**Boston University**

**Electrical & Computer Engineering**

**EC463 Senior Design Project**

First Semester Report

Recent Progress of Aerobatic BlackBox

Submitted to

Dr. Kenneth D. Sebesta

CyPi Ltd.

859-648-0272

[kenn@cypiltd.com](mailto:kenn@cypiltd.com)

by

Team 9

Aerobatic Black Box

Team Members

Darcy Meyer darcym@bu.edu

Pai Liu paiii@bu.edu

Radhey K. Patel rexu6@bu.edu

Wenjun Ma wjm123@bu.edu

Xinyu Liu xinyubu@bu.edu

Submitted: Dec 12, 2021

#### Table of Contents

[Table of Contents](#_heading=h.z5tuqhgkwwxv) 2

[Executive Summary (Pai Liu)](#_heading=h.3ze3umhfv9ig) 3

[Introduction (Pai Liu)](#_heading=h.1fob9te) 4

[Concept Development (Xinyu Liu)](#_heading=h.3znysh7) 5

[System Description (Wenjun Ma)](#_heading=h.tyjcwt) 8

[First Semester Progress (Radhey K. Patel)](#_heading=h.3dy6vkm) 10

[Technical Plan (Darcy Meyer)](#_heading=h.1t3h5sf) 11

[Budget Estimate (Wenjun Ma)](#_heading=h.4d34og8) 13

[Attachments](#_heading=h.2s8eyo1) 14

[Appendix 1 – Engineering Requirements](#_heading=h.17dp8vu) 14

[Appendix 2 – Gantt Chart](#_heading=h.3rdcrjn) 15

[Appendix 3 – Other Appendices](#_heading=h.26in1rg) 16

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Executive Summary (Pai Liu)

*Aerobatic BlackBox*

*Team 9 – Aerobatic BlackBox*

Aerobatic performances are very different from ordinary flying tasks. General pilots use scientific methods to avoid risks during training, while aerobatic pilots take the initiative to face risks and use their lives to challenge the limits of aircraft performance and flying skills. In the field of aerobatics, there is no limit to the exploration of pilots' flying skills. In different aerobatics, the aircraft's flight status, altitude, speed, direction, and overload parameters change drastically. The pilot must be aware of the flight status at any time, understand the changes in motion parameters, and control the aircraft to move on a predetermined trajectory in a timely and accurate manner. During competition training, ordinary avionics cannot record and replay 3D demonstrations of flight training, and pilots cannot make precise movement adjustments.

In order to help pilots to understand the status of the aircraft, the AHRS came into being, which can help pilots understand the real-time aircraft attitude. However, this system lacks 3D playback of aerobatics, cannot help the pilot to watch the aircraft attitude from the judge's point of view, and cannot record real-time pilot input, which does not meet the needs of aerobatic training.

Our Aerobatic Black Box will address these issues by delivering a hardware part which contains AHRS, INS, GPS and a camera to collect pilot input, and a software part which allows pilot replay 3D demonstration of flight training, and also provides pilot input corresponding to every maneuver. Aerobatic Black Box is specially designed for introductory aerobatics pilots and aerobatics training. This system will allow pilots to observe different angles of flight attitude for the next flight adjustment, and also solves the lack of digital AHRS in some aircraft, maneuver log and greatly reduces the difficulty of aerobatics training.

.

# Introduction (Pai Liu)

During aerobatic competition the aircraft's flight status, altitude, speed, direction, and overload parameters change drastically. Polite must be aware of the flight status at any time, understand the changes in motion parameters, and control the aircraft to move on a predetermined trajectory in a timely and accurate manner. During competition and training, AHRS will help pilots understand the real-time aircraft attitude, however this device cannot record and replay 3 demonstrations of flight training, and pilots cannot make precise movement adjustments based on ordinary avionics.

Under this premise, the client put forward the following requirements to us: First, the equipment must be able to record flight data (including altitude, GPS, and flight attitude). Second, the equipment must be able to record the pilot's input to the aircraft. Third, the equipment must be able to replay 3D flight simulations to help pilots adjust flight movements. Fourth, flight simulation must be able to switch perspectives to help pilots observe aerobatics.

Our Aerobatic Black Box will address client’s issues by providing a hardware part which contains AHRS, INS, and a camera to collect pilot input; the software part which allows pilot replay 3D demonstration of flight maneuvers, and also provides pilot input corresponding to every maneuver. When collecting pilot input, we are faced with the problem that the aircraft does not have a collection interface, so we innovatively use image recognition technology to collect airspeed and pilot input.

The Aerobatic Black Box is specially designed for introductory aerobatics pilots and aerobatics training. This system will allow pilots to observe different angles of flight attitude for the next flight adjustment, and also solves the lack of digital AHRS in some aircraft, and greatly reduces the difficulty of aerobatics training.

# Concept Development (Xinyu Liu)

**2.1 Customer’s problem**

The client wants us to correlate the pilots’ control inputs to the airplane’s behavior, and this correlation should be able to visualize what the maneuver looked like from the ground. According to the client’s requirement, we need to create a system installed in a plane that records the pilot's inputs and the state of the plane over time then visualizes them after the flight. For the statues of the plane, it contains 3 basic variables in 3 axes: roll, pitch, and yaw. We change the status of the plane by changing the value of these variables. Besides, to represent the plane under the actual situation, we need 5 extra variables: altitude, longitude, latitude, ground speed, and airspeed. Therefore, by these 8 variables, we can record the status of the real plane. For recording the pilot’s inputs, we use the camera to capture the pilot’s behavior. There are 2 basic elements in a pilot’s behavior: rudder pedal and flight stick. The rudder pedal controls the rear rudder of the plane and the flight stick controls the elevator and the aileron of the plane. To precisely capture the pilot’s input, we need to quantify the movement of the pedal and stick. For the visualization, we need to use a 3-D plane module that can represent the corresponding plane and insert it into a real map.

**2.2 Customer’s need**

According to the client’s problem, we separate the system into 3 parts which contain data collection, operating environment, and visualization system.

## 

## 2.2.1 Data Collection

1. The data collection system must measure and record the position (latitude, longitude, and altitude) and attitude (pitch, roll, and yaw) of the aircraft over time
2. The data collection system must measure and record the positions of the flight stick, rudder pedals, and throttle (collectively the pilot’s control inputs) over time
3. The data collection system must measure and record the value of the airspeed indicator dial located in the cockpit of the aircraft over time
4. The data collection system must continuously record each of the above data points at a frequency of at least 20Hz
5. The data collection system must record a timestamp alongside each of these data points

**2.2.2 Operating Environment**

1. The data collection system must survive temperatures of -40°C to 50°C in storage and 0°C to 40°C in operation
2. The data collection system must continue to collect data according to the Measurement Requirements and remain securely attached to the interior of the plane while under a g-force of at least 15G
3. The data collection system must survive the vibrations of an aerobatic plane in flight – to be tested in a Citabria 7ECA
4. The data collection system must operate for at least 30 minutes during flight without any interaction from the pilot
5. The data collection system must be powered either by a 12V bus or by battery
6. The data collection system must not modify any part of the aircraft to comply with FAA regulations

**2.2.3 Visualization System**

1. The visualization system must provide a 3D representation of an aerobatic flight reconstructed from the flight data recorded by the data collection system
2. The visualization system must allow the user to view the flight from any camera angle
3. The visualization must play at 20Hz or more

**2.3 Approach**

According to the customer’s needs, we create a system which contains Raspberry Pi 4B, 10 DOF IMU, GPS Receiver with external antenna, Raspberry Pi4 Camera IR, Image Processing Method, and Xplane11 Simulator.

1. **Raspberry Pi 4B**

Raspberry Pi 4B is a SBC (single board computer) which not only has a small size but also has the full function of a regular computer. We only need Raspberry Pi 4B to extract the data from the data and to process the image so we don’t need a full-size computer which is impossible to place inside a plane. Besides, compared to Arduino, it has the full functionality of the normal computer which can provide a larger range of flexibility. We don’t need to use a specific programming language to reach our goal. This provides us more choice to choose the sensors and to write algorithms.

1. **10 DOF IMU**

10 DOF IMU (10 Degree of Freedom Inertial Measurement Unit) contains 3 DOF gyrometer, 3 DOF accelerometers, 3 DOF magnetometers, and a pressure sensor. By combining the data from gyrometer, accelerometer, and magnetometer, we can get the precise value of roll, pitch, and yaw of the plane. By getting the data from the pressure sensor, we can get the value of the altitude. Compared to 6 DOF IMU which doesn’t include the magnetometer, the 10 DOF IMU can resolve the shifting data from gyrometer and accelerometer. Besides, 10 DOF IMU can get the value of the altitude, which reduces the cost of the whole project since we don’t need to buy another pressure sensor.

1. **GPS Receiver with external antenna**

GPS Receiver with external antenna can provide the precise data of longitude, latitude, ground speed, and altitude. The GPS receiver we use is incorporated with the 10 DOF IMU. Compared to other GPS receivers, this GPS receiver significantly reduces the size of the whole system since we don’t need to use extra wires to connect the sensors with Raspberry Pi. Also, with the external antenna, this GPS receiver can get more precise data and faster start speed.

1. **Raspberry Pi4 Camera IR**

Raspberry Pi4 Camera IR can record the pilot’s inputs and airspeed gauge in the plane. Compared to other regular cameras, this camera can automatically switch between the day and night modes, so we don’t need to create an extra night vision camera to capture the pilots in the night. This reduces the costs and the workloads of the project. Besides, it records the video with 1080p resolution so our image processing algorithm will receive a more precise image to measure the movement of the stick, pedal, and the airspeed gauge.

1. **Image Processing Method**

The method we use is to place a tape over the stick and pedal. Then we set the initial position of the stick, pedal, and the pointer of the airspeed meter. By measuring the movement between the initial position and the final position, we can get the change of the pilot’s inputs and the airspeed. Compared to other methods which need to insert a wire into the plane’s control system which is not approved by the client, our method can provide a precise measurement of the movement of pilot’s inputs and the airspeed gauge.

1. **Xplane11 Simulator**

The Xplane11 Simulator can create the visualization of a plane by providing corresponding roll, pitch, yaw, longitude, latitude, altitude, ground speed, airspeed, the movement of stick and pedal. This simulator has a large number of authorized aircraft modules and a complete real-time map. Compared to building a new application with 3-D engine and map, this simulator extremely reduces the workload of the project and allows the users to choose the plane that they fly in the real world, which gives them a more real experience to review their flights.

# System Description (Wenjun Ma)

To meet all of the engineering requirements, we design the system as below:

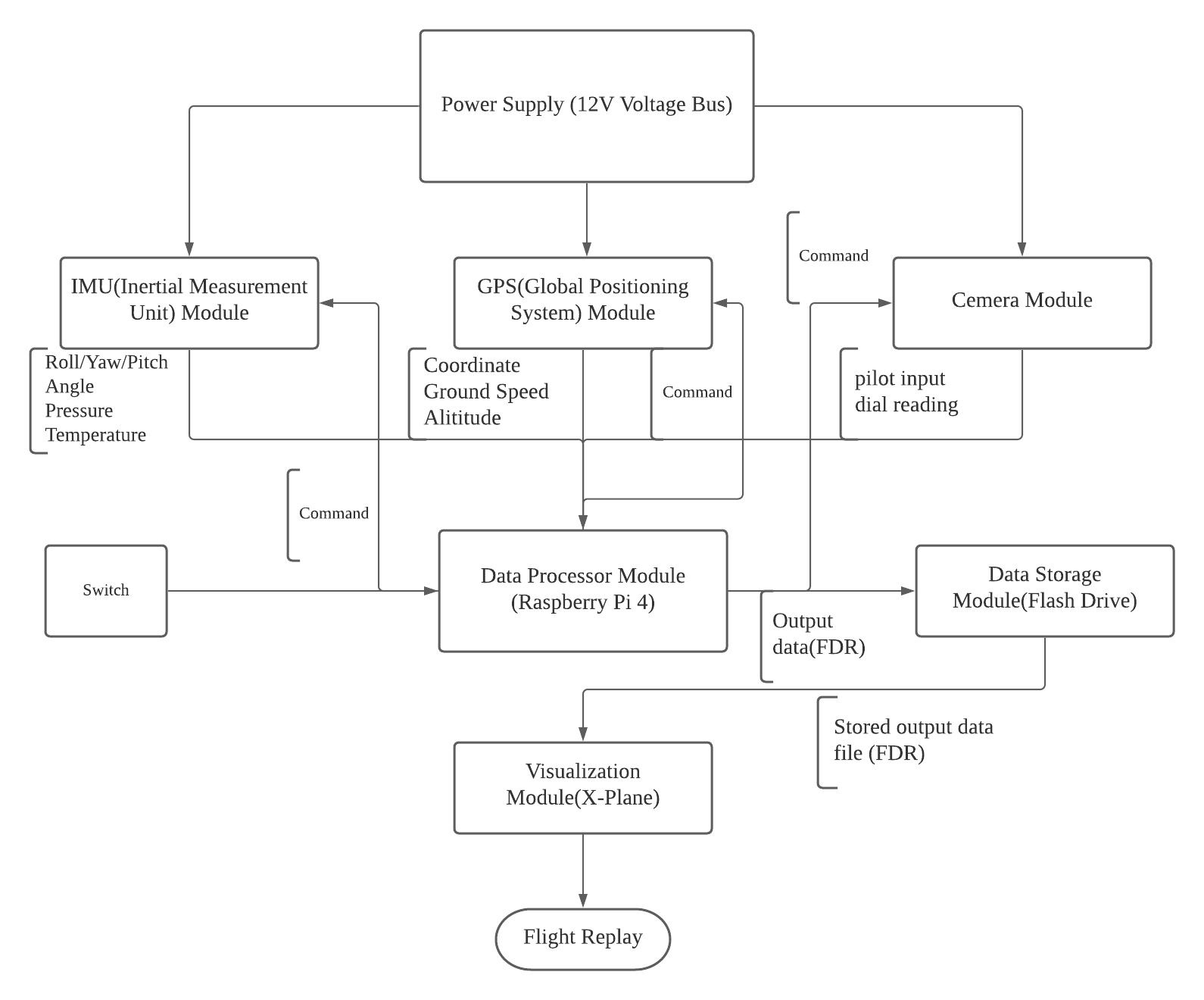


Figure 3.1: Block diagram of the Aerobatic Black Box

* IMU (Inertial Measurement Unit) Module: BerryGPS-IMU GPS and 10DOF
* GPS (Global Positioning System) Module BerryGPS-IMU GPS and 10DOF, GPS antenna (AA.162.301111)
* Camera Module: MakerFocus Raspberry Pi4 Camera IR, Raspberry Pi4 Camera Night Vision Fisheye Camera 5mp OV5647
* Data Processor Module: Raspberry Pi4 B
* Data Storage Module: Flash drive
* Visualization Module: Xplane

Our Aerobatic Black Box will address issues by delivering a hardware part which contains AHRS, INS, GPS and a camera to collect pilot input, and a software part which allows pilot replay 3D demonstration of flight training, and also provides pilot input corresponding to every maneuver. The system is functionally consisting of 6 parts. IMU (Inertial Measurement Unit) Module, GPS (Global Positioning System) Module, Camera Module, Data Processor, Data Storage Module and Visualizing Module, which are all supplied by a 12V voltage bus mounted in the aerobatic plane. The IMU module could not only measure the roll,yaw and pitch angle of the plane while the plane is in flight, it also measures the temperature and the surrounding pressure. The GPS module measures the specific coordinates of the global coordinate system, altitude and calculates the ground speed by processing the coordinates. The camera module, which needs to be mounted in an appropriate place in the aerobatic plane, will record the pilot inputs by measuring the movement of Rudder cables and sense the dial reading from the dashboard on the aerobatic plane. Raspberry Pi 4 B is selected as the data processor of the system since its small size and great capability to work. We would also use a flash drive to save the data which was produced by the Raspberry to be later analyzed for visualization on Xplane.

Once the switch of the system is turned on, the raspberry pi will receive the signal to work and send the command to IMU,GPS and Camera module to start collecting data. The output data will be transmitted back and processed by the raspberry pi. The raspberry pi will save these processed data which will be in fdr format into the flash drive which is needed to be inserted into the system. After the flight is over, the flash drive can be taken out. The data which is stored in the flash drive could be extracted and read by Xplane. Finally, the Xplane will read and analyze the data and generate a whole flight replay which allows pilot replay 3D demonstration of flight training, and also provides pilot input corresponding to every maneuver.

# First Semester Progress (Radhey K. Patel)

Currently, our system constitutes Raspberry Pi 4B, 10 DOF IMU, GPS Receiver with external antenna, Raspberry Pi4 Camera IR, Image Processing Method, and Xplane11 Simulator as well as data extraction programs for the aforementioned sensors. The team initiated the progress with client meetings, user interviews, site visitation followed by a thorough investigation of the functionalities and benefits requirement for the project to determine feature selection (detailed overview of these components have been mentioned in section 2.0).

The current blackbox hardware has been calibrated and tested for measurements from the GPS, Inertial Measurement Unit (IMU), barometric pressure sensor, gyroscopic as well as magnetometric alterations. Moreover, the software based calculation system for altitude (both pressure dependent and altitude), trajectory recorder based on the sensor data in addition to a kalman filter for noise correction; though, needed to be time-synchronized, the data collector presently can write the data independent of frequency coordination to be sampled for the visualization component. The Xplane 11 has been customized to a multi-perspective view, where a .fdr file feeder can initialize a simulation generator based on the data. This file type can be altered to a .sit (situation) or .rec (recording) file type to log alteration point during the pilot’s feedback for training/flight.   
 The testing parameters and successful results for the current semester included:

1. Data extraction from IMU sensor and outputting data stream for changes in Yaw, Roll, and Pitch/Heading in live time.
2. Data extraction from pressure-temperature sensor and outputting data stream for changes in barometric pressure in live time as well as calculating altitude based on it.
3. Calibration of a 3-satellite based GPS location and live time data streaming and recording.
4. Calculating alterations on the instrumentation dials with visible angular alterations using a machine learning algorithm.
5. Deploying .fdr into Xplane 11 to model flight trajectory and adding alternative data writers within the software.

# Technical Plan (Darcy Meyer)

**5.1 Data Collection**

*Read GPS/IMU sensor outputs*

As stated. Since we have already figured out how to read all these raw values, this task is a final verification that we are doing this correctly. Deliverable: all data values from the GPS/IMU sensors are read using scripts on the Pi. Lead(s): Allen, Pai.

*Airspeed dial image analysis*

Given a video (or sequence of images) of an airspeed dial, use image processing to read the value of the airspeed dial. Output the value in kph. Deliverable: A sequence of scalar values that show the value of the airspeed dial over time. Lead(s): Darcy.

*Cable positions image analysis*

Given a video of some cables with tape marks on the cable, track the motion of the cables over time. Deliverable: A sequence of scalar values that show the position of the cable over time. Lead(s): Darcy.

**5.2 Data Processing**

*Synchronize data arrival times*

Compensate for different sensor latencies as much as possible, to get accurate timestamps on each data point. Deliverable: accurate timestamps on data points from each data input method. Lead(s): Allen, Rex.

*Smooth data with Kalman filter*

Given telemetry data points for gps and acceleration, create a Kalman filter to model the path of the plane in the presence of noise in the sensor readings. Deliverable: a script implementing a Kalman filter that denoises our data. Lead(s): Darcy, Rex.

*Interpolate between data points*

Turn lower frequency data into higher frequency data by interpolating between sparse data points. The data frequency used for the visualization should be consistent with the project requirements. Deliverable: Given some sparse data points of different types, generate a data stream that is dense. It should be reasonable given the original data. Lead(s): Allen.

*Data unit conversion*

Convert all recorded data into the units required by the .FDR file format. Deliverable: a component in our data path that converts all data into the correct units. Lead(s): Rex, Pai, Marvin.

*Automatic .FDR file creation*

Collect the data into a correctly formatted .FDR file that can be read by XPlane. Deliverable: there should be a direct data path from the sensors to the .FDR file with no manual steps. Lead(s): Marvin, Rex.

**5.3 User Experience**

*Pi/sensor housing*

Make a box to contain and protect the Pi and the sensors (except cameras). Ports on the Pi should be accessible, and the box should withstand the operating conditions listed in the requirements. Deliverable: a physical box that houses the Pi and attached sensors. Lead(s): Pai.

*Pi/sensor mounting*

Create a system to securely mount the Pi/sensor box in the plane. It should be able to connect to the plane’s power source. Deliverable: the Pi/sensor box will not bounce around the inside of the plane during flight. Lead(s): Marvin.

*Camera mounting*

Create a system to securely mount the cameras inside the plane. The cameras should be able to connect to the Pi/sensor box. Deliverable: the cameras will not bounce around the inside of the plane during flight. Lead(s): Pai, Marvin.

*Create user-friendly interface*

Write a procedure, and create hardware and code to execute it, that allows the user to easily install and calibrate the project in their plane. This will include a method to calibrate the extreme positions of the cables for image analysis, and a method to transfer the .FDR file off of the Pi. Deliverable: successful test of a procedure that calibrates the data collection system such that data is collected correctly, and transfers the .FDR file. Lead(s): Darcy, Allen.

**5.4 Testing**

*Test flight 1*

Test our project in the client’s plane (client willing) to try and find previously unseen problems. Deliverable: creation and execution of an in-situ test of how our project performed, along with identification of any problems. Lead(s): everyone must lead for their own portion of the project.

*Feedback implementation*

Fix any problems that we found from the test flight. Deliverable: all problems identified from the test flight are remedied. Lead(s): everyone must lead for their own portion of the project.

**5.5 Client Hand-off**

*User manual*

Write a manual explaining exactly how the average aerobatic pilot can install the system into their plane. Deliverable: a document given to the client such that the client should be able to install this project without our help. Lead(s): everyone must lead for their own portion of the project.

*Installation*

Install the system into the client’s plane. Deliverable: fully functional product according to the engineering requirements, installed in the client’s plane. Lead(s): everyone must lead for their own portion of the project.

*Finalization*

This is a period of time at the end of the semester to finish up anything remaining for the class.

# Budget Estimate (Wenjun Ma)

| **Item** | **Description** | **Cost** |
| --- | --- | --- |
| 1 | BerryGPS-IMU GPS and 10DOF for The Raspberry Pi\*2 | $71.20\*2 |
| 2 | Raspberry Pi 4 4GB starter kit | $109.95 |
| 3 | GPS antenna (AA.162.301111) | $12.39 |
| 4 | MakerFocus Raspberry Pi4 Camera IR, Raspberry Pi4 Camera Night Vision Fisheye Camera 5mp OV5647 | $34.15 |
| 5 | PRT-12794 female to male wire | $2 |
| 6 | JBtek BMP180 Barometric Pressure, Temperature and Altitude Sensor | $6.99 |
| 7 | IMU 9DOF MPU-9250 | $13.88 |
|  | Total Cost | $321.76 |

The main budget implicant we encountered is that we wasted some budget during the testing time. Our system functionally consists of 3 parts on collecting data: IMU, GPS, and camera modules. Item 6 and 7 are the parts which we first selected and bought for the system IMU part. However, these components need to be soldered to work properly. Then, we chose item1 which includes both the GPS and the IMU module for reducing the required system space and convenience. The reason why we bought two BerryGPS-IMU GPS and 10DOF for The Raspberry is that we accidentally burned one component when we first tried to get some data from the BerryGPS-IMU GPS and 10DOF for The Raspberry Pi.

# Attachments

# Appendix 1 – Engineering Requirements

Team # 9   
Team Name: Aerobatic Black Box

Project Name: Aerobatic Black Box

| **Requirement** | **Value, range, tolerance, units** |
| --- | --- |
| Data Collection | Measure and record the position (latitude, longitude, and altitude) and attitude (pitch, roll, and yaw) of the aircraft |
|  | Measure and record the positions of the flight stick and rudder pedals |
|  | Measure and record the value of the airspeed indicator dial |
|  | Record each of the above data points at a frequency of at least 20Hz |
|  | Record a timestamp alongside each of the above data points |
| Operating Environment | -40°C to 50°C in storage  0°C to 40°C in operation |
|  | Continue to collect data according to the Measurement Requirements and remain securely attached to the interior of the plane while under a g-force of at least 15G |
|  | Survive the vibrations of an aerobatic plane in flight – to be tested in a Citabria 7ECA |
|  | Operate for at least 30 minutes during flight without any interaction from the pilot |
|  | Powered either by a 12V bus or by battery |
|  | Not modify any part of the aircraft to comply with FAA regulations |
| Visualization System | Provide a 3D representation of an aerobatic flight reconstructed from the flight data recorded by the data collection system |
|  | Allow the user to view the flight from any camera angle |
|  | Play at 20Hz or more |

# Appendix 2 – Gantt Chart

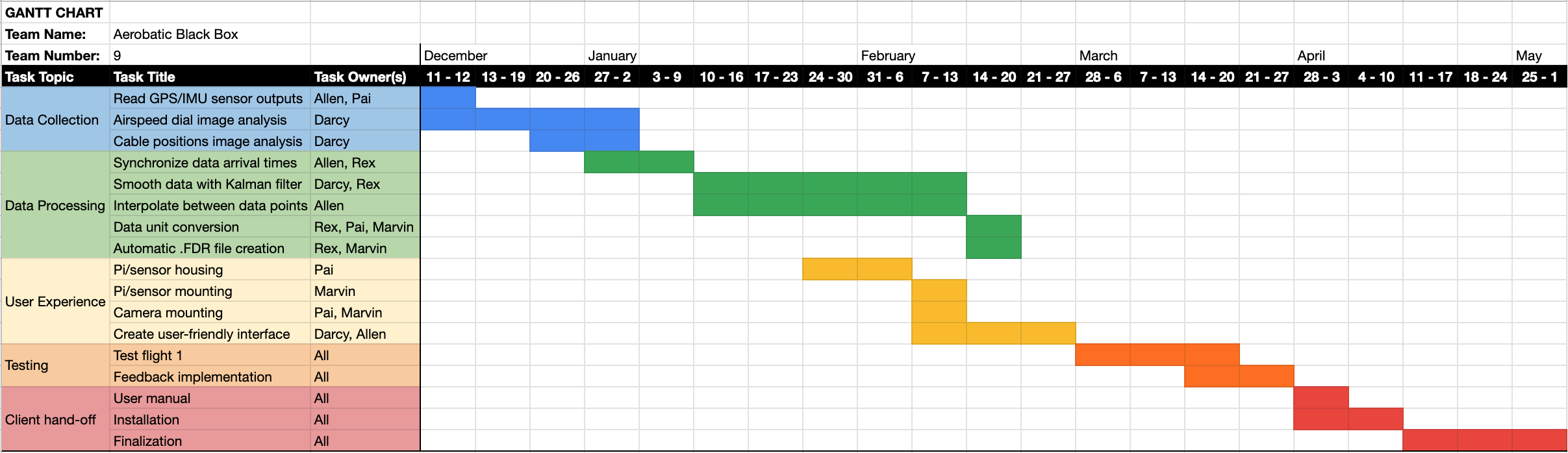


Figure 7.2.1: Gantt Chart of Team 9’s schedule for the rest of the school year.

# Appendix 3 – Other Appendices

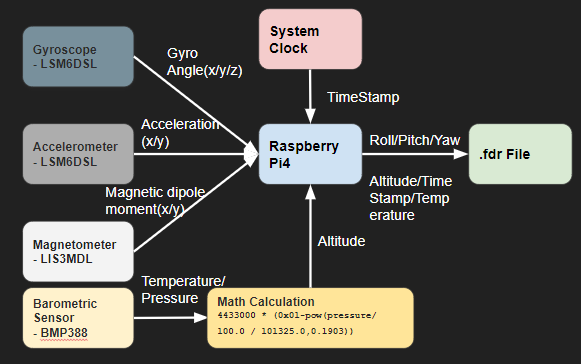


Figure 7.3.1: The schematic of the IMU system

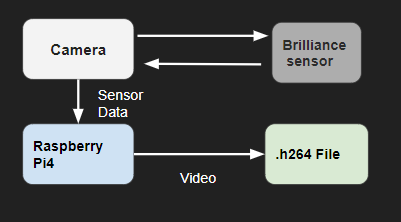


Figure 7.3.2: The schematic of the Camera System

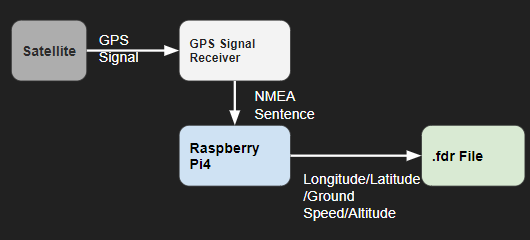


Figure 7.3.3: The schematic of the GPS system

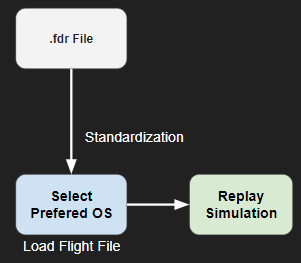


Figure 7.3.4: The schematic of the Visualization System

Table 7.3.1: Team Information

| Xinyu Liu  Phone: 6174169301  Email: [xinyubu@bu.edu](mailto:xinyubu@bu.edu)  Bio: Student of Department of Electrical and Computer Engineering, Boston University | | | |
| --- | --- | --- | --- |
| Pai Liu  Phone: 8572600461  Email: [paiii@bu.edu](mailto:paiii@bu.edu)  Bio: Student of Department of Electrical and Computer Engineering, Boston University | | | |
| Wenjun Ma  Phone: 6173207151  Email: [wjm123@bu.edu](mailto:wjm123@bu.edu)  Bio: Student of Department of Electrical and Computer Engineering, Boston University | | | |
| Radhey K. Patel  Phone: 413453464  Email: [rexu6@bu.edu](mailto:rexu6@bu.edu)  Bio: Student of Department of Electrical and Computer Engineering, Boston University | | | |
| Darcy Meyer  Phone: 2013127510  Email: [darcym@bu.edu](mailto:darcym@bu.edu)  Bio: Student of Department of Electrical and Computer Engineering, Boston University | | | |

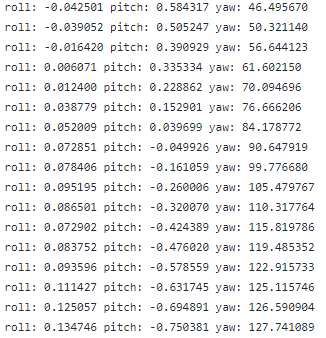


Figure 7.3.5: IMU Testing Data Results

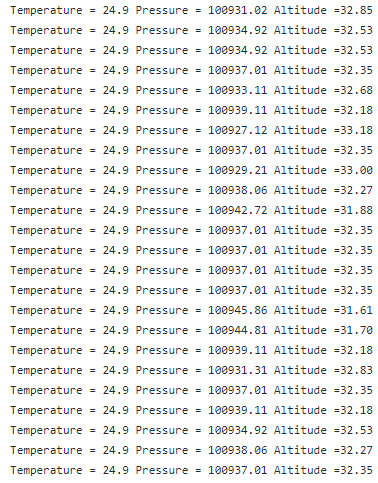


Figure 7.3.6: Pressure-Temperature-Altitude Sensor Data

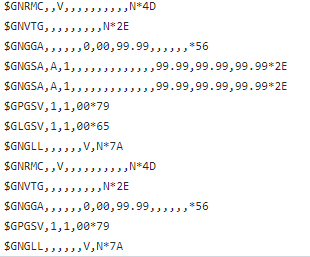


Figure 7.3.7: GPS Data from 4-Satellite systems

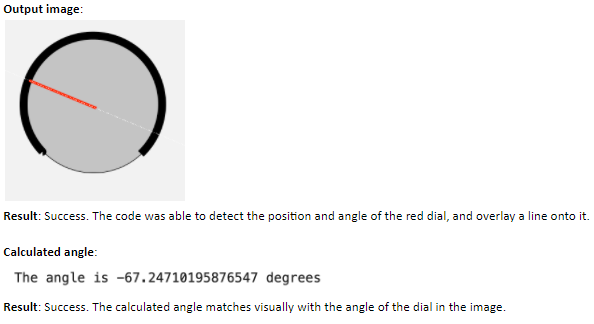


Figure 7.3.8: Instrumentation Dial Angular Image Processing