Team 01 Testing Document

**Project:** Obstacle track-racer – A competitive robot design project

**Task:** Design and construct a machine that can autonomously navigate to a race track on an island and complete as many laps as possible within a 5 minute period, eventually returning to its starting point.

**Document Version Number**: 2.0

**Date:** 12/03/2021

**Edit History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Editor | Date | Operation |
| 1.0 | Junjian Chen | 11/03/21 | The introduction, the table of contents and the brief test timeline. |
| 2.0 | Junjian Chen, Shichang Zhang | 12/03/21 | The test plan explanation, the general purpose and test ideas of each test.  Complete the initial test plan. |

### 

### Table Of Contents

[**1.0 Introduction**](#_heading=h.3znysh7) **2**

[**2.0 Test Plan**](#_heading=h.2et92p0) **2**

[**3.0 Test Record**](#_heading=h.tyjcwt) **2**

[**3.1 Hardware Test**](#_heading=h.3dy6vkm) **2**

[**3.2 UltrasonicLocalizer Test**](#_heading=h.1t3h5sf) **2**

[**3.3 LightLocalizer Test**](#_heading=h.4d34og8) **2**

[**3.4 Navigation Test**](#_heading=h.2s8eyo1) **2**

[**3.5 Obstacle Avoidance Test**](#_heading=h.17dp8vu) **2**

### 

### 1.0 Introduction

The objective of the testing document is to build a system that can help to evaluate and optimize the performance of our project design.

The testing document consists of two parts-Test Plan (Part 2.0) and Test Record (Part 3)

The Test Plan involves a timeline of tests to be performed by weeks. Each test has a simple description of its objective, the methods used to test .

The Test Record will record every test performed and be classified by its component. A record has the following information:date, tester, hardware/software version number, purpose, objective, procedure, expected result output, test report, conclusion, action need to be done and distribution.

### 2.0 Test Plan

In order to get a design that performs desirable functionality of client requirements, testing is an essential process of design. The testing plan of our project is divided into three main stages: tests on basic components, tests on combinational functionalities, tests on the completed design. These tests are from simple to complicated. They are carried step by step. The tests on basic components (ex. ultrasonic reading, moving forward for a long distance) will tell us whether these basic components are reliable when working individually. If we know these basic components are reliable, when we deal with comprehensive functionalities such as navigation, we can pay more attention to how to combine these components to work out the complex function instead of going back to check these components’ functionality. Except for these three main test stages, we will also conduct tests when a hardware or software component is integrated to know whether the integrated part of hardware or software is consistent with other components and works better than the old one. In the test plan, tests will be listed with some general information and a timeline that indicates when the test will be performed. The detailed information and implementation of listed tests is in the test record part of this testing document.

Our current hardware design generally satisfies our needs for the final project. So we will regard the current hardware design as our project hardware design. We will test the alternative hardware designs functionalities and compatibilities soon. During the development of software, if we encounter problems that must need the integration of hardware, we will switch the current hardware design to alternative hardware designs or integrate some component of current hardware design. At that time, we will conduct the test on integrated hardware components. This test is depending on the software development, so there is no clear timeline for testing integrated hardware components.

* **Hardware Component Development Test**

The test is focusing on the integrated part of the hardware. In this test we just run the tested software functions such as move straight or rotate and so on to test whether original desirable functionalities are damaged. And then we will run the software function that asks us to integrate the hardware component to check whether the integrated hardware can handle the problem.

Timeline for Tests:

**Week of 3.14(3.14-3.21)**

We will perform comprehensive tests on basic components and some simple functionality. The tests are expected to pass in a short period of time.

* **Ultrasonic Reading Test**

The test mainly tests the helper methods improving the ultrasonic sensor reading. Actually, it is also a test for whether the robot can detect an obstacle on the path. From previous labs, we observed that the accuracy of feedback data of the ultrasonic sensor was undesirable. The ultrasonic sensor reading tends to fluctuate considerably when the robot orients to open space. We have developed some methods to eliminate the inaccuracy. Since the project is processed in a new world with greater size, we need to test whether the helper methods we developed can assist us to get a precise ultrasonic reading.

For the test, the general idea is to put the obstacle on the path and examine whether the ultrasonic readings achieve our expectation. We will change the position, orientation of the robot to test some normal cases and extreme cases.

* **Rotating Test**

The test mainly tests whether the current hardware parameters such as base width and wheel radius are accurate enough.

For the test, the general idea is just to let the robot execute a series of rotating actions and check whether the final angle meets our expectation. For example, we will let the robot rotate 360° for 5 times and check whether the final angle is close to the start angle in a tolerable range.

* **Moving Test**

The test mainly tests the stability of current hardware design. From previous labs, we observed that when the robot moves for a long distance, the robot sometimes deviates from the planned path. The phenomenon is mostly caused by the inaccurate position of wheels, symmetry of the robot and so on. We want to check whether the current hardware design is able to help the robot run with a tolerable error.

For the test, we should firstly confirm that the Rotating Test passes. We will run the robot as a square driver seen in the lab 2 or a rectangular driver. But, we will run the robot for not one but many rounds of the square or rectangle. We will use relatively large side length to examine the extreme case.

* **Ultrasonic Localizer Test**

The test mainly tests whether Ultrasonic Localizer works accurately. In our Lab 5 Demo, we find that unknown errors existed when performing Ultrasonic Localization. To find the cause of it and in what circumstance it will happen, tests need to be performed. We need to confirm whether the ultrasonic Localizer can be directly used as a part of the final design.

For the test, we will simulate the first step of the project, placing the robot with the random initial angle in the corner of the world (red region or green region). And we will run the ultrasonic localizer to see whether the robot will finally orient to the angle we expect. We will choose plenty of different initial angles to test, and we expect that the final angle error is within 1-2 degrees.

* **Light Localizer Test**

The test mainly checks whether the Light Localizer can help the robot localize to the point we expect (the start point at beginning or the nearest point during the process of navigation).

For the test, we will perform two kinds of tests. The first one is that we will test after the ultrasonic localization, whether the Light Localizer can bring the robot to the start point. After passing the Ultrasonic Localizer Test, we will just let the robot use Light Localizer to localize to the start point. The second one is that we will test whether the light localizer can help to localize the nearest point while navigating. We will just let the robot move from a random start point to a random endpoint on the island. At the end of moving, we will let the robot use Light Localizer to localize to the endpoint.

* **Alternative Hardware Design Test**

The test will examine whether the alternative hardware designs can meet our needs and whether they are compatible with current software design.

Our hardware team currently has two alternative hardware designs.

During this test, we will apply our software to these two models and compare their performances with our current used hardware design to observe whether they meet our expectation.

* **Obstacle Avoidance Test**

This test focuses on evaluating the performance of obstacle avoidance functionality. The expected obstacle avoidance system enables the robot to avoid the obstacle on the path regardless of how the obstacle is placed, the angle the robot is facing.

In this test we will change the position of the obstacle relative to the robot to examine the robot to limits and find out the case that our obstacle avoidance system does not work. For example, we will place an obstacle on the navigation path and place another obstacle near the one on the path. In this condition, the robot is expected to not only avoid the obstacle on the path, but also avoid the obstacle closed to the former obstacle when avoiding the former obstacle. .

**Week of 3.22(3.22-3.28)**

* Navigation System Development Test

The test mainly checks whether the navigation system can function desirably in the final project. Our current Navigation System can localize the robot to waypoints with low errors. However, the large time consumption of it will lower the performance of the robot. The Navigation System software will be modified and different versions of it will be tested and evaluated.

For the test, similar to lab 5, we will pass some random way points to our design. We expect the robot to move to these way points in order. We expect to observe the error is within the same range of the current design. But we want the robot to complete the map for a much shorter time. Moreover, since the final project asks the robot to run the laps as many as possible, we will also conduct a test on whether the nagevation system can assist the robot to run a map for multiple times and finally end with small errors.

* **Obstacle Avoidance Test**

If we fail the test in the previous week, and we do some integrations on our design, we will apply similar tests to check whether the improved design can pass the test.

**Week of 3.29(3.29-4.4)**

* **Complete System Test**

The test is examining whether the complete system can meet the client requirements. The test ideas for this test will be detailed when we get clear final project requirements.

**Week of 4.5(4.5-4.11)(Final Week)**

* **Complete System Test**

Continue to check the performance of our complete system. According to the test results, keep working on improving our design.

### 3.0 Test Record

### 3.1 Hardware Test

### 3.2 UltrasonicLocalizer Test

### 3.3 LightLocalizer Test

### 3.4 Navigation Test

### 3.5 Obstacle Avoidance Test

template :

Date : 2021/3/13

Tester:

Author:

Hardware version: 1.0

Software version: 1.0

Purpose of Test: Determine whether the ultrasonic localizer designed in the previous labs can be directly used in the project.

Test Objectives: Determine whether the ultrasonic localizer can accurately localize.

Procedure:

Expected Result:

Output:

Test Report:

Conclusion:Pass/Fail?

Action:

Distribution:software development, project management

### 