Building Interior

Reconnaissance Drone

ECE 445 Design

Document - Spring 2026

Project # 25

Jack Lavin, Jacob Witek, Mark Viz

Professor: Joohyung Kim

TA: Shiyuan Duan

Contents

1 Introduction

Our goal is to design a system that allows a user to view accurate, real-time information about the presence of people in rooms that the user is physically near but not inside. The user should be able to "see people through walls" and be alerted to the exact location of where an individual may be, as well as some information about their speed. Through the use of a drone housing a curated selection of sensors and an augmented-reality software program we will design a functioning prototype of this system.

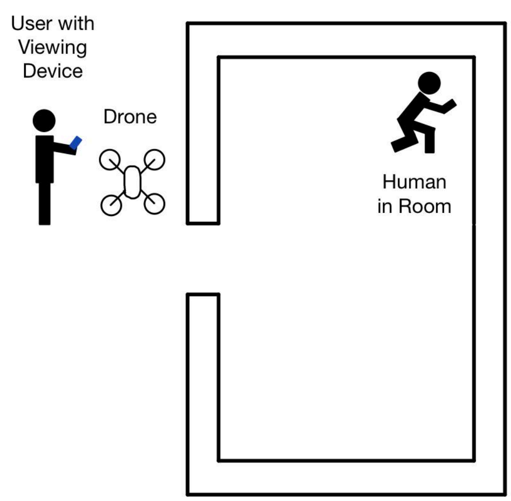
1.1 Problem

There are many situations when law enforcement or emergency medical service professionals need to quickly know real-time, useful information about a non-visible location without sending a human to gather this information due to potential or confirmed risks. One of the most important things to know in these situations is if there are people in a room or area, and if so, where they are located and if they are moving. While many of the present solutions in practice by professionals offer a promising effectiveness, they can rarely be operated by one person and take away valuable time and manpower from situations which almost always require great amounts of both. Our solution attempts to address these existing issues, provide a clean and easy-to-use setup and interface, and most importantly, effectively improve the response capabilities of law enforcement or emergency medical professionals.

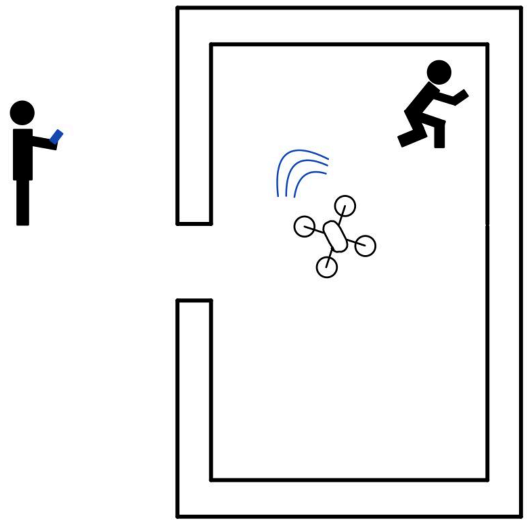
1.2 Solution

Our solution to this issue is to use a reconnaissance drone, acronymed by B.I.R.D, primarily equipped with a mmWave Human detections sensor, an 3x3 ultrasonic receiver/transmitter array module, and an ESP32 microcontroller. This drone will enter a hazardous room and perform a simple 360 maneuver while hovering. If the sensors detect the presence of a human, the drone will send a notice along with relevant position information directly to the AR software over 2.4G Wi-Fi using the ESP microcontroller. This received information will then be used by a user's device running our custom written augmented-reality viewing program. This program will calculate the distance and angle at which the person is to the user's device. Then the program will display an icon and a distance value overlaid on the device's front camera feed to provide the user with instant and accurate position information of humans present in the adjacent room. The user can then take the appropriate actions or assist other responders present to ensure that they along with the detected human presence are able to continue navigating the situation as safely as possible.

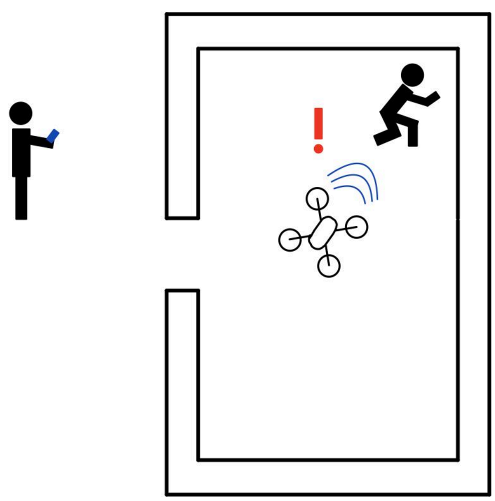
1.3 Visual Aid



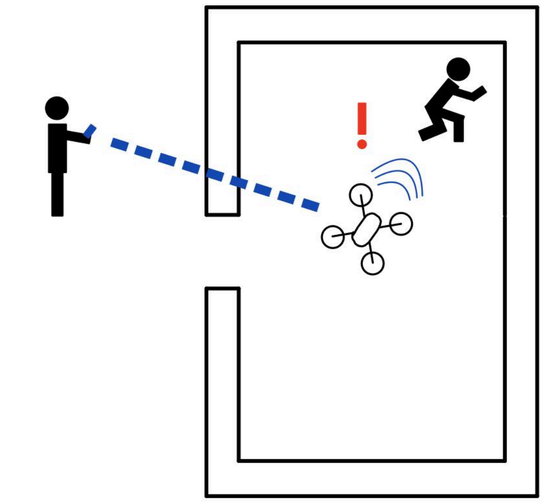
(Figure 1 Hypothetical Use Case)



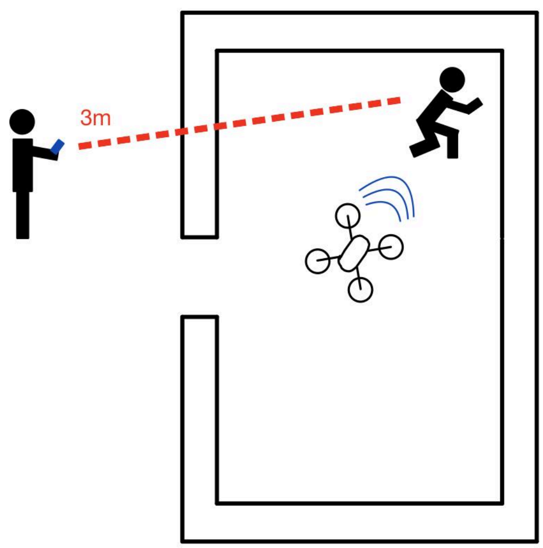
(Figure 2 Drone Sent into Room)



(Figure 3 Drone Detects Human)



(Figure 4 Drone Alerts Exterior Device)



(Figure 5 Outside Device Alerts User)

(Figure 6 Hypothetical View From AR app)

1.4 High Level Requirements

The BIRD will have 3 main criteria for success:

- A semi-autonomous vehicle that can provide our sensors with a vantage point with five modes of movement: lifting, hovering, landing, 360 degree rotation, and forward movement.
- Onboard sensors that are able to detect human presence and map the local environment of detected persons.
- Transmit, calculate, and display the relevant human presence data onto an alternate reality camera feed as demonstrated by *figure 6* above.

2 Design

This section aims to go in depth on both the hardware and software designs we have created to allow us to create a functional final product.

2.1 Physical Design

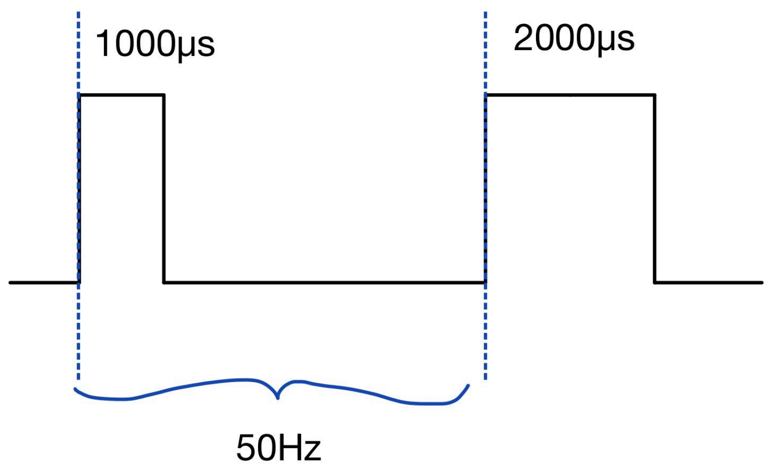
After careful consideration of different vehicle options, a lightweight drone was chosen as the most versatile platform for our desired sensing capabilities. Given the time and resources involved with designing and fabricating a frame through the help of the machine shop, we also chose to base the structure of the drone on a commercial carbon-fiber frame for the most optimal structural properties. On the frame, we will fasten our brushless DC motors and speed controllers using machines screws, adhesives, and ties. The LiPo battery will also be secured to the underside of the frame using its included strap. Our PCBs are designed to align with the existing mounting holes on the frame for easy installation on extended standoffs. The main PCB, which handles flight and sensing controls, will be mounted to the top panel, and the secondary PCB, which routes high-current battery power to the ESCs, will be mounted internally to the bottom panel. Extra sensors that are not mounted directly to the PCB and instead use pin headers will be mounted to the proper area of the frame securely using custom 3D printed adapters. All of the components are chosen and/or designed to interface properly together as one powered, functioning unit.

2.1.1 Drone Frame

In choosing a frame for the drone, we had to ensure that it would be large enough to house all of our electronics, strong enough to withstand early testing or unplanned collisions, but also still small and light enough to achieve stable indoor flight and maneuverability. Since FPV drone racing has risen in popularity, there are a number of robust carbon fiber frames available from online retailers. Ultimately, our chosen frame featured two body plates at 2 mm thick, four motor arms at 5 mm thick, and various standoffs and machine screws for assembly. The total diagonal length between the center shaft of two opposite motors is 225 mm in total. The body and arm pieces appear cut or milled from single, woven and laminated sheets with various mounting holes and hollow cuts for weight reduction. After assembling and assessing the frame, we determined it to have no loose tolerances, minimal flex, and easy adaptability for extra custom hardware.

2.1.2 Drone Motors, Battery, and ESCs

Given that the drone frame we purchased is designed for 5’’ propeller blades we had to base the motor, battery and ESCs on a 225mm wheelbase system. We will be using four 2306 motors. The motors’ stator is 23mm wide and 6mm tall, as indicated by their name. Each motor completes 2400 rotations per minute for each volt applied to the motor up to 22 volts. This motor provides smooth throttling from low to high and precise control at the expense of being slightly less powerful. Each motor can draw up to 40A of current at max speeds. Each motor will run off of a 40A ESC. These ESC’s convert pulse width modulation signals (PWM) that are easily sent by the ESP32 into signals the motors can use. Each ESC will run at 50hz with 1,000 **µ**s being the minimum pulse and 2,000 **µ**s being the max. Below is an example PWM signal generation that shows the drone going from 0% thrust (1,000 **µ**s) to 100% thrust (2,000 **µ**s) across one duty cycle:

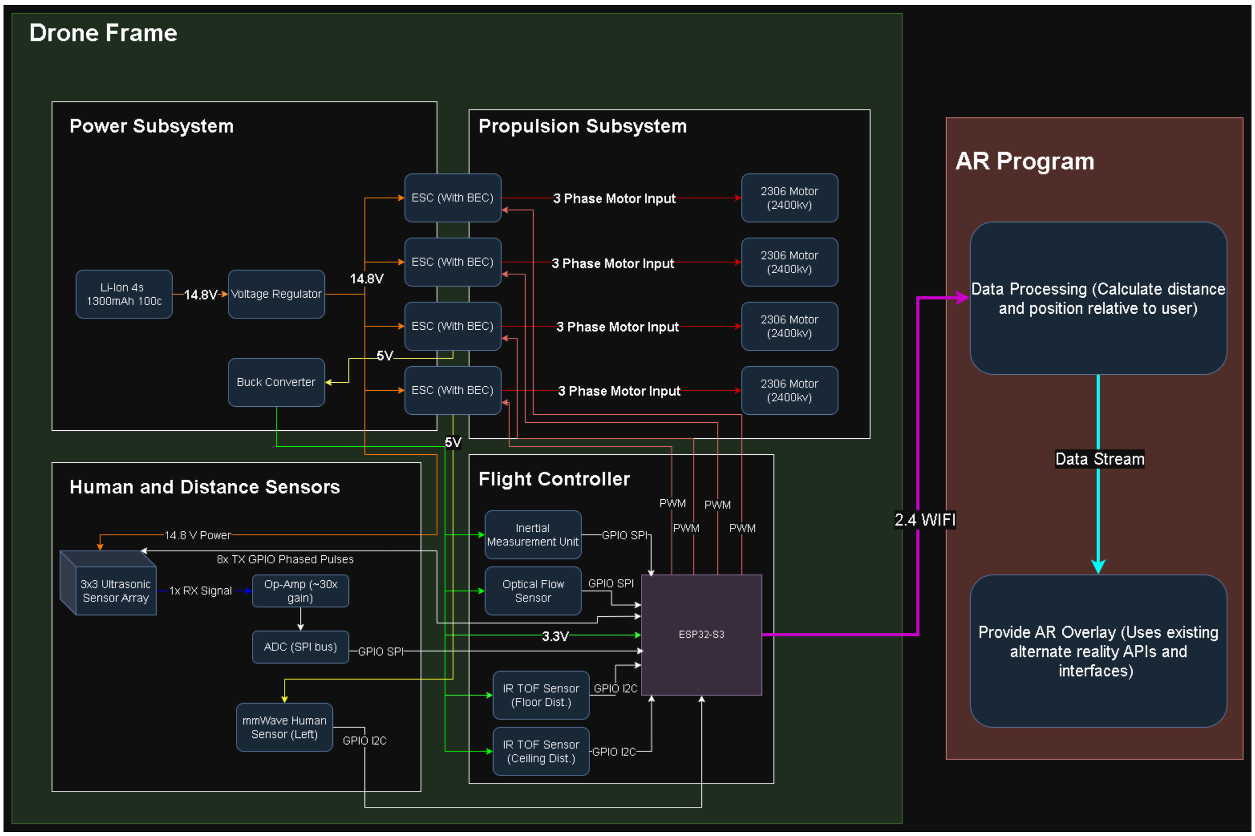


(PWM at 0% Motor Speed and 100% Motor Speed)

The battery we have chosen for our system is a 4s (14.8V) 1300mah Li-Po with 100c. This battery will provide enough power to the motors to allow for extended flights, while also being light enough to not drag the drone down. Something of note is that the battery can safely discharge at 130A. This means that if all motors are max (40A per motor), this will overdraw current from the battery. However, we have done the calculations as shown by the tolerance analysis equations below, and verified that each motor will nominally be in the range 25%~35% which will keep us well within the range of the batteries current draw. We will include fuses to ensure that the current is never too high for the battery.

Each ESC features a voltage step down converter in the form of a battery eliminator circuit (BEC) with an output of 6V and maximum current draw of 4A. Using one of the ESC in combination with a buck converter to step down to 3.3V, it will be used to power the ESP32 microcontroller as well as the IMU, optical flow sensor, and Time of Flight sensors. A second ESC will output 6V and in combination with a 5V output buck converter, will be used as the power supply for the mmWave human presence sensor.

2.2 Block Diagram



2.3 Functional Overview of Block Diagram

2.3.1 Power Subsystem

The power subsystem is responsible for distributing power to all other subsystems found in the block diagram. The primary current draw from the battery will be from the four electronic speed controllers from the propulsion system, each being rated for a maximum of 40 Amps at peak sustainable thrust. After initial tests with all four motors and battery installed, we determined that the drone can hover at only ~25% total thrust, giving us decent longevity out of our 1300 mAh battery. In addition to powering the motors, it is also responsible for providing stable 3.3 V and 5 V power to various control and sensing IC's on the main PCB. To do so, we utilize two of the 6 V output pins provided by each ESC along with buck converters to achieve proper voltage levels on the board for these components. To ensure the stability of these sources under the peak current draw of our components, we also included an inductor/capacitor network on each buck converter in the recommended manufacturer configurations. The last component that requires power from the LiPo battery are the eight transmitting ultrasonic sensors, each of which directly uses the 14.8 V battery voltage. These components can be driven at up to 30 V, and do not draw much current during pulses, so they will be powered via a simple jumper wire pair from the power PCB to the main PCB, which also serves to establish a common ground.

Requirements	Verification
Total output current from LiPo battery is less than a total of 130A and less than 32.5A per electronic speed controller.	Generate our maximum programmed thrust PWM signal of 40% (1400**µ**s per duty cycle) and observe if 30A fuse connecting LiPo battery to ESC is blown. Generate a 60% (1400**µ**s per duty cycle) PWM signal and observe if 30A fuse connecting LiPo battery to ESC is blown.
Provide stable power to each electronic speed controller with up to 32.5 A of current	Run motors through full thrust range using a test program and monitor battery and ESC currents using bench instruments
Provide stable 3.3 V power to integrated circuit components via buck converter	Power each 3.3 V component through the battery-buck circuit, monitor total current draw and stability while active
Provide stable 5 V power to mmWave sensor via buck converter	Power the 5 V sensor through the battery-buck circuit, monitor total current draw and stability while active

Provide 14.8 V for transducer amplification supplying ~100 mA	Perform a beam sweep across the full range of motion, amplifying the phased pulse signals through a transistor on battery power
Provide stable power at varying voltages to **all** circuit components including those listed above	Secure full drone assembly to bench for thrust testing while sensing, ensure all components active, monitor battery supply using bench instruments

2.3.2 Propulsions Subsystem

The propulsion subsystem is responsible for ensuring ascent, stable hovering, and simple maneuvers all under the final operating weight. As mentioned above, the motors are driven with a three phase output signal from the speed controllers, which in turn are powered by the battery and controlled by PWM signals from the flight controller. This PWM control, also described above, allows for full throttling of each motor, and when paired with IMU, optical flow, and IR range sensor data will enable basic autonomous flight with proper programming.

Requirements	**Verifications**
When the user sends an activation signal from their device, the drone turns on at 0% thrust.	A confirmation message is sent from the ESP32 to the user indicating successful activation.
The drone lifts and hovers in place at a set height 5 seconds after user activation.	-Set thrust to 35% to lift -Set thrust to 25% when bottom ToF sensor measures the set height setting or if top ToF sensor measures less than 0.5m to ceiling. -Ensure there is minimal drift through the use of an Inertial Measurement Unit (IMU) sensor
The drone flies forward a set distance and hovers.	-Set front thrust to 25% and rear thrust to 30% thrust, and return to 25% for all after the optical flow sensor measures the set distance

	measurement from initial liftoff point.
The drone performs a 360 degree rotation	-Set the two clockwise motors to 25% thrust and the two counterclockwise motors to 30% thrust to perform a yaw rotation. -Ensure that there is no vertical movement with ceiling and floor ToF sensors. -Ensure circular spin without drift with the optical flow sensor.
The drone lands in place and sets to 0% thrust.	-Set thrust to ~20% thrust to slowly lower the drone. -Keep drone stable on the x-y plane with information from IMU's gyroscope readings -Set thrust to 0% when ToF sensor measures a stable 2cm

2.3.3 Sensor Subsystem

The human sensing subsystem is responsible for acquiring accurate positional and situational information about humans within the sensors' overall range, which is primarily achieved using the mmWave GHz sensor module and ultrasonic sensor array. The mmWave module utilizes 24 GHz RF technology with a simple two-antenna approach to obtain reflected human position data. This data is then communicated to our microcontroller over the I2C bus and used for our initial human position determination. The ultrasonic sensor array utilizes nine total 40 kHz transducers, configured as eight transmitters and one receiver. The eight transmitters are pulsed using a consistent phase offset that can be varied, allowing for rudimentary beam forming. The 40 kHz resonant frequency of the transducers allows for easy deterministic control of each transmit signal given our microcontroller's maximum clock speed of 240 MHz. Since the GPIO pins only switch voltage at the level of the ESP32 chip, the GPIO outputs are instead used to control the gate of an amplifying nMOS transistor driving each transmitter from the battery voltage. To amplify the weak received pulses of each directed beam pulse, the receiving transducer's signal is first amplified using a lower noise JFET op-amp with a tunable gain, and then sampled by the microcontroller through an analog-to-digital converter on the SPI bus. This full sensor array can be used to conduct rough sweeps of an area suspected of having human presence to build a simple depth map around it and differentiate its position from other physical landmarks. Ultimately, all important pre-processed data is sent over Wi-Fi from the microcontroller to the Android software for further processing and use in the AR app.

Requirements	Verification
When a human body is within the determined/effective range of the sensor instruments, other system components are notified.	The ESP32 sends a confirmation message over Wi-Fi that a human being has been detected within the sensor range.
When a human body is detected and the system is notified, all relevant measurements regarding the distance from the drone to the human are reported.	The ESP32 sends a distance value from the mmWave and the basic depth value data from the ultrasonic scan along with the Wi-Fi confirmation message. The distance value and depth data are used to determine the most accurate true distance from drone to human.

2.3.4 Flight Controller

The flight controller, an ESP32-S3 module, is primarily responsible for driving the motor PWM signals according to the received sensor data. It uses data from the aforementioned telemetry and attitude sensors collected along the SPI and I2C busses; managing them with various GPIO pins for chip select signals. The IMU will provide 3D acceleration data, as well as magnetometer data that can be used for azimuth calculation. The optical flow sensor will provide 2D positional drift data through the use of LiDAR sensing to allow for translational corrections for stable hovering. The two time-of-flight IR sensors utilize similar technology to the optical flow sensor, but simply communicate a distance to a reflected object up to 4 meters away. These will be used to obtain accurate distances between the drone and the ceiling or floor, which will be used in the overall informational pipeline as well as for object avoidance and sensor positioning.

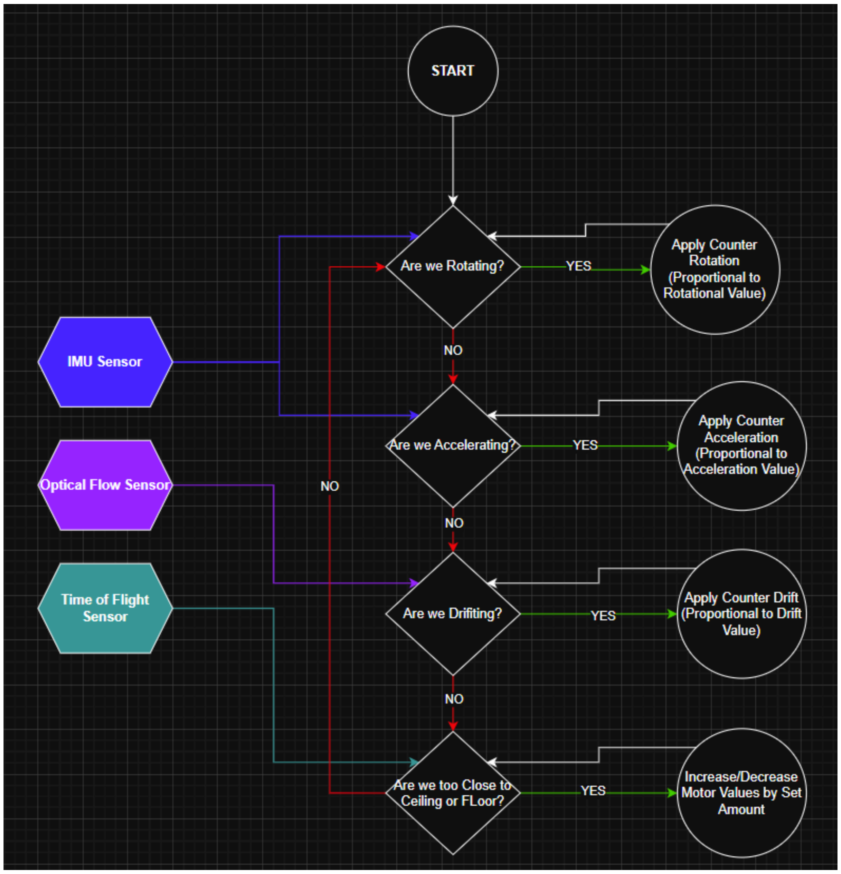
Requirements	Verification
During the appropriate portions of flight paths, transmit all sensor data as specified by the Sensor Subsystem Requirements table continuously.	Monitor data sent as text or otherwise through Wi-Fi from the ESP32 during the sensing portion of flight paths. This data is collected and referenced against the real physical measurements for accuracy.
Properly manage all sensor module chips on SPI and I2C protocols using context for priority.	During the initial portions of flight paths, the ESP32 will only be using and sending data over Wi-Fi from the IMU, optical flow, and IR range sensors. While it is inside the room to be investigated, it will enable sensor modules and begin transmitting their data over Wi-Fi, adapting its flight to the new bandwidth limitation on the protocol buses which the flight sensors use.
Drive the ultrasonic array with phased 40 kHz PWM pulses using a beam forming algorithm to send directed pulses for depth measurement at the receiver using the reflection time-of-flight.	Whenever the ultrasonic sensor is to be used, the ESP32 will drive 8 separate 40 kHz GPIO outputs with controlled phased differences, and immediately following the end of one pulse, the ESP32 will record receiver data from the ADC on the SPI bus and send the reflection profile over Wi-Fi to be processed further. If desired, a basic TOF measurement can be determined on the ESP32 before sending over data.

2.5 Software Design

2.5.1 Drone Flight Control & Sensors

The software on the drone will be run on the ESP32-S3 module. This program has two main jobs: ensure stable flight and process incoming and outgoing data. To allow our drone to hover autonomously we need a program that is able to detect movements and acceleration and apply proper counter forces. To start off, the program needs to be able to control the motors. The ESC signal pins will be connected to GPIOs on the ESP32-S3. The ESP is then able to control the motors based on pulse width modulation (PWM) signals it generates. It is able to do this with the help of the ESP32Servo header file. Upon startup the drone sets the PWM ranges and frequencies to be outputted using the *setPeriodHertz* and *attach* functions. In our case our ESCs will be calibrated to 1,000 **µ**s to 2,000 **µ**s at 50 Hz. We then specify the GPIOs that are being used for the PWM generation (4, 5, 6, 7). After the program has successfully configured the ESC we are able to begin generating signals. The ESCs need to be set to detect the lowest PWM input before allowing themselves to be armed and so the program will set the PWM to 1,000 **µ**s using the *writeMicroseconds* function. Now that the setup is complete the ESP is ready to control the motors.

Once the drone has motor control we need to make use of sensor data to ensure we are stable in flight. The sensors the drone will be using for this purpose include the LSM6DSOXTR IMU sensor, the two IOR time of flight sensors, and the optical flow sensor. The stability program combines the data from these 3 sensors to crosscheck its movement and position. The IMU sensor provides both acceleration and rotation data. If the drone notices acceleration or rotation in a certain direction it will use the PWM controls stated above to increase the corresponding motors proportionally to the acceleration value. This section of the program will require fine tuning and value adjustments which will be achieved through safe testing. We will use the Adafruit_LSM6DSOX header file to collect data from this module. The data from the optical flow sensor will work to detect slight drifts that the IMU may not be able to pick up on. Again the program will adjust motor speeds corresponding to the magnitude of the drift. We will use the Bitcraze_PMW3901 header file and its associated functions to use this module. The time of flight sensor modules serve to correct vertical adjustment. One of these sensors will be facing up and the other will be facing down, so we are able to detect the floor and ceiling within a 4m range. This part of the program will work by making sure the drone stays within a certain threshold between the floor and ceiling. If the drone gets to close it will decrease or increase all motors depending on the desired altitude. We will make use of the provided VL53L1X header file to make use of this module.



(State Diagram of Flight Control Software)

Once we have achieved stable flight we can now focus on the scanning portion of the software. This will require data to be collected and then transmitted across the server described below. The two sensors of note here are the mmWave human sensor and the phase array system. Both the human presence and distance data will be broadcasted to the client upon detection.

Finally once all data is collected we need to transmit the data over to the AR program. To do this we are making use of the arduino WiFi system. This allows the ESP32 to host its own server that an outside source can wirelessly connect to. The server will be hosted on port 1000. Once a secure connection is established with a client the drone will stream the mmWave human presence sensor and the transceiver/receiver beam data.

Something of note is that each of these sensors makes use of SPI protocol except for the mmWave human sensor. Each of these SPI sensors will be sharing a bus (GPIO 18, 19, 20) and each have their own GPIO for chip select. This will be done using the arduino provided SPI functions to easily transmit and receive data from the sensors. The mmWave human sensor will be communicating through I2C protocol which can be accessed through the provided Wire library.

2.5.2 User AR Interface

The user interface AR part of the software will be running on the android device. We will make use of android studio and the provided APIs to create an app that can communicate with the data being transmitted by the drone. This app will have 3 main functionalities: receive data from the drone, process data, display data in an AR format. To receive the data we will be setting up the app as a “client” for the drone server. When the app first opens it will attempt to make a socket connection with the drone. If successful we will move onto data processing. The drone will be transmitting data regarding its current position and heading relative to its start position (using dead reckoning) and where a human presence is detected. Once a human is detected the app will be able to triangulate the distance and angle of the human from the start position which is where the AR user will be standing. This can be achieved through the law of cosines:

$$a^2 = b^2 + c^2 - 2bc * cos(A)$$

Once we have the positional data we can finally display the data to the user’s screen. We will make use of Google’s ARCore library which provides all the functions required to access the phones positional and motion sensors to display our drone data on the device screen.

2.7 Tolerance Analysis

<u>Total Mass Calculation (estimate):</u>

- Carbon fiber drone frame: ~125 g
- LiPo Battery: 156 g
- Brushless DC motors: 30.2 g x 4 = 120.8 g
- Electronic speed controllers: 44.2 g x 4 = 176.8 g
- Sensor array: 4.8 g x 9 + ~10 g = 53.2 g
- PCB: ~60 g
- Accounting for propellers, wiring, other incidentals: ~100 g
- **Total: ~800 g**
- **Absolute max: 1100 g**

<u>Thrust Calculation:</u>

Each motor has a value of 2400kv. This means that each motor can perform 2400 rotations per minute per volt. We will be running each motor at 4s (~14.8V).

- RPM Per Motor: 2400RPM/V * 14.8V = 35,520RPM
- Propeller Pitch: 4.5in or 0.114m
- Diameter: 5in or 0.127m

Drone Thrust Equation [5]:

$$Thrust = 4.392 * 10^{-8} * RPM * d^{3.5}/\sqrt{pitch} * (4.233 * 10^{-4} * RPM * pitch)$$

$$Thrust = 4.392 * 10^{-8} * 35520 * 5^{3.5}/\sqrt{4.5} * (4.233 * 10^{-4} * 35520 * 4.5)$$

Total Maximum Thrust Per Motor = 13.9N

<u>Operating Current Calculation:</u>

To hover, the force generated by thrust must be equal to the gravitational force (mg):

$$4 * F_{thrust} = mg$$

$$F_{thrust,\,hover} = \frac{1}{4}(1.1kg)(9.81m/s^2) = 2.698N$$

Each motor must generate 2.698N of thrust in order to hover. As a ratio to maximum thrust:

$$\frac{F_{hover}}{F_{max}} = \frac{2.698N}{13.9N} = 0.194$$

From experimental testing, we have determined that running the motors at ~25% total thrust allows for the drone to hover, which closely matches the theoretical expectation of 19.4% of maximum thrust needed to hover and thus verifies that the theoretical equation is accurate for the BIRD. Applying the experimental hovering thrust:

$$13.9N * 0.25 = 4.392 * 10^{-8} * RPM * 5^{3.5}/\sqrt{4.5} * (4.233 * 10^{-4} * RPM * 4.5)$$

With the above equation, RPM ~ 17760 per motor in order to hover. From the motor manufacturer's specifications [6], operating the motors at 17800 RPM requires 8.4A at 16.8V (8.4*16.8 = 141.12W), and in our case of 14.8V, I = 141.12W/14.8V = 9.535A.

From experimental testing, the drone safely rises at 35% total thrust, however we will use 40% thrust to account for future implications, where we may want to make the drone rise quicker than in our first iteration. Applying the drone thrust equation to find RPM:

$$13.9N * 0.40 = 4.392 * 10^{-8} * RPM * 5^{3.5}/\sqrt{4.5} * (4.233 * 10^{-4} * RPM * 4.5)$$

The equation results in RPM ~ 22,458. The manufacturer specifications [6] indicate that at RPM = 21,200, current is 13.6A at an operating voltage of 16.8V, resulting in 228.48W of power. Applying our operating voltage of 14.8V, I = 228.48W/14.8V = 15.44A. While our maximum RPMs are 1258 more than the specifications, this will not increase the current significantly, as a change of 3400 more RPM increases the current by 15.44-9.535 = 5.9A when comparing the two closest specifications for the BIRD's hovering versus lifting. Thus, as the RPM difference from experimental to specification is 1/3 the change from the two manufacturer specifications, we may say that each motor pulls approximately 15.44+(0.33*5.9) = 17.4A for lifting.

Operating the BIRD at **25% thrust for hovering requires 9.5A** and operating at **40% thrust for lifting requires ~17.4A** - both of which are well within our safety limit of 32.5A per motor in order to not exceed the LiPo battery maximum ratings for current output.

3 Cost and Schedule

3.1 Cost Analysis

By current estimations, each group member is contributing, on average, 10 hours of work on the project every week. Assuming a new graduate electrical/computer engineering salary of $35/hr, with a company overhead of x2.5, we have a labor cost per member of:

$35/hr * 2.5 * 10hr/week * 11 weeks = $9625

With three group members, our total labor cost is 9,625*3 = $28,875 and a total parts cost of $335.05. This brings the total cost of this project to be $29,210.05.

Description	**Manufacturer**	**Quantity**	**Extended Price**	**Link**
ESP32-S3-WROOM1 Devboard	YEJMKJ	2	$20.89	Amazon
ESP32-S3-WROOM1	Espressif Systems	1	$6.13	Digikey
FPV Brushless Motors 4-pack	Emax	1	$65.99	Amazon

2400kV 3-4S				
40A Electronic Speed Controller (ESC)	Toytensi	4	$83.96	Amazon
Connector Plugs 20-pack	Dgzzi	1	$7.59	Amazon
Carbon Fiber Quadcopter Frame	Mark4	1	$33.99	Amazon
4S LiPo Battery (100C @ 1300mAh 14.8V)	Ovonic	1	$18.99	Amazon
VL53L1X IR Time of Flight Sensor	Dweii	2	$29.98	Amazon
PMW3901 Optical Flow Sensor 2-pack	PixArt	1	$30.99	Amazon
LSM6DSOX Accelerometer & Gyroscope IMU	Adafruit	1	$11.95	Adafruit
C4001 mmWave Presence Sensor	DFRobot	1	$13.90	DFRobot
XL1509-3.3 Buck Converter	Evvo	1	$0.35	Digikey
XL1509-5.0 Buck Converter	Evvo	1	$0.35	Digikey
TCT40 R/T Transducer	HiLetgo	20	$9.99	Amazon
STP22NF03L	STMicroelectronics	8	*student self-service	N/A
LF357 JFET Op-Amp	Texas Instruments	1	*student self-service	N/A
Galaxy Smartphone (Android)	Samsung	1	*borrowed	N/A
Miscellaneous PCB Components (capacitors, resistors, inductors, terminal blocks)	N/A	N/A	*student self-service	N/A

3.2 Schedule

Week	**Task**	**Person**
Week 1 **2/1 - 2/7**	Research wireless presence systems (RF/sound)	Jacob Witek
		Mark Viz
	Research power system and propulsion system components	Jack Lavin
Week 2 **2/8 - 2/14**	Finalize research on feasibility of various Human Presence sensors, choose one as final	Jacob Witek
	Research ultrasonic array theory & implementation	Mark Viz
	Finalize power system, propulsion, and drone stability components research	Jack Lavin
Week 3 **2/15 - 2/21**	-Begin initial KiCAD schematic design -Purchase components for initial drone construction	Everyone
Week 4 **2/22 - 2/28**	-Attach motors to drone frame -Solder motors to ESC with connector plugs and connect ESC to devboard -Test motors in lab	Everyone
	Code and test program to use C4001 sensor	Jacob Witek
	Create initial PCB layout for first round submission	Mark Viz
	Code WiFi connection program to ESP32	Jack Lavin
Week 5 **3/1 - 3/7**	-Test what percent of power allows for a drone to hover with full load. -Initial stability test integrating IMU sensor	Everyone
	-Test C4001 sensor's detection plot, and begin to create a table of vector x^ and y^ measurements from scalar distance data. -Support Jack in creating programs for IMU and optical flow sensors	Jacob Witek
	Continue ultrasonic array prototyping & create Time of Flight sensor program	Mark Viz
	Code program to use IMU sensor and Optical Flow sensor	Jack Lavin
Week 6	- Create breadboard that integrates C4001, IMU, Time of	Everyone

3/8 - 3/14	Flight, and optical flow sensors with ESP32 devboard. -Ensure basic stability for hovering with the drone -Solder initial PCB and perform various testing	
Week 7 **Spring Break 3/15 - 3/21**	- Continue PCB testing - Revisions to PCB design - Submit 2nd PCB for 4th round orders	Everyone
Week 8 3/22 - 3/28	Create a program to process ultrasonic sensor data to create 2D mapping. Test various physical geometries to determine best fit for beamforming.	Jacob Witek Mark Viz
	Create an interface for esp32 data to an Android application	Jack Lavin
Week 9 3/29 - 4/4	-Test and solder 2nd edition PCB -Create and test drone forward and 360 degree rotation movement	Everyone
Week 10 4/5 - 4/11	-Continue drone movement and rotation testing -Write program to process data for distance and position relative to user with android application and create augmented reality display (AR overlay)	Everyone
Week 11 4/12 - 4/18	Finalize the project, ensuring all subsystem requirements are met.	Everyone

3.3 Risk Analysis

The main risk in operating and designing this project is the physical use of an aeronautical drone. While in the initial design stage, we realized that the motors would be operating at levels that are dangerous to touch by hand (40 A and 14.8V at max), and as such we ensured to use a quality ESC that has been tested to operate safely within our maximum levels, rather than designing the component ourselves. Furthermore, our selected LiPo battery is only able to safely discharge at 130A, while the maximum pull would be 40*4 = 160 A for the motors operating at maximum. For the safety of the drone and people around it, we have the following four safety features.

The drone will initiate at 0% thrust and safely turn off at 0% thrust, which occurs when the PWM signal has a pulse of 1000 **µ**s for each period. Secondly, the drone will never operate above 40% maximum thrust, which as indicated in the Tolerance Analysis section, ensures that the output amperage of the battery is never exceeded. Thirdly, there will be a fuse on each ESC line to prevent any unexpected current spikes. Lastly, the ESP32-S3 microcontroller is fully autonomously controlled in regards to flight - there will never be a case where a lost signal connection between user and drone causes concerns with drone flight.

4 Ethics and Safety

For the past two decades, society has been shown a wide-scale adaptation of the use of drones in both military and civilian circumstances. As such, it is prudent to specify that the Building Interior Reconnaissance Drone (BIRD) is intended solely for use in search and rescue and/or police-sanctioned operations. Thus, any use of this drone and related software for unlawful entrance or harassment is strictly forbidden in accordance with IEEE Code of Ethics, Sec. I.1 [1]. While we do not currently intend to commercialize our project; if our stance was to change - we would insure to only sell the BIRD to law enforcement professionals or fire and rescue departments in order to ensure the responsibility of potential users of our project and to eliminate the use of the BIRD in nonprofessional settings in accordance with IEEE Code of Ethics. Sec. I.3 [2] and Sec I.4 [3].

The BIRD contains no devices that are unstable or intentionally harmful towards its users and recipients. Our project utilizes human presence radio sensors that operate at 24 GHz.

According to the FCC [4], the 24.05 GHz - 24.25 GHz frequency band is designated for industrial, scientific, and medical equipment and allows for the use of unlicensed devices within this range, therefore we may operate our human presence sensors freely without license. The ultrasonic sensors will be used in the 40 kHz sound frequency, and as such should not affect human hearing in any way, despite potential audible emissions due to the operational design of the array. The major safety concern of our project is the use of a 4S LiPo battery with a discharge rate of 100C and capacity of 1300mAh, operating at 14.8V. For the protection of the battery, motors, and our PCB - we use four ESCs (electronic speed controller) rated at 40A for 2-4S, which also each contain a BEC (battery eliminator circuit) with output of 5V. Thus, each ESC includes power-on protection, over-temperature protection, low-voltage cutoff, and signal-loss protection which aim to protect the ESCs, motors, and circuit components while also acting as voltage regulators to step-down the LiPo voltage to 5V. This 5V output is passed through a buck-converter in order to step down to the required 3.3V needed to supply the ESP32-S3 microcontroller chip.

The BIRD will protect the safety of officers and other government personnel from blindly clearing rooms that may otherwise contain a threat, either in the form of a person or a structural instability. The important distinction between the BIRD and currently available commercial detection drone systems is that the BIRD is safely usable by one individual. In other words, current drone systems require the user to control the drone through a physical controller - therefore preoccupying them - and thus requiring two people to ensure safety of the drone user. On the contrary, BIRD requires the user to send in the drone initially while having full control of the situation afterwards, be it to avoid falling objects from a collapsing structure or an assailant exiting the selected room during drone operation. In a search and rescue scenario, this allows for a theoretical increase of x2 speed at clearing a residential building - as we may now have twice the number of room clearing operations occurring with the same manpower as before.

References

[1] IEEE, *IEEE Code of Ethics*, sec. I.1. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[2] IEEE, *IEEE Code of Ethics*, sec. I.3. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[3] IEEE, *IEEE Code of Ethics*, sec. I.4. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[4] “Federal Communications Commission,” Dec. 2001. Accessed: Feb. 14, 2026. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-01-357A1.pdf

[5] A. Staff, “How to Calculate & Measure Propeller Thrust,” *Tyto Robotics*, Jan. 04, 2026. https://www.tytorobotics.com/blogs/articles/how-to-calculate-propeller-thrust?srsltid=AfmBOooleXPEWdIui0XQD1EwkoKIBElChglpRuKlnJJXuC1sXqsJpkZn (accessed Feb. 28, 2026).

[6] E. II, “Emax ECO II Series 2306 1700KV 1900KV 2400KV Brushless Motor for RC Drone FPV Racing,” *Emax*, 2020. https://emaxmodel.com/products/emax-eco-ii-series-2306-1700kv-1900kv-2400kv-brushless-motor-for-rc-drone-fpv-racing (accessed Feb. 28, 2026).