Autonomous Aerial Mapping

Team Name: Game of Drones

Authors

Trenton Cisewski – trentoncisewski@gmail.com
Sam Goyal - s4mgoyal@gmail.com
Chet Koziol - chet.koziol@gmail.com
Mario Infante - marioinfantejr@gmail.com
Peter Ginty - peterginty13@gmail.com

Introduction

Applications for unmanned aerial vehicles have allowed industries to embrace revolutionary strategies for technological growth and development. This project aims to tackle one of these applications, developing new three dimensional mapping solutions with industry leading drone and lidar technologies. Our system will tackle the difficulties in gathering 3D map data in areas with limited topographic information immediately available. The current availability of 3D map data is limited and often presents inaccurate information. We aim to eliminate these issues by developing a drone able to autonomously map nearby terrain by integrating various sensors to produce an interactive model of the area on the user's phone.

The fundamental problem this project solves is how to rapidly create a three dimensional model of an unknown area using autonomous technologies, and display this data to a user. This process can be broken down into four main components. The first is interfacing with the drone itself. We will be using a DJI Matrice 100 quadcopter to develop and deploy our aerial mapping system. The second major component is data collection, which will be done by mounting a lidar sensor to our quadcopter. Lidar sensors can output industry standard binary files, or object files, containing the coordinates for a point cloud. This point cloud will then be rendered into a 3D model, which will match the landscape surrounding the quadcopter's flight path. The lidar sensor we have selected is the Sweep V1 360°. The Sweep V1 outputs a distance to the nearest point and a rotation angle and signal strength associated with that point. It rotates 360° at a frequency between 2-10 Hertz, allowing for 2D mapping out-of-the-box. The third component and brain of the system is the Raspberry Pi. The Pi is connected to the DJI Matrice as well as the Lidar Sensor. The input will be the point distance, as well as the telemetry data of the drone. Our algorithm will then integrate this data to create the corresponding .OBJ file, containing all the data points. The Raspberry Pi will then send this file via Wi-Fi to our mobile device application. This mobile application will be the fourth component, rendering the file and displaying the corresponding terrain model to the user.

The second problem we will address is designing the drone to be fully autonomous. This process begins with the mobile application. The user will select the autonomous flight option, enter the square meters they want scanned, and select the where they want the drone to scan. This command will be sent from the phone to the Raspberry Pi, which will relay the information to the drone. The drone will then slowly fly over the area, gathering the best possible point data. Attached to the drone will be the DJI Guidance System, which has four sensor bars and cameras to alert the drone when an object comes within 20 meters. The drone will then be able to avoid obstacles during this autonomous flight. This process will repeat itself until the surrounding area has been scanned.

Along with the two main problems, there will be many features associated with the mobile application to develop. The app will display information about the drone such as battery life, velocity, and geographical position. Additionally, automatic landing options will be presented to the user. On the screen of the application, the user will be able to zoom in and pan around the 3D model, to view the landscape from every angle. The user can also choose to get a bird's eye view through the drone, while the drone flies on their phone screen.

Our main objective is to get the drone to fly over an area, first manually and then autonomously, and produce a 3D rendered model of the area that the user can interact with. Our objective relies most heavily on our ability to successfully integrate lidar and drone data to transform it into an accurate point cloud, which can then be rendered by additional software to create the model. As a team, we want to be able to successfully create a new architecture, integrating both hardware and software components, to bring technical advancements to drone-mapping, with a low cost solution.

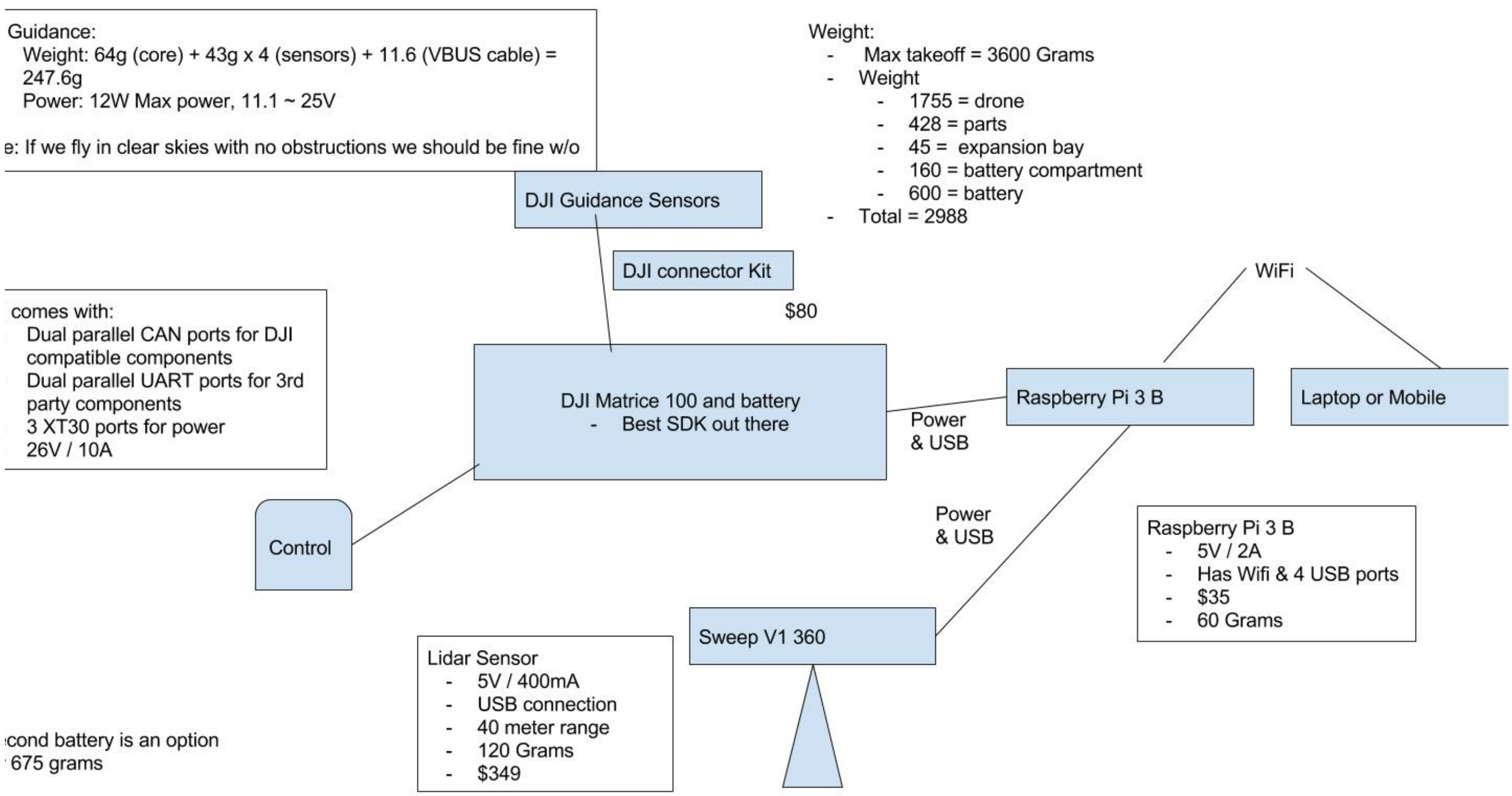
The core technological advancement is designing this low cost solution. Typical lidar sensors for this application cost thousands of dollars, while our Sweep V1 costs only a few hundred. This sensor does not provide the typical point cloud output, but our data collection algorithm can fill this gap, and provide 3D output. This advancement has far reaching applications such as autonomous flying and driving on a mass scale due to the cost efficiency alone.

Some assumptions shall be made about this system when it is interacting with the user. These include:

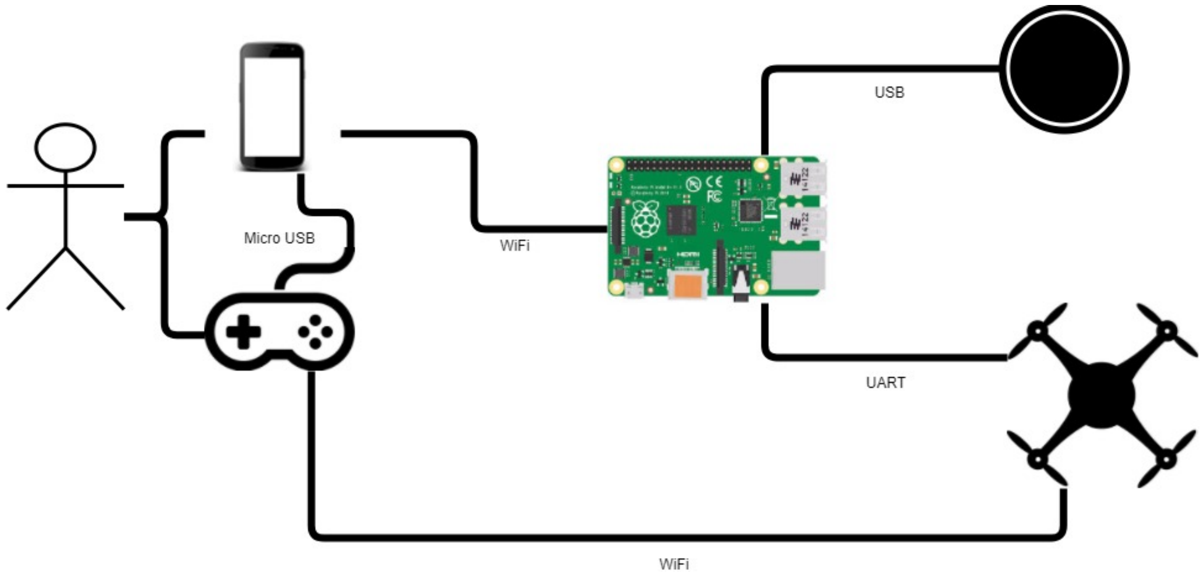
- Cannot accurately map an environment/object that dynamically moving and changing
- Ensuring the drone will be flying within the Controller and Wi-Fi ranges
- Making sure the drone is flying in appropriate weather conditions (i.e. not raining, no heavy wind, no storms, etc.)
- Mapping generally flat areas/Avoiding terrain that varies greatly

System Architecture

High Level Diagram



User Interaction Diagram



Requirements

This project requires many different use cases from the system’s perspective. Many of these cases can be broken down further, but some of these include:

1) The system shall transmit an object file from a Raspberry Pi to an Android Mobile Device.
    a) Acceptance criteria: visual inspection of object file on mobile device
        i) Make sure format of file is correct on mobile device
    b) Time to complete: 8 hours
2) The system shall transmit data from the lidar sensor to the Raspberry Pi.
    a) Acceptance criteria: visual inspection that lidar data is present on raspberry pi
        i) Make sure lidar data is properly formatted for .obj file formats on the pi
    b) Time to complete: 4 hours
3) The system shall transmit drone telemetry to the Raspberry Pi
    a) Acceptance criteria: visual inspection that telemetry data is present on raspberry pi
        i) Make sure data is properly formatted for .obj files and have the proper time-stamps for interpolating/dead-reckoning data
    b) Time to complete: 4 hours
4) The Raspberry Pi subsystem shall provide mathematical computations such that the data provided from the sensors can be transformed accordingly and accurately.

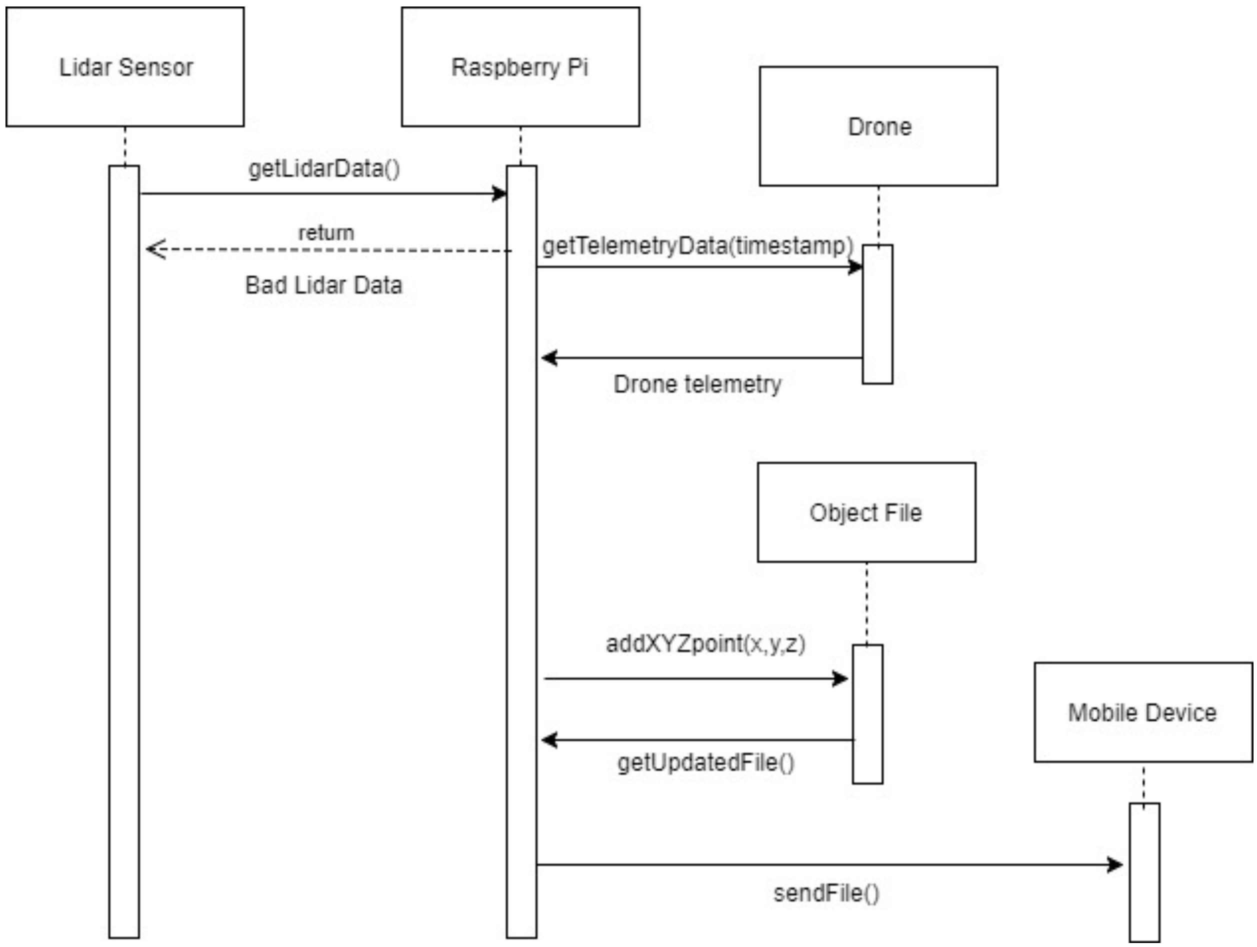
    a) Acceptance criteria: transformations pass all the unit tests defined in Github
        i) Transformations output the data into the proper .obj file format using interpolation/dead-reckoning
    b) Time to complete: 20 hours
5) The system shall transmit data from the drone to the Android Mobile Device.
    a) Acceptance criteria: drone battery life, telemetry, etc. is shown on mobile device
    b) Time to complete: 12 hours
6) The Raspberry Pi subsystem shall convert the received transformed data and convert it into an object file.
    a) Acceptance criteria: check the file built to make sure the data is correctly inputted in the stream
7) The Android Mobile Device Subsystem shall be able to display the object file to the user.
    a) Acceptance criteria: Object file is viewable on android device
    b) Time to complete: 10 hours
8) The Android Mobile Device Subsystem shall allow the user to interact with the 3D model. Interactions include zoom in/out, rotating, etc.
    a) Acceptance Criteria: the file is able to be altered by the user
9) The Android Mobile Device Subsystem will allow the user to input a size on a map to constrain the drone for autonomous flight.
10) The system will return back mid-flight if it senses that it is running out of battery, or if the weather conditions suddenly get unpleasant during flight.
    a) Acceptance criteria: drone lands itself automatically when it has low battery
    b) Time to complete: 6 hours
11) The system shall be wired according to specifications to ensure connectivity and proper functionality.
    a) Acceptance criteria: Drone behaves as expected and works.
    b) Time to complete: ? hours
12) The Android Mobile Device Subsystem shall allow the user to trigger the sensors to scan while flying the drone in the manual mode.
    a) Acceptance criteria: User can start and stop the sensor from scanning while drone is in flight.
    b) Time to complete: 10 hours
13) The system shall create a wireless connection between the phone and Raspberry Pi upon power up.
14) The transformation function subsystem shall timestamp telemetry and lidar data for synchronization.
    a) Acceptance criteria: the lidar timestamp matches the telemetry timestamp
    b) Time to complete: 8 hours
15) The system shall exclude erroneous points from the 3D model.

    a) Acceptance criteria: lidar data with a low signal strength and lidar data associated with the bottom of the drone will not be included in the object file.
    b) Time to complete: 8 hours
16) The drone will be flown autonomously
    a) Acceptance criteria: The drone maps an area that the user predetermined on their mobile device
    b) Time to complete: 40 hours
17) The system will be able to build and display the map on the mobile device in real time.

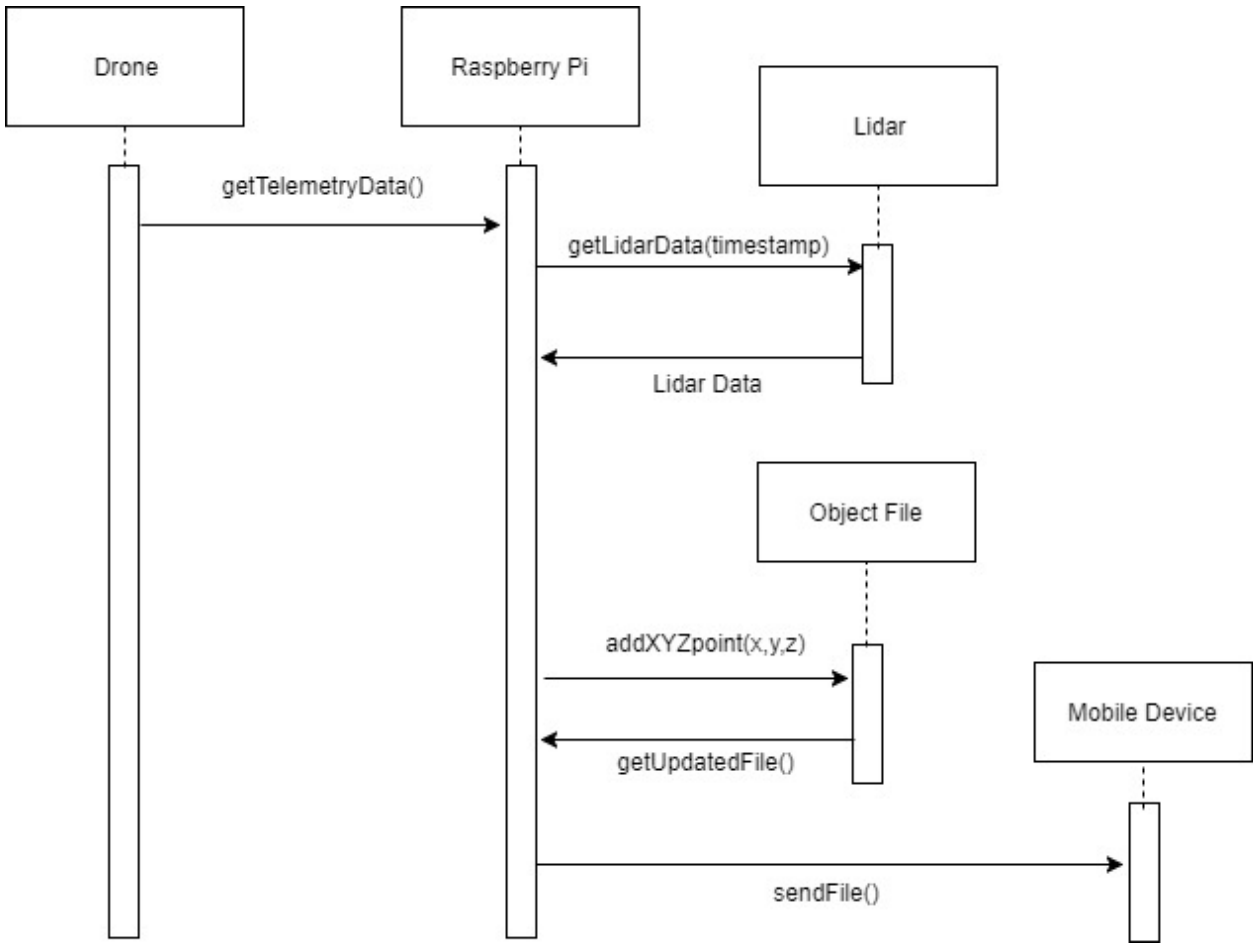
System Models

Activity Diagram

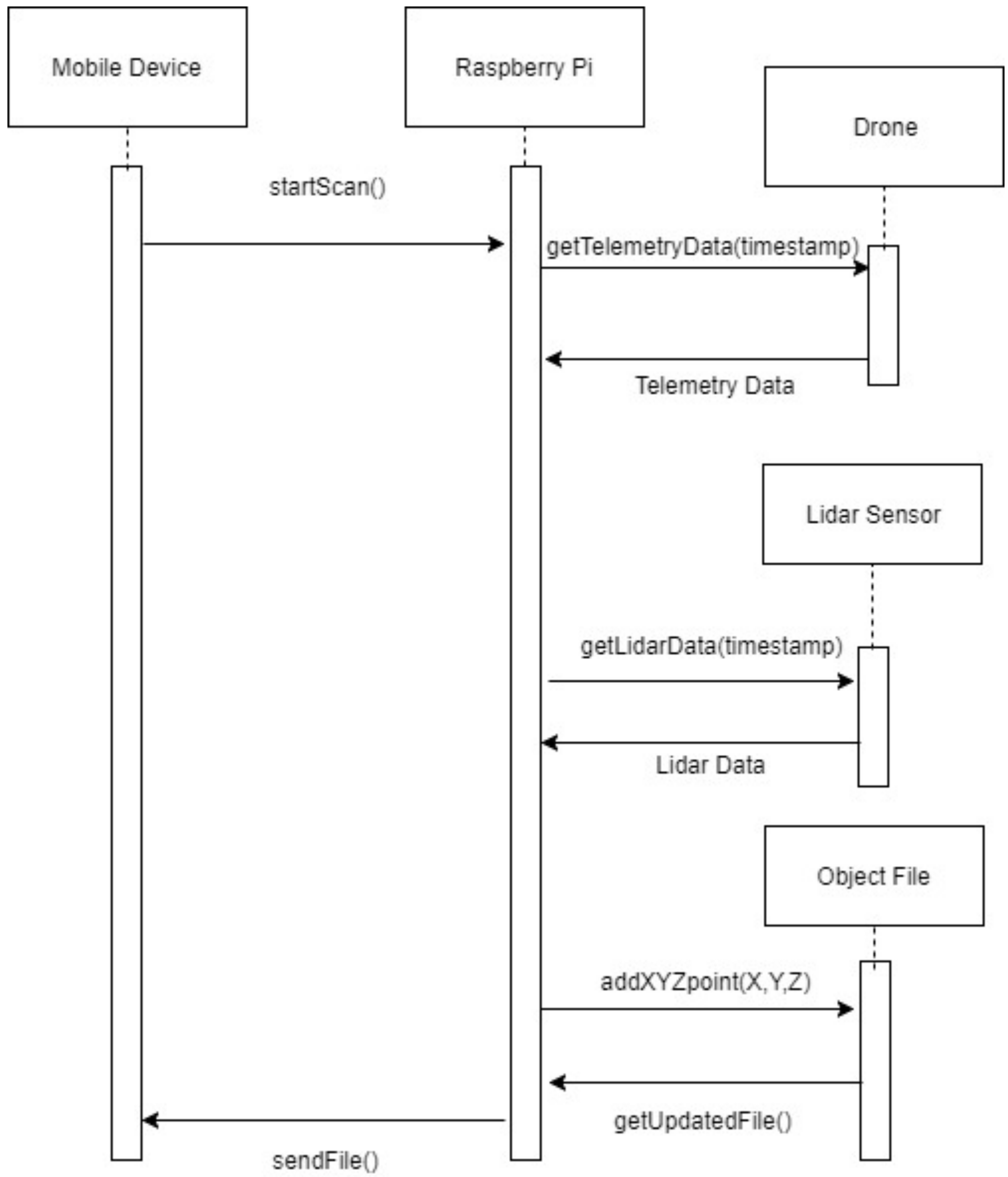
Lidar Data Sequence



Telemetry Data Sequence



Mobile Data Sequence



Technologies Employed

This project requires the use of many different technologies coming together. These include:

- DJI Matrice 100 - Drone
- DJI Guidance System - Has built in sensors and cameras used for obstacle avoidance, but data can be independently read off of them as well
- Sweep V1 360 Degrees Laser Scanner - Lidar sensor used to collect data to help create our point clouds
- Raspberry Pi - Used for computing as well as sending data to the user.