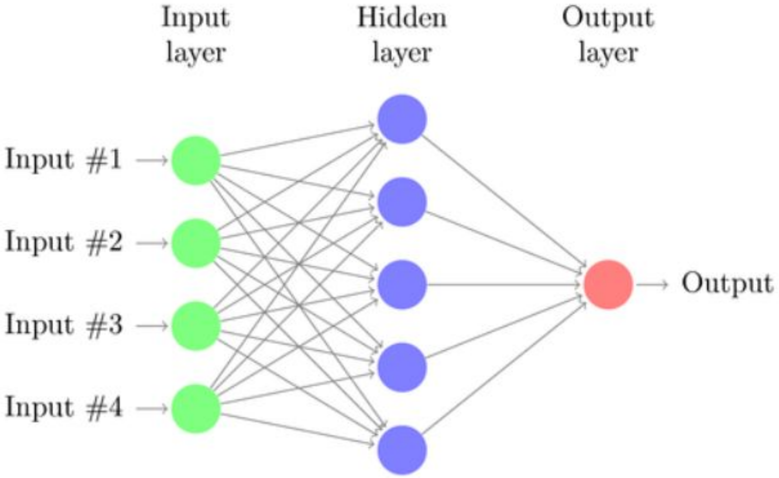
Object Detection Drone

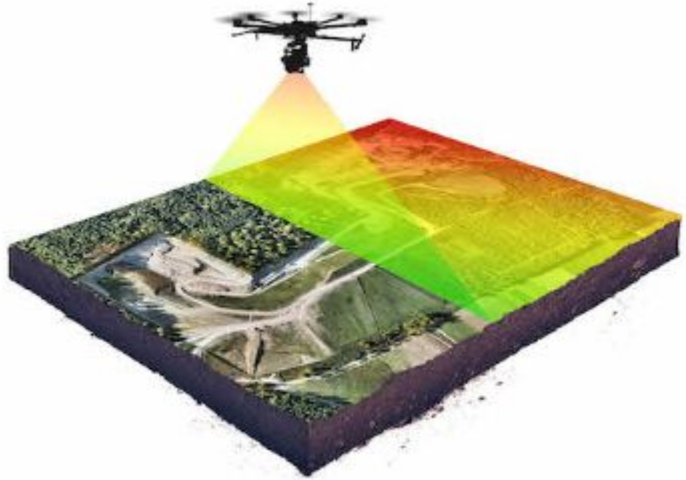
Final Presentation - Group Three

Sponsor: UCF ECE Department

UCF Senior Design
Spring 2023

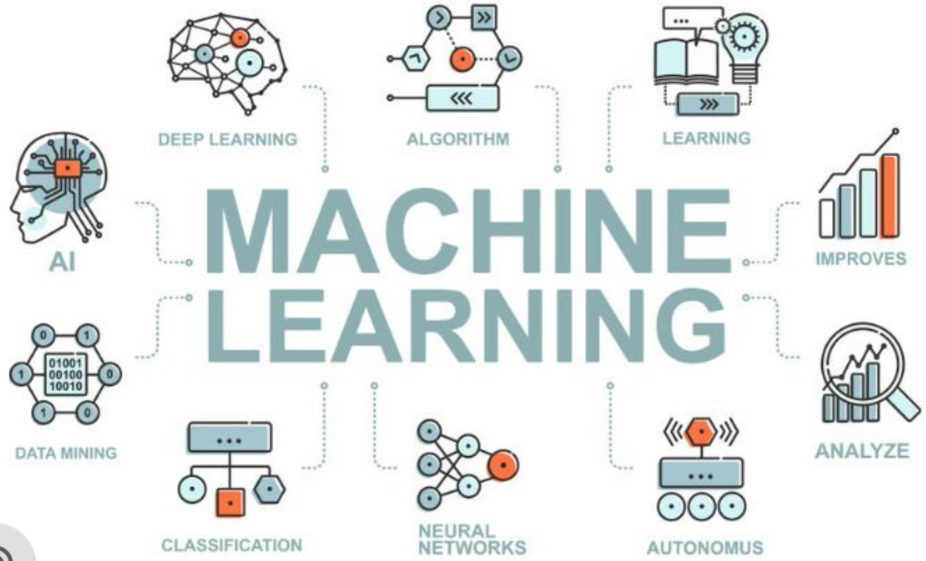
Introduction





Jazmine Roman (EE)

Motivation

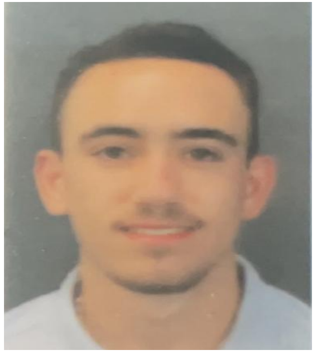
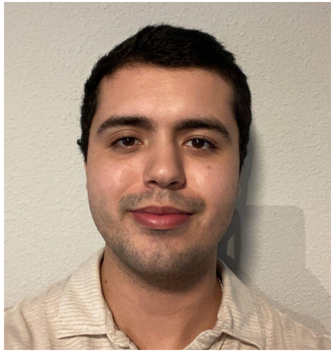



Jazmine Roman (EE)

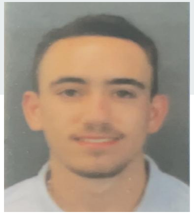
Purpose

- Develop a drone that has object detection and distance detection.
- The drone has full flight capability.
- The process included schematics, construction, simulation, calibration, and flight testing.
- Allowed essential contribution from all three disciplines within the group

Meet the Team

Derek Murdza *Computer Engineering*	Cannen Carpenter *Computer Engineering*	Jazmine Roman *Electrical Engineering*	Kevin Nilsen *Photonic Science and Engineering*
1. Flight / Navigation 2. Calibrations 3. RC Tuning	1. Machine Learning 2. Embedded Systems 3. Calibrations	1. Drone Build 2. Part Management 3. Calibrations	1. PCB Design 2. LiDAR Design 3. 3D Printing



Derek Murdza (CpE)

Sponsor Information

- This project was sponsored by the **University of Central Florida Department of Electrical and Computer Engineering**
- Received funding for drone components
- Offered areas for flight testing

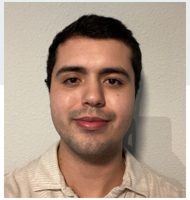

Jazmine Roman (EE)

Goals and Objectives

1. The drone must satisfy all pre-arm and safety checks
2. The drone must be able to takeoff, hover, maneuver, and land
3. The drone must be able to classify objects
4. The drone must be able to detect the distance from an object

Stretch Goal:

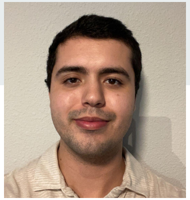
1. The drone must be able to fly autonomously



Cannen Carpenter (CpE)

Requirement Specifications

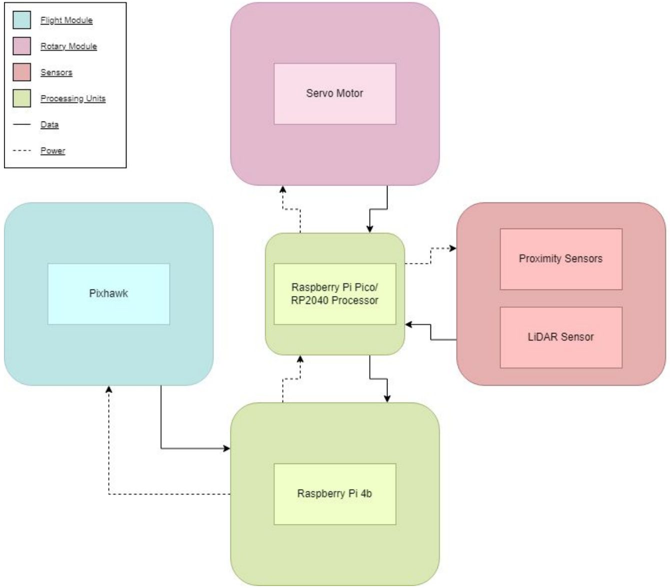
1	The drone should be able to fly in all three axes (X/Y/Z)
2	The system should be able to hover
3	The sensors must measure distance of target objects up to at least 1 meter with a 0.1 m accuracy
4	The machine learning model must classify objects within 5.00m and have a confidence above 85%
5	The drone must be calibrated to pass pre-arm checks for safety purposes
6	The drone should only be flown in open indoor areas



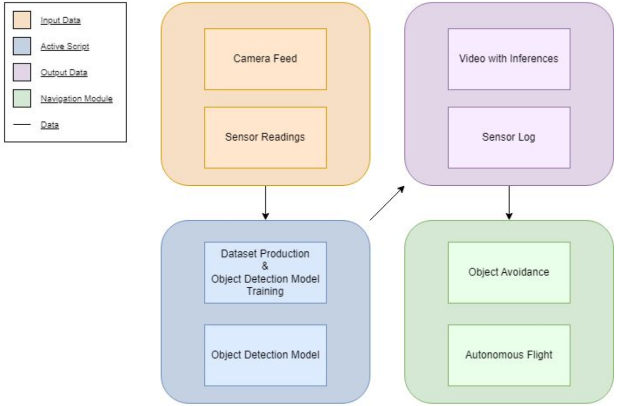
Cannen Carpenter (CpE)

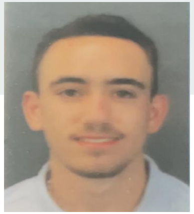
Block Diagrams

Hardware Diagram



Software Diagram




Derek Murdza (CpE)

Physical and Safety Constraints

- The prototype must weigh less than 10 lbs
- The prototype must utilize an even weight distribution
- The prototype must only operate in a safe area indoors as permitted by UCF
- The prototype must not exceed an overall power usage of 11.1V
- The prototype must utilize propeller fasteners to safely secure to motors


Jazmine Roman (EE)

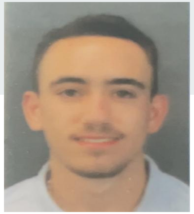
Drone Build and Design

Soldering, Planning, and Designing Component Layouts for Optimal Performance

Major Components Comparison

Jazmine Roman (EE)

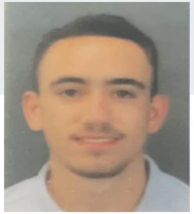
Motors	**Brushless**	-Very durable	- Complex and driven by a specialized circuit	**Brushed**	-2 leads only + & -	-Not durable
Flight Controller	**PixHawk 4**	-High performance -Customizable -Stable -User-friendly	-High cost -Complex	**Ardupilot**	-Open-source -Versatile -Cost-effective	-Constant software updates -Limited documentation
Companion Computer	**Single-Board Computer**	-High process power -Customizable -Cost-effective	-Power consumer -Large	**FPGA Board**	-High Performance -Low latency -Power efficient	-High cost -Difficult integration with flight controller systems
Microcontroller	**Raspberry Pi Pico**	-Low cost -High Performance -Versatile	-Limited memory	**ARM Cortex-M series**	-Low power consumption -Easy to program	-Limited memory and storage -Low processing power
Quadcopter Frame	**X Frame**		-Aerodynamic -Lightest frame	**H Frame**		-Simple to build -Heaviest frame



Derek Murdza (CpE)

ReadyToSky RS-2212 Motors

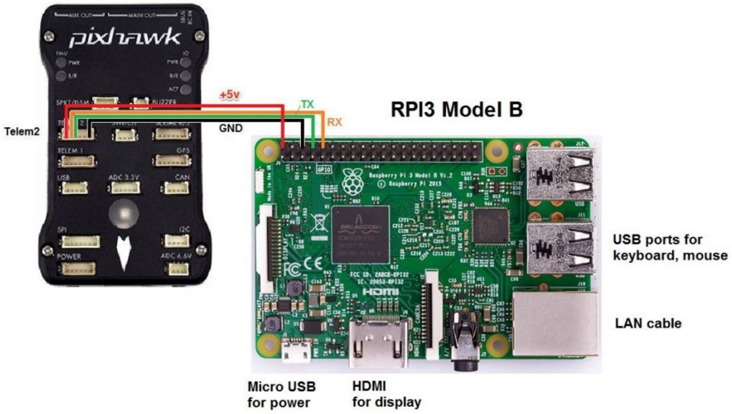
- Brushless for Noise Reduction
- 920kV Capacity
- Black Caps: Clockwise Rotation
- Silver Caps: Counter-Clockwise Rotation
- Connects directly to ESCs
    - Connections insulated with electrical tape to reduce any small electric magnetic field

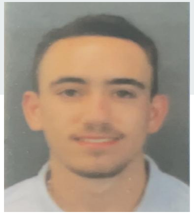


Derek Murdza (CpE)

PixHawk Module

- Responsible for sending data to Mission Planner and the Single Board Computer (RPi 4B)
- Crucial for flight control and calibration
- Allows user interfacing for adjustments to navigation
- Controls telemetry and GPS data

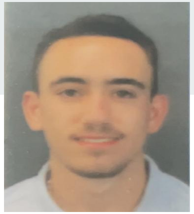




Derek Murdza (CpE)

LiPo Battery

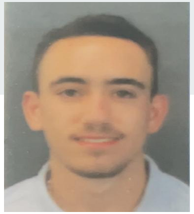
- Powers all drone components
- 11.1V Maximum Output
- 3000 mAh Charging Capacity



Derek Murdza (CpE)

ReadyToSky Electronic Speed Controllers

- Handles throttle control based on user input
- Powered by LiPo battery
- Requires calibration to spin motors
- Utilizes a crystal oscillator for highest accuracy
- 16 KHz motor frequency



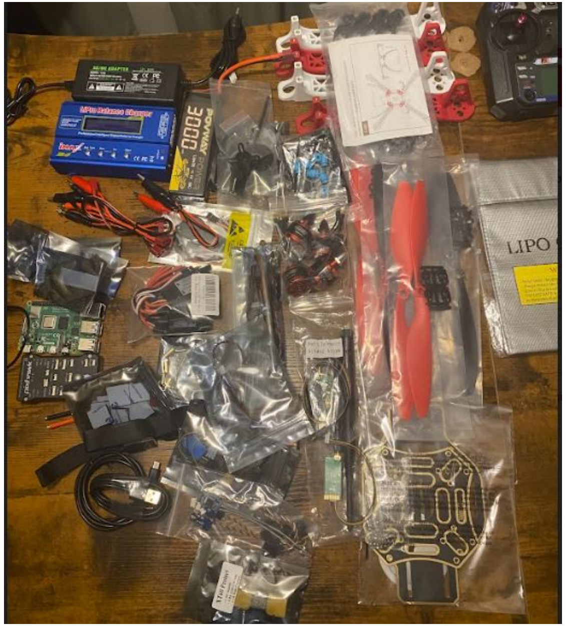
Derek Murdza (CpE)

Navigation Transmitter

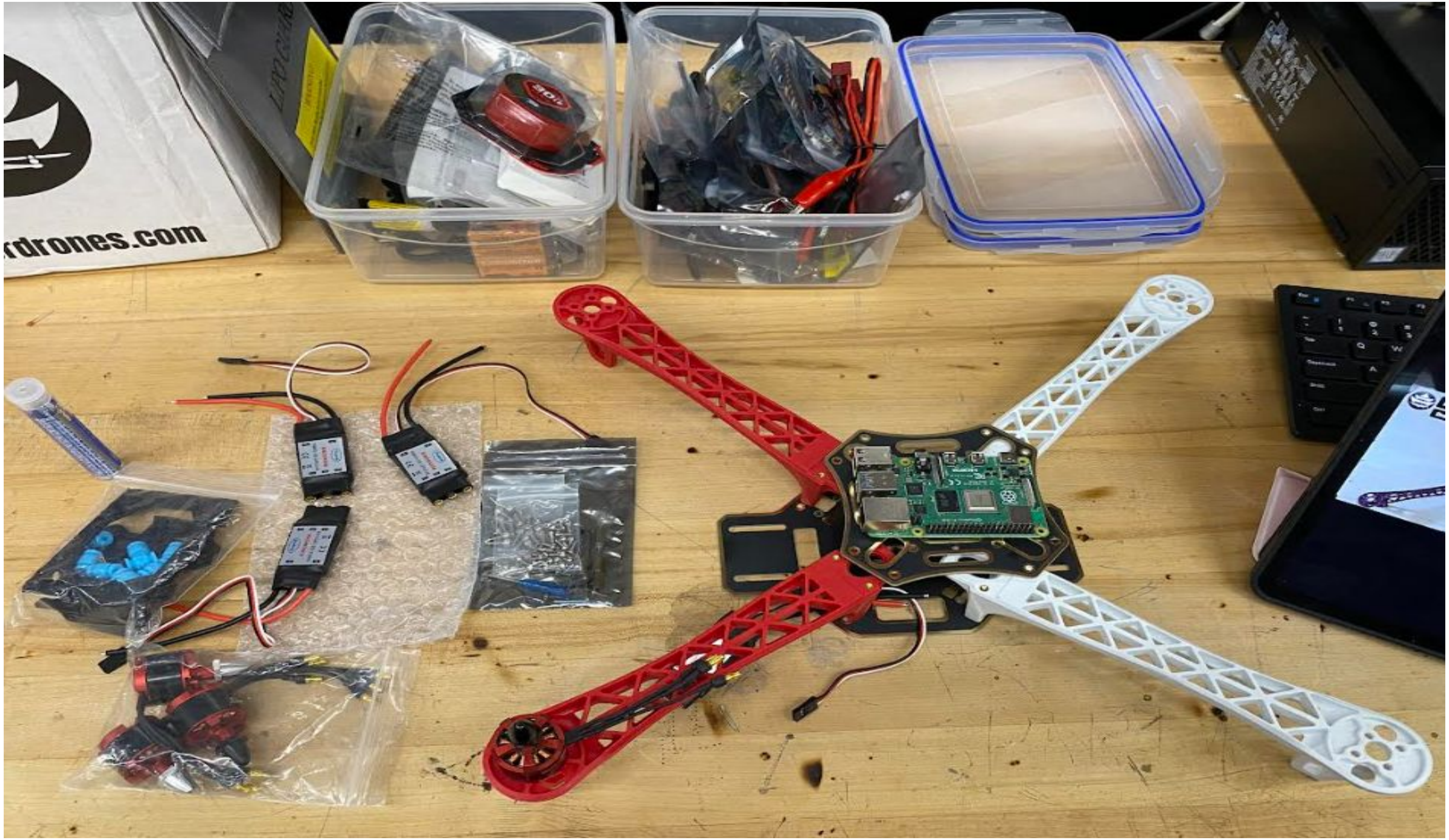
- The FlySky FS-i6 remote control is used for all flight tests
- The four main controls include:
  - Throttle (Altitude)
  - Pitch (Forward/Backward Flight)
  - Yaw (Rotation)
  - Roll (Side-to-Side Flight)

Jazmine Roman (EE)

Parts



Part Placement Planning

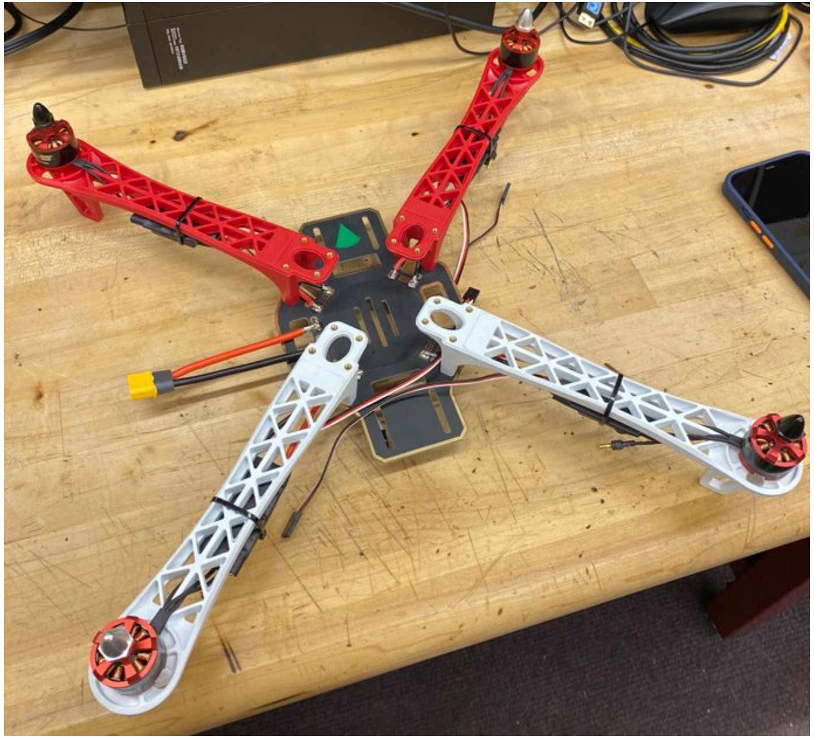


Jazmine Roman (EE)

Jazmine Roman (EE)

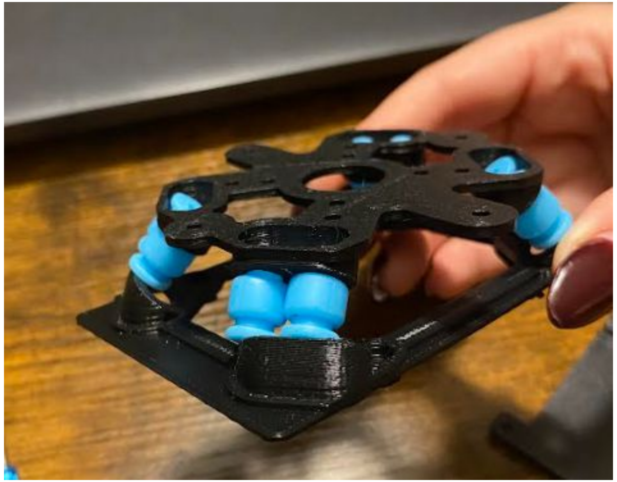
Power Distribution Board

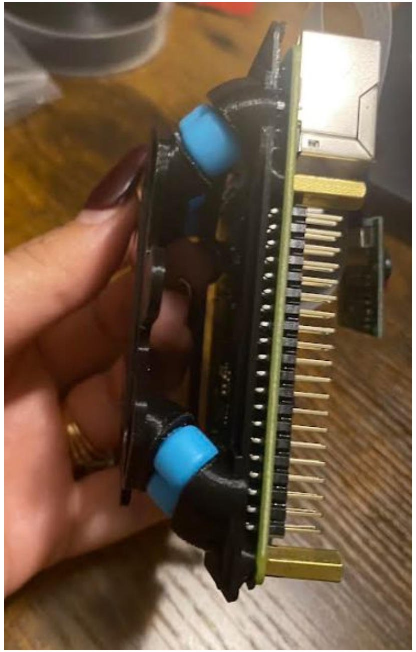
- Electric Speed Controllers (ESCs) & battery connector soldered onto PDB



Jazmine Roman (EE)

Vibration Dampener





Jazmine Roman (EE)

Top Plate

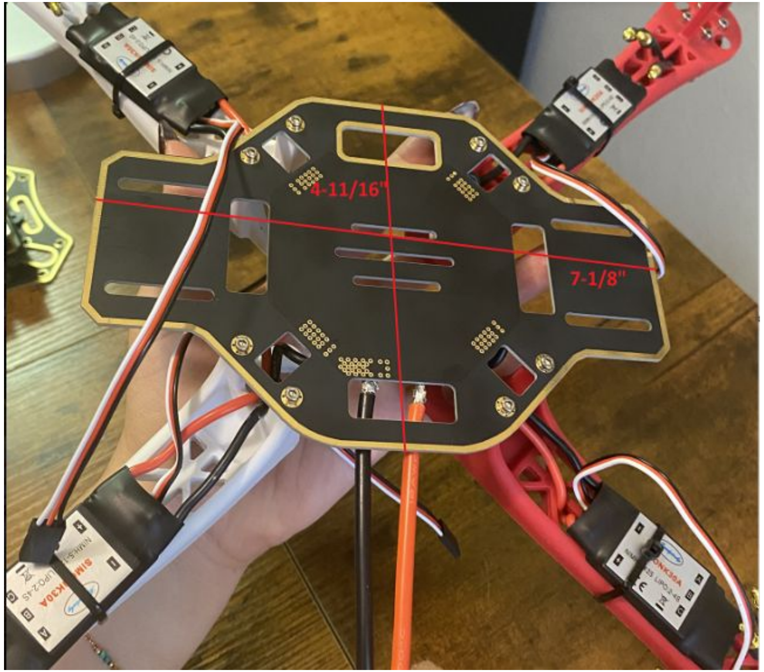
- Raspberry Pi with PiCam installed and mounted onto top plate

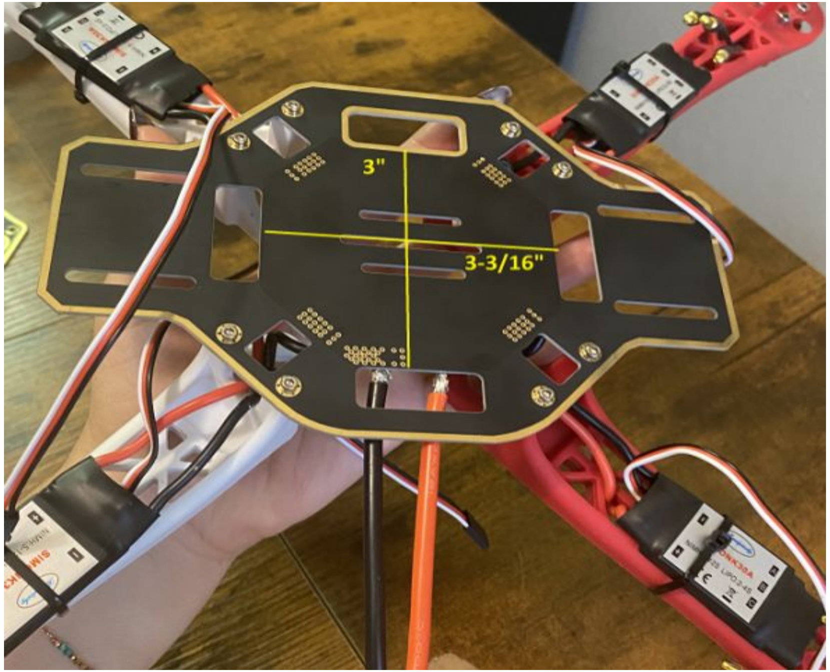
Jazmine Roman (EE)

Flight Controller Placement

Jazmine Roman (EE)

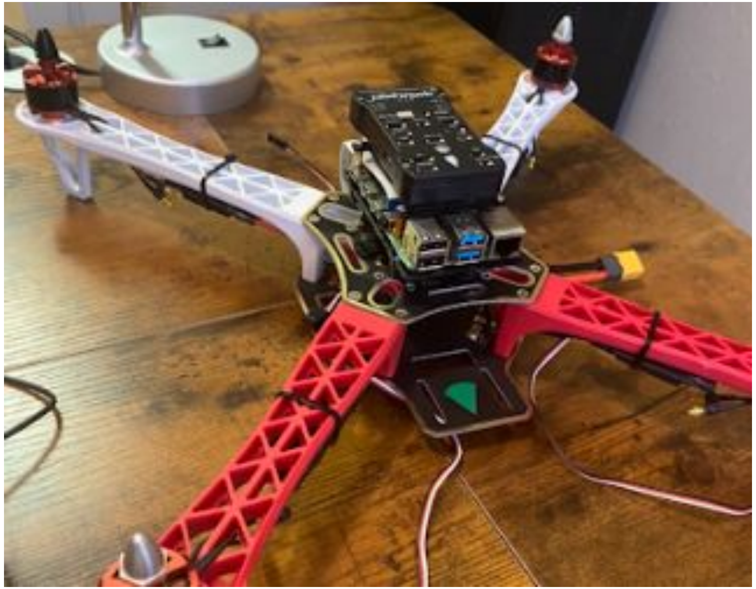
PCB Placement Planning

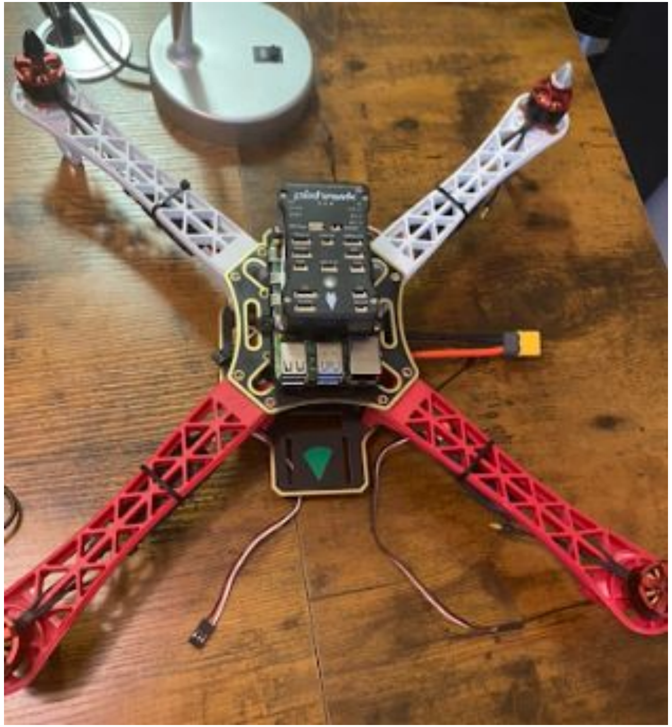




Jazmine Roman (EE)

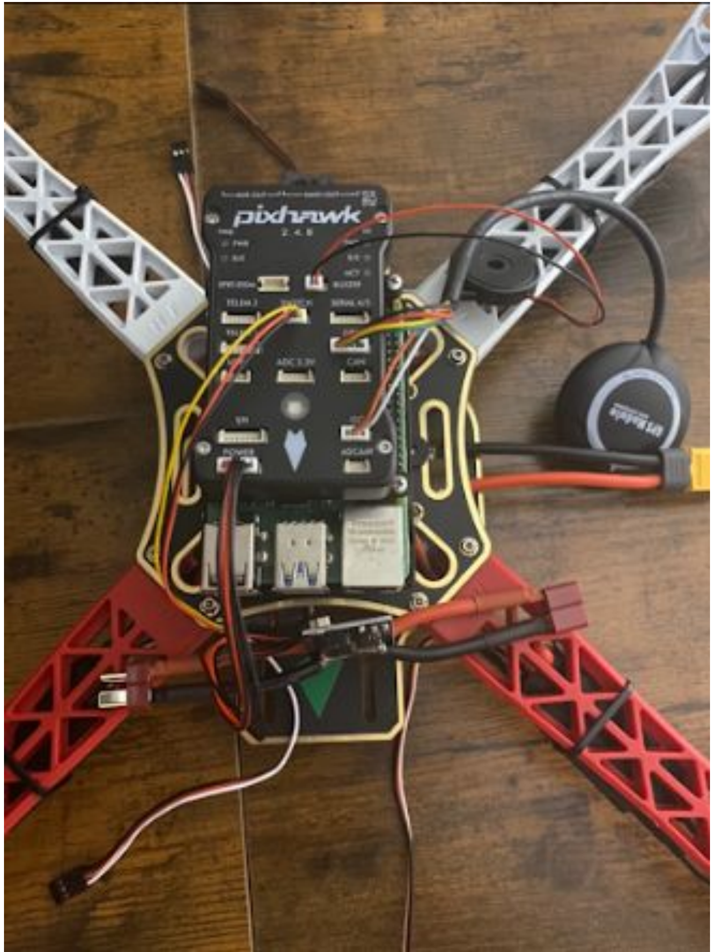
Top Plate Installation





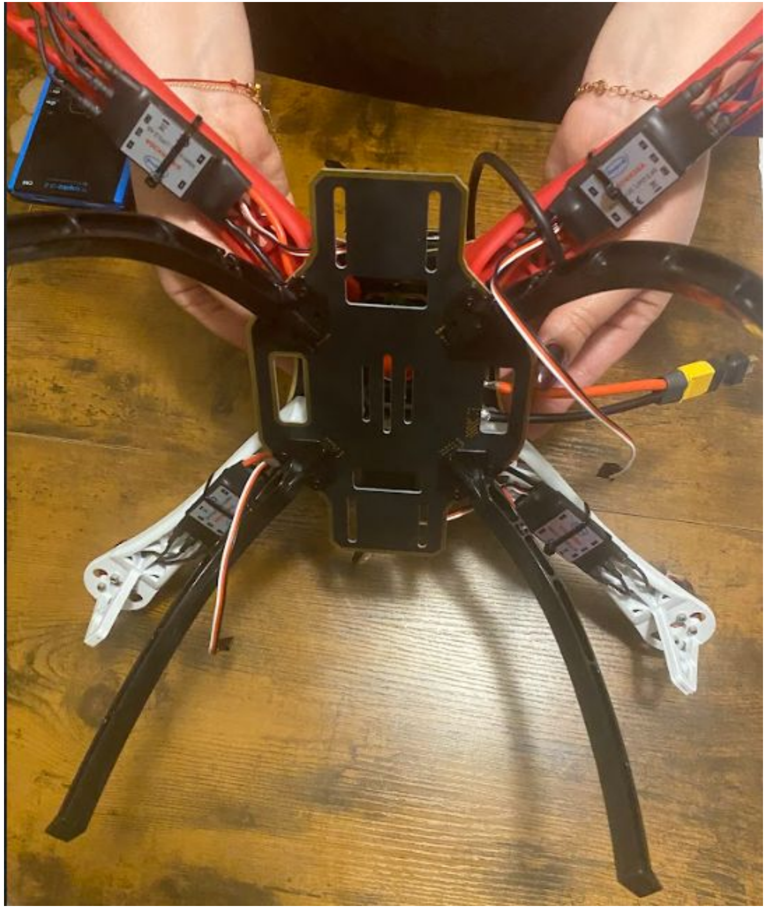
Wiring PixHawk

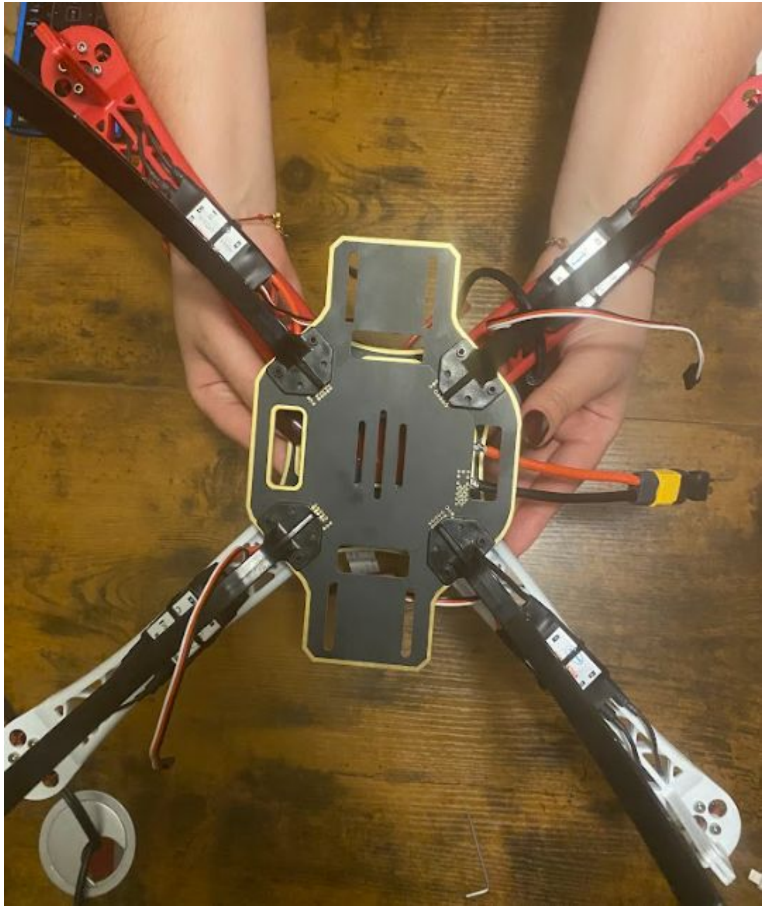
- Buzzer
- Safety switch
- GPS module with I2C connection
- Power module to battery




Jazmine Roman (EE)

High Landing Gear Legs





Jazmine Roman (EE)

Jazmine Roman (EE)

Mounting GPS and Camera

- GPS mounted away from PixHawk
- Camera mounted on top of raspberry pi USB ports

Bill of Material

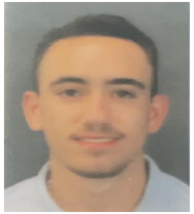
Part	Market Price
Frame legs	$25
High Landing Gear legs	$20
Propellers w/fastener	$23
Propeller guards	$20
Electric speed controller	$40
Power Distribution board	$52
UBEC power board module	$16

Part	Market Price
PiHawk flight controller	$189
Raspberry Pi4	$235
Picam	$10
4G TF Memory Card w/wires	$12
16G VHS-1 w/reader	$35
Brushless motors	$62
Vibration damper	$8

Part	Market Price
GPS module	$38
GPS mount	$9
Pinspice wires	$10
AC/DC 12V adapter battery charger	$12
LiPro Balance Charger	$32
Power 3000mAh Lipo3S 11.1V 50C	$40
XT60 Protect	$8

Part	Market Price
Screws and bolts	$7
Power module v1.0	$17
Radio Telemetry	$84
Zip ties	$5
Sticky fasteners	$19
USB to Type C	$8
USB to Micro Connector	$8
AA battery pack	$15

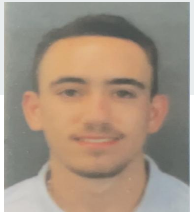
Part	Market Price
Velcro fasteners	$9
Fireproof case	$30
RC controller	$53
Pico microcontroller	$22
IR PCB board	TBD
LIDAR PCB board	TBD
Total without kit	$1,116
Total with kit	$849



Derek Murdza (CpE)

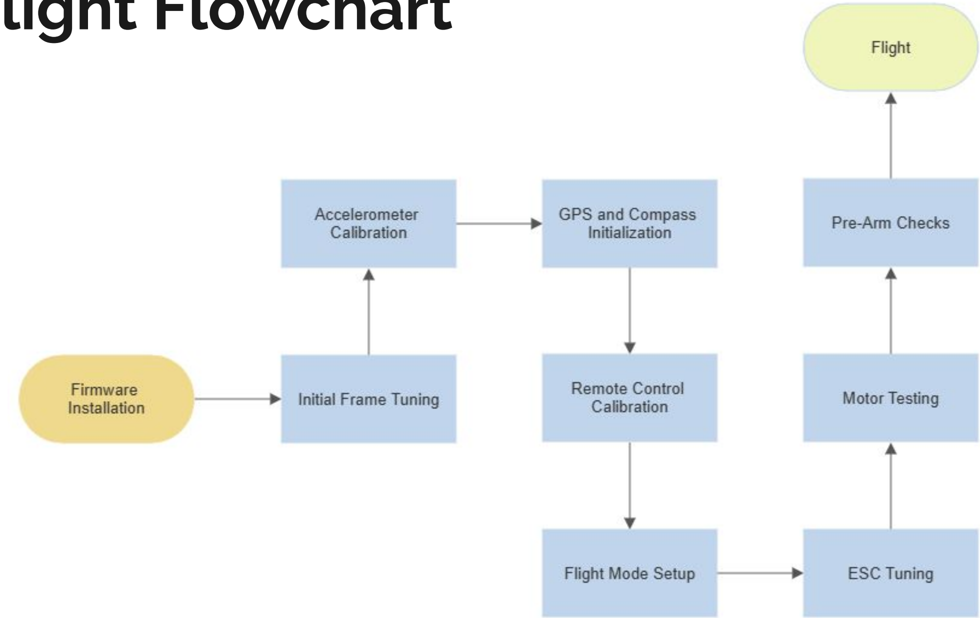
Navigation and Calibration

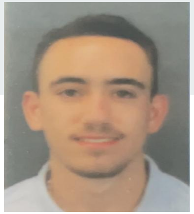
Utilizing Essential Firmware and Calibration to Achieve Successful Flight



Derek Murdza (CpE)

General Flight Flowchart

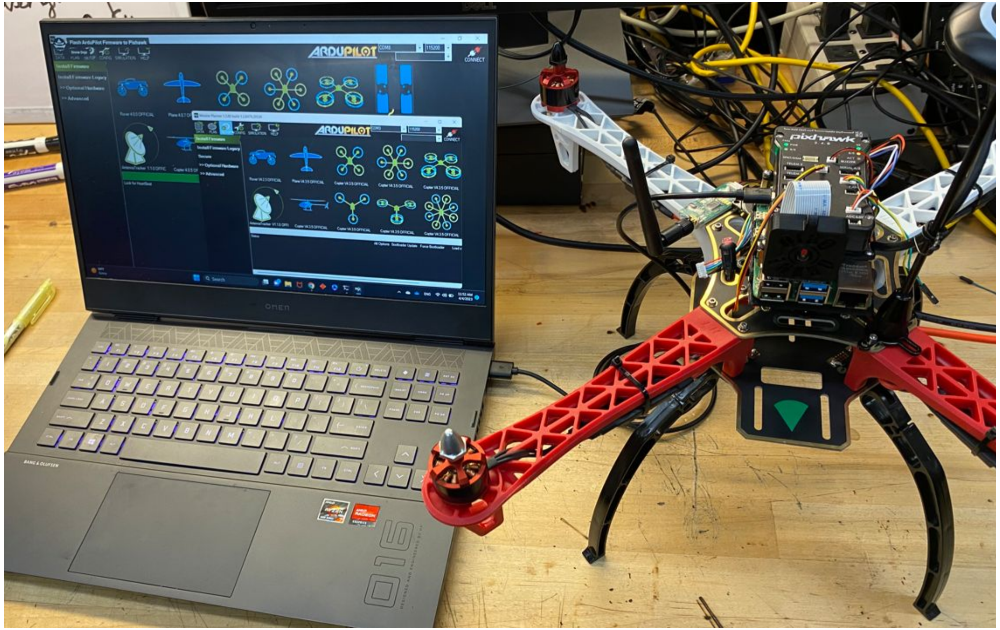


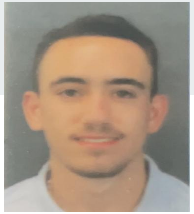


Derek Murdza (CpE)

Mission Planner

- Mission Planner is the flight controller used to monitor all flight data
- The primary functions include:
  - Drone Calibration
  - Pre-Arm Checks
  - Flight and GPS Data
  - Motor Function and Direction Testing

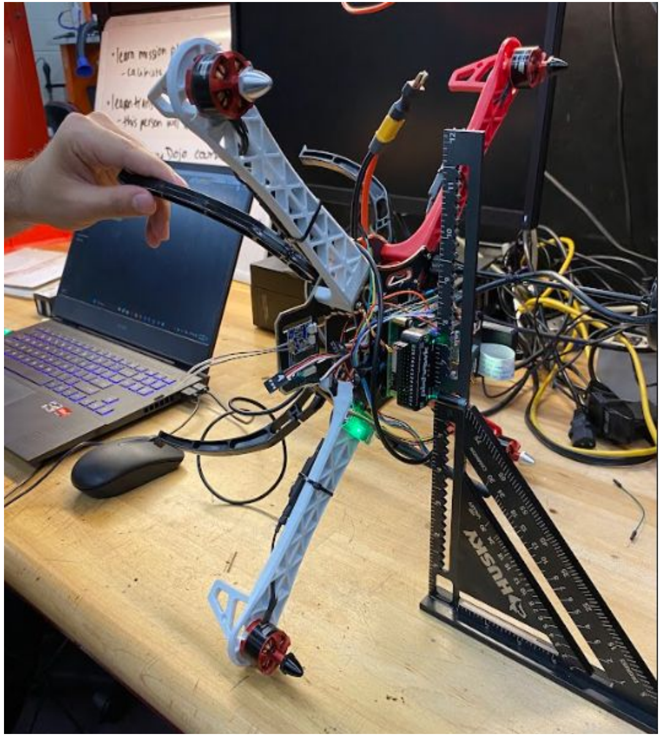




Derek Murdza (CpE)

Drone Calibrations

- Essential Calibrations include:
  1. Accelerometer
  2. GPS Fixation
  3. Compass
  4. Radior
  5. ESC
  6. Motor Test



GPS Calibration

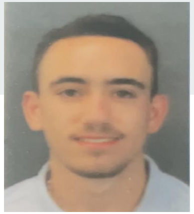
- Performed outside to ensure satellites connection
- Crucial for satisfying pre-arm checks

Jazmine Roman (EE)

Jazmine Roman (EE)

Ready for RC flight!

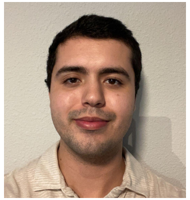
- Performed indoors for safety of others and the robot



Derek Murdza (CpE)

Potential for Autonomous Flight

- Determined as a stretch goal
- Python scripts were tested using the Gazebo flight simulator and PX4 Autopilot
    - Functions Tested:
        - Takeoff and Landing
        - Directional Maneuvering
- This prototype has great potential to be fully-autonomous with more training

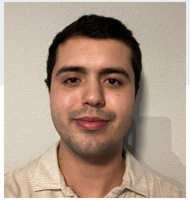


Cannen Carpenter (CpE)

Object Recognition

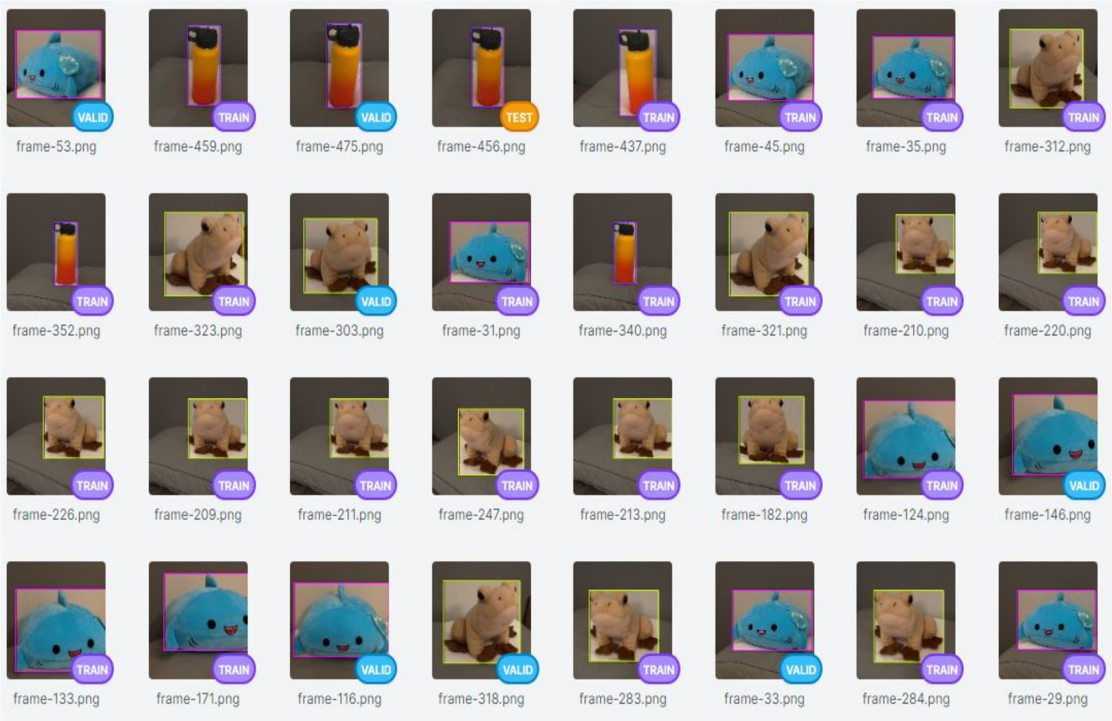
Implementing Machine Learning to Detect and Predict Objects Mid-Flight

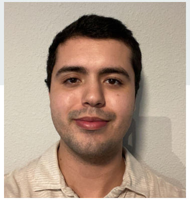
Custom Dataset Production



Cannen Carpenter (CpE)

- Developed dataset using Roboflow's platform
- Recorded video of objects, each frame was selected as an image for our dataset
- Dataset is comprised of 550+ manually annotated images
    - Augmented to produce 700+ more annotated images





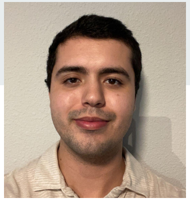
Cannen Carpenter (CpE)

Model Training

- Custom Dataset
  - 1200+ custom annotated images
- YOLOv8 model
  - Implementation is done with PyTorch, rather than Darknet
- Training Specs:
  - Image Size: 800x800
  - # of Epochs: 25
- Last few epochs and overall results shown in the figure on the right

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      23/25      7.2G     0.1643     0.1687     0.8115          8        800: 100% 64/64 [00:28<00:00,  2.24it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 5/5 [00:02<00:00,  1.76it/s]
                   all        150        159      0.992      0.996      0.993      0.979
                 Coqui        150         30      0.999          1      0.995      0.993
         Rey the Shark        150         75      0.998      0.987      0.991      0.974
          Water Bottle        150         54      0.979          1      0.992      0.971

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      24/25      7.2G     0.1498     0.1505     0.8052          9        800: 100% 64/64 [00:28<00:00,  2.28it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 5/5 [00:03<00:00,  1.33it/s]
                   all        150        159      0.991      0.996       0.99      0.973
                 Coqui        150         30          1          1      0.995      0.993
         Rey the Shark        150         75      0.996      0.987      0.991      0.979
          Water Bottle        150         54      0.975          1      0.985      0.947

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      25/25      7.2G     0.1353      0.139     0.7977          8        800: 100% 64/64 [00:28<00:00,  2.24it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 5/5 [00:06<00:00,  1.24s/it]
                   all        150        159      0.992      0.996      0.992      0.983
                 Coqui        150         30      0.999          1      0.995      0.993
         Rey the Shark        150         75      0.998      0.987      0.991      0.978
          Water Bottle        150         54      0.978          1      0.989      0.977

25 epochs completed in 0.308 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.20 🚀 Python-3.9.16 torch-2.0.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 5/5 [00:06<00:00,  1.22s/it]
                   all        150        159      0.992      0.996      0.992      0.983
                 Coqui        150         30      0.999          1      0.995      0.993
         Rey the Shark        150         75      0.998      0.987      0.991      0.978
          Water Bottle        150         54      0.978          1       0.99      0.977
Speed: 4.0ms pre-process, 7.0ms inference, 0.0ms loss, 4.3ms post-process per image
```

Model Deployment



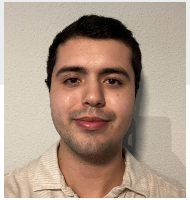
Cannen Carpenter (CpE)

- Roboflow Raspberry Pi Docker
  - Used to install the Roboflow inference server for deploying our model
- Roboflow inference server
  - Ability to download our trained model and make inferences
- Installed packages using MiniConda & PIP
  - Roboflow, OpenCV, and PyDrive
- Model performance standards:
  - Confidence in correct detection of at least 85%
  - Differentiate between objects consistently

PyTorch

CONDA®

python™

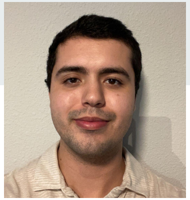


Cannen Carpenter (CpE)

Scripting

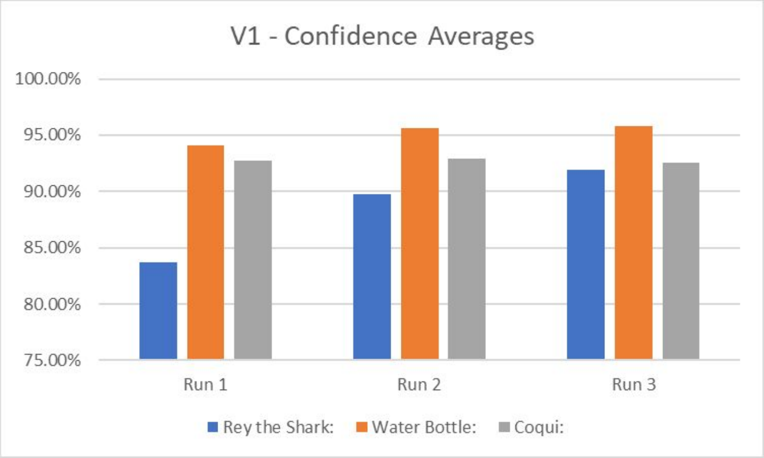
- Central processing unit for our drone is our Raspberry Pi 4B
  - Powered through our battery, which then transmits power to our PCB
- Flashed with the latest version of Ubuntu, 22.10, using the Raspberry Pi Imager
  - Done to utilize software packages that help deploy our model onto the Pi
- Scripts are created in Python, and activated through the Ubuntu terminal on startup
  - Can end recording session through wireless keyboard interrupt
- Our scripts hold a few purposes:
  - Video Recording
  - Inferencing
  - Receives Sensor Data
  - Uploads results to our Google Drive
- Created two versions, one without LiDAR integration, and one with LiDAR integration
  - Used to reduce computing power

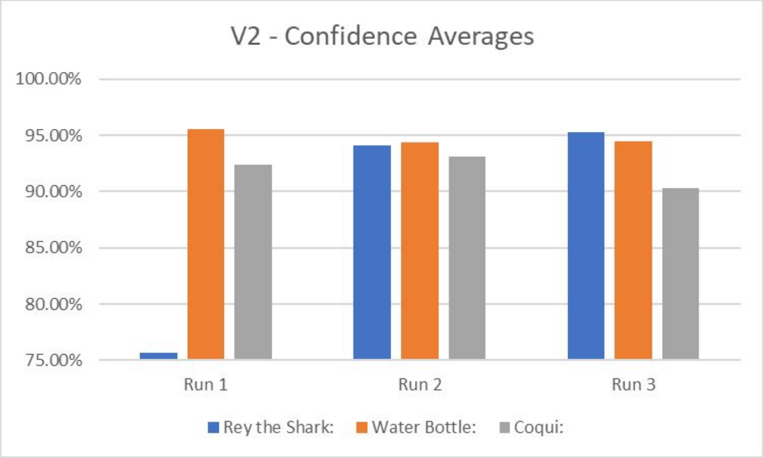
Object Detection Results



Cannen Carpenter (CpE)

- Our selection of objects were all somewhat difficult for different reasons
  - Reflectivity of the metal water bottle and shape
  - Size and shape inconsistency of the shark plushie
  - Shape of the frog plushie
- Water Bottle ended with the highest average confidence, with Rey giving the most trouble
  - Successful in managing a confidence average of over 85%






Kevin Nilsen (PSE)

Distance Detection

Implementing LiDAR to Detect Objects and Record Distance Mid-Flight

Kevin Nilsen (PSE)

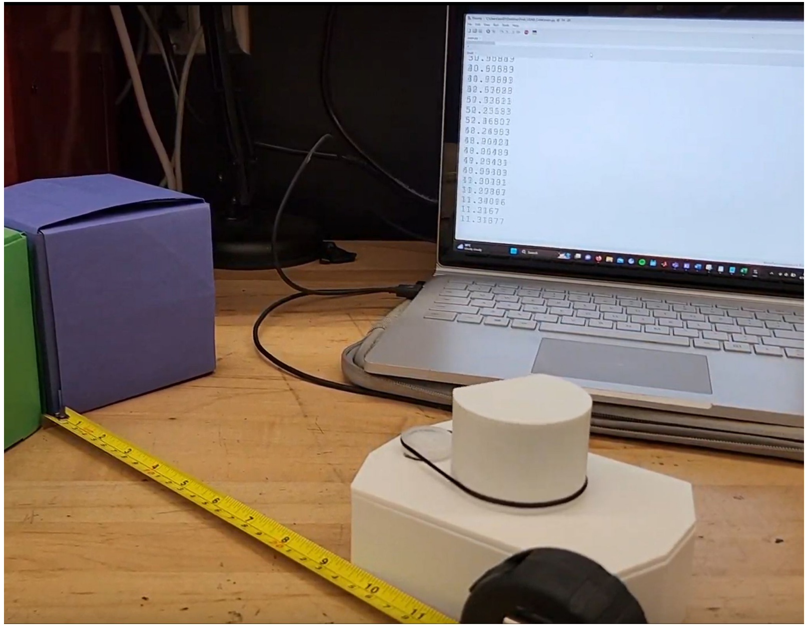
Sensor Comparison

Method	Pros	Cons	Comments
Proximity Sensors	• Cheap • Scalable	• Short range • Imprecise • Noise sensitive	Simple and easy to implement. Multiple sensors can be used
Distance Detection	• More accurate	• Short range • More complex	Likely provides little benefit over object detection
2D Light Detection and Ranging (LIDAR) Scanning	• Accurate • Long Range	• Inaccurate for very short distances • Complex driving circuitry	Worth exploring due to far more accurate environment data

Infrared Detection Methods

- LiDAR **Range is 1 meter** (3 ft), with an accuracy within 8 centimeters
- Wavelength at 940 nm for LiDAR and 980 nm for proximity sensors
- LiDAR struggles with objects less than 30 cm away
- Four sets of IR proximity sensors using LEDs are also situated around the drone

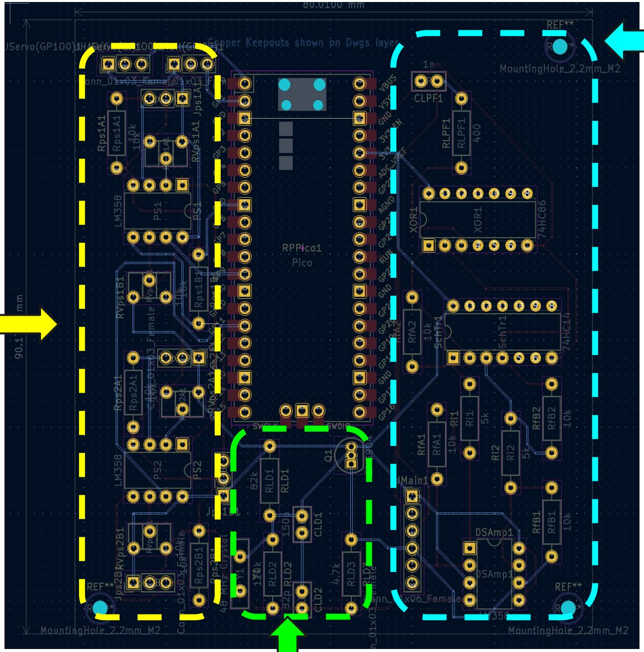
Kevin Nilsen (PSE)



PCB - Distance Sensing

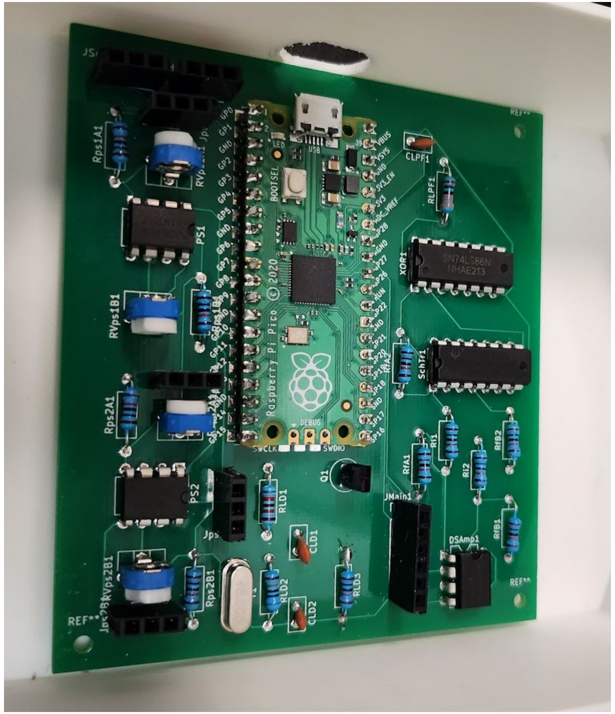
Kevin Nilsen (PSE)

Proximity sensor control



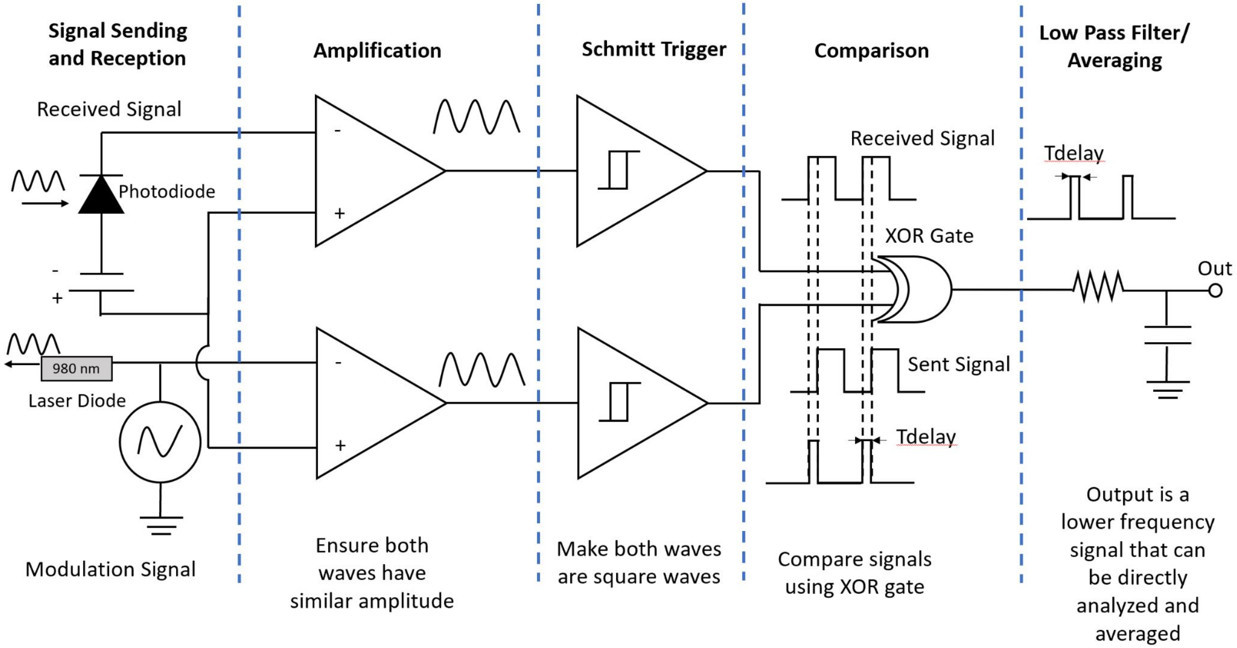
Time of flight analysis

Modulator



Time of Flight Circuit


Kevin Nilsen (PSE)



Based on S. Lineykin et. al. (2019)

Kevin Nilsen (PSE)

Notes on LiDAR

- The LiDAR and proximity sensors were originally made to assist with an autonomous navigation system, but that ended up being cut from the final prototype
  - So, the design needs to adapt
- Scanner has the capability for full two dimensional mapping, there is an option to disconnect the motor and have the scanner point only in the direction the camera is facing
  - Lets us only run the object detection program when the scanner senses an object
  - Dramatically reduces computing power
- In any future work, this system could be used in order to avoid environmental objects so the drone does not fly into anything

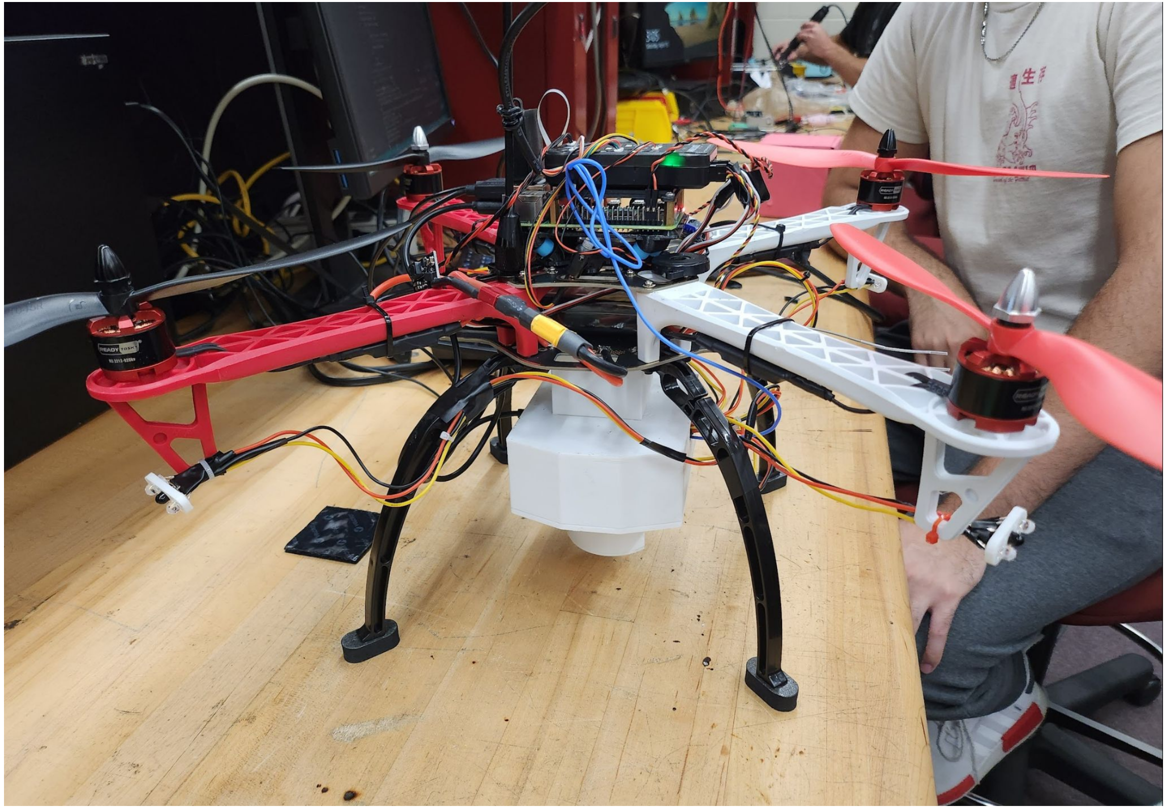
Kevin Nilsen (PSE)

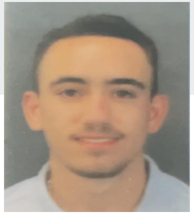
Future Improvements to LiDAR

- This system has some limitations in its accuracy and range
- Max range can be improved
  - Higher quality sensors
  - More powerful lasers
- In this particular system, the main limitation was the fact that the power of the laser was quite low in order to keep the cost down and to improve safety.
  - Perhaps in a future system, the laser could be changed, and a higher quality sensor could be used
- Additionally, a method exists to use two photodiodes and a beam splitter
  - This lets us compare the laser output to the received signal directly. Giving us a close comparison
  - More costly and bulky

Final Design

Jazmine Roman (EE)




Derek Murdza (CpE)

Conclusion and Acknowledgements

- This project allowed us to intertwine skills and knowledge gained from multiple disciplines to create a fully-functional drone and has been a very rewarding experience overall
- We are proud of the work that has been put into this project even with setbacks and challenges where we were able to find solutions to solve all problems that appeared
- We would like to acknowledge the following:
    - UCF Department of Electrical and Computer Engineering
    - Dr. Lei Wei
    - Dr. Samuel Richie
    - Dr. Aravinda Kar
    - Review Committee

Thank You!