|  |
| --- |
| Rochester Institute of Technology |
| Final Project: Cyclometer |
| Test Plan & Test Results |
|  |
| **Nsama Chipalo & Jenny Zhen** |
| **5/20/2014** |

|  |
| --- |
|  |

Table of Contents

[Introduction 2](#_Toc388354490)

[Purpose of the Test Plan 2](#_Toc388354491)

[Audience of the Test Plan 2](#_Toc388354492)

[Current State of the Project 2](#_Toc388354493)

[Test Plan 2](#_Toc388354494)

[Type of Testing 2](#_Toc388354495)

[Scope 2](#_Toc388354496)

[Critical Areas to Focus 2](#_Toc388354497)

[Entry and Exit Criteria 2](#_Toc388354498)

[Pass/Fail Criteria 2](#_Toc388354499)

[Test Cases 3](#_Toc388354500)

[Reflection of Design Choices 5](#_Toc388354501)

[Changes Made in Design since the Design Document 5](#_Toc388354502)

[Changes in Design, if Project was redone 6](#_Toc388354503)

# Introduction

## Purpose of the Test Plan

This document states and tracks the appropriate information for effectively and thoroughly testing the Final Project, Cyclometer. The goal of this test plan is to define the areas of the system that will be tested, as well as, the areas that will not be tested, and to track the expected results and outcome.

## Audience of the Test Plan

Affected audiences of this test plan document include:

* Developers (and Testers) of the Final Project: Cyclometer
* Users of the Final Project: Cyclometer

## Current State of the Project

# Test Plan

## Type of Testing

The primary type of testing that will be described in the test plan is system testing. The reason for this is to place a focus on ensuring that the complete product meets the end-user requirements and functional specifications. Additionally, black box testing allows the developers and testers to focus on what the end-user will see.

## Scope

System testing assumes that unit and integration testing has been done (Unit and integration testing was briefly performed prior to system testing, but without any formal procedures). Scope of the system testing follows the scope of a black box testing procedure, where no knowledge of the inner design of the code or logic needs to be known.

## Critical Areas to Focus

The critical areas to focus include:

* Initializing the system correctly.
* Performing calculations correctly.

## Entry and Exit Criteria

System testing may begin once a majority of the features have been implemented. System testing will end once all test cases have been gone through by the developers, who are also the testers for this project.

## Pass/Fail Criteria

A test case will pass if it meets all the criteria listed in the expected outcome. If the actual results do not match completely and is a minor issue, it may be marked as neutral. Otherwise, the test case will fail if the criteria are not implemented, or are implemented incorrectly.

## Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Suite Name** | **Pre-conditions** | **Steps to Test** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| 1 | System Reset | 1. Cyclometer is powered off (system is to be initially started). | 1. Power on the cyclometer. | 1. Cyclometer enters manual mode with trip calculations turned off.  2. Trip values are set to 0.  3. User needs to select miles or kilometers using the Steps to Test described in Test Case #3.  4. User needs to set tire circumference using the Steps to Test described in Test Case #4.  5. Cyclometer does #1-4 in the given order, and then displays the current and average speeds. | Expected | Pass |
| 2 | System Reset | 1. Cyclometer is powered on. | 1. Press the Mode, Start/Stop, and Set buttons simultaneously for at least 2 seconds. | Expected Results are the same as Test Case #1. | Expected | pass |
| 3 | Initial Settings | 1. Cyclometer is powered on.  2. Cyclometer is selecting miles or kilometers. | 1. Confirm display shows a 1.  2. Press the Mode button once.  3. Press the Mode button once.  4. Press the Set button once. | 1. Pressing the Mode button toggles between 1 (kilometers) and 2 (miles).  2. Pressing the Set button sets the units. | Expected | pass |
| 4 | Initial Settings | 1. Cyclometer is powered on.  2. Cyclometer is selecting tire circumference. | 1. Confirm display shows the current value, or 210.  2. Press the Mode button once.  3. Hold the Mode button down for a few minutes.  4. Press the Set button once. | 1. Pressing the Mode button once increases the value by 1.  2. Holding the Mode button down for at least 2 seconds increases the value at a rate of 1 per second.  3. The maximum value is 220.  4. The minimum value is 190.  5. The value goes to the minimum value when the maximum is reached.  6. Pressing the Set button sets the circumference. | Expected | pass |
| 5 | Settings Modification | 1. Cyclometer is powered on and initialized.  2. Cyclometer is displaying the distance. | 1. Press the Set button once.  2. Set the Tire circumference using the Steps to Test in Test Case #4. | 1. Cyclometer allows setting of the tire circumference. It first displays the current circumference setting.  2. Setting of the circumference is as described in the Expected Results of Test Case #4.  3. The cyclometer continues performing trip calculations until the Set button is pressed.  4. The wheel rotations continue to be displayed until the Set button is pressed.  5. The new circumference is stored when the Set button is pressed. | Expected | pass |
| 6 | Start/Stop Calculations | 1. Cyclometer is powered on.  2. Cyclometer is in Manual mode.  4. Cyclometer is performing calculations. | 1. Press the Start/Stop button once.  2. Press the Start/Stop button once, again. | 1. Cyclometer stops the calculations on the first press. Cyclometer turns the Auto LED off. Cyclometer continues to compute the current speed.  2. Cyclometer starts the calculations on the second press. Cyclometer blinks the Auto LED with a 1 Hz rate. | Expected | pass |
| 7 | Start/Stop Calculations | 1. Cyclometer is powered on.  2. Cyclometer is in Auto mode. | 1. Perform wheel rotations.  2. Stop wheel rotations. | 1. Cyclometer starts the calculations when wheel rotations are performed. Cyclometer blinks the Auto LED with a 1 Hz rate.  2. Cyclometer stops the calculations when wheel rotations are stopped. Cyclometer lights the Auto LED steady on. Cyclometer continues to compute the current speed. | LED not lit | fail |
| 8 | Set Auto/ Manual | 1. Cyclometer is powered on.  2. Cyclometer is displaying either the elapsed time or speed. | 1. Press the Set button once.  2. Press the Set button once, again. | 1. Pressing the Set button toggles between Auto mode and Manual Model. | Expected | pass |
| 9 | Cycle Displays | 1. Cyclometer is powered on.  2. Cyclometer is displaying the current speed/average speed. | 1. Press the Mode button once.  2. Press the Mode button once, again.  3. Press the Mode button, one more time. | 1. Cyclometer displays the distance on the first press.  2. Cyclometer displays the elapsed time on the second press.  3. Cyclometer displays the current speed/average speed on the third press.  4. Cyclometer displays updated values when cycled. | Expected | pass |
| 10 | Reset Calculations | 1. Cyclometer is powered on.  2. Cyclometer is not performing either a System Reset or a Settings Modification. | 1. Press the Mode and Start/Stop buttons simultaneously for at least 2 seconds. | 1. Cyclometer resets the trip values to 0. | Expected | pass |
| 11 | Display Wheel Activity | 1. Cyclometer is powered on. | 1. Perform wheel rotations.  2. Stop wheel rotations. | 1. Cyclometer blinks the Wheel LED with a 1 Hz rate, when wheel rotations are performed.  2. Cyclometer turns off the Wheel LED, when wheel rotations are stopped. | Passes as the wheel led does blink. Though the it does not when setting circumference. | Fail/pass |
| 12 | Display Units | 1. Cyclometer is powered on.  2. Cyclometer is set to use kilometers. | 1. Confirm cyclometer has the Units LED off. | 1. Cyclometer has the Units LED off for kilometers. | Expected | pass |
| 13 | Display Units | 1. Cyclometer is powered on.  2. Cyclometer is set to use miles. | 1. Confirm cyclometer has the Units LED steady on. | 1. Cyclometer has the Units LED steady on for miles. | Expected | pass |
| 14 | Display Speed | 1. Cyclometer is powered on.  2. Cyclometer is displaying current speed. | 1. Perform wheel rotations, at a moderate speed.  2. Perform wheel rotations, at a very high speed.  3. Stop wheel rotations. | 1. Cyclometer displays a value between 6.8 and 99, exclusively, at a moderate speed.  2. Cyclometer displays a value of 99, at a very high speed.  3. Cyclometer displays a value of 0, at no speed.  3. Cyclometer shows 6.8 at 1hz. | Expected | pass |

# Reflection of Design Choices

## Changes Made in Design since the Design Document

The Reason for all our changes since the design documentation was the need to implement the state walker pattern. Prior to use using the state walker you design did not use any obvious pattern, as such when we decided to use the state walker we had to change everything. As per the requirements of the state walker our new design contains a context class, State nodes, transition objects and an iterator.



NOTE: the ModeButtonTransition is not properly named. It supports transitions of all button presses. It should be seen as just a transition class that accepts multiple events from multiple classes. The reason it is like this is because the original design called for a specific transition class for each type of transition. However as implementation was underway the ModeButtonTransition class came to compartmentalize all the functionality of transitions. At the time refactoring the name change was not seen as a priority and was left as is.

The sate node is abstract enough that various states can be created, however that is not the case with the mode button class. Another difference with the design in comparison to the actual state walker is that out transition cannot send messages to the state nodes to tell them to tell the iterator to change states. In the design this is accomplished via the transition telling the iterator to change states

The state chart diagram has not received any changes since it was turned in.

## Changes in Design, if Project was redone

The first change I would make would be to have a central location that controls the writing to the board. Having each state class control their writing to the board later on caused issues with managing the blinking of the LEDs. I now see that having a central location to handle the writing to the board would have prevented other threads from stepping on each other’s toes even with the best of synchronizing. Next would be to properly implement the state walker without the couple of deviations introduced.