



UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Building Interior Reconnaissance Drone (B.I.R.D)

Team 25

Jack Lavin

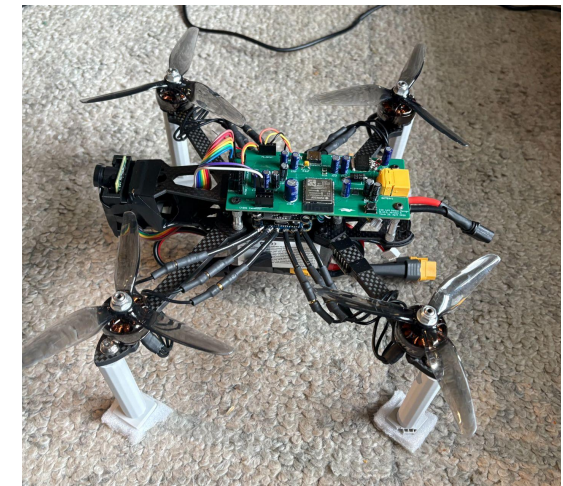
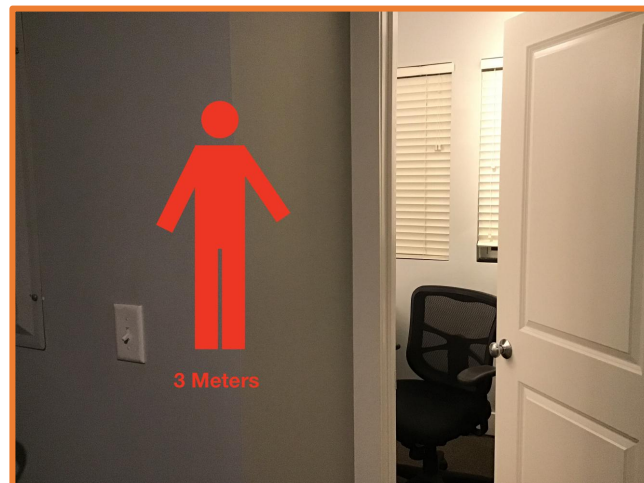
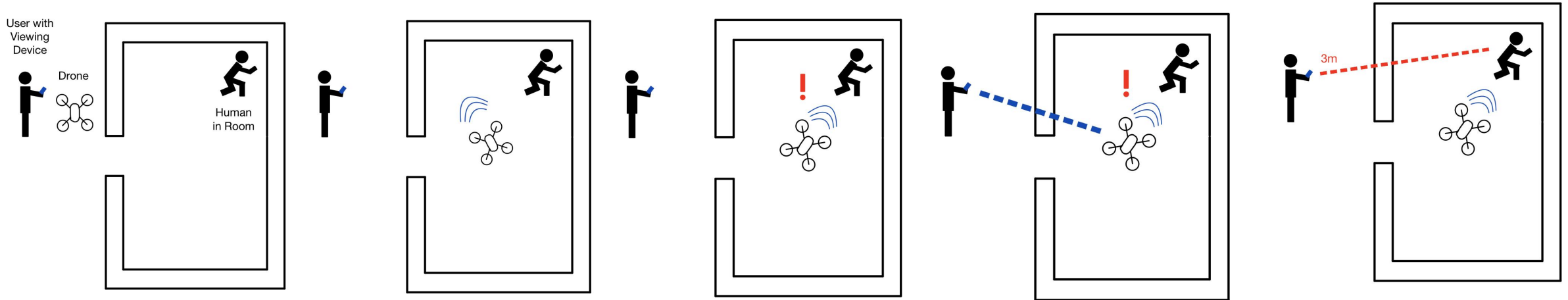
Mark Viz

Jacob Witek

Contents

- Introduction
- Objective
- Design
 - Block Diagram
 - Schematic & Printed Circuit Board (PCB)
 - Sub-System Overview
 - Presence Sensing
 - Power & Propulsion
 - Flight Control & AR Interface
- Mechanical Design
- Successes & Challenges
- Conclusions & Future Work

A Semi-Autonomous Quadcopter



(1) Reduces Risk of Room Entry in Emergencies

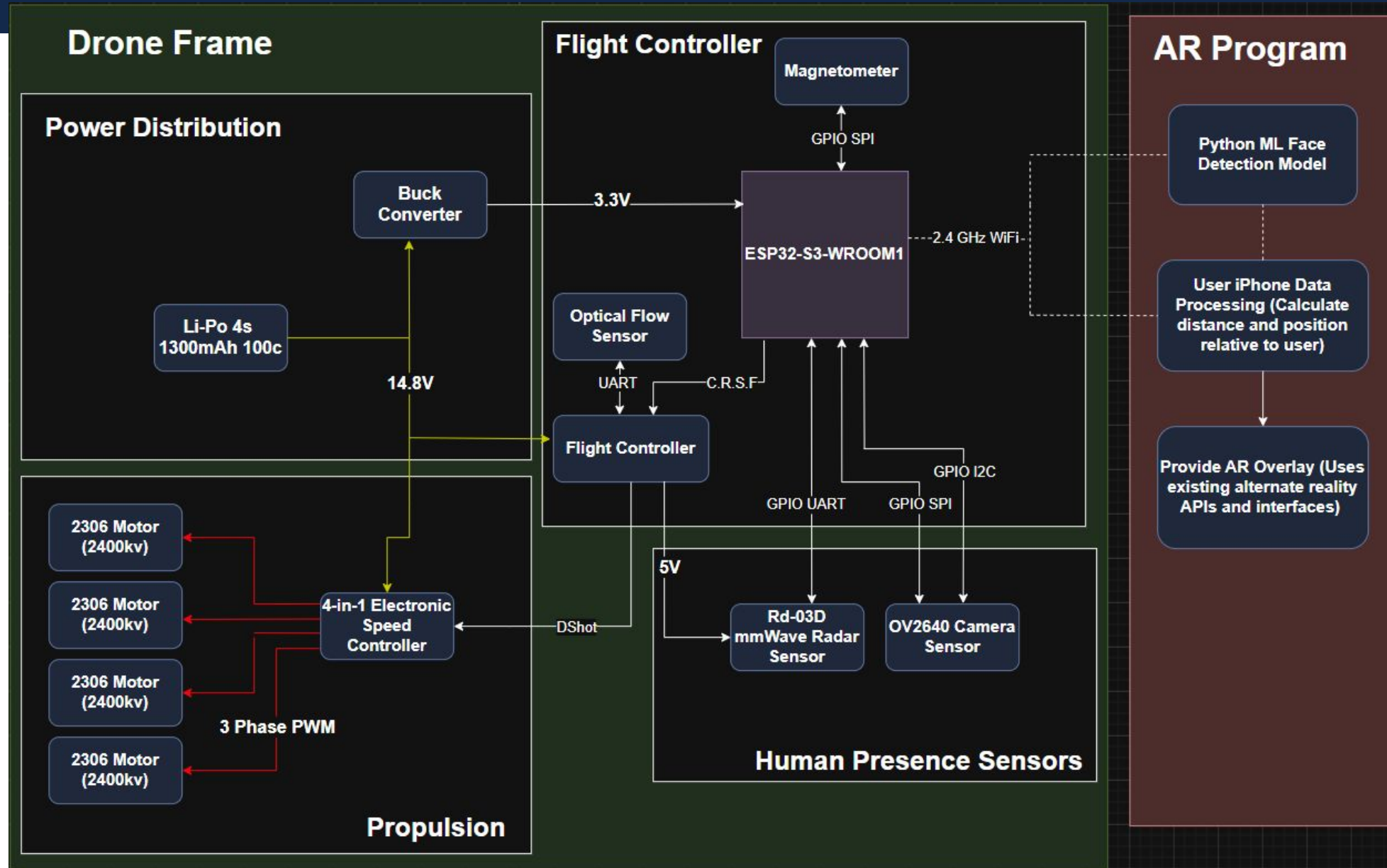
- Manual entry carries a risk towards emergency responders
 - Unstable flooring or ceiling
 - Armed assailants

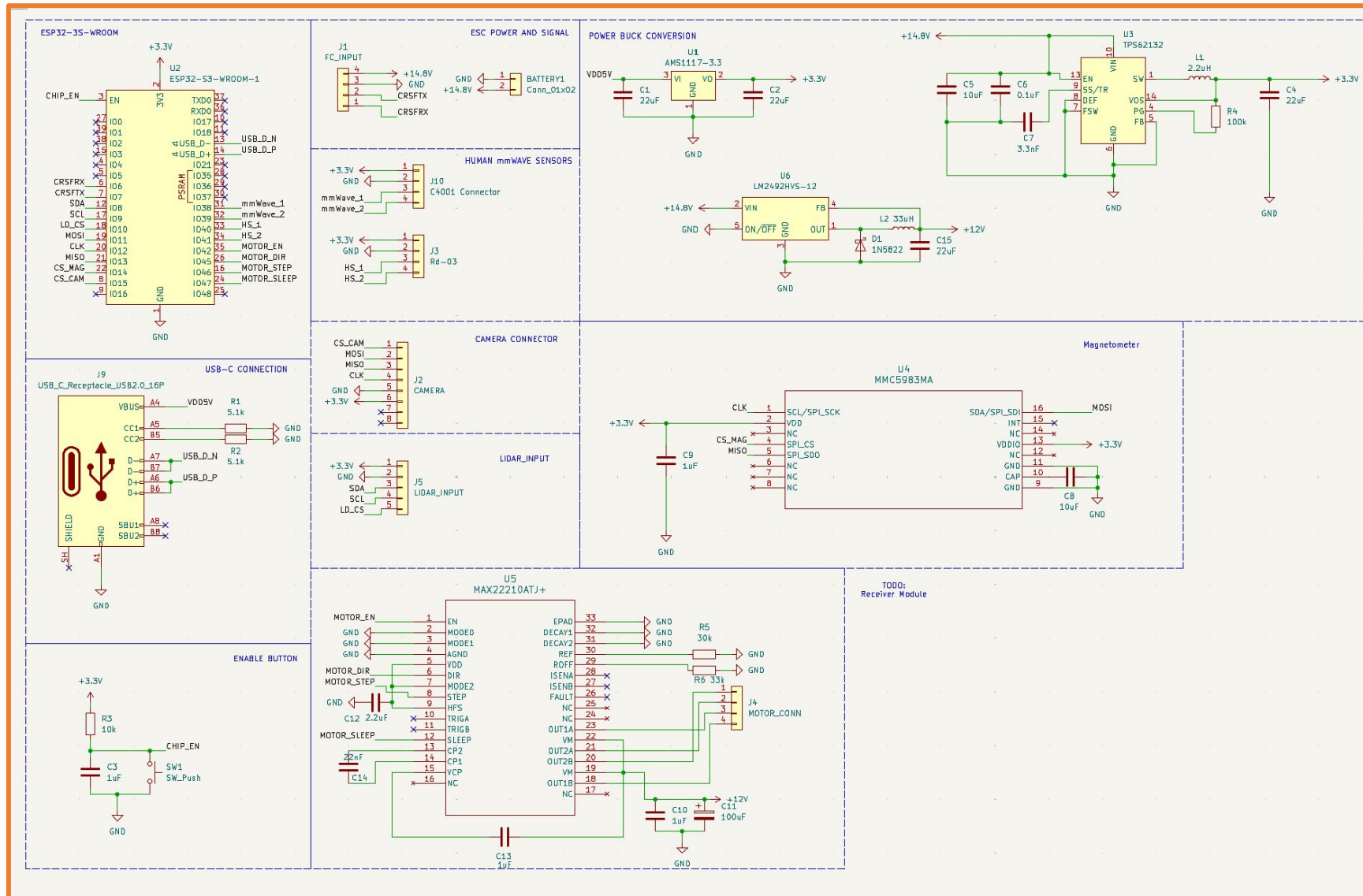
(2) Frees up the User

- Current implementations require the user to control a drone's movement and visually confirm the situation
 - The B.I.R.D only requires initial drone placement and a single button press on the iPhone App

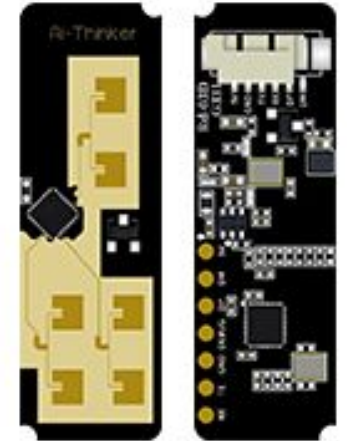


Block Diagram



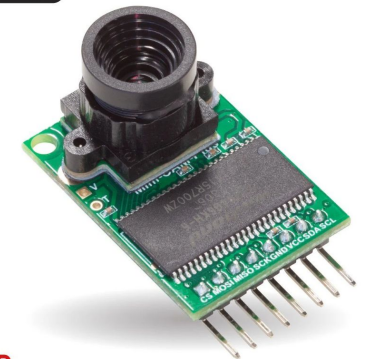


Requirements	Verification
Relay number of humans and local angle of each individual relative to each 60° increment at a maximum distance of 8 meters.	Transmit 320x240 image every 3 seconds to host server. Observe correct 'human' classification and angle offset from central image point in high light environment via an ML model
Constantly relay local angle and distance information of stationary humans at a maximum distance of 8 meters.	Observe human presence with $\pm 5^\circ$ angle offset from centerpoint and $\pm 0.5\text{m}$ distance measurement.



ArduCam
SKU: B0067

60°(H)



2 MP

ArduCam OV2640 Camera & ML Model

- Python script runs retrieved image through Ultralytics' YOLO11 model and only plots bounding boxes on humans with confidence levels over 0.25
- 60° FOV per image at 320 pixels horizontal resolution
→ 0.1875° per pixel
- $\text{pixel_offset} = \text{bbox_center_x} - 160$
- $\text{offset_angle} = \text{pixel_offset} * 0.1875^\circ$

$$\text{offset_angle} = (240 \text{ pxl} - 160 \text{ pxl}) * 0.1875^\circ = 15^\circ$$

Rd-03D 24 GHz FMCW Sensor

- Contains S5KM312CL FMCW transceiver SoC, three inset-edge fed patch antennas (2 receiver, 1 transmit), and SoC to transform output FFT into UART

Datasheet Parameters

Experimental Parameters

Azimuth Angle: $\pm 60^\circ$

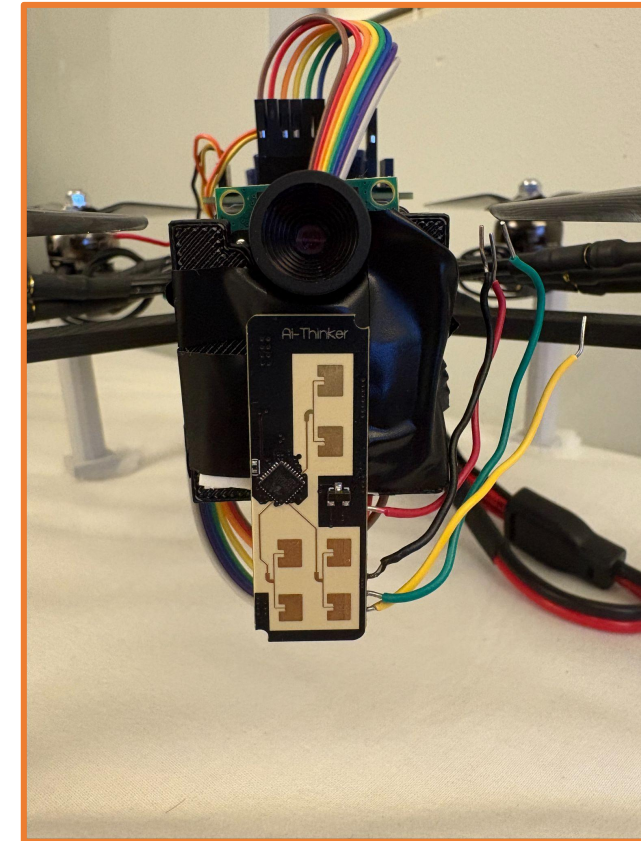
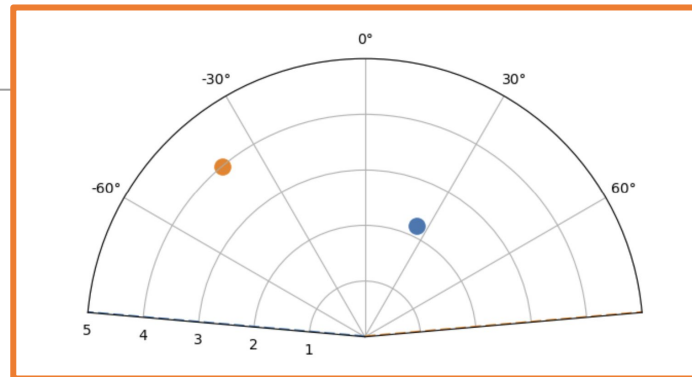
$\pm 70^\circ$ without F.F.

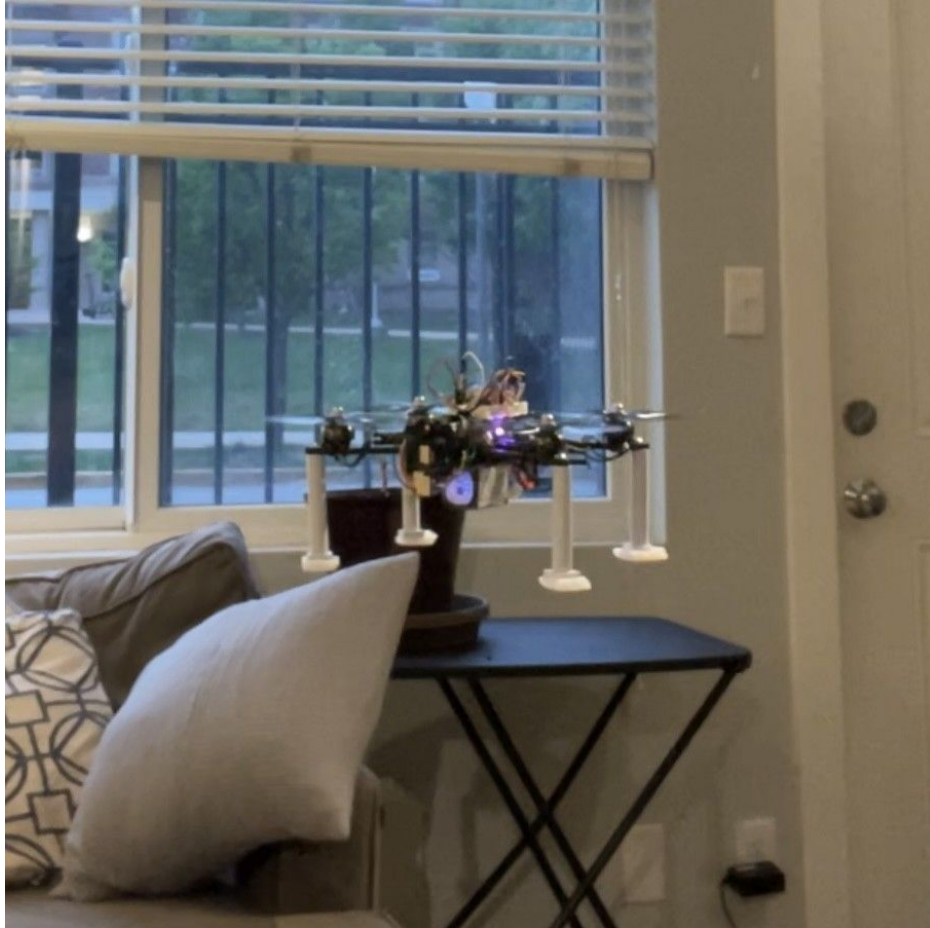
Ranging Accuracy: 0.15 m

~ 0.4 m

Max Sensing Range: 8 m

~ 5 m





Motor Considerations

- Four EMAX ECO II 2306 2400kV BDC motors at ~30 gram each
- One motor at 25% thrust (rough hover baseline): ~165 g thrust @ 4.5 g/W efficiency and 2.4 A current
- Total thrust range from idle to 100%: 180 g (3.2 A) to 5400 g (140 A) (actually lower due to current limit)

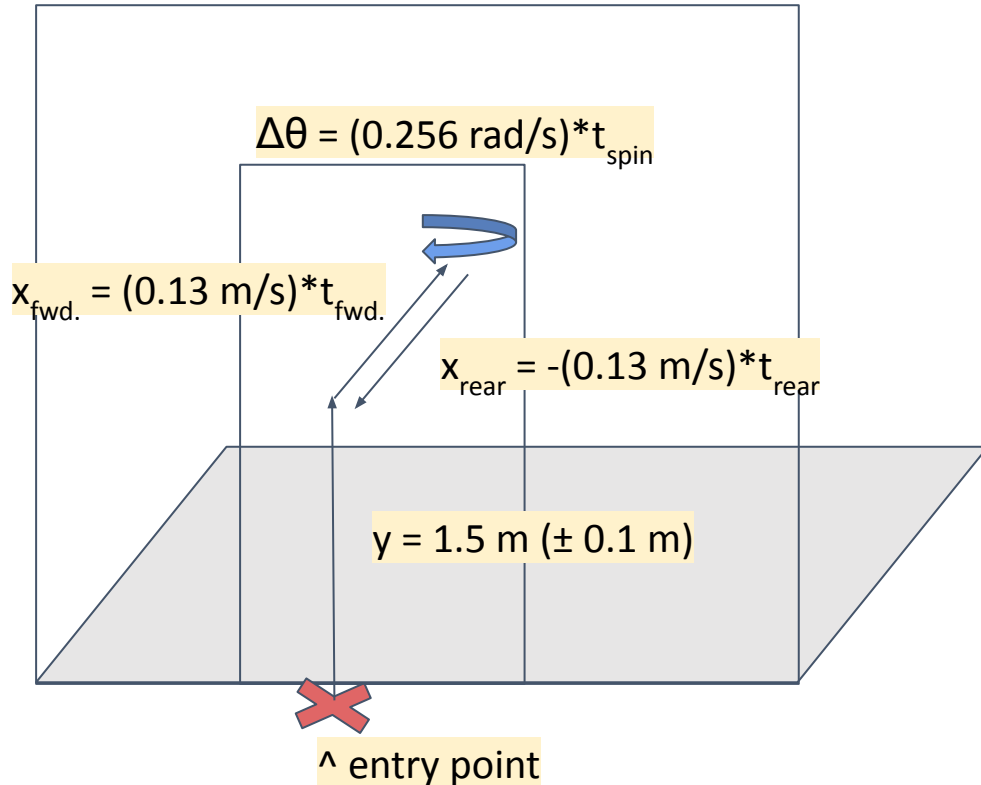
Battery Considerations

- OVONIC 4S 1300 mAh LiPo (100C => 130 A limit)
- 185 g

Flight Weight and TWR

- 135 g frame, account for ~160 g in PCB/sensors
- Total weight: 600 g
- Max TWR: 9.0 (allows for heavier future payload)

Movement and Positioning



- LiDAR range sensor ensures stable 1.5 m hover height ($\pm 10 \text{ cm}$ accuracy*)
- 360° yaw spin achieved with constant angular velocity input for timed duration ($\pm 1^\circ$ accuracy*)
- Forward/rear translation movements achieved using a constant linear velocity ($\sim 0.13 \text{ m/s}$) for timed duration ($\pm 0.5 \text{ m}$ accuracy*)
- *Accuracies verified with both measurements and visual observations

Requirements

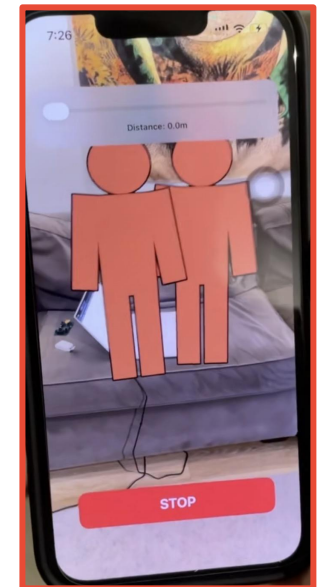
Safely and efficiently take off, move forward a set distance, survey the room, and return to take off position.

Relay information collected from previously discussed presence sensing and display it in a AR format that is easily understandable and provides relevant position information.

Verification

Run flight tests at different distances and voltage levels to ensure drone consistently and accurately takes off and moves set distances. Once test is complete verify drone has completed its flight path accurately using measurement devices.

Run presence detection tests with multiple people in different configurations and ensure that the app overlay is reflective of the people's positions.



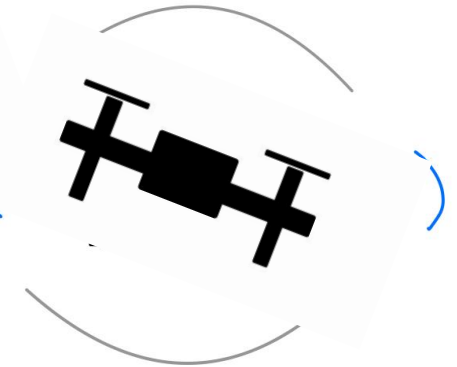
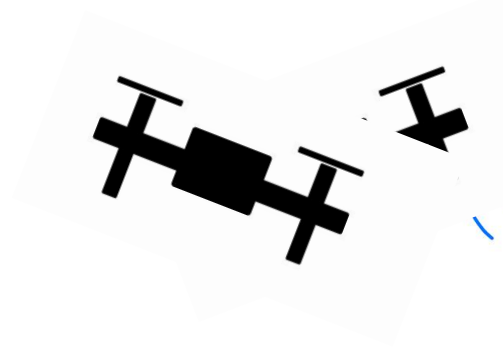
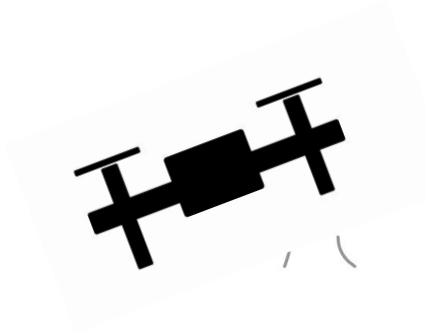
Flight Control System

- Drone is made up of flight “states” which it rotates through.
- Each state has a different goal which it must accomplish before moving onto the next.
- User input and sensor information control the states through different communication protocols.



Rolling Flight:

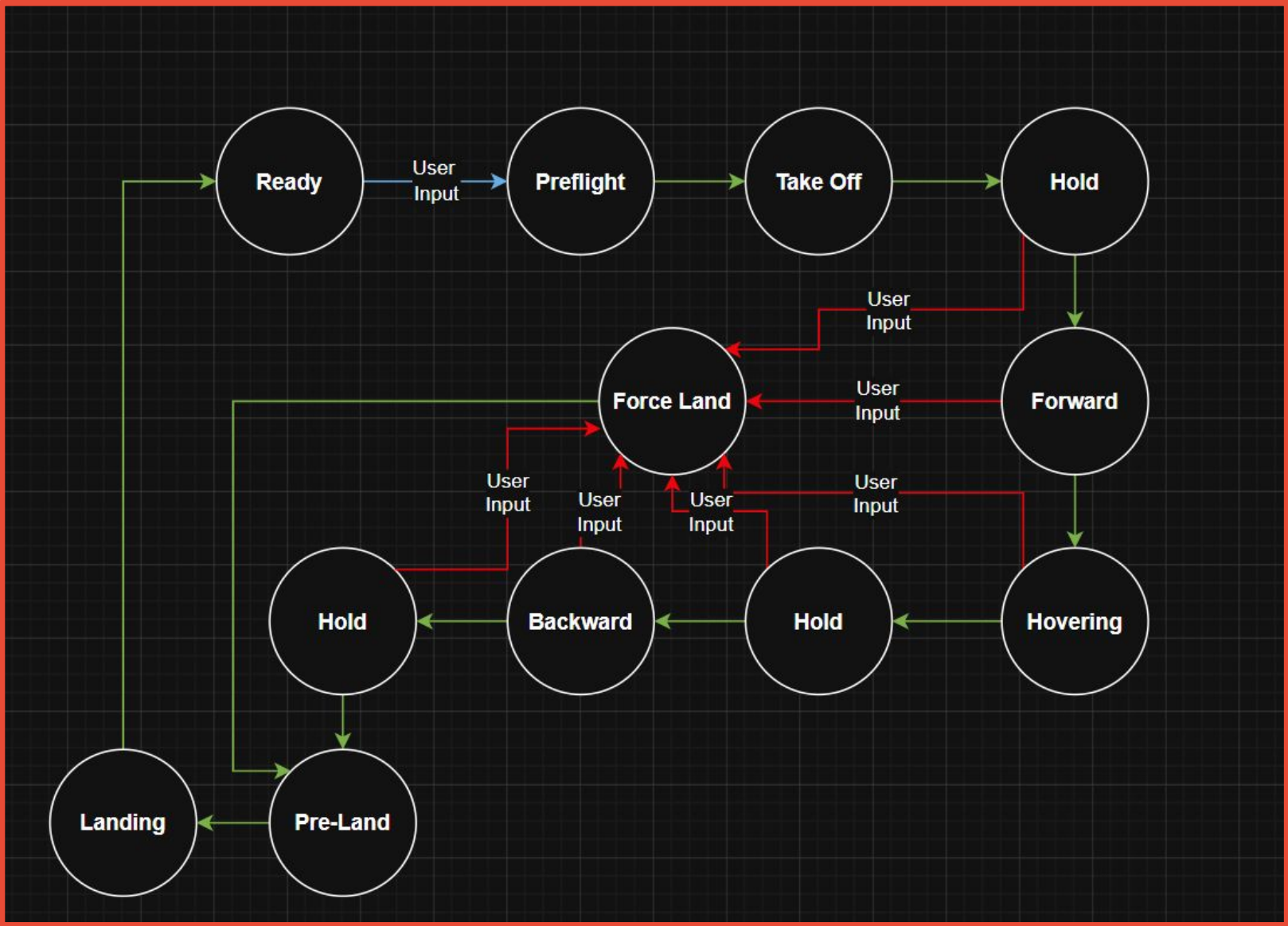
Drone stabilizing.
Spinning the propellers and
adjusting its speed for
a set time.



Drone Status:

Armed

Position Locked

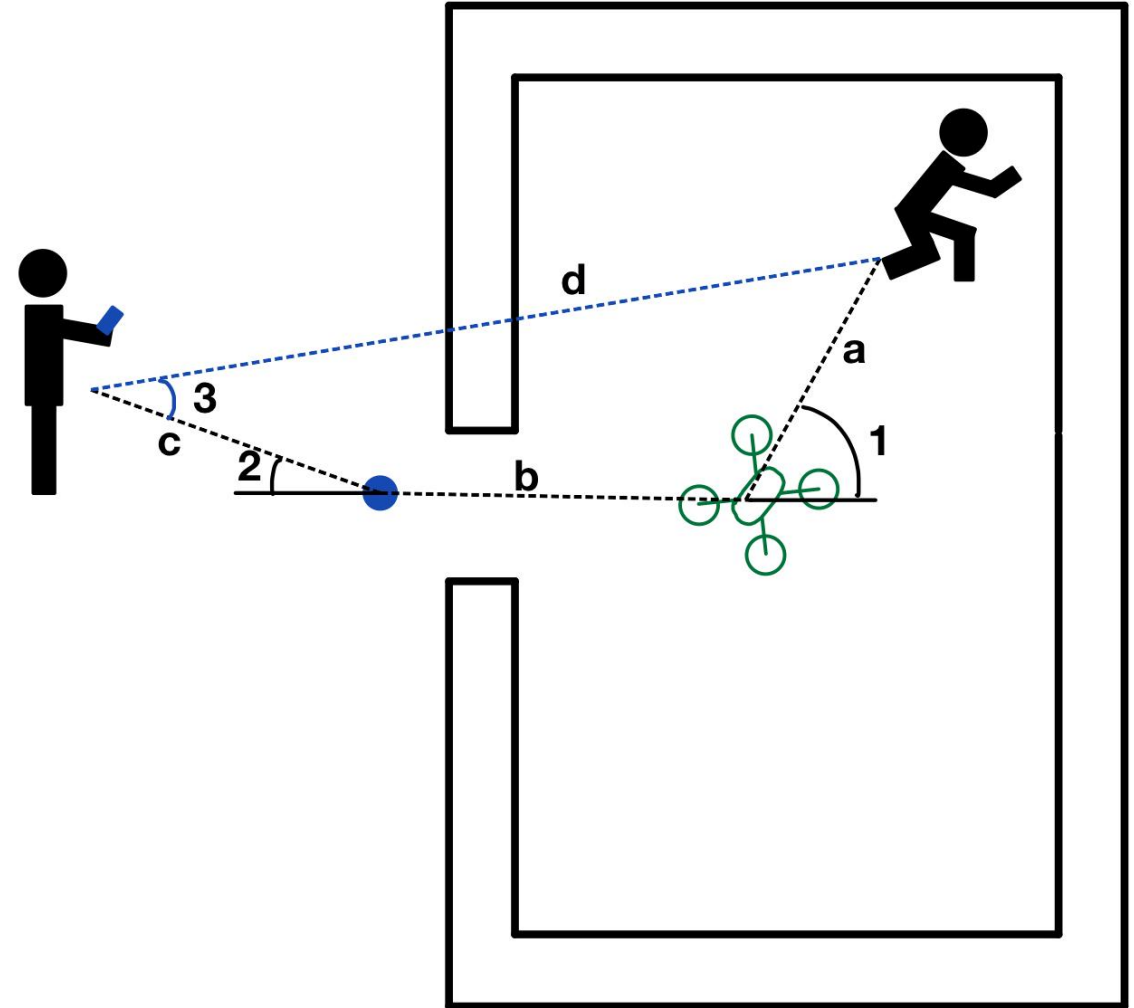


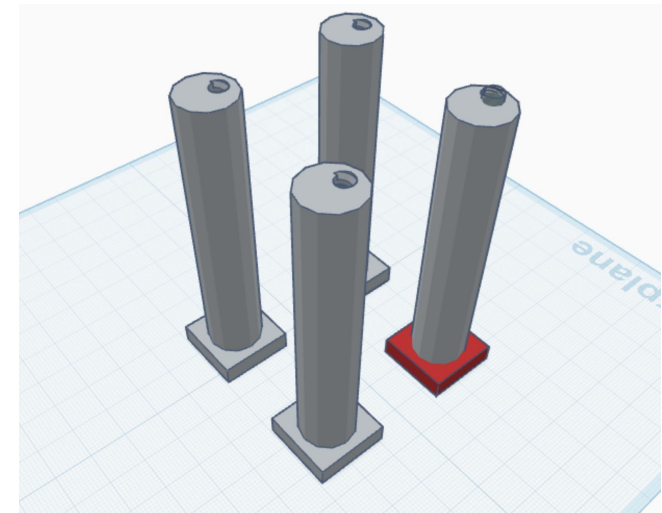
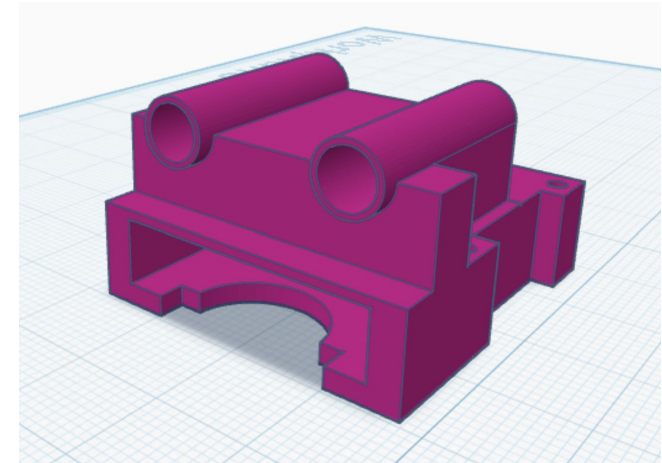
AR Interface

- Uses Apple's ARKit to form an XY grid of the local area.
- On startup the app sets an "anchor point" which it bases all positioning data from.
- Once a presence is detected the app receives angle, and distance data.

Equations:

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





- Successfully completed all three *High Level Requirements*
 - Lifting, hovering, forward movement, 360° rotation, reverse movement, landing
 - Correctly identifies moving and stationary humans in cluttered rooms
 - Displays accurate measurements on an iPhone app
- Use well known and well documented sensors, do not rely on them without full testing
- Leave free GPIO pins from MCU for unexpected future use cases on PCB
- Change from dead-reckoning to a LIDAR S.L.A.M approach for local positioning

Thank You!

<https://youtu.be/ANaeb1gEpyA>

Questions?