Autonomous Vehicles Research Studio

Setup Guide – QBot 2/2e I/O Check

Table of Contents

A. QBot 2/2e I/O Check

1. From the same folder containing this file, open the folder for your vehicle, QBot 2 or QBot 2e IO Check, and open the Stabilizer_*vehicle*.slx file.

2. In the model that loads, from the **HARDWARE** tab on top of your Simulink model, click on **Hardware Settings** (Gear Icon). (If using an older version, click on **Model Configuration Properties** under the **Simulation** drop menu.)

3. Expand **Code Generation** on the left side of the window, click on **interface** and set the **MEX-file arguments** (Figure 1) as follows

   ```
   '-w -d /tmp -uri %u -URI_Host tcpip://192.168.2.H:18000','tcpip://192.168.2.X:17001'
   ```

   where 192.168.2.X refers to the IP address of the QBot 2/2e you are using (found on the side of the robot), and 192.168.2.H refers to the IP address of the ground control station PC (default IP is 192.168.2.5, but you can find this by typing ipconfig in the command prompt). Press OK.
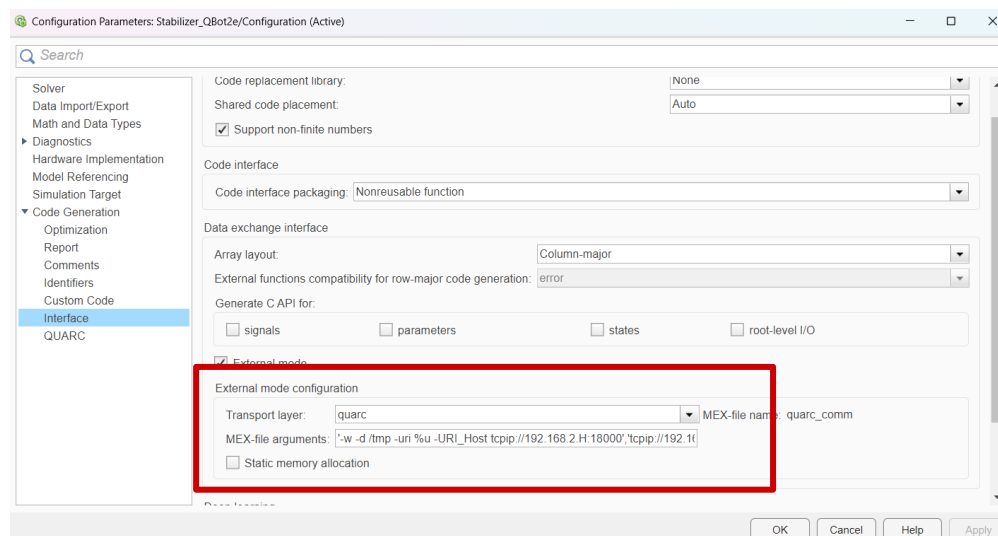

Figure 1. Setting up the MEX-file arguments

4. Turn on the vehicle and ensure that a connection to the vehicle is established by pinging to it. See the vehicle communication document for more information.

5. Open another MATLAB instance.

6. In this new MATLAB window, open the file in the same folder called Mission_Server_*Vehicle*_IO_Check.mdl. If you have an FrSky joystick (silver with a screen at the bottom) open the one that says _FrSky.mdl

7. Ensure that the joystick is turned ON and the USB dongle is plugged into the ground control station PC.

8. In the model that loads, open the **MISSION SERVER QBOT2 IO CHECK** subsystem. Under it, open the **JOYSTICK** subsystem (Figure 2a), and double-click on the **Host Game Controller** block (Figure 2b). Ensure that the Controller number (Figure 2c) drop down menu has selected the item labelled **FrSky RECEIVER**

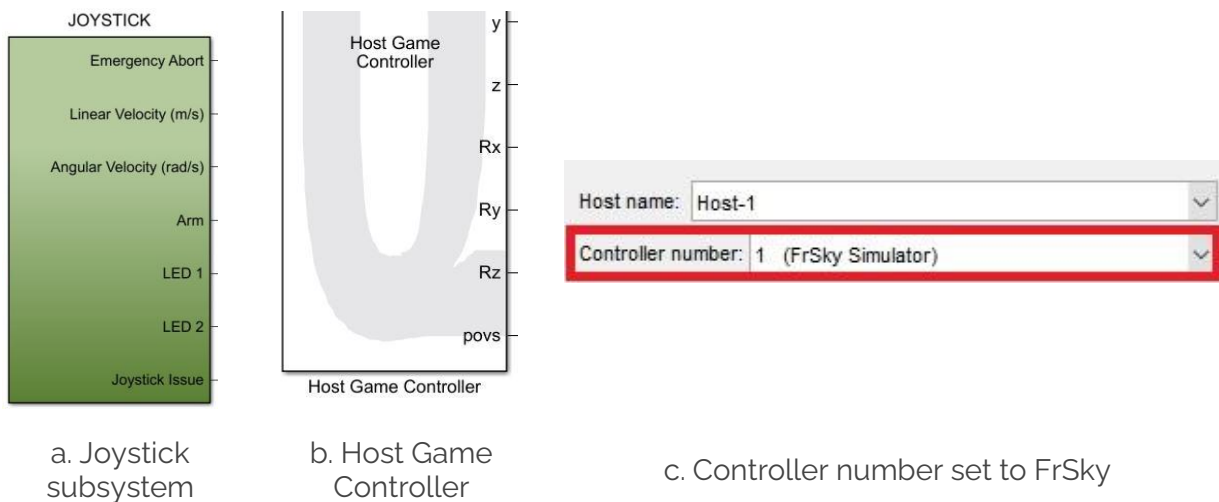a. Joystick subsystem   b. Host Game Controller   c. Controller number set to FrSky

Figure 2: Selecting the FrSky receiver in Simulink/Quarc

9. On the Mission Server model, click on the **HARDWARE** tab on the top menu, and then click the green play button (**Monitor & Tune**), It should build and start the model. If you have an older version of Simulink that does not have a Hardware tab, under the **QUARC** drop down menu, click **Build** and once it finishes click **Start**.

10. Ensure the mission server starts running by checking the simulation time bar at the bottom right of the model (figure 3).

Figure 3: Simulation Time bar showing current time stamp

11. Ensure that the **Joystick Issue** display reads 0 (no joystick issues). Ensure that all the toggles are set away from the user (see FrSky Joystick documentation). Make sure that the joystick's Arm/Disarm toggle is set as in Figure 4a and LED toggle is set as in Figure 4c.

a. Disarm   b. Arm   c. Ready LED Green   d. Ready LED Red
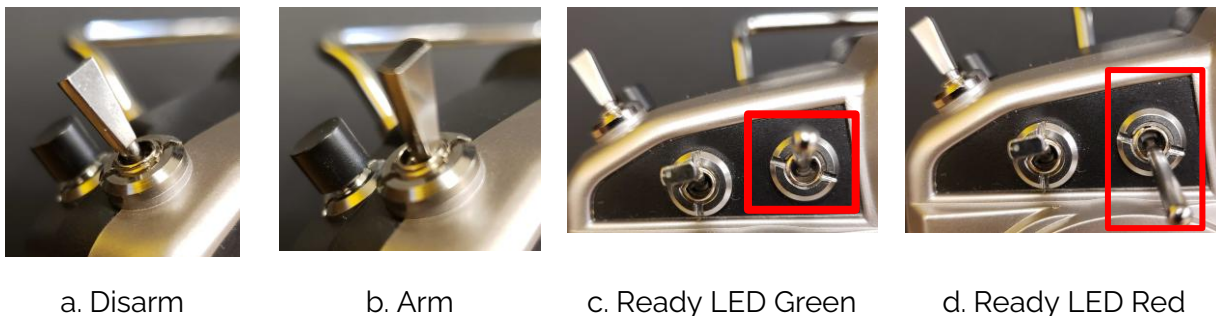
Figure 4: Joystick Arm/Disarm toggle (a and b) and LED toggle (c and d)

12. Follow step 9 again but run the **Stabilizer** model instead.

13. You should hear a sequence of beeps to signify that the **Stabilizer** model is running on the QBot 2. The user programmable LED (LED #2) on the QBot 2 should turn ON a solid green.

14. Move the Takeoff/Land toggle to the 2 position (Figure 4d) (Move the toggle all the way towards you). The LEDs should now switch to a solid red.

15. Switch the Arm/Disarm toggle to 2 (Figure 4b) (Move the toggle all the way towards you). This should arm the QBot 2. Use the Roll/Pitch stick on the joystick to move the QBot 2 forward/backward and make it turn. The commands from the stick are mapped to the linear and angular velocity of the QBot 2.

16. Move the Arm/Disarm toggle to 0 (Figure 4a). Move the LED toggle to 0 (Figure 4c). The QBot2_Stabilizer and Mission_Server_QBot2_IO_Check models can now be stopped.

This completes the **QBot 2 IO check** task and confirms that your vehicle is functioning correctly. If you have any errors, make sure that all the steps prior to this checkpoint have been followed. If further issues persist, please contact Quanser technical support (tech@quanser.com).

QUANSER
INNOVATE·EDUCATE

Solutions for teaching and research. Made in Canada.