Building Interior Reconnaissance Drone

By

Jack Lavin
Jacob Witek
Mark Viz

Final Report for ECE 445, Senior Design, Spring 2026
TA: Shiyuan Duan

6 May 2026
Project No. 25

Abstract

There are many situations which require first responders, whether law enforcement, medical, or rescue services, to enter rooms with dangerous conditions to verify the presence and location of people within and act accordingly. Currently, the best methods to do so with acceptable accuracy generally require a human or animal to also enter the hazardous environment. Since the increase in popularity and commercialization of quadrotor drones across various sectors of the U.S. market, there has been an increase in the utilization of drones by military, police, and EMS to enhance their capabilities. Despite this increase, most modern systems require a pilot's full attention to control the drone and relay information while a separate operator receives the information verbally. We propose a system to identify human presence and relative position within hazardous rooms using a semi-autonomous quadrotor drone with human sensing and an augmented-reality companion app to eliminate the risk to human life in these scenarios.

Contents

1. Introduction

1.1. Problem……………………………………………..…………………………..3
1.2. Solution……………………………………………..…………………………..3
1.3. High Level Requirements………………………………………..………..3
1.4. System Overview…………………………………..………………...5

2. Design

2.1. Power & Propulsion………………………………………..……………...6
2.2. Presence Sensing……………………………………..…………………7
2.3. Flight Control……………………………………..……………………9
2.4. Augmented Reality Interface………………………………………………….12
2.5. Physical Design…………………………………..…………………14

3. Requirements & Verifications

3.1. Verification Procedures……………………………………………………..15

4. Cost & Schedule

4.1. Cost……………………………………..……………………………..17
4.2. Schedule……………………………………..…………………………...18

5. Conclusion

5.1. Accomplishments…………………………………..……………….19
5.2. Uncertainties…………………………………..……………………20
5.3. Future Considerations…………………………………..….………..21
5.4. Ethical Considerations……………………………………………………...21
6. References……………………………………..……………………………..22

Appendix A: Requirements & Verification Tables…………………………………………….23
Appendix B: Circuit Schematic………………………………..…...……………24
Appendix C: Printed Circuit Board (PCB)……………………………………………………25

1 Introduction

1.1 Problem

First responders find themselves in hazardous situations on a daily basis. Obtaining information about the location of people in their immediate surroundings quickly and easily, without putting themselves in danger, is of vital importance to ensure they are able to do their job effectively and safely. There is currently no comprehensive system which allows for one operator to do this effectively.

1.2 Solution

The Building Interior Reconnaissance Drone, further acronymed as B.I.R.D, aims to quickly provide users with information on people present in an unseen room in an easy to understand format. The system uses a drone to fly into a room and perform a 360° rotation while scanning for the presence of humans. This data is then sent back to a user's device running our custom alternate reality program. Human presence data is then displayed on a custom app along with distance information, allowing users to quickly and easily understand exactly where people are present in a room without entering that room first.

1.3 High Level Requirements

To successfully accomplish the above basic description of operation, the B.I.R.D must satisfy the three following requirements.

1. A semi-autonomous vehicle that can provide presence sensors with a vantage point with six modes of movement: lifting, hovering, forward, 360° rotation, reverse, and landing as demonstrated by Figure 1.
2. Onboard sensors that are able to detect stationary human presence while on a moving platform.
3. Transmit, calculate, and display the relevant human presence data onto an alternate reality camera feed as demonstrated by Figure 2.

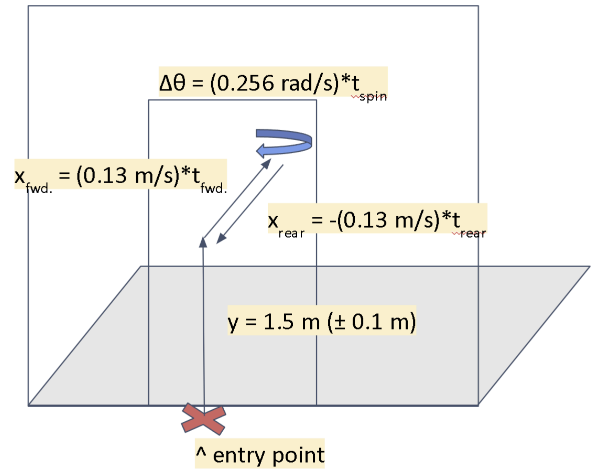


Figure 1: Predetermined Autonomous Flight Path

Figure 2: Hypothetical View From Augmented Reality (A.R) App

The first high level requirement is to create a moving device that can be deployed from outside a potentially hazardous room and can autonomously move inside the room. It also needs to house human presence detection sensors to observe the entire environment while the sensors themselves are stationary. Furthermore, this first requirement allows the device to be retrievable by the user at the same launching position and thus allows for multi-use cases.

The second high level requirement is responsible for determining local positioning of people within a room with respect to the device's current location. For the current use case of this project, the presence sensors are able to detect people that are either standing or sitting and may be behind slight obstruction, such as desks, chairs, or other objects that would still have some line of sight between themselves and the device.

The third requirement relays local positional information about people from the device to the user's mobile device and applies triangulation to create a direct line of sight between the user's current location and a person within the room.

1.4 System Overview

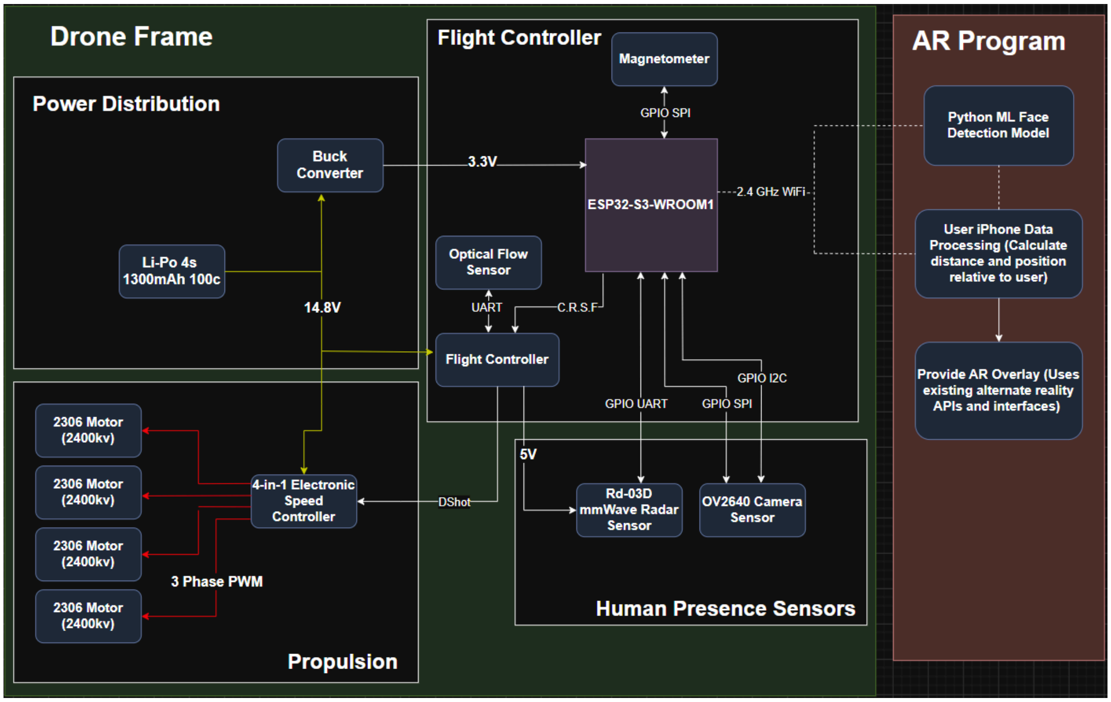


Figure 3: Top Level Block Diagram of the B.I.R.D System

Three of four subsystems are located on the physical drone frame, and their placement is detailed in section 2.5 below. The section responsible for supplying power and performing the physical movement is the joint Power Distribution & Propulsion subsystem. It does so by supplying the required nominal 14.8 [V] to 16.2 [V] to the electronic speed controller and flight controller as well as a stable 3.3 [V] to the microcontroller (ESP32-S3). Furthermore, the propulsion component of this subsystem receives a signal from the flight controller and responds by sending specific throttle values to each of the four motors.

The next subsystem is the Human Presence Sensor System. This system communicates with the ESP32-S3, and consists of a camera and radar sensor that work in tandem.

The flight controller subsystem is responsible for flight correction through the use of an optical flow sensor. It is also responsible for stability through the use of a flight controller module. The ESP32-S3 moreover operates as a wireless host to a WiFi web server that allows for local information to be transmitted to the external subsystem, the AR program interface.

Lastly, the AR subsystem consists of an external device that runs a machine learning algorithm with Python to process camera frames from the sensor subsystem as well as an iPhone that is used to display human position data from radar presence information and post processed camera frame information.

2 Design

2.1 Power and Propulsion

The primary considerations when designing and implementing the power/propulsion system were the thrust-to-weight ratio (TWR) and flight time, which is dictated by the battery capacity and motor efficiency. During the initial design process, the first decision made was to use a standard carbon fiber frame designed for a 5” propeller quadrotor drone. Since our sensor module was still being designed, we wanted to ensure a large TWR to support a potentially heavy sensor apparatus if needed.

A 1300 mAh capacity 4S Ovonic Lithium-Polymer battery was used to supply power to all modules through various regulators and converters. The battery is rated for a discharge rate of 100C, or one hundred times its capacity per second, allowing for a maximum of 130 A of continuous current to the flight control system, motors, and sensors. Initially, we purchased four TOYTENSI ESC’s rated for 40 A to supply the proper three-phase current to the motors we would use. These ESC’s were chosen so that they would not limit the max current capability of our battery, however, during our extensive PID loop testing, we determined that they did not have a fast enough sampling rate to support stable flight with our IMU data. After further research, we determined that these ESC’s were primarily intended for use with fixed wing RC aircraft, leading us to source new Sequre brand ESC intended for quadrotor drones with a 65 A (per motor) rating and DSHOT support for better control.

With our battery and ESC chosen, we purchased four ECOMAX II 2306 brushless DC motors with RPM capabilities of 2400 RPM per volt of battery potential. Each motor has a maximum thrust of around 1350 grams force or 13.224 N. With all subsystems assembled on the frame, the drone has a mass of about 750 grams, resulting in a final TWR of 7.2. This is well beyond our requirement for stable hovering; when maintaining a constant height from the ground the motors maintain 25-30% of their max thrust output.

25%	Thrust (g)	Amps	Volts	RPMs	Watts	G/W
HQ v1s 5x4x3	139	1.79	16.11	10346	29	4.8
T-Motor 5143 5.1x4.3x3	148	2.29	16.11	10208	37	4
HQ v1s 5x4.3x3PC	157	2.28	16.12	10275	37	4.3
DAL Cyclone 5x4.5x3	158	2.29	16.11	10223	37	4.3
Lumenier Buttercutter 5x5x3	161	2.29	16.11	10258	37	4.4
HQ 5x4x3GF	162	2.26	16.11	10266	36	4.4
Gemfan 5149 5.1x4.9x3	170	2.36	16.11	10170	38	4.5
DAL Cyclone 5x4.6x3	171	2.42	16.11	10124	39	4.4
HQ v1s 5x5x3	185	2.6	16.11	9962	42	4.4
HQ 6x3.5	191	2.45	16.11	10104	39	4.8
KingKong 6x4	195	2.6	16.11	9970	42	4.6
HQ v1s 5.1x4.6x3	198	2.79	16.11	9786	45	4.4

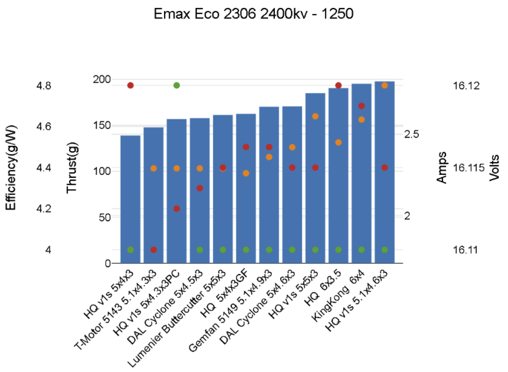


Figure 4: ECOMAX II 2306 2400kV Brushless DC Motor Test Data from MNTB

Figure 5: View of B.I.R.D. Sustained Hover from AR Application

2.2 Presence Sensing

The presence sensing subsystem consists of two sensor modules, an Ai-Thinker Rd-03D 24 GHz radar and an ArduCam OV2640 camera. In the current implementation of the project, these sensors work in tandem such that both provide independent information on the number of people and the detected persons angle relative to the front of the drone. Given the limitations of the computer vision model used to process images, the distance measurement can only be obtained by the Rd-03D radar sensor. When there is discrepancy between the data measurements, an algorithm running on the ESP determines which data to use.

The Rd-03D is a radar module that integrates an S5KM312CL IC, which is a microcontroller that operates frequency modulated continuous wave (FMCW) radar at 24 GHz with a bandwidth of 250 MHz. To briefly explain FMCW radar, a transmitted signal (a Tx chirp) is compared to a received signal (a Rx echo) and is then transformed to the frequency domain via a Fast Fourier Transform (FFT). For distance measurement, a greater difference in frequency indicates a greater distance and for angle measurement, the shift in phase of the frequency peak indicates a shift in angle. For example, an angle of 0° would produce a phase of 0+j0 whilst an angle of 90° would produce a phase of 0+j1. A visual description of a distance measurement is shown on the left in of *Figure 6* whilst a visual description of an angle measurement is shown on the right of *Figure 7*.

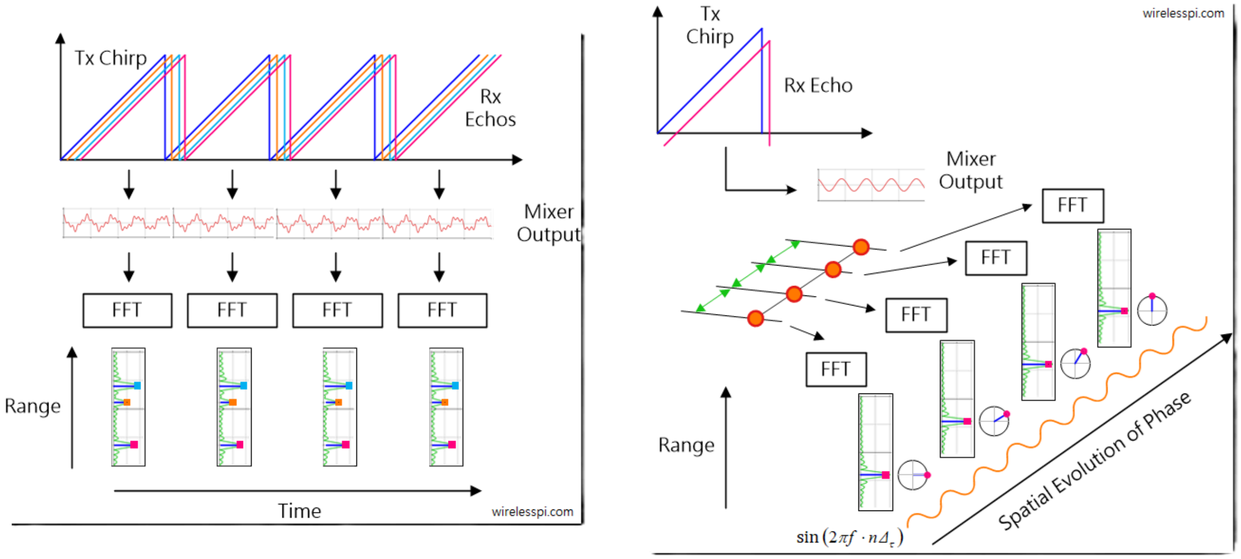


Figure 6 (Left) Theoretical Distance Measurement,
Figure 7 (Right) Theoretical Angle Measurement

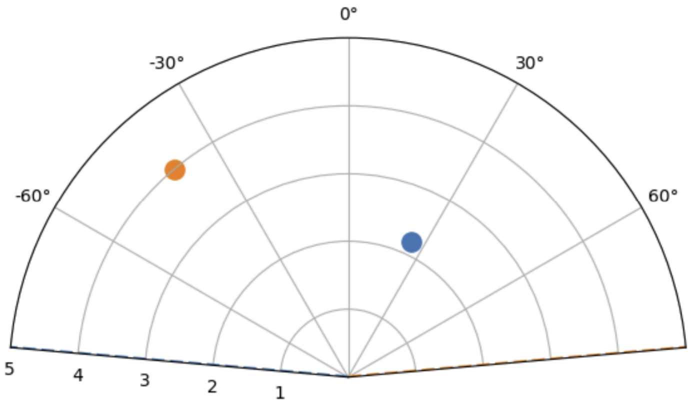


Figure 8: RD-03D Calculated Range Polar Plot

From the verification procedure for the Rd-03D as specified in section 3.1, the polar plot of the visible region above was constructed using Python, and contains two separate people at distances of approximately 4 meters and 2.2 meters. Some details that are not distinguishable from the polar plot are that the radar is not able to correctly discern two people standing right next to each other and marks them as a single person as well as two people standing one behind the other, in a column fashion. Thus, this led to the use of a second sensor to clear up these two issues.

The second sensor that is used in this subsystem is the ArduCam AV2640 camera module. This module operates using I2C communication for register initialization, and SPI communication to transfer images containing a 60° horizontal field of view from the module to the ESP32-S3 microcontroller. For this sensor, the ESP32-S3 displays the images on the /capture path on the WiFi-hosted web server. Each 360° rotation takes approximately 22 seconds, and thus to capture the whole field of view without creating duplicate edges for the image corners a 320x240 image is transmitted to the web-server every 3.5 seconds. This timing is sufficiently long enough to allow for each image buffer to finish uploading and not be overwritten.

To analyze each image, an external laptop running a Python script passes each retrieved image through Ultralytics' YOLO11 vision model and plots bounding boxes on humans when the confidence level is over 0.25. As such, this vision model is able to differentiate two people that are close together if both of their faces are visible. To calculate the respective angle of each person relative to the front of the camera, the horizontal pixel of the bounding box of each human is compared to the central horizontal pixel. Below is an example, where the bounding box of a human is at 240 pixels, and thus is located 15° to the right of the camera.

$$60^\circ\, FOV\ /\ 320\, pixels\ =\ 0.1875^\circ\ /pixel$$
$$pixel\_offset\ =\ bbox\_center\_x\ -\ 160$$
$$offset\_angle\ =\ pixel\_offset\ *\ 0.1875^\circ$$
$$offset\_angle\ =\ (240\, pxl\ -\ 160\, pxl)\ *\ 0.1875^\circ\ =\ 15^\circ$$

In previous editions of this project, we had used the DFRobot C4001 FMCW sensor that operated on the same principle as the RD-03D. During testing with vibrations (to simulate the sensor being on a drone), the sensor displayed significantly incorrect distance measurements as well as significant false negative that would not detect people while they were present, and thus was not able to sufficiently satisfy this subsystems' requirements. To attempt to fix this issue, the tolerance of the sensor was increased - however this fine tuning was not able to find a middle ground that would not have false negatives with people within 5 meters. This led us to look into the RD-03D. The RD-03D was significantly better in this regard, as it contained two receiving antennas versus the C4001's just one - and thus was able to more accurately compare the received data.

2.3 Flight Control

The flight control subsystem is responsible for ensuring stable flight such that the sensor subsystem is able to perform its functionalities. The flight stabilization is made up of multiple hardware components, a robust stabilization interface, and a state based flight path program.

To ensure that the drone is stable while flying we made use of two main hardware components. The first component is the ICM42688P inertial measurement unit (IMU). This component includes a 3-axis gyroscopic sensor and a 3-axis acceleration sensor. From this IMU we are able to derive information on the drones orientation, linear velocity, and rotational velocity. These sensors work in tandem to determine the drones' overarching motions at any given time. This information is passed onto the flight stabilization interface discussed below to allow the drone to perform the motions necessary for our flight path. The second piece of hardware critical for keeping our drone from drifting is the MTF-01. This component includes both a LIDAR and optical flow component. The LIDAR sends out a pulse and times how long it takes for the pulse to return. This allows for accurate distance measurements. When pointed at the ground, this gives our drone information on its current height. The optical flow sensor is a type of low-resolution camera. It periodically scans the ground below it and compares pixels to previous frames. If the pixels have shifted we know the drone is drifting in a certain direction and must be corrected. This component is essential as it picks up on the slight drifts that the IMU usually misses. One oversight we had while working with earlier IMUs (such as the LSMD6SOX) was we had not taken into account the severe vibrations the drone caused. This led to the IMU readings to be inaccurate and noisy. We had to make a design alteration, and find IMUs more resistant to noise such as the ICM42688P.

Using the data provided by the sensors described in the paragraph above, we now need to make individual motor adjustments such that the drone is able to reach a desired pitch and angle. This was done using the HGLRC flight controller while running the open source flight software INAV. The program feeds the IMU and MTF-01 data into a proportional integral derivative (PID) equation. This equation has 3 components. The first component is proportional to the current angle of the drone. So if the angle is high this component will be high. The second component is an integral of past angles of the drone; So if the drone is not rotating despite being at an undesired angle, the integral component will slowly add more force over time until it is back to our desired angle. The last component is the derivative, which takes into account the direction of the rotation and applies a counter force; For example if the drone is rotating towards an undesired angle this component will apply a force against it. If the drone is rotating towards a desired angle this component will again apply a force against it so the drone doesn't overshoot. These 3 PID components together are used to determine the throttle value of each motor. PID constants were determined through drone auto-hover tests. One major design alternative that we had to take was switching from a custom PID system to the more readily available PID system INAV. We had made significant progress with our own PID program, but it was not sufficient for the stability and precision required for our drone to fly through rooms. If given more time we would like to go back and continue our custom PID and flight controller work.

The final component of the flight control is the state based flight path system. This program runs entirely on our ESP and communicates with the flight controller using CSRF packets. The program consists of 10 total states which each need to achieve a certain goal before moving onto the next state. The states and their functionalities along with the state machine are described in the table and image below:

State	Functionality
Ready	The drone is ready for user input. Motors and stability systems are off.
Preflight	The drone is armed and the motors begin to power up.
Take Off	Flight stability is enabled and the drone begins to lift off to its desired position.
Hold	The drone stabilizes itself in preparation for the next state.
Forward	The drone moves at a set speed forward for a distance indicated by the user.
Hovering	The drone activates all human detecting sensors and begins to rotate. The drone completes a full 360 in this state.
Backward	The drone flies back to the same location it took off at.
Pre-Land	The drone begins to descend.
Landing	The drone disarms motors and stability systems.
Force Land	During most states this state can be called to send the drone straight to the pre-land state. This is dependent on user input.

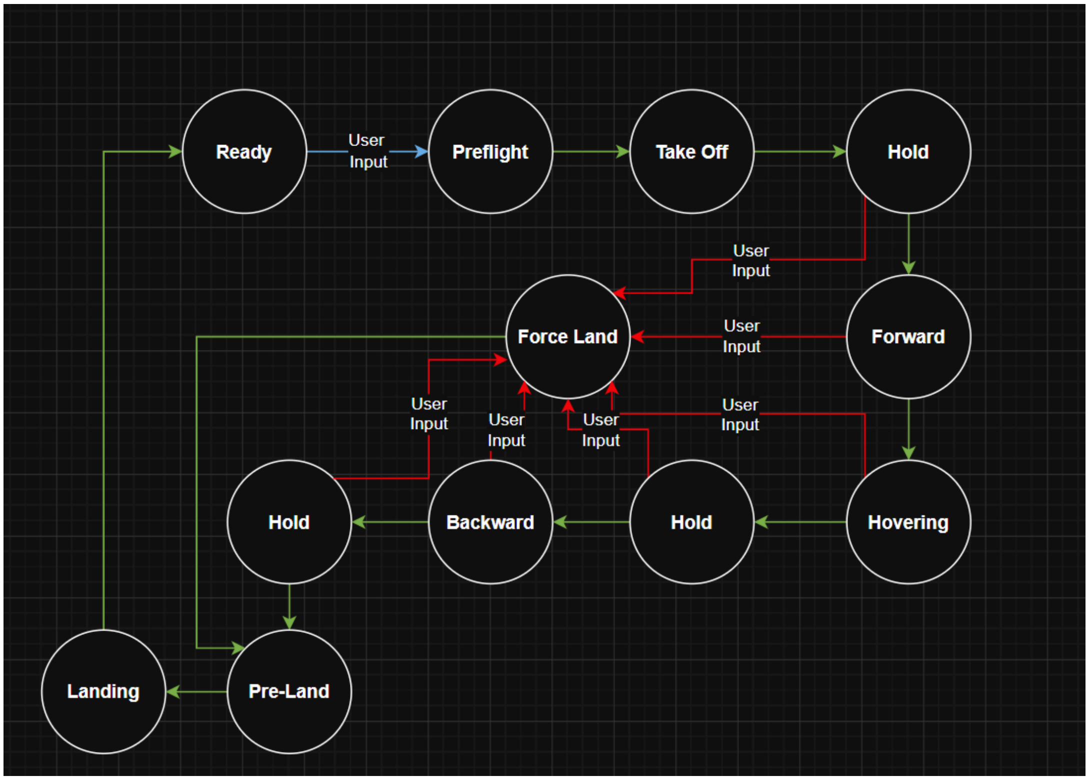


Figure 9 Finite State Machine of Flight Control

The ESP flight path program has full control over throttle, pitch, roll, and yaw values and sets them depending on the current state to execute the drone movement expected.

2.4 Augmented Reality Interface

The augmented reality (AR) app allows the users to view the human presence information in an informative and easy to understand format. The app displays each human detected as a red person icon with a size that is proportional to the distance that the user is relative to that detected person. The app works by first connecting to the ESP's webserver. The user must be standing directly behind the drone when connecting on the app. Once connected the drone and the app create an "anchor point" and an x-y grid from which all positional data is calculated from. The app is written in the swift language and makes use of Apple's ARKit to provide an accurate and effective AR experience.

When the user hits "Start" on the app, the drone enters its *Pre-Flight* state described above. User imputed flight distance information is also sent at this time. As the drone continues its flight path and detects humans it posts detected humans to the <drone_ip>/data part of the webserver. The app is constantly polling this address for information. Once data is posted to the webserver the app collects angle, distance, and human count information. It then calculates the position of the icon relative to the anchor point. While moving around the app also keeps track of the user's current position from the anchor point. Knowing these three positions, the app is able to calculate where the human is relative to the app's user. An example use scenario and the app's calculations are shown below. We are trying to calculate the blue line in the image below:

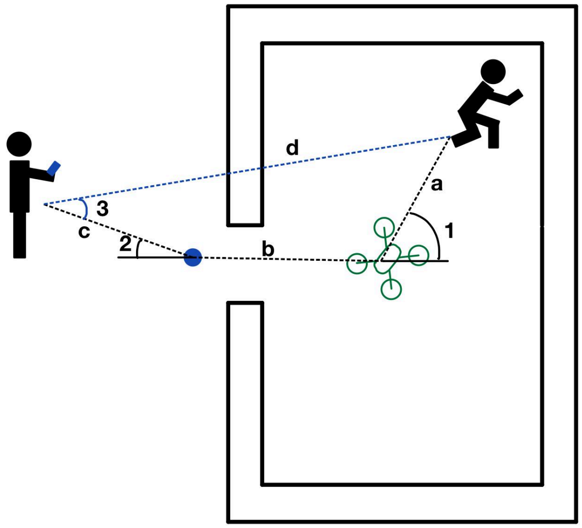


Figure 10 Example Scenario and Related Equations

X-Component: cos(1) * a + b + cos(2) * c
Y-Component: sin(1) * a - sin(2) * c

Depending on how many people are detected the app is also able to display different icons. If more than one person is determined to be present, the app uses a multi-person icon. Otherwise a single icon is used. The app is also able to display distance information if you tap on a detected person icon relative to where the user is currently standing. Lastly you are also able to see the voltage and heading values. This information is collected by polling the /volt and /head sections of the webserver which are given using an onboard magnetometer:

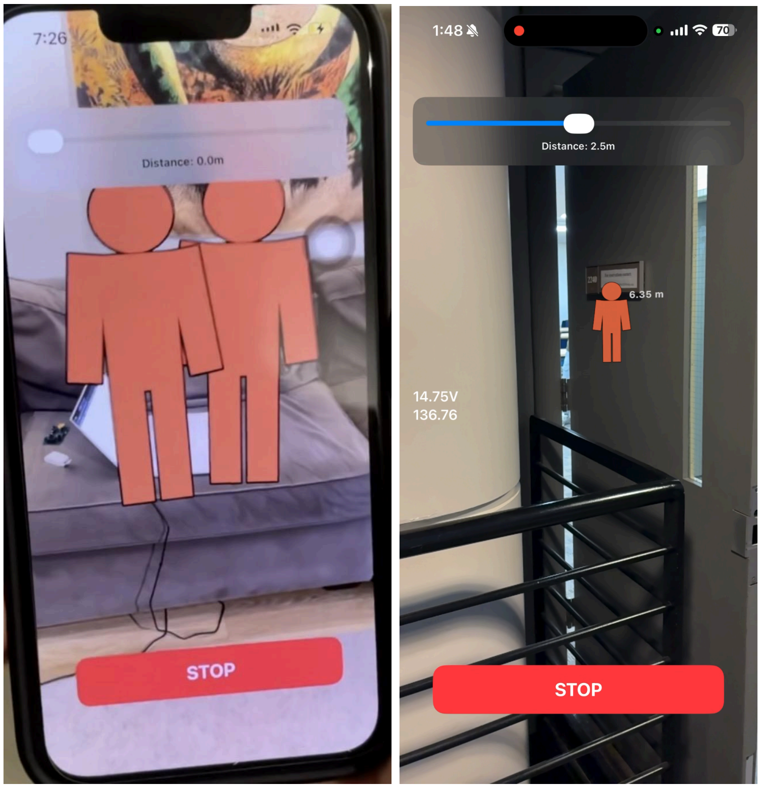


Figure 11 (Left) Multiperson Icon, Figure 12 (Right) Single Person Icon with Distance

2.5 Physical Design

The body of the drone is separated in three layers. From *Figure 13*, the top layer contains the PCB, the middle layer contains the flight controller and electronic speed controller modules on top of each other, and the bottom layer contains the LiPo battery. The battery is contained via velcro straps that are taped onto the bottom of the frame and top of the battery. Moreover, there were three 3d prints required - four legs colored in grey, four propellor guards colored in blue and orange, and a third print being a frontal mount to contain the camera and radar sensor modules.

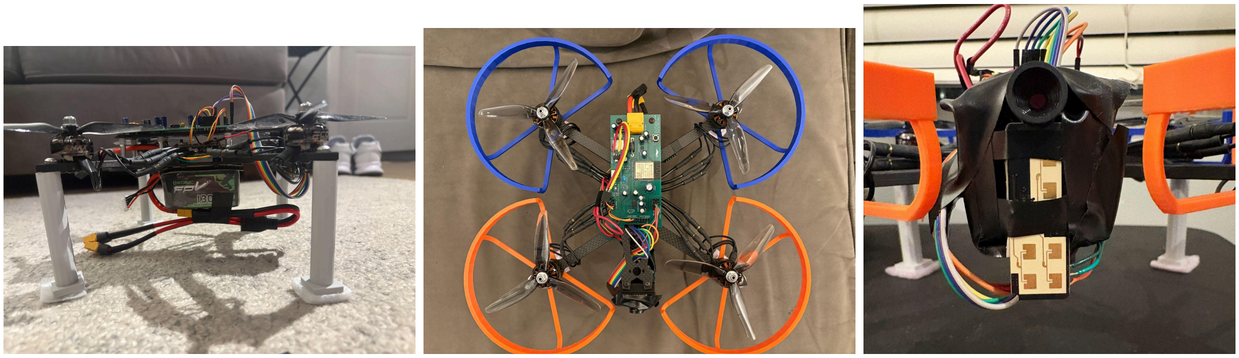


Figure 13 (Left) Drone Side, Figure 14 (Middle) Drone Top, Figure 15 (Right) Drone Front

3 Requirements & Verification

3.1 Verification Procedures

Power and Propulsion

We first observed that the drone would begin to lift itself at approximately 40% thrust. This thrust percentage would be the theoretical maximum that the drone motors would occur during a flight test as lifting requires the most force upwards. We ran the motors in place at 40% thrust while holding the frame down for 1 minute and 30 seconds which is the time required to perform a flight test with a distance of 2 meters. From testing, the first test on average drops the battery voltage by 0.95V, the second run drops the voltage by 0.55V, and the third test by 0.50V. Thus, once the drone landed from the third test, the battery voltage was on average 14.2V. We ran this test 5 times, and verified that the battery is capable of running 3 consecutive flights. From flight tests, when the battery voltage dropped below 14V the power distribution to the four motors was not evenly distributed and caused the drone to always lose stability and fall, therefore three tests is the maximum number that this drone could perform without having to recharge.

Presence Sensing

For the OV2640 camera, we ensure that images are being accurately captured onto the WiFi web-server by observation on the /capture path through code logs. Once these images are ensured to be correct, the images are passed through the Python script running the YOLO11 model, and output images with bounding boxes around all the determined humans as well as the model's confidence that these bounding boxes are correct. Once the setup is complete, we placed the camera at a height of 1.5m above the ground, as this is the height that the drone will fly at when it performs a 360° rotation. Next, we created a 60° cone at a maximum distance of 5 meters in front of the camera that a person would stand at in different positions. Running the model, we observed that it was able to detect people that were facing forward towards the camera as well as facing sideways. The model would also detect people who were standing and in sitting positions, such as on a couch or chair. We ran this test 10+ times with multiple configurations and verified the model was sufficient for our needs. Some drawbacks of this camera is that it would not accurately display people who were sitting but facing away from the camera, and it would not work well in a low light environment - such as a room with all lights off and blinds closed.

For the RD-03D radar module, we placed this sensor at the same position as the camera as they are placed on each other as shown in *Figure 15* on the drone. During testing, a person would stand at some spot within the 60° cone and an observer would mark down the position that the radar states the person is in. Then, we used a tape measure to determine the correct distance to the sensor as well as the iPhone's compass for angle. Running this test 15 times in areas throughout the region, we determined that on average the RD-03D was within 0.4 meters of accuracy to the actual distance, and their angle was within ±5° from the correct reading. Furthermore, there were no false readings - both false negatives or false positives during any of the tests within the 5 meter region. In determining this maximum range of operation, we performed an additional 15 tests at locations such as 80° from center, 20° at 7m, etc, but began to observe fluctuation in the distance readings such that the accuracy was now ±0.8m for some of the extreme readings, while about within 0.6m on average. We verified that these parameters would be suitable for our use case.

Flight Control

The first part of verifying the flight controller entails ensuring that the drone could stably fly with minimal drift. To verify this we did multiple rounds of preset test programs which would fly the drone to a set height, hover in place, then land again. We would then use rulers to mark the takeoff and landing positions and ensure that the drone did not drift by more than 20 cm. We are using 20 cm as we deemed this to be the maximum allowable drift value that could allow the drone to reasonably fly around an enclosed room. After testing 10+ times we were able to verify the drone is capable of this task.

The next part of verification entailed ensuring forward movement, 360 rotation, and backward motion. This would also verify that the flight state program was working. To do this we repeatedly tested the flight program and visually observed that the drone would complete its tasks. After testing we were able to verify the drone is capable of this task.

Lastly we had to verify that the drone would move forward and backward distances that were reflective of what the user set the drone distance to be. We gave an allowable drift of 0.5 m of set distance. We determined 0.5m to be a maximum allowable error while still being able to achieve accurate functionality. To test this we repeatedly tested setting the drone's distance, running the program, and measuring the actual distance traveled. After testing 20+ times we were able to verify the drone is capable of this task.

Augmented Reality Interface

To verify the functionality of the AR interface required testing multiple arrangements of people in a room, and then running the flight program and visually verifying that the people showed up on the screen within 5° and within 0.5 m. We determined these values to be within the range acceptable for a user to verify the location of a person in a room. We also verified that the correct person icon appeared on the screen depending on how many people the drone detected. After running the program with 20 different configurations, around 16/20 of the tests were perfect, with all 20 of the tests detecting people in some regard. This was acceptable for our tasks.

4 Cost & Schedule

4.1 Cost

Description	Manufacturer	Quantity	Extended Price	Link
ESP32-S3-WROOM1	Espressif Systems	1	$6.13	Digikey
FPV Brushless Motors 4-pack 2400kV 3-4S	Emax	1	$65.99	Amazon
65A 4-in-1 Electronic Speed Controller	SEQUIRE	1	$39.99	Amazon
Flight Controller	HGLRC	1	$34.99	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
LiPo Battery Charger	Tenergy	1	$57.99	Tenergy
VL53L1X IR Time of Flight Sensor	Dweii	2	$29.98	Amazon
PMW3901 Optical Flow Sensor 2-pack	PixArt	1	$30.99	Amazon
RD-03d mmWave	Ai-Thinker	1		
XL1509-3.3 Buck Converter	Evvo	1	$0.35	Digikey

XL1509-5.0 Buck Converter	Evvo	1	$0.35	Digikey
User Phone	iPhone	1	N/A	N/A
Miscellaneous PCB Components (capacitors, resistors, inductors, terminal blocks)	Digikey Order	N/A	$66.59	N/A

By current estimations, each group member is contributing, on average, 8 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 * 8hr/week * 13 weeks = $9100

With three group members, our total labor cost is 9100*3 = $27,300 and a total cost for each drone being $393.3. This brings the total cost of this project to be $27,693.

4.2 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 RD-03D 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 **3/8 - 3/14**	- Create breadboard that integrates RD-03D, IMU, Time of Flight, and optical flow sensors with ESP32 devboard -Ensure basic stability for hovering with the drone -Solder initial PCB and perform various testing	Everyone
Week 7 ****Spring Break**** **3/15 - 3/21**	- Continue PCB testing - Revisions to PCB design - Submit 2nd PCB for 4th round orders -Switch to INAV based stability system	Everyone
Week 8 **3/22 - 3/28**	Test new INAV systems and create ESP32 interface to allow for full drone control from ESP	Jacob Witek Mark Viz
	-Create an web server interface for ESP32 data to an Apple application -Begin work on flight state system for movement control	Jack Lavin
Week 9 **3/29 - 4/4**	-Test and solder 2nd edition PCB -Create and test drone forward and 360° 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 Apple application and create augmented reality display (AR overlay)	Everyone
Week 11 **4/12 - 4/18**	Stress test the project, ensuring all subsystem requirements are met	Everyone
Week 12 **4/19 - 4/25**	Finalize all testing and make tuning adjustments to flight paths and user interface	Everyone
Week 13 **4/26 - 5/2**	Design, print, and install drone guards	Everyone

5 Conclusion

5.1 Accomplishments

Regarding base functionality, we believe the drone has completed its goals exceptionally. The first critical goal we were able to achieve was stable autonomous flight. The drone is able to consistently and safely fly forward into a room, do a full 360, and return back to its takeoff position. The second goal of note is that we were able to provide relevant accurate human presence information for individuals standing in a room. Our sensors and algorithms that collect data are able to effectively determine humans and their distances’. Lastly we were able to display this information in an intuitive format such that it is clear how many people are present and where they are in relation to the user. These were the main goals that we set for our project originally. From our tests the distance, angle, and number of humans present was consistent and accurate to within a few degrees and ± 0.4m. We believe that in emergency situations the B.I.R.D in its current state has the potential to provide useful information.

5.2 Uncertainties

There are a few uncertainties we have encountered while testing our drone. First is with regards to flight stability in certain undesirable conditions. For example, if the floor is excessively reflective or in some way changing, this can throw off the drones positioning sensors and cause undesirable movements. Furthermore, if the lighting in the room is dark, the optical positioning sensors again could be thrown off resulting in sporadic movement. Another problem is the drone has no understanding of its environment. If it has obstacles that have fallen or moved in front of its path it will be unable to dynamically adjust to move to safer conditions. These are issues that, given more time, we would like to address.

Another uncertainty that we considered is how the drone would react when people move while the drone is sensing. As the drone rotates, if one person is caught moving between frames they would show up as two people. This could cause confusion on the true number of people in a room. Also, if a person quickly moves from a spot they were in when the drone sensed them, the user’s icons will become void and inaccurate. We designed the sensing assuming static people as we ideally are trying to identify people who are stuck and can't make their way to the door themselves, but we acknowledge that emergency situations are dynamic, and we should account for all movement possibilities.

5.3 Future Considerations

Given more time to work on this project we would like to cover 3 major areas. The first is expanding the B.I.R.D’s working environment. This would entail adding low light mode and LEDs so the optical positioning sensors would not be compromised in low light scenarios. Furthermore, in low light mode, the B.I.R.D would automatically resort to its radar based human presence sensing as opposed to its optical ones. We also would like to add a SLAM based positioning system using LIDAR to allow the B.I.R.D to react dynamically to obstacles in front of it. To take it a step further we could have it map out rooms and algorithmically choose the best point to begin scanning for humans.

The second major area is further improving the B.I.R.D's speed and stability. Currently the drone is relatively slow and wastes precious time in emergency situations. We would like the drone to move faster when it comes to entering the room, rotating, and leaving the room. This would provide first responders with the critical information they need faster.

The final area of improvement is with regards to more dynamic human sensing. As discussed above, we would like to expand the B.I.R.D's sensing capabilities to account for already seen people and moving people such that the information relayed is as accurate as possible. Overall we are excited with the possibilities on where we could take the B.I.R.D.

5.4 Ethical Considerations

Aerial drones are currently a major topic of debate regarding the ethics of their use for both surveillance and military purposes. Recognizing the potential for misuse of this system and technology, the B.I.R.D. is strictly intended for use to aid emergency medical and rescue service professionals. In accordance with the IEEE Code of Ethics, we insist that any use of this technology or implementation of this system "... hold[s] paramount, the safety, health, and welfare of the public, … strive[s] to comply with ethical design and sustainable development practices, [and] … protect[s] the privacy of others."

The IEEE Code of Ethics also requires that members "disclose promptly factors that might endanger the public or the environment." To ensure compliance with this principle, we hereby make known to any user of this system the potential hazards involved with operating the B.I.R.D. system in its entirety:

- The B.I.R.D. is powered by a LiPo battery with significant capacity and discharge capability. LiPo batteries pose many potential risks to personal safety if punctured, exposed to excess heat, or operated outside rated voltage ranges. The B.I.R.D. should never be charged or operated unless full consideration of LiPo hazards are understood and proper safety measures are taken. To mitigate the risk of electrical failure or battery depletion during flight, the B.I.R.D. should never be launched if the voltage of the LiPo battery under no load exceeds 16.2 V or is below 14.6 V.
- The B.I.R.D. and its propellers are capable of extreme acceleration and top speed. Although the autonomous flight operation and paths have been tested rigorously, the B.I.R.D. should never be operated without propeller guards installed in the case of lost control or electrical failure.
- In addition to the considerations above, it must also be noted that the B.I.R.D. in its current implementation is a prototype, and as such should not be utilized as-is for the

described purposes in its current state. Those who wish to attempt to replicate the system for testing or improvement must understand the potential risks to the safety of both personnel and property.

6 References

[1] Ai-Thinker, "Ai-Thinker LoRa Series Modules Specification," *Ai-Thinker*, Oct. 16, 2023. [Online]. Available: https://en.ai-thinker.com/Uploads/file/20231016/20231016032622_13559.pdf. [Accessed: May 6, 2026].

[2] Q. Chaudhari, "FMCW Radar Part 1 – Ranging," *Wireless Pi*, Sep. 30, 2023. [Online]. Available: https://wirelesspi.com/fmcw-radar-part-1-ranging/. [Accessed: May 6, 2026].

[3] Q. Chaudhari, "FMCW Radar Part 2 – Velocity, Angle and Radar Data Cube," *Wireless Pi*, Nov. 6, 2023. [Online]. Available: https://wirelesspi.com/fmcw-radar-part-2-velocity-angle-and-radar-data-cube/. [Accessed: May 6, 2026].

[4] Mini Quad Test Bench, "Emax Eco 2306 2400kv," *Mini Quad Test Bench*. [Online]. Available: https://www.miniquadtestbench.com/assets/components/motordata/motorinfo.php?uid=259. [Accessed: May 6, 2026].

[5] Apple Inc., "ARKit," *Apple Developer Documentation*. [Online]. Available: https://developer.apple.com/documentation/arkit. [Accessed: May 6, 2026].

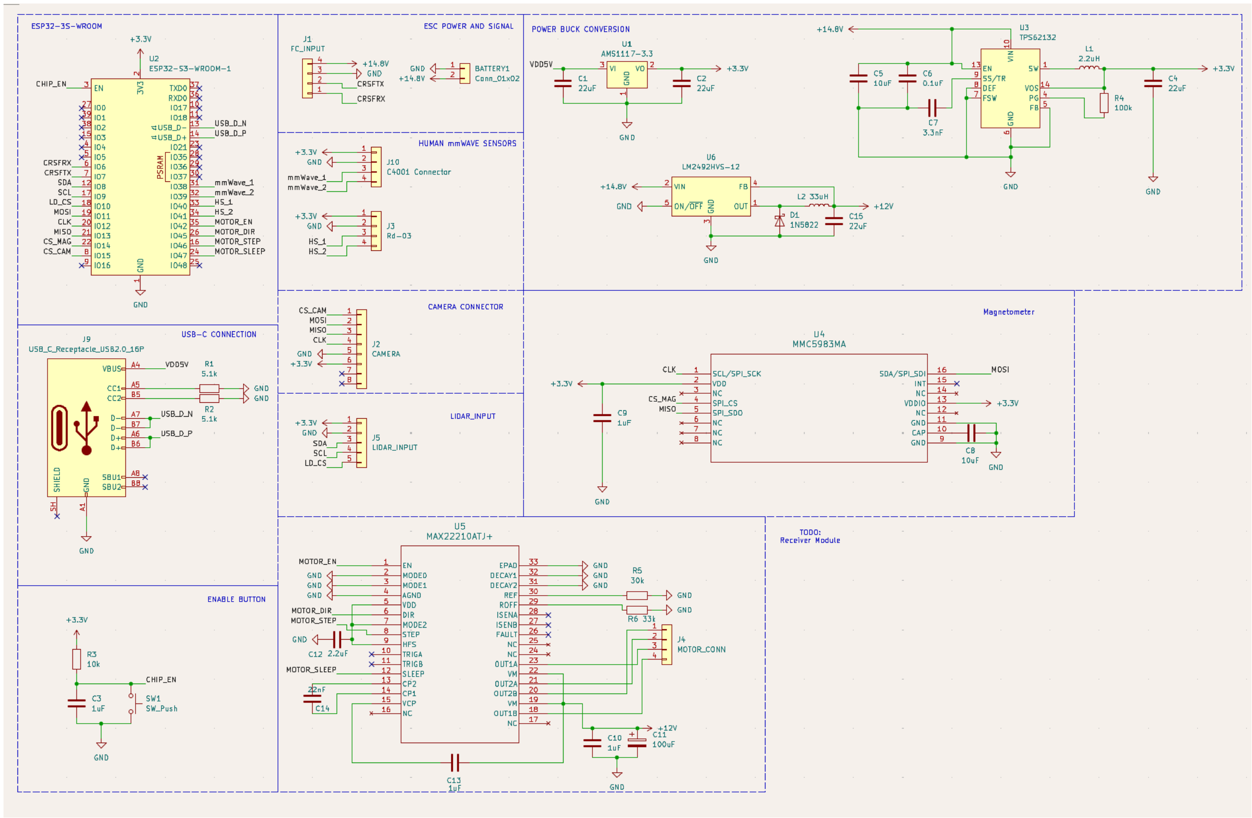
[6] IEEE, "IEEE Policies, Section 7.8," *IEEE*, 2026. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8. [Accessed: May 5, 2026].

Appendix A: Requirements & Verification Tables

Requirements	**Verification**
One battery shall provide enough nominal power to run three sequential flight tests.	Charge LiPo battery to approximately 16.2V, and read the voltage after each flight test with 2 meter set forward movement using the battery charger. Measure the voltage drop across each test, and ensure that the third test does not start when the voltage is below 14.5V.
Relay number of humans and local angle of each individual relative to each 60° increment at a maximum distance of 5 meters.	Transmit 320x240 image every 3 seconds to host server. Observe correct 'human' classification and angle offset from central image point in high light environment via an ML model.
Constantly relay local angle and distance information of stationary humans at a maximum distance of 5 meters.	Observe human presence with ±5° angle offset from centerpoint and ±0.5m distance measurement.
Safely and efficiently take off, move forward a set distance, survey the room, and return to take off position.	Run flight tests at different distances and voltage levels to ensure the drone consistently and accurately takes off and moves set distances. Once test is complete, verify the drone has completed its flight path accurately using measurement devices.

Relay information collected from previously discussed presence sensing and display it in with AR format that is easily understandable and provides relevant position information.	Run presence detection tests with multiple people in different configurations and ensure that the app overlay is reflective of the people's positions.

Appendix B: Circuit Schematic



Appendix C: Printed Circuit Board (PCB)