**System Test Plan**

**For**

**RF Direction Detection**

Cassandra Harrison

Robert Kramer

Sofia Mvokany

Krishna Patel

Kyle Reagan

|  |  |
| --- | --- |
| Version/Author | Date |
| 1.0 Krishna Patel | 11/3/2020 |
| 1.1 All | 11/9/2020 |
| 1.2 Krishna Patel | 11/17/2020 |
| 1.3 Krishna Patel | 11/19/2020 |
| 1.4 Krishna Patel | 11/20/2020 |
| 1.5 All | 11/30/2020 |

**Table of Contents**

[**1. Introduction**](#_yoywm2c9jxv8)4

[1.1 Purpose](#_lkab4rfbi8iu) 4

[1.2 Objectives](#_mwmqwimmwuap) 4

[**2. Functional Scope**](#_wsnnygb5g0t6)4

[**3. Overall Strategy and Approach**](#_nhrw6ey22vv2)4

[3.1 Testing Strategy](#_sb7ofl4jdcue) 4

[3.3 Testing Types](#_ii58g9ro20hb) 5

[3.3.1 Usability Testing](#_1p5dpotzzuhk) 5

[3.3.2 Functional Testing](#_6d66u3vzrs48) 7

[**4. Execution Plan**](#_4tt3qwthk5jm)8

[4.1 Execution Plan](#_bkhhl3k1hdzr) 8

[**5. Traceability Matrix & Defect Tracking**](#_holdni9wnooa)8

[5.1 Traceability Matrix](#_ywy42pa7peye) 9

[**6. Environment**](#_9ro7oueepje3)12

[**7. Assumptions**](#_rb8b743z9wql)12

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

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

This document is a test plan for RF Direction Detection System Testing. Testing will be done on the system itself in different components. The rotating platform, stepping motor, loop antenna, and software-defined radio all will be first tested separately. Once all separate parts pass the tests the system will be assembled and will be tested for further analysis.

## **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 loop antenna and software-defined radio will be tested through software Airspy and Mission Planner. To simulate a drone to be tested we will be using a Pixhawk in varying positions. For best results of testing will have to be done outdoors to receive GPS signals.

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

The Modules in the scope of testing for the RF Direction Detection System Testing. In scope aspects of the system are the Stepping motor, the loop antenna, and the RTL-SDR. The out-scope aspects of the system are the rotating platform that will be attached to the stepping motor and the loop antenna. The testing for the in-scope aspects are mentioned in the document attached 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**

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, loop antenna, SDR, Mission Planner, 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 | a. Open Arduino Source Code  b. Go to line 12 and change stepDelay value to the desired value.  c. Compile and Upload code to Arduino Board  d. 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 | a. Open Arduino code  b. Go to the “steps” variable (default should be 17.778)  c. 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 | a. Open Arduino code  b. Plugin 12 V supply to CNC shield.  c. Upload code to the board  d. 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 | a. Open Arduino code  b. Plugin 12 V supply to CNC shield.  c. Upload code to the board  d. Press end stop switch  e. Stop pressing endstop switch | The motor should resume rotation. | Pass | Stepping motor resumes rotation when the endstop switch is not pressed anymore. |
| 5 | The drone is simulated via one telemetry radio configured with Pixhawk and Mission Planner | a. Connect telemetry radio to specified Pixhawk port using wired cable  b. Connect Pixhawk to computer  c. 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 | a. Connect telemetry radio 1 to Pixhawk and telemetry radio 2 to laptop.  b. Observe blinking LEDs: green blinking: seeking telemetry radio connection; green steady: connection established; red blinking: data transferred.  c. 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. |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Test Description/Requirement | Inputs/Action | Expected Output | Test Result | Test Comments |
| 1 | Rotate Antenna 180 degrees | a. Plug in a 12V power source.  b. 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. |
| 2 | Rotate Antenna at a speed of 1000 microseconds | a. Plug in a 12V power source.  b. Connect Arduino Board to PC and upload source code. | The antenna should rotate at a speed of 1000 microseconds | Pass | The antenna rotates at the speed expected. |
| 3 | Arduino Serial Port Displays angle of rotation | a. Plug in a 12V power source.  b. Connect Arduino Board to PC and upload source code.  c. 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) |
| 4 | Telemetry radios connect and transmit data to each other | a. Connect both radios to power sources via supplied cables  b. Open SDR to the desired frequency (915MHz) | 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. |
| 5 | The antenna will pick up signals at 915 MHz | a. Connect the antenna to the SDR  b. Connect the SDR to the computer | At 915 MHz, the SDR software will show a detected signal | Pass | At times we will receive signals most times we get frequency hops |
| 6 | The antenna will have a minimum gain when the signal direction is perpendicular to the plane of the loop | a. Set up the antenna for signal detection  b. Place the signal source in front of the loop and observe the signal  c. Place the signal source to the side of the loop and observe the signal | The observed signal from side placement will be significantly larger than the signal from the front placement | Pass | Receives minimum gain |

# **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 |
| 3.2.7 | The loop antenna shall constantly be rotating 180/360 degrees through the stepping motor which is powered by Arduino. | 1 | Tested/Done | Pass | Decided on only rotating 180 degrees, but is still tested for 360 degrees |
| 3.2.6 | The base station shall consist of a stepping motor with a loop antenna design attached to the stepping motor. | 1 | Tested/Done | Pass | Was attached using a 3D printed part |
| 4.2.3.1 | The stepping motor shall be able to rotate 180 degrees clockwise. | 1 | Tested/Done | Pass |  |
| 4.2.3.2 | The stepping motor shall be able to rotate 180 degrees counterclockwise. | 1 | Tested/Done | Pass |  |
| 4.2.3.3 | The stepping motor shall be able to rotate at any numerical speed the user inputs. | 1 | Tested/Done | Pass |  |
| 4.2.3.4 | The stepping motor shall be able to stop when the end stop switch is pressed and held | 1 | Tested/Done | Pass |  |
| 4.2.3.5 | The stepping motor shall be able to rotate the attached antenna. | 1 | Tested/Done | Pass |  |
| 4.1.2 | The user will manually tell the antenna when to rotate and when to stop. | 1 | Tested/Done | Pass |  |
| 3.4.1 | The system shall be able to detect signals operating at 915 MHz | 1 | Tested/Done | Pass |  |
| 3.4.2 | The system shall observe a minimal signal gain when the source is in front of the plane of the loop | 0 | Untested |  |  |
| 3.1.1 | The user shall be able to connect the Pixhawk interface to Mission Planner via a connection though the computer’s ports | 1 | Tested/Done | Pass |  |
| 3.1.2 | The user shall be able to specify the types of connection for the Pixhawk device in Mission Planner. | 1 | Tested/Done | Pass |  |
| 3.1.3 | The user shall be able to simulate a GPS based flight path in Mission Planner | 1 | Tested/Done | Pass |  |
| 3.1.6 | Rotation (Line 12): This line controls the degrees the stepping motor turns in any direction. | 1 | Tested/Done | Pass |  |
| 3.1.6 | Speed (Line 13): This line controls the speed at which the stepping turn by adding a delay between each pause. | 1 | Tested/Done | Pass |  |
| 3.4.1 | The system shall be able to detect signals operating at 915 MHz | 1 | Tested/Done | Pass |  |

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

To execute these test cases the team will use a Pixhawk simulator, to test the software-defined radio and antenna components of the system. 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**

There are no assumptions that need to be made when testing this product.

**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 as the budget is $1500 and the cost may overrun as there are different aspects being added last minute 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. There may be a MATLAB integration to the system but will cause little to no issues.