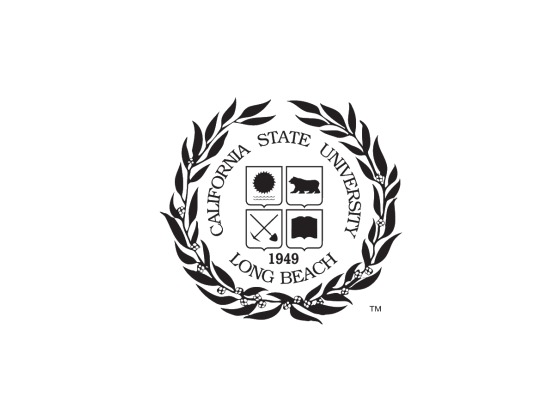
**California State University, Long Beach, Department of Electrical Engineering**

**EE 400D Verification Test Plan, Spring 2020**

Submitted on 05/10/2020

****

****

**Project:** Mini-Rosco

**Project Manager:** Alex Margaris (014575156)

**Electronics & Control Engineer:** Giann Bullo (015728113)

**Mission, Systems, and Test Engineer:** Alex Margaris (014575156)

**Manufacturing Engineer:** Darren Chan (015864587)

Table of Contents

[1 Introduction 3](#_Toc479597921)

[1.1 Purpose 3](#_Toc479597922)

[1.2 Project ConOps/Mission 3](#_Toc479597923)

[1.3 Document Overview 3](#_Toc479597924)

[2 Applicable Documents 4](#_Toc479597925)

[3 Testing Methodology 5](#_Toc479597926)

[3.1 Master Verification and Validation (V&V) Matrix 5](#_Toc479597927)

[3.2 Testing Types and Methods 6](#_Toc479597928)

[3.2.1 Verification by Analysis 6](#_Toc479597929)

[3.2.2 Verification by Demonstration 6](#_Toc479597930)

[3.2.3 Verification by Inspection 6](#_Toc479597931)

[3.2.4 Verification by Test 7](#_Toc479597932)

[3.2.5 Validation by Analysis 7](#_Toc479597933)

[3.2.6 Validation by Demonstration 7](#_Toc479597934)

[3.2.7 Validation by Inspection 7](#_Toc479597935)

[3.2.8 Validation by Test 7](#_Toc479597936)

[3.3 Master Test Case List 7](#_Toc479597937)

[3.3.1 TC-01: Mini-Rosco Basic Operations 8](#_Toc479597938)

[3.3.2 TC-02: Mini-Rosco Assembly and Disassembly 8](#_Toc479597939)

[3.3.3 TC-03: Mini-Rosco Mission Operation 8](#_Toc479597940)

[3.3.4 TC-04: Mini-Rosco Logistical Elements 8](#_Toc479597941)

[4 Test Procedures 9](#_Toc479597943)

[4.1 TC-XX:xxxxxxx(Title) 9](#_Toc479597944)

[4.1.1 Detailed Description 9](#_Toc479597945)

[4.1.2 Test Case Traceability and Pass/Fail Matrix 9](#_Toc479597946)

[4.1.3 Detailed Success Criteria 9](#_Toc479597947)

[4.1.4 Test Environment 9](#_Toc479597948)

[4.1.5 Assumptions and Preconditions 10](#_Toc479597949)

[4.1.6 Procedure Steps 10](#_Toc479597950)

[5 Appendices 11](#_Toc479597951)

[5.1 xxxxxxx 11](#_Toc479597952)

[5.2 xxxxxxx 11](#_Toc479597953)

[5.3 xxxxxxx 11](#_Toc479597954)

[5.4 xxxxxxx 11](#_Toc479597955)

# Introduction

This is the Mini-Rosco Spring 2020 Verification Test Plan

## Purpose

The purpose of this document is to provide a comprehensive Verification (V) Test Plan of the Spring 2020 Mini-Rosco, including the Project ConOps/Mission, Test Methodology, Verification and Validation Matricies, and Test Cases.

## Project ConOps/Mission

The Mini-Rosco is the first generation Mini-Rosco and will enable trade show specialists to capture footage of Arxterra, Humans for Robots, and EE 400D robots. To meet our Program Objectives the first generation of Mini-Rosco will introduce new design features such as autonomous locating features, autonomous following features, and a miniaturized form factor than that of Rosco. The objective of the Mini-Rosco project is to create a robot that can assist the TRC company in trade shows in order to capture footage showcasing the capabilities of the 3DoT board. There will be Arxterra and ArxRobot controls integrated into the overall design.

The prototype Mini-Rosco is designed for robot enthusiasts to use for footage of other robots. Footage captured via a smartphone’s camera mounted on Mini-Rosco can be viewed in a real-time stream and by everyone logged into our telepresent application.

Another robot's location will be acquired by Mini-Rosco via the Arxterra and ArxRobot applications. Mini-Rosco’s position will be acquired, in addition to another robot’s position.  Mini-Rosco will then make any necessary calculations in order to move to the other robot's location within a maze. The other robot’s location will be transmitted to the user through the use of a custom GPS PCB.

Once Mini-Rosco reaches the other robot's location, Mini-Rosco will demonstrate fully autonomous following of another robot through a maze in order to capture video footage at different angles. Mini-Rosco’s handheld size allows it to maintain speeds faster than or equal to that of other robots, while maintaining stability required to securely hold a smartphone and maneuverability required to follow the TRC robots at the convention (built in class).

## Document Overview

This document is organized as follows:

* Section 2 contains links to relevant and applicable project reference documents and presentations for this Test Plan.
* Section 3 contains a description of the Testing Methodology utilized in this Test Plan, including the Master Verification and Validation Matrix, a description of the 4 types of V&V testing performed, the Test Environment(s) description(s), and a Master Test Case List of all 4 Test Cases for this project.

# Applicable Documents

This section contains a table of all relevant and applicable project reference documents and presentations for the Mini-Rosco Spring 2020 Verification and Validation Test Plan.

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Document Description** | **Document Link** |
| Trade Study #1 | Trade study involving GPS Module | [Link](https://www.arxterra.com/?p=159439&preview=true) |
| GPS PCB Design Blog Post | Details on the design process of the GPS PCB | [Link](https://www.arxterra.com/?p=159492&preview=true) |
| PDR | Preliminary Design Review Presentation. Contains L1 and System Block Diagram, Resource Allocation Reports, Recovery Plan, and design methodology | [Link](https://docs.google.com/presentation/d/1ntCX-vsxMV4u-Rq4VdlW_eGogdiW6tdGl_neRTcX8xI/edit?usp=sharing) |
| Ultrasonic Testing Blog Post | Testing of the Ultrasonic Sensor | [Link](https://www.arxterra.com/?p=161609&preview=true) |
| Final Project Code | Final Document of all MCU Firmware Code. | (Paste Google Drive Link here) |
| Final Project Summary (final blog post) | Final Presentation of completed Project. Contains all material included in PDR plus detailed descriptions of all processes employed by Mini-Rosco | [Link](https://www.arxterra.com/?p=153570&preview=true) |
| NASA Systems Engineering Handbook (2007) | Document containing Test Methodologies in Section 3 | <http://www.acq.osd.mil/se/docs/NASA-SP-2007-6105-Rev-1-Final-31Dec2007.pdf> |

# Testing Methodology

This section contains the Master Verification and Validation Matrix, as well as detailed descriptions of the various Test Methods and Test Cases utilized in this Test Plan.

## Master Verification Matrix

This matrix provides complete traceability of every requirement. Specifically, every requirement is mapped to its description, success criteria, V&V testing designation and method, and Test Case(s) where the requirement will be tested. Note that some overlap between Test Cases’ requirements V&V is okay. **ALL REQUIREMENTS MUST BE MAPPED HERE AND ACCOUNTED FOR.**

**\*NOTE\* the verification matrix below also includes constraints, as they are considered L1 Functional Requirements.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement Number** | **Requirement Text** | **V Success Criteria** | **V Method**  (Analysis, Demonstration, Inspection, Test?) | **Test Case(s) where Requirement is Tested** |
| L.1.1 | Mini-Rosco shall operate for a maximum of 30 minutes. It should be noted that this 30 minute timeframe begins only after the GPS PCB has acquired a signal. This may take an average of 10 minutes. | Record how long each part of the process takes to complete. Sum of all three recordings is < 30min | Demonstration | TC-03 |
| L.1.2 | Mini-Rosco shall be able to articulate the camera feed using a pan angle of 360 degrees and a tilt angle of 135 degrees using the mounted iOS/Android smartphone. These are the optimal camera degree field of views due to the fact that Mini-Rosco will be filming other moving robots. The tallest robot is 2 feet, the booth is 10 feet wide and the Mini-Rosco will be located inside the booth. | Use protractor to see if angles can be accomplished | Test | TC-01 |
| L.1.3 | Mini-Rosco shall have the ability to make turns to the left and to the right within the range of 0-360 degrees with an accuracy of +/-10 degrees. This is necessary for navigating tight turns within the booth. | Use protractor to see if angle can be accomplished | Test | TC-01 |
| L.1.4 | Mini-Rosco shall have the option of being controlled in Remote Control mode. All movement requirements shall be demonstrated remotely. This will be done through the ArxRobot application. | See if commands sent from app trigger an appropriate response | Test | TC-01 |
| L.1.5 | Mini-Rosco shall be controlled for the duration of the mission through the Arxterra control panel. Mini-Rosco is operated from our video booth. | See if commands sent from Arxterra control panel trigger appropriate response | Demonstration | TC-03 |
| L.1.6 | Mini-Rosco will be able to accept user input through a waypoint for a specific destination to travel to. | User can enter destination information and sees waypoint on control panel | Test | TC-03 |
| L.1.7 | Mini Rosco shall be able to navigate to specific locations based on GPS coordinates within an accuracy of a minimum of 4 ft. | If received waypoints cause robot to reach specified destination | Demonstration | TC-03 |
| L.1.8 | Mini-Rosco shall recognize another robot within a 4 feet radius using an ultrasonic sensor after arriving at the specified destination.  The other robot that is being followed is 1 foot in height. | Verify if ultrasonic can detect within 4 feet in front of it | Test | TC-03 |
| L.1.9 | Mini-Rosco shall maintain a following distance of 1 to 2 feet after locating another robot. The other robot’s speed is within the range of 1 to 3 miles per hour. | Use ruler and hold it near Mini-Rosco while following | Demonstration | TC-03 |
| L.1.10 | Shall have autonomous navigation mode to travel to destination | Robot will arrive at the destination | Demonstration | TC-03 |
| L.1.11 | Mini-Rosco shall avoid obstacles up to 70cm away while in autonomous mode. | Robot will avoid obstacles in autonomous mode | Test | TC-03 |
| L.1.12 | Mini-Rosco shall execute its mission in three stages. The first stage will be the Locating stage seen in requirement L.1.7. The second stage will be the Acquisition stage see in requirement L.1.8. The third stage will be the Following stage seen in requirement L.1.9. | Stages are executed in order | Demonstration | TC-03 |
| L.1.13 | Mini-Rosco shall employ a custom PCB to extend the functions of the 3DoT by allowing control over where Mini-Rosco travels via a set of coordinates | Is the board fabricated and does the GPS work | Test | TC-03 |
| L.1.14 | Mini-Rosco shall be a smaller scale model of 70% to 80% the size of Rosco for storage purposes. | Perform measurements and confirm that it is 70% to 80% smaller | Test | TC-02 |
| L.1.15 | Mini-Rosco will be designed such that the 3DoT board fits onto the chassis. | Does the 3DoT fit into the designated spot | Test | TC-01, TC-02 |
| L.1.16 | Mini-Rosco’s expenditures shall not exceed $500. | Show a cost resource report | Inspection | TC-04 |
| L.1.17 | Mini-Rosco will be designed such that the same number of motors and servos are equal to that of the original Rosco. | Are the number of motors and servos equal | Inspection | TC-02 |
| L.1.18 | Mini-Rosco will utilize the 3DoT board to control the robot. | Is the 3DoT controlling anything | Inspection | TC-01 |
| L.1.19 | Mini-Rosco shall provide a live stream video feed via the Arxterra control panel throughout the duration of its mission. | Can a video feed be seen | Demonstration | TC-01, TC-03 |
| L.1.21 | Mini-Rosco shall contain a custom 3DoT shield incorporating interface electronics between the 3DoT board and sensors and/or actuators unique to the robot. | Is a 3DoT present and does it interface with the ultrasonic sensor on Mini-Rosco. | Demonstration | TC-03 |
| L.1.22 | Mini-Rosco’s form factor will be completed by the date of May 15, 2020 | Is Mini-Rosco’s form factor finalized | Test | TC-02 |
| L.1.23 | Mini-Rosco shall be disassembled and reassembled in 20 minutes or less. | Time the disassembly and reassembly of Mini-Rosco, including motors, servos, the 3DoT, and the PCB | Demonstration | TC-02 |
| L.1.24 | Mini-Rosco shall be designed such that there are no dangling wires. | Brush a finger over the top of Mini-Rosco’s chassis to verify that it doesn’t get caught | Inspection | TC-02 |
| L.1.25 | Mini-Rosco shall incorporate the 3DoT v9.05 or later series of boards. | Look at the back of the 3DoT board | Inspection | TC-02 |
| L.1.26 | Software shall be written in the Arduino [De facto Standard](https://www.arduino.cc/reference/en/) scripting language and/or using the GCC C++ programming language, which is implements the ISO C++ standard ([ISO/IEC 14882:1998](https://www.iso.org/standard/25845.html)) published in 1998, and the 2011 and 2014 revisions. | Look at the code and verify that it is written in c++ | Test | TC-01 |
| L.1.27 | Mini-Rosco shall be controlled via Bluetooth 4.0 in compliance with the Bluetooth Special Interest Group (SIG) Standard (supersedes IEEE 802.15.1). | Verify that a Bluetooth connection is established between the 3DoT and the cell phone. | Analysis | TC-01 |
| L.1.28 | Manufacturability of 3D printed robots shall utilize as few files as possible when 3D Printing the Robot and its parts. | Ensure that there are not more than 10 SolidWorks files when 3D Printing. | Inspection | TC-04 |
| L.1.29 | All Lithium (Li-ion, Li-polymer) batteries shall be purchased with and stored, when not in use, in a fire and explosion proof battery bag. | Ensure that battery is stored in correct bag when not in use. | Inspection | TC-04 |

## Testing Types and Methods

This subsection contains the 4 types of Verification and Validation (V&V) testing, as derived from the NASA Systems Engineering Handbook referenced above in Section 2. Material is taken from Chapter 5 in the NASA Handbook, and replicated below.

**Verification** proves that a realized product for any system model within the system structure conforms to the build-to requirements (for software elements) or realize-to specifications and design descriptive documents (for hardware elements, manual procedures, or composite products of hardware, software, and manual procedures). In other words, Verification is requirements driven; verification shows proof of compliance with requirements; that the product can meet each “shall” statement as proven through performance of a test, analysis, inspection, or demonstration.

**Validation** is conducted under realistic conditions (or simulated conditions) on any end product for the purpose of determining the effectiveness and suitability of the product for use in mission operations by typical users; and the evaluation of the results of such tests. Testing is the detailed quantifying method of both verification and validation. However, testing is required to validate final end products to be produced and deployed. In other words, Validation is ConOps/Mission driven; validation shows that the product accomplishes the intended purpose in the intended environment; that product meets the expectations of the customer and other stakeholders as shown through performance of a test, analysis, inspection, or demonstration.

### Verification by Analysis

The use of mathematical modeling and analytical techniques to predict the suitability of a design to stakeholder expectations based on calculated data or data derived from lower system structure end product verifications. Analysis is generally used when a prototype; engineering model; or fabricated, assembled, and integrated product is not available. Analysis includes the use of modeling and simulation as analytical tools. A model is a mathematical representation of reality. A simulation is the manipulation of a model.

### Verification by Demonstration

Showing that the use of an end product achieves the individual specified requirement. It is generally a basic confirmation of performance capability, differentiated from testing by the lack of detailed data gathering. Demonstrations can involve the use of physical models or mockups; for example, a requirement that all controls shall be reachable by the pilot could be verified by having a pilot perform flight-related tasks in a cockpit mockup or simulator. A demonstration could also be the actual operation of the end product by highly qualified personnel, such as test pilots, who perform a one-time event that demonstrates a capability to operate at extreme limits of system performance, an operation not normally expected from a representative operational pilot.

### Verification by Inspection

The visual examination of a realized end product. Inspection is generally used to verify physical design features or specific manufacturer identification. For example, if there is a requirement that the safety arming pin has a red flag with the words “Remove Before Flight” stenciled on the flag in black letters, a visual inspection of the arming pin flag can be used to determine if this requirement was met.

### Verification by Test

The use of an end product to obtain detailed data needed to verify performance, or provide sufficient information to verify performance through further analysis. Testing can be conducted on final end products, breadboards, brass boards or prototypes. Testing produces data at discrete points for each specified requirement under controlled conditions and is the most resource-intensive verification/validation technique. As the saying goes, “Test as you fly, and fly as you test.” (See Subsection 5.3.2.5.).

### Validation by Analysis

The use of mathematical modeling and analytical techniques to predict the suitability of a design to stakeholder expectations based on calculated data or data derived from lower system structure end product validations. It is generally used when a prototype; engineering model; or fabricated, assembled, and integrated product is not available. Analysis includes the use of both modeling and simulation.

### Validation by Demonstration

The use of a realized end product to show that a set of stakeholder expectations can be achieved. It is generally used for a basic confirmation of performance capability and is differentiated from testing by the lack of detailed data gathering. Validation is done under realistic conditions for any end product within the system structure for the purpose of determining the effectiveness and suitability of the product for use in NASA missions or mission support by typical users and evaluating the results of such tests.

### Validation by Inspection

The visual examination of a realized end product. It is generally used to validate physical design features or specific manufacturer identification.

### Validation by Test

The use of a realized end product to obtain detailed data to validate performance or to provide sufficient information to validate performance through further analysis. Testing is the detailed quantifying method of both verification and validation but it is required in order to validate final end products to be produced and deployed.

## Master Test Case List

A **Test Case** can be described as *a scenario containing a sequence of detailed test steps, in order to perform verification/validation testing on multiple requirements that are similar in nature.*

For example, if a group has multiple requirements regarding starting up their robot project, they can group all these requirements to be verified/validated in a single test case. Similarly, if a group has multiple requirements that can be verified/validated via inspection, they can group all of them together in a single test case.

The purpose of this subsection is to provide a High-Level overview of all Test Cases utilized in this Test Plan. Each item in this subsection will contain the following: Test Case Number and Name, High Level Scenario Description, and Test Environment Description.

**The best way to approach Test Cases is to GROUP REQUIREMENTS THAT ARE SIMILAR IN NATURE FIRST, then write procedure steps for each Test Case.**

### TC- 01: Mini-Rosco Basic Operations

Description: This test case is used to verify requirements related to functions such as basic movement. Not related to mission.

Test Environment: Living Room

### TC-02: Mini-Rosco Assembly and Disassembly

Description: This test case is used to verify the requirements related to the assembly, disassembly, and visual inspection of Mini-Rosco. Not related to mission.

Test Environment: Living Room

### TC-03: Mini-Rosco Mission Operation

Description: This test case is used to verify the requirements related to the mission operations of Mini-Rosco. This includes the three stages of the mission.

Test Environment: Living Room

### TC-04: Mini-Rosco Logistical Elements

Description: This test case is used to verify requirements related to parameters such as cost and deadline. In addition, requirements not verifiable by TC-01, TC02, or TC03 will be classified here.

Test Environment: Living Room

# Test Procedures

This section contains details of every Test Case utilized for V&V of project requirements. Each Test Case subsection within this section will contain the following: Test Case number and name, detailed scenario description, Test Case Traceability Matrix, detailed success criteria, detailed Test Environment description, Test Assumptions/Preconditions, Detailed Test Procedure Steps, and a Pass/Fail Matrix of success criteria per Test Case.

## TC-01: Mini-Rosco Basic Operations

### Detailed Description

In this test case, Mini-Rosco’s , motors and servos are connected to a 3DoT board. Then, the ArxRobot application is connected to the 3DoT board via Bluetooth. Finally, Mini-Rosco will have input commands sent to its motors and servos in order to verify and validate that they meet the specified requirements.

### Test Case Traceability and Pass/Fail Matrix

This matrix shall show all requirements that are being tested in this test case. The Pass/Fail Column is populated after the Test Case has been run via the Procedure Steps.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement Number** | **Requirement Text** | **V Success Criteria** | **V Method**  (Analysis, Demonstration, Inspection, Test?) | **Procedure Step(s) where Requirement is tested** | **Pass/Fail?** |
| L.1.2 | Mini-Rosco shall be able to articulate the camera feed using a pan angle of 360 degrees and a tilt angle of 135 degrees using the mounted iOS/Android smartphone. These are the optimal camera degree field of views due to the fact that Mini-Rosco will be filming other moving robots. The tallest robot is 2 feet, the booth is 10 feet wide and the Mini-Rosco will be located inside the booth. | Use protractor to see if angles can be accomplished | Test | 7 |  |
| L.1.3 | Mini-Rosco shall have the ability to make turns to the left and to the right within the range of 0-360 degrees with an accuracy of +/-10 degrees. This is necessary for navigating tight turns within the booth. | Use protractor to see if angle can be accomplished | Test | 4 |  |
| L.1.4 | Mini-Rosco shall have the option of being controlled in Remote Control mode. All movement requirements shall be demonstrated remotely. This will be done through the ArxRobot application. | See if commands sent from app trigger an appropriate response | Test | 4 |  |
| L.1.15 | Mini-Rosco will be designed such that the 3DoT board fits onto the chassis. | Does the 3DoT fit into the designated spot | Test | 1 |  |
| L.1.18 | Mini-Rosco will utilize the 3DoT board to control the robot. | Is the 3DoT controlling anything | Inspection | 4 |  |
| L.1.19 | Mini-Rosco shall provide a live stream video feed via the Arxterra control panel throughout the duration of its mission. | Can a video feed be seen | Demonstration | 6 |  |
| L.1.26 | Software shall be written in the Arduino [De facto Standard](https://www.arduino.cc/reference/en/) scripting language and/or using the GCC C++ programming language, which is implements the ISO C++ standard ([ISO/IEC 14882:1998](https://www.iso.org/standard/25845.html)) published in 1998, and the 2011 and 2014 revisions. | Look at the code and verify that it is written in c++ | Inspection | 8 |  |
| L.1.27 | Mini-Rosco shall be controlled via Bluetooth 4.0 in compliance with the Bluetooth Special Interest Group (SIG) Standard (supersedes IEEE 802.15.1). | Verify that a Bluetooth connection is established between the 3DoT and the cell phone. | Analysis | 3 |  |

### Detailed Success Criteria

* + Use protractor to see if desired pan/tilt angles can be accomplished
  + Use protractor to see if desired turning angles can be accomplished
  + See if commands sent from ArxRobot application trigger an appropriate response with Mini-Rosco
  + See if commands sent from the Arxterra control panel trigger an appropriate response with Mini-Rosco
  + See if 3DoT fits into the designated position
  + See if the 3DoT is controlling anything.
  + User can see a video feed

### Test Environment

This test case will take place and will be executed in Giann’s Living room within his apartment.

### Assumptions and Preconditions

* + Mini-Rosco is fully built and basic software is integrated for this use case
  + Mini-Rosco is connected to the ArxRobot application
  + Mini-Rosco is connected to the Arxterra application in community mode
  + 3DoT board can be easily placed on the chassis and all necessary wired connections can be made

### Procedure Steps

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step Number** | **Step Description** | **Pass Criteria** | **Recorded Data** | **Requirement(s) Tested** | **Test Type** | **Test Method** |
| 1 | Place 3Dot onto chassis behind the servo tower | 3DoT must fit onto the chassis |  | L.1.15 | Inspection | Test |
| 2 | Connect motors and servos to 3DoT |  |  |  |  |  |
| 3 | Connect phone to 3DoT | Phone connects to 3DoT via Bluetooth |  | L.1.27 | Analysis | Analysis |
| 4 | Manually turn Mini-Rosco to the right and then to the left using the ArxRobot application | Mini-Rosco will execute desired movements |  | L.1.18, L.1.4, L.1.3 | Test | Inspection, Test, Test |
| 5 | Place phone app in community mode and place it in the phone holder on Mini-Rosco |  |  |  |  |  |
| 6 | Access control panel on a computer and activate video feed | Live video feed should be seen within control center |  | L.1.19 | Demonstration | Demonstration |
| 7 | Issue pan and tilt commands through the control center | Servos should exhibit appropriate responses |  | L.1.2 | Test | Test |
| 8 | Look at raw code controlling the robot | Code should be written in C++ |  | L.1.26 | Inspection | Inspection |

## TC-02: Mini-Rosco Assembly and Disassembly

### Detailed Description

In this test case, Mini-Rosco has a 3DoT board placed onto the appropriate place on the chassis. Then Mini-Rosco has its motors, servos, and PCB connected to the 3Dot. Once this has all been done, Mini-Rosco will now be disassembled to its original state prior to this test case.

### Test Case Traceability and Pass/Fail Matrix

This matrix shall show all requirements that are being tested in this test case. The Pass/Fail Column is populated after the Test Case has been run via the Procedure Steps.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement Number** | **Requirement Text** | **V Success Criteria** | **V Method**  (Analysis, Demonstration, Inspection, Test?) | **Procedure Step(s) where Requirement is tested** | **Pass/Fail?** |
| L.1.14 | Mini-Rosco shall be a smaller scale model of 70% to 80% the size of Rosco for storage purposes. | Perform measurements and confirm that it is 70% to 80% smaller | Test | 6 |  |
| L.1.15 | Mini-Rosco will be designed such that the 3DoT board fits onto the chassis. | Does the 3DoT fit into the designated spot | Test | 3 |  |
| L.1.17 | Mini-Rosco will be designed such that the same number of motors and servos are equal to that of the original Rosco. | Are the number of motors and servos equal | Inspection | 2 |  |
| L.1.22 | Mini-Rosco’s form factor will be completed by the date of May 15, 2020 | Is Mini-Rosco’s form factor finalized | Test | 6 |  |
| L.1.23 | Mini-Rosco shall be disassembled and reassembled in 20 minutes or less. | Time the disassembly and reassembly of Mini-Rosco, including motors, servos, the 3DoT, and the PCB | Demonstration | 5 |  |
| L.1.24 | Mini-Rosco shall be designed such that there are no dangling wires. | Brush a finger over the top of Mini-Rosco’s chassis to verify that it doesn’t get caught | Inspection | 4 |  |
| L.1.25 | Mini-Rosco shall incorporate the 3DoT v9.05 or later series of boards. | Look at the back of the 3DoT board | Inspection | 3 |  |

### Detailed Success Criteria

* + Mini-Rosco must be within the range of 70%-80% smaller than Rosco
  + The 3DoT fits onto the appropriate spot on Mini-Rosco’s chassis
  + Number of motors and servos used should equal that of the original Rosco
  + No further adjustments should be made to Mini-Rosco’s form factor for the Spring 2020 semester
  + Mini-Rosco should be assembled and disassembled within 20min
  + No dangling wires
  + Correct version of the 3DoT

### Test Environment

This test case will take place and will be executed in Giann’s Living room within his apartment.

### Assumptions and Preconditions

* + Mini-Rosco is disassembled prior to this test case
  + Mini-Rosco has all 3D printed parts screwed / fixed in place

### Procedure Steps

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step Number** | **Step Description** | **Pass Criteria** | **Recorded Data** | **Requirement(s) Tested** | **Test Type** | **Test Method** |
| 1 | Begin with Mini-Rosco disassembled including motors, servos, 3DoT, and PCB | \*Timer for 20 min begins\* |  |  |  |  |
| 2 | Attach motors and servos | 2 motors and 2 servos should be attached |  | L.1.17 | Inspection | Inspection |
| 3 | Place 3DoT behind servo Tower after checking model and version number | v9.05 of the 3DoT will fit into the desired location |  | L.1.25, L.1.15 | Test | Inspection, Test |
| 4 | Run finger lightly along the top of the chassis | No wires catch on finger |  | L.1.24 | Inspection | Inspection |
| 5 | Disassemble Mini-Rosco, including motors, servos, 3DoT, and PCB | The end of this step happens before the 20 min timer reaches its end |  | L.1.23 | Test | Demonstration |
| 6 | Show Solidworks files | Mini-Rosco is in its final form factor that is 70%-80% smaller than Rosco |  | L.1.22, L.1.14 | Test | Test |

## TC-03: Mini-Rosco Mission Operation

### Detailed Description

In this test case Mini-Rosco will execute its mission in its entirety. First, Mini-Rosco will travel to a waypoint coordinate while detecting and avoiding obstacles in automated drive mode. Once Mini-Rosco arrives at the inputted waypoint coordinate, it will be switched into remote control mode so that the user can position it in front of an object to follow. Mini-Rosco will then be switched back into automated mode and follow the desired object.

### Test Case Traceability and Pass/Fail Matrix

This matrix shall show all requirements that are being tested in this test case. The Pass/Fail Column is populated after the Test Case has been run via the Procedure Steps.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement Number** | **Requirement Text** | **V Success Criteria** | **V Method**  (Analysis, Demonstration, Inspection, Test?) | **Procedure Step(s) where Requirement is tested** | **Pass/Fail?** |
| L.1.1 | Mini-Rosco shall operate for a maximum of 30 minutes. It should be noted that this 30 minute timeframe begins only after the GPS PCB has acquired a signal. This may take an average of 10 minutes. | Record how long each part of the process takes to complete. Sum of all three recordings is < 30min | Demonstration | 7 |  |
| L.1.5 | Mini-Rosco shall be controlled for the duration of the mission through the Arxterra control panel. Mini-Rosco is operated from our video booth. | See if commands sent from Arxterra control panel trigger appropriate response | Demonstration | 2, 6 |  |
| L.1.6 | Mini-Rosco will be able to accept user input through a waypoint for a specific destination to travel to. | User can enter destination information and sees waypoint on control panel | Test | 2 |  |
| L.1.7 | Mini Rosco shall be able to navigate to specific locations based on GPS coordinates within an accuracy of a minimum of 4 ft. | If received waypoints causes robot to reach specified destination | Demonstration | 4 |  |
| L.1.8 | Mini-Rosco shall recognize another robot within a 4 feet radius using an ultrasonic sensor after arriving at the specified destination.  The other robot that is being followed is 1 foot in height. | Verify if ultrasonic can detect within 4 feet in front of it | Test | 6 |  |
| L.1.9 | Mini-Rosco shall maintain a following distance of 1 to 2 feet after locating another robot. The other robot’s speed is within the range of 1 to 3 miles per hour. | Use ruler and hold it near Mini-Rosco while following | Demonstration | 6 |  |
| L.1.10 | Shall have autonomous navigation mode to travel to destination | Robot will arrive at the destination | Demonstration | 4 |  |
| L.1.11 | Mini-Rosco shall avoid obstacles up to 70cm away while in autonomous mode. | Robot will avoid obstacles in autonomous mode | Test | 3 |  |
| L.1.12 | Mini-Rosco shall execute its mission in three stages. The first stage will be the Locating stage seen in requirement L.1.7. The second stage will be the Acquisition stage see in requirement L.1.8. The third stage will be the Following stage seen in requirement L.1.9. | Stages are executed in order | Demonstration | 2, 3, 4, 6 |  |
| L.1.13 | Mini-Rosco shall employ a custom PCB to extend the functions of the 3DoT by allowing control over where Mini-Rosco travels via a set of coordinates | Is the board fabricated and does the GPS work | Test | 4 |  |
| L.1.19 | Mini-Rosco shall provide a live stream video feed via the Arxterra control panel throughout the duration of its mission. | Can a video feed be seen | Demonstration | 8 |  |
| L.1.21 | Mini-Rosco shall contain a custom 3DoT shield incorporating interface electronics between the 3DoT board and sensors and/or actuators unique to the robot. | Is a 3DoT present and does it interface with the ultrasonic sensor on Mini-Rosco. | Demonstration | 6 |  |

### Detailed Success Criteria

* + Mini-Rosco’s mission duration does not exceed 30 minutes
  + Mini-Rosco successfully executes commands issued from the Arxterra Control Panel
  + User has the ability to enter waypoints through the Arxterra Control Panel
  + Mini-Rosco travels to and arrives at desired waypoint
  + Can detect another robot in front of it within a 4 foot range
  + Mini-Rosco will remain 1 to 2 feet away when following
  + Mini-Rosco will autonomously travel to destination and autonomously avoid obstacles
  + Mini-Rosco’s mission will consist of three stages
  + Mini-Rosco uses a custom PCB for at least one of the stages in its mission
  + Mini-Rosco will provide a live video feed throughout the mission
  + 3DoT controls Ultrasonic sensor

### Test Environment

This test case will take place and will be executed in Giann’s Living room within his apartment.

### Assumptions and Preconditions

* + Mini-Rosco will be fully assembled and initialized.
  + Mini-Rosco will have an ultrasonic sensor mounted in the designated position
  + “Automated Mode” will be left in the “ON” default position to start the mission.
  + Mini-Rosco will be navigating across a room in Giann’s apartment.
  + Mini-Rosco will be in community mode and connected through the Arxterra Control Panel.
  + Mini-Rosco will have acquired a GPS signal prior to beginning its mission
  + Other “robots” are not available and as such a piece of carboard or a textbook will be used in its place.
  + Objects Mini-Rosco must avoid will not be moving objects.
  + Objects Mini-Rosco must follow will be in motion.

### Procedure Steps

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step Number** | **Step Description** | **Pass Criteria** | **Recorded Data** | **Requirement(s) Tested** | **Test Type** | **Test Method** |
| 1 | Ensure that Mini-Rosco is initialized (automated mode on), assembled, and connected to control panel |  |  |  |  |  |
| 2 | Enter a waypoint on the Arxterra Control Panel | Waypoint is visible on the Control Panel and Mini-Rosco begins to move to waypoint without any more user input |  | L.1.5, L.1.6, L.1.12 | Test | Demonstration, Test |
| 3 | Cardboard or book is held in Mini-Rosco’s path. Object is not moving | Mini-Rosco navigates around the object |  | L.1.11, L.1.12 | Test | Test, Demonstration |
| 4 | Mini-Rosco is within 4 feet of waypoint and can be considered at destination | Mini-Rosco is within four feet of entered waypoint |  | L.1.7, L.1.10, L.1.12, L.1.13 | Test | Demonstration, Test |
| 5 | Mini-Rosco is switched to RC mode and positioned in front of object it is to follow | Mini-Rosco is successfully manually positioned in front of object to follow |  |  |  |  |
| 6 | Mini-Rosco is switched to following mode with automated on | Mini-Rosco will begin to follow a moving object within a distance of 1ft to 2ft |  | L.1.5 L.1.8, L.1.9, L.1.12, L.1.21 | Test | Demonstration, Test |
| 7 | Record time at this point in the mission | Time should not exceed 30 min |  | L.1.1 | Test | Demonstration |
| 8 | View live stream at the end of the mission | Mini-Rosco has provided a live stream throughout the course of the mission |  | L.1.19 | Demonstration | Demonstration |

## TC-04: Mini-Rosco Logistical Elements

### Detailed Description

In this test case, Mini-Rosco itself is not present or needed. First, a cost report demonstrating line by line items purchased do not exceed budget will be presented. Then 3D printing costs will be analyzed to determine if too many files were used. Finally, any batteries present will be stored safely.

### Test Case Traceability and Pass/Fail Matrix

This matrix shall show all requirements that are being tested in this test case. The Pass/Fail Column is populated after the Test Case has been run via the Procedure Steps.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement Number** | **Requirement Text** | **V Success Criteria** | **V Method**  (Analysis, Demonstration, Inspection, Test?) | **Procedure Step(s) where Requirement is tested** | **Pass/Fail?** |
| L.1.16 | Mini-Rosco’s expenditures shall not exceed $500. | Show a cost resource report  (See Appendix for Cost Breakdown) | Inspection | 1 |  |
| L.1.28 | Manufacturability of 3D printed robots shall utilize as few files as possible when 3D Printing the Robot and its parts. | Ensure that there are not more than 10 SolidWorks files when 3D Printing. | Inspection | 1 |  |
| L.1.29 | All Lithium (Li-ion, Li-polymer) batteries shall be purchased with and stored, when not in use, in a fire and explosion proof battery bag. | Ensure that battery is stored in correct bag when not in use. | Inspection | 2 |  |

### Detailed Success Criteria

* + Mini-Rosco must not exceed the total allotted budget
  + Mini-Rosco must have been 3D printed efficiently
  + Battery used to power 3DoT must be stored safely

### Test Environment

This test case will take place and will be executed in Giann’s Living room within his apartment.

### Assumptions and Preconditions

* Mini-Rosco will not be present while completing this test case
* A computer sharing its screen over zoom will be used
* Documents will be shown over Zoom to prove cost constraints

### Procedure Steps

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step Number** | **Step Description** | **Pass Criteria** | **Recorded Data** | **Requirement(s) Tested** | **Test Type** | **Test Method** |
| 1 | Show cost breakdown spreadsheet | Total cost will not exceed budget and 3D print cost  total will be under $60  (See Appendix for Cost Breakdown) |  | L.1.16, L.1.28 | Inspection | Inspection |
| 2 | Show bag in which battery is stored in | Battery stored in safe bag |  | L.1.29 | Inspection | Inspection |

# Appendices

This section will contain any addition documentation needed to verify/validate requirements. For example, if a project has a cost constraint requirement, include the cost breakdown spreadsheet below as a subsection and reference the appendix subsection in the related Test Step in the Test Procedure. If another group needs to verify something by hand via calculation, include the calculations as a subsection below and reference the appendix subsection in the related Test Step in the Test Procedure.

## [Cost Structure Breakdown](https://docs.google.com/spreadsheets/d/1gRW-f5E5iHtfcufZo57qlCJMpQFvgarNjTIKy8gjOdI/edit?usp=sharing) : TC-04