Toronto Metropolitan University

AER817 Systems Engineering

Systemic Engineering Design of a Visual Tracking Drone

Final Report

Authors: Aman Gilani, Muhammad Mohsin Mukhtar, Brijeshkumar Patel, Muhammad Abdul Rafay, Ibrahim Khan, MHD Nazem Eylji

Table of Contents

List of Figures

List of Tables

1. Introduction & Summary of the Work Done by Team Members

The Visual Tracking System was designed for activities that require a constant close-up view for the audience such as many sporting events like racing and paragliding. This system is useful for careful surveillance, making it a very marketable product due to its multiple uses. In this phase III report, the objectives and requirements are restated alongside the degree of meeting the requirements matrix in order to portray how the team was able to meet the necessary requirements for the systems. In addition, the report provides a brief system overview of the power subsystem, payload subsystem, structures & mechanisms, command & data handling, communication subsystem, and the ground station. The system interfaces between each system that show the connectors, inputs, and outputs of each system are also detailed. Finally, test results of all the systems of the Visual Tracking System are provided based on the demonstration that was conducted in the gym.

Table 1 - Work distribution for each team member

Name	Task Distribution for Phase III Report	Percentage
Muhammad Mohsin Mukhtar	Introduction, Payload Subsystem, Power Subsystem, Work Packages, Degree of Meeting Requirements Matrix, Conclusion, software code	14.28%
Aman Gilani	Sub-system Testing, TRL, N-2 Diagram, Work breakdown structure, software code, Command, and Data Handling and Communication system.	14.28%
Muhammad Abdul Rafay	Structures and mechanisms, Ground Station	14.28%
Ibrahim Khan	System Interface, BOM, report formatting.	14.28%
Brijeshkumar Patel	System Interface, Gantt Chart, report formatting.	14.28%
MHD Nazem Eylji	System Objectives, System Requirements, CAD Modeling, software code.	14.28%

2. System Objectives & Requirements

2.1 System Objectives

Mission Main Objectives

- Reading the object's GPS signal successfully
- Converting the GPS reading into specific Coordinates
- Transferring the data using Arduino to command the camera to move
- Transmit a video signal from the camera using the wifi transmitter

Mission Secondary Objectives

- Ensuring the component do not get overheated to avoid damaging the components
- Ensuring the power supply is enough to power the system for a specific duration of time
- Reading GPS signals to get an accurate location of where the drone is in case of visual loss
- Keeping the drone at a safe distance to avoid accidents or any parts from getting wet

2.2 System Requirements

2.2.1 Requirements

Payload Requirements:

- The volume of the payload shall be less than 8000 cm^3.
- The total weight of the payload shall be no more than 300 grams.
- The payload shall be able to attach to the given platform on the quadcopter.
- The total cost of the equipment including the box shall be less than $50.

Functionality:

- The GPS module shall be able to obtain GPS-specific coordinates.
- The wifi transmitter shall be able to communicate with the other object's GPS system to obtain the coordinates information and process it to compare it to the box's GPS coordinates so the camera can get an accurate location of the object.
- The wifi transmitter shall be able to send live video feed within its specified range.
- The GPS signals of the tracking device shall be captured by the visual tracking system in real-time.

Performance:

- The battery system shall be enough to power everything.

- There shall be no heating issues encountered during the demonstration.
- All components shall be securely attached to the payload without significant vibrations.
- The camera module shall have a 360 degrees rotation capability and 60 degrees pivot relative to the ground.

2.2.2 Degree of Meeting Requirements Matrix

Table 2 - Degree of Meeting Requirements Matrix

Requirement Description	Specification	Validation Method	Test Results
Payload System			
Payload weight	The mass of the payload shall not exceed 300 grams.	The length, width, and height are altered to achieve a mass of less than 300 grams.	The mass was measured to be 244 grams for the entire Visual Tracking System
Payload volume	The volume of the payload shall be at most 20 cm x 20 cm x 20 cm.	The length, width, and height are altered to achieve a volume of less than 8000 cm^3.	The payload dimensions were 15 cm x 10 cm x 5 cm. Which gives 750 cm^3.
Interface attachment	The payload subsystem shall be able to attach to the quadcopter's base interface.	The base of the quadcopter attaches firmly to the payload.	The payload was firmly attached to the quadcopter during the demonstration.
Product deadline	The final product shall be delivered on/before November 21st, 2022.	The design development process is followed using the previously made Gantt Chart.	The final project report and demonstration were completed before the deadline.
Total cost	The total cost of the final payload subsystem shall be at most the budget of $50.	The sum of the cost of the GPS receiver and camera module is approximately $31, which is less than the	The total cost was approximately $41.

		$50 budget.	
Arduino microcontroller functionality	The Arduino microcontroller shall be able to receive and store information from the GPS module and WIFI transceiver.	After the completion of the Arduino Uno code, the Visual Tracking System is to be tested in a test environment using a target object with an in-built GPS.	The Arduino microcontroller operated successfully during the demonstration.
Servo and gimbal system attachment	The servo and gimbal subsystem shall be able to attach to the payload.	The servo and gimbal subsystems are attached to the payload using an adhesive.	The servo and gimbal system was attached to the payload using super glue.
Self-contained system	The payload shall be a self-contained system	The 9 V battery is used and no power is extracted from the quadcopter.	A 9 V battery powered the system without any power extracted from the quadcopter.
Power System			
Sufficient power for operation	The power subsystem shall provide sufficient power to all the components for successful operation	All the components are functioning after connecting the battery.	All components functioned successfully.
9 V battery	A 9 V power supply shall be used for the final product.	The battery works at its full capacity after checking with a voltmeter.	A 9 Volt battery was utilized for the demonstration.
Power supply for GPS receiver	The power supply for the GPS receiver shall be at 40 mA and 3.3 V.	The 3.3 V pin from the Arduino microcontroller is connected to the GPS module	The 3.3 V pin from the Arduino microcontroller was connected to the GPS module.
Electrical grounding	The power subsystem shall have an electrical grounding	A wire from the Arduino board is connected to the ground on the	A wire from the Arduino board was connected to the ground on the

	to protect the components from the accumulation of static electric charges and high voltages.	Arduino microcontroller.	Arduino microcontroller
Functions properly for the duration of the demonstration	The power subsystem shall provide adequate power to the Arduino microcontroller for the duration of the demonstration.	The battery works for about 10 minutes in the test environment.	The battery powered the systems successfully throughout the demonstration
Communication System			
GPS Satellite lock.	The GPS receiver shall be able to get a lock on to at least 4 satellites.	After cold starting the GPS module, the GPS automatically searches for satellites for a signal fix.	The GPS successfully received information for the latitude and longitude of the target. Therefore, it was successfully able to connect to the satellites.
GPS data update speed.	The GPS receiver shall be able to update its position at least every 0.5 seconds.	The GPS module will communicate the quadcopter GPS data to the Arduino with a 0.5 seconds delay.	A delay of 0.5 seconds was done in the Arduino code with a delay function.
GPS latitude accuracy.	The GPS receiver shall have a horizontal positioning accuracy of less than 2 m.	The GPS module bought has a latitude accuracy of less than 2 m.	The latitude and longitude information received had an accuracy of less than 2 m.
GPS data transfer rate.	The GPS receiver shall have a time transfer accuracy of 30 ns.	The GPS module bought has a data transfer rate of 30 ns.	Since the data was received rapidly from the GPS, this requirement was also met.
Command and Data Handling			

GPS data memory allocation.	The flight software shall be able to save the GPS data in the allotted memories onboard the Arduino Microcontroller.	The data received from the GPS is saved in the respective variables in the Arduino RAM.	The GPS data was saved in the variable assigned for it in the Arduino Uno program.
Uplink and Downlink Speeds.	The flight software shall have fast data uplink and downlink speeds.	The flight software is programmed with faster baud rates for uplink and downlink speeds.	The baud rate was 11500.
Efficient programming methods.	The flight computer shall follow efficient programming methods.	The flight program will use multiple loops to perform separate tasks instead of one loop. The program uses safety partitioning methods to limit failures.	Multiple loops were made in the program to make the code simple.
Sensors communication.	The flight computer shall be able to communicate with other sensors for data acquisition.	The flight computer is programmed to receive data from all the sensors in a loop and in real-time.	The Arduino microcontroller was able to communicate with the GPS receiver.
Software and hardware redundancies.	The flight computer shall have redundancies in case of sensor failure.	The program is coded with software redundancies and safety partitioning.	Multiple versions were made of the program for redundancy.
Body: Structures and Mechanisms			
Body structure volume.	The body structure shall not exceed 8000 cm^3 volume.	The designed payload is compact and uses optimal shapes.	The volume was 750 cm^3.
Body structure mass.	The body structure mass shall not exceed 150g.	The material used is lightweight along with optimal structure volume.	The body weight of the box was 144 grams.

Body structure interfaces with avionics.	The payload shall be securely attached to an internal plate using adhesive.	The adhesive used is strong enough to securely attach the avionics. The avionics are reinforced with screws.	The superglue was strong enough to securely attach the avionics systems.
Quadcopter interface with payload.	The body structure shall interface with the quadcopter.	The base interface for the quadcopter securely attaches to the payload.	The quadcopter is securely attached to the payload using the drone attachment disk.
Ground Station			
Data acquisition and storage.	Data shall be received and stored by the Arduino Microprocessor.	The latitude and longitude variables in the flight computer program store the data received from the GPS module and camera module	The data was received and stored in the Arduino microprocessor. However, the camera video feed was not received.
GPS data processing.	Data received by the GPS modules shall be processed in real-time.	The data received in the flight software.	The data from the GPS module was processed in real-time.
Commands to servo and gimbal subsystem.	The ground station shall command the servo and gimbal subsystem with rotation and pitch sequences.	The connection between the flight computer and the servo and gimbal subsystem is stable.	The connection between the Arduino microcontroller and the servo and gimbal subsystem was stable and superglue was used to attach them to the payload in a secure manner.

2.3 Mass and Power Budget

2.3.1 Mass Budget

Table 3 - Mass budget

Component	Mass (g)	Cost ($)
Arduino UNO	25	Included
Camera Module	9	16.99 [1]
Servo Motor (x2)	18	Included
GPS Module	10	19.5 [2]
Transceiver (x2)	10	Included
Battery Pack	34	5
3D PLA Print Shell	50 (approx)	Included
Total	**156**	**41.49**

2.3.2 Power Budget

Table 4 - Power budget

Component	Voltage	Power
Arduino UNO	5 V	160 mW
Camera Module	3.3 V	150-200 mW
Servo Motor (x2)	4.8 V	275 mW
GPS Module	3.3 V	132 mW
Transceiver (x2)	3.3 V	100 mW
Battery Pack	9 V	N/A
3D PLA Print Shell	N/A	N/A
Total	**9 V**	**867 mW**

2.4 Technology Readiness Level

The Technological Readiness Level is a method that helps in estimating the maturity of the technology or the product developed during the acquisition phase of the product life cycle. The life cycle of a product begins from its initial concept design and ends when the product is launched in the market. Throughout its life cycle, the design of the product goes through various stages of design, development, and testing.

The following are the Technological Readiness Levels and their Descriptions:
Level 1: Basic principle observed and reported.
Level 2: Technology concept and/or application formulated.
Level 3: Analytical and experimental critical function and/or characteristic proof-of-concept.
Level 4: Component and/or breadboard validation in the laboratory environment.
Level 5: Component and/or breadboard validation in the relevant environment.
Level 6: System/subsystem model or prototype demonstration in a relevant environment (ground or space).
Level 7: System/subsystem model or prototype demonstration in a target/space environment.
Level 8: Actual system completed and "flight qualified" through test and demonstration (ground or flight).
Level 9: Actual system "flight proven" through successful mission operation

By the end of this project, the developed prototype of the visual tracking system was aimed to achieve the TRL of level 6. This was achieved by demonstrating the prototype (system/subsystem) in the relevant environment. The prototype was able to show functions regarding GPS data acquisition and servo motion (rotation and pitch). However, during the flight demonstration, the camera module wasn't able to transmit the live video feed due to the wifi's unavailability. Therefore, the technology readiness assessment failed to qualify the prototype since the sensor package did not achieve all of its targeted mission objectives.

3. Project Gantt Chart, Work Breakdown Structure, and Work Packages

3.1 Project Gantt Chart

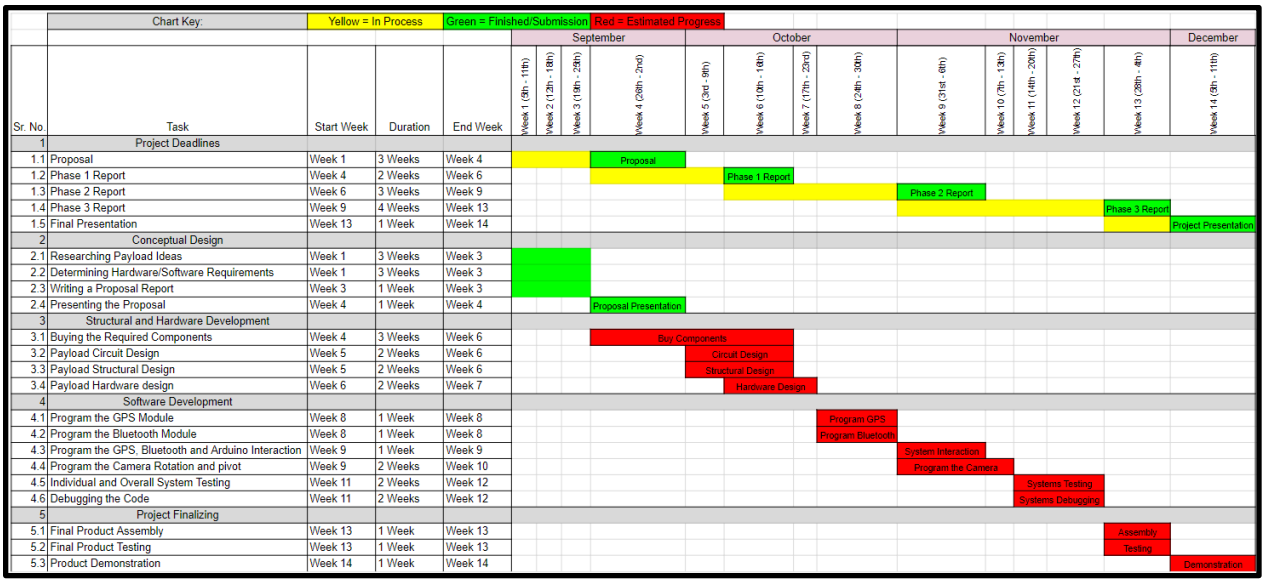

Figure 1 - Gantt chart

3.2 Work Breakdown Structure

The figure below illustrates the project phases and their subsequent work breakdown structure. The completion of the project is divided into three following phases: conceptual design, structure and hardware development, and software development. Then each phase is divided into subtasks adopting a top-down approach for the project. The approach is also an object-oriented one that focuses on the requirements and objectives stated in the project description.

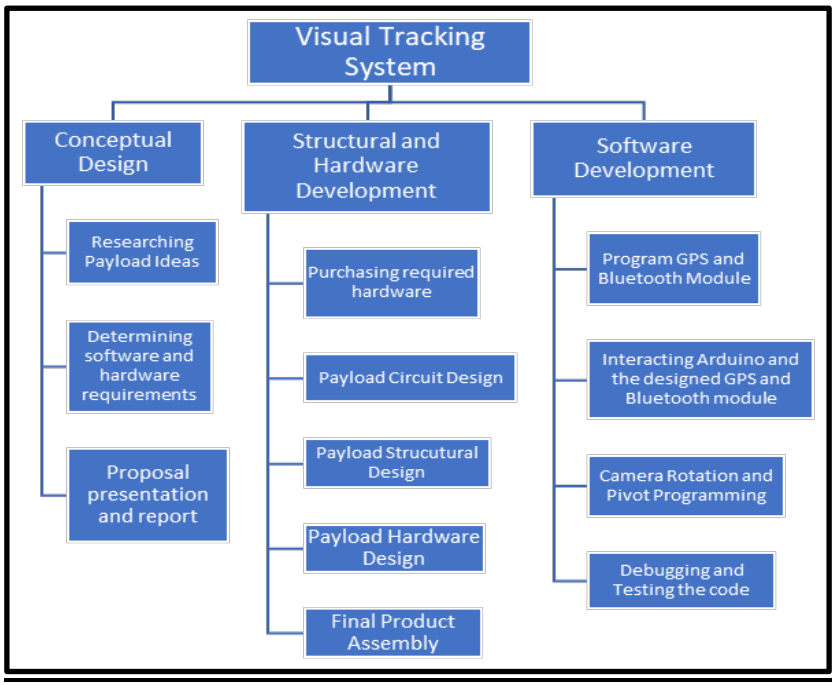

Figure 2 - Product Design Phases Breakdown Structure

The figure below shows the work breakdown structure for the project. The project is divided into various phases and each phase is divided into subsequent tasks. Each task is defined as a simple internal milestone that is expected to be completed as per the summary of deliverables table. The figure represents a flow chart to better understand and present the process for the course of this project. The following figures (*figure 3 and figure 4*) illustrate the detailed work breakdown structure for WBS 2000 and 4000 respectively.

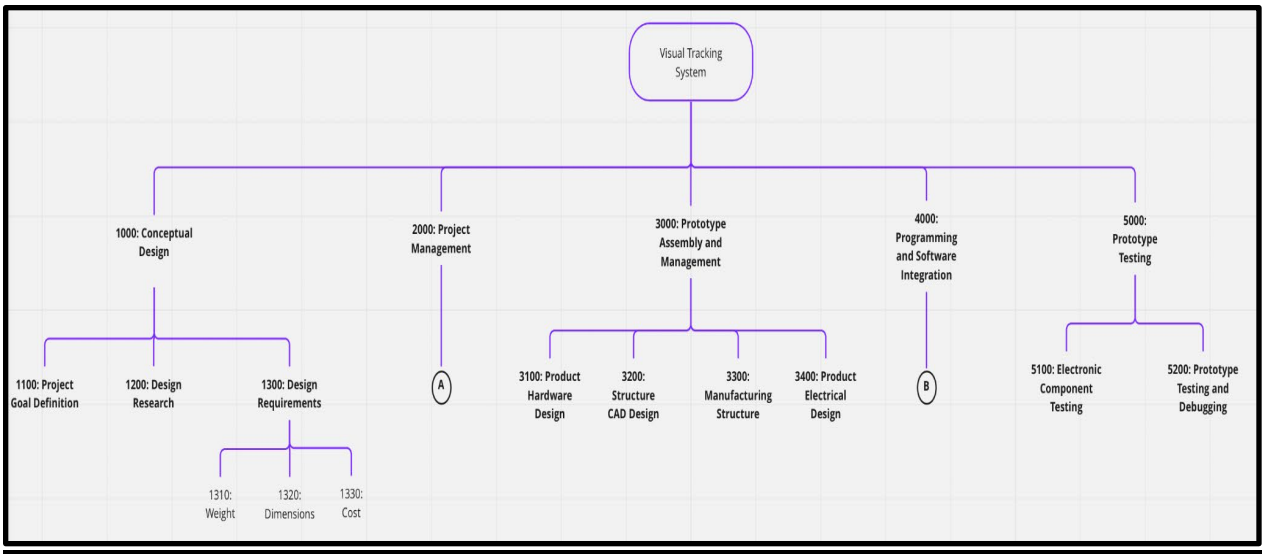

Figure 3 - Work Breakdown Structure

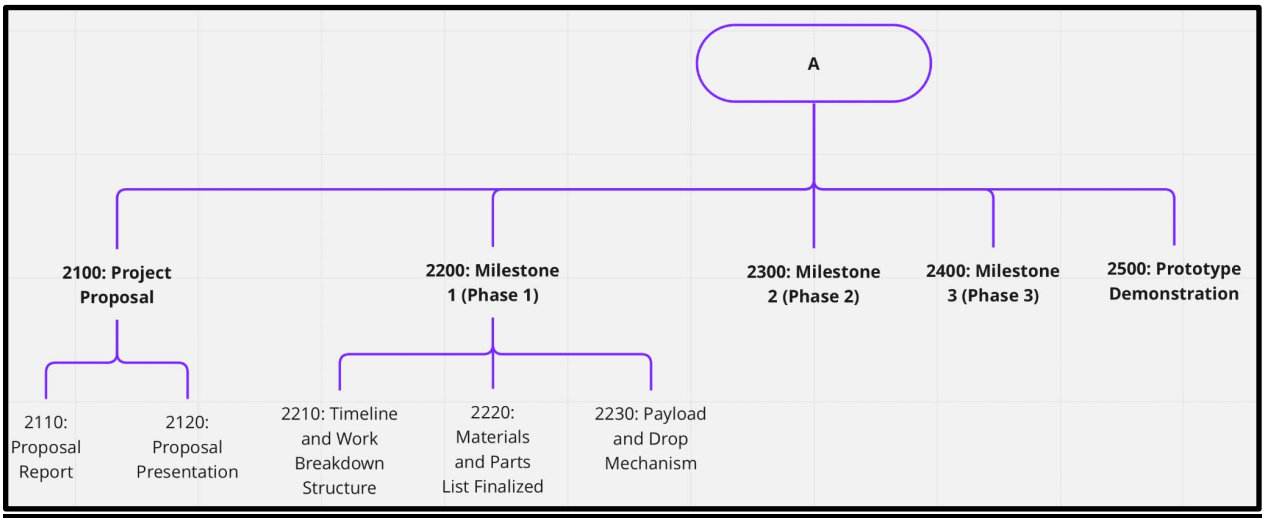

Figure 4 - Detailed Work Breakdown Structure for WBS 2000

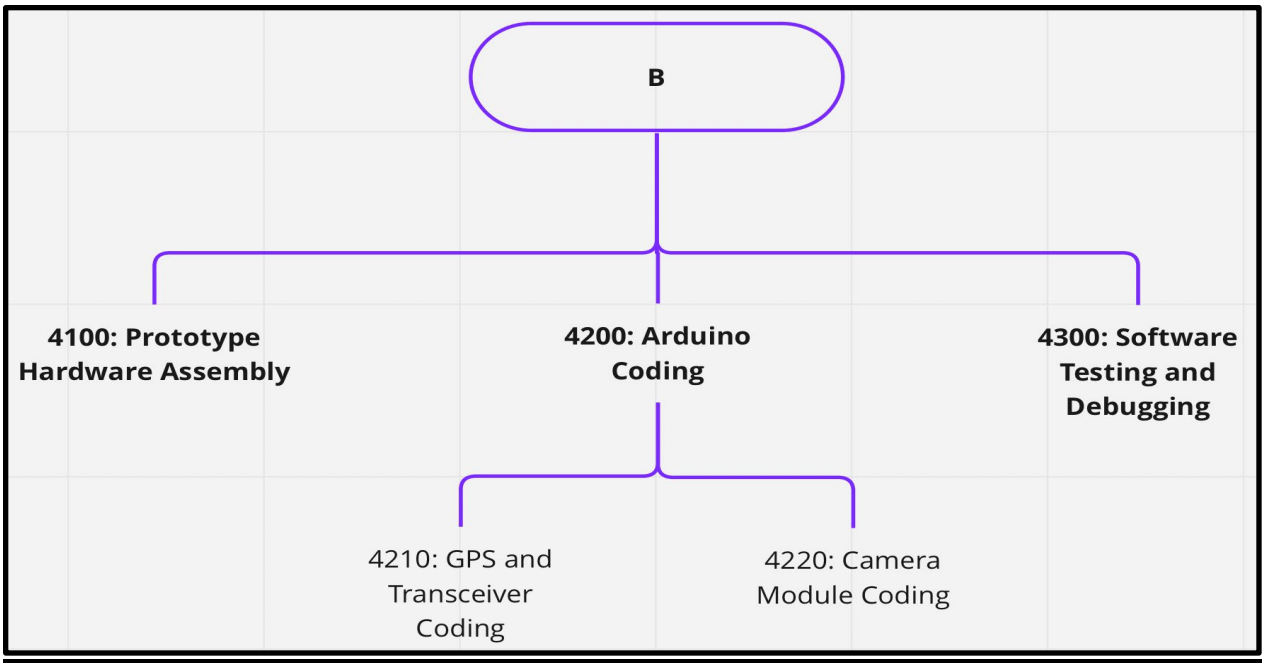

Figure 5 - Detailed Work Breakdown Structure for WBS 4000

3.3 Work Packages

Based on the work breakdown structure, the following work packages can be constructed for each phase of the project. The work packages demonstrate the objectives, tasks, inputs, and outputs of each phase. This allows the Systems Engineering team to systematically develop the final product.

Table 5 - Work Package Definition - WBS #1000

Project: Visual Tracking System (VTS)			
Work Pack Title:	**Conceptual Design**	**WBS Ref: 1000**	
Sheet: 1 of 1			
Scheduled start: September 19,2022		Accountable Manager:	Muhammad Mohsin Mukhtar
Scheduled end: October 10, 2022		Resources	

Estimated effort: 10 hours		Aman Gilani, Ibrahim Khan, Muhammad Abdul Rafay, Brijeshkumar Patel, MHD Nazem Eylji, Muhammad Mohsin Mukhtar

Objectives:

- Identify the primary and secondary design requirements
- List the System requirements specifications
- Ensure all the design requirements stated in the design definition are satisfied
- Research pre-existing designs
- Finalize all the components of the visual tracking system

Inputs:

- From Dr. K. Kumar
- From Cameron Paul (AER817 Section 2 Teaching Assistant)
- From team members
- From research articles on similar projects

Tasks:

- Conduct research of pre-existing designs and list the advantages and disadvantages of each design to choose the best feasible design
- Research the camera module, GPS module, and Wifi transceiver that are compatible with Arduino IDE and cost-effective.
- Determine the dimensions of the payload
- Make a rough sketch of the final product

Outputs/Deliverables:

- The primary and secondary design requirements
- The system requirements specifications
- Compatible camera module, GPS module, and Wifi transceiver
- Rough sketch of the final product

Table 6 - Work Package Definition - WBS #2000

<table>
<tr><td colspan="4">Project: Visual Tracking System (VTS)</td></tr>
<tr><td colspan="2">Work Pack Title:</td><td>Project Management</td><td>WBS Ref: 2000</td></tr>
<tr><td colspan="2">Sheet: 1 of 1</td><td colspan="2"></td></tr>
<tr><td>Scheduled start:</td><td>September 19,2022</td><td>Accountable Manager:</td><td>Muhammad Mohsin Mukhtar</td></tr>
<tr><td>Scheduled end:</td><td>December 6, 2022</td><td>Resources</td><td rowspan="2">Aman Gilani, Ibrahim Khan, Muhammad Abdul Rafay, Brijeshkumar Patel, MHD Nazem Eylji, Muhammad Mohsin Mukhtar</td></tr>
<tr><td>Estimated effort:</td><td>75 hours</td><td></td></tr>
<tr><td colspan="4"><u>Objectives:</u>
● Confirm with all group members the weekly meeting time
● Communicate the project plan with all team members
● Ensure all the project deliverables such as the project reports and presentations are completed according to the project timeline while fulfilling all the requirements</td></tr>
<tr><td colspan="4"><u>Inputs:</u>
● From Dr. K. Kumar
● From Cameron Paul (AER817 Section 2 Teaching Assistant)
● From team members
● From research articles on similar projects
● From AER817 course notes</td></tr>
<tr><td colspan="4"><u>Tasks:</u>
● Discuss with each group member to come up with a meeting time suitable for all members
● Hold a meeting to come up with a feasible project schedule
● Complete all the project reports and presentations in an efficient manner by the date indicated for each in the project timeline</td></tr>
<tr><td colspan="4"><u>Outputs/Deliverables:</u>
● Meeting time schedule
● Project timeline
● Project reports and presentations</td></tr>
</table>

Table 7 - Work Package Definition - WBS #3000

<table>
<tr><td colspan="4">Project: Visual Tracking System (VTS)</td></tr>
<tr><td colspan="2">Work Pack Title:</td><td>Prototype Assembly and Manufacturing</td><td>WBS Ref: 3000</td></tr>
<tr><td colspan="2">Sheet: 1 of 1</td><td colspan="2"></td></tr>
<tr><td>Scheduled start:</td><td>September 19,2022</td><td>Accountable Manager:</td><td>Muhammad Mohsin Mukhtar</td></tr>
<tr><td>Scheduled end:</td><td>November 10, 2022</td><td rowspan="2">Resources</td><td rowspan="2">Aman Gilani, Ibrahim Khan, Muhammad Abdul Rafay, Brijeshkumar Patel, MHD Nazem Eylji, Muhammad Mohsin Mukhtar</td></tr>
<tr><td>Estimated effort:</td><td>25 hours</td></tr>
<tr><td colspan="4"><u>Objectives:</u>
● Purchase the required hardware
● Display a CAD model of the final assembly
● Complete the circuit design using the provided Arduino kit and the purchased components
● Complete the payload structural design
● Produce the final product assembly</td></tr>
<tr><td colspan="4"><u>Inputs:</u>
● From team members
● From research articles on similar projects
● From AER817 course notes
● From Arduino hardware guides
● From assembly guides</td></tr>
<tr><td colspan="4"><u>Tasks:</u>
● Create a CAD model of the Visual Tracking System on either CATIA or Solidworks
● Connect the GPS module, camera module, and the Wifi transceiver in the circuit using the provided Arduino kit
● Obtain the 3D printed model of the structure using the final geometry
● Produce the final assembly of Visual Tracking System</td></tr>
<tr><td colspan="4"><u>Outputs/Deliverables:</u></td></tr>
</table>

● End product detailed design ● Circuit design ● 3D printed model ● Final assembly

Table 8 - Work Package Definition - WBS #4000

<table>
<tr><td colspan="4">Project: Visual Tracking System (VTS)</td></tr>
<tr><td colspan="2">Work Pack Title:</td><td>Programming and Software Integration</td><td>WBS Ref: 4000</td></tr>
<tr><td colspan="2">Sheet: 1 of 1</td><td colspan="2"></td></tr>
<tr><td colspan="2">Scheduled start: October 10, 2022</td><td>Accountable Manager:</td><td>Muhammad Mohsin Mukhtar</td></tr>
<tr><td colspan="2">Scheduled end: November 15, 2022</td><td rowspan="2">Resources</td><td rowspan="2">Aman Gilani, Ibrahim Khan, Muhammad Abdul Rafay, Brijeshkumar Patel, MHD Nazem Eylji, Muhammad Mohsin Mukhtar</td></tr>
<tr><td colspan="2">Estimated effort: 15 hours</td></tr>
<tr><td colspan="4"><u>Objectives:</u>

● Learn the Arduino programming language
● Complete the final code of Visual Tracking System
● Testing procedure</td></tr>
<tr><td colspan="4"><u>Inputs:</u>

● From team members
● From Arduino coding guides
● From AER817 course notes</td></tr>
<tr><td colspan="4"><u>Tasks:</u>

● Using the Arduino coding guides, familiarize with the Arduino programming language
● Type the code for the Visual Tracking System such that the horizontal and vertical distance can be determined between the quadcopter and target</td></tr>
</table>

Outputs/Deliverables:
● Final typed code on Arduino IDE

Table 9 - Work Package Definition - WBS #5000

Project: Visual Tracking System (VTS)			
Work Pack Title:	**Prototype Testing**		**WBS Ref: 5000**
Sheet: 1 of 1			
Scheduled start: November 15, 2022,		Accountable Manager:	Muhammad Mohsin Mukhtar
Scheduled end: November 25, 2022,		Resources	Aman Gilani, Ibrahim Khan, Muhammad Abdul Rafay, Brijeshkumar Patel, MHD Nazem Eylji, Muhammad Mohsin Mukhtar
Estimated effort: 5 hours			
Objectives: ● Test the Visual Tracking System			
Inputs: ● From team members ● Product prototype			
Tasks: ● Attach the final product to the quadcopter and determine whether the Visual Tracking System can track the target			
Outputs/Deliverables: ● Fully-functional Visual Tracking System			

4. System Overview

4.1 Payload Subsystem

4.1.1 Requirements

The following requirements have been stated in the design definition for the payload subsystem:

1. The mass of the payload shall not exceed 300 grams.
2. The volume of the payload shall be at most 20 cm x 20 cm x 20 cm.
3. The payload subsystem shall be able to attach to the quadcopter's base interface.
4. The final product shall be delivered on/before November 21st, 2022.
5. The total cost of the final payload subsystem shall be at most the budget of $50.
6. The Arduino microcontroller shall be able to receive and store information from the GPS module and WIFI transceiver.
7. The servo and gimbal subsystem shall be able to attach to the payload.
8. The payload shall be a self-contained system.

4.1.2 Design

The payload incorporates the Arduino microcontroller, camera module, GPS module, WIFI transceiver, and servo and gimbal subsystem. All these individual components are assembled in the payload such that the camera has a clear vision of the surroundings.

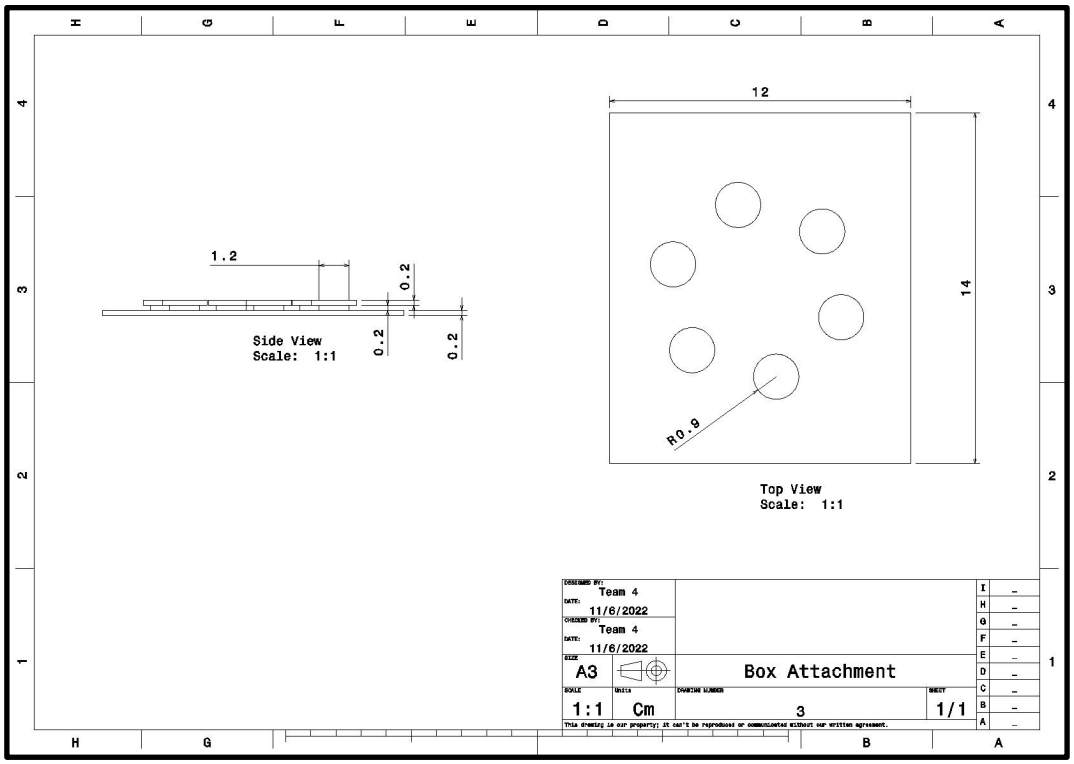

Figure 6 - Drawing for the Quadcopter Disk Attachment

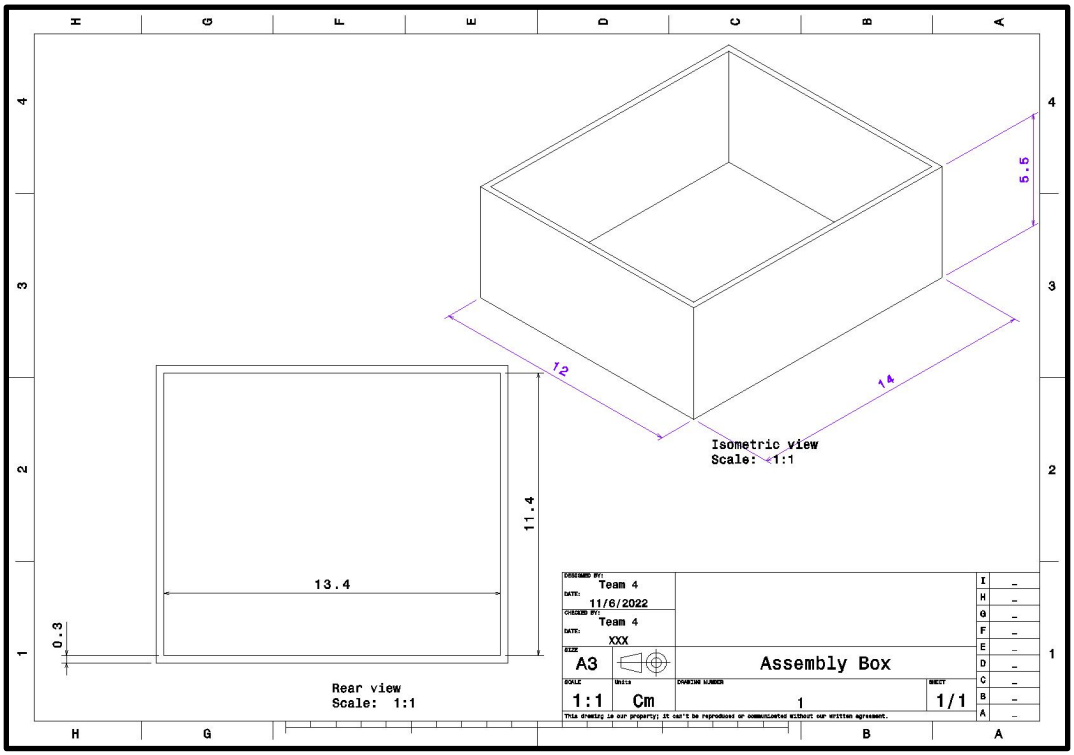

Figure 7 - Drawing for the Payload Structural Case

4.1.2.1 Hardware

The block diagram shown below in figure 6 portrays how each component interacts with the Arduino microcontroller in the payload. The power is supplied to the Arduino microcontroller by a 9 V battery. The GPS sensor, WIFI transceiver, servo and gimbal system, and camera are connected to the Arduino microcontroller which serves as an onboard computer. Utilizing the GPS sensor, the Arduino microcontroller will receive the exact coordinates of the reference object in terms of latitude and longitude. Since the payload will be attached to the quadcopter, the reference object is taken to be the quadcopter itself. Using the WIFI transceiver, the Arduino microcontroller will receive the exact coordinates in terms of latitude and longitude of the target object. Once the two sets of coordinates have been obtained, the servo and gimbal system will rotate and/or pitch the camera module as necessary to lock on to the target object. The servo and gimbal system is responsible for the rotation and pitching of the camera. The servo and gimbal subsystem consists of two servo motors. The servo at the top is responsible for rotating the gimbal. The servo at the side is responsible for pitching the camera.

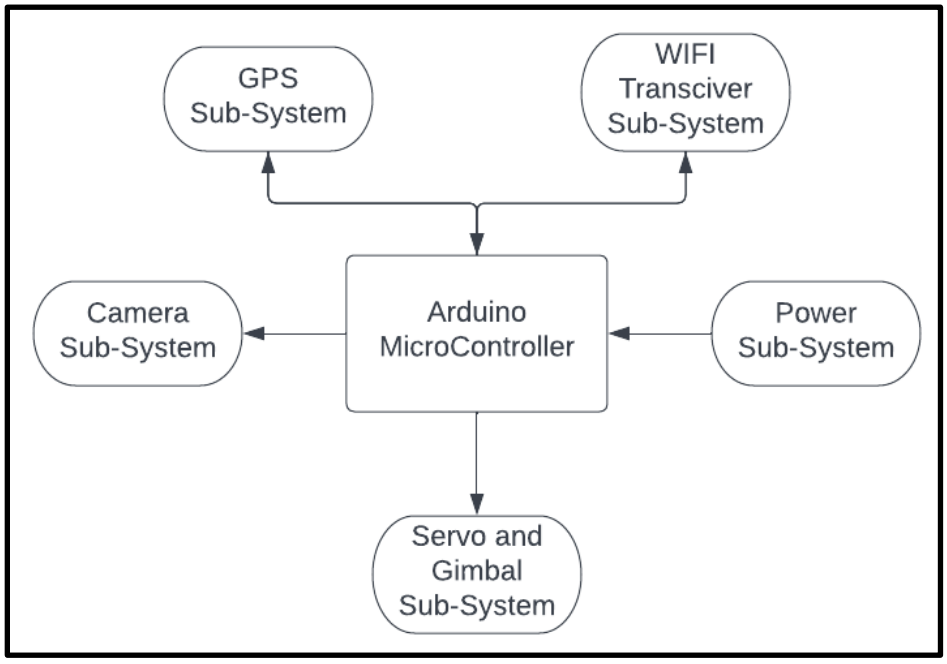

Figure 8 - Block diagram of payload subsystem

The bill of materials (BOM) of the payload subsystem is shown in the table below. The Arduino microcontroller, servo motors, and WIFI transceiver were obtained from Toronto Metropolitan university.

4.1.2.2 Software

The software of the payload subsystem is responsible for allowing the camera module to lock onto the target object once the reference and target locations are known. The camera module has to be programmed to rotate and pitch to point toward the target after the coordinates of the target object have been processed in the Arduino microcontroller.

Sub-function 4: Camera Module and Servo and Gimbal System

Calibrate the camera to point north during the initiation

Declare an angle variable

Store angle data received from the data processing subsystem

declare servo I/O pins

void setup ()

define the servo I/O pin numbers.

Change register values from default to custom

Add required delay based on Arduino microcontroller

Configure resolution

Set color

Capture the image with the relevant number of pixels

void loop ()

Instruct the top servo to rotate at the given angle

4.2 Structures & Mechanisms

A moving or fragile piece of equipment that is enclosed and safeguarded by a solid casing is referred to as a body structure. Its main functions are to provide a protective exterior to make handling easier, to provide attachment points for internal mechanisms, to maintain the cleanliness of the contents by shielding them from fouling contaminants, and to protect interior mechanisms from structural stress and potential damage from the environment. In our design, the payload will be secured by a cuboid-shaped enclosure that will be 3D printed using ABSplus-P430 thermoplastic.

4.2.1 Requirements

The main function of the body structure is to securely hold the payload and attach it to the drone. Additionally, there are secondary functions as well such as protecting the payload from

physical and chemical damage caused by surroundings. Therefore, for optimal performance of the sensor package, the requirements for the body structure are as follows:

1. The body structure shall not exceed 8000 cm^3 volume.
2. The body structure mass shall not exceed 100g.
3. The dimensions of the body structure shall not exceed 20 cm x 20 cm x 20 cm.
4. The payload shall be securely attached to an internal plate using glue.
5. The body structure shall interface with the quadcopter.

4.2.2 Design

The payload's block diagram, which includes the GPS module, Wi-Fi transceiver, camera module, Arduino microprocessor, servo, and gimbal subsystem, served as the basis for the design of the body structure. The body structure was developed to support the payload with the least amount of tools for easy accessibility and maintenance. The walls are 3mm thick, and numerous holes have been inserted to reduce costs and wireless transmission disruptions. Additionally, the holes allow for heat dissipation of the components during extensive use. The structure's overall measurements are 15cm x 10 cm x 5 and its volume is 750 cm^3.

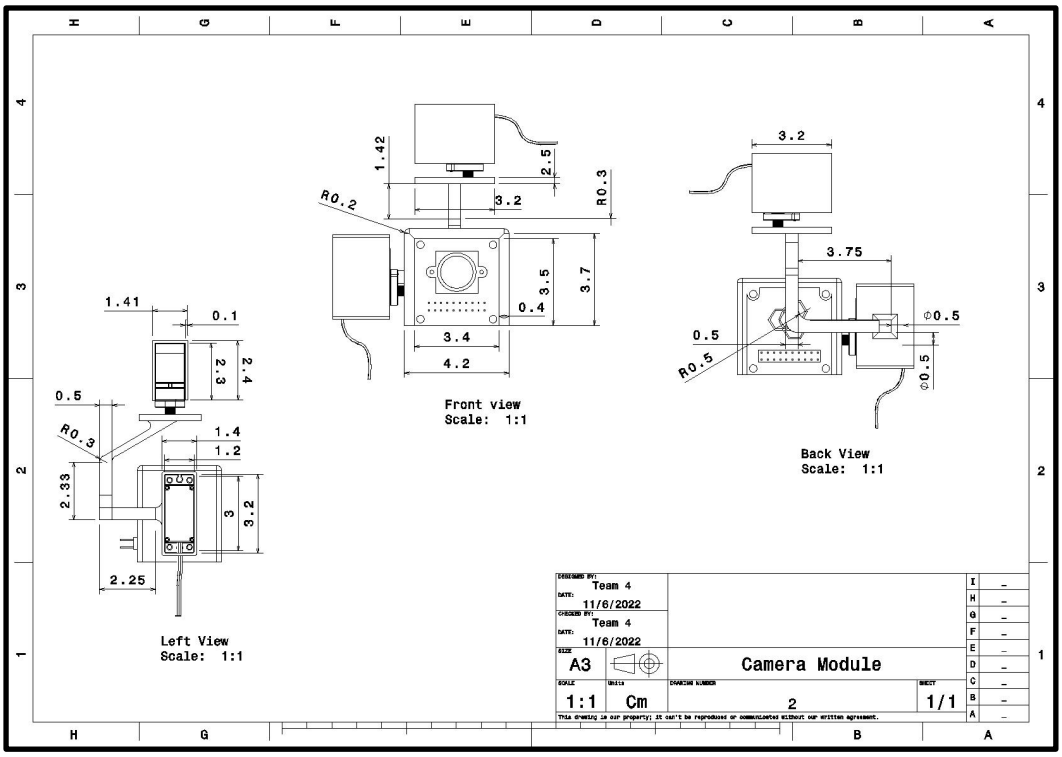

Figure 9 - Drawing for the Camera Module

4.2.2.1 Hardware

The hardware design for the structures and mechanisms ensured that all the components were able to fit inside the structural case of the payload system. The aim of the outer case was to protect the sensors and other systems inside from vibration and other factors that would disrupt the payload function. The case was designed to be lightweight and strong. The payload case was 3D printed using ABSplus-P430 thermoplastic. The gimbal mechanism was designed to achieve the camera rotation and pitch motion without transmitting the quadcopter vibrations onto the camera module.

4.3 Command & Data Handling Subsystem

The Command and Data Handling (C&DH) subsystem is essentially the brains of the visual tracking system and controls the servo and gimbal movement on the payload. The C&DH subsystem manages all forms of data received by the Arduino microcontroller. The system is responsible for computing the distance and angle between the target and the quadcopter using the haversine formula. This is achieved by receiving the GPS data of the quadcopter from the GPS sensor present inside the payload. The GPS data of the target object is obtained from the WIFI transceiver. With the calculated distance and angle, the C&DH subsystem instructs the servo and gimbal subsystem to perform necessary maneuvers. Furthermore, the C&DH subsystem autonomously monitors the position of the servo and gimbal subsystem and responds accordingly for subsequent maneuvers.

A key aspect of the C&DH subsystem is the Arduino microcontroller, which is also the flight computer for the designed payload system. The flight software on the Arduino board is responsible for all the data processing and mathematical calculations necessary for the payload to function.

4.3.1 Requirements

The objective of the Command and Data Handling subsystem is to provide the payload with maneuver sequence to various subsystems. Due to the requirements of the payload, the C&DH subsystem needs to be efficient and easy to integrate with other subsystems. The requirements for the C&DH subsystem for the designed payload are as follows:

1. The flight software shall be able to save the GPS data in the allotted memories onboard the Arduino Microcontroller.
2. The flight software shall have fast data uplink and downlink speeds.
3. The flight computer shall follow efficient programming methods.
4. The flight computer shall be able to communicate with other sensors for data acquisition.
5. The flight computer shall have redundancies in case of sensor failure.

4.3.2 Design

The proposed design for the Command and Data handling subsystem must meet the above-stated requirements to perform the payload's function. The flight software is designed to save the GPS information from the quadcopter and the target object on the Arduino's onboard memory. The flight software then uses efficient programming methods (for example: using multiple loops instead of one big loop) to calculate the distance and angle between the quadcopter and the target object. The flight software uses a baud rate of 19200 for faster uplink and downlink speeds between the Arduino and the sensors. For the designed payload, the Arduino microcontroller is treated as a ground station that carries out all the necessary data processing.

In case when the flight computer cannot receive data from a sensor or when there is a loss of power to a sensor, the flight software tags the specific sensor as failed. The flight software separates the failed subsystem from affecting the entire payload function. This method is called safety partitioning by limiting the fault of a failed system to that system. Since the data acquisition is not completed due to the failed sensor, the flight computer blinks a red LED to warn of the sensor failure. Due to the weight limitation of the payload a backup sensor is not used in case of any sensor failure. This decision was made using the trade-off study.

4.3.2.1 Hardware

The figure below illustrates a block diagram of the flight computer system. The diagram highlights the communication between the flight software and the onboard memory of the Arduino microcontroller. The flight software performs the command and data handling functions necessary for the overall payload function and performance. The Arduino microcontroller is built with three onboard memory modules called RAM, flash memory, and EEPROM. The flash memory is used to store the main operating software and other computation sequences. The RAM will be used to store the data acquired from the sensors. The EEPROM memory is used to store a backup copy of the flight software in case the main software fails or gets corrupted.

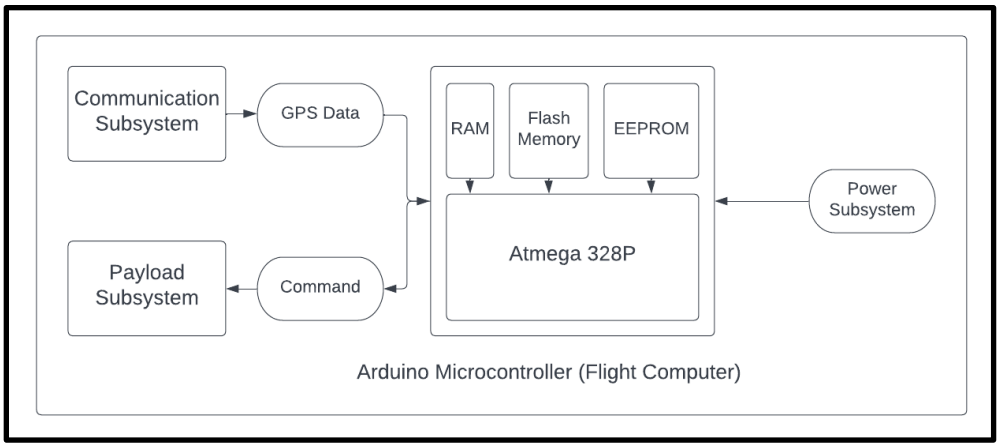

Figure 10 - Block diagram of Command and Data Handling subsystem

The table below tabulates the bill of materials for all the components used in the Command and Data Handling subsystem. The Arduino microcontroller required for the subsystem is obtained from the Toronto Metropolitan University. The Arduino microcontroller has inbuilt memory modules necessary for the project.

4.3.2.2 Software

The software design for Command and Data Handling is responsible for calculating the distance and the angle between the quadcopter and the target object. This is accomplished by using the haversine formula. The haversine formula is a very accurate and common method of calculating the distance between two points on a sphere using the latitudes and longitudes of the two points. Furthermore, the program also calculates the angle of the target object with respect to the quadcopter. This is achieved using if statements as shown in the following pseudo-code.

Sub-function 3: Data Processing

Include math library.

Declare latitude and longitudinal variables for target and reference (quadcopter).

Void loop ()

Get the longitudinal and latitude values from Sub-functions 1 and 2.

Set the latitude and longitudinal values for sub-functions 1 as reference.

Set the latitude and longitudinal values for sub-functions 2 as the target.

Use the haversine formula to calculate the distance between the target and the reference using their GPS data.

Calibrate the servo and gimbal subsystem such that the camera points North at the cold start.

If (the target location latitude is smaller than the reference latitude)

{

The target is south of the reference

Else

The target is north of the reference

}

If (the target location longitude is smaller than the reference longitude)

{

The target is east of the reference

Else

The target is west of the reference

}

Find the angle between the displacement line (line joining the target object and reference object) and the latitude line using the flat earth approximation.

4.4 Power Subsystem

The power subsystem is responsible for supplying power to the Arduino microcontroller. As outlined in the design definition, the final product has to be a self-contained system. Therefore, the Visual Tracking System has to have an internal power supply as it cannot use the power from the quadcopter.

4.4.1 Requirements

The power subsystem of the Visual Tracking System must satisfy the following requirements.

1. The power subsystem shall provide sufficient power to all the components for successful operation
2. A 9 V power supply shall be used for the final product.
3. The power supply for the GPS receiver shall be at 40 mA and 3.3 V.
   The working voltage of the GPS receiver shall be 3.3 V.
4. The power subsystem shall have an electrical grounding to protect the components from the accumulation of static electric charges and high voltages.
5. The electrical signals shall not interfere with data processing and transmission.

6. The power subsystem shall provide adequate power to the Arduino microcontroller for the duration of the demonstration.

4.4.2 Design

The power subsystem shall adequately supply power to the different components of the Visual Tracking System. In order for the system to have a safe operation, the voltage supplied by the battery should not exceed the normal operating range of each individual component of the Visual Tracking System.

4.4.2.1 Hardware

The Arduino microcontroller is able to accept a power supply from a 9 V battery or USB 5 V power. However, the USB input will add additional weight to the Visual Tracking System and would be more difficult to replace when depleted compared to a battery. Since weight is one of the constraints outlined in the design definition, it is important to minimize it in our design whenever possible. Moreover, the Visual Tracking System is a self-contained system that is not monitored by a computer on the ground. This makes the USB input difficult to incorporate into the design. Therefore, a 9 V battery is the best option for the design. The 9 V battery shall be incorporated into the power subsystem using the snap-on connector clips which will then be connected to the Arduino microcontroller. This is illustrated in the block diagram in figure 7. The Arduino microcontroller will provide the necessary power to the GPS receiver, camera module, WIFI transceiver, and 2 servo motors. This is illustrated in the block diagram in figure 8.

The rated power of a 9 V battery is approximately 450 mAH. This means it can pull 450 mA for one hour The GPS Receiver GP-20U7 has a power consumption of 40 mA at 3.3 V. The working voltage for the Treedix Camera Module OV2640 is 3.3 V. The normal operating voltage range for the WIFI transceiver is between 1.9 V and 3.6 V. The two servo motors each require a voltage of 5 V. Since the GPS module and camera module require the same amount of voltage, they are connected in a parallel connection to the Arduino microcontroller. Since 3.3 V is within the normal operating voltage range of the WIFI transceiver, it can also be connected in parallel to the GPS module and camera module. In a parallel connection, the voltage between the GPS module, camera module, and the WIFI transceiver would be 3.3 V. Since the two servo motors also require the same amount of voltage, they can be connected in parallel to each other. The total voltage of the circuit of the Visual Tracking System is 8.3 V. On the Arduino microcontroller, there are 3.3 V and 5 V input pins. The GPS receiver will be connected to the 3.3 V input, and the servo motors are connected to the 5 V input pin.

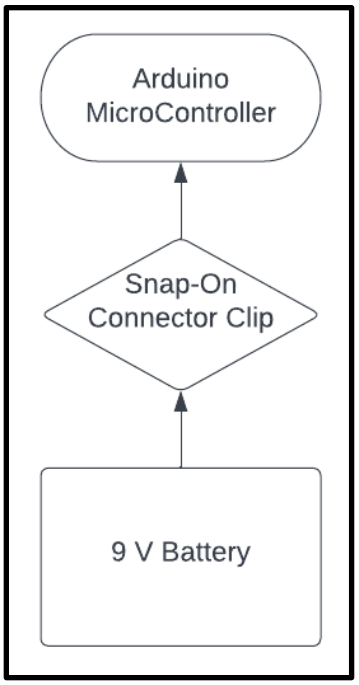

Figure 11 - Hardware connection of the battery to Arduino microcontroller

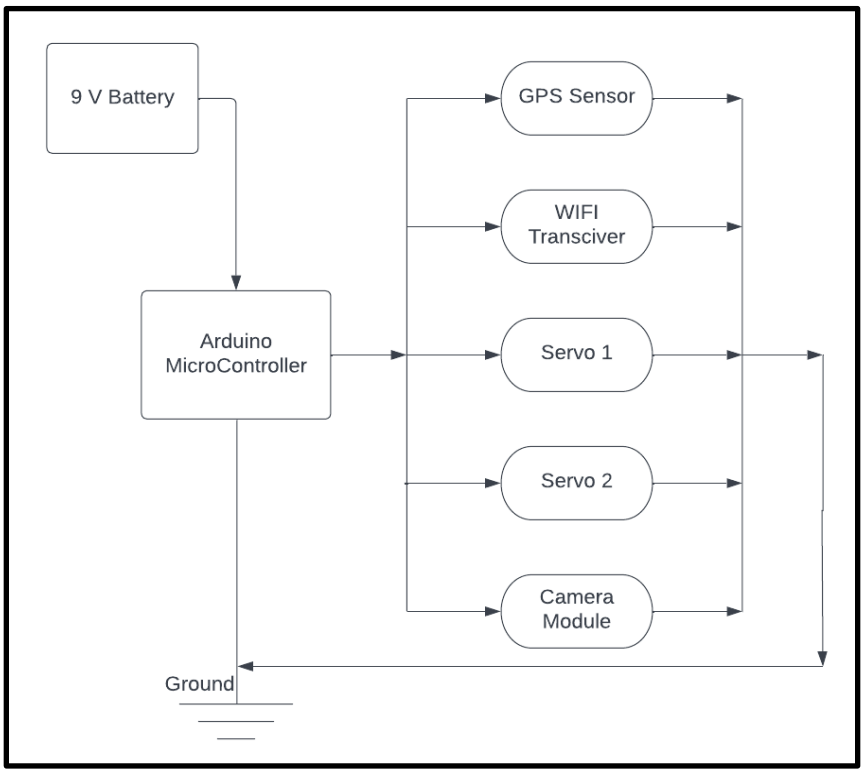

Figure 12 - Block diagram of power subsystem

The table below shows the different components of the power subsystem. The Arduino Microcontroller, 9 V battery, and snap-on connector clips were obtained from Toronto Metropolitan University. They are not included in the $50 budget for the project.

4.4.2.2 Software

For the power subsystem, it has to be ensured that the required voltage by each component is distributed and regulated by the Arduino microcontroller. This is achieved by utilizing the built-in voltage regulators in the Arduino microcontroller. The built-in voltage regulators will control the amount of power provided to each component so that the components can continue to obtain voltage within their respective normal operating voltage range.

4.5 Communication Subsystem

The communication subsystem for the designed payload consists of sensors, transmitters, and receivers that are responsible for real-time data communication to the Arduino microcontroller. The sensors/transmitters include a GPS sensor that communicates with the satellites to obtain the location of the quadcopter. The receiver includes a WIFI transceiver which acts as a data receiver for the target object's GPS coordinates.

4.5.1 Requirements

The requirements for the communication subsystem are divided into GPS communication requirements and WIFI transceiver requirements. These requirements are necessary for the payload system to work at optimal capacity.

1. The GPS receiver shall be able to get a lock on to at least 4 satellites.
2. The GPS receiver shall be able to update its position at least every 0.5 seconds.
3. The GPS receiver shall have a horizontal positioning accuracy of 2.5 m.
4. The GPS receiver shall have a time transfer accuracy of 30 ns.
5. The WIFI transceiver receiver shall have the same radio frequency set as its transmitter.
6. The WIFI transceiver shall have an open space communication range of 100 m.

4.5.2 Design

The design of the communication subsystem requires the connection of each component to the Arduino microcontroller. This design focuses on maximizing the working capacity of the GPS sensor and the WIFI transceiver to the onboard computer. The design of the communication subsystem is further divided into two categories: Hardware design and Software design. The ideal location for the GPS sensor is on the top section of the payload structure, where it can get a satellite fix and receive a clear and open signal. This is also a similar case for the WIFI transceiver receiver.

The software design focuses on optimizing the data acquisition and data processing from the two sensors in the communication subsystem.

4.5.2.1 Hardware

The figure below illustrates the communication system for the two sensors present in the payload. Upon power on, the GPS module communicates with the Arduino which initializes the satellite fix sequence. The GPS module then communicates with the overhead satellites to compute the quadcopter's position. This information is communicated to the Arduino microcontroller for processing. The Arduino is programmed to store the quadcopter's latitude, longitude, elevation, and time. The GPS module is attached to the payload structure on the top section. The GPS sensor is attached to the structure with liquid adhesive such that it has a zero degree of freedom.

The WIFI transceiver is connected to the Arduino microcontroller such that it can catch a clear signal from the target object. The figure below illustrates a block diagram to explain the connection between the sensors and the Arduino microcontroller. The WIFI transceiver receives the target object's GPS data and communicates with the microcontroller for further data processing. Similar to the GPS module, the WIFI module is attached to the payload structure with a liquid adhesive to restrain the sensor with zero degrees of freedom.

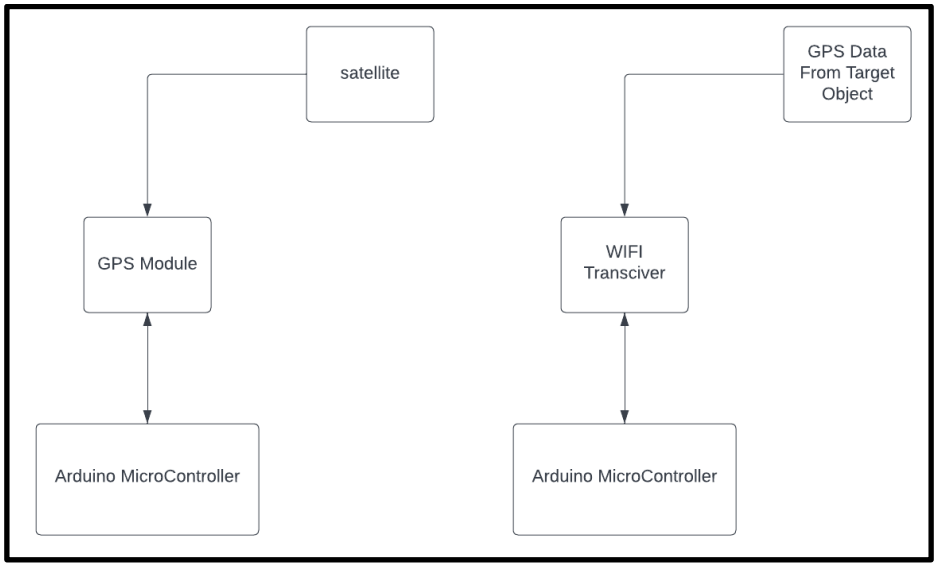

Figure 13 - Block diagram of communication subsystem

The table below tabulates the bill of materials for all the components used in the communication subsystem. The Arduino microcontroller and the WIFI Transceiver were obtained from the Toronto Metropolitan University.

4.5.2.2 Software

The software design for the communications subsystem is divided into two sub-functions: a GPS Module and a WIFI Transceiver. The following is a pseudo-code for the two sub-functions. The first sub-function is responsible for computing the GPS data for the quadcopter. The second sub-function is responsible for receiving the GPS data for the target object. The pseudocode is written for programming in the Arduino IDE software.

Sub-Function 1: GPS Module.

Include libraries required for the GPS module.

Define variables for Quadcopter latitude and longitude.

Define pin numbers on Arduino (as connected to the Arduino board) for the GPS module.

Void setup ()

    Define the baud rate for the serial monitor.

    Define the I/O pins for the GPS module.

Void loop ()

Check for GPS data. //wait for the first satellite fix and perform tests to check for the GPS data.

While (the Arduino receives information from the GPS module)

{

    If (the data encoding is successful from the GPS)

    {

        Obtain GPS position in latitude, longitude, elevation, and time.

        Save the latitude and longitudinal data in respective variables.

    }

}

Sub-Function 2: WIFI Transceiver (Signal Receiving).

Include libraries for the WIFI transceiver module.

Include libraries for SPI device communication.

Include libraries for radio communication.

Declare CE and CSN pins for the transceiver.

Declare receiver address.

Declare variables for the target object GPS (latitude and longitude) data.

Void setup ()

Initiate radio.

Setting the receiver address.

Set the power level for the transceiver.

Set the transceiver as the receiver.

Define the I/O pins for the WIFI transceiver module.

Void loop ()

If (the radio signal is available)

{

Save target object GPS (latitude and longitude) data in the respective variables.

}

4.6 Ground Station

A ground station is a terrestrial radio station designed for receiving radio waves from aeronautical radio sources or communicating with airborne devices. Ground stations may be situated either on the surface of the earth or somewhere within its atmosphere. In the case of this design, the ground station is the Arduino microprocessor itself and is responsible for commanding the camera system with necessary instructions to always keep a direct line of sight of the tracked object. Radio waves in high-frequency bands are typically used by ground stations for communication, and a telecommunication link is established when radio waves are successfully transmitted by a ground station.

4.6.1 Requirements

The primary function of the ground station is to perform mathematical calculations with the data received by the GPS modules to provide appropriate instructions to the servo and motor to control the camera. As a result, the following are requirements pertaining specifically to the ideal operation of the ground station unit:

1. Data shall be received and stored by the Arduino Microprocessor.
2. Data received by the GPS modules shall be processed in real-time.
3. The ground station shall command the servo and gimbal subsystem with rotation and pitch sequences.

4.6.2 Design

The Ground station subsystem is responsible for logging and processing data received from the GPS module on the visual tracking system attached to the quadcopter, as well as GPS data from the module on the tracked object via a Wi-Fi transmitter. It consists of an Arduino microprocessor on the payload which receives data from both GPS modules in real time to evaluate the distance and angle between them. The ground station then commands the servo and gimbal system with rotation and pitch sequences.

4.6.2.1 Hardware

The block diagram below showcases the relationship between the ground station, which is the Arduino Microcontroller, itself, and the rest of the system. The wireless transceiver converts digital signals into analog signals for wireless transmission which are then converted back to digital signals for processing by the Arduino Microcontroller. The ground station is then programmed to perform mathematical computations to determine critical geometrical parameters such as distance and angle differences to provide appropriate commands to the servo and gimbal subsystem. These commands are then used to position the camera directly toward the object of interest.

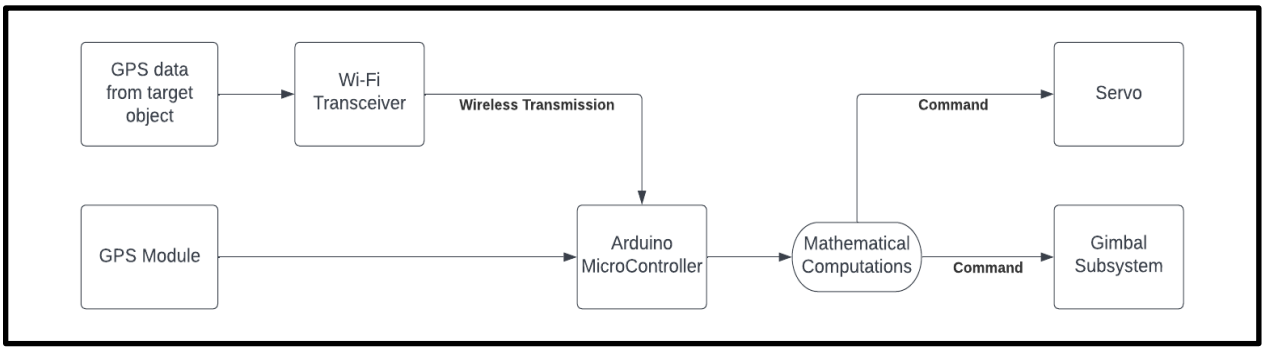

Figure 14 - Block diagram of Ground Station subsystem

4.6.2.2 Software

Since the Arduino microcontroller is a substitute for the ground station system for the Visual Tracking System, the relevant software algorithm is discussed in the Command & Data Handling System.

4.7 Bill of Materials (BOM)

The table below tabulates the bill of materials for the project. Some of the materials were provided by the Toronto Metropolitan University. This is indicated by the term “included”.

Table 10 - Bill of Materials

Component	Part Quantity	Cost ($)
Arduino Microcontroller	1	Included
Treedix Camera Module OV2640 [1]	1	$11.99
Servo motor	2	Included
GPS Receiver - GP20U7 [2]	1	$19.50
WIFI Transceiver - nRF24L01	1	Included
9 V Battery	1	Included
Clip Connectors	1	Included

5. System Interface

Figure 15 below shows the different interfaces that will be used in the payload. From the figure, we can establish a system interface showing all the connectors, inputs, and outputs between each subsystem. The system interface consists of two interfaces namely Internal and External interfaces. There are three subsystems interfacing in the internal interface. They are as follows:

1) Power Subsystem interfaces with Payload Subsystem
2) C&DH Subsystem interfaces with Payload Subsystem
3) Structured Subsystems interfaces with Payload Subsystem

There is only one subsystem interface in the external interface which is;

1) Structure Subsystem interfaces with Drone

The housing-mate component shown in the figure below is used to connect the structure subsystem to the Quadcopter base. There are six equal-space circular structures in a circular pattern that fits the circular slots on the base of the quadcopter establishing an interface. This base interface has to be attached to the payload package and all the components inside the payload package must be accessible.

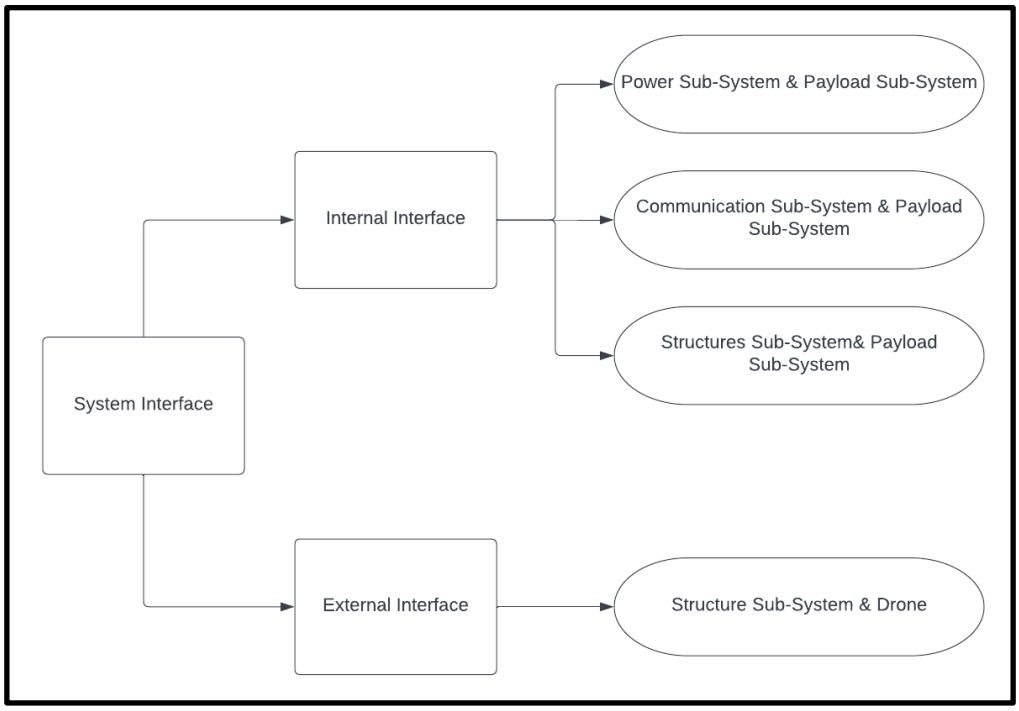

Figure 15 - System Interface Overview

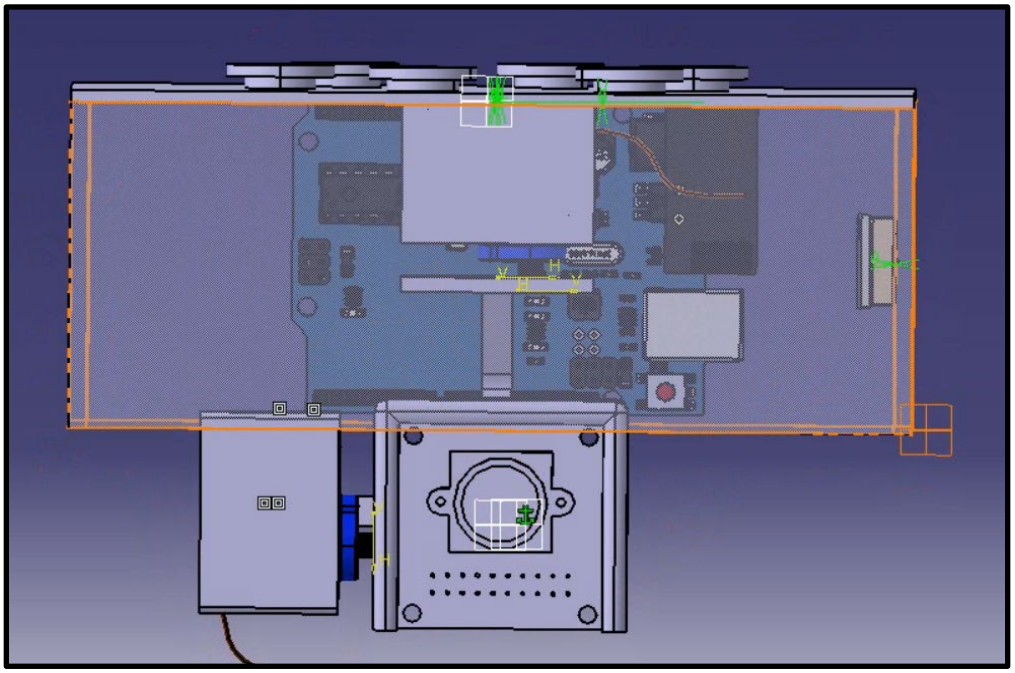

Figure 16 - Front View Drawing for Component Interface

5.1 System Interface Diagram <Data and Power>

From figure 16 below shows the overall connections and interfaces between the subsystems. From the figure, it can be seen that the Power subsystem is connected to the payload load system. This interface will be used to power the Arduino Microcontroller. The second interface is between the communication subsystem and the payload subsystem. Here, there are two components from the communication subsystem namely Wi-Fi Transceiver and GPS module connecting to the Arduino Microcontroller in the payload subsystem. These interfaces are used for locating the target object.

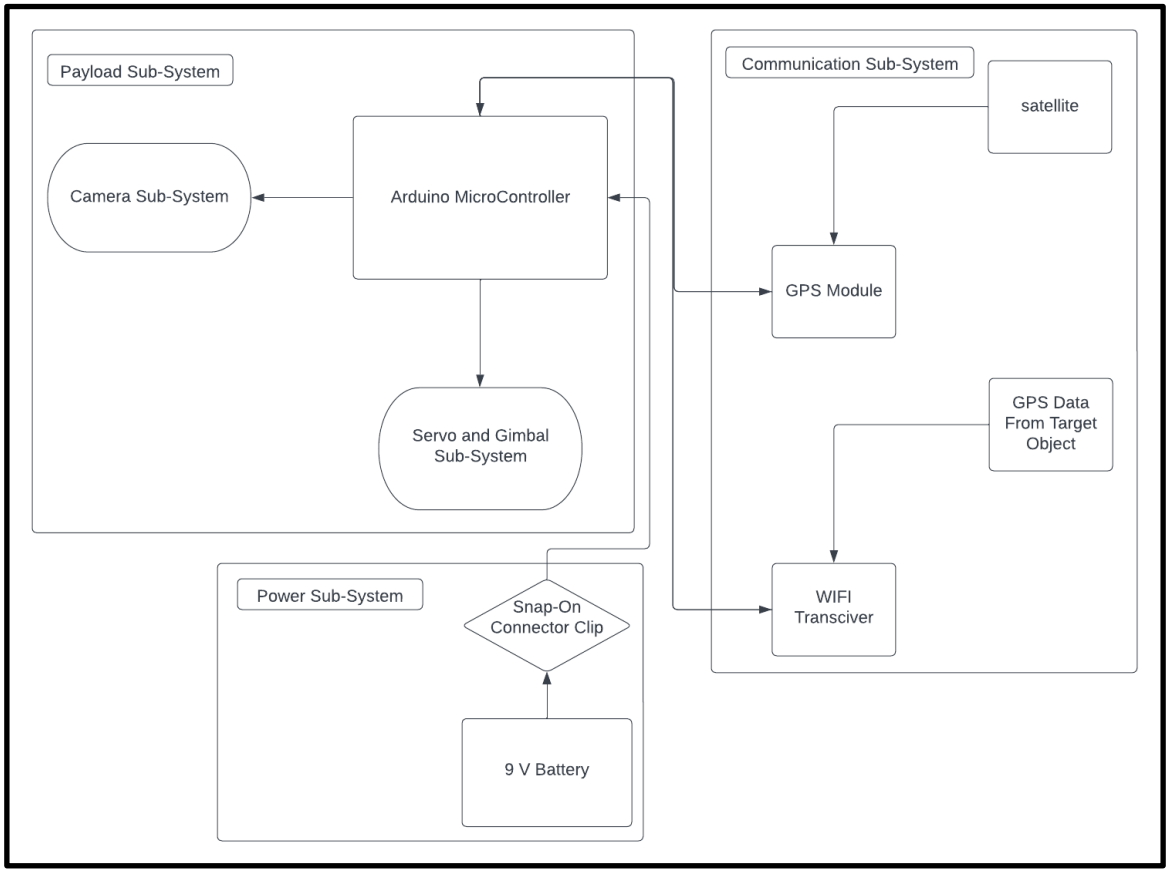

Figure 17 - System Interface Diagram

5.2 System Interface between Subsystems

5.2.1 Power subsystem and Payload subsystem

The power and payload subsystem interface are straightforward. As shown in figure 17, the power will be supplied from the 9V battery in the power subsystem to the Arduino Microcontroller in the payload subsystem.

5.2.1.1 Power Connectors

For the payload and power subsystem interface, we will be using the Snap-On connectors to supply the power to the Arduino Microcontroller. Jumper wires will be used to connect the battery to the other components.

5.2.2 Communication Subsystem and Payload Subsystem

For the communication and payload subsystem interface, the GPS data from the target object will be received by the Wi-Fi transceiver. The Wi-Fi transceiver in the communication subsystem is connected to the Arduino Microcontroller in the payload subsystem. Similarly, the GPS module will receive the target object coordinates and will send them to the Arduino Microcontroller. These subsystem interfaces will help in determining the location of the target object. An Arduino microcontroller is programmed to make the necessary calculations for the distance and angle of the received target coordinates. These calculations are then used by the Servo and Gimbal subsystem to turn and pitch the camera as per the target location.

5.2.2.1 Power Connectors

Jumper wires are used for connection between the Wi-Fi Transceiver and Arduino Microcontroller and the GPS module and the Arduino Microcontroller.

5.2.3 Structures Subsystem and Payload Subsystem

For the structures subsystem and the payload subsystem, the interface is a 3D-printed structure that will host the payload subsystem. As seen in the table above as to the items the payload system consists of these items will be inside the 3D-printed structure and will be secured. This structure will be mounted on the quadcopter using a twist and lock mechanism.

5.2.3.1 Power Connectors

The power connectors between the structures subsystem and the payload subsystem will be inapplicable because there is no need for electrical power transfer for the payload subsystem to interface with the structural subsystem.

5.3 N-Squared Diagram

The N-squared diagram for the end product is illustrated in the figure below. The N-squared diagram represents the logical data flow between various subsystems present in the designed payload (overall system). The overall items or functions are represented on the diagonal. For the designed payload the diagram indicates 8 items. The items are - Structure, Arduino

microcontroller, Power supply, GPS module, Camera Module, WIFI transceiver, Servos, and Camera gimbal. Each of these components has inputs and outputs. The inputs are indicated in columns and the outputs are indicated in rows. The N-squared diagram pinpoints areas where conflicts may arise between interfaces. The interface between the functions can be checked against the interface design illustrated in figure 14.

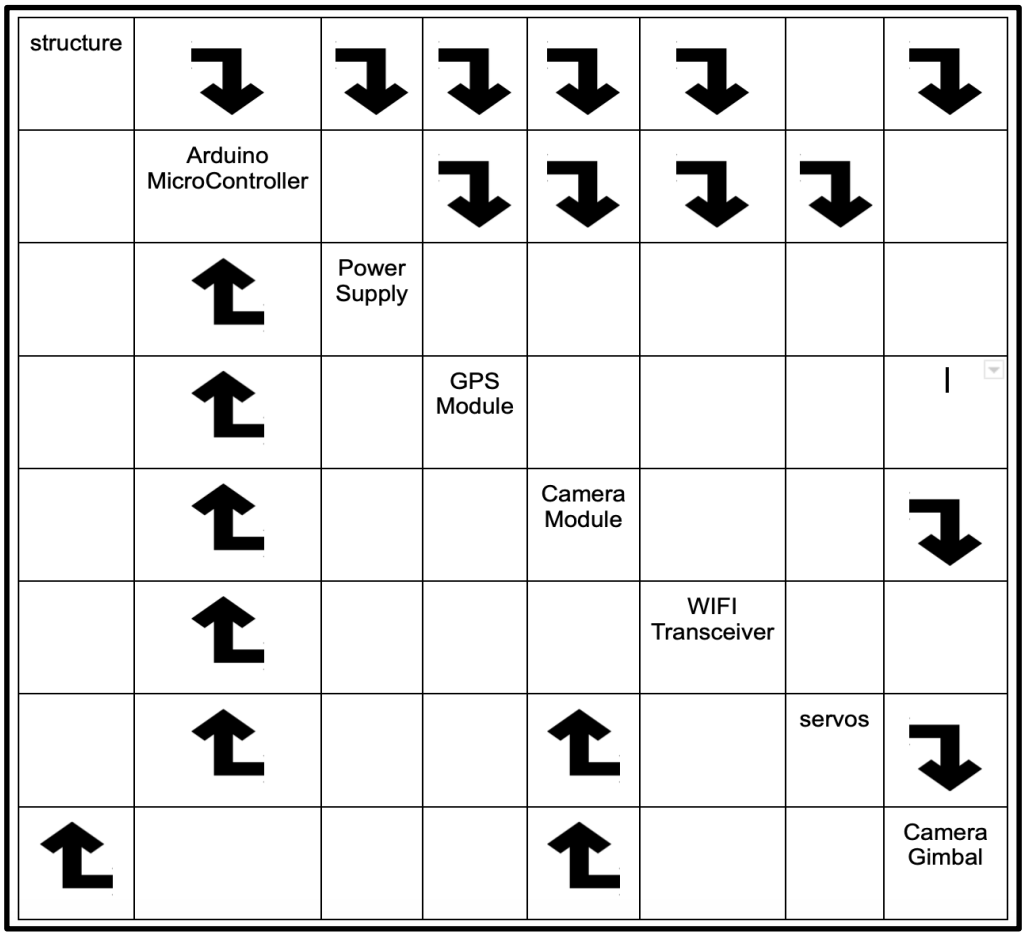

Figure 18 - N-Squared Diagram for the Payload

As seen in the figure above, the structure's function interfaces with the Arduino microcontroller, the power supply(batteries), the GPS module, the Camera module, the WIFI transceiver, and the Camera gimbal. This is because all the components are attached to the payload structure. The power supply module (battery pack) has an output that only shares the input with the Arduino microcontroller. The flight computer (Arduino) shares a feedback loop with the GPS module, the Camera module, the WIFI transceiver, and the Servos. The Arduino is responsible for power and voltage regulation for all the components. The Arduino is responsible for commanding the servos for rotation and pitch motion. The Arduino receives the GPS data from the GPS module

and the WIFI transceiver. The Camera module is attached to the Camera Gimbal, therefore, they also have a feedback loop. The servos are responsible for rotating and pitching the camera. Therefore the camera module has input from the servos.

6. Subsystem Testing

6.1 Payload Subsystem

The payload subsystem consists of the Arduino microcontroller, GPS module, camera module, WIFI transceiver, and the servo and gimbal system. After connecting these components with the Arduino Uno board, the functionality of each of these components was tested. The GPS module provided the correct output for the GPS coordinates of the target object. These individual components were connected to the gimbal and servo subsystem which was enclosed by the 3d-printed box. Each individual component was attached to the box using super glue. During the demonstration, it was observed that each of the components was secured in their respective places, and they were providing the relevant information.

6.2 Structures & Mechanisms

The Structures and Mechanism subsystem consists of a moving and fragile piece of equipment that is enclosed and safeguarded by a solid casing referred to as a body structure. Its main functions are to provide a protective exterior to make handling easier, to provide attachment points for internal mechanisms, to maintain the cleanliness of the contents by shielding them from fouling contaminants, and to protect interior mechanisms from structural stress and potential damage from the environment. In our design, the payload will be secured by a cuboid-shaped enclosure that will be 3D printed using ABSplus-P430 thermoplastic.

The system had three main components that required testing and analysis. First is the drone attachment disk that interfaces with the quadcopter. The disk was tested and modified to ensure that the attachment securely holds the payload system. The disk was modified to have the most minimal weight with high functioning capability. The second was the servo/gimbal mechanism. The mechanism was tested with a vibration test and structural integrity test. This was completed to ensure that the gimbal mount is steady during flight and can withstand strong gusts. The objective of the gimbal mechanism was to use the servo to rotate and pitch the camera module. Several tests were performed to ensure that there was no time lag for its operation. Lastly, the outer structural case was tested for its structural integrity and to ensure that all the components fit inside the case. After recording a few test results, the case dimensions were modified to optimize the weight-to-functionality ratio.

Table 11 - Validation Methods and Results for Structures and Mechanisms

Validation Subsystem	Objective	Method	Result
Structures and Mechanisms	Ensure that the structures and mechanisms for the system were strong and capable of their basic functionality.	1. Vibration test for gimbal mechanism. 2. Structural integrity test for the gimbal and the outer case. 3. Drone attachment disk test.	1. The gimbal proved to withstand the vibrations from the quadcopter rotors. 2. The drop-tests proved that the gimbal and the case were strong enough to withstand a drop from 3 ft. 3. A lightweight model was selected from the multiple disk models tested with the drone attachment system.

6.3 Command & Data Handling

The Command and Data Handling (C&DH) subsystem is essentially the brains of the visual tracking system and controls the servo and gimbal movement on the payload. The C&DH subsystem manages all forms of data received by the Arduino microcontroller. Therefore, it is essential to understand the process of data acquisition and data processing. This section will explain the subsystem's testing methods and their results. The system first connects to the GPS module on the payload to acquire the location of the quadcopter in terms of its altitude, latitude, and longitude. This data will then be stored on board the Arduino's memory and set as the datum for further calculations. Using the GPS data from the target object, the microcontroller calculates

the angle of rotation and pitch with the help of haversine formulas. The Arduino then commands the servo modules to rotate and/or pitch accordingly. The data acquired from the GPS is updated every 0.5 seconds. In order to receive the correct and incorrupt GPS coordinates, the module was tested in an open environment and cross-checked with the coordinates on google maps.

The primary objective for testing the sub-system was to ensure that it records and calculates the accurate angles for servo rotation and pitch. After several tests and code amendments, it was decided that the subsystem was reliable with a very low probability of failure. Although, it was discovered that if the system were to fail, it would result in category 2 level severity impacts on the mission objective. The following figure illustrates a demo test for data acquisition and data processing.

```
Data Set 1:
$GPGLL,4341.41875,N,07917.98709,W,225614.00,A,A*77
$GPRMC,225615.00,A,4341.41876,N,07917.98698,W,0.191,,041222,,,A*6B
$GPVTG,,T,,M,0.191,N,0.353,K,A*2F
Lat, Lon, and Altitude: $GPGGA,225615.00,4341.41876,N,07917.98698,W,1,04,2.11,133.2,M,-35.9,M,,*6E
$GPGSA,A,3,08,07,09,27,,,,,,,,,5.55,2.11,5.13*01
$GPGSV,2,1,06,04,,,23,07,59,307,31,08,74,141,35,09,41,228,32*48
$GPGSV,2,2,06,26,02,080,,27,52,059,25*78

Data Set 2:
$GPGLL,4341.41876,N,07917.98698,W,225615.00,A,A*7C
$GPRMC,225616.00,A,4341.41882,N,07917.98703,W,0.118,,041222,,,A*61
$GPVTG,,T,,M,0.118,N,0.218,K,A*20
Lat, Lon, and Altitude: $GPGGA,225616.00,4341.41882,N,07917.98703,W,1,04,2.11,133.8,M,-35.9,M,,*6F
$GPGSA,A,3,08,07,09,27,,,,,,,,,5.54,2.11,5.13*00
$GPGSV,2,1,06,04,,,23,07,59,307,30,08,74,141,35,09,41,228,31*4A
$GPGSV,2,2,06,26,02,080,,27,52,059,25*78

Data Set 3:
$GPGLL,4341.41882,N,07917.98703,W,225616.00,A,A*77
$GPRMC,225617.00,A,4341.41889,N,07917.98706,W,0.183,,041222,,,A*6C
$GPVTG,,T,,M,0.183,N,0.339,K,A*20
Lat, Lon, and Altitude: $GPGGA,225617.00,4341.41889,N,07917.98706,W,1,04,2.11,134.2,M,-35.9,M,,*6D
$GPGSA,A,3,08,07,09,27,,,,,,,,,5.54,2.11,5.12*01
$GPGSV,2,1,06,04,,,22,07,59,307,29,08,74,141,34,09,41,228,32*41
$GPGSV,2,2,06,26,02,080,,27,52,059,26*7B
```

Figure 19 - Data Acquisition from the GPS Module

The GPS module returns with the latitude, longitude, and altitude of the quadcopter. Along with it, the data indicates the number of satellites connected to the GPS module. This is necessary to acquire accurate geographic data. For the above displayed data set, 4 satellites were connected to the GPS module.

Table 12 - Validation Methods and Results for Command & Data Handling

Validation Subsystem	Objective	Method	Result
Command & Data Handling	Ensure correct data acquisition and data processing.	4. Testing: Data from the GPS is checked with google maps. 5. Testing: Data is processed to calculate the rotation and pitch angles.	All data is received and processed accurately without corruption.

6.4 Power Subsystem

The power subsystem is responsible for supplying power to the Arduino microcontroller. As outlined in the design definition, the final product has to be a self-contained system. Therefore, the Visual Tracking System has to have an internal power supply as it cannot use the power from the quadcopter. The power subsystem shall adequately supply power to the different components of the Visual Tracking System. In order for the system to have a safe operation, the voltage supplied by the battery should not exceed the normal operating range of each individual component of the Visual Tracking System.

The verification and validation method for this subsystem was completed in terms of voltage, current, and power analysis. From the available data of each sensor/component, the maximum power consumption was calculated. Based on this, a battery was selected amongst various choices to optimize the power delivery and weight of the power subsystem. The power and mass budget are tabulated and discussed in the above sections.

6.5 Communication Subsystem

The communication subsystem for the designed payload consists of sensors, transmitters, and receivers that are responsible for real-time data communication to the Arduino microcontroller. The in-built memory modules on the Arduino microcontroller were utilized to store the incoming and outgoing flight data from and to the sensors. It was essential to test the communication methods for each component separately and then for the entire payload system. This method was adopted to identify and mitigate the bugs in the communication code. To ensure safe and reliable communication with each component, the wiring connections were tested using a vibration test. This was completed to ensure that the wires wouldn’t become loose during the actual flight demonstration due to the vibrations from the quadcopter.

The communication modes for each sensor/module were tested prior to the testing. The GPS module transmitted the correct coordinates of the quadcopter, and the servo modules output the rotation and pitch motion from the data processed by the microcontroller. During the pre-flight testing, the camera module was coded separately to communicate with the Arduino microcontroller(flight computer). However, the camera module wasn't able to produce the output video feed as it was supposed to.

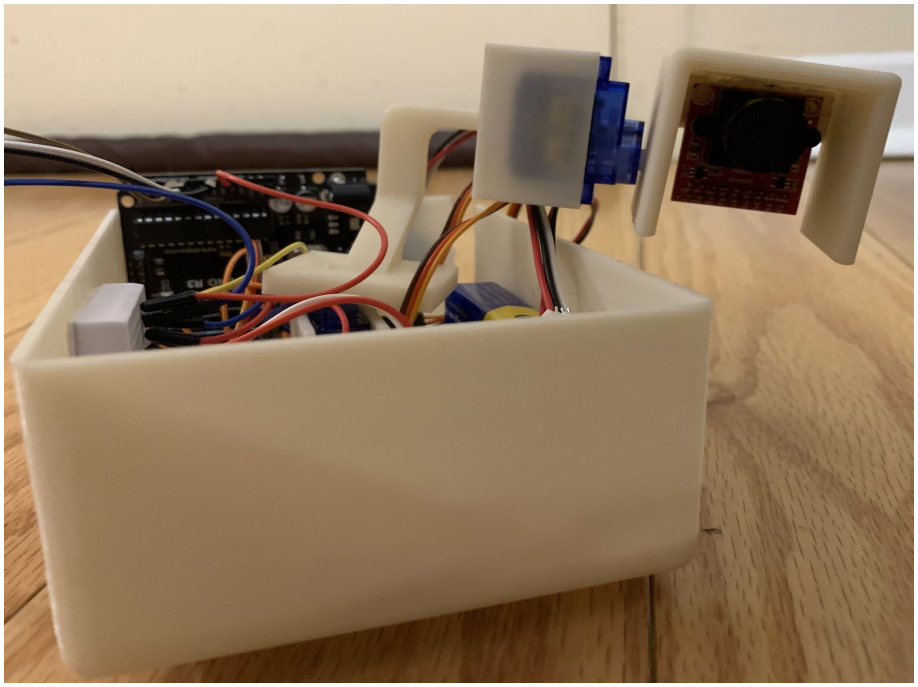

Figure 20 - Sensor Communication Connection

Table 13 - Validation Methods and Results for Communications

Validation Subsystem	Objective	Method	Result
Communication	Ensure proper communication with each sensor.	1. Vibration Test. 2. Communication with the GPS Module. 3. Communication with the Servo Modules. 4. Communication with the Camera Module.	1. All wiring connections were strong. 2. All GPS data is received without corruption. 3. The servos functioned without any time lag.

			4. The Camera Module didn't transmit any video feed to the computer.

6.6 Ground Station

A ground station is a terrestrial radio station designed for receiving radio waves from aeronautical radio sources or communicating with airborne devices. Ground stations may be situated either on the surface of the earth or somewhere within its atmosphere. In the case of this design, the ground station is the Arduino microprocessor itself and is responsible for commanding the camera system with necessary instructions to always keep a direct line of sight of the tracked object. The primary function of the ground station is to perform mathematical calculations with the data received by the GPS modules to provide appropriate instructions to the servo and motor to control the camera.

The testing methods for the ground station, the Arduino microcontroller for this project, are similar to the methods from the Command and Data Handling, Communications, and Power subsystems combined. These methods were utilized to ensure that the payload accomplishes all of its functions.

6.7 Integrated System (Payload Unit)

The overall payload system is the integration of all the subsystems and their interaction with the flight computer. The subsystems included the Camera module, GPS module, Gimbal Subsystem, and power module. The interaction between the sensors and the Arduino was tested to ensure satisfactory operation for the payload system in the intended environment. The operation/interaction of each subsystem was tested during the communication and Command and Data Handling testing phase. These tests confirmed the satisfactory performance of the GPS and the servo modules. However, the camera module was not successful during this phase. The camera module was tested several times for its output video feed but the data communication between the camera and the flight computer failed due to the lack of Wi-Fi connectivity.

Following the individual testing, the integrated system was tested to make sure they functioned together and performed optimally in their intended environment. The tests included drone-like simulations (vibration, strong gust from the rotor).

Table 14 - Validation Methods and Results for Integrated Systems

Validation Subsystem	Objective	Method	Result
Integrated Systems	Ensure that the system works in its intended environment.	1. Vibration Test. 2. Wind Test. 3. Servo Functionality Test. 4. GPS Data acquisition Test.	1. The payload structure was successful in its vibration and wind testing. 2. The servo rotated and pitched according to the calculated angles. 3. The GPS sensor communicated accurate GPS data.

7. Conclusion

In conclusion, the project was successful in creating the Visual Tracking System which transmitted and received data from the GPS and gimbal, and servo system. The data was successfully processed into latitude and longitude coordinates which corresponded to the correct location of the target. The requirements for mass and volume were met as the total mass was less than the maximum allowable mass by 56 grams, and the volume of the payload was 750 cm^3 which was significantly less than the maximum allowable volume of 8000 cm^3. Moreover, the demonstration also proved that the payload securely attaches to the quadcopter without significant vibrations on the payload. All the individual components were securely glued to the payload using the superglue which prevented them from falling or destabilizing during the flight demonstration of the quadcopter.

8. References

[1] *Treedix camera module OV2640 2MP STM32F4 driver source code support ...* (no date). Available at: https://www.amazon.com/Treedix-Camera-STM32F4-Support-Arduino/dp/B085XV999Q (Accessed: October 2, 2022).

[2] #563666, M. *et al.* (no date) *GPS receiver - GP-20U7 (56 channel)*, *GPS-13740 - SparkFun Electronics*. Available at: https://www.sparkfun.com/products/13740 (Accessed: November 27, 2022).

9. Appendix A: C&DH and Communication Subsystems code

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include "esp_camera.h"
#include <WiFi.h>
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"
const char* ssid = "Aman's Iphone";
const char* password = "Aman1234";

void startCameraServer();

Servo servo1;
Servo servo2;
int GPSVal;
Int latVal;
Int lonVal;
SoftwareSerial GPS_Serial(10, 11); // RX, TX

void setup()
{
 GPS_Serial.begin(115200);
 servo1.attach(8);
 servo2.attach(9);
 Serial.begin(115200);
 Serial.setDebugOutput(true);
 Serial.println();

 camera_config_t config;
 config.ledc_channel = LEDC_CHANNEL_0;
 config.ledc_timer = LEDC_TIMER_0;
 config.pin_d0 = Y2_GPIO_NUM;
 config.pin_d1 = Y3_GPIO_NUM;
 config.pin_d2 = Y4_GPIO_NUM;
 config.pin_d3 = Y5_GPIO_NUM;
 config.pin_d4 = Y6_GPIO_NUM;
 config.pin_d5 = Y7_GPIO_NUM;
 config.pin_d6 = Y8_GPIO_NUM;
 config.pin_d7 = Y9_GPIO_NUM;
 config.pin_xclk = XCLK_GPIO_NUM;
```

```
 config.pin_pclk = PCLK_GPIO_NUM;
 config.pin_vsync = VSYNC_GPIO_NUM;
 config.pin_href = HREF_GPIO_NUM;
 config.pin_sscb_sda = SIOD_GPIO_NUM;
 config.pin_sscb_scl = SIOC_GPIO_NUM;
 config.pin_pwdn = PWDN_GPIO_NUM;
 config.pin_reset = RESET_GPIO_NUM;
 config.xclk_freq_hz = 20000000;
 config.pixel_format = PIXFORMAT_JPEG;
 if(psramFound()){
   config.frame_size = FRAMESIZE_UXGA;
   config.jpeg_quality = 10;
   config.fb_count = 2;
 } else {
   config.frame_size = FRAMESIZE_SVGA;
   config.jpeg_quality = 12;
   config.fb_count = 1;
 }
#if defined(CAMERA_MODEL_ESP_EYE)
 pinMode(13, INPUT_PULLUP);
 pinMode(14, INPUT_PULLUP);
#endif


 esp_err_t err = esp_camera_init(&config);
 if (err != ESP_OK) {
   Serial.printf("Camera init failed with error 0x%x", err);
   return;
 }

 sensor_t * s = esp_camera_sensor_get();

 if (s->id.PID == OV3660_PID) {
   s->set_vflip(s, 1); // flip it back
   s->set_brightness(s, 1); // up the brightness just a bit
   s->set_saturation(s, -2); // lower the saturation
 }
 // drop down frame size for higher initial frame rate
 s->set_framesize(s, FRAMESIZE_QVGA);
```

```
#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
 s->set_vflip(s, 1);
 s->set_hmirror(s, 1);
#endif

 WiFi.begin(ssid, password);

 while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }
 Serial.println("");
 Serial.println("WiFi connected");

 startCameraServer();

 Serial.print("Camera Ready! Use 'http://");
 Serial.print(WiFi.localIP());
 Serial.println("' to connect");
}
}

void loop()
{
char rc;
if (GPS_Serial.available())
{
        rc = GPS_Serial.read();
        Serial.write(rc);
        LocationVal = GPS_Serial.getGeolocation();
        latVal = LocationVal.latitude;
lonVal = LocationVal.longitude;

GPSVal = latVal;
        GPSVal = map (GPSVal, 0, 1023, 0, 180);
        servo1.write(GPSVal);
GPSVal = longVal;
GPSVal = map (GPSVal, 0, 1023, 0, 180);
        servo2.write(GPSVal);
```

```
        delay(15);
}
delay(10000);
}
```