MULTIPLE BRICKS TEST

GROUP 4

2nd December 2016
Version 1

Abstract

This test will be used to check whether multiple bricks can communicate with each other. We realised that this is possible, but errors resulting from multiple bricks may be hard to debug.

Contents

1 BACKGROUND

1.1 Edit History

Jake Zhu: 2016-10-28, Initial set up

Kareem Halabi and Jake Zhu: 2016-11-01, Tested out and wrote the document

1.2 Test Information

Tester: Kareem Halabi

Author: Jake Zhu

Hardware Version: Two Bricks connected via USB

Software Version: BrickComs.java

2 GOAL

To see whether it is feasible to use two bricks and have them communicate with each other.

3 PROCEDURE

Setup:

1. Using the included USB cable, connect the mini-USB side to the master brick and the full-USB side to the slave brick.
2. Connect a Large EV3 Motor to master brick port A and another Large EV3 Motor to slave brick port A.
3. Connect a EV3 Light Sensor to master brick port S1 and another EV3 Color Sensor to slave brick port S1.
4. Connect a EV3 Ultrasonic Sensor to master brick port S4 and another EV3 Ultrasonic Sensor to slave brick port S4.
5. Run BrickComs.java

Tests:

1. Make sure the motors run correctly
2. Make sure the light sensors run correctly

3. Make sure the ultrasonic sensors run correctly
4. Make sure the audio run correctly
5. Make sure LCD screen is correctly displayed
6. Test extreme case

4 EXPECTED RESULT

We expect that there will be some delay because of the Master-Slave relationship. This is because the slave brick has to wait for a signal from the master brick to execute an instruction. The Remote brick will then run based on the received signal from the Master brick.

5 TEST REPORT

5.1 Motors

- The biggest issue is the fact that the remote motor objects will have to be closed before the master brick terminates the program
- If this does not occur, the ports will remain locked on the remote brick preventing them from being opened the next time the program runs. In the event of an unexpected termination of the master program, the ports can be forcefully unlocked through the "Reset" option in the LeJOS menu.

5.2 Audio

- The audio worked fine. Both robots beeped when instructed. Audio objects do not need to be closed.

5.3 LCD Screen

- To access the slave LCD, we used `slave.getTextLCD();` However, this method did not work with the master LCD. We had to use `LocalEV3.get().getTextLCD();` instead.
- Another issue with the slave LCD screen is that the top menu displaying the battery level and IP address cannot be cleared. Therefore we were restricted to the number of usable lines on the slave LCD.

5.4 Ultrasonic Sensor

- Wanted to poll two ultrasonic sensors sequentially rather than simultaneously to avoid interference between the two if they are at a close proximity. We wrote a test that sequentially polled each sensor by only enabling one at a time. To do that, we had to use an `EV3UltrasonicSensor` object to access the `enable()` and `disable()` methods (otherwise the sensors continue emitting pulses even if no sample is being fetched).
- Remote sensors can only be accessed via a `RMISampleProvider`, an interface that can represent various local and remote sensors, thus `enable()` and `disable()` are not visible.
- Realised that if the sensros are perpendicular to each other, they will not interfere.

5.5 Color Sensor

- The light sensors worked just as we expected. There was no interference between either light sensors when both are polled in parallel.

5.6 Extreme Case

As a final test, we connected one of each three types of motors (`EV3LargeRegulatedMotor` `EV3MediumRegulatedMotor` and `NXTRegulatedMotor`) to both bricks and one of each three relevant sensors (`EV3UltrasonicSensor`, `EV3ColorSensor` and `EV3TouchSensor`) to both bricks. We then had three threads running in parallel, one for each type of sensor, polling continuously from the master and the slave. Then, each motor's speed was coupled with the value from a sensor, changing the speed as the sensor value changes.

Even in this extreme case, everything continued to run smoothly and as expected.

6 CONCLUSION

Refer to Section 6.5 and 7.5 on the HARDWARE Document.

7 ACTION

This test report should be sent to the Software Team to confirm that multiple brick communication works. They will decide whether they want to use two bricks or not.

8 DISTRIBUTION

Software Development, Hardware Team