System Test Plan

emStart - Emulator for Satellite Transceiver (communications) and Radio Telescope

Team members: Matthew Selph, Matthew Gasper, Matthew Grabasch, TJ Scherer, Ivan Borra

Version/Author	Date
1.0 All	10/17/2021
1.1 All	10/19/2021
1.2 All	12/6/2021

Table of Contents

Table of Contents **2**

Introduction **3**
- Purpose 3
- Objectives 3

Functional Scope **3**

Overall Strategy and Approach **3**
- Testing Strategy 3
- System Testing Entrance Criteria 3
- Testing Types 4
  - Usability Testing 4
  - Functional Testing 5
- Suspension Criteria and Resumption Requirements 6
  - Suspension Criteria 6
  - Resumption Requirements 6

Execution Plan **6**

Traceability Matrix & Defect Tracking **9**
- Traceability Matrix 9
- Defect Severity Definitions 11

Environment **11**

Assumptions **11**

Risks and Contingencies **12**

1. Introduction

1.1. Purpose

This document is a test plan for emStart System Testing, produced by the System Testing team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

1.2. Objectives

Meets the requirements, specifications and the Business rules.
Supports the intended business functions and achieves the required standards.
Satisfies the Entrance Criteria for User Acceptance Testing.
Follows any FCC Communications Standards or Restrictions

2. Functional Scope

The modules in the scope of testing for the emStart system testing are mentioned in the following areas: the System Requirement Specification document and section 3.1 of this document.

3. Overall Strategy and Approach

3.1. Testing Strategy

emStart System Testing will include testing of all functionalities that are in scope (Refer Functional Scope Section) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, work flows, functionality access, testing of internal & external interfaces.

3.2. System Testing Entrance Criteria

In order for the full system to be tested the robotic arm must be connected to the central computer and the main program must be loaded in order to begin synchronization. Once this is completed the rotator program/ground station program can be loaded and prepared to sync up with the earth system. On tangent to all of this the RF system can be turned on at any time which only requires the Hack RF to be connected to a transmitting computer while the RTL be connected to the receiving Raspberry Pi 4 ground station and set to receive on the correct frequency.

3.3. Testing Types

3.3.1. Usability Testing

User interface attributes, cosmetic presentation and content will be tested for accuracy and general usability. The goal of Usability Testing is to ensure that the User Interface is comfortable to use and provides the user with consistent and appropriate access and navigation through the functions of the application (e.g., access keys, consistent tab order, readable fonts etc.)

Documentation	Description
System Requirement Specification, 3.1.1	Diagram showing current azimuth angle of antenna.
System Requirement Specification, 3.1.2	Diagram showing current altitude angle of antenna.
System Requirement Specification, 3.1.3	Button allows the user to stop movement of the antenna
System Requirement Specification, 3.1.4	Button that allows manual control of the antenna for testing.
System Requirement Specification, 3.2.1	The Earth System shall consist of a computer connected to a 6-DOF robotic arm via a USB cable.
System Requirement Specification, 3.2.2	The Ground Station shall consist of a computer connected to the antenna microcontroller via a USB, which is used to control the 2-DOF antenna.
System Requirement Specification, 3.3.1	The Ground Station shall control a 2-DOF antenna.
System Requirement Specification, 3.3.2	The antenna shall be utilized using a microcontroller connected to a computer via USB.
System Requirement Specification, 3.3.5	The “Earth” will be fully controlled by a computer, working without reference to the other systems via a 3-DOF robotic arm connected to a computer via USB.
System Requirement Specification, 3.4.2	The Ground Base shall use the astronomical program to aid in tracking the signal by converting the data into movements for the 2-DOF Antenna.

System Requirement Specification, 4.3.3.5	The system shall transmit at a frequency of 910Mhz in order to emulate that of an actual received frequency of a ground station.
System Requirement Specification, 4.3.3.6	The system shall receive a frequency of 910Mhz in order to emulate that of the actual frequency transmitted from a satellite.
System Requirement Specification, 3.3.1.1.	The user interface shall display the current position of the system in real time.
System Requirement Specification, 3.3.1.2.	The emulation shall use a configuration file to determine how to obtain data.
System Requirement Specification, 3.3.1.3.	The emulation shall allow the user to load a variety of configurations without resetting the system.
System Requirement Specification, 3.3.1.4.	The emulation shall allow the user to load a CSV file containing altitude and azimuth coordinates associated with a time.
System Requirement Specification, 3.3.1.5.	The emulation shall accept commands which allow the user to play, pause, rewind, and reset the emulation time.
System Requirement Specification, 3.3.1.6.	The emulation shall synchronize the current time with the ground station using a bluetooth connection.

3.3.2. Functional Testing

The objective of this test is to ensure that each element of the component meets the functional requirements of the business as outlined in the:

- Business / Functional Requirements
- Business rules or conditions
- Other functional documents produced during the course of the project i.e. resolution to issues/change requests/feedback

System Requirement Specification, 3.2.3	The Ground Station shall work independently from the Earth System, meaning that the antenna must compensate for the movement of the “Earth
System Requirement Specification, 3.2.4	The System in Space shall work using a computer connected via USB to a SDR and GPSDO with a fixed antenna, sending the frequency towards the Ground Base.

System Requirement Specification, 3.3.3	The microcontroller will be fed the astrological data, which then can be converted into movements for the antenna. The system in space will only be sending signals, which will then be received by the antenna.
System Requirement Specification, 3.3.4	The software-defined radio will be used to communicate analog signals to the Ground Station. The Earth system shall use the 6 DOF arm to move the “Earth” in different positions, thus causing the Ground Base to be utilized to keep the antenna aligned with the space object.
System Requirement Specification, 3.4.1	The Ground Base shall track the analog signal that is being transmitted by the System In Space.

3.4. Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

3.4.1. Suspension Criteria

Testing will be suspended if the incidents found will not allow further testing of the system/application under-test. If testing is halted, and changes are made to the hardware, software or database, it is up to the Testing Manager to determine whether the test plan will be re-executed or part of the plan will be re-executed.
If from the outputs of the GUI, it is found that proper signal integrity is not found as defined by the software defined radio signal, testing will be halted to remediate the issue. If any piece of hardware does not function as specified in the requirements, testing will also be halted. Finally if any piece of the User Interface does not display or react as listed in the requirements, testing must be halted and ensure that proper data display is achieved.

3.4.2. Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been retested successfully.

4. Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their inter dependency.

Test Case Identifier	Input	Expected Behavior
TC01	Enter the following into the windows terminal in the location of the “Earth” program: python emStart.py -g	Successful launch of the web interface, displaying an empty plot alongside some data input fields, showing current azimuth and altitude
TC02	Click the button to stop movement of the antenna	The robotic arm should stop moving
TC03	Use the button allowing manual control of the earth from the earth computer	The earth robotic are should react to the control call and move to the specified location
TC04	Run the program again	A graph of the current status of alignment will appear indicating tracing of the signal
TC05	Run the earth system though a test movement	The system shall be a 2 DOF robotic arm connected to a computer which is independently controlled
TC06	Run a ground station test	The system shall be a 2 DOF rotator which has an antenna mounted to it, which is independently controlled by a RPI4
TC07	Feed the ground station data from astropy to be run	The ground station will go to the required location as specified by astropy
TC08	Run the program and open the link displayed in the terminal window	The browser should display a plot with one pink and one yellow line, which correspond to the altitude and azimuth angles.

TC09	Run the program	The program should automatically load the [DEFAULT] configuration defined in “config.ini”.
TC10	Run the program and type “load [CONFIG]” where [CONFIG] is the name of the user configuration	The program should load the new data for the specified configuration.
TC11	Run the program and type “load OVERRIDE”	The program should load custom data values contained in “data.csv”.
TC12	Run the program and type “play” After 5 seconds, type “pause” Type “rewind”, type “play” After 2 seconds, type “pause” Type “reset”	The emulation should progress for 5 seconds, then pause, then rewind for 2 seconds, then pause, then reset back to the beginning.
TC13	Run the appropriate programs on the Earth subsystem and the ground station	The ground station computer should receive aliveness pulses from the Earth, which contain the current emulation time.
TCRF01	When transmitting a 910MHz sinusoidal signal; at 1cm away from the receiver.	The signal can be picked up from the receiver and shown its graph with some gain greater than the noise ceiling.
TCRF02	When transmitting a 910MHz sinusoidal signal; at 10cm away from the receiver.	The signal can be picked up from the receiver and shown its graph with some gain greater than the noise ceiling.
TCRF03	When transmitting a 910MHz sinusoidal signal; at 100cm away from the receiver.	The signal can be picked up from the receiver and shown its graph with some gain greater than the noise ceiling.
TCRF04	When transmitting a 910MHz sinusoidal signal; at 1000cm away from the receiver.	The signal can be picked up from the receiver and shown its graph with some gain greater than the noise ceiling.
TCRF05	When transmitting a 910MHz sinusoidal signal; at 1000cm away from the receiver but facing the receiver 90 degrees to the right hand side from the perspective of the receiver..	The signal can be picked up from the receiver and shown its graph with some gain greater than the noise ceiling.
TCRF06	When transmitting a 910MHz sinusoidal signal; at 1000cm away from the receiver but facing the receiver 90	The signal can be picked up from the receiver and shown its graph

	degrees to the left hand side from the perspective of the receiver.	with some gain greater than the noise ceiling.
TCRF07	When transmitting a 910MHz signal; at 1000cm away from the receiver but facing the receiver 180 degrees from the perspective of the receiver.	The signal can be picked up from the receiver and shown its graph with some gain greater than the noise ceiling.
TCRF08	When transmitting a 910MHz sinusoidal signal; at 1000 cm away from the receiver have the receiver moving in a random fashion in all which way constantly.	The signal can be picked up from the receiver and shown its graph with some gain greater than the noise ceiling.
TCRE01	Assemble the system	The system shall be mounted on a wooden plate, with a 2 degree of freedom rotator which mounts the Patch Antenna
TCRE02	Run the 2 degree of freedom rotator in the full range of its motion	The patch antenna shall be attached to the RTL SDR with no issues of rotation in regards to the RPI 4 Mounting
TCRE03	Send the Arduino Nano azimuth and angle values from the RPI4 through UART	The rotator shall navigate to the required location as specified by the RPI4
TCRE04	Send the Arduino Nano azimuth and angle values using astropy from the RPI4 over UART	The rotator shall navigate to the astropy specified location

5. Traceability Matrix & Defect Tracking

5.1. Traceability Matrix

Requirement	Test Case Identifier	Pass / Fail	Defect Severity
3.1.1 3.1.2	TC01	Pass	Critical
3.1.3	TC02	Pass	Medium
3.1.4	TC03	Pass	Low
3.1.5	TC04	Pass	Medium
3.2.1 3.3.4 3.3.5 3.4.3	TC05	Pass	Medium
3.2.2 3.2.3 3.3.1 3.3.2	TC06	Pass	Critical
3.3.3 3.4.2	TC07	Pass	Critical
3.3.1.1.	TC08	Pass	Low
3.3.1.2.	TC09	Pass	Critical
3.3.1.3.	TC10	Pass	Medium
3.3.1.4.	TC11	Pass	Low
3.3.1.5.	TC12	Pass	Low
3.3.1.6.	TC13	Pass	Critical
4.2.3.1 4.2.3.4 4.2.3.5	TCRE01	Pass	Medium
4.2.3.2 4.2.3.3	TCRE02	Pass	Critical
4.2.3.6 4.2.3.7	TCRE03	Pass	Critical

4.2.3.8 4.2.3.9			
4.2.3.10 4.2.3.11 4.2.3.12	TCRE04	Pass	Medium
4.3.3.5 4.3.3.6	TCRF01	Pass	Medium
4.3.3.5 4.3.3.6	TCRF02	Pass	Medium
4.3.3.5 4.3.3.6	TCRF03	Pass	Medium
4.3.3.5 4.3.3.6	TCRF04	Pass	Medium
4.3.3.5 4.3.3.6	TCRF05	Pass	Medium
4.3.3.5 4.3.3.6	TCRF06	Pass	Medium
4.3.3.5 4.3.3.6	TCRF07	Pass	Medium
4.3.3.5 4.3.3.6	TCRF08	Pass	Medium

5.2. Defect Severity Definitions

Critical	The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows: ● Communication between the systems is disrupted. ● Communication between the systems is corrupted. ● Hardware malfunctioning.
Medium	The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows: ● Arm movement is incorrect. ● Plotly Data incorrectly displayed
Low	The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows: ● Repositioning of plotly graph on screens ● Text displayed incorrect

6. Environment

In order to conduct the testing the tester needs to have the following installed onto their computer:

- Python version 3.9.7 and up
- Arduino version 1.8.16 and up

7. Assumptions

Earth

- The astronomy data downloaded by astropy is accurate
- The astropy calculations are correct
- The latency of sending commands to myCobot is constant

Ground Station

- The antenna mount will never rotate more than 360 degrees
- The antenna mount will never rotate more than 90 degrees vertically
- The received radio signals will be noisy

System in Space

- The transmitted radio signals are weak enough to be contained in the lab

8. Risks and Contingencies

Risk #	Risk	Impact	Contingency Plan
1	Signal given too much power.	High	Keep possible max strength low.
2	Incompatible Compiler	High	Testing can only be done through python compliant IDEs.
3	Incorrect data given to Ground Base	High	Add redundancy to double-check the data given.