**System Requirements Specification**

**RF Direction Detection Project**

**CEC/EE 420 Fall 2020**

Cassandra Harrison

Robert Kramer

Sofia Mvokany

Krishna Patel

Kyle Reagan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date | Reason for Change | Names | Version |
| Initial Release | 9/10/20 | Initial Release | ALL | 1.0 |
| Update Sections | 9/22/20 | Updated all sections and Table of Contents | Krishna Patel | 1.1 |
| Hardware, Communication & Security Requirements | 9/24/20 | Hardware, Communication & Security Requirements | Krishna Patel | 1.2 |
| System Features | 9/25/20 | System Features | Robert Kramer | 1.3 |
| Communication Requirements | 9/25/20 | Communication Requirements | Sofia Mvokany | 1.4 |
| Software & Security Requirements | 9/25/20 | Software & Security Requirements | Cassandra Harrison | 1.5 |
| Assumptions | 9/25/20 | Assumptions | Kyle Reagan | 1.6 |
| Update Sections | 10/22/20 | Update to 1.1, 1.2, 1.3, 2.1, 2.2 | Robert Kramer | 1.7 |
| Update Sections | 10/27/20 | Update to 1.1, 2.7, 3 | Robert Kramer | 1.8 |
| Introduction & Software Interfaces | 11/17/20 | Update to 1.1, 1.3 and 3.4 | Krishna Patel | 1.9 |
| Product Overview | 11/18/2020 | Update to 2.3,2.4 & 2.8 | Krishna Patel | 2 |
| External Interface Requirements | 11/19/2020 | Update to section 3 and Formatting | Krishna Patel | 2.1 |
| Update Sections | 11/20/2020 | Update to all of section 4, 5 & 6 | Krishna Patel | 2.2 |
| Update Sections | 11/30/2020 | Updated all sections & Formatting | ALL | 2.3 |

[**1 Introduction**](#_sy8nnwxfddx3) **4**

[1.1 Purpose](#_xblgphdmmx1e) 4

[1.2 Intended Audience and Reading Suggestions](#_nx4q5qgdw90y) 4

[1.3 Product Scope](#_nx4q5qgdw90y) 4

[**2 Product Overview**](#_k9wiwsm6e0d2) **4**

[2.1 Product Perspective](#_nx4q5qgdw90y) 4

[2.2 Product Functions](#_nx4q5qgdw90y) 4

[2.3 User Classes and Characteristics](#_nx4q5qgdw90y) 4

[2.4 Use Cases](#_nx4q5qgdw90y) 5

[2.5 Operating Environment](#_nx4q5qgdw90y) 6

[2.6 Design and Implementation Constraints](#_nx4q5qgdw90y) 6

[2.7 Assumptions and Dependencies](#_clz50qcxjayr) 6

[2.8 Stakeholders](#_nx4q5qgdw90y) 6

[**3 External Interface Requirements**](#_12zne1gcnqfm) **6**

[3.1 User Interfaces (Software)](#_nx4q5qgdw90y) 7

[3.2 Hardware Interfaces](#_nx4q5qgdw90y) 8

[3.3 Software Interfaces](#_nx4q5qgdw90y) 9

[3.4 Communications Interfaces](#_nx4q5qgdw90y) 9

[**4 System Features**](#_qrg6g15w47t1) **9**

[4.1 RF Directional Antenna & Airspy](#_nx4q5qgdw90y) 10

[4.1.1 Description and Priority](#_ebndkbpeon4f) 10

[4.1.2 Stimulus/Response Sequences](#_ebndkbpeon4f) 10

[4.1.3 Functional Requirements](#_ebndkbpeon4f) 11

[4.2.1 Description and Priority](#_i129lwft4cx) 11

[4.2.2 Stimulus/Response Sequences](#_ilt73hjvw6ur) 11

[4.2.3 Functional Requirements](#_440tp0yuf6n1) 11

[**5 Other Nonfunctional Requirements**](#_j2zlfze7bchg) **11**

[5.1 Performance Requirements](#_nx4q5qgdw90y) 11

[5.2 Safety Requirements](#_nx4q5qgdw90y) 11

[5.3 Security Requirements](#_nx4q5qgdw90y) 12

[5.4 System Quality Attributes](#_nx4q5qgdw90y) 12

[5.5 Business Rules](#_nx4q5qgdw90y) 12

[**6 Appendix A: Glossary**](#_1nbehjs3lrvz) **12**

# **1 Introduction**

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

This document will describe the functions, user classes, external interface requirements, system features and nonfunctional requirements for the RF Direction of Arrival system. The product is designed to detect the radio frequency given off by commercial drones. It will detect telemetry signals and determine its direction and distance by analyzing the strength of the signal. The main reason for the development of this product is so that airports can detect and locate unauthorized drone use over and near airports.

## **1.2** **Intended Audience and Reading Suggestions**

The intended audience is those in the EECSE field that study signals as well as airport management and drone enthusiasts.

## **1.3** **Product Scope**

The scope of this product is to have a loop antenna capable of detecting a frequency of 915 MHz that can rotate to obtain the direction that the signal is coming from. The telemetry signal received by the antenna will be fed directly into a computer. It is also designed to determine the strength of the signal so that one can tell how close the source is. The direct application of this product is so that airports can use it to locate drones that are flying illegally near the airport.

# **2 Product Overview**

## **2.1** **Product Perspective**

This product will allow the user to obtain the direction of a radio frequency, specifically the signal given off by a drone. It will consist of a loop antenna, stepping motor, and software defined radio. The product will be able to rotate freely and find signals within the 915 MHz range.

## **2.2** **Product Functions**

The loop antenna will obtain the signal and the software defined radio will help make the signals readable to the computer. The stepping motor will rotate so that the antenna can find the direction that the signal is coming from. The motor will output the angle that it is at compared to the starting direction.

## **2.3** **User Classes and Characteristics**

2.3.1 Airport Users - Will be using the product at the airport to detect illegal drones in the area. If possible, should get the highest level of accuracy for detecting the signal because of its use.

2.3.2 Drone Users - Will be using the product for fun or as a hobby. Don’t need a high level of accuracy because they will know the general area where their drones will be.

2.3.3 FAA Users - Will not be using the product but need to see the history of how many drones and where.

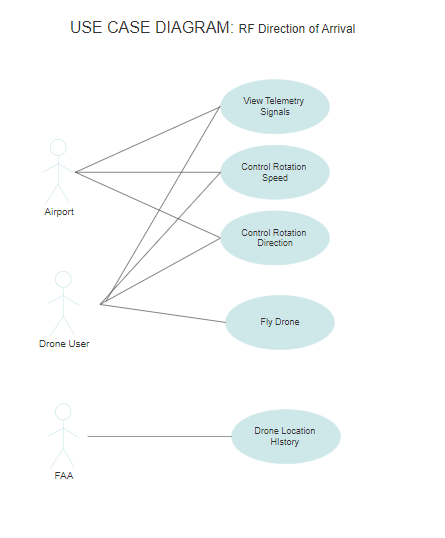
**2.4** **Use Cases**

Figure 1: Use Case Diagrams

2.4.1 View Telemetry Signals- User will be able to view the telemetry signals through Airspy and see the direction the signal is coming from in Mission Planner.

2.4.2 Control Rotation Speed - User will be able to change the speed of rotation for the loop antenna through the Arduino code provided.

2.4.3 Control Rotation Direction - User will be able to change the direction of rotation for the loop antenna through the Arduino code provided.

2.4.4 Fly Drone - The drone user will have control of the drone and can fly it and test the system if need be.

2.4.5 Drone Location History - User will have access to past history of drone locations.

2.4.6 View Rotation Angle - User will be able to see the angle the antenna is at while it is rotating

2.4.7 Stop Motor Rotation - User will be able to stop the motor from rotating by holding a button connected to the Arduino

## **2.5** **Operating Environment**

2.5.1 User Interface - Signal is processed on a computer that displays the angle, strength, direction, and frequency of the signal.

2.5.2 Data Collections - The loop antenna will collect the signal and send it to the central computer to be processed and displayed. The step motor runs independently and continuously updates the angle that’s displayed.

2.5.3 Data Transmission - No data is transmitted from the product.

## **2.6** **Design and Implementation Constraints**

The product will be limited to a small range, though is dependent on the strength of the signal being received. As a loop antenna is being used, it is bidirectional and thus the direction will be set to 2 directions. Possible constraint with the computer being used. If not powerful enough, may not be able to handle the data load from the antenna and may not have enough USB ports to plug in every appliance.

## **2.7** **Assumptions and Dependencies**

The detected signal shall come from a source emitting a continuous, constant frequency signal. The detected signal shall not move around at high speeds.

## **2.8** **Stakeholders**

Customer: Wants a way to detect where drones are coming from

FAA: Being able to detect illegally flying drones is useful for airports

Drone users: Be able to find lost drones or just keep track of drones from far away

# **3 External Interface Requirements**

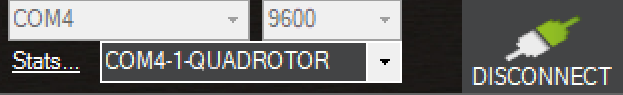
Section 3 covers information regarding the usage of the system’s interface, including both hardware and software interfaces. Sections 3.1 and 3.3 outline the software interfaces, including the user interface. Sections 3.2 and 3.4 outline the hardware interfaces and communication interfaces.

## **3.1** **User Interfaces (Software)**

The users shall interface with the RF Direction of Arrival system using Mission Planner, Airspy and Arduino. The following requirements describe how the user interface shall be displayed and utilized.

3.1.1 The user shall be able to connect the Pixhawk interface to Mission Planner via a connection though the computer’s ports

3.1.2 The user shall be able to specify the types of connection for the Pixhawk device in Mission Planner.

*Figure 2: Customization of the type of connected device after connection*

3.1.3. The user shall be able to simulate a GPS based flight path in Mission Planner

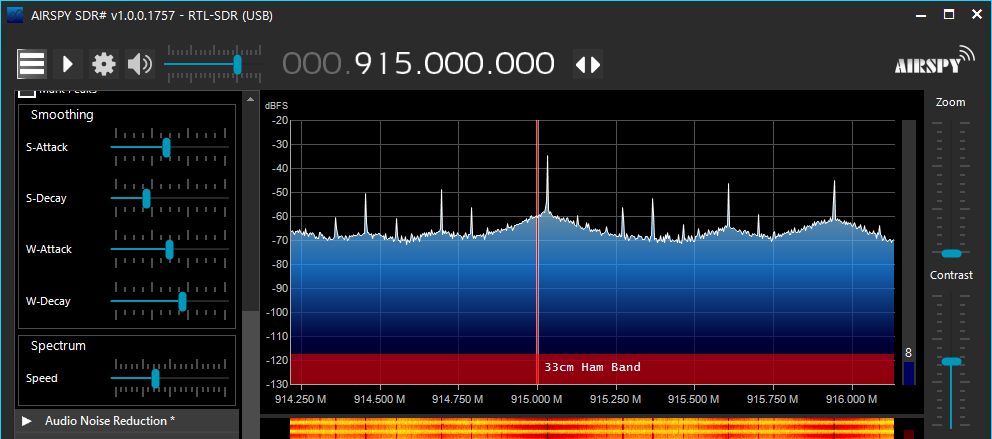
*Figure 3: Mission Planner/Pixhawk simulation display*

3.1.4 The user shall interface with Arduino to be able to control the rotation and direction of the stepping motor. The user shall change the following lines of code:

3.1.5 Rotation (Line 12): This line controls the degrees the stepping motor turns in any direction.

3.1.6 Speed (Line 13): This line controls the speed at which the stepping turn by adding a delay between each pause.

3.1.7 The user shall use software Airspy to visualize the spectrum of data transferred at the selected frequency.



*Figure 4: Screenshot of captured telemetry radios spectrum on Airspy.*

## **3.2** **Hardware Interfaces**

The RF Direction of Arrival system hardware consists of several components working in tandem to control the system. The RF DOA hardware consists of a Laptop acting as the base-station and a drone platform carrying testing and control the stepping motor.

3.2.1 The drone platform shall consist of the Pixhawk telemetry module and a GPS module.

3.2.2 The system shall operate as followed:

3.2.2.1. The PixHawk shall simulate the flight pattern of a drone.

3.2.2.2. Mission Planner shall track the Pixhawk’s location and flight parameters using the GPS.

3.2.2.3. The telemetry radios shall establish connections and data transfers between the PixHawk, laptop and Mission planner.

3.2.3 The testing hardware shall be coordinated by a single-board computer.

3.2.4 The testing hardware shall consist of the following:

3.2.4.1 Software defined radio

3.2.4.2 GPS module

3.2.5 The base station shall consist of a laptop running Mission Planner, Airspy and Arduino.

3.2.6 The base station shall consist of a stepping motor with a loop antenna design attached to the stepping motor.

3.2.7 The loop antenna shall constantly be rotating 180/360 degrees through the stepping motor which is powered by Arduino.

3.2.8 The stepping motor shall consist of the following:

3.2.8.1 CNC Shield Board

3.2.8.2 Arduino Uno Board

3.2.8.3 1.5m USB cable for Board

3.2.8.4 Nema 17 Stepper Motor 1.7A

3.2.8.5 Mechanical Switch Endstop

3.2.8.6 3 pin 70cm cable

3.2.8.7 DRV8825 Stepper Motor Driver

3.2.8.8 Aluminum heatsink

3.2.8.9 12-volt battery supply

3.2.9 The loop antenna shall consist of the following:

3.2.9.1 The loop antenna shall be approximately 32.7 mm in diameter, 10% of the target wavelength.

3.2.9.2 The antenna shall be made out of a conductive material, in this case, copper.

## **3.3** **Software Interfaces**

The RF Direction of Arrival system is composed of several software systems working in unison to receive data. The drone simulation platform consists of a Laptop and running Mission Planner. To configure the software defined radio receiver, the Laptop shall run Airspy. The rotating platform consists of a Laptop running Arduino.

3.3.1 The operating software shall be able to process communication between all components.

3.3.2 The system shall use the software Arduino to be able to run the stepping motor.

3.3.3 The system shall use the software Airspy to be able to read and display the signal received by the loop antenna.

3.3.4 The system shall use the software Mission Planner to be able to read and display the Pixhawk drone simulation.

3.3.5 The Laptop shall have Mission Planner, Airspy and Arduino installed.

3.3.6 The Laptop shall be running Windows 7 or above.

## **3.4** **Communications Interfaces**

3.4.1 The system shall be able to detect signals operating at 915 MHz

3.4.2 The system shall observe a minimal signal gain when the source is located in front of the plane of the loop.

3.4.3 The SDR is connected to the loop antenna that picks up the signal exuded by the Pixhawk. The telemetry radio 1 is connected to the Pixhawk while the telemetry radio 2 is connected via USB to the same laptop that holds the SDR. The user connects and starts the data exchange between both telemetry radios by connecting them using Mission planner. A GPS module is connected to the PixHawk and Mission planner is able to display the flight parameters of the Pixhawk as well as the geographical location.

# **4 System** **Features**

Section 4 details a more specific list as put forward in Section 2.2. This includes the functions associated with the RF Directional Antenna in 4.1, and the Stepping Motor in 4.2.

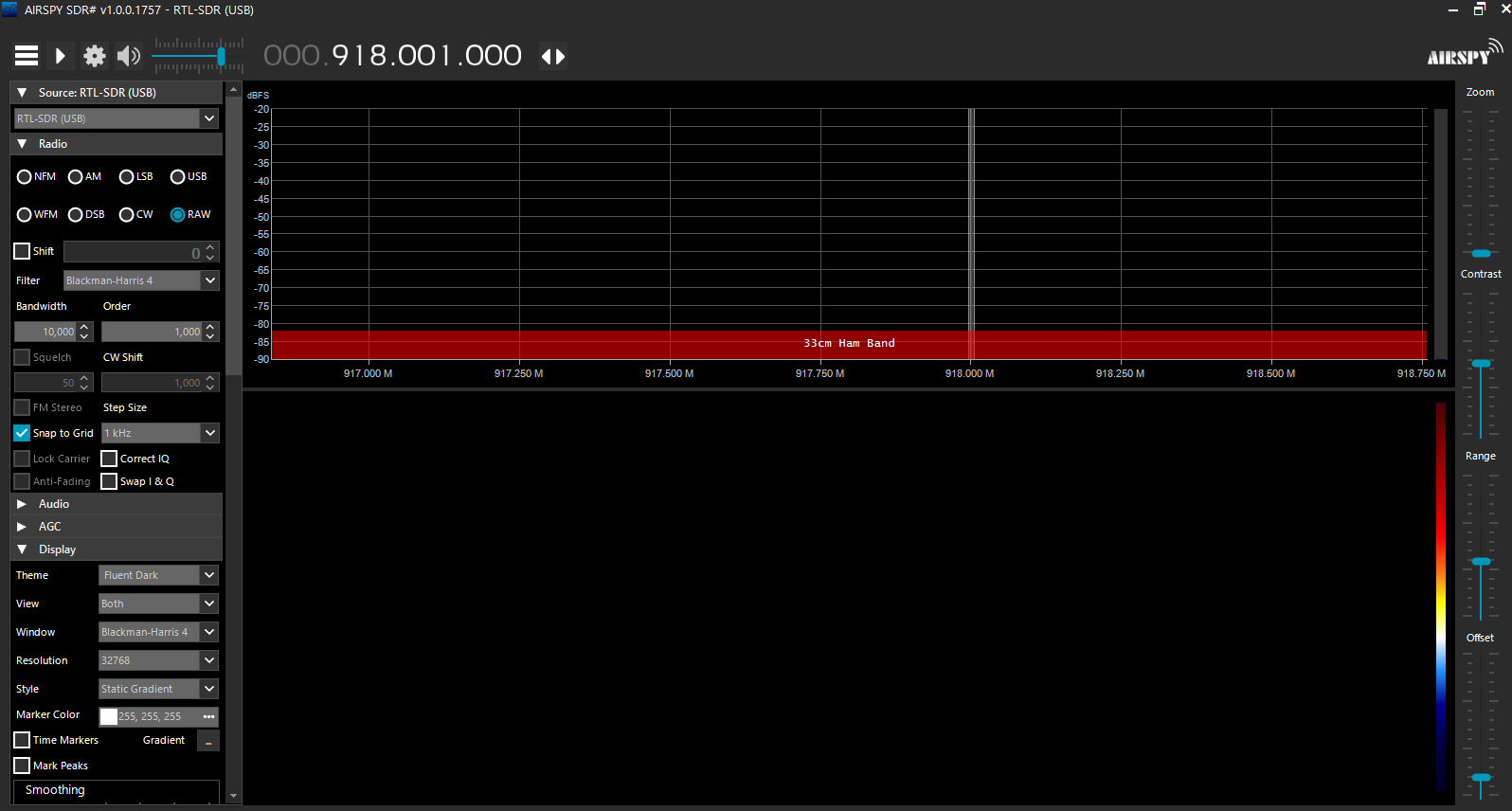
## **4.1** **RF Directional Antenna & Airspy**

### **4.1.1 Description and Priority**

The RF directional antenna will be tuned to detect a specific frequency given off by a drone. These frequencies will be displayed through a software called Airspy. Airspy is the software used by the SDR to observe and analyze the data exchanged by the system (Pixhawk, telemetry radios and GPS). Through Airspy, the user can analyze the data spectrum i.e. see the bandwidth, extract the audio file and observe the frequency and dB loss of the spectrum.

### **4.1.2 Stimulus/Response Sequences**

The Software Defined Radio dongle must be plugged into a computer with Airspy. The dongle must be attached to an antenna to be able to pick up signals at the various displayed frequencies. When opening Airspy after connecting the dongle, the user will have to make sure that the correct source is selected (Here RTL-SDR (USB)).



*Figure 5: Example frequency display when SDR is connected*

*.*

The user has the options to choose various radios such as Narrow-band FM (NFM) or Wide-band FM (WFM) as well as the desired frequency at the top. Once the investigated frequency is chosen by typing it in the frequency bar at the top of the interface (Here, 918 MHz), press the play button. The user should be able to see and hear the audio of data that is being transferred on the chosen frequency.

### **4.1.3 Functional Requirements**

4.1.3.1 The Antenna shall be able to rotate at least 180° and detect RF signals operating at 915 MHz

4.1.3.2 The user shall be able to connect the telemetry radios to each other using Mission planner.

4.1.3.3 The Airspy software shall display the spectrum of the data exchanged between the Pixhawk, GPS and telemetry radios.

**4.2** **Stepping Motor**

### **4.2.1 Description and Priority**

The system must be able to rotate so the loop antenna can pick up frequencies from the entire area. The antenna will rotate by attaching it to the stepping motor. The stepping motor gets its directions through an Arduino code that tells it to move a certain amount of steps in a certain direction along with controlling the speed of rotation.

### **4.2.2 Stimulus/Response Sequences**

The user will manually tell the stepping motor when to rotate and when to stop. The stepping motor will rotate 180 degrees before turning the other way for another 180 degrees. The user will also manually tell at what speed the stepping motor will turn by adding a step delay in the code.

### **4.2.3 Functional Requirements**

4.2.3.1 The stepping motor shall be able to rotate 180 degrees clockwise.

4.2.3.2 The stepping motor shall be able to rotate 180 degrees counterclockwise.

4.2.3.3 The stepping motor shall be able to rotate at any numerical speed the user inputs.

4.2.3.4 The stepping motor shall be able to stop when the end stop switch is pressed and held.

4.2.3.5 The stepping motor shall be able to rotate the attached antenna.

# **5 Other Nonfunctional Requirements**

Section 5 details the current known requirements and regulation associated with this product as of the Initial Release, Version 1.0.

## **5.1** **Performance Requirements**

There are no performance requirements for this system.

## **5.2** **Safety Requirements**

5.2.1 No voltage line shall exceed 32 volts.

5.2.2 All electrical connections shall be either concealed or wrapped up.

## **5.3** **Security Requirements**

As of right now, there are no applicable security requirements for this system.

## **5.4** **System Quality Attributes**

5.4.1 The Laptop shall display the drone location/direction.

5.4.2 Antenna should have a diameter of 10% of the operating wavelength. At 915 MHz, that would be approximately 33 mm.

## **5.5** **Business Rules**

5.5.1 If a drone user is using this system only the FAA and the airport shall be able to authorize drone testing within 5 miles of the airport.

# **6 Appendix A: Glossary**

CNC - Computerized Numerical Control

DOA -Direction of Arrival

EECSE - Electrical Engineering and Computer Science

FAA - Federal Aviation Administration

MHz - Mega Hertz

NFM - Narrow-band FM

RF- Radio Frequency

SDR - Software Defined Radio

WFM - Wide-band FM