

# Test Document

## Project: LIBERTY

**Task:** Test the integral system

**Document Version Number:** 1.0

**Date:** 20/11/2016

**Author:** Edward Son

**Editor:** Andi-Camille Bakti

**Edit History:** [https://github.com/Gabetn/DPM\\_01\\_Project\\_Documentation](https://github.com/Gabetn/DPM_01_Project_Documentation)



# McGill

# TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b>	<b>2</b>
<b>1.0 TESTS:</b>	<b>3</b>
1.1 Green integration corner one test	3
Test 1: Integration (Localization + Navigation + Dismount)	3
<b>2. HARDWARE</b>	<b>7</b>

# 1.0 TESTS:

## 1.1 Green integration corner one test

### Test 1: Integration (Localization + Navigation + Dismount)

Date: 19/11/2017

Tester: Edward Son

Author: Edward Son

- 1) This test will validate the functionality of the integrated code so far, in terms of getting to the other end of the zipline.
- 2) This test should make the robot drive to the coordinates over wifi, more specifically to the red zipline other point (ZO\_R\_x, ZO\_R\_y). At that point, it should localize and face true zero.
- 3) The brick should be positioned near corner 1, in any orientation. The robot receives data over wifi, the zipline coordinates (ZO\_R\_x, ZO\_R\_y and (ZO\_G\_x, ZO\_G\_y). Once data is acquired, the command “start” is used on the server GUI and the robot starts localization. After localization, the angle is measured, and a button press continues to navigation where it turns to true zero, and the angle is measured again. The robot proceeds to navigate to the zipline point, where it localizes again, and the X, Y and angle error are measured. A button press gives it the go start the zipline traversal, and stops when dismounted for another angle check. Finally, localization is run and angle error is measured at the end.
- 4) The brick should end up at the point (ZO\_R\_x, ZO\_R\_y), after successfully completing each individual component
- 5)

Test Run #	Theta final error (true zero) (in °)	Theta error after localization (in °)
1	3.0	0.0
2	10.0	5.0

3	4.0	1.0
4	6.0	2.0
5	2.0	0.0
6	4.0	0.0
7	4.0	1.0
8	8.0	3.0
9	4.0	0.0
10	5.0	1.0

**Figure 1:** Position after localization (angle precision)

<b>Test Run #</b>	<b>Starting Position (x,y)</b>	<b>Xf (cm)</b>	<b>Yf (cm)</b>	<b>X(cm)</b>	<b>Y(cm)</b>
1	(1,1)	1.0	3.0	0.5	0.0
2	(1,1)	11.0	5.8	0.0	0.0
3	(1,1)	1.5	2.0	0.5	0.5
4	(1,1)	12.8	5.5	0.0	0.0
5	(1,1)	0.0	1.8	0.0	0.0
6	(1,1)	1.0	0.0	0.0	0.5
7	(1,1)	1.5	2.0	0.5	0.0
8	(1,1)	2.0	1.9	0.0	0.0
9	(1,1)	0.5	1.7	0.0	0.5
10	(1,1)	0.8	1.3	0.0	0.0

**Figure 2:** Navigation test (positioning precision)

Test Run #	Theta error final (in °)
1	3.8
2	14.0
3	4.5
4	21.0
5	2.0
6	0.0
7	-1.0
8	0.0
9	1.0
10	1.0

**Figure 3:** Navigation test (angle precision)

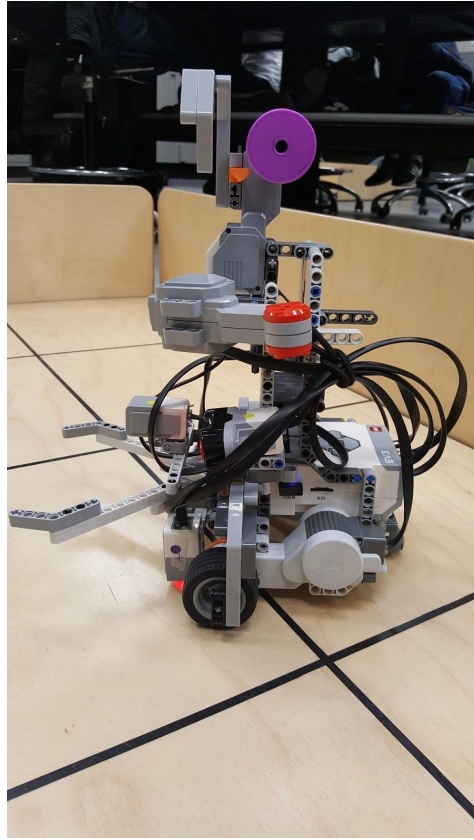
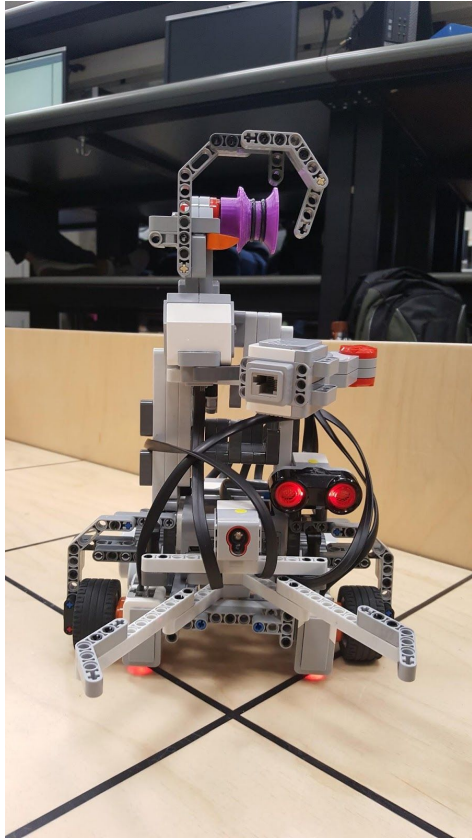
Test Run #	Theta final (after localization)	Theta after dismount
1	0.0	3.0
2	N/A	N/A
3	0.0	0.0
4	N/A	N/A
5	0.0	1.0
6	N/A	N/A
7	0.0	3.0
8	N/A	N/A

9	0.5	4.0
10	1.0	3.0

**Figure 4:** Dismounting localization (angle precision)

- 6) The brick successfully reaches the end point 60% of the test runs. Most of the times it failed is due to the navigation turns being off, which causes the robot to not properly localize in front of the zipline. Thus, the robot misses the zipline. This may be due to the wheel base or wheel radius values being off.
- 7) There should be an investigation about navigation turning to figure out why the turns are not accurate. Then, include these fixing in the next integration test, and verify that the zipline mounting success rate goes up.

## 2. HARDWARE



See *HARDWARE - 2.0*.