Specialty Plants

User Manual

Quanser Qbot

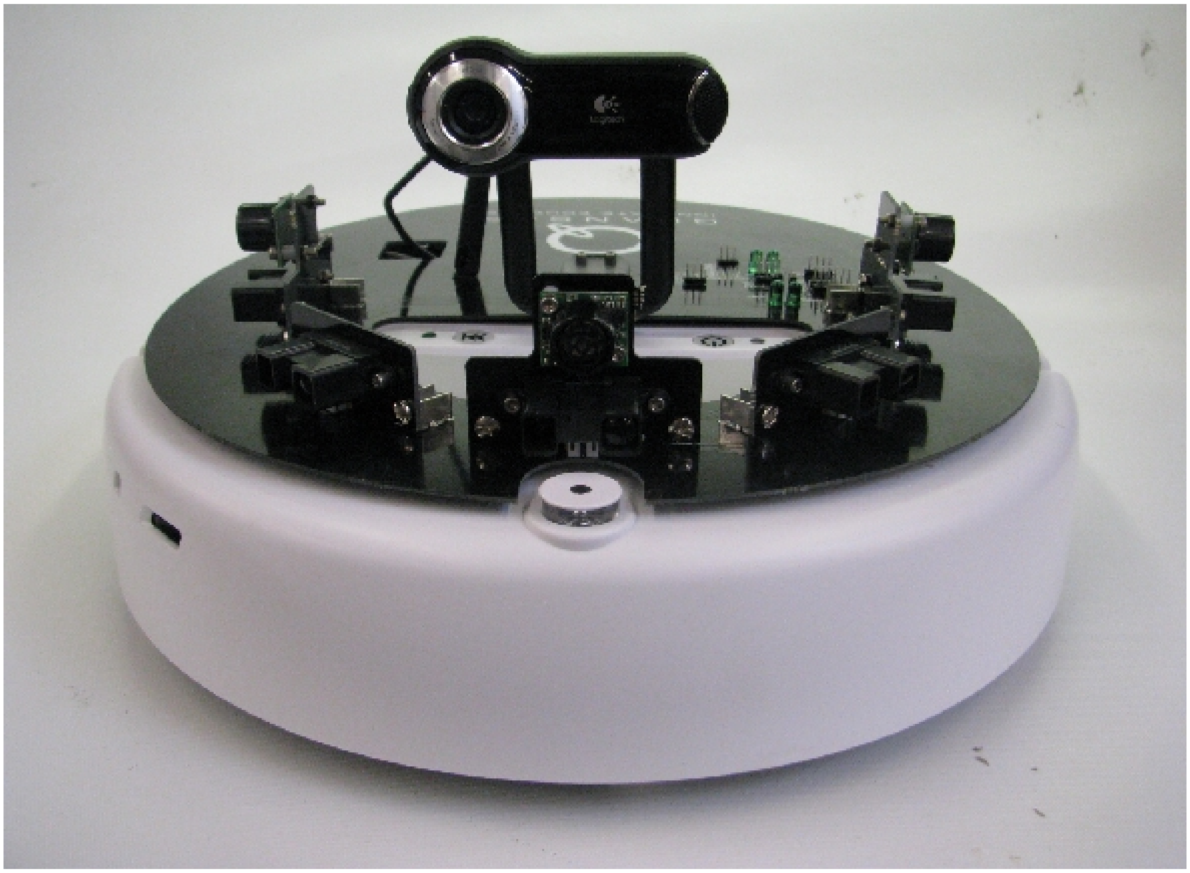


Table of Contents

1. Introduction

The Quanser Qbot (Figure 1) is an innovative autonomous ground robot system incorporating a robust educational ground vehicle with the Quanser Controller Module (QCM). The vehicle is comprised of a iRobot Create® robotic platform, an array of optional infrared and sonar sensors, and a Logitech Quickcam Pro 9000 USB camera. The QCM is an embedded system mounted on the vehicle, which uses the Gumstix computer [2] to run QUARC, Quanser's real-time control software, and the Qbot data acquisition card (DAC).

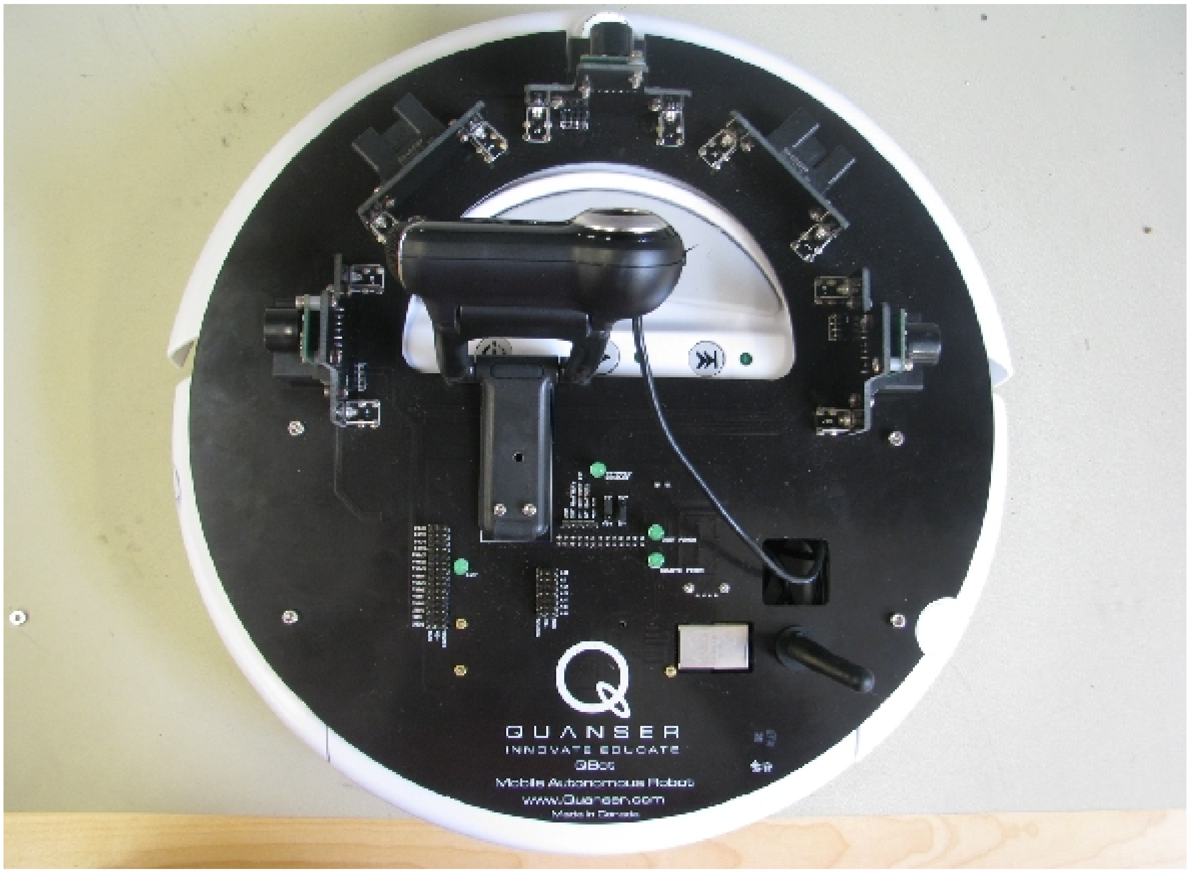


Figure 1: The Quanser Qbot - top view

The interface to the QCM is MATLAB Simulink with QUARC. The Qbot is accessible through three different block sets: the Roomba block set to drive the vehicle, the HIL block set to read from sensors and/or write to servo outputs, and finally the OpenCV block set to access the camera. The controllers are developed in Simulink with QUARC on the host computer, and these models are downloaded and compiled into executables on the target (Gumstix [2]) seamlessly. A diagram of this configuration is shown in Figure 2.

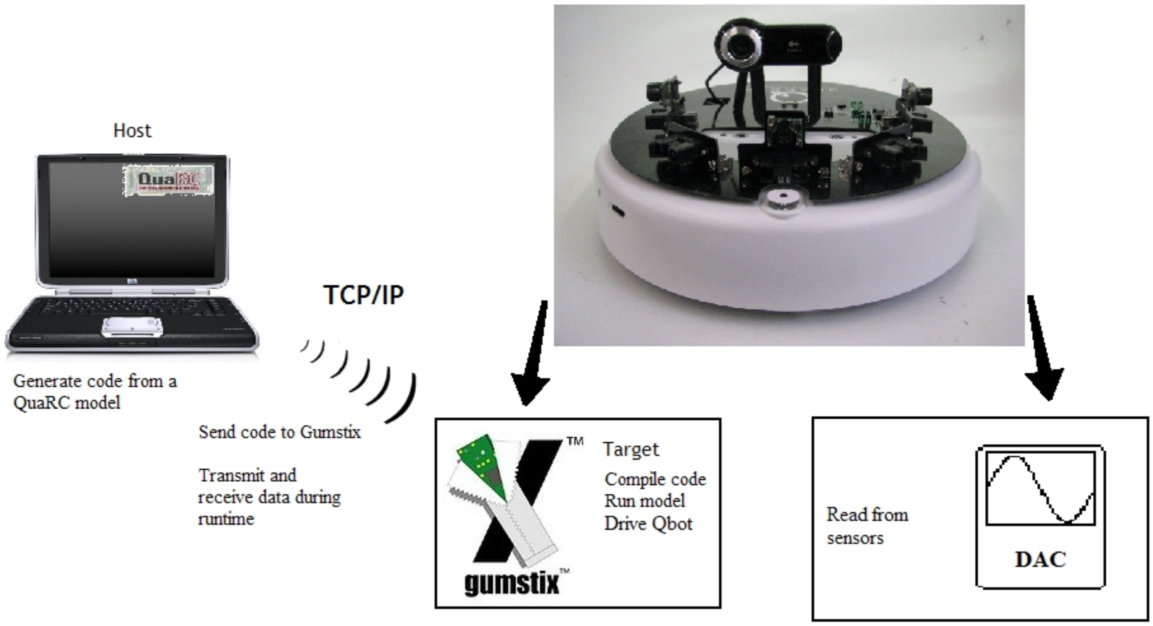


Figure 2: Communication Hierarchy

Section 2 goes through the laboratory requirements, and Section 3 lists various documents that are referenced in this manual. The general system description, component nomenclature, specifications, and model parameters are all given in Section 4. Section 5 goes into detail on how to setup the Qbot. Lastly, Section 6 contains a troubleshooting guide.

2. Prerequisites

To successfully operate the Qbot, the prerequisites are:

i) To be familiar with the components of the Qbot.

ii) To have QUARC version 2.2 or later installed (**NOTE: You must install the beta components by selecting them during installation, since some blocks used with the Qbot are found in the QUARC beta library**) and properly licensed.

iii) To be familiar with using QUARC to control and monitor the vehicle in real-time, and in designing a controller through Simulink. See Reference [2] for more details.

3. References

[1] *Gumstix:*
http://gumstix.com/

[2] *QUARC User Manual* (type `doc quarc` in Matlab to access)

[3] *iRobot Create Open Interface* Specification Sheet
[4] *iRobot Create Owner's Guide*
[5] *SHARP 2Y0A02 IR Range Sensor.pdf*
[6] *LV-MaxSonar-EZ0-Datasheet.pdf*

4. System Hardware Description

4.1. Qbot Ground Vehicle

4.1.1. The iRobot Create®

The Qbot uses an iRobot Create® frame (Figure 3). The Qbot 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 34 cm, and its height (without camera attachment) is 7 cm. and it is driven by two differential drive wheels. The iRobot Create® comes with a bumper sensor and an omni-directional infrared receiver. The QCM can access data from these sensors.

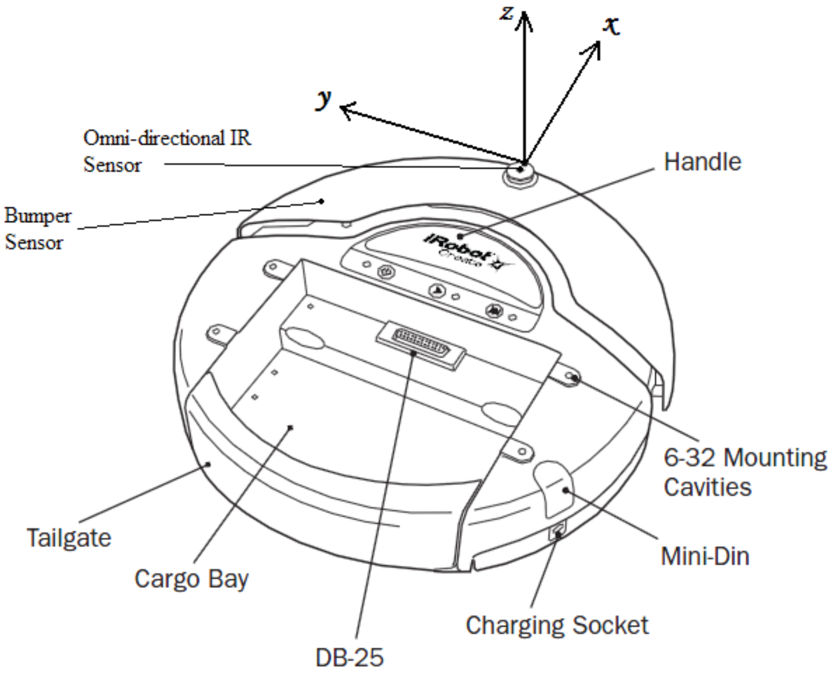


Figure 3: Anatomy of the Qbot, showing various components and body axes

The Qbot is turned on by pressing the power button (more on Section 5.1). Figure 4 below shows the buttons that are used to operate the iRobot Create®. The Play and Advance buttons are for built-in demos for the vehicle, and are not necessary for Qbot purposes.

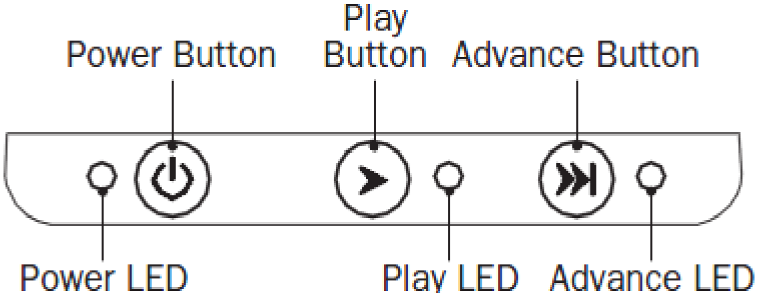


Figure 4: Buttons on the Qbot Frame

4.1.2. The Printed Circuit Board (PCB)

The printed circuit board (PCB) contains the wiring and circuitry for the Qbot. It also acts as a cover for the DAC and Gumstix, and mounts for infrared sensors, sonar sensors, and webcam. Figure 5 shows the accessible pins for the user. In particular, the DIO, PWM output, and analog input pins have been labeled for clarity.

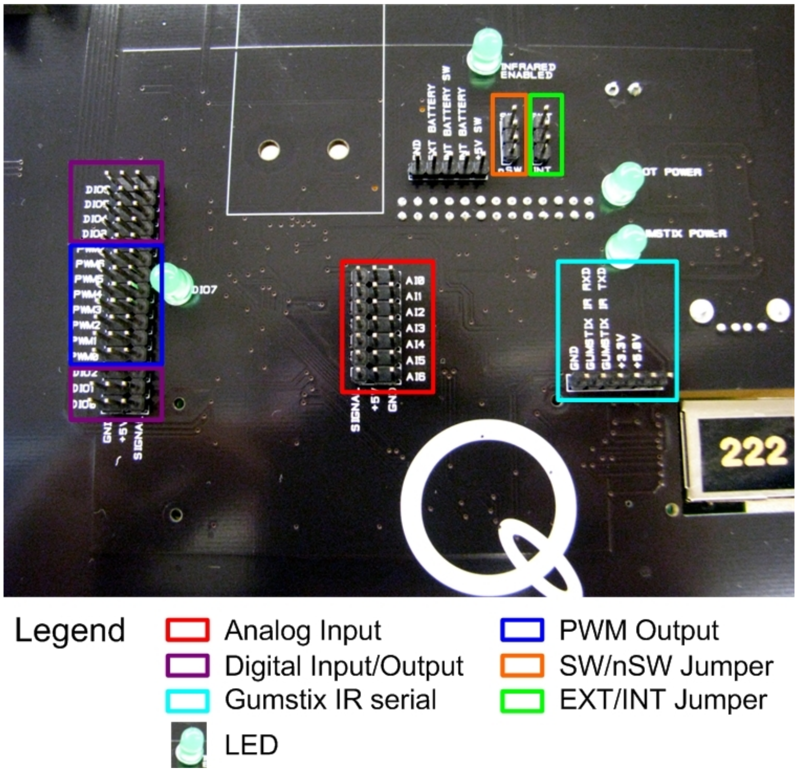


Figure 5: Qbot PCB showing available pins for the PWM output, Digital Input/Output and Analog Input pins, serial pins, and jumpers for INT/EXT power

4.1.2.1. Digital Input/Output Pins (DIO #)

The DIO channels (0 to 6) are set as inputs by default. The DIO channels need to be configured as either inputs (or outputs, but not both) using the HIL Initialize block. Also, if an output needs to be in a known state on power up, it is recommended that a 10k resistor is put from the I/O to 5V or GND as needed.

There is a final digital channel (7) that is a fixed output, and it is represented by an LED labeled DIO7.

4.1.2.2. Gumstix IR Serial Pins

The Qbot provides a TTL serial connection to the Gumstix IR serial port (port number 2). The serial port consists of ground (GND), receive (GUMSTIX IR RXD), transmit (GUMSTIX IR TXD), and power (+3.3V or +5.0V) pins. The serial port is accessed through the QUARC Stream blockset or Stream API. For more information on accessing serial data see the QUARC help QUARC Targets/User's Guide/Communications.

4.1.2.3. SW/nSW and INT/EXT Jumpers

The INT/EXT jumper switches the Qbot from internal power from the iRobot Create battery (INT) and an external battery power supply (EXT). Note that no external battery is supplied with the Qbot, so this jumper should be left in the INT position to power the Qbot. When the power source jumper is in the INT position, the SW/nSW jumper indicates whether the iRobot Create must be switched on (SW) for the Qbot to receive power, or whether the Qbot should always draw power even when the iRobot Create is off (nSW).

4.2. Detailed Description of Qbot Components

4.2.1. Qbot DAC

The Qbot DAC is a data acquisition board, capable of receiving analog inputs and other inputs (for sonar sensors). It is also capable of writing PWM outputs for possible servo actuators. It is located underneath the black cover of the Qbot, and it is not intended for it to be physically accessed.

4.2.2. USB Camera

The Logitech Quickcam Pro 9000 USB camera is mounted on top of the Qbot (Figure 6). The QUARC block set that uses the Open Source Computer Vision library (in Section 5.3) allows the user to capture and display images in real time, process them, and save them in a disk for future analysis.

Figure 6: Camera

4.2.3. Gumstix

The Gumstix is a small-scale, fully functional, open source computer. With QUARC installed, code generated from MATLAB is downloaded, compiled, and executed directly on the Gumstix.

The Gumstix motherboard is connected directly to the Qbot DAC. The Gumstix also comes with the Wifi attachment board to allow wireless connection between the target Gumstix and the host computer and/or other vehicles.

The IP of the Gumstix will be provided.

4.2.4. Battery

The Qbot is powered by the Advanced Power System (APS) Battery (Figure 7) provided by iRobot. The battery fits underneath the Qbot (Figure 8), and can last continuously for about 2 hours after a full charge. The Qbot's power light indicates the power level of the battery. The light is green when the batteries are fresh, then gradually turns red as the battery discharges. The battery takes less than 3 hours to charge. While charging, the power light pulses slowly with an orange colour. A battery charger will be provided.

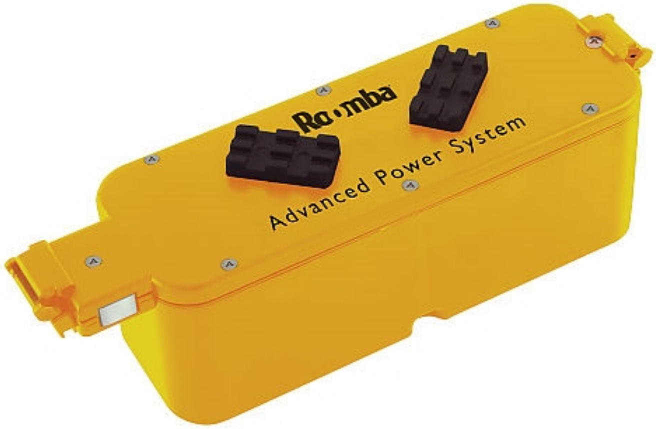


Figure 7: The Qbot Battery, the Advanced Power System

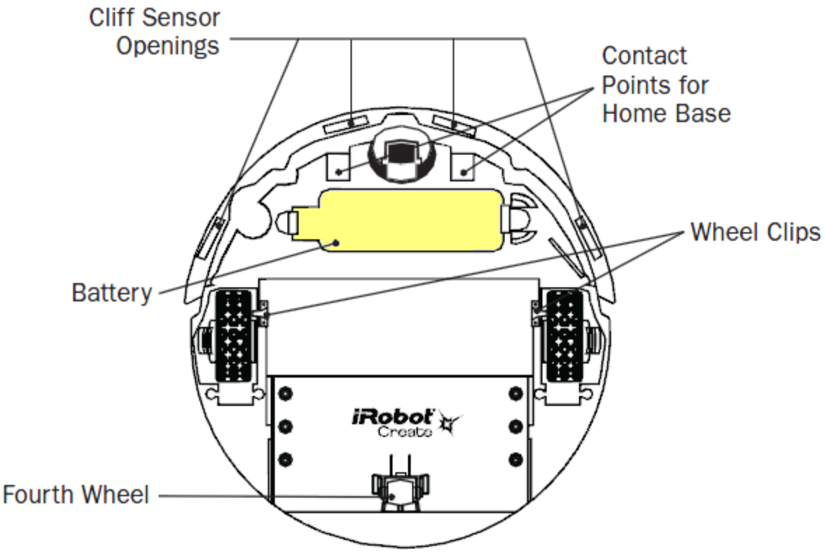


Figure 8: Bottom view of the Qbot, highlighted with battery location

4.2.5. Infrared Sensors

The SHARP 2Y0A02 [5] is a low cost infrared range sensor (20-150 cm). There are five of these sensors included with the Qbot. The sensors are connected to the analog input channels of the Qbot DAC, which can then be read using the HIL Read Write block (Section 5.3).

Figure 9: SHARP GP2D12 IR Sensor

4.2.6. Sonar Sensors

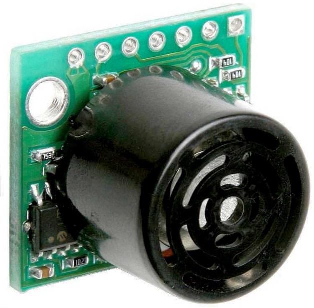


Figure 10: LV-MaxSonar-EZ0 Sonar Range Finder

The MaxSonar-EZ0 [6] is a very short to long-range detection and ranging sonar sensor (Figure 10). The MaxSonar-EZ0 detects objects from 0-inches to 254-inches (6.45-meters) and provides sonar range information from 6-inches out to 254-inches with 1-inch resolution. Objects between 0-inches and 6-inches range as 6-inches. There are three of these sensors included with the Qbot. The sensors are connected to the other input channels of the Qbot DAC, which can then be read using the HIL Read Write block (Section 5.3).

4.3. System Specifications and Model Parameters

Table 1 below lists the main parameters associated with the Quanser Qbot.

Symbol	*Description*	*Value*	*Unit*
d	Diameter of the Qbot	0.34	m
h	Height of the Qbot (with camera attachment)	0.19	m
v_{max}	Maximum speed of the Qbot	0.5	m/s
m	Total mass of the Qbot	2.92	kg

Table 1 Qbot Model Parameters and System Specifications

5. System Setup

In Section 5.1, instruction is given on how to assemble the vehicle. Section 5.2 walks through setting up a wireless connection with Windows. Finally, Section 5.3 explains how to properly setup the QUARC-Simulink interface so that the user can communicate with the Gumstix.

5.1. Setting up the Qbot

Follow these steps to setup the Quanser Qbot:

1. Insert the battery underneath the Qbot.
2. Press power button shown in Figure 4. This turns on both vehicle and Qbot DAC/Gumstix.

5.2. Establishing Wireless Connection

In order to establish wireless communications between the host PC and the Gumstix embedded computers, a wireless ad-hoc (peer-to-peer) network is used. Each Gumstix target is assigned a unique IP address between 182.168.1.20 and 182.168.1.254. A wireless adapter is provided with the system for the host PC. The IP address of this wireless adapter must also be a unique IP address that is not already used by any Gumstix computer or another host PC. The host PC wireless IP address should be set to an unused value between 182.168.1.10 and 182.168.1.19. If you are setting up only one host PC, set the wireless IP address to 182.168.1.10. For additional host PCs, set their IP addresses by incrementing the last digit, e.g., 182.168.1.11, then 182.168.1.12, etc.

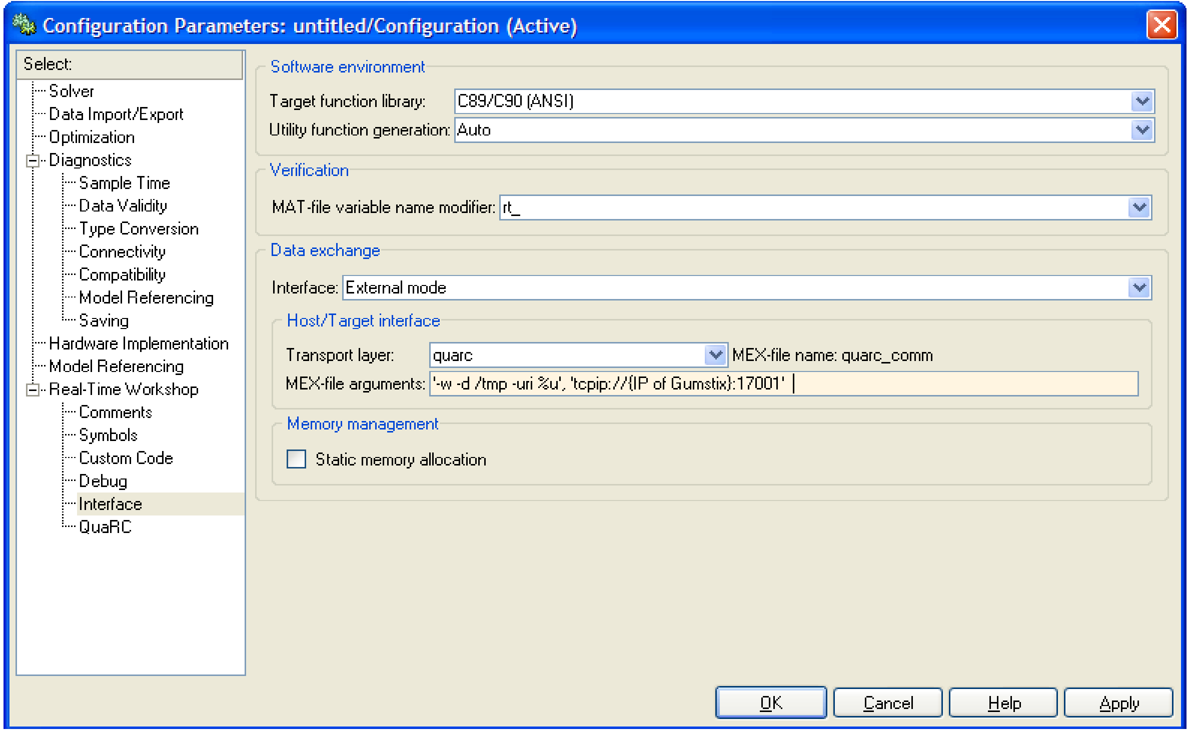
1. Windows should detect a network called GSAH, an “Unsecured Computer-to-Computer Network”. Connect to it.
2. Open the **Status** of the wireless network, then click on Properties.
3. Under “This connection uses the following items:”, scroll down to Internet Protocol (TCP/IP), then double click on it.
4. Instead of obtaining an IP address of the computer automatically, enter the following:
   IP address: 182.168.1.xxx (any unused number)
   Subnet mask: 255.255.255.0
   Default gateway: 182.168.1.xxx (same as IP)
5. The Gumstix can be pinged by typing “ping {IP of Gumstix}” in the Run box in Windows. If the connection is successful you will see the ping replies in the command window. **Note: you may need to disable Windows firewall to establish a connection.**

5.3. QUARC – Quanser Real-Time Control

Simulink should have a new menu item called QUARC once QUARC has been installed. **Note: QUARC should be installed with beta library components selected since the Qbot uses some QUARC components that are found in the beta library.**

The following steps are required to setup QUARC for the Qbot:

1. Create a new Simulink model, or open an existing model to be run on the Gumstix.
2. Click on the QUARC menu, then select Options.
3. The System target file under Real-Time Workshop should be quarc_linux_verdex.tlc. Browse through the system target list to locate the proper file if necessary.
4. Next, choose Interface on the left hand pane. Under the MEX-file arguments, type '-w -d /tmp -uri %u', 'tcpip://{IP of Gumstix}:17001'. Include the single quotation marks (Figure 11). Replace {IP of Gumstix} with the IP of your Gumstix, e.g. 'tcpip://182.168.1.200:17001'.


Configuration Parameters: untitled/Configuration (Active)

Software environment

Target function library: C89/C90 (ANSI)

Utility function generation: Auto

Verification

MAT-file variable name modifier: rt_

Data exchange

Interface: External mode

Host/Target interface

Transport layer: quarc MEX-file name: quarc_comm

MEX-file arguments: '-w -d /tmp -uri %u', 'tcpip://{IP of Gumstix}:17001'

Memory management

□ Static memory allocation

Figure 11: Configuring MEX-file Arguments

5. In the Simulink Library Browser, navigate down QUARC Targets → Devices → Third-Party → iRobot → Roomba → Interfacing. The Roomba Initialize block is required for the Gumstix to communicate with the Qbot. Change the URI to “serial://localhost:1?baud=57600,word=8,parity=none,stop=1” (Local host is 2 by default). Use default send receive bytes.

6. In the Simulink Library Browser, navigate down QUARC Targets → Data Acquisition → Generic → Configuration. The HIL Initialize block is required to communicate with the Qbot DAC. Choose a Board name. Select the board type "qbot" to target the Qbot when running the model.
7. Add the HIL Read Write block from QUARC Targets → Data Acquisition → Generic → Immediate I/O to read infrared sensor (analog) and sonar measurements from the vehicle. The Qbot DAC also supports PWM write channels [0:3].
8. For image processing using the on-board camera, use the blocks in QUARC Targets Beta → Image Processing → Open Source Computer Vision.
9. Select “External” for simulation mode, instead of “Normal”, which indicates that the model is to be run on the target machine (Gumstix) rather than simulating the model on the host machine. Table 2 and Table 4 list the block sets supported by the Quanser Qbot, and Table 3 highlights the more important Qbot blocks.

Block set	***Description***
Interface (Figure 12)	This block set implements the basic application program interfaces (APIs) provided by iRobot Create®. The APIs can be broadly categorized based on the following functionalities: ● serial connection configuration between the high level controller (e.g., PC or Gumstix) and Qbot, ● setting operation mode of Qbot, ● accessing Qbot's sensory information, ● other hardware configuration (e.g., setting digital and analog I/O ports, and changing LED colours.
Application (Figure 13)	This block set allows a user to implement navigation algorithms using available sensory information. The present blocks use only wheel encoder data and bump sensors for navigation and obstacle avoidance.
Image Processing (Figure 14)	This block set implements image acquisition and processing functionalities using the OpenCV library. The image processing block set allows to capture images from a USB camera, to process it on-line, and to save it to a disk for further analysis.

Table 2: Description of the Relevant Qbot Block Sets for the Qbot in QUARC

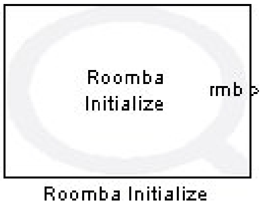
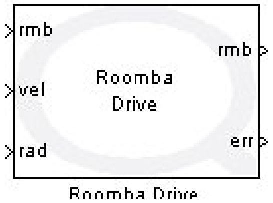
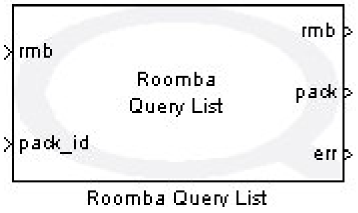
Block	***Description***
	This block is required to establish a serial connection to a Qbot. A Qbot is identified by a Universal Resource Identifier (URI), such as `serial://localhost:1baud=57600,word=8,parity=none,stop=1`, where communication port 1 of the target is connected to the Qbot serial port with the specified parameters. Note: The "rmb" output of this block must be connected to the "rmb" input of the blocks in the Roomba block set in order to access the Roomba.
 rmb vel rad Roomba Drive rmb err Roomba Drive	This block drives the Qbot with two inputs: velocity and radius. To drive straight, the radius input takes 32768 or 32767. This is a special case, and refer to the MATLAB Help page for a more complete list of possible commands.
 rmb pack_id Roomba Query List rmb pack err Roomba Query List	This block retrieves various sensor data from the Qbot. The *pack_id* input ranges from 7 to 42, and each input will display a value from the *pack* output. See the MATLAB Help page for a full description for each of the packet IDs. Notable pack_id values: 8 = wall (0 = no wall, 1 = wall seen) 19 = distance (The distance that the Qbot has traveled in millimeters) 20 = angle (The angle in degrees that the Qbot has turned) 22 = voltage (Voltage of the Qbot's battery in millivolts) 23 = current (Current in milliamps flowing into or out of the Qbot's battery)

Table 3: Important Qbot Blocks

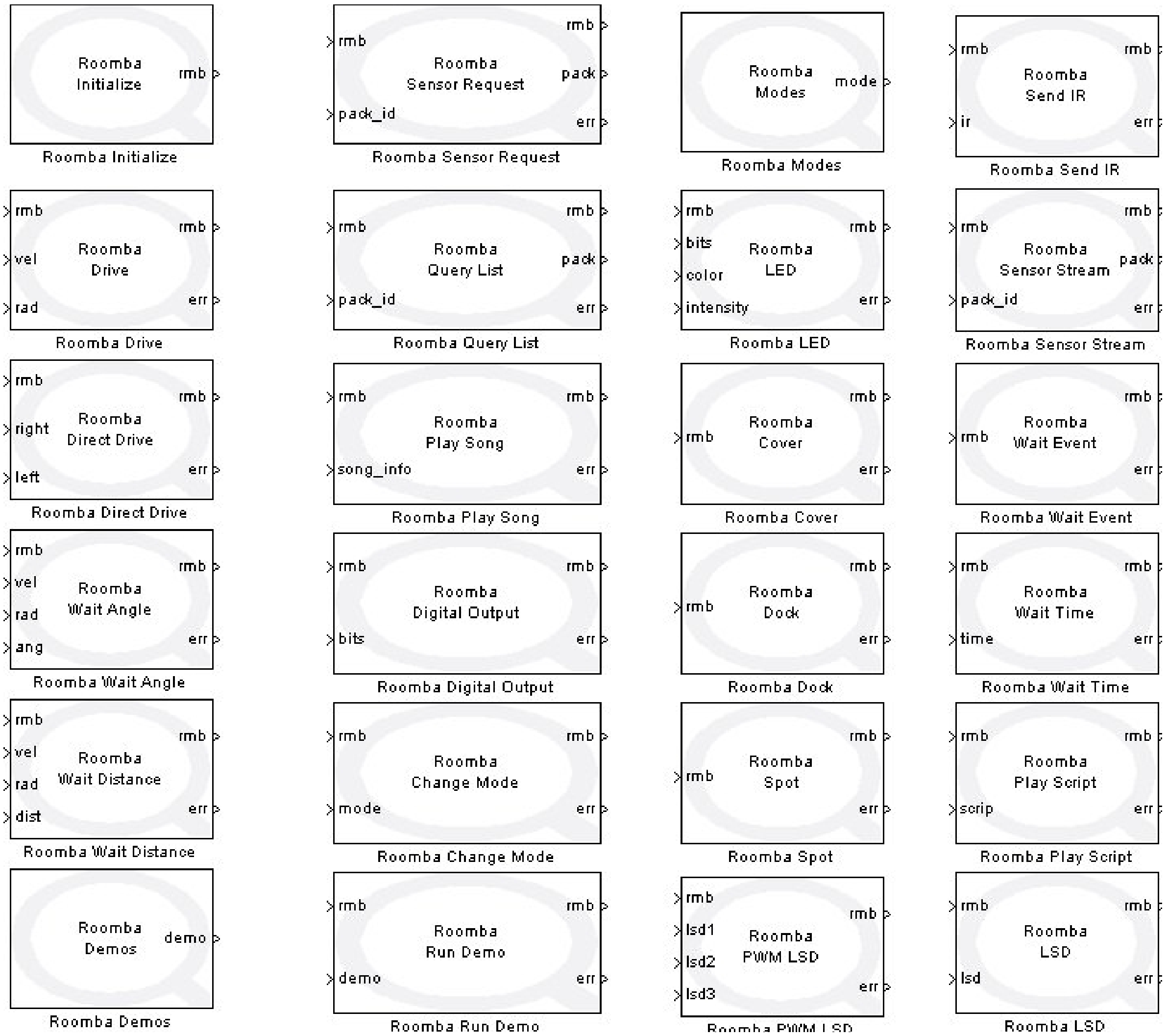


Figure 12: Interface Block Set

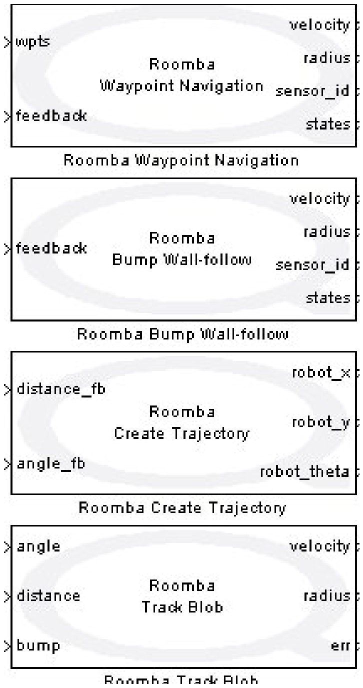


Figure 13: Application Block Set

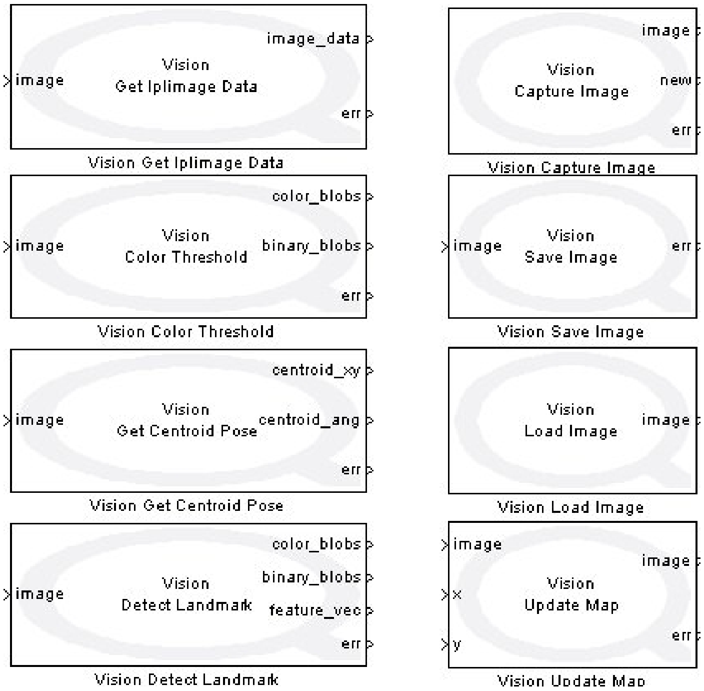


Figure 14: Image Processing Block Set

<table>
<tr><th><em>Block</em></th><th><em>Description</em></th></tr>
<tr><td>
QuaRC
controls development made easy
HIL Initialize
HIL-1 (qbot-0)

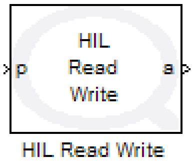
</td><td>Initialization block required in all QUARC models using a HIL card. Note: only one HIL Initialize block is needed in the model to setup one vehicle. The HIL Read Write block must reference this block to determine the board being used.</td></tr>
<tr><td>
</td><td>The current version of the Qbot DAC can read from analog channels (for the infrared sensors), and sonar channels. PWM channels are supported for write operations.
<table>
<tr><td>Channel Type</td><td>Valid Channel Ranges</td><td>Description</td></tr>
<tr><td>IR Input (Analog Read)</td><td>[0:4]</td><td>Reads infrared sensor measurements</td></tr>
<tr><td>Analog Input
(Read)</td><td>[0:6]</td><td></td></tr>
<tr><td>Sonar Input (Other Read)</td><td>[1:3]</td><td>Reads sonar sensor measurements</td></tr>
<tr><td>PWM (Write)</td><td>[0:7]</td><td>Gives duty cycle commands as a percentage (0 to 1) of a 20ms period, where values range from 0.025 to 0.125.</td></tr>
</table></td></tr>
</table>

Table 4: Description of the Relevant HIL Blocks for the Qbot in QUARC

At this point, the model is ready to be compiled and executed on the target.

5.4. Using the on-board camera

The Qbot is equipped with a Logitech QuickCam Pro 9000 USB camera. When connected to the on-board Gumstix Verdex computer, this camera is capable of capturing up to 9 frames per second (fps). In order to acquire and display images on the host computer during execution some additional blocks are required. Figure 15 shows an example model where images are acquired and displayed.

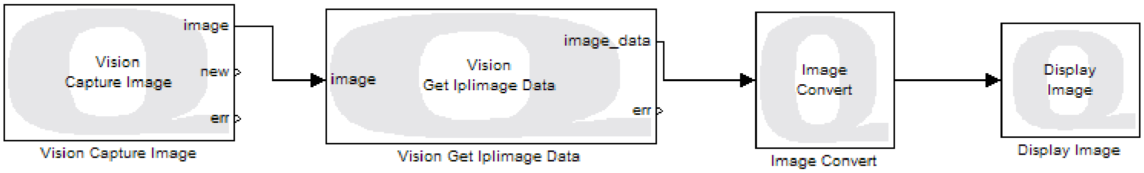


Figure 15: Capturing and displaying images in QUARC.

The Vision Capture Image block is used to capture image data from the camera. The Vision Get Iplimage Data block simply takes the image and outputs the image data as a matrix. For the Qbot camera, color images are output as 3-dimensional matrices (3 x ImageWidth x ImageHeight) with each pixel represented by a 3-vector of [Blue, Green, Red] values between 0 and 255.

The Display Image block is used to display images on the host. This block acts like a scope and will not delay the execution of the real-time model. The images displayed in the Display Image block may be delayed compared to the actual image data being acquired and used in the real-time model. The Image Convert block is used to convert the image format from the Qbot camera (BGR8) to a format that can be displayed in MATLAB.

Note: since displaying images on the host requires a large amount of bandwidth, it is recommended that the Display Image block be run with a sample time of 0.1 seconds or slower. If the error “Warning: Not enough memory on the target to process the packet: EXT_SELECT_SIGNALS” is displayed in the MATLAB Command window or data on Scopes and the Display Image block are not displayed during model execution, the size of the external mode buffer is too large. The Display Image block acts like a Scope and uploads its signal (the image signal) to the host for display through Simulink's external mode buffers. In order to resolve this issue so that images and data will be displayed the buffer size should be reduced. To reduce the size of the buffer, either increase the model's base rate so that it is faster or reduce the number of base rate samples used in the model's external mode buffer by decreasing the Duration (recommended 1 to 10) in the Signal & Triggering menu under Tools > External Mode Control Panel > Signal & Triggering.

Because the vision blocks require OpenCV2.2, models using these vision blocks must be configured for code generation in C++. To set this parameter select the menu option

QUARC > Options and navigate to the Code Generation page and select C++ under Language. The LLVM-GCC cross-compiler (default compiler for the quarc_linux_verdex target type) will not compile models using the vision blocks. The compiler can be changed under the QUARC > Options window on the Code Generation > QUARC page. Selecting Native GCC will use the compiler on the Verdex, which is slower than the cross-compiler since it downloads the generated code and compiles natively on the Gumstix. To utilize a cross-compiler, the CodeSourcery Lite cross-compiler can be used. Use of the CodeSourcery Lite cross-compiler results in much faster compilations, but requires that the CodeSourcery Lite cross-compiler be installed (see the Installing CodeSourcery help page in the MATLAB help for details on downloading and installing CodeSourcery).

6. Troubleshooting Guide

For any issue, the first and easiest troubleshooting solution on any electronic device is to reboot the device. Turn off the Qbot, then turn it back on again. For troubleshooting any problem with the Qbot, 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 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://182.168.1.xxx:17000”, where xxx corresponds to the IP address of the Qbot.

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

6.1. Qbot Drive or Direct Drive commands aren't responding or are causing the robot to move incorrectly, or the Qbot bump sensor inputs are not functioning.

1. Check that the Qbot wheels are retracted fully into the wheel bays. If any wheel is out of its bay, then the Qbot will stop driving for safety reasons. Makes sure to operate the Qbot on a hard, flat surface.
2. Turn off the Qbot and iRobot Create power and remove the jumper on the INT/EXT battery header on the Qbot. Turn on the iRobot Create and run the built-in demos and troubleshooting steps found in the iRobot Create Owner's Guide supplied with your Qbot. If the troubleshooting steps in the iRobot Create Owner's Guide show that the robot is working, then replace the INT/EXT battery jumper on the Qbot and continue. If the iRobot Create demos do not work as expected, please contact iRobot technical support.

3. Turn on the Qbot and open the VAL Read Sensors block in one of the supplied Simulink models. In the VAL Read Sensors parameters under the "Other" channels, select the bump sensors by typing [13000:13002]. Attach a scope or display block to the VAL Read Sensors "Other" channels output and compile and run the model. Verify that the bump sensors can be read by the Qbot computer by pressing the Qbot bumper and watch the bump sensors signals go from 0 to 1 when pressed. If these signals do not change, then open the Qbot lid by removing the four lid screws shown in Figure 16 and verify that the Qbot computer is connected to the robot. Make sure the DB-25 connector is firmly in place, replace and secure the Qbot lid, and retry the bump sensor test.

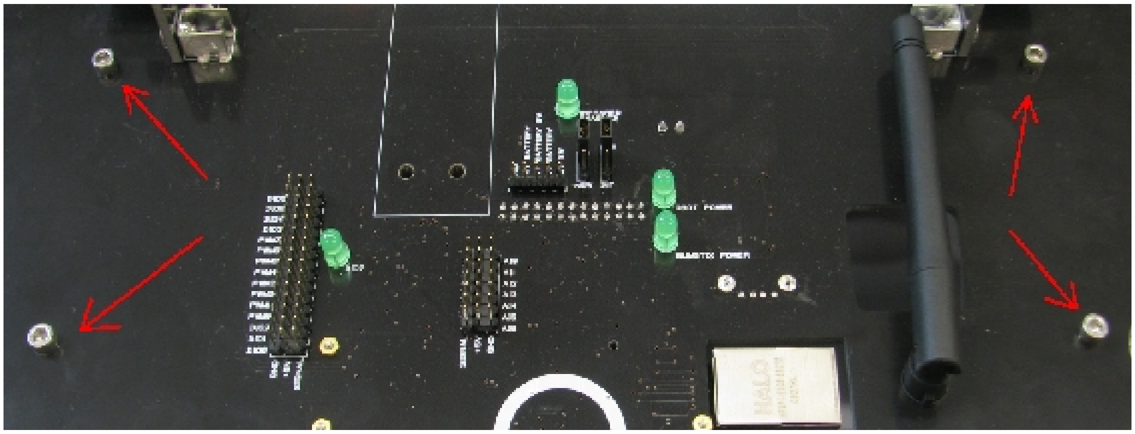


Figure 16: The Four Qbot Lid Screws

6.2. The model fails to build/connect or the QUARC console does not successfully open.

1. Turn on the Qbot and look through the square window located on the lower right of the Qbot lid. When the Qbot powers up, you will see an orange LED come on in the inner cargo bay of the robot. After approximately 30 seconds, a blue LED will flash to indicate the Qbot wifi is powering on, and is attempting to connect to another computer on the ad-hoc wifi network. If the blue LED flashes and remains on, then the wifi module is functioning and is able to find another node on the ad-hoc wireless network. If the blue LED flashes and then turn off, the Qbot is not able to detect other nodes (e.g., the host PC) on the ad-hoc wireless network. Check that the wifi adapter is inserted properly in the host PC and is configured according to the network configuration procedure outlined in this manual (System Setup\Establishing Wireless Connection). Verify that the host PC is connected to the wireless ad-hoc (GSAH) network and try to successfully ping the Qbot by going to the Windows Start → Run and typing "ping 182.168.1.xxx" (without quotation marks), where xxx corresponds to the IP address of your Qbot. If the blue LED

never flashes, the wireless antenna or wireless module may be disconnected. Turn off the Qbot power and open the Qbot lid by removing the four lid screws shown in Figure 16 and verify that the Gumstix wireless module and antenna are properly connected. The wireless module shown in Figure 17 is located on the bottom side of the green Gumstix computer inside the Qbot cargo bay. Make sure the wireless module and antenna are securely connected and retry the above steps to establish a wireless connection.

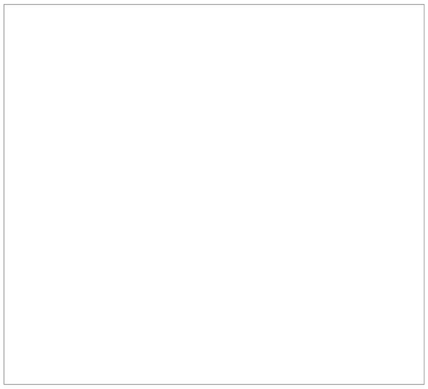


Figure 17: Wifi Module

6.3. The Qbot sensors are not being read correctly or they are stuck at some constant value.

1. Using the VAL Read Sensors block, output all possible channels. Check these outputs using scopes and displays, and determine if the problem lies with a particular sensor, or set of sensors, or if the issue is global across all sensors.

2. Create a new Simulink model and add a HIL Initialize and HIL Write block. Select the “qbot” board type in the HIL Initialize block. Open the HIL Write block and select only digital output channel 7. Connect either a square wave going from 0 to 1, or a constant block to the input of the HIL Write block. Compile and run the model. Verify that the LED leveled DIO7 on the Qbot lid turns on and off when the input to the HIL Write block goes from 1 to 0, respectively. If the DIO7 LED does not turn on, there is a communication failure between the Gumstix and the Qbot computer.

6.4. The Qbot magnetometer values drift while the Qbot is moving in a straight line.

1. The magnetometer detects the Earth's magnetic fields as wells as external magnetic fields. Metal underneath the floor or in surrounding structures will affect the direction of the magnetic fields, thus affecting the Qbot magnetometer measurements. This is a fundamental limitation of any magnetometer device.

6.5. 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”.

1. The maximum sample rate recommended for the Qbot is 500 Hz (0.002 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. Try reducing the model sample rate in the menu QUARC\Options\Solver by increasing the “Fixed-step size (fundamental sample time)” parameter.

2. The HIL Read Write block and the VAL Read Sensors block should only be used once in a diagram. These blocks perform large data transfers between the Qbot computer and the controllers, so placing more than one of these blocks will cause multiple reads to be performed in the same sample instant, which is unnecessary. Use only one HIL Read Write or VAL Read Sensors block for the entire model.

3. 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.

6.6. Trying to start the Qbot model results in the error “Unable to locate the dynamic link library or shared object.”

1. This error indicates that the Qbot driver is not found on the target. Make sure that the model target type is set to Linux Verdex by navigating to the QUARC menu QUARC\Options\Real-Time Workshop pane and changing the System target file to quarc_linux_verdex.tlc. Open a console through the QUARC menu QUARC\Console for all, and verify that the console window displays the target IP of your Qbot in the window title.

6.7. The LEDs QBOT POWER and GUMSTIX POWER do not illuminate when the Qbot robot is powered on.

1. Verify that the jumper on the INT/EXT battery header on the Qbot lid is securely placed in the INT position. This jumpber switches the Qbot power source from the robot's internal poewr (INT) or an external battery source (EXT). Note: an external battery to power the Qbot is not provided with the system.
2. If the jumpers are securely in place and the power LEDs remain off when the iRobot Create is on, turn off the robot, open the Qbot lid by removing the four lid screws shown in Figure 16, and verify that the DB-25 cable to the robot is securely connected. Replace the lid and try to power on the Qbot.

6.8. Building a model fails with the error “Not enough system resources are available to perform the operation.” The hard disk is full on the Gumstix computer.

1. When several models are compiled, the disk space on the Gumstix may become full, and you will no longer have space to build models. Using the clean option in the QUARC menu under QUARC\Clean all will remove all generated code and compiled code for the current model, but this will only free up the space used by the current model. Clearing all models from the Gumstix hard disk must be done manually using PuTTY, or any ssh client. Using PuTTY, log into the Gumstix by connecting to it using the IP listed on the Qbot, and use the user name “root”, and password “quanser” (without quotation marks). Navigate to the /var/spool/quarc directory, which is where all generated code and compiled models reside. Use the “ls” command to list all contents of this directory to see all models that have been downloaded to the Gumstix. **Note: be careful executing any remove/delete command on your Gumstix, as incorrect usage could permanently damage the file system, rendering the Qbot unusable.** Individual models/code can be manually removed from the /var/spool/quarc directory. To remove all generated code and executable controllers, run the command “rm -rf /var/spool/quarc/*” (without quotation marks). To confirm there are no longer any models on the Gumstix, execute the command “ls” and there should be nothing listed.

6.9. The Display Image block is not showing any image.

See Section 5.4 for an example using the camera acquisition and display blocks.

1. Ensure that the Vision Capture Image block is acquiring images by checking the “new” output signal. If necessary, reduce the image resolution and output the image as matrix data.

If the image is output as matrix data remove the Vision Get Iplimage Data block.

2. Check that the Image Convert block is set to convert the source image format BGR8 to MATLAB RGB.

3. Set the Display Image sample time to 0.5 seconds.

4. If the error “Warning: Not enough memory on the target to process the packet: EXT_SELECT_SIGNALS” is displayed in the MATLAB Command window or data on Scopes and the Display Image block are not displayed during model execution, the size of the external mode buffer is too large. The Display Image block acts like a Scope and uploads its signal (the image signal) to the host for display through Simulink's external mode buffers. In order to resolve this issue so that images and data will be displayed the buffer size should be reduced. To reduce the size of the buffer, either increase the model's base rate so that it is faster or reduce the number of base rate samples used in the model's external mode buffer by decreasing the Duration in the Signal & Triggering menu under Tools > External Mode Control Panel > Signal & Triggering.