles.

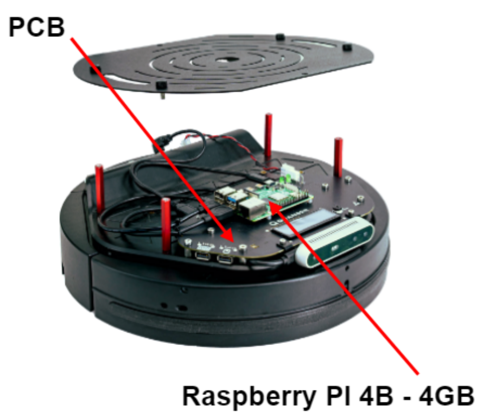



Figure 2.5: The Raspberry Pi Model 4B mounted on the QBot 3 PCB

Below are the full specifications of the Raspberry Pi 4 Model B:

- Broadcom BCM2711, Quad-core Cortex-A72 (ARMv8) 64-bit SoC @ 1.5GHz
- 4GB LPDDR4-3200 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 5.0, BLE
- Gigabit Ethernet

- Extended 40-pin GPIO header
- 2 micro-HDMI ports (up to 4kp60 supported)
- 2 USB 3.0 ports
- 2 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading the operating system and storing data
- 5V/3A DC power input over USB-C
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

There are 5 connections from the PCB/Raspberry Pi 4B that must be connected for full functionality. These are displayed in Figure 2.6. These connections are,

1. PCB to Kobuki base via USB 2.0 for data
2. PCB to Kobuki base for power
3. PCB to Raspberry Pi 4B via USB 2.0 for data
4. PCB to Raspberry Pi 4B via USB 2.0 for power
5. Raspberry Pi 4B to Intel RealSense D415 via USB 3.0 for data

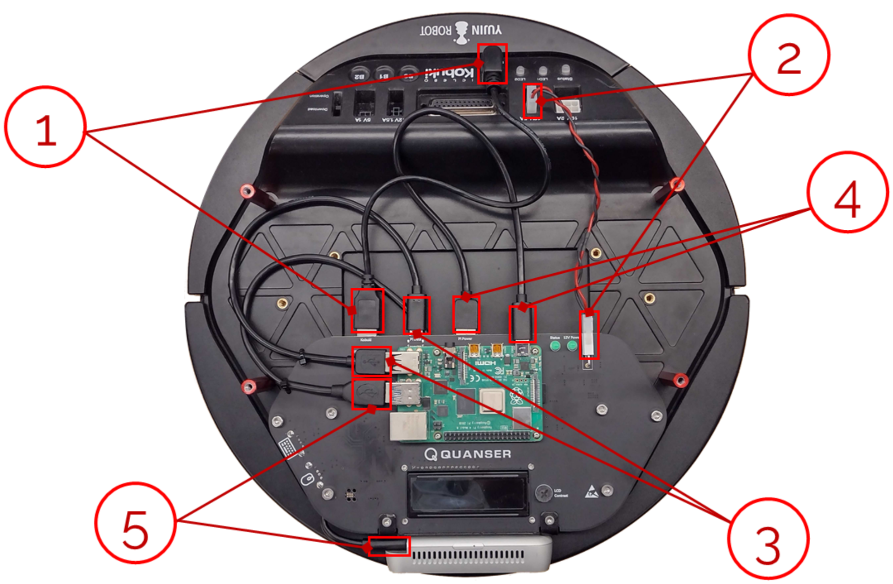



Figure 2.6: The QBot 3 PCB

For further information on the Raspberry Pi 4 Model B refer to [1]. Note that one of the USB 3.0 ports is used for the Intel RealSense D415. A USB 2.0 port is used for communications with the PCB (Pi Data label on PCB). 2 additional USB 2.0 ports are provided on the PCB near the front of the QBot 3.

## 2.5 LCD

The QBot 3 uses a user-programmable 32-character LCD that is mounted on the PCB (16 characters per line). Generally, it displays the current battery voltage, as well as IP address associated with the current network connection, unless your applications are writing custom messages. The LCD displays a variety of standard messages as described below,

- qbot3-XXXXXX starting up
  This is the default starting message after the LCD briefly displays QUANSER INNOVATE EDUCATE.
- Raspberry Pi disconnected
  This indicates a communication issue between the embedded computer and PCB. Check the Pi Data and Pi Power USB cables connecting the embedded computer to the PCB. Cycle the power if the cables are fine and the Raspberry Pi is still not responding.
- Writing to LCD too fast
  This indicates that the user application is attempting to write to the LCD faster than supported.
- Supply voltage out of range
  This indicates a problem with the Kobuki 12V power supply. Try charging the Kobuki battery. If this does not resolve the problem, contact Quanser tech support. See Section 5 for more details.
- Shutting down
  This message is displayed when the embedded computer is shutting down correctly.
- Safe to turn off
  It is now safe to turn off the QBot 3 using the power switch.

## 2.6 Model Parameters

Table 2.1 lists the main parameters associated with the QBot 3.

| Symbol | Description | Value | Unit |
|---|---|---|---|
| $D$ | Diameter of the QBot 3 | $0.35$ | m |
| $d$ | Distance between the left and right wheels | $0.235$ | m |
| $h$ | Total height of the QBot 3 | $0.16$ | m |
| $\nu_{max}$ | Maximum speed of the QBot 3 | $0.7$ | m/s |
| $m$ | Total mass of the QBot 3 | $3.48$ | kg |

Table 2.1: QBot 3 specifications

# 3 Setup

The QBot 3 package comes with a pre-configured wireless router and automatically connects to the WiFi network Quanser_UVS-5G. It uses TCP/IP connection for communicating with the host computer and/or other Quanser unmanned vehicles. The Host PC and each of the vehicles must have unique IP addresses and the range of these addresses are defined below:

| | |
|---|---|
| Host PC(s) | 192.168.2.2 to 192.168.2.10 |
| Reserved for DHCP Connections | 192.168.2.11 to 192.168.2.254 |

Table 3.1: Valid Host PC and QBot 3 IP ranges

These Steps outlined below for setting up the host computer wireless connection only need to be performed once.

1. Power up and turn on the wireless router.
2. After turning on the router that is provided, wait for about 60 seconds for the wireless network to establish.
3. Connect the battery inside the Kobuki robot.
4. Press the power switch shown in Figure 2.2c. This should turn on both the vehicle and PCB, in turn powering the Pi computer. The QBot 3 will automatically connect to the provided router using the **Quanser_UVS-5G network**.
5. Connect your PC network card to any of the ports of the router (e.g. port number 1 to 4) using the network cable provided (you can also connect to the **Quanser_UVS-5G** wireless network if your computer has wireless adapters, however, wired connection between your PC and the router is preferred for better performance). If you choose wireless connectivity between your PC and the router, you should use the password **UVS_wifi** to connect to the Wi-Fi network.

   **Note:** You may need to disable Windows firewall to establish a connection.

6. Launch a command prompt and ensure that you can ping the router by using the command `ping 192.168.2.1`. If the connection to the router is successful you will see the ping replies in the command window. If you cannot ping the router, check network connectivity and your IP address before going to the next steps.
7. With the QBot 3 turned on, the LCD will display the IP address of the QBot. From the command prompt, ping the QBot 3.
8. Deploy applications as needed.
9. To turn OFF the QBot 3, download PuTTy here. Once installed, browse to the `Technical Resources/Remote Shutdown` directory in the provided content. Right-click the `shutdown.bat` file and click edit. Modify the IP address `192.168.2.xxx` to match that displayed on the LCD of your QBot 3. Save and close the file. Double-click on the `shutdown.bat` file to execute a remote shutdown. Wait for the LCD to display a message indicating it is safe to use the power switch. Check the LCD subsection under Section 2.

   **Note:** Alternatively, launch PuTTy and connect to the Raspberry Pi 4B using its IP address as the hostname. This is launch a terminal asking for credentials. Use `pi` (username) and `QuanserPi4` (password). Once logged in, use the command `sudo shutdown now` in the terminal to shut the embedded computer off. Wait for the LCD to display a message indicating it is safe to use the power switch. Check the LCD subsection under Section 2.

   **Note:** Alternatively, connect the QBot 3 to a monitor, keyboard and mouse. The login credentials for the QBot 3 are `pi` (username) and `QuanserPi4` (password). Once logged in, shutdown the Raspberry Pi 4 correctly. Wait for the LCD to display a message indicating it is safe to use the power switch. Check the LCD subsection under Section 2.

**Note:** Do NOT use the power switch on the Kobuki to turn OFF the QBot 3 directly. This can cause loss of data, SD card corruption, and an inability to operate the QBot 3.

# 4 Troubleshooting

For any issue, the first and easiest troubleshooting solution on any electronic device is to reboot the device. Use VNC or PuTTy to launch a session to access the QBot 3 and execute the sudo reboot now command. For more details on connecting to the QBot 3 using PuTTy, see the Section 3 section. For troubleshooting any problem with the QBot 3, it is always a good idea to open the QUARC® console in case additional information is printed to the console by going to the QUARC® menu and clicking on Console for all.... The console must be opened after the QBot 3 has booted and established a WiFi connection. If the console is opened successfully it establishes a connection to the target and the console window has the title `QUARC Console for * at tcpip://192.168.2.xxx:17000`, where *xxx* corresponds to the IP address of the QBot 3.

If you are still unable to resolve the issue after reading through this section, contact tech@quanser.com for further assistance.

**Q1** You cannot ping the QBot 3

Make sure the router is on and the WiFi light on the router is on. Check that the network adapter of the host PC is connected to the router (or the wireless network Quanser_UVS or Quanser_UVS-5G) and is configured according to the setup procedure outlined in this manual (Section Section 3). Verify that you can successfully ping the QBot 3.

□□□

**Q2** The model fails to build/connect or the QUARC console does not successfully open

Turn on the QBot 3 and verify that the LED labeled PWR on the Raspberry Pi 4 Model B is lit and has turned solid red. This ensures that power is being supplied to the Raspberry Pi. If the LED does not turn on, check the that the USB-A socket on the PCB labelled Pi Power is securely connected to the USB-C connector on the Raspberry Pi 4 Model B using the provided cables. Also, check that the network adapter of the host PC is connected to the router (or the wireless network Quanser_UVS or Quanser_UVS-5G) and is configured according to the network configuration procedure outlined in this manual (Section 3). Verify that you can successfully ping the QBot 3.

□□□

**Q3** The Simulink model appears to run slowly (i.e., the simulation time runs slower than actual time), or the console displays the message *Sampling rate is too fast for base rate*

(a) The maximum sample rate recommended for the QBot 3 is $1000$ Hz (sample time $0.001$ s). However, if there are complex calculations (such as image processing) performed within the model, then this could potentially limit the sample rate of the model (suggested sample rate of 100 Hz or 50 Hz). Try reducing the model sample rate in the menu *QUARC* | *Options* | *Solver* by increasing the *Fixed-step size (fundamental sample time)* parameter.

(b) To determine the execution time of blocks or subsystems within the model, use the Computation Time block found in the *QUARC library* | *Sources* | *Time*. This block outputs the computation time of a function call subsystem, measured using an independent high-resolution time source. Blocks can be placed inside a function call subsystem and connected to the *Computation Time* block to determine their execution time during each sample instant. This helps identify the bottlenecks in the model (blocks/subsystems with the highest execution time) and can identify blocks/subsystems whose computation time is greater than the sample time of the model. Try increasing the sample time of those blocks whose computation time is greater than the sample time of the model so that the blocks run in a slower rate thread.

(c) If you are using image processing blocks, ensure that they are only executed when there is new image data available or at a slower rate than the image acquisition rate. Otherwise, the embedded computer is spending time processing image data that has already been processed, which can cause the model's computation time to increase undesirably. For complex image processing, consider lowering the rate of the image processing blocks and/or downsampling the source images to a lower resolution to speed up processing time.

□□□

**Q4** MATLAB returns an error ‘Model failed to download on target... The code being downloaded or run is not compatible with the type of target referenced by the target’ when building the model.

This error is returned by MATLAB when there is a mismatch between the ‘Target type’ referenced in ‘QUARC | Preferences’ and the actual target type. Ensure that ‘Target type’ is set to *linux_pi_4*.

# 5 Technical Support

To obtain support from Quanser, go to http://www.quanser.com/ and click on the Tech Support link. Fill in the form with all the requested software and hardware information as well as a description of the problem encountered. Also, make sure your e-mail address and telephone number are included. Submit the form and a technical support person will contact you.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

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