ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Drone Reconnaissance

Team #100

ANNA SAKO (sako2@illinois.edu)
ELIJAH SUTTON (esutton3@illinois.edu)
JAMES TANG (jhtang2@illinois.edu)

TA: Lukas Dumasius

May 6, 2026

Abstract

This project develops a low-cost, modular aerial reconnaissance platform that collects flight data, communicates with a ground station, and supports future expansion into advanced sensing and autonomy. Hazardous reconnaissance tasks require systems that reduce operator risk while remaining practical to deploy in constrained environments. To address this need, the system uses a manually controlled drone platform with onboard Global Positioning System (GPS) and inertial measurement unit (IMU) sensing to collect position, motion, and orientation data during flight. An onboard microcontroller unit (MCU) interfaces with the flight controller and supports telemetry transfer through a long-range radio (LoRa) communication link. The ground station receives and displays flight data, including mapped coordinate points for post-flight path reconstruction. The final deliverables include a functional drone platform, GPS and IMU data collection, MCU-to-flight-controller communication, LoRa-based drone-to-ground-station communication, map-based visualization of recorded coordinates, and a modular hardware and software framework for future autonomous flight, radar feedback, and expanded reconnaissance visualization.

Contents

1 Introduction

Hazardous or unstable environments often require preliminary field data collection before personnel can safely enter the area. In applications such as warzones, damaged infrastructure sites, or unfamiliar terrain, direct human surveying can be dangerous, costly, and inefficient. These challenges create a need for a remote reconnaissance platform capable of gathering useful environmental data while minimizing operator exposure to unsafe conditions.

1.1 Problem Statement

The problem addressed by this project is the lack of a low-cost, easily deployable reconnaissance platform for collecting field data in environments that may be hazardous or inaccessible to human operators. Existing reconnaissance methods often rely on expensive equipment, direct human involvement, or highly complex systems that are impractical for rapid deployment and implementation within a semester-scale prototype.

1.2 Proposed Solution

The proposed solution is a semi-autonomous drone-mounted ground-penetrating radar platform intended for mine detection and general reconnaissance applications in hazardous environments. The system is designed to provide a lower-cost and more operationally accessible alternative to traditional reconnaissance approaches while reducing the need for direct human involvement in dangerous areas. Additionally, the platform emphasizes modularity and scalability, enabling future expansion of sensing, communication, and autonomous navigation capabilities while building upon existing military reconnaissance concepts and technologies.

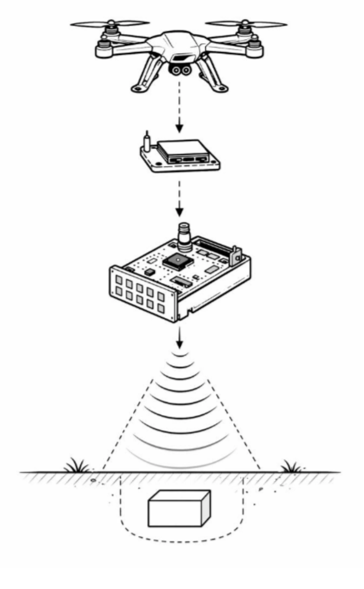


Figure 1: Proposed Solution Concept Art

2 Design & Functionality

The design consists of 3 main subsystems: drone, radar, and ground station. Each of these subsystems are integrated together using a micro-controller on our custom PCB so that we can establish seamless used control over each system from the ground station. Each of these 3 subsystems were developed independently and then combined together for the final product.

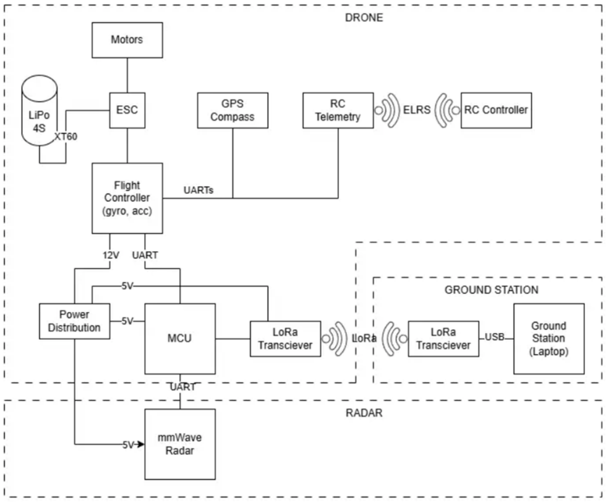


Figure 2: Block Diagram System Overview

2.1 Drone

The drone accomplishes 3 main tasks. It achieves flight, enables flight commands from the ground station, and sends telemetry data to the ground station.

There were a few different criteria by which we made all of our design decision.

- Cost - The design should be as cheap as possible. This solution is inherently expensive since it requires both a drone and radar which are known to be costly. Any money we can save is very valuable.
- Modular - Most of the design was chosen at the beginning with the understanding that changes and problems would come up as the semester progressed. Making the design as modular as possible allowed us to pivot when necessary alternative solutions.
- Documented - This project involves integrating together many different complex and obscure systems. The more documented the components in our design are, the faster we can progress towards a working solution.

The first of these tasks, which is also the most important, is achieving flight. This is accomplished with a quadcopter drivetrain which consists of a battery, motor controller, flight controller (with gyroscope + accelerometer), motors, and propellers. These are generally paired together in different drone platforms based on propeller size. Smaller propellers (4/5″) are used for racing while larger propellers (7/10″) are used for carrying payloads and extended flight. Although in practice, the 7″ or 10″ platforms would be used for an ideal solution, due to our primary criteria being cost, we decided to use the cheaper and smaller 5″ design. This also makes our design more modular and documented since the 5″ platform is the most popular.

Although flight dynamics can be calculated by hand, because of the complexity of real world conditions and inconsistency with parts its much more common to use online calculators that account for these. We used eCalc which takes each part of the drivetrain and creates a report on flight dynamics [1]. Our components were selected by a process of iterating through various cheap and available parts in the calculator and maximizing flight time with an estimated weight of 500g.

The final drivetrain components used were the following:

- Battery - Ovonic LiPo 4S 1550mAh 100C 14.8V XT60
- Motor Controller - Betaflight 60A ESC
- Flight Controller - Betaflight F722 Beastfpvf722
- Motors - AKK RS2205 2300kV
- Propellers - 5″ Triblade 5140

This drivetrain has an estimated 11.7 thrust-to-weight ratio, an estimated flight time of 9mins, and a peak efficiency around 20mph.

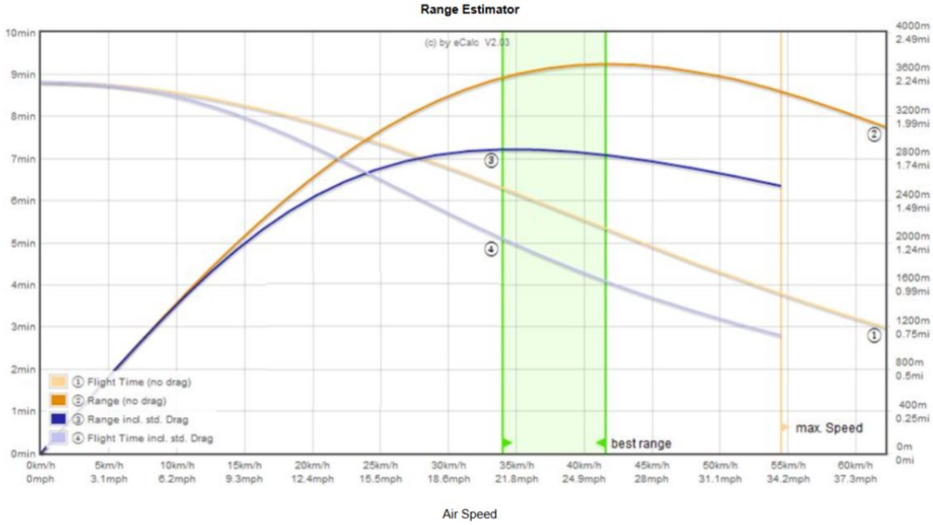


Figure 3: Drone Drivetrain Dynamics Plot

The second task of the drone is to enable flight commands from the ground station. Our original assumption was that since there are many flight controllers using the F722 chip as ours does and since the connection to the flight controller to send flight commands is UART (which is very low level) that we would be able to get the cheapest F722 flight controller available and it would work with all relevant software. This was a mistake that ended up costing a lot of time and effort. Although the cheap flight controller had all of the same hardware as the others for half the price, it still had to be configured with software that needs firmware for that specific flight controller. This meant that instead of using configuration software such as arduflight (specifically designed for interfacing with a flight controller from an MCU) or INAV (supports autonomous control and routing), we only had a single option which was betaflight (used for racing only with no automation support). This meant that instead of the flight controller knowingly interfacing with the microcontroller, we had to trick it into interfacing with the microcontroller. There were a few different ways to do this: MSP protocol which is antiquated and supports only telemetry rather than flight commands, MAVLINK protocol which is designed for two-way ground station communication, and CRSF crossfire protocol which is designed for RC remotes to use when sending joystick inputs. We tried both MAVLINK and CRSF using popular arduino libraries but neither worked. Since the libraries didn't work and we didn't want to write our own libraries we ended up using our microcontroller as a pass-through for the RC receiver which already had CRSF signals. This meant that instead of having to send complex signals written by our microcontroller, all we needed to do was modify the signals sent by the RC receiver and send them to the flight controller. This works perfectly and allows us to have very low level control over the inputs of the drone flight.

The third task of the drone is to send telemetry data back to the ground station. The first part of this design is deciding what telemetry information is necessary for the ground station operator to make flight decisions. We ultimately decided on the following telemetry data:

- IMU - The inertial measurement unit gave us gyroscope and accelerometer data. Although the data rate sent to the ground station would not be fast enough to make quick adjustments, the IMU data gives us an idea of the orientation of the drone in the air.
- GPS - The GPS gives us latitude, longitude, altitude, and magnetometer (compass) data. Although it is only accurate within a couple of meters, this data is vital for informing long autonomous flights.
- Lidar - The lidar tells us the distance of the drone from the ground. This was a later addition to the design that was very vital to autonomous flight. Although the drone could blindly fly patterns laterally, it had no way without the lidar to maintain its height. Although this information is sent back to the ground station. It is first sent to the microcontroller where it is used in a control loop to set the thrust.

Once we decided the telemetry information that is pertinent to our goals, we had to decide how to send it. Our original design for this was very simple. Since we had a UART connection to the flight controller we were able to send CLI commands requesting various telemetry packets. However, this design was changed when we configured the microcontroller to send commands to the flight controller with CRSF. CRSF devices automatically get sent all available telemetry data in set intervals. This meant that once we had the CRSF set up, all we had to do was listen to the telemetry signals sent.

The following figure shows a picture of the final drone with all the drivetrain components, the telemetry components, and the ground station/PCB components attached.

Figure 4: Final Drone Construction

2.2 Radar

The first task of the drone was to configure the drone radar subsystem. The radar subsystem had multiple configuration settings including maximum range. The subsystem able was to identify a metallic object of roughly diameter 10 cm on the ground. The radar used the AWR6843ISK Texas Instruments module for object detections.

2.2.1 Design Description & Justification

We used the AWR6843ISK because it is capable of taking point cloud measurements from a single sensing position. This reduces localization constraints on the drone compared to other radar systems that require multiple radar measurements from different positions. The goal of the project was to maintain a modular system architecture, and this design choice reduced the requirements on the drone subsystem for precise localization and improved performance in environments in external disturbances, such as wind.

The AWR6843ISK performed all radar DSP processing on chip and returned data in the form of range-azimuth heatmaps and point cloud data, providing a list of different detected object positions.

The initial plan was to integrate the AWR6843ISK with the host MCU, using a fanout board to expose the MISO, SPI, and MOSI pins from the 60-pin connector on the radar board to the MCU.

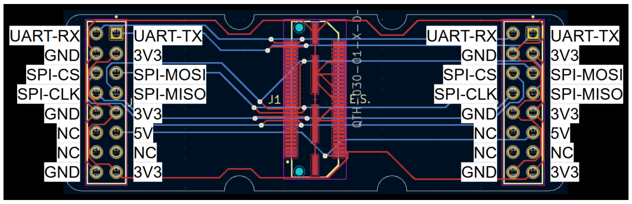


Figure 5: AWR6843ISK Fanout Board

2.2.2 Equations & Simulations

We integrated the AWR6843ISK with a host PC to detect a small object on the ground using a range-azimuth visualization.

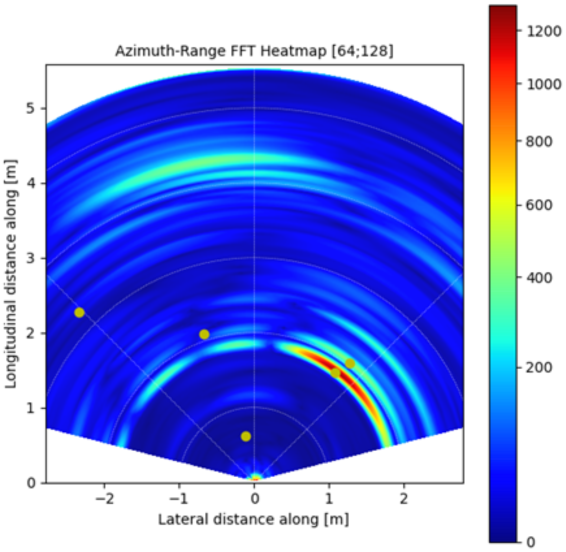


Figure 6: Range Azimuth Plot

The diagram shows an object at an angle of approximately 45 degrees relative to the radar module when the system was connected to the host PC. The range-azimuth plot is overlaid with yellow points representing detected targets within the radar's field of view. The AWR6843ISK provides an approximate 60° field of view in azimuth. Most of the yellow points correspond to noise, while a small cluster aligns with the red-highlighted detection corresponding to the target.

The AWR6843ISK operates in the 60 GHz frequency band, corresponding to a wavelength of approximately 5 mm. The attenuation constant α of a material is

$$\alpha = \omega \sqrt{\frac{\mu\epsilon}{2}\left[\sqrt{1+\left(\frac{\sigma}{\omega\epsilon}\right)^2}-1\right]}$$

where α is the attenuation constant, $\omega = 2\pi f$ is the angular frequency, and μ, ϵ, and σ are the permeability, permittivity, and conductivity of the material, respectively.

For low-loss dielectrics such as dry sand, where $\sigma \ll \omega\epsilon$, this expression simplifies to:

$$\alpha \approx \frac{\sigma}{2}\sqrt{\frac{\mu}{\epsilon}}.$$

[2]

Thus, in low-conductivity materials such as dry sand, attenuation is primarily controlled by conductivity rather than frequency, allowing greater penetration of 60GHz signals through dry sand. In a real-world test of the AWR6843ISK, the radar would work best in sandy or desert environments. For greater ground penetration, a lower frequency radar would be needed.

2.2.3 Design Alternatives

One of the issues was the failure of the AWR6843ISK at the time of demo. The interface of the AWR6843ISK chip is primarily used to connect to other Texas Instruments chips, such as the MMWaveICBoost evaluation board. The AWR6843ISK primarily sends point cloud data in SPI format, and one of the challenges was finding the documentation to decode the AWR6843ISK data format.

One of initial goals of the AWR6843ISK radar was to integrate with the sixty pin connector on the main MCU board, and we have had multiple board failures throughout the course of the project, that caused delays to the integration of the AWR6843ISK and MCU, even though integration of the AWR6843ISK with the host PC was completed.

2.3 Ground Station

The ground station performs two main tasks. It uses the custom PCB to integrate each on-board component together and it provides intuitive two-way communication between the user and the drone. There were a few different criteria by which we made all of our design decisions.

- Reliability - This was our main concern. The drone involved complex flight of expensive and vital components. Any kind of failure could lead to the complete destruction of the project. This meant that we wanted the communication to work over long distances and through buildings. It also meant that we didn't was the user commands to be so vital to the drone flight that if a problem were to occur it leads to a complex loss of control.
- Transmission rate - We didn't have much data to send because telemetry information, flight commands, and processed radar data can be simplified a lot. Even so, it was important that all of this data would be communicated between the ground station and the drone in a small time frame so that live control from the ground station is very reasonable.

- User friendly - This design has to take lots of complex data and present it to an operator who responds with complex flight instructions. It is the job of the software to make that experience as intuitive and seamless as possible.

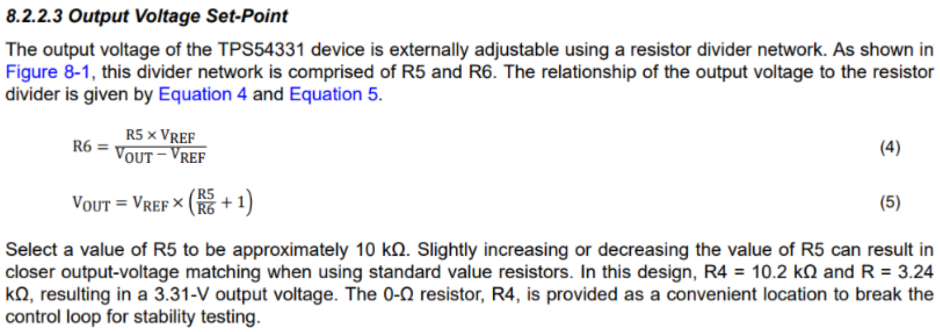
The first of the ground station tasks is complete integration of all on-board components. This was done with our custom PCB microcontroller.

The first part of PCB creation is MCU selection. We used the following criteria to select the MCU:

- Very Fast - must process data from many sources including potentially IQ data from radar
- RAM - must be able to store data from radar to process it
- High baud rate - must be able to communicate with the drone as fast as possible for autonomous control
- UARTS - must have many UART connections
- USB support - for ease of use and programming
- Cheap - most MCUs are very cheap but cheaper is still better

We found that the STM32H753IIT6 fit these criteria best. It runs at 550MHz, it has 16KB of RAM, it has a max baud rate of 12.5MB/s, it has 9 UARTS, USB support, and is a maximum of $18. This excels in nearly every criteria. The next part of the PCB design is power management. The entire circuit and system is attached to the drone. This means that all power comes from the drone's LiPo battery and is dropped down with a buck to 5V and then with an LDO to 3V3. The custom PCB then routes the necessary power to each component. There were two calculations necessary for the power management circuit design[3]. The first was tuning the buck to 5V. This was done using the buck tuning equation below.

$$V_{\text{out}} = V_{\text{ref}} \left(\frac{R_5}{R_6} + 1 \right) = 0.8 \left(\frac{52.3\text{k}}{10\text{k}} + 1 \right) = 4.984\text{V}$$



8.2.2.3 Output Voltage Set-Point

The output voltage of the TPS54331 device is externally adjustable using a resistor divider network. As shown in Figure 8-1, this divider network is comprised of R5 and R6. The relationship of the output voltage to the resistor divider is given by Equation 4 and Equation 5.

$$\text{R6} = \frac{\text{R5} \times \text{V}_{\text{REF}}}{\text{V}_{\text{OUT}} - \text{V}_{\text{REF}}} \quad (4)$$

$$\text{V}_{\text{OUT}} = \text{V}_{\text{REF}} \times \left(\frac{\text{R5}}{\text{R6}} + 1 \right) \quad (5)$$

Select a value of R5 to be approximately 10 kΩ. Slightly increasing or decreasing the value of R5 can result in closer output-voltage matching when using standard value resistors. In this design, R4 = 10.2 kΩ and R = 3.24 kΩ, resulting in a 3.31-V output voltage. The 0-Ω resistor, R4, is provided as a convenient location to break the control loop for stability testing.

Figure 7: Buck Tuning Equation

The second was for resistor selection for LED indicators at each voltage level. We used ohms law to get our resistor values. $R = \frac{V_s - V_f}{I_f}$ Our chosen LED has a voltage drop of 2V and operates at 2mA. I then put each voltage level into the equation and chose standard resistors close to the desired values.

$$R = \frac{12\text{V} - 2.0\text{V}}{0.002\text{A}} = \frac{10\text{V}}{0.002\text{A}} = 5000\Omega \approx 4.7\text{k}\Omega$$

$$R = \frac{5\text{V} - 2.0\text{V}}{0.002\text{A}} = \frac{3\text{V}}{0.002\text{A}} = 1500\Omega = 1.5\text{k}\Omega$$

$$R = \frac{3.3\text{V} - 2.0\text{V}}{0.002\text{A}} = \frac{1.3\text{V}}{0.002\text{A}} = 650\Omega \approx 680\Omega$$

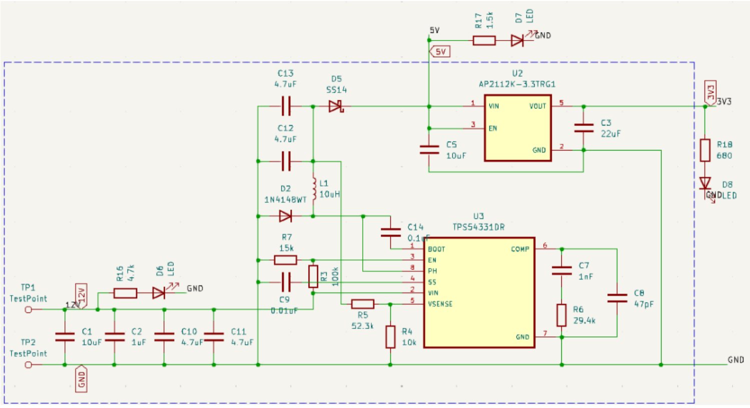


Figure 8: Power Management Circuit

The next part of the PCB design was wiring the programming ports and component interfaces. The complete schematic is shown in the figure below. This includes the SWD interface for the ST-Link, the USB-C connection, the LoRa, flight controller, radar, and additional pins [4].

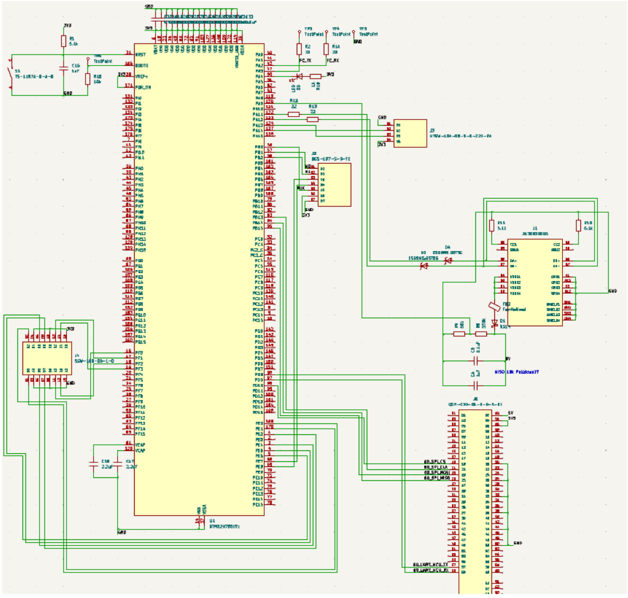
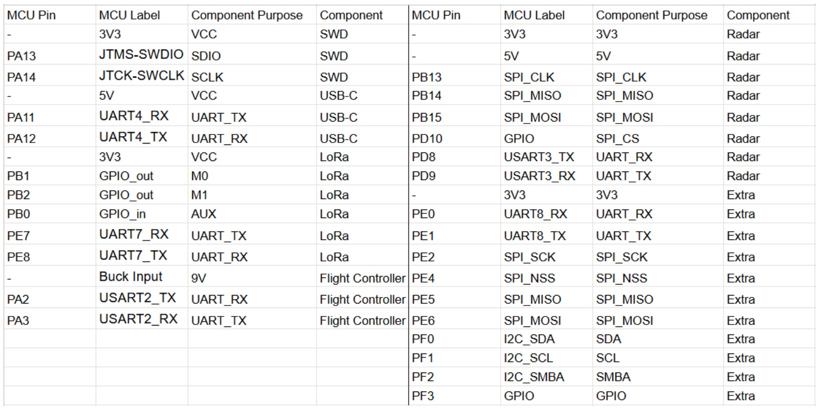


Figure 9: PCB Schematic



MCU Pin	MCU Label	Component Purpose	Component	MCU Pin	MCU Label	Component Purpose	Component
-	3V3	VCC	SWD	-	3V3	3V3	Radar
PA13	JTMS-SWDIO	SDIO	SWD	-	5V	5V	Radar
PA14	JTCK-SWCLK	SCLK	SWD	PB13	SPI_CLK	SPI_CLK	Radar
-	5V	VCC	USB-C	PB14	SPI_MISO	SPI_MISO	Radar
PA11	UART4_RX	UART_TX	USB-C	PB15	SPI_MOSI	SPI_MOSI	Radar
PA12	UART4_TX	UART_RX	USB-C	PD10	GPIO	SPI_CS	Radar
-	3V3	VCC	LoRa	PD8	USART3_TX	UART_RX	Radar
PB1	GPIO_out	M0	LoRa	PD9	USART3_RX	UART_TX	Radar
PB2	GPIO_out	M1	LoRa	-	3V3	3V3	Extra
PB0	GPIO_in	AUX	LoRa	PE0	UART8_RX	UART_RX	Extra
PE7	UART7_RX	UART_TX	LoRa	PE1	UART8_TX	UART_TX	Extra
PE8	UART7_TX	UART_RX	LoRa	PE2	SPI_SCK	SPI_SCK	Extra
-	Buck Input	9V	Flight Controller	PE4	SPI_NSS	SPI_NSS	Extra
PA2	USART2_TX	UART_RX	Flight Controller	PE5	SPI_MISO	SPI_MISO	Extra
PA3	USART2_RX	UART_TX	Flight Controller	PE6	SPI_MOSI	SPI_MOSI	Extra
				PF0	I2C_SDA	SDA	Extra
				PF1	I2C_SCL	SCL	Extra
				PF2	I2C_SMBA	SMBA	Extra
				PF3	GPIO	GPIO	Extra

Figure 10: PCB Peripheral Wiring

The last part of the PCB design was layout. Layout design consisted of adding a ground plane, increasing trace thickness for power traces, and ensuring that no added components mechanically collide. The finished PCB is shown below in a 3D rendering.

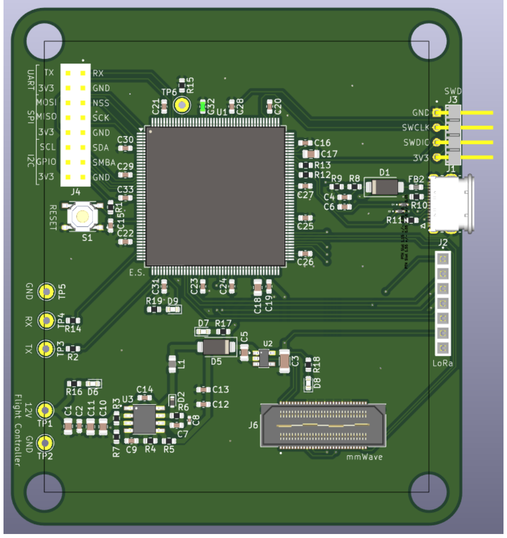


Figure 11: PCB 3D Rendering

Besides the PCB which interfaced with the on-board components, there is also ground station design housed on a laptop. Our goal was to make the drone as configurable and intuitive as possible for the pilot experience. This meant having the pilot operate the drone primarily with an RC remote while watching their laptop to update commands and view telemetry information. This was achieved by putting a matching LoRa transceiver on the ground station side.

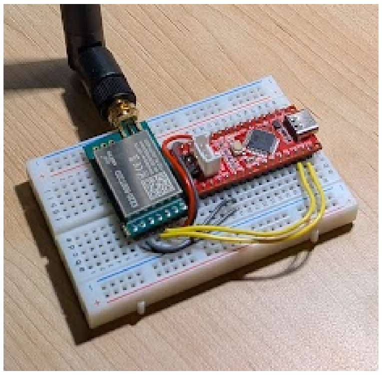
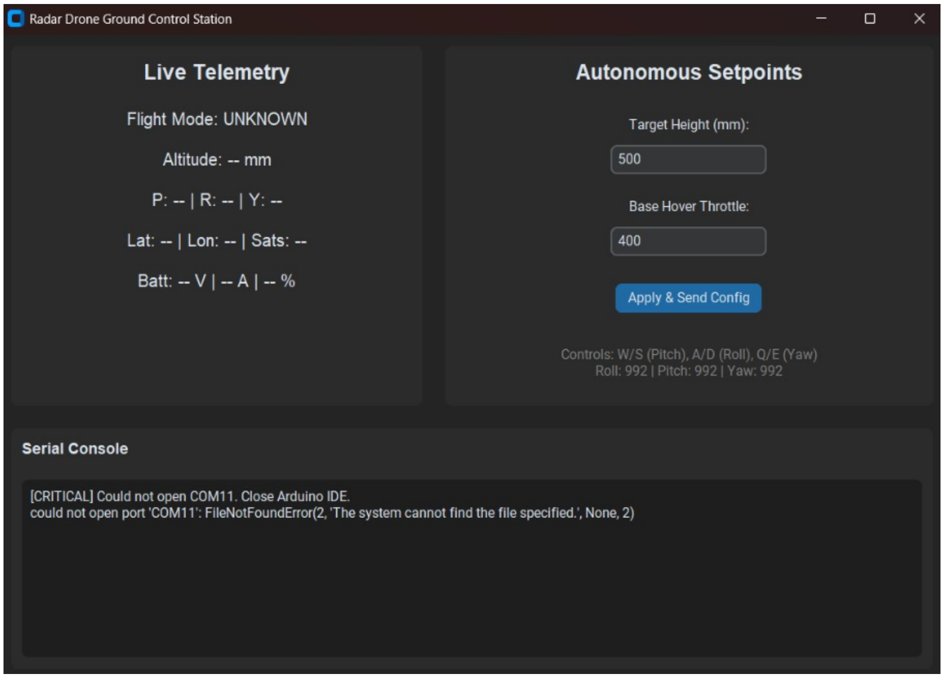


Figure 12: Ground Station Hardware

This hardware consisted of the LoRa module and an Arduino Nano. The Arduino Nano was running code to directly translate the LoRa serial information onto a laptop serial port. Then a Python script was able to interface with the information via the serial com port.

The python script had three primary displays. One was for explicit serial com printing. This helped with debugging and gives the pilot direct access to raw data and data history. The second is an updated display of all current telemetry data. This ensures that the user always gets the most recent data to make a decision with. The last display is used to adjust the autonomous flight. The farthest we were able to get with the autonomous flight display was allowing the user to tune the autonomous hovering by selecting a desired hover distance and a neutral throttle strength.



Flight Mode: UNKNOWN
Altitude: -- mm
P: -- | R: -- | Y: --
Lat: -- | Lon: -- | Sats: --
Batt: -- V | -- A | -- %

Target Height (mm): 500
Base Hover Throttle: 400

Controls: W/S (Pitch), A/D (Roll), Q/E (Yaw)
Roll: 992 | Pitch: 992 | Yaw: 992

```
[CRITICAL] Could not open COM11. Close Arduino IDE.
could not open port 'COM11': FileNotFoundError(2, 'The system cannot find the file specified.', None, 2)
```

Figure 13: Ground Station GUI

3 Requirements & Verification

3.1 Drone

The drone addresses a few key parts of the overall problem statement: low cost, easily deployable, and works in hazardous environments. The drone introduces a solution to each of these by allowing the radar payload to be cheaply, easily, and quickly transported to the hazardous environment without touching anything.

3.1.1 Requirements, Verification Process, & Quantitative Results

The drone subsystem was designed with two primary objectives: minimizing the cost of reconnaissance missions and ensuring the radar could be rapidly deployed in unstable terrain too hazardous for humans. By using a drone, we eliminated all contact with the ground and rough terrain which often increases both the cost and the mechanical failure rate of the mine detection system.

The primary motivation for our design was to create a platform that is significantly more affordable than existing specialized hardware. This was not only to improve the quality of the solution but also to work within the constraints of the class and budget. Aerial systems are inherently more cost-effective than ground-based mine detection systems because they require less heavy housing to navigate the ground. We verified this by comparing the cost of our drone prototype and industry-standard ground-based mine removal systems. Our final drone and radar cost totaled $823.34. Compared to typical specialized mine removal or reconnaissance systems, which can cost tens of thousands of dollars, our platform is substantially cheaper. This makes it more practical for high-risk, "disposable" deployment.

The secondary requirement is to be effective in hazardous zones. Our drone can operate without physically interacting with dangerous terrain and remain airborne long enough to survey a meaningful area. We verified this requirement through flight testing with a simulated payload. The drone achieved stable flight while maintaining a safe altitude, controlled manually by us. Quantitatively, the drone demonstrated a flight time of over 10 minutes while carrying a total takeoff weight of 750g, which accounts for the drone itself and a payload simulating the radar. This flight time is sufficient for short-range reconnaissance flights.

3.2 Radar

The goal of the radar system was to detect metal objects with a diameter greater than 10 cm within a 30° azimuth field of view on both the left and right sides.

3.2.1 Requirements, Verification Process, & Quantitative Results

To verify the performance of the radar system, it was positioned directly above the target and rotated in the azimuthal direction until the target either disappeared from the detec-

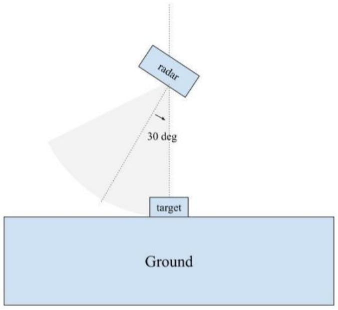


Figure 14: Radar Verification

tion frame or could no longer be distinguished from background clutter. The radar was able to detect objects within an angular range of 37 ± 1° when tested at a height of 0.5 m above the ground, 34 ± 1° at 1.0 m, and 30.5 ± 1° at 1.5 m above the ground.

3.3 Ground Station

The ground station addresses a few parts of the overall problem statement: low cost and easily deployable. The ground station design is a solution to each of these by working with a typical laptop with little extra hardware and provides a telemetry/command interface for the pilot through the software.

3.3.1 Requirements, Verification Process, & Quantitative Results

The ground station serves as the command-and-control hub. It directs communication between the pilot and drone. Its performance is measured by two main objectives: the speed of its data transfer, and the clarity with which it presents information to the user.

The primary requirements was reliable and timely data transfer. This is critical for maintaining situational awareness of a drone mid-flight. We verified this by implementing a communication link that handles all necessary telemetry. This includes GPS coordinates, IMU data, and command inputs via a Long-Range (LoRa) radio transceiver. The LoRa system provides a data rate that, while lower than alternatives, is highly optimized for our use case. Our goal was to transfer the telemetry data and commands with as little loss as possible since the drone needs these commands in flight. In our testing, the data rate was sufficient to update the ground station interface in near real-time. This allows the pilot to constantly operate on current data. This throughput is adequate for our modular reconnaissance purposes where the focus is on coordinate tracking rather than high-bandwidth video or radar data streaming.

The secondary requirements for the design was intuitive presentation of the data to the pilot. We verified the functionality of this system by successfully displaying live GPS coordinates and orientation data on our custom Graphical User Interface (GUI). However, while the system is functional, it currently fails to meet the highest standards of ”intuitive” design. Due to the semester-scale timeline, we were unable to integrate more advanced mapping APIs that would properly visualize the drone orientation, GPS location, and current heading. While all necessary raw information is present for the operator it was never intuitive enough to be worth looking at while flying the drone. While we would consider the data transfer a success for getting all data necessary communicated between pilot and drone, the GUI did not succeed in the same way we desired. This was only a matter of time and could easily be solved by giving more time and effort to the software.

4 Cost & Schedule

Effective project management is critical to the successful development of a semester-scale engineering system, particularly under strict budgetary, manufacturing, and time constraints. Both cost analysis and schedule planning directly influenced the technical direction and achievable scope of this project, requiring continuous evaluation of subsystem complexity, component availability, and integration feasibility throughout the design process. Budget limitations impacted major architectural decisions, including the selection of the drone platform and radar subsystem, while schedule constraints required iterative refinement of project objectives and subsystem responsibilities. The following sections present the project cost breakdown and development schedule used to guide technical decision-making, resource allocation, and overall project execution.

4.1 Cost

A complete bill of materials (BOM) and associated project costs are included to document all hardware, software, fabrication, and subsystem expenses required for implementation of the proposed system.

4.1.1 Labor Cost

Weekly labor cost can be estimated using the following formula:

$$\textbf{Labor Cost} = (\$/\textbf{hour}) \times 2.5 \times \textbf{Hours to Complete}$$

According to the 2023–2024 Illini Success report, the reported median starting salary for electrical engineering graduates is $84,250. Based on a standard 40-hour workweek and 52 working weeks per year (2,080 total working hours annually), this corresponds to an equivalent hourly rate of approximately **$40.50 per hour**.

We also assume that each team member is averaging a total of **10 hours** of work, including research, documentation, and testing, for a length of **10 weeks**.

Table 1: Labor Cost Breakdown by Week and Entire Project

Team Member	Hourly Rate ($/hr)	Hours	Overhead (2.5)	Total Labor Cost ($)
Anna Sako	40.50	10	2.5	1,012.50
Elijah Sutton	40.50	10	2.5	1,012.50
James Tang	40.50	10	2.5	1,012.50
			Total Weekly Labor Cost ($)	3,037.5
			Total Project Labor Cost ($)	30,375

4.1.2 Parts Cost

Table 2: Parts List and Bill of Materials

Item	Description	Manufacturer / Part #	Qty	Unit Cost ($)	Total Cost ($)
1	Bulk Component Order	UIUC RDQ Bulk Order	1	162.86	162.86
2	60-Pin Adapter Connector	Samtec QTH-030-01-L-D-A-K-TR	1	35.59	35.59
3	Heat Gun	Generic Heat Gun	1	12.76	12.76
4	LoRa Module	EBYTE E220-900T22D	1	21.57	21.57
5	LoRa Module	EBYTE E220-900T22D	1	14.02	14.02
6	ESC + Flight Controller	F722 FC + 60A ESC	1	66.23	66.23
7	ESP32 Development Board	ESP32	1	21.65	21.65
8	LiDAR Sensor	Generic LiDAR Module	1	9.82	9.82
9	ELRS Receiver	ELRS 2.4GHz Receiver	1	18.56	18.56
10	USB-UART Adapter	USB to UART Adapter	1	7.64	7.64
11	ST-Link Programmer	ST-Link V2	1	6.54	6.54
12	ST-Link Programmer	ST-Link V2	1	6.54	6.54
13	LoRa Dongle	USB LoRa Modem Dongle	1	28.39	28.39
14	Radar Module	TI mmWave AWR6843ISK	1	412.17	412.17
Grand Total of Parts ($)					**823.34**

4.2 Schedule

The following schedule outlines the major project milestones and subsystem development activities completed throughout the semester. Responsibilities are divided among team members to reflect contributions toward drone development, radar integration, software implementation, PCB design, and ground station development.

Table 3: Project timeline and division of responsibilities

Week(s)	**Milestone**	**Primary Responsibilities**
1–5	Project formation, proposal development, and team contract submission	Team formation, project scope definition, preliminary subsystem research, proposal preparation, and team contract completion/submission.
6	Design document submission	Continued subsystem research, initial hardware/software planning, and completion of the overall system design document.
7–8	PCB revisions, BOM finalization, and subsystem development	Anna – documentation support and ground station development. James – mmWave/radar research, interface development, and software investigation. Elijah – drone hardware development, PCB revisions, firmware development, and component ordering.
9	Spring Break	Limited project activity and schedule reassessment.
10–12	System integration, testing, and progress assessment	Anna – ground station implementation, documentation, and presentation support. James – radar processing, GUI development, and subsystem testing. Elijah – drone software, automation features, and overall subsystem integration.
13	Working prototype completion	Subsystem finalization, prototype integration, debugging, and demonstration preparation.
14–16	Mock demonstrations through final presentation	Final system testing, presentation preparation, technical documentation, and overall system validation for final demonstration.

5 Conclusion

This project demonstrated the design and implementation of a low-cost aerial reconnaissance drone platform with integrated sensing, communication, and ground-station support. Through the final system, the team established a functional proof of concept for collecting flight data, transmitting telemetry, and visualizing drone position data. Overall, the completed work represents a meaningful step toward a scalable reconnaissance platform and provides a strong foundation for future improvements in autonomous flight, radar integration, and expanded sensing functionality.

5.1 Accomplishments

The project achieved controllable manual drone flight while carrying the required onboard electronics and communication hardware, validating the main mechanical, electrical, and power-system design choices. The system collected Global Positioning System (GPS) and gyroscope/inertial measurement unit (IMU) data during operation, allowing position, motion, and orientation information to support post-flight path reconstruction.

The onboard microcontroller unit (MCU) communicated with the flight controller to receive flight-state, GPS, and IMU telemetry. The project also demonstrated ground station communication through the long-range radio (LoRa) framework, allowing flight-related data to be transmitted and displayed in a readable format. Overall, the completed platform provides a modular foundation for future work in autonomous flight, radar feedback, radar data visualization, and expanded reconnaissance functionality.

5.2 Uncertainties

During the final two weeks of the project, we encountered power-related failures with the AWR6843ISK following its integration with the custom 60-pin connector board. After connection to the host system through this interface, the device exhibited overheating and intermittently lost connection to the host PC, including failure to enumerate COM ports, indicating a likely power instability or reset condition rather than a communication-level fault. A continuity test of the 60-pin connector confirmed that no pins were shorted together, suggesting the issue was not caused by direct pin-to-pin shorts in the connector assembly.

5.3 Future Work/Alternatives

The primary limitations of this project were the time and budget constraints of the course. Given additional development time, future work should focus on improving platform stability, expanding sensing capability, and creating a more complete operator interface. One major alternative would be to experiment with a custom radar design rather than relying on the AWR6843ISK module. A custom radar approach could provide greater control over the radar interface, data output, and integration with the onboard microcontroller unit (MCU).

The drone platform could also be upgraded from a 5 inch frame to a 7 inch frame. While the 5 inch platform was inexpensive and modular, a larger frame would provide greater stability, improved payload capacity, and better flight performance with mounted electronics. Using a name-brand flight controller would also improve documentation, software support, and debugging reliability.

Future versions of the system could include advanced autonomous flight, radar feedback to the ground station, radar data visualization, and an app-based ground station user interface. Adding onboard cameras would also improve usability and debugging by giving the operator visual feedback during flight. Together, these improvements would make the platform more reliable, expandable, and practical for continued aerial reconnaissance development.

5.4 Ethical Considerations

This project was developed as a proof-of-concept aerial reconnaissance platform, so the ethical focus shifted from high-stakes mine detection to responsible data collection, operator safety, and future system use. In accordance with the IEEE Code of Ethics [5], the design prioritizes reducing risk to human operators by enabling remote collection of flight and environmental data while maintaining direct operator control during testing. The system does not perform safety-critical autonomous detection decisions in its final implementation, reducing the likelihood of false positives or false negatives affecting civilian safety.

Relevant risks remain for future versions of the platform. Wireless telemetry and command links should be secured to prevent unauthorized access, command spoofing, or data manipulation. Future radar, camera, or autonomous navigation capabilities should also incorporate safeguards related to privacy, reliability, and safe operation. Because reconnaissance technologies may be applied in sensitive or harmful contexts, future development should emphasize humanitarian and safety-focused applications, comply with applicable flight and communication regulations, and undergo appropriate validation and testing prior to deployment.

References

[1] eCalc. “xcopterCalc - Multicopter Design and Performance Calculator,” Accessed: May 6, 2026. [Online]. Available: https://www.ecalc.ch/xcoptercalc.php.

[2] C. S. Analysis. “”Wave Propagation in Lossy Dielectrics”,” Accessed: May 6, 2026. [Online]. Available: https://resources.system-analysis.cadence.com/blog/msa2021-wave-propagation-in-lossy-dielectrics.

[3] Texas Instruments, *TPS54331 3-A, 28-V Input, Step Down DC-DC Converter With Eco-mode Datasheet*, Rev. F, Texas Instruments, Dallas, TX, USA, 2014. Accessed: May 6, 2026. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps54331.pdf.

[4] STMicroelectronics. “STM32H753 High-Performance Microcontroller Datasheet,” Accessed: May 6, 2026. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32h753vi.pdf.

[5] IEEE. “”IEEE Code of Ethics”,” Accessed: May 6, 2026. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.