Toronto Metropolitan University

Department of Aerospace Engineering
Faculty of Engineering & Architectural Science

Semester (Term, Year)	Fall 2024
Course Code	AER 817
Course Section	Section 03
Course Title	System Engineering
Course Instructor	Dr. Krishna Kumar
Course TA	Alessandro Ciniello
Submission	04
Submission Due Date	12/02/2024
Title	Project Phase III Report
Submission Date	December 02, 2024

Submission by (Name):	**Student ID (XXXX1234)**	**Signature**
Khadeeja Azizi	XXXX47707	Khadeeja
Abigail Marsella	XXXXX3590	Abigail Marsella
Fadia Matti	XXXXX9338	Fadia Matti
Elliott Arpino	XXXXX5958	EA
Maia Elizabeth Gorham	XXXX53988	MEG

By signing the above you attest that you have contributed to this submission and confirm that all work you contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, and "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Academic Integrity Policy 60, which can be found at www.torontomu.ca/senate/policy60

RadiationXplorer
Phase III Report

AER817: SYSTEMS ENGINEERING
Department of Aerospace Engineering
Toronto Metropolitan University
350 Victoria St. Toronto, Ontario, M5B 2K3

Version:	1		
Author:	Elliott Arpino Khadeeja Azizi Maia Elizabeth Gorham Abigail Marsella Fadia Matti	Supervisor:	Krishna Kumar
Date:	December 2, 2024		

Table of Contents

List of Figures

List of Tables

1.0 Introduction & Summary of the Work Done by Team Members

The primary mission objective is to autonomously monitor radiation levels in radioactive wastelands using a payload attached to a drone. The payload is equipped with a Geiger counter, GPA, and data storage and communication capabilities. Secondary mission objectives include real-time data transmission and geospatial mapping for radiation level monitoring.

The following describes the mission overview. The payload is attached to the drone via a simple 3D printed attachment component. The drone is flown around various locations. Radiation levels are taken via the Geiger counter as well as their corresponding coordinates which are taken by the GPS module all autonomously. This information is relayed to the ground station (a phone or laptop) in real-time via the ESP32's wifi transceiver. The same information is also stored in the payload onto a microSD card, which can later be extracted and uploaded to a computer. This information is useful because it allows humans to decide if certain areas are safe or still harmful without risking human lives. It is also useful to monitor how radiation levels change over time and if they align with our expectations for research purposes.

This design is innovative because of its lightweight design, dual data handling system for redundancy, and the integration of low cost components. It is very easily reproducible and accessible. Figure 1.1 depicts a visual illustration of the mission concept.

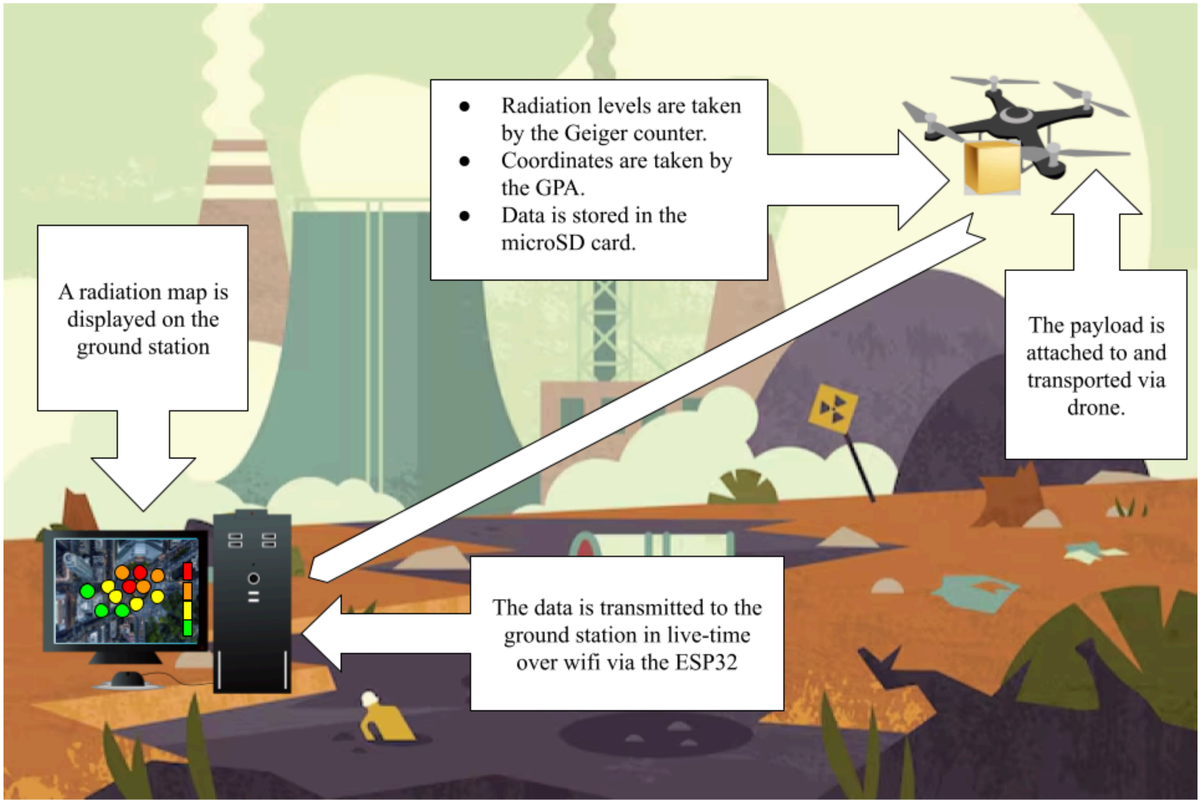

Figure 1.1. Mission Concept

Figure 1.1 depicts the CAD model of the interior of the payload unit with all the components housed inside.

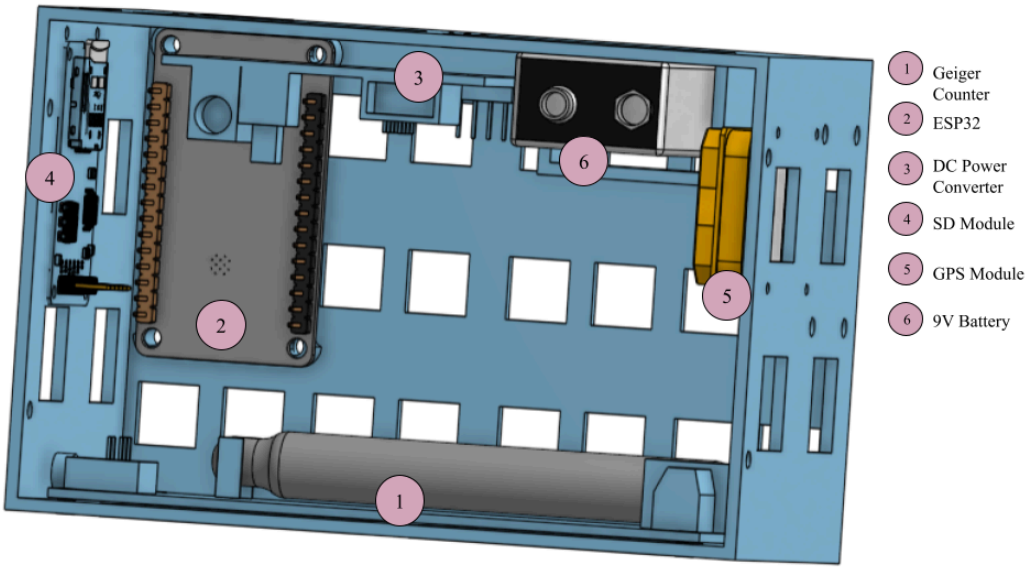

Figure 1.2. CAD Payload Unit Model

Figure 1.3 shows an image of the final payload unit fully assembled and ready to be tested.

Figure 1.3. Payload Unit

Table 1.1 below lists the work breakdown and contribution for each member of the group.

Table 1.1. Work Breakdown and Contribution Summary

Elliott Arpino	• Body subsystem • Ground station subsystem • Command and data handling • Coding • Wiring • Assembly • Flight Test
Khadeeja Azizi	• Payload subsystem • System interface between payload and power • Assembly • Mission concept • Flight Test • Report Formatting and Editing
Maia Elizabeth Gorham	• CAD design and 3D printing • Hardware assembly and wiring • Mission concept diagram
Abigail Marsella	• Power subsystem • Communication subsystem • System interface between payload and power • Assisted Coding • Assembly • Flight Test • Report Formatting and Editing
Fadia Matti	• Command and data handling (C&DH) subsystem • System interface diagram of data and power • N2 diagram • Assembly • Flight Test • Report Formatting and Editing

2.0 System Objectives & Requirements

2.1 System Objectives

The following lists the main system objectives:

- **Radiation Detection:** The system must reliably measure and log radiation levels in counts per minute using a Geiger-Müller counter. The data must be accurate enough for use in environmental assessments, enabling authorities to determine radiation intensity and its spatial distribution across affected areas.
- **Geospatial Mapping:** The system must integrate a GPS module to track the drone's position during flight, correlating radiation measurements with specific locations. This will allow for the creation of detailed radiation maps, showing how radiation levels vary geographically within the monitored area.
- **Lightweight Design**: The payload must weigh less than 300 grams to ensure that the rotor copter drone can maintain stable flight while carrying it. This constraint requires careful selection of lightweight components, such as sensors, communication devices, and power supplies.
- **Low Cost**: The total cost of all new hardware, excluding pre-existing components like the Geiger counter, must not exceed $50. This makes the system accessible for applications in regions with limited financial resources, allowing for wider use in developing countries or for educational purposes.
- **Autonomous Operation**: The system must be capable of collecting data autonomously, with minimal human intervention required during flight. Once deployed, the drone must fly predetermined routes while continuously gathering radiation data, eliminating the need for human presence in hazardous areas.

The following lists the secondary system objectives:

- **Environmental Durability**: The system must be designed to operate in harsh environments, such as radiation-rich areas, where electronic components may be at risk of interference or damage. Shielding sensitive components, like the Arduino board and communication modules, with lightweight protective materials (e.g., aluminum foil) is essential to ensure the system remains functional throughout the mission.
- **Power Efficiency**: The system must prioritize energy efficiency to extend the drone's flight time. Non-essential sensors and communication modules should be powered down when not in use, ensuring that battery life is conserved for the most critical operations. Efficient power management is vital for long-duration missions in remote areas where recharging or battery replacement may not be feasible.
- **Scalability for Broader Applications**: Although the initial deployment focuses on monitoring radiation levels in contaminated sites, the system should be designed with the potential for scalability. By making minor adjustments to the payload, the system could be adapted for other environmental monitoring applications, such as detecting chemical contamination or air quality, expanding its use in disaster response and environmental research.

2.2 System Requirements

The payload subsystem for the RadiationXplorer drone mission has specific, measurable requirements to achieve reliable, autonomous radiation monitoring and are depicted in Table 2.1. Each requirement focuses on functional, performance, environmental, and budgetary constraints for the payload subsystem.

Table 2.1. Payload Subsystem Requirements.

ID	**Requirement**	**Verification Method**	**Status**
RQ-1.1	The payload shall have a total mass of less than 300 grams.	Measure the fully assembled payload on a scale.	Met

ID	**Requirement**	**Verification Method**	**Status**
RQ-1.2	The payload shall fit within a 20 cm x 20 cm x 20 cm volume.	Measure the dimensions of the fully assembled payload with a ruler.	Met
RQ-1.3	The Geiger-Müller counter shall detect and record radiation levels with ±10% accuracy.	Test by collecting data and comparing it to theoretically known radiation levels.	Met
RQ-1.4	The GPS module shall provide geospatial data with positional accuracy of ±5 meters, updated every second.	Test by collecting data and comparing it to known geospatial positioning locations.	Met
RQ-1.5	The SD card module shall store a minimum of 1 GB of radiation and GPS data per mission.	Test by collecting data and ensuring it matches the live-time data values.	Not met
RQ-1.6	The payload shall autonomously collect and log data for at least 30 minutes per mission.	Measure the time the payload can collect data with a timer.	Met
RQ-1.7	The ESP32's built-in Wi-Fi transceiver shall provide optional real-time data transmission to the ground station.	Test by collecting data and receiving it without any wires plugged into the ground station.	Met
RQ-1.8	All payload components shall operate within a temperature range of -10°C to 40°C.	Measure the temperatures the payload can withstand using a thermometer.	Not met
RQ-1.9	Sensitive electronic components shall be shielded from radiation interference.	Test that accurate data is collected by all the components in the presence of radiation.	Met

ID	Requirement	Verification Method	Status
RQ-1.10	Total expenditure for new payload components shall not exceed $50.	Calculate the total price of the entire payload to ensure the 50$ budget is met.	Met
RQ-1.11	The payload's power supply shall support all components continuously for a minimum of 30 minutes.	Test the battery lasts for a minimum of 30 minutes using a timer.	Met
RQ-1.12	The payload enclosure shall be robust enough to protect components during drone flight and minor impacts.	Test that the payload enclosure safely houses the components during drone flight movements, forces, and vibrations.	Met

2.2.1 Degree of Meeting Requirement Matrix

The following Meeting Requirements Matrix, shown in Table 2.2, outlines the payload requirements, their rationale, verification method, and priority level. This highlights the significance of each requirement for optimal performance and safety, and priority level of high, medium, or low.

Table 2.2. Meeting Requirements Matrix for Payload Subsystem.

ID	Requirement	Design Rationale	Verification Method	Priority Level
RQ-1.1	The payload shall have a total mass of less than 300 grams.	Ensures drone stability and efficient flight by minimizing weight, thus reducing the risk of payload-induced destabilization.	Record and verify each component's weight prior to full payload integration.	High

ID	Requirement	Design Rationale	Verification Method	Priority Level
RQ-1.2	The payload shall fit within a 20 cm x 20 cm x 20 cm volume.	Compact size minimizes air resistance and improves flight duration, ensuring compatibility with the drone's payload bay constraints.	Use 3D modeling and CAD to confirm that the design complies with volume limitations.	Medium
RQ-1.3	The Geiger-Müller counter shall detect and record radiation levels with ±10% accuracy.	Ensures reliable environmental data for post-mission radiation mapping, meeting mission goals of radiation assessment.	Validate radiation readings using a controlled environment with known radiation sources.	High
RQ-1.4	The GPS module shall provide geospatial data with positional accuracy of ±5 meters, updated every second.	Enables precise geolocation of radiation data, supporting accurate mapping and data correlation across the target area.	Test GPS accuracy in open-sky conditions against benchmarked locations.	High
RQ-1.5	The SD card module shall store a minimum of 1 GB of radiation and GPS data per mission.	Provides adequate storage for a full mission's dataset, enabling detailed post-flight data analysis without risk of data loss due to limited storage.	Conduct continuous data logging for a simulated mission to ensure storage capacity is sufficient.	High
RQ-1.6	The payload shall autonomously collect and log	Supports continuous, autonomous data collection for the mission	Test the system's autonomous operation in a controlled setting	High

ID	Requirement	Design Rationale	Verification Method	Priority Level
	data for at least 30 minutes per mission.	duration, eliminating the need for ground-based control during flight.	to confirm 30+ minutes of data logging.	
RQ-1.7	The Wi-Fi transceiver shall provide optional real-time data transmission to the ground station.	Enables immediate access to radiation and GPS data, facilitating rapid analysis and situational awareness during mission-critical flights.	Test Wi-Fi connectivity and data transmission within the mission's operational range.	High
RQ-1.8	All payload components shall operate within a temperature range of -10°C to 40°C.	Ensures reliable operation in varied environmental conditions, as expected during missions.	Perform thermal tests in controlled conditions to verify each component's operational temperature range.	Medium
RQ-1.9	Sensitive electronic components shall be shielded from radiation interference.	Protects against data corruption and potential hardware damage from radiation exposure, improving mission resilience.	Incorporate shielding materials and confirm performance in a simulated radiation environment.	High
RQ-1.10	Total expenditure for new payload components shall not exceed $50.	Ensures cost-effectiveness, supporting a design that is economically viable and accessible for similar projects.	Verify total expenses by itemizing costs in the Bill of Materials (BOM) and confirm compliance.	High

ID	Requirement	Design Rationale	Verification Method	Priority Level
RQ-1.11	The payload's power supply shall support all components continuously for a minimum of 30 minutes.	Ensures sufficient energy for the mission duration, allowing all sensors and data storage systems to operate without interruption.	Conduct a continuous power test, recording current draw and power consumption to verify battery life.	High
RQ-1.12	The payload enclosure shall be robust enough to protect components during drone flight and minor impacts.	Ensures that all components are secure within the payload, reducing risk of damage from vibrations or minor impacts during transport or flight.	Perform drop tests and vibration tests to confirm structural integrity under standard flight conditions.	High

2.3 Mass and Power Budget

The mass and power budget provides a breakdown of the weight and energy consumption of each component in the system. It ensures that the payload meets the mass limit of 300 grams and operates efficiently within the power constraints to maintain optimal flight time. Table 2.3 below demonstrates the mass and power for the corresponding components. Power was calculated when components are operating at their maximum.

Table 2.3: Mass and Power Budget

Component	Mass (g)	Power (W)
Arduino Uno	25	25
Geiger Counter	69	25

Component	Mass (g)	Power (W)
GPS Module	20	2
SD Module	10	6.6
Wifi Transceiver	3	2.31
Micro SD Card	0.5	/
USB	110	/
Enclosure	33[1]	/
Totals	270.5	60.91

[1] *This number is an estimated value.*

2.4 Technology Readiness Level

At the end of this AER817 course, the product is at a technological readiness level of 9. This means the product has completed the basic research, product concept formulated, and the system and subsystems are formulated and honed. The assembly is completed and works accurately, efficiently, and correctly. The body structure subsystem has been 3D printed, the payload system added to the structure and readily available to be assembled to the drone provided, code is formulated to the subsystems, and all subsystems are tested and working, payload and user device are in constant wireless connection. Finally, the system is completed and ready for the flight test, members saw RadiationXplorer having a successful flight test, showing the system created works as it was designed, and data output presumed correct. The data output is proven to be correct since the radiation levels in CPM, coordinates, and time are presented simultaneously.

3.0 Project Gantt Chart, Work Breakdown Structure, and Work Packages

3.1 Project Gantt Chart

This section presents the project timeline in the form of a Gantt chart, illustrating the key phases and tasks. The following Figure 3.1 displays the phases of the project from the initial idea through fabrication and testing.

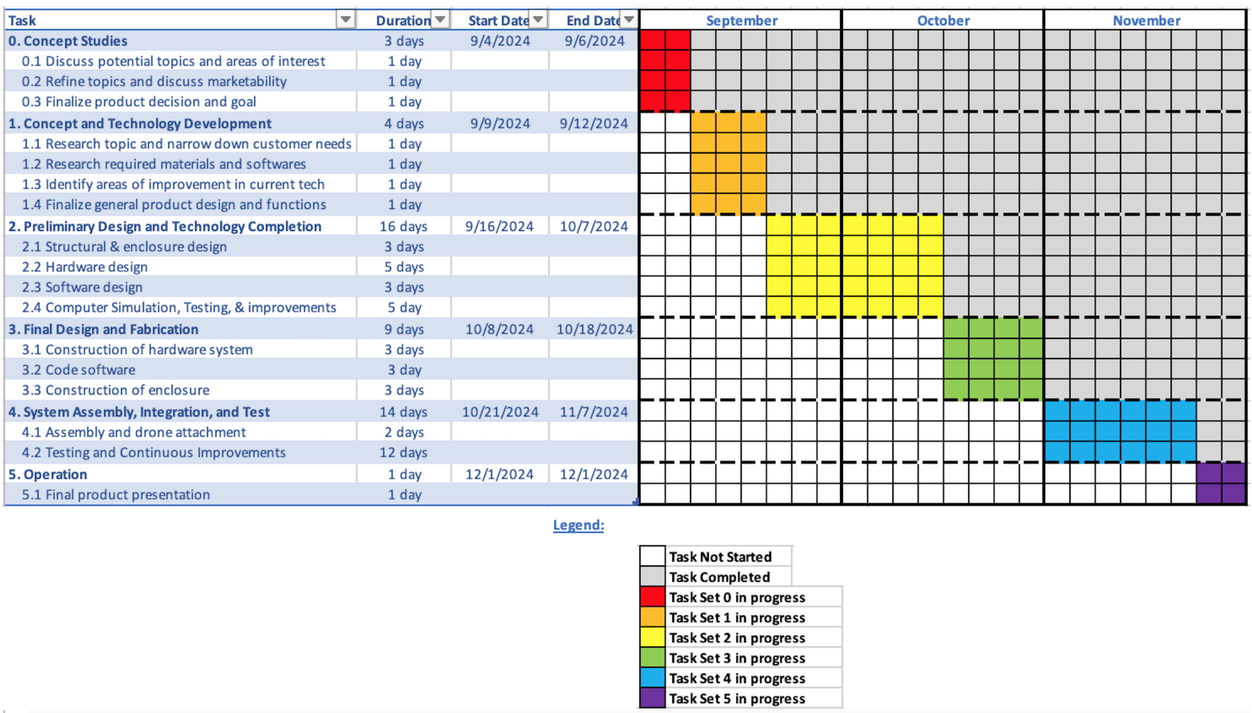

Figure 3.1: Gantt Chart

3.2 Work Breakdown Structure

This section outlines the Work Breakdown Structure (WBS) of the project, breaking down the key tasks and deliverables into manageable components. Figures 3.2 and 3.3 illustrate the overall WBS and provide a detailed breakdown for WPS 5000, helping to clarify responsibilities and ensure that all aspects of the project are accounted for and systematically organized.

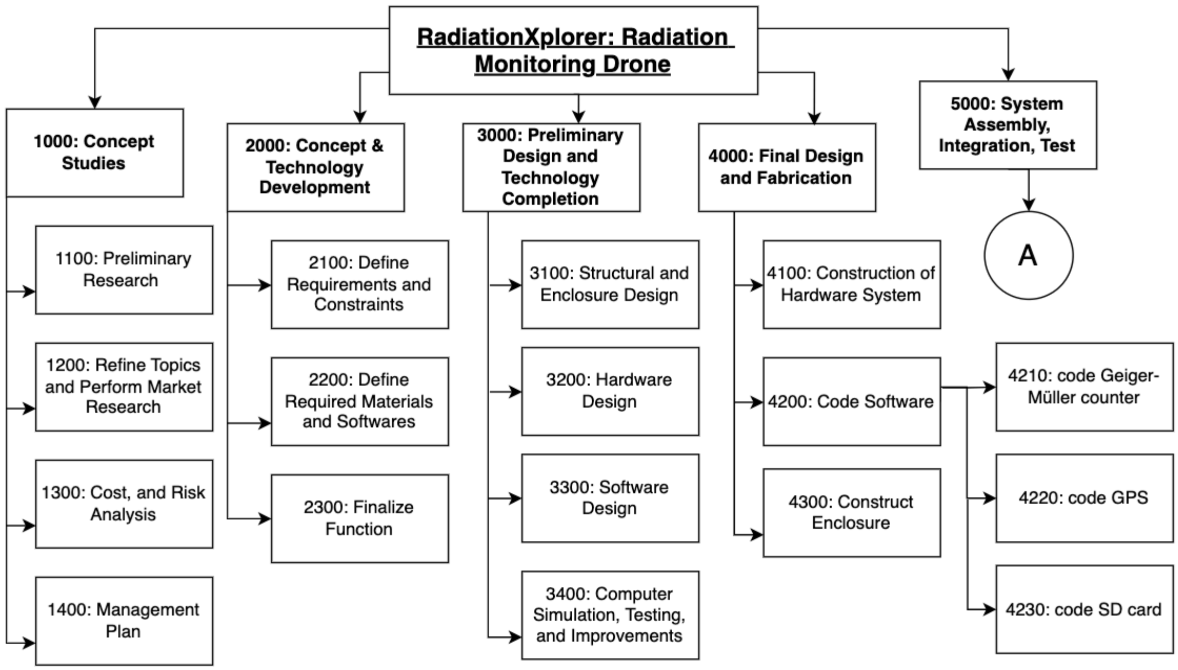

Figure 3.2: Work Breakdown Structure.

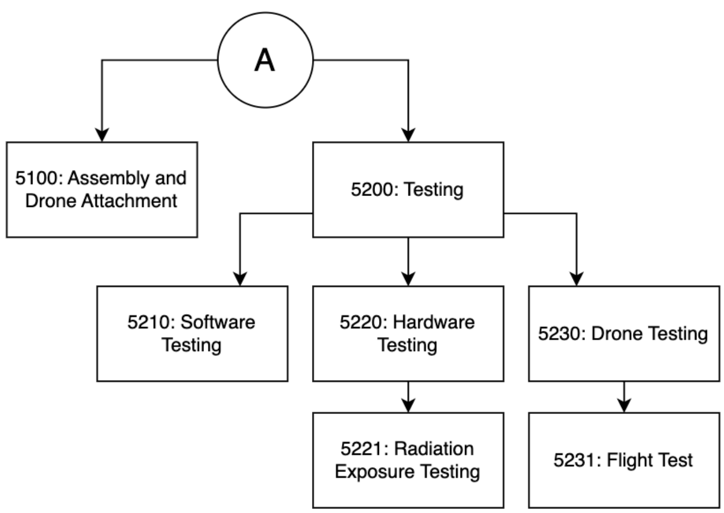

Figure 3.3: Detailed Work Breakdown Structure for WPS 5000.

3.3 Work Packages

The following Tables 3.1-3.5 depict the work packages referenced in Section 6.0 of this report, and correspond to the work breakdown structure seen in Section 3.2.

Table 3.1: Work Package Definition - WBS #1000

<table>
<tr><td colspan="4">Project: RadiationXplorer Radiation Monitoring Drone</td></tr>
<tr><td>Work Pack Title:</td><td>Concept Studies</td><td colspan="2">WBS Ref: 1000</td></tr>
<tr><td>Sheet:</td><td>1 of 1</td><td colspan="2"></td></tr>
<tr><td>Scheduled start:</td><td>September 4, 2024</td><td>Accountable Managers:</td><td>E. Arpino, K. Azizi, M.E. Gorham, A. Marsella, F. Matti</td></tr>
<tr><td>Scheduled end:</td><td>September 10, 2024</td><td rowspan="2">Resources</td><td rowspan="2">E. Arpino, K. Azizi, M.E. Gorham, A. Marsella, F. Matti</td></tr>
<tr><td>Estimated effort:</td><td>48 hours</td></tr>
<tr><td colspan="4"><u>Objectives:</u>
● Preliminary Research
● Topic Definition
● Cost and Risk Analysis</td></tr>
<tr><td colspan="4"><u>Inputs:</u>
● Research of current technology
● Research of relevant applications</td></tr>
<tr><td colspan="4"><u>Tasks:</u>
● Host initial meeting, discuss potential topics
● Perform market research
● Perform cost analysis
● Perform risk analysis
● Division of work and Management Plan</td></tr>
</table>

Outputs/Deliverables: ● Meeting Minutes ● Project Plan ● Finalize Concept and Project ● Outline steps to achieve goal

Table 3.2: Work Package Definition - WBS #2000

<table>
<tr><td colspan="4">Project: RadiationXplorer Radiation Monitoring Drone</td></tr>
<tr><td>Work Pack Title:</td><td>Concept and Technology Development</td><td colspan="2">WBS Ref: 2000</td></tr>
<tr><td>Sheet: 1 of 1</td><td colspan="3"></td></tr>
<tr><td rowspan="2">Scheduled start: September 12, 2024

Scheduled end: September 24, 2024</td><td colspan="2">Accountable Manager:</td><td>E. Arpino, K. Azizi</td></tr>
<tr><td colspan="2" rowspan="2">Resources</td><td rowspan="2">A. Marsella</td></tr>
<tr><td>Estimated effort: 45 hours</td></tr>
<tr><td colspan="4"><u>Objectives:</u>

● Decide Requirements and Constraints
● Define Necessary Materials and Software</td></tr>
<tr><td colspan="4"><u>Inputs:</u>

● Market needs
● Competition Technology</td></tr>
<tr><td colspan="4"><u>Tasks:</u>

● Determine Functional requirements
● Determine Control system requirements
● Determine System test requirements
● Determine what materials will be necessary to complete the objectives</td></tr>
</table>

Outputs/Deliverables:
• Finalize Function and Methods of Operations of Device

Table 3.3: Work Package Definition - WBS #3000

Project: RadiationXplorer Radiation Monitoring Drone			
Work Pack Title:	**Preliminary Design and Technology Completion**	**WBS Ref: 3000**	
Sheet: 1 of 1			
Scheduled start: September 26, 2024		Accountable Managers:	A. Marsella, F. Matti
Scheduled end: October 7, 2024		Resources	K. Azizi, M.E. Gorham
Estimated effort: 80 hours			

Objectives:

- Develop and construct the preliminary design

Inputs:

- From Laboratory Technician
- Phase I review
- Requirement and constraints

Tasks:

- Develop a structural and enclosure design
- Develop a hardware design
- Develop a software design
- Create a computer simulation
- Testing the design
- Identify possible improvements

Outputs/Deliverables:
● Documentation of the finalized design ● Data from computer simulation and testing

Table 3.4: Work Package Definition - WBS #4000

Project: RadiationXplorer Radiation Monitoring Drone			
Work Pack Title:	**Final Design and Fabrication**	**WBS Ref: 4000**	
Sheet: 1 of 1			
Scheduled start: October 8, 2024		Accountable Manager:	M.E. Gorham, F. Matti
Scheduled end: October 20, 2024		Resources	E. Arpino, K. Azizi
Estimated effort: 80 hours			
Objectives: ● Construction of Hardware System ● Code Software - Geiger Muller Counter - GPS - SD Card ● Construct Enclosure Structure			
Inputs: ● Data from Geiger-Muller Arduino Library ● Tutorials to incorporate various sensors into Arduino hardware			
Tasks: ● Construct Structure to affix payload to drone ● Assemble the Arduino circuitry ● Compile the code to accompany hardware in completing the objective			

Outputs/Deliverables: ● Completed First Rendering of Device

Table 3.5: Work Package Definition - WBS #5000

Project: RadiationXplorer Radiation Monitoring Drone			
Work Pack Title:	**System Assembly, Integration, Testing**	**WBS Ref: 5000**	
Sheet:	1 of 1		
Scheduled start:	October 21, 2024	Accountable Manager:	E. Arpino, A. Marsella
Scheduled end:	November 30, 2024	Resources	F. Matti
Estimated effort:	200 hours		
Objectives: ● Affix Payload to Drone ● Rigorous, continuous testing and improvements to ensure quality and longevity			
Inputs: ● Radiation Testing from Physics Department ● Flight testing with Drone operator			

Tasks:
 	
● Affix payload to drone
● Software testing, ensure code is successfully compiled
● Hardware testing, ensure wiring is successfully composed
  - Radiation Testing of sensor
● Drone testing
  - Flight test			

Outputs/Deliverables:
● Complete and successfully present final product

4.0 System Overview

4.1 Payload Subsystem

The RadiationXplorer payload subsystem is designed to autonomously monitor radiation levels in hazardous environments. By integrating modular, lightweight, and cost-effective components, the payload accomplishes geospatially accurate data logging and efficient power management, while meeting strict constraints on mass and budget. This section outlines each component's role, functionality, and contribution to the payload's overall mission.

The payload subsystem for the RadiationXplorer drone mission has specific, measurable requirements to achieve reliable, autonomous radiation monitoring and are depicted in Table 4.1. Each requirement focuses on functional, performance, environmental, and budgetary constraints for the payload subsystem.

Table 4.1. Payload Subsystem Requirements

ID	Requirement
RQ-1.1	The payload shall have a total mass of less than 300 grams.
RQ-1.2	The payload shall fit within a 20 cm x 20 cm x 20 cm volume.
RQ-1.3	The Geiger-Müller counter shall detect and record radiation levels with ±10% accuracy.
RQ-1.4	The GPS module shall provide geospatial data with positional accuracy of ±5 meters, updated every second.

ID	Requirement
RQ-1.5	The SD card module shall store a minimum of 1 GB of radiation and GPS data per mission.
RQ-1.6	The payload shall autonomously collect and log data for at least 30 minutes per mission.
RQ-1.7	The ESP32's built-in Wi-Fi transceiver shall provide optional real-time data transmission to the ground station.
RQ-1.8	All payload components shall operate within a temperature range of -10°C to 40°C.
RQ-1.9	Sensitive electronic components shall be shielded from radiation interference.
RQ-1.10	Total expenditure for new payload components shall not exceed $50.
RQ-1.11	The payload's power supply shall support all components continuously for a minimum of 30 minutes.
RQ-1.12	The payload enclosure shall be robust enough to protect components during drone flight and minor impacts.

With the ESP32 at its core, the RadiationXplorer payload subsystem effectively integrates multiple sensors and instruments, each contributing to the mission goals.

ESP32

The ESP32 serves as the central processor, chosen for its dual-core processing, built-in Wi-Fi, and efficient GPIO support, enabling seamless sensor integration, real-time data transmission, and optimized data management within a compact and power-efficient design [1].

- Functionality: The ESP32 centralizes all data acquisition, processing, and transmission functions [1]. Its built-in Wi-Fi handles real-time data streaming, while GPIO and UART interfaces allow seamless integration with the Geiger-Muller counter, GPS, and SD card module for comprehensive radiation monitoring.

- Power Handling: The ESP32's lower power consumption, compared to the Arduino Mega, enhances system efficiency, allowing the drone to operate for longer periods [1]. Its integration minimizes power draw by consolidating processing and Wi-Fi transmission in a single module, optimized to manage multiple components with reduced energy demand.

Geiger-Müller Counter (Radiation Sensor)

The Geiger-Müller counter detects ionizing radiation, a primary mission function, by measuring radiation intensity in counts per minute (CPM).

- Functionality: The sensor's digital output sends pulses to the ESP32 board with each ionizing event, which the board processes and timestamps for post-mission data mapping. This enables accurate radiation level reporting across varied locations.

- Design Considerations: Lightweight and power-efficient, the Geiger-Müller counter fits within the payload's compact design and meets accuracy requirements for hazardous environment applications.

GPS Module (ATGM336H)

The GPS module provides precise geolocation data, essential for mapping radiation intensity across the monitored region.

- Functionality: Outputting positional data (latitude, longitude, altitude) every second, the GPS module communicates with the ESP32 to timestamp location coordinates. This data is synchronized with radiation measurements, ensuring spatial accuracy in radiation distribution mapping.

- Specifications: Operating at 3.3V, with a positional accuracy of ±5 meters, the GPS meets the payload's precision requirements, supporting high-resolution geospatial analysis [1].

SD Card Module

The SD card module offers onboard storage for sensor data, ensuring all radiation and positional information is securely recorded during flight.

- Functionality: Via SPI interface, the SD card logs data from the ESP32, storing each sensor reading in a CSV format for efficient retrieval. This allows for detailed post-flight data analysis, essential for generating radiation maps.
- Power and Size: Operating at 3.3V and with a minimal footprint, the SD card module aligns with the payload's weight and power limitations, optimizing storage without compromising space or energy.

Micro SD Card

- A compact, high-capacity storage medium compatible with the SD Card Module, suitable for continuous data logging over extended monitoring periods.

DC Power Converter

- Converts 9V from the battery to the lower voltage levels required by ESP32 and other components ensuring consistent, safe operation .
- Reduces risk of power surges, which could damage sensitive electronics in the system .

Power Supply (9V Battery)

A 9V rechargeable battery powers the entire system, providing consistent voltage to all components.

- Functionality: The battery connects to the DC power converter, which regulates and distributes power to all sensors and modules at appropriate levels (3.3V or 5V). This configuration ensures that each component receives stable power, supporting extended operational periods.

- Battery Capacity: Sufficient for at least 30 minutes of continuous operation, the battery meets mission duration requirements, ensuring all data is collected and stored.

The block diagram, shown by Figure 4.1, provides an overview of the payload subsystem, highlighting data flow and power management. The ESP32 acts as the central hub, connecting the Geiger-Müller counter, GPS, SD card module, and utilizing its built-in Wi-Fi for data transmission.

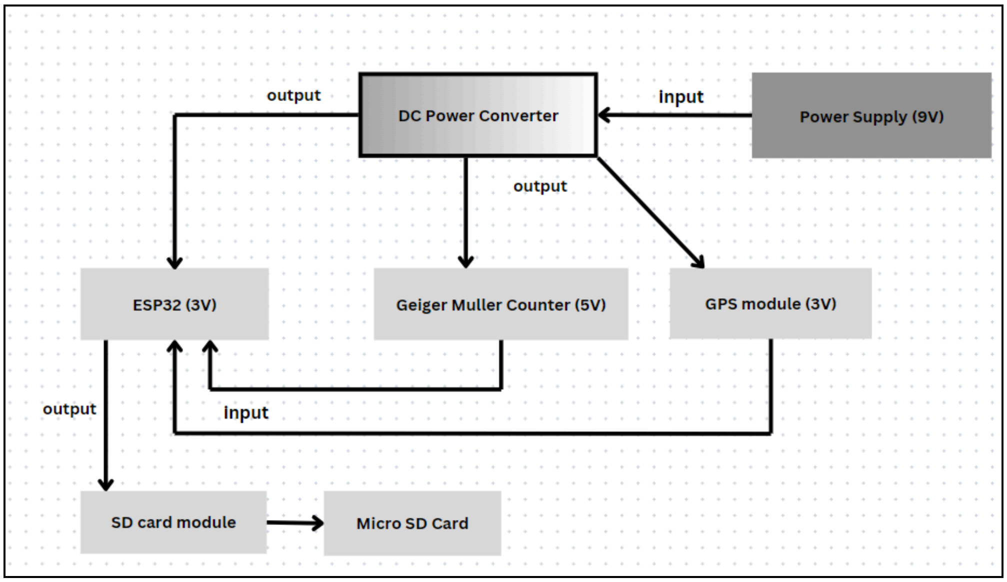

Figure 4.1. Block diagram for payload subsystem.

4.2 Structures and Mechanisms

The body subsystem consists of the plastic housing for all of the electrical components. The enclosure serves three main purposes. One, to attach the payload to the drone. Two, to protect the electrical components from any potential environmental hazards during the flight. Three, to allow for the electrical components, especially the geiger-muller counter, to function and retrieve data properly. Table 4.2 is a list of all the requirements for the body subsystem.

Table 4.2. Body Subsystem Requirements.

ID	Requirement
RQ-5.1	The mass of the product must be less than 300 grams
RQ-5.2	The volume of the product must be less than 20 cm x 20 cm x 20 cm
RQ-5.3	The enclosure must allow unobstructed data transmission and reception for all internal electrical components
RQ-5.4	The enclosure must protect components from environmental hazards

The design for the electrical component enclosure is simple. Through the provided quadcopter base interface, the electrical component enclosure is linked to the bottom of the drone. Figure 4.2 shows a block diagram of the body subsystem.

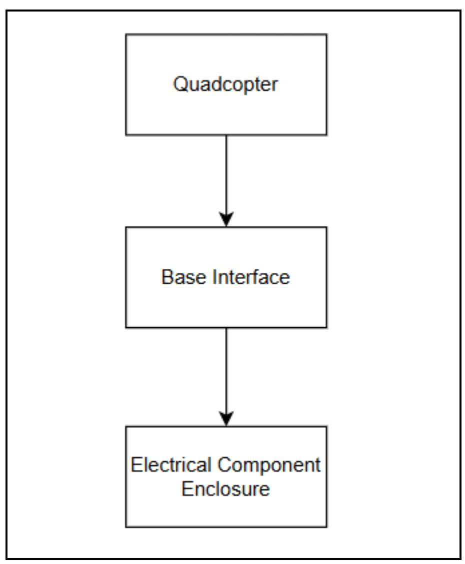

Figure 4.2. Block diagram of Body Subsystem.

4.3 Command and Data Handling (C&DH) Subsystem

The command and data handling subsystem accounts for data processing, data storage, and communication, a crucial subsystem responsible for the completion of the mission's objective. The SD module and micro SD card are responsible for storing data into its memory system to later be transmitted and processed. Additionally, the ESP32 has built-in Wi-Fi responsible for real-time data processing, collecting, and storing data obtained from the sensors. This section will outline the requirements and design of the hardware and software of the command and data handling subsystems.

The command and data handling subsystem shall work together in order to ensure the RadiationXplorer drone operates efficiently while responding to both command and autonomous changes during its mission. The requirements are listed in Table 4.3.

Table 4.3. Command and Data Handling Subsystem Requirements.

ID	Requirement
RQ-4.1	The micro SD card shall have the capacity of 1 GB to store radiation and GPS data per mission.
RQ-4.2	The SD card shall be compatible with the SD module.
RQ-4.3	The data handling shall have efficient algorithms for reading and data wiring.
RQ-4.4	The SD module shall have interface capabilities such as SPI to connect with ESP32
RQ-4.5	The SD module shall have a proper power management circuitry of voltage levels of 3.3 V.
RQ-4.6	The built-in Wi-Fi transceiver shall handle real-time data streaming and transmit to ground station and user device.

RQ-4.7	The data shall be collected from the sensors to be stored on the micro SD card.
RQ-4.8	The built-in Wi-Fi shall output data to the user device.

4.3.1 Hardware

Figure 4.3 shows a block diagram of the command and data handling subsystem and Table 4.4 depicts the BOM for this subsystem.

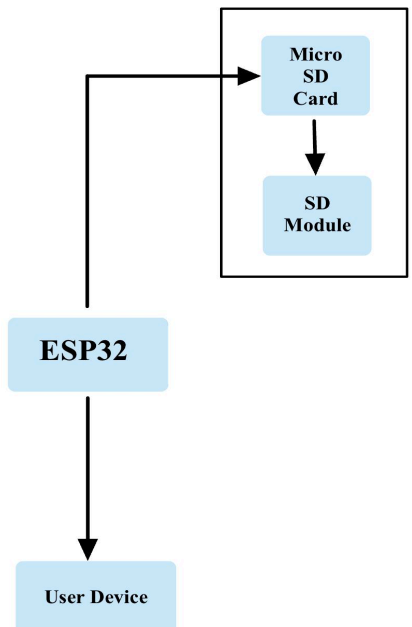

Figure 4.3. Block diagram of command and data handling subsystem.

Table 4.4. Command and Data Handling Subsystem Bill of Materials.

Part	Quantity	Cost
SD Module	1	$11.00
Micro SD Card	1	$3.09
ESP32 Built-in Wi-Fi	1	Borrowed (N/A)

4.3.2 Software

The expected outcome of the code for Command and Data Handling is listed below:

1. Initialize buffers to temporarily hold data and set up the file system on the SD card.
2. While the payload is active:
    2.1. Collect data from the Geiger counter and GPS module.
    2.2. Save this data to the SD card for long-term storage.
    2.3. Forward the data to the Communication subsystem for real-time transmission.
3. If the SD card runs out of space, stop saving new data on the SD card to prevent data corruption.

4.4 Power Subsystem

The power subsystem refers to the source of power and the supply of power to the entirety of the system. This power allows all sensors to operate and data to be collected. The system would not be able to operate to any extent without the power subsystem. The requirements and design features are presented in the following sections.

Considering the power subsystem, its requirements are listed in Table 4.5 to ensure the product is supported by a well designed power system.

Table 4.5. Power Subsystem Requirements.

ID	**Requirement**
RQ-2.1	There shall be enough power to receive, and transmit information.
RQ-2.2	There shall be enough power for the system to operate 30 minutes between charges.
RQ-2.3	The power unit shall not overheat.

Figure 4.4 depicts a block diagram of the power subsystem and Table 4.6 depicts the cost of each component for the power subsystem.

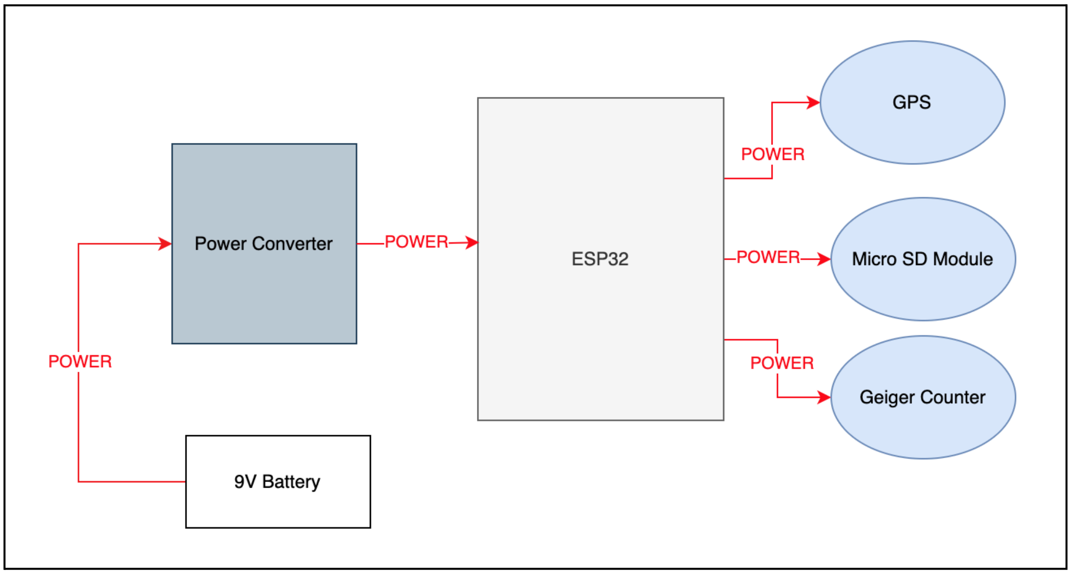

Figure 4.4 Block diagram of the power subsystem.

Table 4.6. Bill of Materials for Power Subsystem.

Part	**Quantity**	**Cost**
9V Battery	1	$1.50
Battery Holder	1	$3.95

4.5 Communication Subsystem

The communication subsystem refers to the wireless communication between the device and the user, in the case of the RadiationXplorer, this is the relay of data via the built-in wifi of the ESP32. The requirements and design features of the communication subsystem are presented in the following sections.

Table 4.7 depicts the requirements for the communication subsystem.

Table 4.7. Communication Subsystem Requirements.

ID	Requirement
RQ-3.1	There shall be a constant line of communication between the drone and the user.
RQ-3.2	The data shall be able to be transmitted at a maximum range of 200m.

Figure 4.5 shows a block diagram of the communication subsystem and Table 4.8 depicts the BOM for the communication subsystem.

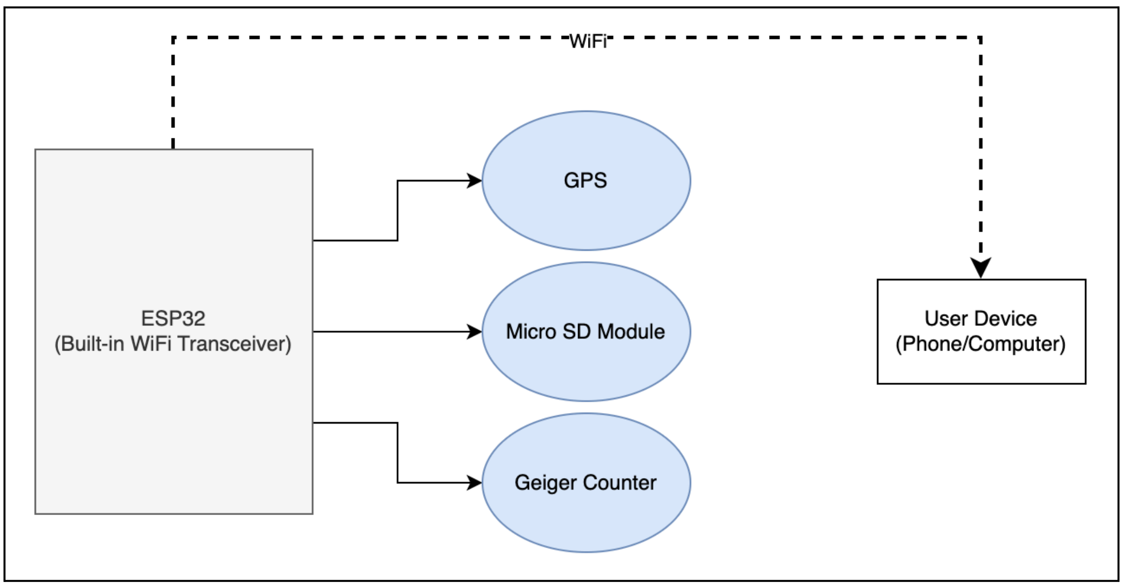

Figure 4.5. Block diagram of the communication subsystem.

Table 4.8. Bill of Materials for Communication Subsystem.

Part	Quantity	Cost
ESP32	1	Borrowed (N/A)
Phone/Laptop	1	Already owned (N/A)

4.6 Ground Station

The Ground Station Subsystem consists of a laptop equipped with both MATLAB software and an SD card slot. The ESP32 receives real-time readings from the Geiger counter and relays this data to the laptop. An SD module also stores all the data on a card, which will be transferred to the laptop after the drone flight. MATLAB is then utilized for data processing, data analysis, and data display. Table 4.9 lists the requirements for the ground station subsystem.

Table 4.9. Ground Station Subsystem Requirements.

ID	Requirement
RQ-6.1	All data from the ESP32 and SD card must be time stamped and synchronized to accurately align GPS and Geiger counter readings.
RQ-6.2	Data processing must include a noise reduction method to improve the accuracy of radiation readings
RQ-6.3	System must detect and log any data discrepancies or errors between real-time ESP32 data and SD card data, with an overall data integrity of at least 95%.
RQ-6.4	GPS and radiation data must be mapped to a geographical overlay in MATLAB to provide a clear visual of radiation intensity across different locations.
RQ-6.5	System must display radiation data and location on a live plot during flight

The interaction of the ground station subsystem with all its components can be visualized in the block diagram in Figure 4.6, with its hardware highlighted in red. Table 4.10 lists the BOM for the ground station subsystem.

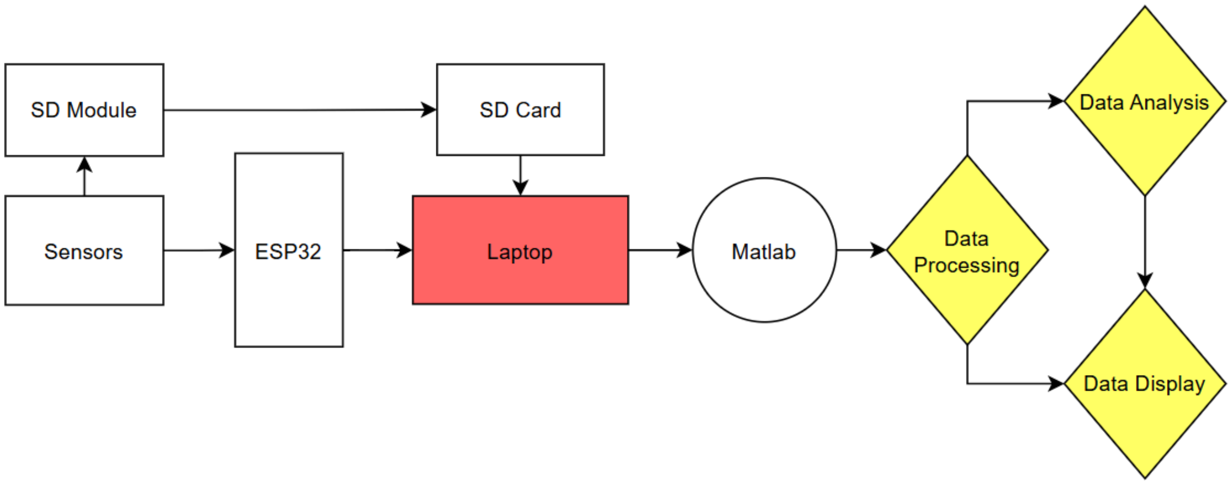

Figure 4.6. Block diagram of ground station subsystem.

Table 4.10. Ground Station Subsystem Bill of Materials.

Name	**Cost**
Laptop (with MATLAB & SD card slot)	0 (owned)

4.7 Bill of Materials (BOM)

Table 4.11 below presents the list of components in the payload along with their price.

Table 4.11. List of Sensors and Instruments in Payload.

Component	**Description**	**Price**
ESP32 Board	Microcontroller with Wi-Fi and Bluetooth for data transmission and sensor integration.	0$ (borrowed)
Geiger-Muller Counter	5V input, 12-30mA current, for radiation detection.	0$ (already own)
GPS Module	Module for real-time geolocation data.	18$

SD Card Module	Storage module for data logging.	11$
Battery	9V rechargeable, 175mAh capacity.	1.50$
Mini SD card	A compact, high-capacity storage medium compatible with the SD Card Module.	3.00$
DC Power Converter	Converts 9V from battery to lower voltage levels.	0$ (already own)

5.0 System Interface

5.1 System Interface Diagram: Data and Power

The interface diagram for the data and power systems shows the high level of the subsystems interfaced. Figure 5.1 shows all the subsystems interfaced together.

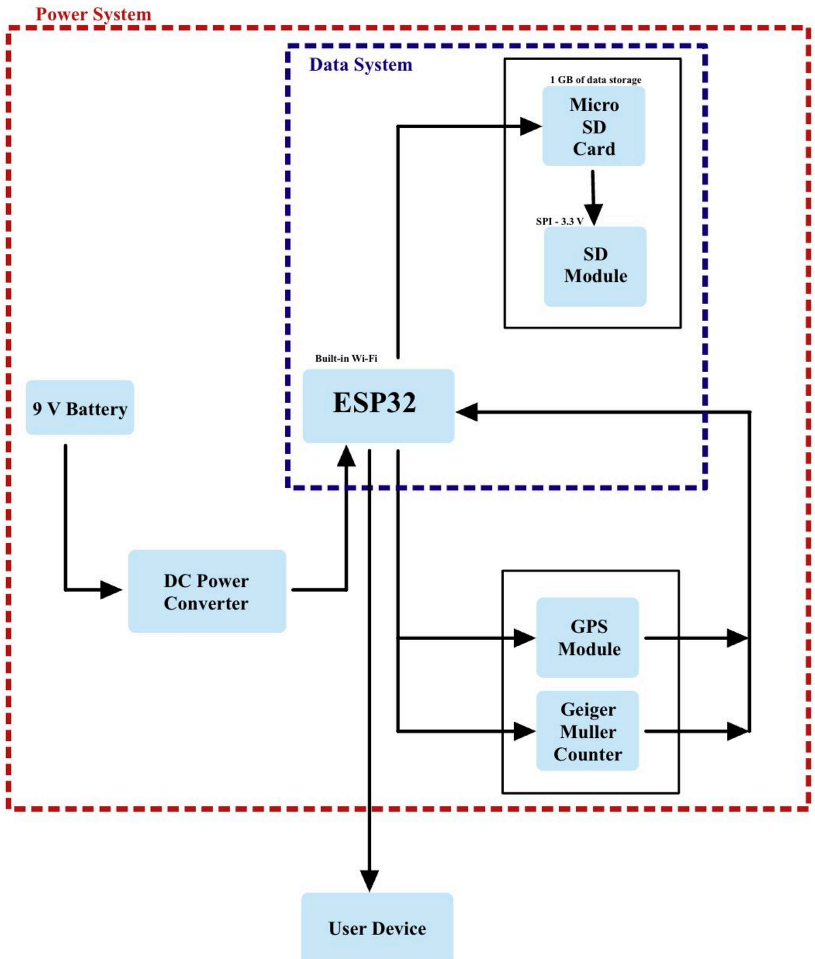

Figure 5.1. System interface diagram of the RadiationXplorer.

5.2 System Interface between Subsystems

An N2 diagram aids in the development of subsystem interfaces [2]. Table 5.1 shows the N2 diagram with all the subsystems listed in Section 5.

Table 5.1. N2 Diagram of all the Subsystems.

Structure	M	M	M	M	M
	Payload	E	E	E	M
		Power	E	E	E
			Communicatio n	E	E
				Command and Data	E
					Ground Station

6.0 Testing

6.1 Payload Subsystem

The payload subsystem for the RadiationXplorer was tested using available resources to validate its design, functionality, and integration with the overall system. While constrained by access to specialized facilities for some tests, practical validation was performed for key components, including real-world testing of the Geiger-Müller counter in a controlled physics lab environment. This section outlines the methods and results of the payload testing process.

The testing plan focused on verifying compliance with the design requirements outlined in the Payload Requirements Matrix (Table 2.2). Tests were performed in a combination of indoor and controlled laboratory environments. The objectives included functional validation of each component, integration testing of the payload subsystem, and simulated mission scenarios.

ESP32 Microcontroller

Objective: Verify core functionality, including data acquisition, processing, and communication with sensors.

Test Method: A test script was uploaded to the ESP32 to interact with the Geiger-Müller counter and GPS module. Wi-Fi transmission was tested within an indoor range of 10–15 meters.

Results: The ESP32 successfully collected, processed, and transmitted data without interruptions.

Geiger-Müller Counter

Objective: Ensure accurate detection of ionizing radiation.

Test Method: The Geiger-Müller counter was tested in the Physics Department lab using radioactive isotopes, including cesium-137 and cobalt-60. Radiation counts were recorded and compared to known activity levels of the sources. Various information was collected from the test as listed below:

- **Cesium 137**, gamma, Nov 1998, 5.0 microCi, 30.2 years (11.55 - 14 cpm/100 our Geiger)
- **Cesium 137**, beta/gamma, Aug 2010, 0.1 microCi, 30.2 years (0.6 - 1.02 cpm/100 our Geiger)
- **Cobalt 60**, gamma, Sept 2019, 1 microCi, 5.27 years (11 - 13.5 cpm/100 our Geiger)
- **Sodium 22**, gamma, Mar 2010, 1 microCi, 2.6 years (0.6 - 1.08 cpm/100 our Geiger)

Results: The counter demonstrated radiation detection within the specified accuracy of ±10%. The readings closely matched the expected radiation levels from cesium-137 and cobalt-60, confirming that the Geiger-Müller counter is fully operational for detecting ionizing radiation in real-world conditions.

Figure 6.1 below shows how the radiation testing was conducted inside the physics lab.

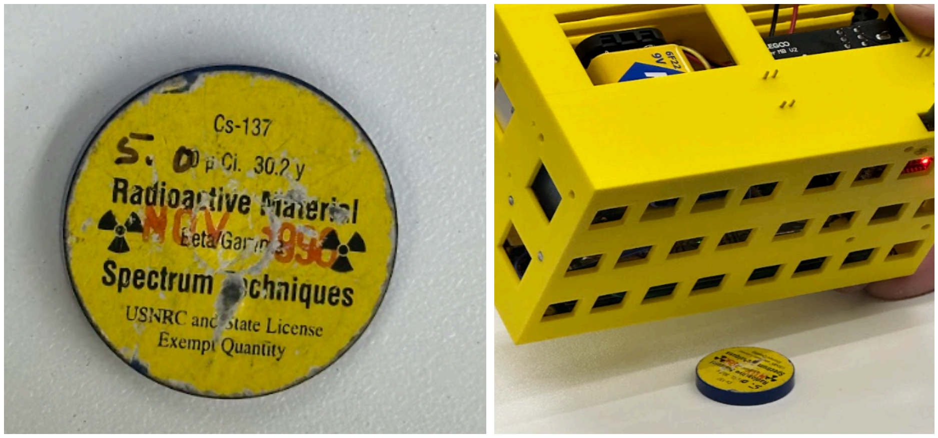

Figure 6.1: Images From Geiger Counter Test Procedure

<u>**GPS Module (ATGM336H)**</u>

Objective: Validate geolocation functionality.

Test Method: The GPS module was tested by walking around with the payload and recording the changes in longitude and latitude values. The logged data was reviewed to confirm that the positional accuracy was within the required ±5 meters.

Results: The GPS module successfully recorded changes in longitude and latitude as expected, with an accuracy of approximately ±5 meters, meeting the design requirements. Figure X shows the final radiation map generated as the payload travelled around the university premises. The colour bar on the side shows the fluctuations in radiation readings while the longitude and latitude can be seen on the x and y axis.

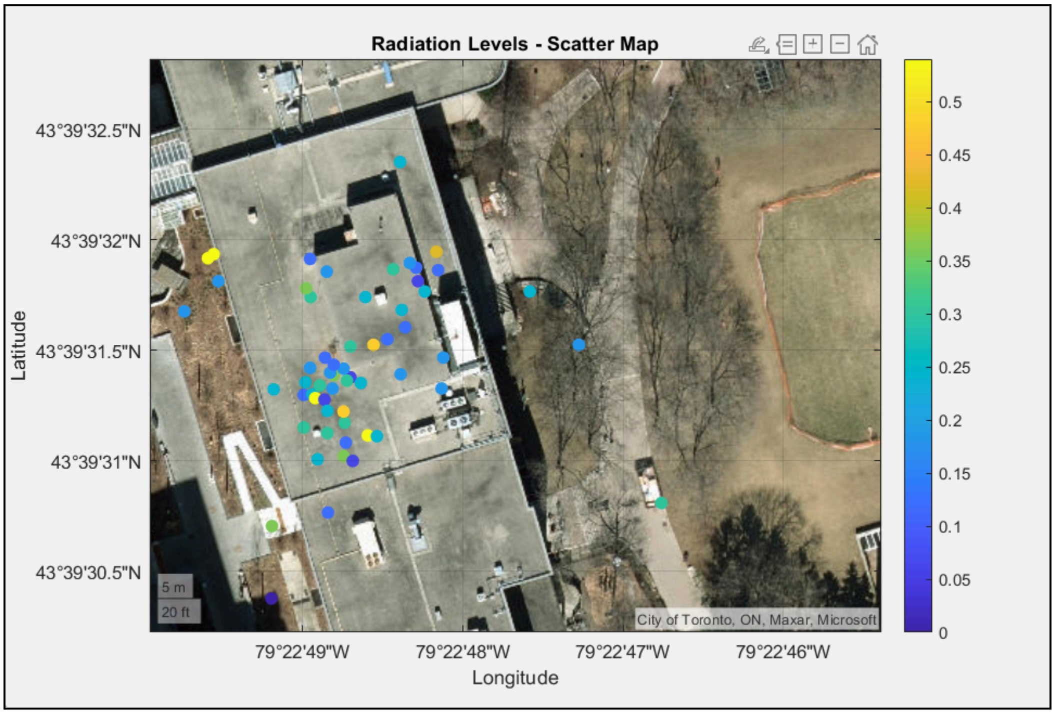

Figure 6.2: GPS Location and Radiation Mapping

Power Supply (9V Battery with DC Converter)

Objective: Verify power stability and sufficiency for a 30-minute mission.

Test Method: The power system was tested under full load with all components active. Voltage and current levels were monitored using a multimeter.

Results: The system required additional current for the wifi transceiver to work as expected. Therefore, a power bank was incorporated into the system which solved the initial power issues.

Payload Enclosure

Objective: Ensure physical protection of components and measure weight.

Test Method: Simple drop tests from a height of 0.5 meters were performed to mimic potential handling impacts. The completed payload was weighed to make sure it was below 300 grams.

Results: The enclosure effectively protected all internal components, with no visible damage or loss of functionality. The payload weighed about 270 grams before the final field test.

6.2 Structures and Mechanisms

The structures and mechanisms were tested in order to verify that the payload meets the requirements for weight and size while ultimately fulfilling its purpose. The following tests were conducted: drop, attachment, and accessibility testing. The following test metrics were used: strength, weight, attachment security, and ease of access.

For strength testing, the payload housing was observed before and after being filled with all the internal components. It was ensured that there was not flexing or cracks in the housing. For the weight testing, the housing was weighed after being fully assembled to ensure the payload started within the 300g limit. For attachment security, screws were used to ensure the housing lid and drone attachment unit was fully attached and secure without any movement between parts. The housing was tested with Peter's drone for assurance. For ease of access, certain slots and holes were added to the housing to ensure ease of access to key components. This was so the Geiger counter could take accurate readings and so that components could be accessed without needing to remove the housing lid each time.

The final housing component met all the requirements. It weighed under 300g and was within the 20x20x20 cm limits. A few small designed adjustments were made such as extra holes being implemented and changing the drone attachment. Figure 6.3 shows an exploded assembly view of the payload including the housing structure.

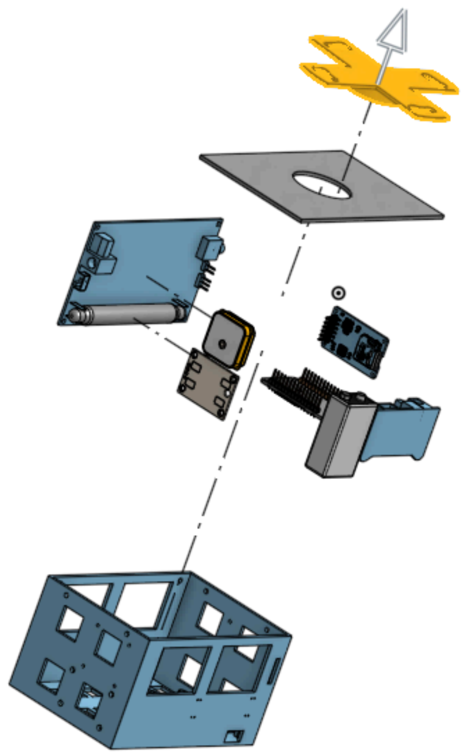

Figure 6.3: Exploded assembly view of the payload including the housing structure.

6.3 Command and Data Handling (C&DH) Subsystem

The command and data handling subsystem would have consisted of the SD module, micro SD card, and ESP32 with the built-in Wi-Fi. During the process of connecting all the hardware and testing the code, it was revealed that the micro SD card and SD module were not connecting efficiently to the entire system nor was it showing any signs of storing data into its memory system to be transmitted and processed. After many attempts, research, and using different methods, it was realized that since the real-time data was working correctly, there was no need for the SD module and micro SD card.

Fortunately, the ESP32 built-in Wi-Fi was working as intended. Once the issue with the power source was resolved, the ESP32 was tested once more. The ESP32 was tested by running the code for a few hours and observing the consistency of the real-time data transported from the Wi-Fi to the ground station. The ESP32 and ground station were connected using a hotspot from a member's phone which transmitted the data from the ESP32 to MATLAB.

Had the issue with the power source been resolved sooner, there might have been a possibility of figuring out how to connect the SD module and micro SD card. However, it was realized that the SD module and micro SD card was not necessary for the field testing. If the design was used for a real-life mission, incorporating the SD module and micro SD card would be essential.

6.4 Power Subsystem

Various power supply and consumption tests were performed on the power subsystem to ensure it could reliably supply sufficient energy to all components of the payload under required operating conditions. Initially, all sensors were powered using a 9V battery and DC converter. During the process of wiring, coding, and testing the hardware it became apparent that the 9V battery would not provide enough power to operate both the ESP32 microcontroller and all sensors; Geiger Counter, GPS module, and SD Card module, and allow for simultaneous operation. This was revealed through attempts to operate the payload that were met with errors, voltage drops under load conditions, instability, and frequent resets of the ESP32. Through various tests and changes to algorithms and wiring it was determined that the 9V battery did not provide enough current to power all of the sensors, thus the power subsystem was not able to achieve the requirements of the payload.

This issue was addressed by swapping the 9V battery out for a USB power bank. A slim and lightweight power bank was able to be installed so that the weight of the system was kept below 300g. This new power method allowed for consistent power to be applied to the system over extended duration, as the power bank was able to store many hours of operational charge. This provided many benefits as it improved the operational duration from what was originally proposed and provided a rechargeable power source.

Comprehensive testing was conducted on the power subsystem with this new power source to validate its performance and ensure all sensors could operate effectively in tandem. Each sensor, Geiger Counter, GPS module, and SD card module, was tested individually and in various combinations to confirm that the power subsystem could effectively handle operation without random voltage fluctuations or performance degradation. In addition to this operational testing, a duration test was performed to determine how long the system could continuously run on a single charge. The device was able to operate successfully for numerous hours with all operational sensors actively engaged. This confirmed that the power bank was an effective power source for the payload.

Further testing could have been performed to explore power optimization techniques, such as sleep modes for unused sensors, which would have effectively extended the runtime of the system. Through the various tests performed it was determined that though the 9V battery proved itself unable to complete the required tasks of the project objective, the USB power bank was an improved solution that ensured stable operations across all sensors and provided the reliability that will be required in real-world employment.

6.5 Communication Subsystem

The communication subsystem consists of a wireless communication between the ESP32 and the ground station device which in this case was MATLAB. While using the 9V battery as the power source. It was difficult to contain the wireless connection between the ESP32 and the user device. Once the power source system was fixed, it became easier to disconnect the USB cord from the ESP32 to the computer and still have the ESP32 transit the real-time data to MATLAB (user device). This was tested by having the payload and user device run for a few hours to make sure there was a constant line of communication between the two. Also, while testing, the payload was in constant movement to ensure the data was able to remain transmitted at different ranges.

6.6 Ground Station

The ESP32 was programmed using Arduino IDE. First, the ground station was tested by making sure Geiger counter readings were being displayed in the serial monitor while the ESP32 was connected to a laptop via a cable. The ESP32's wifi transmitter was tested by sending the radiation data into MATLAB. The GPS module was then tested and the longitudinal, latitudinal, and height measurements were compared to ensure accurate GPS readings.

In the physics lab, the Geiger counter was tested with radioactive materials described in Section 6.1. The radiation readings were displayed in the Arduino IDE serial monitor, with stronger radioactive materials yielding higher numbers. As the Geiger counter approached the radioactive materials, the numbers also displayed higher results in line with our expectations.

While testing the ESP32's wifi transmitter, there were complications using the 9V battery, which is elaborated on in Section 6.4. The wifi transmission did work in the end when using a power bank instead of the 9V battery.

After all the components were tested and data was sent to the ground station, a radiation map was able to be made and is shown by Figure 6.2 in Section 6.1.

6.7 Integrated System (Payload Unit)

The integrated system testing ensured a smooth operation of each subsystem working together. The objective was to validate the device's ability to detect radiation levels, capture geospatial data, store the information locally, transmit the data to a web page, and retrieve the data from the web page using a ground station device. To test the integrated system, the following steps had to be completed:

1. Setting up Esp-32 web server and web page with micro-usb connection:

a. Disconnecting RX and TX wires from GPS Sensor:

i. When uploading code to the Esp-32, it is necessary that any Rx and Tx connections are disconnected.

b. Connecting the payload to the ground station device using the micro-usb port.

c. Updating the local Wi-Fi credentials the Ground Station Device is using into the Arduino IDE code.

i. In order for the Esp-32 to create a web server and web page, the local wifi username and password must be hard-coded and uploaded using Arduino IDE. Not only this, but for the ground station to retrieve the data from the webpage using Matlab, the payload's IP address must be hard-coded into the Matlab code. This is because the IP address becomes the address of the webpage. The IP address is created and assigned by the local Wi-fi.

d. Uploading modified code to the Esp-32.

i. Once the modified code is successfully uploaded, the serial monitor on Arduino IDE displays the device IP address, and this IP address is inputted into the ground station Matlab code.

e. Inputting the Esp-32 IP address into the Ground Station Matlab code.

f. Verifying web-page is set up.

i. There are two ways to ensure the Esp-32 web page is set up. Through loading the webpage manually on the browser of any device, or by running the Ground Station device Matlab code. If the web page is successfully set up, the browser should successfully load a page, or the Matlab program should successfully output the data on the page into the command window.

2. Verifying sensor data collection with micro-usb connection:

a. Checking the web-page for at least 50 seconds:

i. If step (1) is successful - sensor data should be displayed on the webpage. Namely, the radiation level, the latitude, the longitude, and altitude values. This data should also be updated every 10 seconds.

3. Verifying Ground-Station Communication with micro-usb connection :

a. Running the Matlab program:

   i. Sensor data should be displayed clearly with labels in the command window, with values being stored into their own respective vectors.

b. Closing the Matlab program:

   i. Once the Matlab program is terminated, sensor data should be saved in a .mat file.

4. Verifying Integrated System with Independent Power :

a. Switch from micro-usb connection to Ground Station device, to any connection to independent power source:

   i. Repeat steps 1, 2, and 3 to verify sensor data collection, Ground-Station communication, and SD data collection while under independent power.

7.0 Conclusions

The RadiationXplorer project was a successful demonstration of the feasibility of creating a lightweight and cost-effective payload system that is capable of radiation level monitoring and geospatial data collection. Through the seamless integration of a Geiger Counter, GPS Module, SD Card Module and ESP32 Microcontroller, the payload was able to achieve its primary goals of monitoring radiation levels and transmitting them to the user along with GPS coordinates. Where this project fell short was the integration of the SD Card module, due to unforeseen issues with hardware function and wiring the SD Card module was not able to record the data in an external location. This affects the redundancy of the model, as the WiFi transceiver has no backup in case of failure, however, this does not negate the success of the primary function of the payload. The user is still able to receive real-time data which can be mapped through MATLAB and then stored internally on the receiving device.

Looking ahead, further optimization of power consumption and storage capabilities could extend the system's utility for longer missions and broader environmental applications. Overall, the RadiationXplorer's innovative use of affordable, commercially available components made the system highly accessible and reproducible for applications in hazardous environments.

References

[1] All About Circuits. (2015). *Power calculations*. Retrieved January 20, 2015, from All About Circuits: http://www.allaboutcircuits.com/vol_1/chpt_5/5.html

[2] K. Dev Kumar. (2024) *Lecture 3 System Engineering*. Retrieved September 17, 2024, from Toronto Metropolitan University: https://courses.torontomu.ca/content/enforced/929086-aer817_f24_01/AER817_Lecture_Sept%2017_2024.pdf?ou=929086.

[3] Analytical Graphics, Inc. (2005). *STK/PRO Tutorial.* Retrieved October 15, 2014, from University of Colorado: http://www.colorado.edu/engineering/ASEN/asen3200/labs/proTutorial.pdf

Appendix A: C&DH Code

```
% Load data from the file

load('collecteddata.mat'); % Assuming 'collectedData' is loaded

% Remove the first 100 rows

collectedData(1:230, :) = [];

% Remove rows where columns 3, 4, or 5 have a value of 0

validRows = collectedData(:, 3) ~= 0 & collectedData(:, 4) ~= 0 &
collectedData(:, 5) ~= 0;

filteredData = collectedData(validRows, :);

% Extract data

latitudes = filteredData(:, 3);

longitudes = filteredData(:, 4);

radiationLevels = filteredData(:, 2);

% Create a map figure

figure;

geoscatter(latitudes, longitudes, 50, radiationLevels, 'filled');

colorbar; % Show color scale for radiation levels

% Adjust the color bar scale to show better variation

caxis([min(radiationLevels), max(radiationLevels)]);

% Add title and set basemap

title('Radiation Levels - Scatter Map');

geobasemap('satellite'); % Set the basemap
```

Appendix B: Payload Code

```
#include <WiFi.h>

#include <WebServer.h>

#include <TinyGPS++.h>


//****************************//

//          Wi-Fi Setup       //

//****************************//


const char* ssid = "username";

const char* password = "password";


//****************************//

//     GPS Configuration      //

//****************************//


// Define the RX and TX pins for Serial 2

#define RXD2 16

#define TXD2 17

#define GPS_BAUD 9600
```

```
// The TinyGPS++ object

TinyGPSPlus gps;


// Create an instance of the HardwareSerial class for Serial 2

HardwareSerial gpsSerial(2);


//*****************************//

//     Sensor Configuration    //

//*****************************//


// Geiger Counter Configuration

const int geigerPin = 32;

volatile int count = 0;  // Radiation count from Geiger counter

unsigned long lastMillis = 0;  // For timing 10-second intervals

float countsPerMinute = 0.0;  // Radiation counts per minute

float radiationLevel = 0.0;  // Radiation level (calibration required)


//*****************************//

//          Web Server         //

//*****************************//



// Create a web server object on port 80
```

```
WebServer server(80);


//****************************//
//        Functionality       //
//****************************//


// Function to calculate the radiation level
float readSensor() {
  if (millis() - lastMillis >= 10000) {  // Every 10 seconds
    countsPerMinute = (float)count * 6;  // Convert counts to CPM
    radiationLevel = countsPerMinute / 100.0; // 1 CPM ≈ 0.01 µSv/h
    count = 0;  // Reset the count for the next interval
    lastMillis = millis();  // Update the lastMillis timestamp
  }
  return radiationLevel;  // Always return the last calculated value
}


// Function to format GPS and sensor data for server output
String getSensorAndGPSData() {
  float latestRadiation = readSensor();
```

```
  String latitude = gps.location.isValid() ? String(gps.location.lat(), 6)
: "0.000000";

  String longitude = gps.location.isValid() ? String(gps.location.lng(),
6) : "0.000000";

  String altitude = gps.altitude.isValid() ? String(gps.altitude.meters(),
2) : "0.00";


  // Combine all values into one line

  return String(latestRadiation, 2) + " " + latitude + " " + longitude + "
" + altitude;

}


// Function to handle HTTP requests to the root URL

void handleRoot() {

  String response = getSensorAndGPSData();

  server.send(200, "text/plain", response);

}


// ISR Function to count Geiger events

void countRadiation() {

  count++;

}


//*****************************//
```

```
//            Setup                //

//*****************************//


void setup() {

  Serial.begin(115200);


  // Connect to Wi-Fi

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(1000);

    Serial.println("Connecting to WiFi...");

  }

  Serial.println("Connected to WiFi");

  Serial.print("IP Address: ");

  Serial.println(WiFi.localIP());


  // Start the server and define routes

  server.on("/", handleRoot);

  server.begin();

  Serial.println("HTTP server started");


  // Initialize Geiger counter
```

```
  pinMode(geigerPin, INPUT);

  attachInterrupt(digitalPinToInterrupt(geigerPin), countRadiation,
RISING);

  Serial.println("Geiger-Muller Radiation Detector Initialized");


  // Initialize GPS

  gpsSerial.begin(GPS_BAUD, SERIAL_8N1, RXD2, TXD2);

  Serial.println("GPS Module Initialized");

}


//*****************************//

//            Loop             //

//*****************************//


void loop() {

  // Handle incoming client requests

  server.handleClient();


  // Read GPS data

  while (gpsSerial.available() > 0) {

    gps.encode(gpsSerial.read());

  }
```

Appendix C: Ground Station Code

```
% ESP32 IP address and port
esp32IP = '172.20.10.3'; % ESP32's IP address
esp32Port = 80;
% Initialize the matrix to store all data
collectedData = []; % Columns: [timestamp, radiation, latitude, longitude, altitude]
% Continuously fetch data every 10 seconds
while true
  try
     % Fetch the latest log from the ESP32
     logData = webread(['http://' esp32IP ':' num2str(esp32Port)]);

     % Split the data into separate components by space
     dataParts = str2double(strsplit(strtrim(logData)));

     % Check if the data is valid
     if numel(dataParts) == 4
        % Extract the radiation, latitude, longitude, and altitude values
        radiationValue = dataParts(1);   % Radiation level
        latitudeValue = dataParts(2);    % Latitude
        longitudeValue = dataParts(3);   % Longitude
        altitudeValue = dataParts(4);    % Altitude
```

```
        % Get the current timestamp
        timestamp = datetime('now');
        timestampNumber = datenum(timestamp); % Convert to numeric format for storage

        % Append the new data as a row in the matrix
        collectedData(end + 1, :) = [timestampNumber, radiationValue, latitudeValue, longitudeValue,
altitudeValue];

        % Output the new values every time data is fetched
        fprintf('Timestamp: %s | Radiation Level: %.2f µSv/h | Latitude: %.4f | Longitude: %.4f | Altitude:
%.2f meters\n', ...
            timestamp, radiationValue, latitudeValue, longitudeValue, altitudeValue);

        % Save the collected data matrix into a .mat file
        save('collectedData.mat', 'collectedData');
    else
        warning('Received data is incomplete or malformed: %s', logData);
    end

  catch ME
    % Handle errors gracefully
    warning('Failed to fetch data: %s', ME.message);
  end
```

```
    % Pause before the next request (10 seconds)

    pause(10);

end
```