REV3 (Remote EV3 Control) Application Requirements Verification Document

April 1, 2016

ECE 573, Spring 2016
University of Arizona

Bryan Walsh
Team Name: bpwalsh

1 Document Purpose

The REV3 (Remote EV3 Control) application will allow the user to remotely control a Lego EV3 Robot, running the open source, debian based EV3Dev (www.ev3dev.org) operating system, from an Android smart phone. This application will include the ability to control the EV3 via GUI or by use of the phone's accelerometer and gyro sensors.

This document describes the testing that will be conducted to verify that the requirements of the REV3 project have been fully met and implemented.

2 References

The REV3 Design Document is referred to through out this document. Please see bpwalsh-design-document.pdf located in svn.

3 Requirement Verification: Startup and Configuration

3.1 Requirement B.1.1 Verification

Requirement: Upon startup the REV3 application shall immediately enter a configuration screen. The configuration screen shall allow the user to enter the IP address or hostname and port of the EV3 robot that they wish to connect to. The screen shall also have a connect button that upon clicking will establish connectivity to the EV3 robot. (No Dependencies)

Validation: Run Tests B.1.1-1,2,3, and 4.

Test B.1.1-1

Purpose: Verify that "Upon startup the REV3 application shall immediately enter a configuration screen."

Description: Execute unit test where the REV3 application is started and visibility of the Configuration Dialog will be confirmed by querying the dialogs visibility using Espresso's isDisplayed function.

Test B.1.1-2:

Purpose: Verify that the configuration screen will not accept malformed inputs.

Description: Execute unit test where the REV3 application is started and a malformed IP address and port is entered using Espresso. The Edit Text of the Configuration Dialog is inspected to ensure that the malformed inputs were not accepted. Malformed inputs will include: Non numeric characters in the IP address and port, port range outside of 1024-65535, and IP address octet range greater than 255. Test inputs are summarized in the below table.

IP Address Entries To Be Tested	Port Entries To Be Tested
invalid.hostname.com	1023
0.0.0.256	65536
192.168.1.256	invalid_port_data
192.168.1	0
192.168.1.1.1	-1024
192..168.1.1	
0.0.0.-1	

Test B.1.1-3:

Purpose: Verify that the configuration screen will accept properly formed inputs.

Description: Execute unit test where the REV3 application is started and a properly formed IP address and port is entered using Espresso. The Edit Text of the Configuration Dialog is inspected to ensure that the inputs were accepted. Properly formed inputs will consists of an IP addresses with octet ranges between 0 and 255, and port numbers between 1024-65535. Test inputs are summarized in the below table.

IP Address Entries To Be Tested	Port Entries To Be Tested
0.0.0.0	1024
255.255.255.255	65535
10.0.0.0	10000
192.168.1.1	20000
172.16.1.1	30000

Test B.1.1-4:

Purpose: Verify that upon clicking the “connect” button that the “Start Up” sequence (sequence model 5.1.2 from design document) is executed and that connectivity is established to the EV3 robot proxy.

Description: Execute unit test where the REV3 application is started and a properly formed IP address and port is entered using Espresso, the connect button is clicked, and then connectivity is examined by querying the Protocol Processor's isConnected function.

3.2 Requirement A.1.2 Verification

Requirement: If the REV3 application looses connectivity to the EV3 the application shall reinitialize and return to the configuration screen. (Dependencies: B.1.1)

Validation: Run Test A.1.2-1

Test A.1.2-1:

Purpose: Verify that upon disconnect that the REV3 application executes the “Disconnect” sequence (see design document, sequence 5.8.2) and returns to the configuration screen.

Description: Execute unit test where application is started and connected to the EV3 proxy. A disconnect event is simulated by calling the ProtocolProcessor's simulateDisconnect function. This function triggers an IOException, which triggers a reinitialization. Reinitialization state is examined as follows: 1) References to the Protocol and Command Processor is examined from the main activity to ensure that they are Null (ie references have been released). 2)Visibility of the Configuration Dialog is confirmed by querying the dialogs visibility using Espresso's isDisplayed function. A properly formed IP address and port is then entered using Espresso, the connect button is clicked, and then (re) connectivity is examined by querying the Protocol Processor's isConnected function.

4 Requirement Verification: Drive Controls

4.1 Requirement B.2.1 Verification

Requirement: “From the “drive” screen the user shall be able to adjust the commanded forward/reverse (+100%/-100%) speed via touch screen controls. The touchscreen commands will be translated to the proper EV3 protocol message necessary to accelerate/decelerate. (Dependencies: B.1.1, B.3.1)”

Validation: Run Test B.2.1-1

Test B.2.1-1

Purpose: Verify that upon touching forward/reverse touch screen controls that the “Command Power Level (GUI)” sequence (see design document sequence model 5.3.2) is executed. Verify that the bounds (+100%/-100%) are held.

Precondition: Application is started and connected to the EV3 proxy.

Description: Execute unit test where the “Forward” and “Backward” buttons are clicked via Espresso. The sequence of button clicks are summarized in the below table. Upon completion the Protocol Processor's transmit log is examined and compared against the expected responses described below.

Simulated Button Press	Expected Power Command
“Forward”	20
“Forward”	40
“Forward”	60
“Forward”	80
“Forward”	100
“Forward”	100

“Backward”	80
“Backward”	60
“Backward”	40
“Backward”	20
“Backward”	0
“Backward”	-20
“Backward”	-40
“Backward”	-60
“Backward”	-80
“Backward”	-100
“Backward”	-100

4.2 Requirement A.2.2 Verification

Requirement: “From the “drive” screen the user shall be able to adjust the commanded turn rate (+100% turn left/-100% turn right) speed via touch screen controls. The touchscreen commands will be translated to the proper EV3 protocol message necessary to turn. (Dependencies: B.1.1, B.3.2)

Validation: Run Test A.2.2-1

Test A.2.2-1

Purpose: Verify that upon touching left/right touch screen controls that the “Command Power Level (GUI)” sequence (see design document sequence 5.2.2) is executed. Verify that the bounds (+100%/-100%) are held.

Precondition: Application is started and connected to the EV3 proxy.

Description: Execute unit tests where the “Left” and “Right” buttons are clicked via Espresso. The sequence of button clicks is summarized in the below table. Upon completion the Protocol Processor's transmit log is examined and compared against the expected responses described below.

Simulated Button Press	Expected Power Command
“Left”	20
“Left”	40
“Left”	60
“Left”	80
“Left”	100
“Left”	100
“Right”	80

“Right”	60
“Right”	40
“Right”	20
“Right”	0
“Right”	-20
“Right”	-40
“Right”	-60
“Right”	-80
“Right”	-100
“Right”	-100

4.3 Requirement A.2.3 Verification

Requirement: The REV3 application shall be capable of receiving the change in heading (since configuration time) over the EV3 protocol. The application shall display the change in heading via an indicator arrow on the “drive” screen. (Dependencies: B.1.1, A.3.3)

Validation: Run Test A.2.3-1

Test A.2.3-1:

Purpose: Verify that the REV3 application can receive commands from the EV3 Proxy and that the “Display Delta Heading” sequence (see design document sequence 5.7.2) is executed.

Precondition: Application is started and connected to the EV3 proxy.

Description: Execute unit tests where simulated “Delta Heading Commands” are fed to the Protocol Processor via an input file. The rotation of the GUI's indicator arrow is verified by inspecting the GUI elements VectorDrawable.rotation via Espresso.

4.4 Requirement A.2.4 Verification

Requirement: From the “drive” screen the user shall be able to adjust the commanded speed (+100%/-100%) speed by accelerating the phone forward or backward. The user interaction shall be captured by the phones accelerometer sensor. The accelerometer outputs will be translated to the proper EV3 protocol “set speed” message. (Dependencies: B.1.1, B.3.1)

Note: This “A” requirement will be dropped (unless the instructor adamantly objects) from the final release for the spring 2016 semester. Hence the TBDs below.

Test A.2.4-1:

Purpose: Verify that the power filter is behaving in accordance to the reference implementation.

Precondition: Application is started and connected to the EV3 proxy.

Description: Execute unit test where recorded or simulated sensor inputs are played back through the REV3 application, outputs are compared against the power filter's reference implementation. Descriptions of the recorded/simulated playback files are TBD. The outputs of the Power Processor's filter are recorded. Upon completion the recorded file is compared against the reference implementation. The comparison will ensure that each value is within +/- 0.00001 of the values found within the reference implementation.

Test A.2.4-2:

Purpose: Verify that upon accelerating the phone forwards/backwards that the "Set Power Level (Accelerometer)" sequence (see design document sequence 5.4.2) is executed and providing expected outputs. Verify that the bounds (+100%/-100%) are held.

Precondition: Application is started and connected to the EV3 proxy.

Description: Execute unit test where recorded or simulated sensor inputs are played back through the REV3 application. The Power Processor's filter outputs and the commands sent to the EV3 proxy (from Protocol Processor) are recorded. Description of the recorded/simulated playback files are TBD. Upon completion the recorded commands are compared to a command formed (in unit test) from the power filter's output at the corresponding time. The comparison will ensure that each value is within +/- 0.00001.

4.5 Requirement B.2.5 Verification

Requirement B.2.5: From the "drive" screen the user shall be able to adjust the commanded turn rate (+100% turn left/-100% turn right) speed by turning the phone left or right. The user interaction shall be captured by the phones gyro sensor. The gyro outputs will be translated to the proper EV3 protocol "turn" message. (Dependencies: B.1.1, B.3.2)

Validation: Run Tests B.2.5-1 to B.2.5-8

Test B.2.5-1 to B.2.5-6:

Purpose: Verify that the turn filter is behaving in accordance to the reference implementation.

Precondition: Application is started and connected to the EV3 proxy.

Description: Execute unit test where recorded or simulated sensor inputs are played back through the REV3 application, outputs are compared against turn filter reference implementation. A description of the recorded/simulated playback files is provided in the table below. The outputs of the Turn Processor's turn filter are recorded. Upon completion the recorded file will be compared against the reference implementation. The comparison will ensure that each value is within +/- 0.00001 of the values found within the reference implementation.

Test #	Description	Recorded / Simulated	Comments
B.2.5-1	Phone laying flat on its back	Recorded	Filter uses accelerometer and atan(y/x) to estimate roll. Does x= 0 cause issues?
B.2.5-2	Phone laying flat on its back	Simulated	
B.2.5-3	Phone at 0° for 120 seconds	Recorded	Filter drift due to gyro biases?
B.2.5-4	Phone rotated to 0°, rotated to 180° and then back to 0°.	Recorded	Left Command and Bounds Checking
B.2.5-5	Phone rotated to 0°, rotated to -180° and then back to 0°.	Recorded	Right Command and Bound Checking
B.2.5-6	Periodically Rotated Between -90° and 90°. Period 30 secs.	Simulated	Verify Filter Response

Test B.2.5-7 and B.2.5-8:

Purpose: Verify that upon turning the phone left or right that the “Command Turn (Gyro)” sequence (see design document sequence 5.4.2) is executed and providing expected outputs. Verify that the bounds (+100%/-100%) are held.

Precondition: Application is started and connected to the EV3 proxy.

Description: Execute unit test where recorded or simulated sensor inputs are played back through the REV3 application. The Turn Processor's turn filter outputs and the commands sent to the EV3 proxy (from Protocol Processor) are recorded. A description of the recorded/simulated playback files is provided in the table below. Upon completion the recorded commands are compared to a command formed (in unit test) from the turn filter's output at the corresponding time. The comparison will ensure that each value is within +/- 0.00001.

Test #	Description	Recorded / Simulated	Comments
B.2.5-7	Phone rotated to 0°, rotated to 180° and then back to 0°.	Recorded	Left Command and Bounds Checking
B.2.5-8	Phone rotated to 0°, rotated to -180° and then back to 0°.	Recorded	Right Command and Bound Checking

5 Requirement Verification: EV3 Protocol

Requirement B.3.1: The EV3 protocol shall contain a message to command the speed of the EV3 by specifying the percent power level. +100% shall be interpreted as full power forward. -100% shall interpreted as full power backwards. (No Dependencies)

Validation: Run Tests B.2.1-1 (Described Above)

Requirement B.3.2: The EV3 protocol shall contain a message to command the turn rate of the EV3.

+100% shall correspond to a hard left turn. -100% shall correspond to a hard right turn. (No Dependencies)

Validation: Run Tests A.2.2-1, B.2.5-7 or B.2.5-8 (Described Above)

Requirement A.3.3: The EV3 protocol shall contain a message to receive the delta change in heading (since configuration) from the EV3. The delta change in heading shall be expressed in degrees. (No Dependencies)

Validation: Run Test A.2.3-1 (Described Above)