ECE496 Final Report

Project Title: Bluetooth Low Energy (BLE) As Localization and Communication Solution for Drone Swarms

Project Number: 2022617

Supervisor Name: Lacra Pavel

Team Members:

Mohammad Ahmed

Jun Ho Sung

Adam Pietrewicz

Natalia Sydorenko

ECE496Y

Voluntary Document Release Consent Form

To all ECE496 students:

To better help future students, we would like to provide examples that are drawn from excerpts of past student reports. The examples will be used to illustrate general communication principles as well as how the document guidelines can be applied to a variety of categories of design projects (e.g. electronics, computer, software, networking, research).

Any material chosen for the examples will be altered so that all names are removed. In addition, where possible, much of the technical details will also be removed so that the structure or presentation style are highlighted rather than the original technical content. These examples will be made available to students on the course website, and in general may be accessible by the public. The original reports will not be released but will be accessible only to the course instructors and administrative staff.

Participation is completely voluntary and students may refuse to participate or may withdraw their permission at any time. Reports will only be used with the signed consent of all team members. Participating will have no influence on the grading of your work and there is no penalty for not taking part.

If your group agrees to take part, please have all members sign the bottom of this form.

Consent Statement

We verify that we have read the above letter and are giving permission for the ECE496 course coordinator to use our reports as outlined above.

Team #: 617 Project Title: Bluetooth Low Energy (BLE) As Localization and Communication Solution for Drone Swarms

Supervisor(s): Lacra Pavel Administrator: Milan Graovac

Name	Jun Ho Sung	Signature		Date:	4/6/2023
Name	Adam Pietrewicz	Signature	Adam Pietrewicz	Date:	4/6/2023
Name	Mohammad Ahmed	Signature	Mohammad	Date:	4/6/2023
Name	Natalia Sydorenko	Signature	Natalia Sydorenko	Date:	4/6/2023

Executive Summary (Author: A. Pietrewicz)

Drones are becoming a versatile and powerful tool to technological systems both outdoors and indoors. One main area of innovation and study is Indoor Positioning Systems, a system put in place in an indoor environment that calculates and tracks the position of an asset, or in this scope, a drone. There are many technologies available such as GPS and radar, however they are not plausible solutions for indoor drones because they use satellite signals to pinpoint where the drone is, and those signals are weak and can't penetrate through buildings. This highlights the need for alternative sensors or communication technology that can work indoors. Therefore, Bluetooth Low Energy (BLE) is an exciting and strong technology option for drone indoor positioning systems, since its main benefit and selling point is that it is energy efficient and accurate in locating signal origins up to the centimeter accuracy, where usually meter accuracy is standard and expected. Using BLE's Angle of Arrival (AOA) technology is versatile for drone systems because only an BLE tag and AOA antenna anchors placed around a room are necessary for location tracking, where the BLE tag module can be mounted onto any indoor commercial drone. This gives the drone BLE connectivity that advertises its location that can be triangulated, which is a main component of the software that must accompany the BLE AOA anchors. To build such a BLE drone localization system, it must contain 3 main subsystems which include: the drones with BLE broadcasting signal capability, BLE AOA antenna arrays to intercept the drone signals, and a system that contains the triangulation algorithm and control interface where the user can interface with their drone. This system design has been developed to be adaptable and easy to integrate into any custom or commercial drone application. After vigorously testing the design, the accuracy of BLE localization was determined to be feasible but still prone to error which highlights the challenges and limitations of indoor positioning systems for drones, where there are many signal obstructions that degrade accuracy. One main observation is that the higher the number of BLE anchors placed in the indoor environment, the better the accuracy received because it allows for more averaging of data that each pair of anchors detect, and BLE is capable of high scalability while also being energy efficient.

(Note: All sections were written with heavy editing after simple first drafts from all members, which results in the primary author for each section to not be a useful measure of contribution. Overall, each member contributed equally to the document)

Individual Contributions - Adam

- Assessment of hardware compatibility with intended flight control firmware code with the SEEED Sense chip
- Research on hardware parts necessary for the project (finding and evaluating alternatives)
- Worked on flashing SEED micro-controller with drone firmware, conducted research and alternatives
- Tested the SEED microcontroller’s Bluetooth capabilities
- Development of Bluetooth antenna software:
    - Polling from uBlox Bluetooth antennas/anchors
    - Conducted testing of uBlox Bluetooth AOA antenna boards to assess functionality
    - Implemented triangulation program of Bluetooth antenna data (angles)
        - Tested different types of data averaging algorithms and data cleaning to achieve higher localization accuracy
        - Integrations with drone control/localization GUI
- Final design testing (Appendix A)
    - Setting up antenna board configuration and stands
    - Collected cables/USB hub for connecting the boards together
    - Verifying, debugging, and testing accuracy of overall localization system

Individual Contributions - Jun Ho

- Project scoping
    - Initial development of project ideas
    - Shift of project goals when needed
- Parts selection & procurement for drones & BLE modules
- Original custom drone design & building of 2 drones
    - 3d printing of drone frame
    - Circuit design of drone parts
    - Soldering of drone parts parts
    - Tested drone design with custom motor-testing code
- Assessment of hardware compatibility with intended flight control firmware code with the SEEED Sense chip
- UI design & implementation
    - Implementation of UI components using PyQT5
    - Integrations with drone control & triangulation system
- Control system implementation
    - Geofencing
- Drone movement control
    - Tello SDK integration
- Parts of triangulation and averaging algorithm design, but not implementation
- Final design testing (Appendix A)
    - setting up board configuration
    - verifying accuracy of overall localization system

Individual Contributions - Natalia

- Control System Design
    - Initial ideas for location averaging/refinement
    - Research and attempt of designing proper control systems using transfer functions for drone movement and drift
        - Determined unfeasible due to lack of information about drone, DJI Tello ended up having movement commands, and drone commands unable to take in smaller than 20cm of movement
    - Pseudocode for intuitive drift control

Note: Numerous physical and serious mental health problems throughout the year hindered my contribution (can be verified through petitions not limited to interim demo and testing report, academic advisor Leanne Dawkins, Dr. Rachelle Guttman). Made contributions in writing the proposals, implementation plan, final presentation and final report.

Individual Contributions - Mohammad

- Selected a portion of parts needed to be purchased for drone design
- Familiarized myself with SEEED Sense chip by testing its bluetooth functionality
  - simple test program sending data demonstrated in demo early in the semester
- Assessed possibility of programming the SEEED Sense chip to act as a tag
  - Worked on trying to flash tag code onto the SEEED Sense chip to aid with localization
- Initial Triangulation Implementation and Algorithm
  - Researched into how triangulation works, what is required and needed
  - Implemented parts of the triangulation algorithm
- Triangulation Testing
  - setting up board configuration
  - recording measurements
  - verifying accuracy of triangulation code
- Final Design Testing (Appendix A)
  - setting up board configuration
  - verifying accuracy of overall localization system

1. Introduction

This report summarizes the motivation, design, implementation and testing: of Bluetooth Low Energy (BLE) As a Localization and Communication Solution for Drone Swarms as part of our final year design project course ECE496. The report concludes with our final findings and thoughts regarding our design and improvements that could be made in future work.

Background and Motivation: (Author: M. Ahmed)

Our team was initially interested in exploring the world of swarm drones, which refer to a group of Unmanned Aerial Vehicles (UAVs) that operate together to complete valuable tasks such as surveillance, search and rescue, and room mapping. We discovered that drones mainly rely on the Global Positioning System (GPS) or distance sensors such as RADAR and ultrasonic for localization and cellular/radio transceivers for communication. Specifically for indoor drone applications, GPS satellite signals have very low signal strength and accuracy. Therefore, alternative techniques such as ultrasonic beacons are used for indoor positioning systems, improving their accuracy from 1- 400 cm (using GPS) to 1- 100 cm [1]. However, we still believe there is constant room for further indoor drone positioning innovations, such as increasing localization accuracy, improving the reliability and robustness of drone-to-drone communication, and marginal costs. Therefore, our team investigated whether other technologies could achieve higher drone positioning accuracy than ultrasonic technology and eventually learned about Bluetooth Low Energy.

Bluetooth Low Energy Technology: (Author: A. Pietrewicz)

We decided to explore using BLE technology for a drone/swarm drone localization system because we recognized that it is more energy efficient and modern. Our research found that BLE provides two technologies: Angle of Arrival (AOA) and Angle of Departure (AOD). These technologies locate Bluetooth signal origins by calculating the angle at which signals are received by an array of antennas down to centimeter accuracy [2][3], which is competitively better than the accuracy range for GPS and ultrasonic solutions mentioned above. Therefore this sparked the initial motivation and goal for the project; to design and build drones that utilize BLE technology for localization and communication.

However, within the project lifecycle, the team faced a major challenge with designing and constructing a drone from scratch that would be customized to communicate and localize itself using Bluetooth. Specifically, we couldn't finalize a working drone firmware for our microcontroller of choice (due to its small size and weight): the Seeed Studio XIAO nRF52840 [4]. Therefore, the project scope was changed from building a custom drone with Bluetooth communication and localization to focusing on building a Bluetooth indoor positioning system for drones. Additionally, the need for building a swarm of drones was not deemed necessary for demonstrating the functionality of a Bluetooth drone localization system, since it would significantly increase our project material costs and introduce another area of complexity of drone control theory.

Project Goals and Requirements: (Author: M. Ahmed)

As mentioned above, the final project goal was adjusted due to the refinement and change of the project scope. The project goal is to develop a scalable drone infrastructure that utilizes BLE AOA technology for the accurate indoor localization of drones.

Functions:

- Should be able to broadcast commands to drones and receive information about each drone's heartbeat
    - Heartbeat: signal, drone ID number, acknowledgment of received commands
- The system should be able to detect out-of-bound commands
    - Reject commands that lead to drone motion outside of the signal perimeter
    - Flip a kill switch (land immediately) if a drone manages to leave the signal range
- Drones must not crash into each other
    - Drone can't take/be assigned a position in which another drone is already located (or any position in a 15 cm radius of that point)
- Drone must be able to complete the following commands: arm (ready to start), disarm, liftoff, land, left, right, forwards, backwards, up, down

Constraints:

- The system must have a latency no greater than 1s between the drone and GCS

- Should be able to determine the location of a drone within an accuracy of 20cm
- A drone must be a quadcopter equipped with a BLE module and a flight controller with embedded electronic speed controllers
- A drone must weigh less than 250g to qualify within the "micro-drone" class under Canadian law

Objectives:

- The system should be able to correct any drone's drift up to a 20 cm radius at any given time

2. Final Design

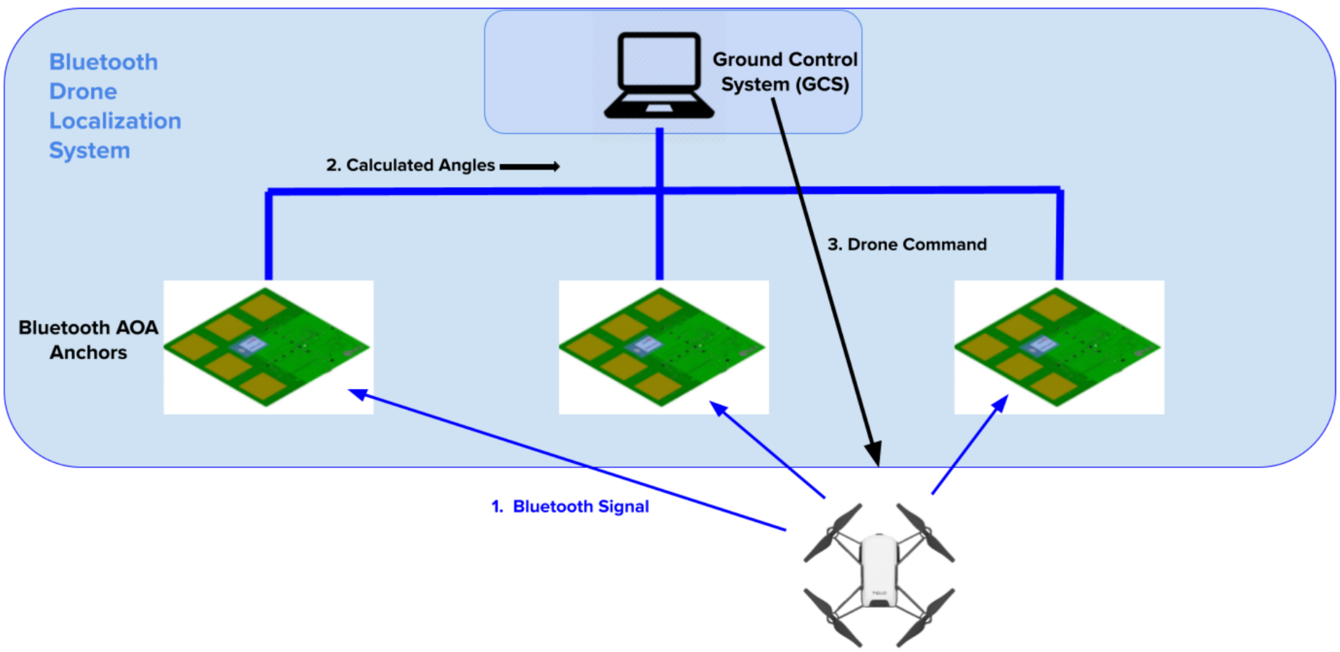


Figure 1: System Level Overview

System-level overview: (Author: A. Pietrewicz, M. Ahmed)

The drone BLE indoor positioning system we have pursued is represented by the system structure shown in Figure 1. This is a system-level overview where the design is divided into three subsystems: the Drone System (functioning drones, each with a BLE broadcasting module), a BLE AOA Array (used for detecting the drone signal's angle of arrival used to localize the drones), and a Ground Control Station (GCS) used for controlling the drone(s) and tracking/displaying their location information. Each subsystem is closely integrated with each other and all must run successfully in order to control and locate the drone in real time, to provide maximum location accuracy to safely control the drone in an indoor environment. The three subsystems are connected like a feedback loop, starting with the drone periodically sending out broadcast BLE signals which get detected/intercepted by the array of BLE AOA boards, which calculate the angle of the signal's arrival. Finally, the developed Ground Control System calculates the Cartesian Coordinates of the drone's location and uses it to properly allow a user to send flying commands and make sure the drone is safe to do so.

System block diagram: (Author: A. Pietrewicz, J. Sung)

Below is a detailed System Block diagram of the drone BLE localization system that the team has finalized and developed. The three subsystems (Drone System, BLE AOA Array and Ground Control Station) now highlight their specific modules providing all the key information necessary for building this system. It begins with mounting an easy-to-use and adaptable battery-powered uBlox BLE tag onto the drone (which can be placed on any commercial drone) and setting it to its maximum broadcast speed (for best accuracy). Next, those Bluetooth broadcast signals (more specifically called Eddystone signals) are intercepted by uBlox AOA anchors placed in the physical setup configuration described in the section below, connected through USB to a single laptop. These anchors communicate using Serial Communication to the Ground Control System, which is a Python program that polls from each USB port and sends its data to a Triangulation System, which is a program that does all the trigonometric calculations for computing the drone's current location. Next, the Triangulation System sends those coordinates continuously in a loop to the Control System, which includes the Python code for the user GUI that allows the user to click buttons on the window to control the drone while simultaneously displaying the drone's position in a plot and its battery life. For our project demonstration we use a DJI Tello drone because it is very user friendly and easy to control using Python functions, therefore the Control System is easily configurable for any user needs for their drone system.

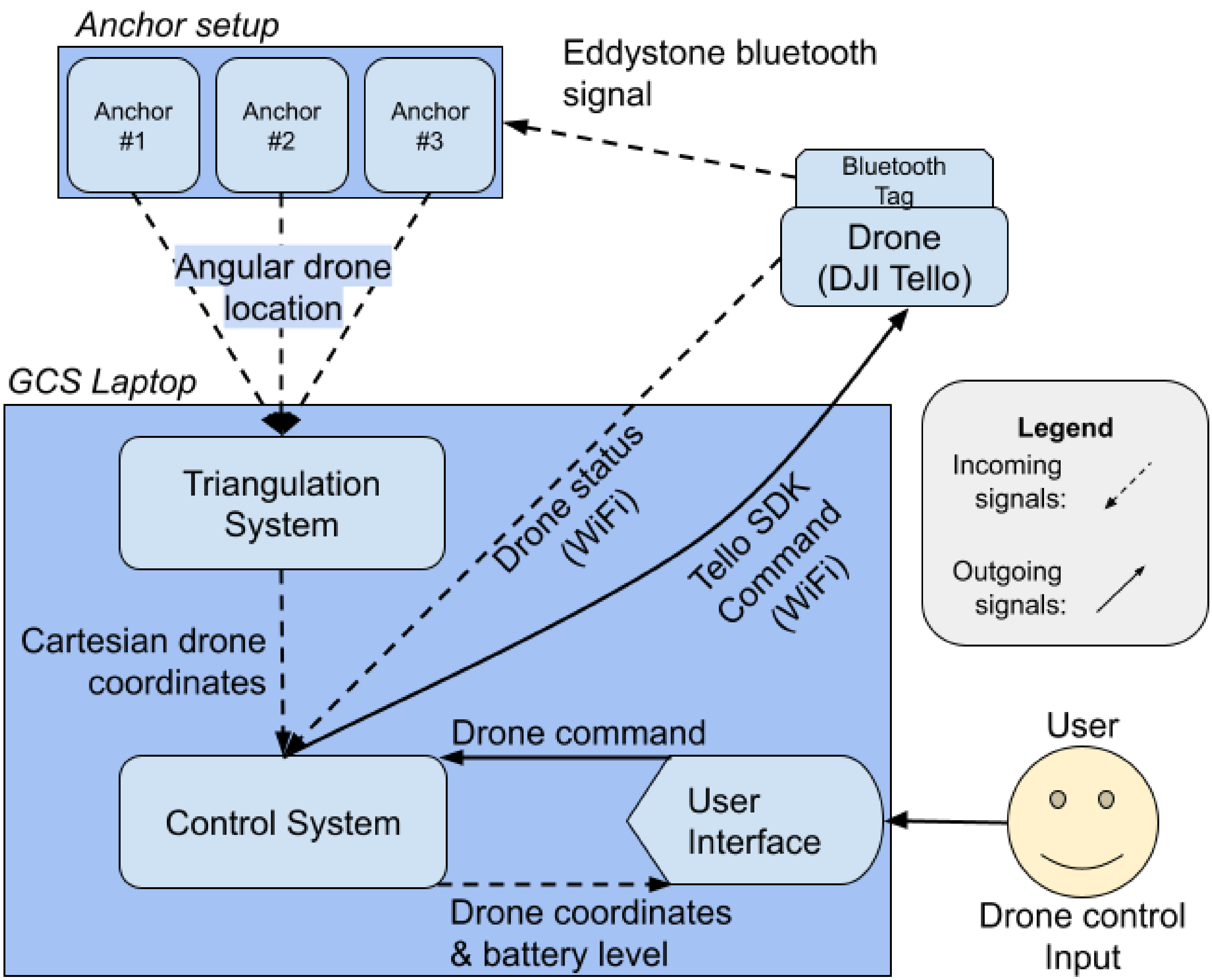


Figure 2 - Block diagram containing all system components

Physical Setup (Author, N. Sydorenko, J. Sung):

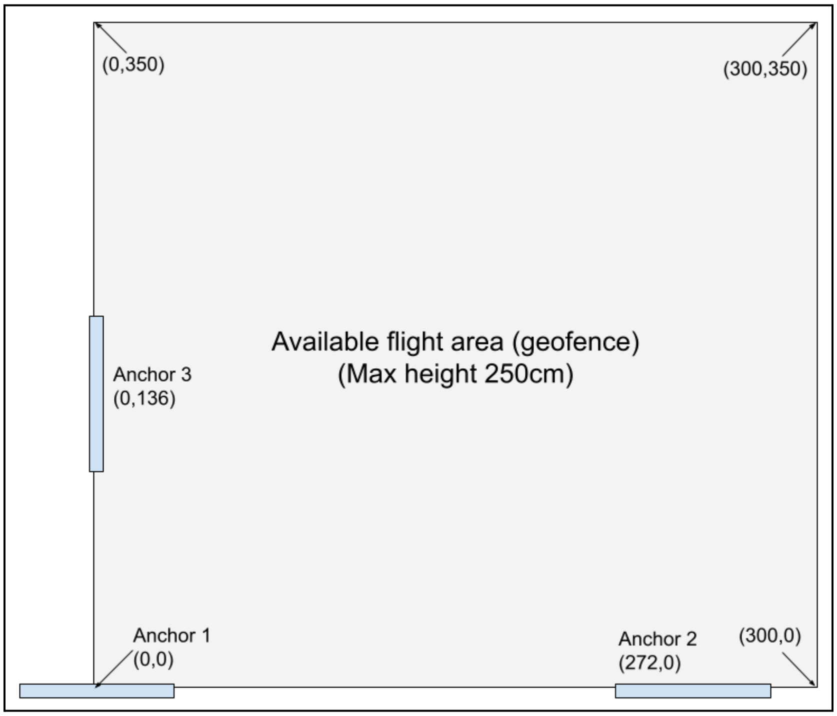


Figure 3 - The anchor setup and geofenced flight area setup. This example was used for testing.

The physical setup for the drone localization system is displayed in Figure 3. This configuration has been determined to provide the most accurate localization results, mainly due to the 3-anchor configuration rather than just 2. These anchors were placed on a table 1m off the ground due to potential signal interference from the possibility of signals bouncing. Using Anchor 1 as a reference point (the origin), Anchor 2 is placed 272cm on the X-axis, and Anchor 3 is placed 136 cm on the Y-axis. This setup creates a clear guide for the drone's geofencing, resulting in the gray area portrayed in the image.

Module-level descriptions and designs:

Drone: (Author: A. Pietrewicz & M. Ahmed)

Our original idea was to build a custom drone from scratch using the Seeed Studio XIAO nRF52840 Sense microcontroller rather than a pre-built commercial drone in order to have a drone that uses Bluetooth 5.0 for communication, to significantly reduce the cost of each drone (~58% from \$198.40 to \$83.14). However, due to our lack of foresight, we failed to note the compatibility of the firmware we wanted to use with the nRF52840 Sense chip and had to push back our milestones.

Due to these hardware difficulties we experienced, the team chose to transition to using an off-the-shelf-drone, the DJI Tello, to allow us to showcase the localization aspect of our project. The drone is connected to using Wi-Fi and controlled using the provided software development kit. The drone localization is done by attaching a Bluetooth tag that emits a signal for the anchor to intercept.

Anchors: (Author: M. Ahmed)

A key component we used to achieve the localization aspect of our project are the uBlox XPLR-AOA-1 BLE AOA boards for the AOA array. The BLE AOA array component of the solution is implemented using the uBlox XPLR-AOA-1 arrays since this product provides everything necessary to localize BLE signals using the software it runs that calculates relevant location information.

The anchor boards are able to detect the angle or direction of the drones within our system using the Bluetooth signal it receives. Each drone is equipped with a Tag which helps the drone act as a beacon by broadcasting Eddystone signals which are received by the anchor boards. The anchors are then able to calculate two key angles, the azimuth and elevation angles which are formed with respect to the location of each drone.

To read these calculated angles from each board, we created a Python program that is run on the Ground Control Station. By polling the port the board is connected to, we can access a variety of information, such as the Eddystone instance ID, azimuth angle, elevation angle, anchor ID and timestamp.

The output of this program (azimuth & elevation angles) is then fed into another program which then calculates the location of each drone in x, y, and z cartesian coordinates using triangulation through various methods. The final coordinates for the location of each drone are then outputted in our GUI.

To improve the drone's localization, we calculated the rolling average of the drone's cartesian coordinates every 20 samples. This was done due to the unreliable nature of Bluetooth localization, as the angles generated were extremely susceptible to any interference (e.g. someone standing in between). As such, many different types of averaging algorithms were tried, with the above method providing the most accurate output.

Ground Control Station: (Author: N. Sydorenko)

Control System:

A big part of any dynamic system is motion planning and collision avoidance which already has many existing solutions [5]. Initially, these existing solutions were to be modified for a custom drone but in the end a DJI Tello was used. The DJI Tello already has usable commands (ex. up, forward, stop, speed) that would control how the drone moves but it was observed that the drone had a tendency to drift which wasn't accounted for. Due to the commands only being able to take in a minimum location change of 20cm, a more intuitive approach was taken to stabilize it since the information given by a transfer function wouldn't work well with the available commands. Therefore a while-if function was created so that after every command was completed, this function would be called that would adjust the drones location if it was outside the desired radius that would be checked and repeated continuously until a new command was issued. This function would be able to take in the continuous flow of any of the four versions of the averaged location data which help refine the location due to the AOA data being inconsistent but more accurate.

Another system had to be implemented to keep the drone within the limits of the space whether it's the physical constraint of the indoor space or the maximum distance between anchors. This geofencing system works by checking the drone's location every 500ms and pushing the drone within the geofence whenever the drone tries to move outside it. This prevents the user from losing signal to the drone or accidentally crashing it into the walls.

User Interface:

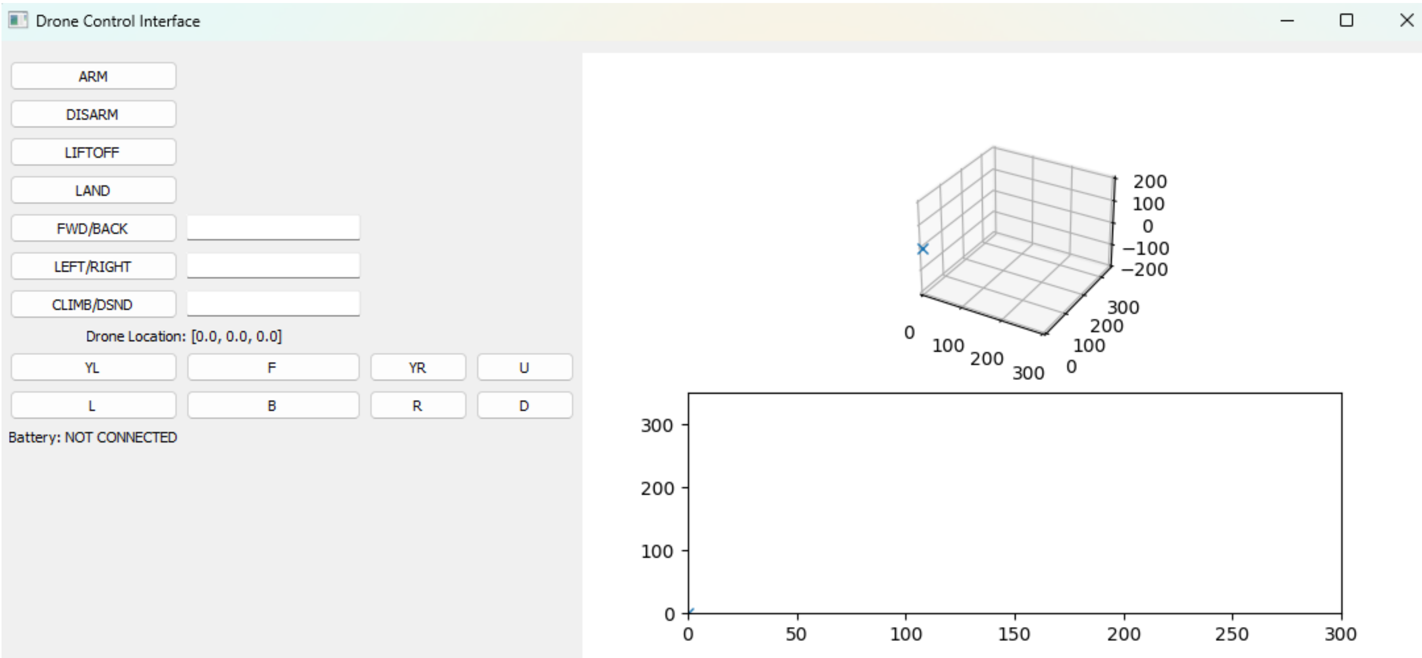


Figure 4 - An initial view of the UI the user sees

Finally, for the user to be able to easily control the drone and view the necessary information, we made a graphical user interface. The UI is made with PyQT5 and sends the control commands to the drone through Wi-Fi. The specific commands are determined by the user where they can click on the command they want to send (e.g. arm, liftoff, left) and also type in the distances they want the drone to move with values >20cm and <-20cm. Below this is a section for manual control where clicking on one of those buttons (e.g. YL - yaw left, U - up, R - right) will allow the drone to move in that direction until instructed to stop. There are also 2 visualizers on the right half for the drone location: 1. 3D drone location display 2. 2D view of drone location on X-Y plane. These, coupled with the drone location coordinates shown middle left will give the user a clear idea where the drone is located in the environment.

Assessment of final design: (Author: M. Ahmed, N. Sydorenko)

After several rounds of thorough testing, there were several conclusions we were able to draw regarding the success and pitfalls of our design. Through various tests, we noticed we achieved the greatest accuracy within our system when we had the anchors placed furthest apart from each other. From this observation, we believe that this design can successfully be used to locate objects in large indoor spaces using BLE AOA technology.

During our testing and experiments, we only used one drone, however our design allows adding additional drones to our system with ease due to each uBlox XPLR-AOA-1 array being able to locate up to 20 tags. The issue with this that is worth mentioning is that a new system would be required to expand on our design to enable controlling all the drones within the system. Currently, only one connection is able to be made using our laptop device with a drone through Wi-Fi.

Other limitations within our design which affect getting consistent accurate results are tied to the uBlox XPLR-AOA-1 arrays only reporting angles in full integers. This is a problem when objects are at longer distances away orthogonal to the plane, the anchors being used to calculate the angle, the derivative of the angle change per distance away from the plane becomes lower resulting in an angle change being smaller than a degree per unit of distance of our requirement which is 20 cm.

Also, only 2 and 3 anchors were used while testing. 2 is the minimum requirement to triangulate the drone's location. Typically indoor drone locating systems use more anchors which would give more location data points (e.g. 1 vs 3 vs 6 for 2, 3, and 4 anchors respectively) for refinement. It was proven that with 3 anchors the location was more accurate so if we increased the number of anchors, we would see more precision in location.

3. Testing and Verification (Author: J. Sung)

ID	Requirement	Verification Method	Verification result and proof
1	Successful communication between the Ground Control System (GCS) and drones	Each communication signal should be properly sent and received by the drone and GCS and the acknowledgement should be shown on the computer terminal.	**PASS:** See testing video in Appendix A

2	The system should be able to detect out-of-bound commands.	A successful test would cause the drone to move 10 meters to the right and then halt within the marked perimeter. The drone should not obey the full command (moving 10.5 meters to the right) as this would cause the drone to go out of bounds.	**PASS:** See testing videos in Appendix
3	Drones must not crash into each other.	*Drone two* should not obey the second command sent to it (move right 1m). A successful test would result in the drone remaining 1m up in the air from its original position, as accepting the move right command would cause a crash to occur with *drone one.*	**UNTESTED** (Were not able to create a multi-drone system)
4	Drone can perform proper commands.	Each drone command should be observed.	**PASS:** See testing videos in Appendix A
5	Low latency between the drone and GCS.	The measured latency should be less than 1 second + the timer's reaction speed.	**PASS:** The latency was within 1m. See testing videos in Appendix A
6	Accurate drone position received by the GCS.	The drone should land within a 20 cm radius of the expected landing position.	**PASS:** Works within a distance of ~200cm within the X-Y plane. See location history in Appendix A.1.1
7	A drone must be a quadcopter.	The drone must have 4 DC motors functioning at a similar level and must take off.	**PASS**: Used a DJI Tello, an off-the-shelf quadcopter
8	Drones must qualify in the "micro-drone" class under Canadian law.	A single drone weighs < 250g	**PASS**: 95g overall weight

4. Summary and Conclusions (Author: M. Ahmed & J. Sung)

Our project has adequately addressed the original problem stated at the start of the project; however, we believe that the proposed solution does not provide the reliability we were expecting. The accuracy of the anchors we used in our design plays a major role in the success of our project. Therefore the number of anchors the team should use was a topic that was discussed in detail as we wanted to design a highly accurate system. The minimum number of anchors required to triangulate the drone's location is two. A question we investigated was whether or not the accuracy improved with the more anchors we used within our system. By testing with two-anchor and three-anchor setups, we found results were more accurate when setting up three anchors. However, even with the slightly increased accuracy, we conclude that the technology we are using does not have the level of reliability we are looking for due to the inconsistent nature of Bluetooth localization.

Another observation we were able to make was that our system was far more accurate in a parallel plane to the pair of anchors than in a plane orthogonal to the anchors. The localization system accuracy was the best when the tag was within a 30-150 cm range on the Y-axis. Based on these observations, we believe that rather than having the anchors at an elevated position, it would be best to mount the anchors on the ceiling or the floor directly for the best localization. The lacking accuracy in the orthogonal (height) axis to the anchors' planes can be assisted with internal barometers or time-of-flight sensors, which can be acquired at a very low cost as well.

Overall, we believe that Bluetooth localization technology is too unreliable at its current stage to be used for drone localization without significant algorithmic support. As such, we expect to see camera systems that utilize computer vision to locate the drone in an indoor space become more prevalent in the near future. However, with sufficiently powerful averaging algorithms and possibly support from machine learning models, a system of Bluetooth-localized drones may be feasible at satisfactory accuracy.

5. References

[1] M. Saad, C. Bleakley, and T. Ballal, "High-accuracy reference-free ultrasonic location estimation - researchgate." [Online]. Available: https://www.researchgate.net/publication/231169921_High-Accuracy_Reference-Free_Ultrasonic_Location_Estimation. [Accessed: April 6, 2023].

[2] Hollander, D. "How aoa & AOD changed the direction of bluetooth location services," *Bluetooth® Technology Website*, March 2022. [Online Serial] Available: https://www.bluetooth.com/blog/new-aoa-aod-bluetooth-capabilities/ [Accessed: October 24, 2022].

[3] Pau, G. *et al.,* "Bluetooth 5.1: An analysis of direction finding capability for high-precision location services," *MDPI. Multidisciplinary Digital Publishing Institute*, May 2021. [Online serial]. Available: https://www.mdpi.com/1424-8220/21/11/3589/htm. [Accessed: October 24, 2022].

[4] "Getting Started with Seeed Studio XIAO nRF52840 (Sense) | Seeed Studio Wiki," *wiki.seeedstudio.com*. https://wiki.seeedstudio.com/XIAO_BLE/ [accessed November 15, 2022].

[5] M.M. Iqbal, *et al.,* "Motion planning of UAV SWARM: Recent challenges and approaches, IntechOpen," *IntechOpen*, June 2022. [Online serial]. Available: https://www.intechopen.com/online-first/82985 [Accessed: March 24, 2023]

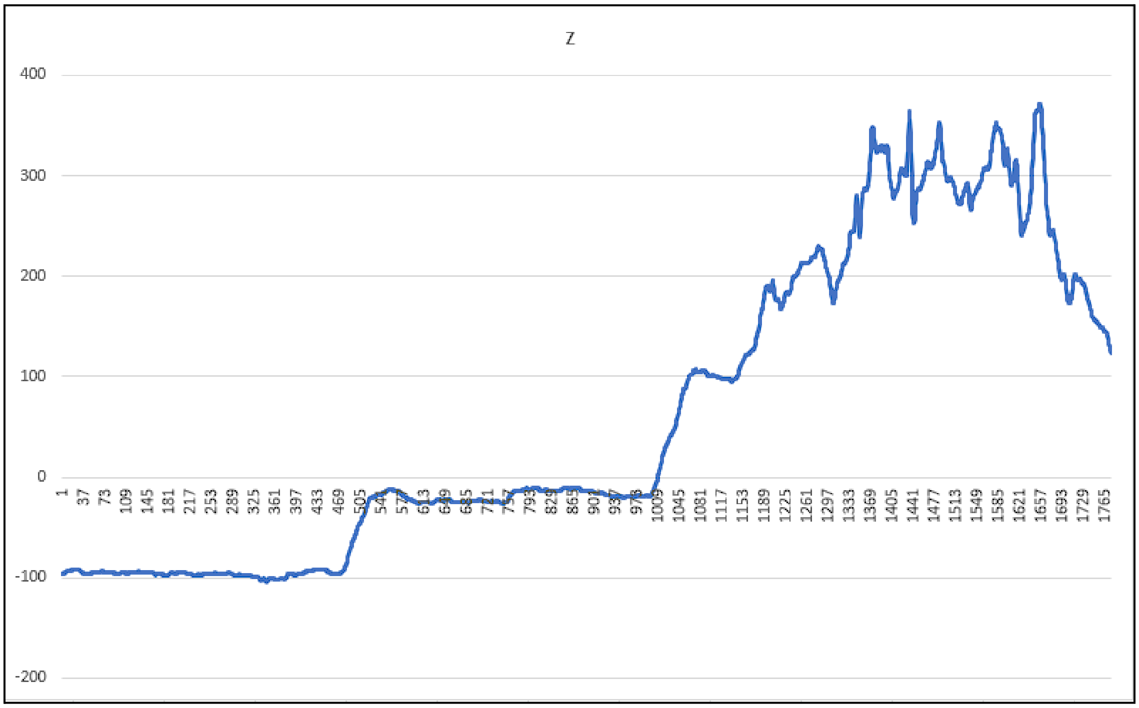
Appendices

Appendix A - Final System Test Data

A.1 - Video of Test #1 (z-axis geofence test)

https://drive.google.com/file/d/1NH0muBSgEOdrxWr0-UtgOob-jijk3Q7C/view?usp=sharing

A.1.1 - Z-axis location history for Test #1 (z-axis geofence test)



The initial position of the drone on the ground, 1m below the origin, clearly shows that the localization works sufficiently within the 20cm accuracy stated within the testing requirements. The geofencing to keep the drone within boundaries is shown here as well.

A.2 - Video of Test #2 (x-axis, y-axis geofence test)

Part 1 of video demo:

https://drive.google.com/file/d/1PkyRYfLR9hi5IELO7VfkS-DKzEzyndVv/view?usp=sharing

A.2.1 - Screen Capture Recording of Test #2 (x-axis, y-axis geofence test)

https://drive.google.com/file/d/1VXc535RcpWyfJE1s8rFmwKORYnf9XFoD/view?usp=sharing

The screen capture shows how the UI was used to display the relevant information to the user while allowing the user to control the drone