**System Test Plan**

**For**

**RF Direction Detection**

Abigail Butka

Cassandra Harrison

Sofia Mvokany

Kyle Reagan

Sabrina Yepez

|  |  |
| --- | --- |
| Version/Author | Date |
| 2.1 All | 3/10/2021 |
| 2.2.1 Sabrina Yepez | 4/2/2021 |
| 2.2 All | 4/6/2021 |
| 2.3 All | 4/20/2021 |

**Table of Contents**

[**1.** **Introduction** 3](#_Toc66278257)

[**1.1** **Purpose** 3](#_Toc66278258)

[**1.2** **Objectives** 3](#_Toc66278259)

[**2.** **Functional Scope** 3](#_Toc66278260)

[**3.** **Overall Strategy and Approach** 3](#_Toc66278261)

[**3.1** **Testing Strategy** 3](#_Toc66278262)

[**3.3** **Testing Types** 4](#_Toc66278263)

[**3.3.1** **Usability Testing** 4](#_Toc66278264)

[**3.3.2** **Functional Testing** 6](#_Toc66278265)

[**4.** **Execution Plan** 7](#_Toc66278266)

[**4.1** **Execution Plan** 7](#_Toc66278267)

[**5.** **Traceability Matrix & Defect Tracking** 8](#_Toc66278268)

[**5.1** **Traceability Matrix** 8](#_Toc66278269)

[**6.** **Environment** 11](#_Toc66278270)

[**7.** **Assumptions** 11](#_Toc66278271)

# **1.** **Introduction**

## **1.1** **Purpose**

This document is a test plan for the RF Direction Detection System. System testing will be broken down into individual component testing and system integration testing. Components to be tested are the rotating platform, stepping motor, Yagi antenna, and software-defined radio. Once all components pass their respective tests, the system components will be integrated and tested once more.

## **1.2** **Objectives**

This section will describe the plan for testing the RF Direction Detection System. The motor that is controlled by the Arduino code and end stop switch will be tested for rotation, speed, and on command rotation pauses. The S11 parameters and antenna characterization of the Yagi antenna are tested to ensure it meets the 33cm frequency band and directionality requirements. The software-defined radio will be tested through software Airspy and Mission Planner and will output magnitude and phase of the received signal using Simulink. To simulate a drone to be tested we will be using a Pixhawk in varying positions. For best results, testing will have to be done outdoors to receive GPS signals.

# **2.** **Functional Scope**

The functional scope of the RF Direction Detection System determines that the system shall function in the 33-cm frequency band and determine the direction of arrival from the AirSpy software. Using the data collected from the RTL-SDR, the system will output the magnitude and phase angle of the received signal using Simulink. Included in the scope are the Stepping motor, the Yagi antenna, and the RTL-SDR. The out-scope aspects of the system are the rotating platform that will attach the stepping motor and the Yagi antenna. The testing for the in-scope aspects are mentioned in the document in the following path:

* Testing Strategy
* System Testing Entrance Criteria
* Testing Types
* Suspension Criteria and Resumption Requirements

# **3.** **Overall Strategy and Approach**

## **3.1** **Testing Strategy**

The RF Direction Detection 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, and testing of internal & external interfaces.

**3.2** **System Testing Entrance Criteria**

In order to start system testing, certain requirements must be met for testing readiness. The readiness can be classified into:

* Availability of the test environment supporting necessary hardware, software, settings, and tools for the purpose of test execution. Such as the stepping motor, Yagi antenna, SDR, Mission Planner, Simulink and Airspy.
* Testers are trained and the necessary resources are available. Testers should have basic knowledge of radio frequencies.
* Requirements should be clearly defined and approved. All requirements can be found in the SRS of the system.
* Test Design and documentation are ready.

## **3.3** **Testing Types**

### **3.3.1** **Usability Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Test Description | Inputs/Action | Expected Output | Test Result | Test Comments |
| 1 | User changes rotation speed | 1. Open Arduino Source Code 2. Go to line 12 and change step delay value to the desired value. 3. Compile and Upload code to Arduino Board 4. Plugin 12 V supply to CNC shield | The antenna should change to the speed the user desired. | Pass | The value can be changed to any value as long as it is a positive integer or decimal value. |
| 2 | User changes angle output | 1. Open Arduino code 2. Go to the “steps” variable (default should be 17.778) 3. Change by a factor of 10 (177.778) | The angle output should be every 10 degrees instead of every 1 degree | Pass | Can be changed to output at any number of degrees, simply have to do the correct calculations for step to degree ratio |
| 3 | User presses end stop switch | 1. Open Arduino code 2. Plugin 12 V supply to CNC shield. 3. Upload code to the board 4. Press end stop switch | The stepping motor should stop rotating. | Pass | Stepping motor pauses rotation as long the end stop switch is pressed. |
| 4 | User lets go of the end stop switch | 1. Open Arduino code 2. Plugin 12 V supply to CNC shield. 3. Upload code to the board 4. Press end stop switch 5. Stop pressing end stop switch | The motor should resume rotation. | Pass | Stepping motor resumes rotation when the end stop switch is not pressed anymore. |
| 5 | The drone is simulated via one telemetry radio configured with Pixhawk and Mission Planner | 1. Connect telemetry radio to specified Pixhawk port using wired cable 2. Connect Pixhawk to computer 3. After opening Mission Planner, connect the Pixhawk configuration | The flight parameters in Mission Planner should update each second. If the Pixhawk setup is moved, the parameters will reflect the numerical changes as well as the visual flight pattern. | Pass | Mission Planner was able to show when GPS was connected. |
| 6 | The telemetry radios communicate with each other | 1. Connect telemetry radio 1 to Pixhawk and telemetry radio 2 to laptop. 2. Observe blinking LEDs: green blinking: seeking telemetry radio connection; green steady: connection established; red blinking: data transferred. 3. Connect telemetry radio 1 to telemetry radio 2 through Mission planner. | Both telemetry radios should blink red LEDs. The  the flight pattern of the Pixhawk is visible on both laptops (the one with the Pixhawk connected and the one with telemetry radio 2 connected). Changes are visible in real time. Airspy displays a frequency-hopping data spectrum. | Pass | The telemetry radios successfully connected and communicated. Mission planner was able to display flight parameters in real time. Airspy displays data spectrum. |
| 7 | Antenna functions in the 33-cm range | 1. VNA signal analyzer is calibrated in the 800 MHz to 1 GHz range 2. Antenna is connected to VNA 3. S11 measurements of antenna are taken from the VNA | S11 spectrum from the VNA should display a band pass in the 902-928 MHz frequency range with a peak at 915 MHz | Pass |  |
| 8 | System Characteristic Impedance | 1. VNA is set to 50Ω in the 33-cm band 2. Antenna is connected to a 50Ω coaxial cable 3. Smith Chart of S11 is obtained | S11 Smith Chat shows a matched impedance at 915 MHz | Pass | Verified using the 3D anechoic chamber results |

### **3.3.2** **Functional Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Test Description/Requirement | Inputs/Action | Expected Output | Test Result | Test Comments |
| 9 | Rotate Antenna 180 degrees | 1. Plug in a 12V power source. 2. Connect Arduino Board to PC and upload source code. | The antenna attached to the stepping motor should rotate 180 degrees clockwise then 180 degrees counterclockwise | Pass | The antenna rotates as expected. |
| 11 | Arduino Serial Port Displays angle of rotation | 1. Plug in a 12V power source. 2. Connect Arduino Board to PC and upload source code. 3. Open Arduino Serial Port | The serial port should show the angle the antenna is at continuously updating | Pass | The angle is printed by default for every 10 degrees. Can be changed by user (see usability test 2) |
| 12 | Telemetry radios connect and transmit data to each other | 1. Connect both radios to power sources via supplied cables | Both radios should have a solid green light and a blinking red light when data is being transmitted. The band shown in the SDR should pulsate when data is being transmitted. | Pass | Unsure if the amount of data communicated by each radio is symmetrical. |
| 13 | The SDR will read and display signals coming from the antenna | 1. Connect the antenna to the SDR 2. Connect the SDR to the computer 3. Open SDR software to the desired frequency (915MHz) | At 915 MHz, the SDR software will show a detected signal | Pass | At times we will receive signals most times we get frequency hops |
| 14 | Single Antenna Directionality | 1. Calibrate anechoic chamber to the 800 MHz to 1 GZ range 2. Set up the antenna in an anechoic chamber 3. Collect gain measurements of antenna in both the theta and phi directions | The Phi and Theta Log Mag polar measurements should from a figure 8 with two dips 180 degrees apart | Pass | The Yagi antenna shows directionality in the desired frequency range |
| 16 | The Pixhawk will establish a connection between the telemetry radios through Mission Planner | 1. Connect telemetry radio 1 to Pixhawk and telemetry radio 2 to laptop with Mission planner via USB 2. Identify the USB port used to connect the telemetry radio 2 to laptop 3. Select the port and 57600 and click on “Connect” | No error message from Mission Planner.  On Airspy, the received signal should be distinct in the 33-cm frequency band. | Pass | After establishing connections between the telemetry radios, the received signal is identifiable on the Airspy display |
| 17 | Perform test of modified telemetry radios with Diamond antenna | 1. Connect telemetry radio 1 to Pixhawk, and plug it into a laptop 1 via USB. Connect telemetry radio 2 to laptop 1 via USB. Set laptop 1 on one end of the room 2. Connect SDR with Diamond antenna to laptop 2 and place laptop on opposite end of the room 3. Open Simulink module and run 4. Maneuver Diamond antenna in a sweeping pattern, from pointing the antenna directly at the telemetry units, to pointing perpendicular in both directions. | Expected output would be for there to be a distinct spike in gain when the Diamond antenna is pointed at the telemetry unit Vs. pointed perpendicularly. | Pass |  |
| 18 | Perform test of final antenna choice and modified telemetry radios | 1. Same steps as test case 17, from a-c 2. Maneuver final antenna choice in a sweeping pattern, from pointing directly at the telemetry units to pointing perpendicular, in both directions. | Expected output would be a distinct spike in gain when the final antenna choice is pointed at the telemetry units, Vs. Pointed perpendicularly | Pass |  |
| 19 | Perform test of final antenna choice with stepper motor | 1. Hook up stepper motor as required (details provided once stepper motor is completed and integrated) 2. Same steps as test case 17, a-d, but with Yagi | Expected output would be a distinct spike in gain when the final antenna choice is pointed at the telemetry units, Vs. Pointed perpendicularly | Pass |  |
| 20 | Perform test of final antenna with stepper motor and MATLAB-Arduino communications code | 1. Hook up stepper motor apparatus and Yagi antenna to laptop 2. Set up Telemetry Radios as described in test case 17, a-b 3. Run MATLAB code CS491\_RTLSDR\_Read.m 4. Load Arduino code and begin running | Expected output would be for the stepper motor to sweep left to right 2 times, stopping ever 10 degrees for a measurement.  The output would be for the Yagi antenna to point in the direction of the highest gain, the direction of the antennas | In-Progress |  |

# **4.** **Execution Plan**

## **4.1** **Execution Plan**

The project is using an agile approach, with monthly iterations. At the end of each month, the requirements identified for that iteration will be delivered to the team and will be tested. Exploratory testing will play a large part of the testing as the team has never used this type of tool and will be learning as they go. Tests for planned functionality will be created and added to the Usability and Functionality testing sections of this document as we get iterations of the product.

# **5.** **Traceability Matrix & Defect Tracking**

## **5.1** **Traceability Matrix**

List of requirements and corresponding test cases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement No. | Requirement Description | Test Cases | Status | Pass/Fail | Comments |
| 3.1.1.1 | The system shall be able to detect signals operating at 915 MHz | 7 | Tested/Done | Pass |  |
| 3.1.1.2 | The telemetry radios shall communicate with each other using Mission Planner | 12 | Tested/Done | Pass |  |
| 3.1.1.3 | The Antenna shall have a peek gain at 915 MHz | 13 | Tested/Done | Pass |  |
| 3.1.2.1 | The stepping motor shall be able to rotate 180° clockwise | 2 | Tested/Done | Pass |  |
| 3.1.2.2 | The stepping motor shall be able to rotate 180° counterclockwise | 9 | Tested/Done | Pass |  |
| 3.1.2.3 | The stepping motor shall be able to rotate at any numerical speed the user inputs | 9 | Tested/Done | Pass |  |
| 3.1.2.4 | The stepping motor shall be able to stop when the end stop switch is pressed and held | 3 | Tested/Done | Pass |  |
| 3.1.2.5 | The stepping motor shall be able to rotate the attached antenna | 9 | Tested/Done | Pass |  |
| 3.2.1.1 | The user shall be able to connect the Pixhawk interface to Mission Planner via a connection though the computer’s ports | 9 | Tested/Done | Pass |  |
| 3.2.1.2 | The user shall be able to specify the types of connection for the Pixhawk device in Mission Planner | 16 | Tested/Done | Pass |  |
| 3.2.1.3 | The user shall be able to simulate a GPS based flight path in Mission Planner | 9 | Tested/Done | Pass |  |
| 3.2.1.4 | The system shall allow the user to control the rotation and direction of the stepper motor | 4 | Tested/Done | Pass |  |
| 3.2.2.7 | The loop antenna shall constantly be rotating 180° through the stepping motor which is powered by Arduino | 10 | Tested/Done | Pass |  |
| 3.2.2.9 | The loop antenna shall be matched to a 50Ω coaxial cable | 7 | Tested/Done | Pass |  |
| 3.2.4.1 | The system shall detect signals in the 33-cm RF band | 7 | Tested/Done | Pass |  |
| 3.2.4.2 | The system shall observe a minimal signal gain when the source is in front of the plane of the yagi antenna | 13 | Tested/Done | Pass |  |
| 3.2.4.3 | The system shall observe a maximum signal gain when the sources is perpendicular to the plane of the yagi antenna | 13 | Tested/Done | Pass |  |

# **6.** **Environment**

To execute these test cases the team will use a simulated drone, using a Pixhawk with two communicating telemetry units, to test the software-defined radio components of the system and antenna. To test the stepping motor, the team will use the actual product and Arduino to execute the test cases concerning the rotating mechanism of the system. For the GPS to work properly, testing will have to be done outside in a relatively open area.

# **7.** **Assumptions**

7.1 Assume there is no interfering signal.

7.2 Assume Cable from antenna to SDR poses no interference and/or loss.

7.3 Assume usage of Yagi antenna.

**8.** **Risks and Contingencies**

Due to continuous changing requirements there is a technical risk which may lead to failure of functionality and performance of the system. This may pose a budget risk as well since the budget is $1500 and the cost may exceed it due to improvement being continually added to the design or different testing approaches added to the system. There are no future contingencies, once the product is built few things will have to change.