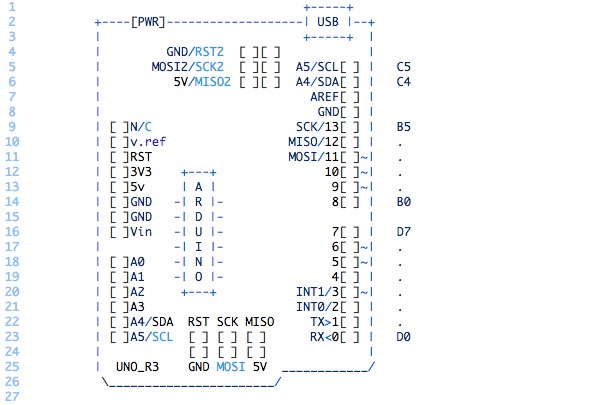
System Requirements Specification for

EGR101 Simulation System

Emily Connearney  
Luke Crump  
Vivian Dang  
Keely Mashburn  
Daniel Khalil



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date | Reason for Change | Initials | Version |
| Initial Release | 9/22/21 | Initial Release | ALL | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |

2

**Contents**

[1 Introduction 2](#_Toc83888446)

[1.1 Purpose 3](#_Toc83888447)

[1.2 Intended Audience and Reading Suggestions 3](#_Toc83888448)

[1.3 Product Scope 3](#_Toc83888449)

[1.4 References 3](#_Toc83888450)

[2 Overall Description 3](#_Toc83888451)

[2.1 Product Perspective 4](#_Toc83888452)

[2.2 Product Functions 4](#_Toc83888453)

[2.3 User Classes and Characteristics 5](#_Toc83888454)

[2.3.1 EGR101 Student: 5](#_Toc83888455)

[2.3.2 EGR101 Instructor: 5](#_Toc83888456)

[2.4 Operating Environment 5](#_Toc83888457)

[2.4.1 User Interface: 5](#_Toc83888458)

[2.4.2 Data Collection: 5](#_Toc83888459)

[2.4.3 Data Transmission: 5](#_Toc83888460)

[2.5 Design and Implementation Constraints 5](#_Toc83888461)

[2.6 User Documentation 5](#_Toc83888462)

[2.7 Assumptions and Dependencies 5](#_Toc83888463)

[2.8 User Interfaces (Software) 5](#_Toc83888464)

[2.9 Hardware Interfaces 5](#_Toc83888465)

[2.10 Software Interfaces 5](#_Toc83888466)

[2.11 Communications Interfaces 5](#_Toc83888467)

[3 System Features 5](#_Toc83888468)

[3.1 Wiring and Design Interface 6](#_Toc83888469)

[3.1.1 Description and Priority 6](#_Toc83888470)

[3.1.2 Stimulus/Response Sequences 6](#_Toc83888471)

[3.1.3 Functional Requirements 6](#_Toc83888472)

[3.1.3.1 The system shall allow the user to select a wire. 6](#_Toc83888473)

[3.1.3.2 The system shall allow the user to move the wire to connect two pin locations. 6](#_Toc83888474)

[3.2 Arduino IDE 6](#_Toc83888475)

[3.2.1 Description and Priority 6](#_Toc83888476)

[3.2.2 Stimulus/Response Sequences 6](#_Toc83888477)

[3.2.3 Functional Requirements 6](#_Toc83888478)

[3.3 PIGEONS Mission Data Viewer 7](#_Toc83888479)

[3.3.1 Description and Priority 7](#_Toc83888480)

[3.3.2 Stimulus/Response Sequences 7](#_Toc83888481)

[3.3.3 Functional Requirements 7](#_Toc83888482)

[4 Other Nonfunctional Requirements 10](#_Toc83888483)

[4.1 Performance Requirements 10](#_Toc83888484)

[4.2 Safety Requirements 11](#_Toc83888485)

[4.3 Security Requirements 11](#_Toc83888486)

[4.4 System Quality Attributes 11](#_Toc83888487)

[4.5 Business Rules 12](#_Toc83888488)

[5 Appendix A: Glossary 12](#_Toc83888489)

[6 Appendix B: Analysis Models 13](#_Toc83888490)

# Introduction

This Product is designed to supplement learning for remote students in the course EGR101. The course utilizes a $229.00 Parallax Boe-Bot Robot Kit to allow students to design and build a functionally autonomous robot. EGR101’s main project deliverables include grades based on performance in four Boe-Bot courses built to challenge students on forming solutions to: basic line following, line following corrected for noise, object avoidance, and resource management. Normally students are split into groups of three, with each group receiving a Boe-Bot kit, which includes its respective sensors, LEDs, and resistors.

Due to the recent pandemic, the role of the course has changed due to variability of student in-person attendance. The current solution to this problem was to make students pay $85 for their own kits and perform the required deliverables remotely.

## Purpose

This document intends to define all requirements and conditions associated with the EGR101-Simulation System. This document covers the product itself, its interaction with the user, and the requirement associated with bringing the system in compliance with the EGR101 course vision.

## Intended Audience and Reading Suggestions

The intended users of this product include students, and the instructor of the EGR101 course. This document contains information on product functionality, requirements, standards, and regulations given by the current EGR101 instructor. Section 2 describes the intent of the project. Section 3 describes the technical aspect most pertinent to the other software developers maintaining or enhancing the project. Section 4 and 5 describe the features and standards the project follows and should be of interest to instructors interested in understanding the scope of this tool.

Along with reading this document, the references outlined in section 1.5 are important to know for the proper operation and testing of the project.

## Product Scope

The scope of this application would be to reduce the cost of eventual replacement of Boe-Bot kits, allow for remote learning through testing electronic based solutions in a sandbox environment, and ease of grading said electronic based solutions. The proposed project would allow students to program Arduino sketches, design a virtual bot through adding components and wire connections, test their virtual bot on the 4 deliverable courses and provide a sandbox environment to improve understanding of basic circuitry and imperative programming. This product could be used in applications far beyond the scope of this course as a virtual electronics test environment could be in-valuable to autonomous vehicle testing

## References

[1] *Language reference*. Arduino Reference - Arduino Reference. (n.d.). Retrieved September 30, 2021, from <https://www.arduino.cc/reference/en/>.

# Overall Description

Section 2 covers purpose of the product in 2.1, the product functions in 2.2 associated user information in 2.3, and the operating environment in 2.4, Sections 2.5, 2.6 and 2.7 contain additional information on the implementation of the functions outlined in section 2.2

## Product Perspective

The EGR101 Simulation System will act as a sandbox to test Arduino code and wiring configurations from the Arduino to its individual components on the simulated bot. The application is controlled via the core Java application which will initialize a Unity application displaying 3-Dimensional Bot customization and simulation display. This lets the core application thread of execution and the Unity thread of execution run in parallel. These threads will communicate with each other over TCP.

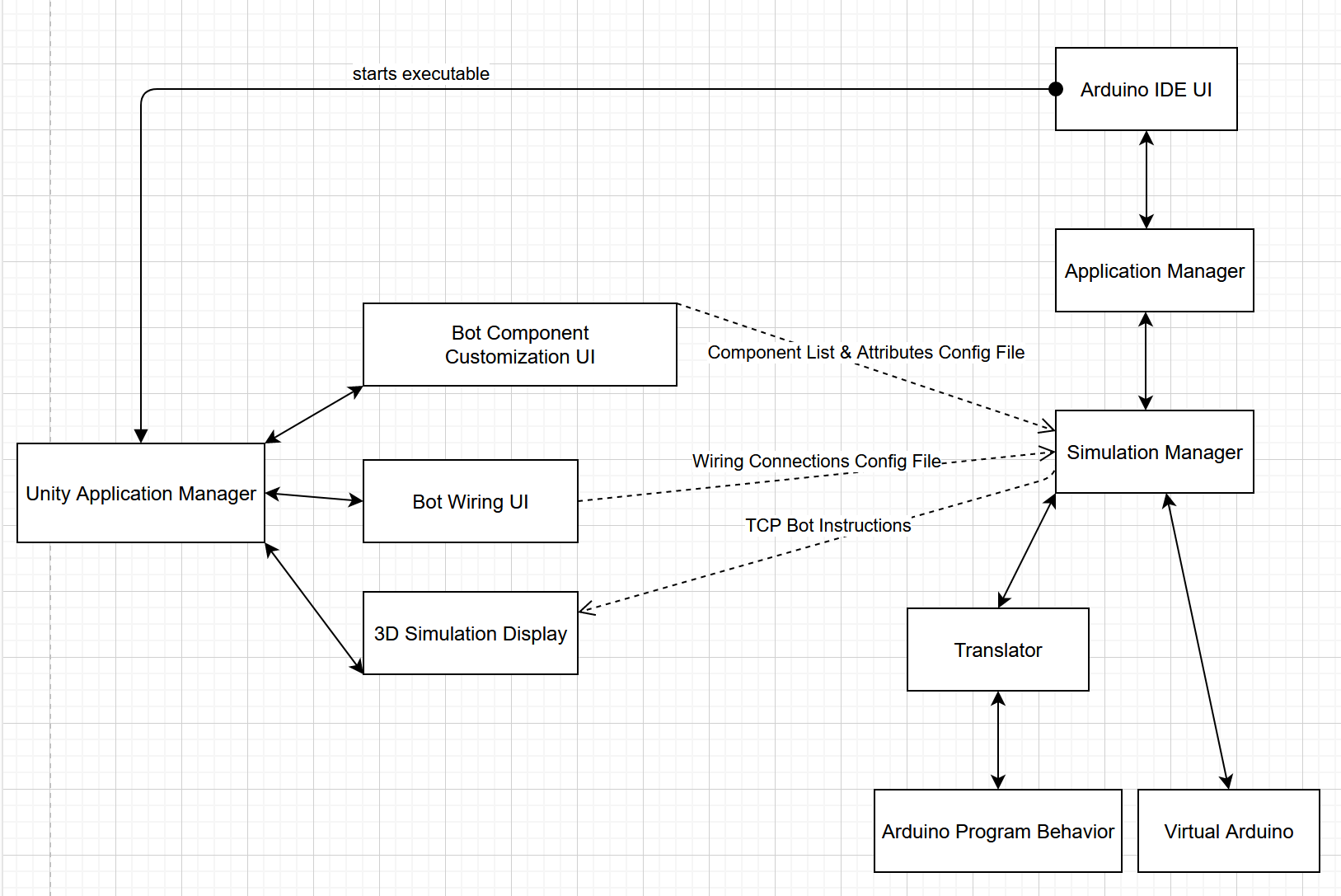


Figure 1: System Configuration Diagram

## Product Functions

2.2.1 Simulate an Arduino IDE to allow users to program virtual bots.

2.2.2 Emulate Arduino and allow the IDE to compile and change Arduino behavior at runtime.

2.2.3 Provide an Arduino API that will modify the emulated Arduino.

2.2.4 Allow components to be added to the bot configuration.

2.2.5 Allow for wired connections between components and Arduino to be specified.

2.2.6 Simulate component functionality based off wired connections.

2.2.7 Display 3-Dimensional autonomous bot course simulation.

2.2.8 Produce recording of 3-Dimensional course simulation.

2.2.9 Produce zipped configuration files for wired connections to components, component list, and Arduino sketches.

## User Classes and Characteristics

### Student

The student is the default user intended to interact with this system. The student is not expected to understand how the Arduino API functions, or how to properly wire Arduino components. Students will interact with the Arduino IDE to produce code, update the bot configurations through adding/removing components, and wiring the components to power, ground, and IO ports on the Arduino, students can also test their configured bots on the 4 deliverable courses. This application will be mainly focused on providing a user-friendly experience for these students.

### Instructor

The instructor is a secondary user who intends to interact with this system. This individual should have proficient knowledge in the systems this application is emulating. This user will require students to have configuration files exported and imported to help ease of grading. There will also be a system in which students must fill out a profile which will be injected into the configuration file which will give the instructor information about student name, student ID, and any other information needed for the identity of the student.

## Operating Environment

### User Interface:

### Data Collection:

### Data Transmission:

## Design and Implementation Constraints

1. Design Constraints:
2. Implementation Constraints:

## User Documentation

## Assumptions and Dependencies

## User Interfaces (Software)

## Hardware Interfaces

## Software Interfaces

## Communications Interfaces

# System Features

## Wiring and Design Interface

### Description and Priority

### Stimulus/Response Sequences

### Functional Requirements

### 3.1.3.1 The system shall allow the user to select a wire.

### 3.1.3.2 The system shall allow the user to move the wire to connect two pin locations.

3.1.3.3 The system shall record the pin locations that are connected by a wire in a file.

3.1.3.4 The system shall have a button to save the current wire configuration.

3.1.3.5 The system shall save the current configuration of wires when the “save” button is pressed.

3.1.3.6 The system shall have a button to exit the Wiring and Design Interface.

3.1.3.7 The system shall exit to the main view screen when the exit button is pressed.

3.1.3.8 The system shall have a button to add a resistor to the board.

3.1.3.9 The system shall be able to connect resistors to pins.

3.1.3.10 The system shall be able to connect resistors to wires.

## Arduino IDE

### Description and Priority

### Stimulus/Response Sequences

### Functional Requirements

3.2.3.1 The system shall have a File button that shows file operations when clicked on.

3.2.3.2 The system shall have a save button appear when the File button is clicked

3.2.3.3 The system shall save the Arduino script when the save button is clicked.

3.2.3.4 The system shall have an open button appear when the File Button is clicked.

3.2.3.5 The system shall open a new Arduino script when selected

3.2.3.6 The system shall have a “save configuration” button appear when the File button is clicked.

3.2.3.7 The system shall save a configuration file when the “save configuration” button is clicked.

3.2.3.8 The system shall produce a configuration file that contains the wiring setup and the Arduino code when the “save configuration” button is pressed.

## PIGEONS Mission Data Viewer

### Description and Priority

The Mission Data Viewer will be built as a standalone software to display current and past PIGEONS data acquired during a mission. The XBEE module will be the means to communicate from the PIGEONS payload system and the PIGEONS Mission Data Viewer. The user will be able to control options relating XBee connection and mission types (ILS/VOR, Both). The flight path from QGroundControl will be used to create a real-time map of the flight path of the vehicle and associated data collected. The map will utilize 3D shapes and color coding scheme defined to indicate the demodulated measurements and the signal strength at any given location.

### Stimulus/Response Sequences

When the UAV flies over a waypoint, the GPS location will be transmitted to the Mission Data Viewer. When the data packet is received, the ground station will parse the packet. The signal strength received will be displayed on the ground station with a color scheme. A cylinder of varying height will be used to denote the altitude the data point was taken. The user shall be able to click the GPS point to display the data values received.

### Functional Requirements

4.3.3.1 The PIGEONS Mission Viewer Start screen shall be displayed upon execution.

4.3.3.2 The PIGEONS Mission Viewer Start screen shall display the Track Live Mission button.

4.3.3.3 The PIGEONS Mission Viewer Start screen shall display the Replay Previous Mission button.

4.3.3.4 Pressing the Track Live Mission button shall display the Remote Vehicle Connection Settings view.

4.3.3.5 The Remote Vehicle Connection Settings view shall allow the ability to select the COM port the XBee is connected to.

4.3.3.6 The Remote Vehicle Connection Settings view shall allow the ability to select the Baud rate the XBee is using.

4.3.3.7 The Remote Vehicle Connection Settings view shall allow the ability to select the Data Bits the XBee is using.

4.3.3.8 The Remote Vehicle Connection Settings view shall allow the ability to select the parity settings the XBee is using.

4.3.3.9 The Remote Vehicle Connection Settings view shall allow the ability to select the Stop Bits settings the XBee is using.

4.3.3.10 The Remote Vehicle Connection Settings view shall allow the ability to select the Flow Control settings the XBee is using.

4.3.3.11 The Test Connection Button shall be displayed in red.

4.3.3.12 A label shall display errors when pressing the Test Connection button results in a failed connection.

4.3.3.13 The Test Connection Button shall change colors to green when a successful connection is established over the XBee module.

4.3.3.14 The Next button in the Remote Vehicle Connection Settings shall display the PIGEONS Mission Settings Dialog when pressed.

4.3.3.15 The Mission dropdown shall display the mission types available (ILS/VOR/Both).

4.3.3.16 The user shall enter numerical frequency values for the ILS frequency.

4.3.3.17 The user shall enter numerical frequency values for the VOR frequency.

4.3.3.18 The Next button in the PIGEONS Mission Settings shall display the PIGEONS Mission Plan Upload Dialog when pressed.

4.3.3.19 Pressing the browse button shall open a file dialog for selecting .plan files.

4.3.3.20 The Next button in the PIGEONS Mission Plan Upload shall display the PIGEONS Live Mission Settings Confirmation Dialog when pressed.

4.3.3.21 The Mission label shall display the selected mission type from 4.3.3.14.

4.3.3.22 The ILS Frequency shall display the entered ILS frequency from 4.3.3.15.

4.3.3.23 The VOR Frequency shall display the entered VOR frequency from 4.3.3.16.

4.3.3.24 The Mission Plan label shall display the selected Mission Plan from 4.3.3.18.

4.3.3.25 The start button shall display the PIGEONS Live Mission View when pressed.

4.3.3.26 The top bar in the Live Mission View shall display the ILS Frequency from 4.3.3.15.

4.3.3.27 The top bar in the Live Mission View shall display the Link status as Connected in green font when connected.

4.3.3.28 The top bar in the Live Mission View shall display the Link status as disconnected in red font when disconnected from XBee module.

4.3.3.29 The map view shall display the path the drone flew with color coded cylinders.

4.3.3.30 The signal strength shall be characterized as red, yellow, green cylinders (red = unacceptable, yellow = acceptable, green = acceptable).

4.3.3.31 Cylinder height in the map view shall be determined on UAV measured altitude.

4.3.3.32 Selecting a point shall display its recorded data.

4.3.3.32.a Recorded data shall be defined as Location, Altitude, Measurement Type, Signal Strength, and Within Range

4.3.3.32.b The Location label shall display the GPS coordinates of the points recorded.

4.3.3.32.c The Altitude label shall display the Altitude in meters of the points height.

4.3.3.32.d The Measurement Type shall display the current point measurement type (ILS/Vor)

4.3.3.32.e Signal Strength shall display the signal strength in dB.

4.3.3.32.f Within Range shall display if the point recorded meets calibration criteria.

4.3.3.33 The Mission Completed dialog shall appear denoting success/failure after UAV has landed.

4.3.3.34 The user shall be able to save Mission replay to the ground station computer.

4.3.3.35 The Replay Previous Mission button shall display the PIGEONS Mission Plan Replay Selection Dialog.

4.3.3.36 Pressing the browse button for Mission plan shall open a file dialog for selecting .plan files.

4.3.3.37 Pressing the Data Recording File browse button shall open a file dialog for selecting PIGEONS Data Record Files (.pdr) files.

4.3.3.38 The Next button in the PIGEONS Mission Plan Replay Selection dialog shall display the PIGEONS Recorded Mission view when pressed.

4.3.3.39 The Recorded Mission view shall display time warp buttons to step through mission time

4.3.3.39.a The Step Backwards button shall jump the display one data point backwards.

4.3.3.39.b The Pause button shall pause the mission from displaying anymore points.

4.3.3.39.c The Play button shall continue playback of the mission.

4.3.3.39.d The Step Forward button shall just the display one data point forward.

4.3.3.40 The onboard computer shall transmit data to the ground station within 3 seconds.

4.3.3.41 The ground station shall receive data from the onboard computer within 3 seconds.

# Other Nonfunctional Requirements

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

## Performance Requirements

5.1.1 Local data writing shall not be interrupted if connection is lost.

5.1.2 Data transmission shall be performed at a speed of at minimum 10 kB/s.

5.1.3 The maximum range of data transfer from drone to base station shall exceed 1 mile.

5.1.4 The drone shall have a maximum flight time exceeding 5 minutes.

5.1.5 Transmission delay from drone to ground station shall not exceed 10 seconds.

5.1.6 Delay of updating overlay shall not exceed 10 seconds.

5.1.7 The various overlay configurations should be evident and simple to understand for the user.

5.1.8 Changing between types of overlay shall take no more than 5 seconds.

5.1.9 The user shall be informed of any connection loss during flight.

5.1.10 The user shall be notified if no signals are being collected during flight.

5.1.11 The unit shall have onboard storage that exceeds 100 gb.

5.1.12 The operating software on the Raspberry Pi shall not exceed 16 gb.

## Safety Requirements

5.2.1 The system shall not be utilized in unapproved airspace unless directly approved by the FAA.

5.2.2 The drone shall not be flown within 5 miles of the airport without the airports consent.

5.2.3 The drone shall have a protocol to come back to the ground station if connection is lost.

5.2.4 All electrical connection shall be concealed.

5.2.5 No voltage line shall exceed 20 volts.

## Security Requirements

5.3.1 No other device shall be able to receive the transmission unless a transmission key is provided.

5.3.2 The unit shall not measure any signals other than ILS and VOR.

5.3.3 The operator of the ground station shall possess a drone license unless flying in a designated area.

5.3.4 The drone shall not capture any images of private property or individuals without consent.

## System Quality Attributes

5.4.1 The signals collected shall be within 2*σ*

5.4.2 The measured carrier frequency shall be within ±0.002% of the intended carrier frequency of the VOR transmitter as highlighted in ICAO Document 8071, Volume 1,

2.2.6.

5.4.3 Changing the polarization of the antenna shall have a variance of ±2.0° on VOR measurements as discussed in ICAO Document 8071, Volume 1, 2.2.34.

5.4.4 The system shall abide to all VOR bearing measurements as discussed in ICAO Document 8071, Volume 1, 2.2.7 and 2.2.8.

5.4.5 The VOR 9 960 Hz carrier shall have a modulation depth between 28% and 32% as discussed in ICAO Document 8071, Volume 1, 2.2.12.

5.4.6 The VOR 30 Hz carrier shall have a modulation depth between 28% and 32% as discussed in ICAO Document 8071, Volume 1, 2.2.12.

5.4.7 The DDM of the two ILS carriers shall be within ±2.0% when directly in line with the center line of the runway as discussed in Section 4.3.14 and 4.3.15 of ICAO Document 8071, Volume 1.

5.4.8 The average error in measurement of ILS location in the horizontal direction shall be within ±10.5 m as discussed in Section 4.3.26 to 4.3.28 of ICAO Document 8071, Volume 1.

5.4.9 The data collection unit shall not exceed 1 lb excluding the antenna.

5.4.10 The data collection unit shall be a standalone unit from the drone.

5.4.11 The data collection utilized without transmitting data if desired.

5.4.12 The ground station shall display the overlay on top of a terrain/satellite map.

5.4.13 The ground station software shall display the correct overlay selected by the user.

5.4.14 The system shall be able to be used during approved flight times by the FAA or respective airport.

## Business Rules

5.5.1 Only the FAA and the airport shall be able to authorize drone testing within 5 miles of the airport.

5.5.2 A flight test engineer should determine the pre-flight path needed to verify the transmitters as described in ICAO Document 8071, Volume 1.

# Appendix A: Glossary

DDM - Difference in depth of modulation. The percentage modulation depth of the larger signal minus the percentage modulation depth of the smaller signal, divided by 100.

ICAO - International Civil Aviation Authority

ILS - Instrument landing system

QGC - QGroundControl, software package controlling autonomous functions.

RPi - Raspberry Pi 3 model B

SDR - Software Defined Radio

UDP - User Datagram Protocol

VHF - Very High Frequency

VOR - VHF omnidirectional radio range

# Appendix B: Analysis Models

