DOMINANCE

Drone Mine Obstacle Avoidance

Group 18

Sponsored by Lockheed Martin

Caleb Jones (CpE)

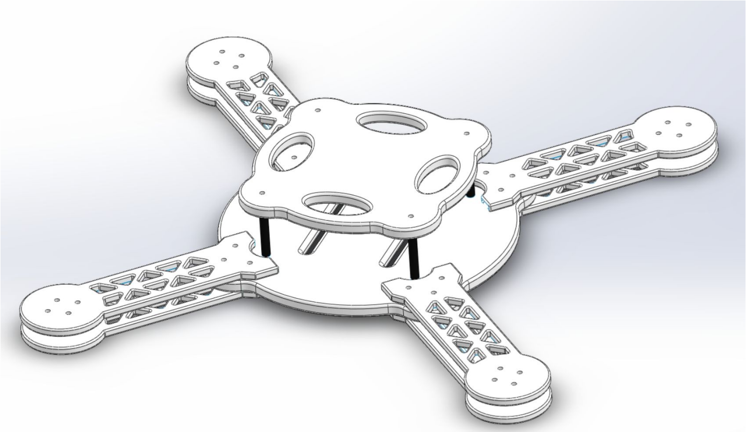
Hamza Siddiqui (CpE)

Rishi Jain (EE)

Ryan Lucas (EE)

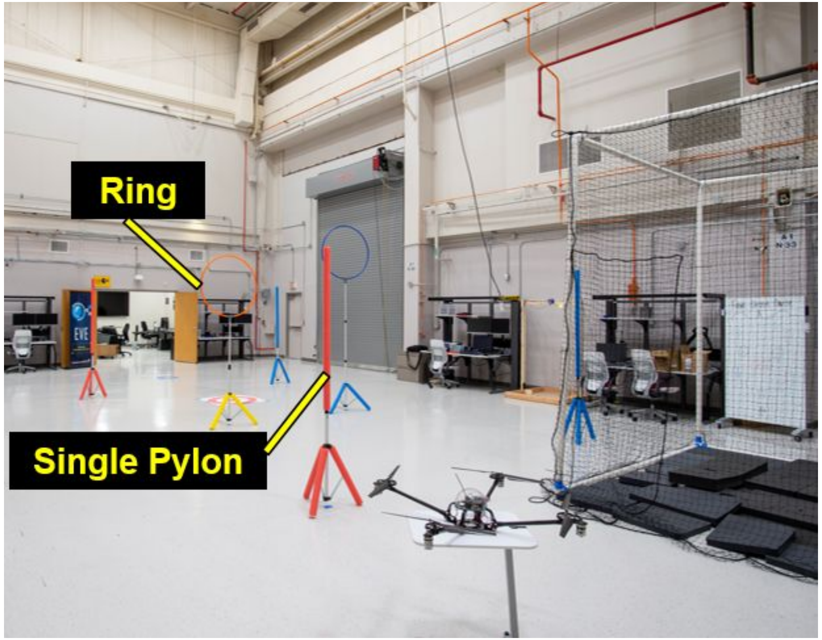
Original Project Objectives

- Create an autonomous drone
- Navigate all the obstacles in the obstacle course
- Avoid mine interference
- Return to starting position after navigating all obstacles
- Fly for at least 10 minutes (in standard operating conditions)



Original Project Scope

- Sponsored project by Lockheed Martin
- Competition project (3 UAV teams and 1 Mine team)
- Compete to earn points
  - Ring = 1 point
  - Single Pylon = 2 points
  - Double Pylon = 3 points
  - Acoustic Waypoint = 4 points
  - Point multiplier for successful consecutive obstacle maneuvering
- Goal: Accumulate enough points to win the competition



Original Customer Requirements (Operational Modes)

- Autonomous Mode: Autonomously Navigates an Obstacle Course
  - Auto Navigation (AutoNav) Submode: Navigate to obstacle
  - Auto Maneuver Submode: Maneuver around obstacle
  - E-Stop Submode: Immediately make a landing (safely stop UAV in case of emergency)
  - Take-off/Landing Submode: For taking-off and landing
- Manual Mode: Provides control to a human operator

Original Customer Requirements
(Object Detection & Vision)

- Detect customer specific obstacles (Ring, Single and Double Pylons)
- Determine distance to target objects
- Determine confidence level of target object
- Mark targets with red “X” on video feed
- Calculate ETA to target object
- Detect acoustic Waypoints and land near waypoints
- Obstacle data along with live video feed
- Communicate with ground station (perform E-stop procedure; Manual Mode)
- Map course and return back to start point after end of run.

Updated Customer Requirements (Post Pandemic Situation)

- Autonomously navigate through a ring
  - Takeoff, navigate to ring, pass through ring, land
- Autonomously navigate around a pylon
  - Takeoff, navigate to pylon, loop around pylon, land

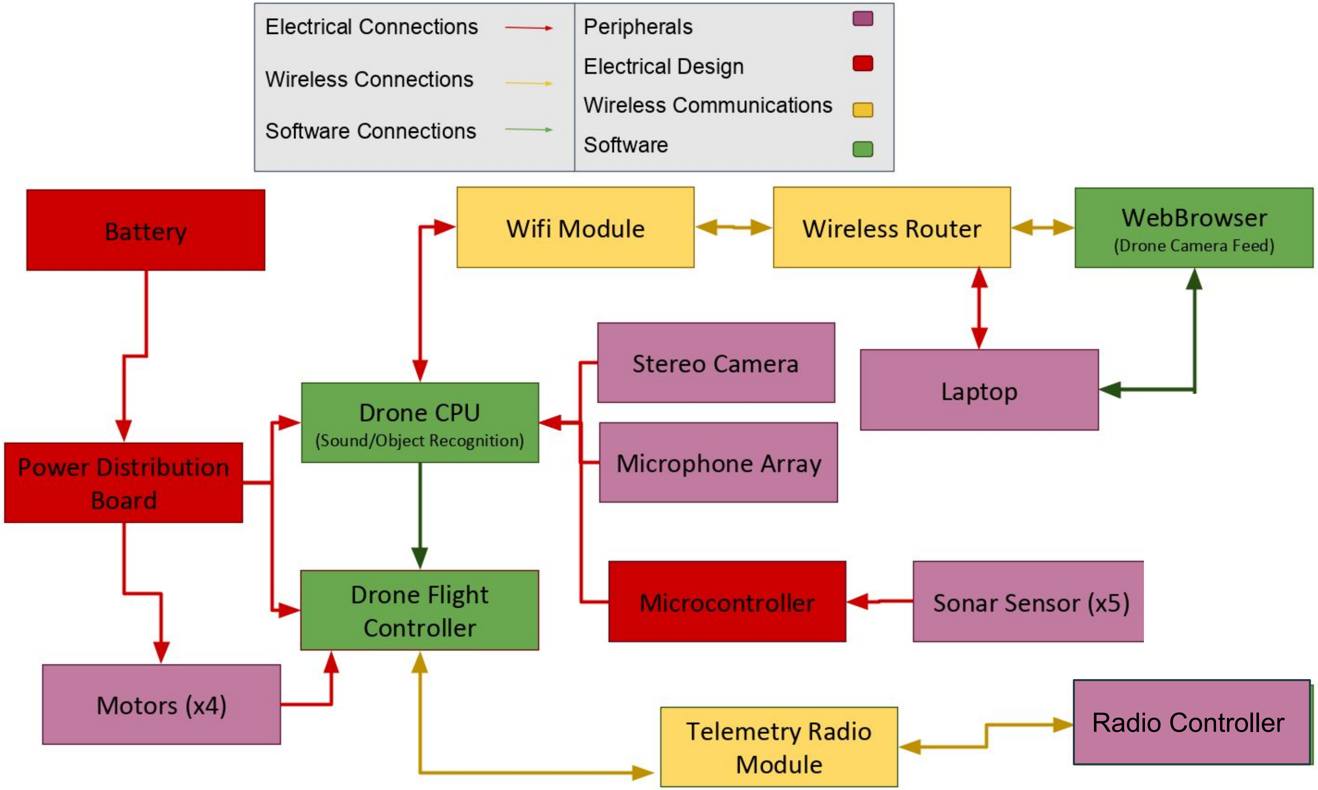
Customer Constraints

- GPS-denied navigation solution (needs to operate inside)
- Use of the YOLO vision algorithm is not allowed
- Budget: $1,650
  - $550 maximum for prototyping
  - $1100 maximum for final build
- Dimensional limits: 1.5ft x 1.5ft x 1.5ft
- Flight height limit: 45ft maximum height (to avoid ceiling collision)
- Maximum flight time: 15 minutes (for the competition)

Design Overview

- A vision algorithm on the drone computer uses an RGB image from a camera to detect objects in its FOV
- A depth camera would provide depth information to determine the distance to the objects
- The drone computer using the position of the objects would determine flight path and send commands to the flight controller in charge of managing motor speed
- The drone computer would send a video stream overlaid with information on obstacles in view to the ground station via a WiFi connection
- A radio controller would send commands to the flight controller to change modes; the drone computer will be able to detect these changes and stop/start issuing commands

Project Diagram



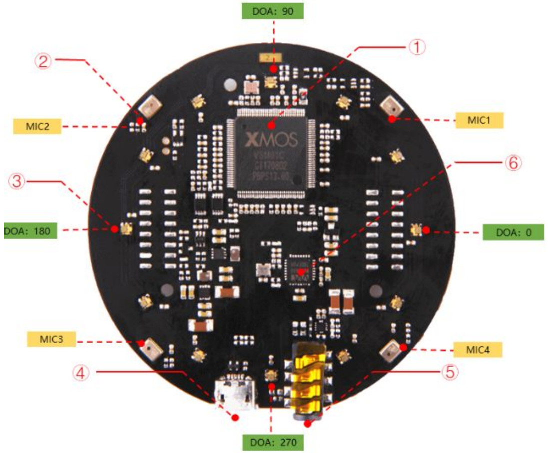
Hardware Part Selection

Microphone Array

Why The Seeed’s ReSpeaker Mic Array v2.0:

- High quality Microphones
- Allows for Acoustic Echo Cancellation (AEC) and Direction of Arrival (DoA)
- Detects sound from 5 meters away (16.4 feet)
- Noise Filtering Parameters

Microphone	Mic Array v2.0	2-Mics Pi	4-Mics Pi
Sound Processor	XMOS-XVF3000 (stereo-AEC voice processor)	WM8960 (low power stereo codec)	X-Power AC108 ADC (x2)
Microphones	MP34DT01TR-M (x4) (digital)	MSM321A3729H 9CP (x2)(analog)	MSM321A3729H 9CP (x4)(analog)
Capture Radius	16.4 feet	10 feet	10 feet
Cost	$64.00	$9.90	$24.90



Seeed’s ReSpeaker Mic Array v2.0

Flight Controller

Purpose:

- Maintain stability of the drone
- Translate user input into engine output
- Gather real-time data

ReadyToSky PixHawk Features:

- Inertial Measurement Units
- System-on-chip with backup system-on-chip
- Able to be flashed with new firmware
- Able to be controlled by an external computer
- ArduPilot Compatible

ReadyToSky Pixhawk

Flight Controller Comparison

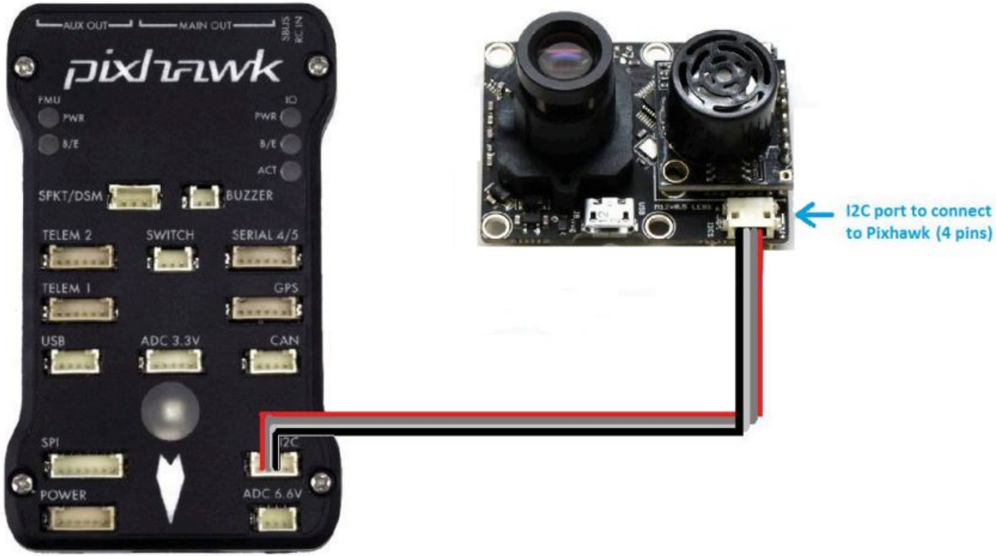
Flight Controller	HGLRC F4.V2	Readytosky Pixhawk
Processor	32-bit	32-bit
Flight Stack	BetaFlight	PX4 or Ardupilot
I2C	No	Yes
SPI	No	Yes
Price	$33.99	$72.99

Optical Flow & Height Sensor

PX4FLOW

- Provides an optical camera designed to be interfaced with the PixHawk
- Ultrasonic height sensor

- Gyroscope
- Flight controller firmware, aware of PX4FLOW
  - Determine height and groundspeed

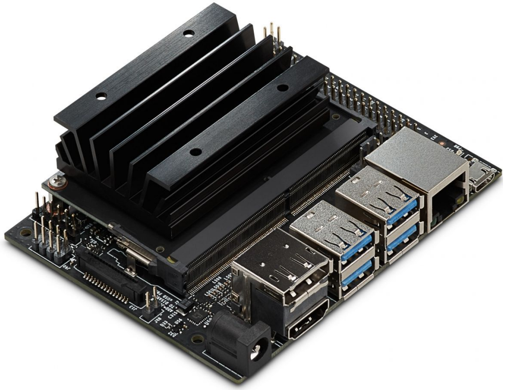


Drone Computer

Computer Requirements:

- Process images from camera to identify obstacles at a rate of at least 5 times per second
- Determine distances to objects using data from depth camera and vision algorithm
- Determine the flight path of the drone
- Send flight commands to the flight controller
- Transmit video to the ground station
- Receive and respond to commands from the ground station
- Support all necessary peripherals (wifi module, microcontroller, camera, microphone array, flight controller)

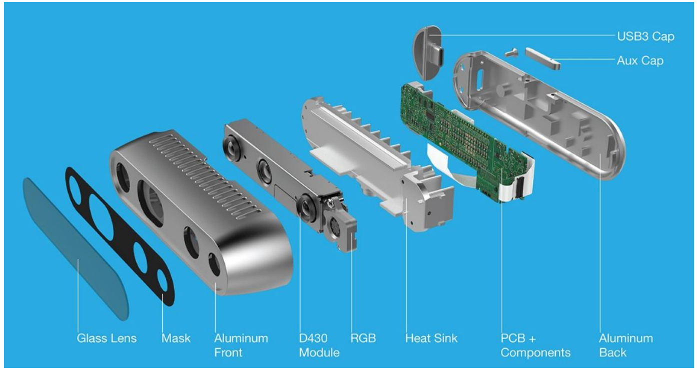
Drone CPU



Computer	Jetson Nano	Raspberry Pi 4B (4GB)
Processor	Cortex-A57 (4 cores)	Cortex-A72 (4 cores)
Clock Rate	1.42GHz	1.5GHz
Power Consumption	10 W	5 W
GPU	Maxwell (128 CUDA cores)	VideoCore VI
Weight	4.1 oz	1.6 oz
Price	$99	$55

- Large amount of computer vision resources (academic and otherwise) use Nvidia GPUs
- Nvidia provides their JetPack SDK for developing computer vision and AI applications

Depth Camera



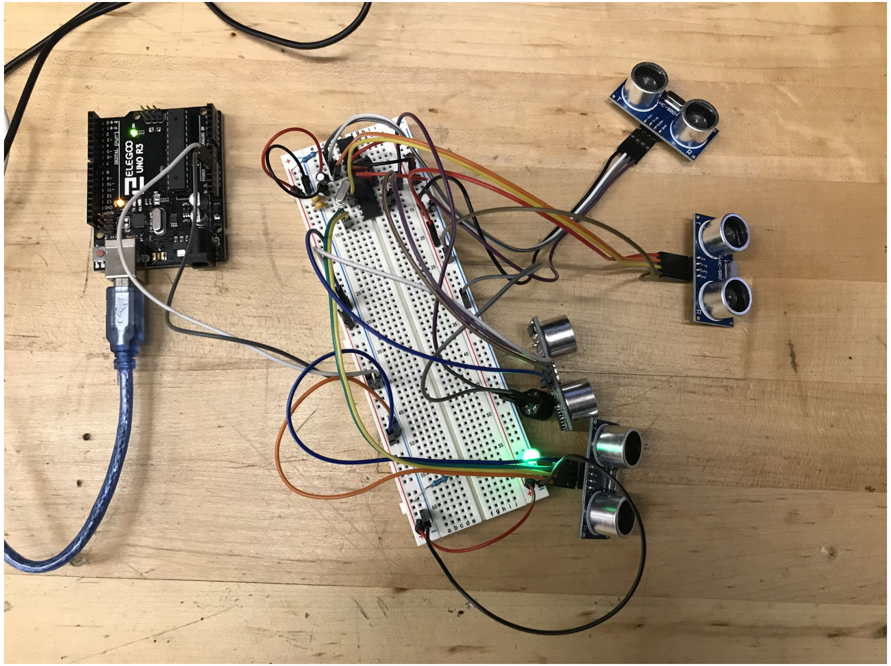
- A camera is necessary to provide a video stream for detecting objects and navigation
- The choice for a depth camera stems from the need to determine distance to the objects

Intel RealSense Depth Cameras

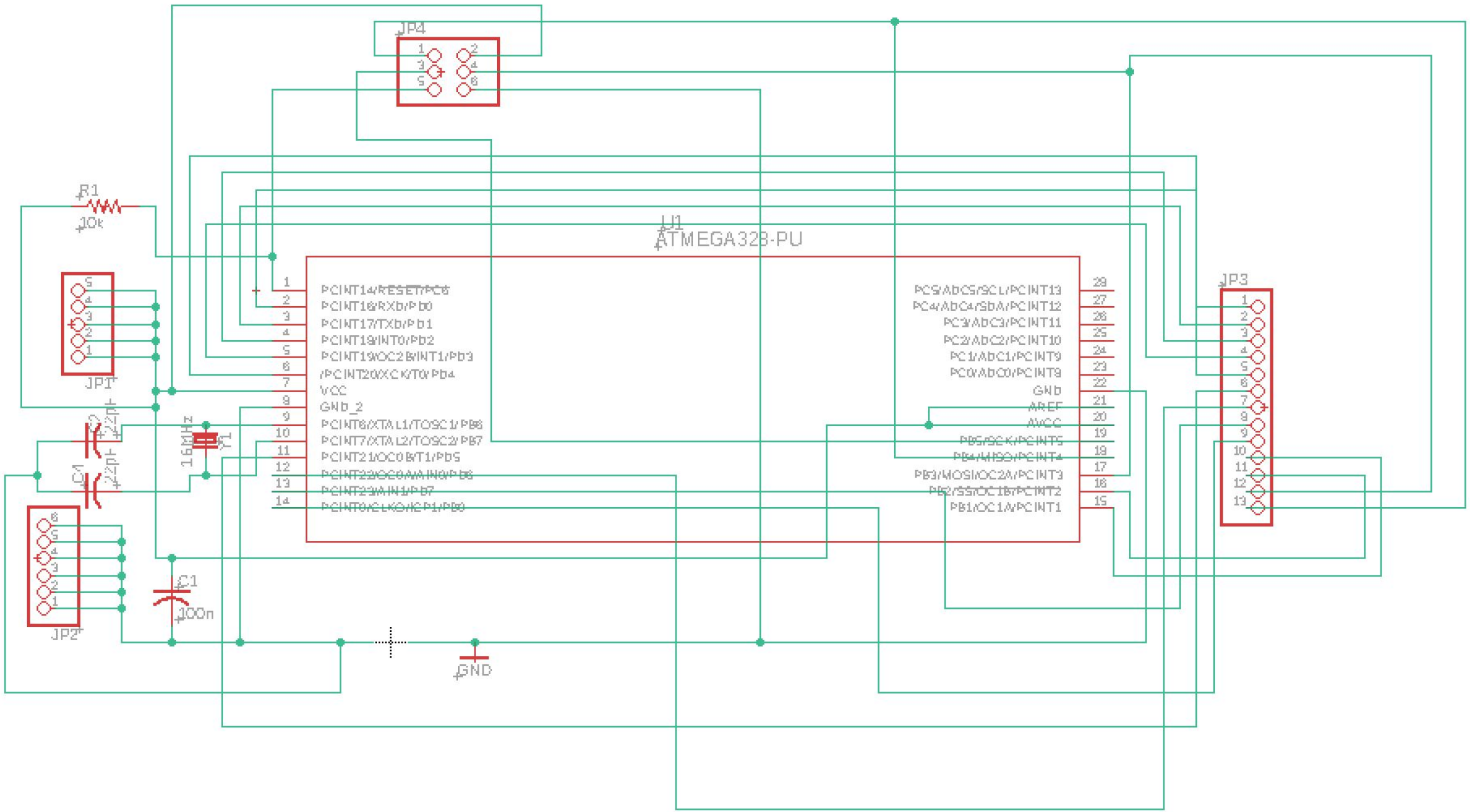
Camera	D415	D435	D435i
Depth FOV	63.4°x40.4°	85.2°x 58°	85.2°x 58°
Depth Resolution	720p @ 90fps	720p @ 90fps	720p @ 90fps
RGB FOV	70° x 42°	70° x 42°	70° x 42°
RGB Resolution	1080p @ 30 fps	1080p @ 30 fps	1080p @ 30 fps
IMU	No	No	Yes
Cost	$149	$179	$199

Hardware/PCB design

- Using an ATmega328pu to manage 4x ultrasonic sensors
- Sensors will be placed on front, left, right, and back sides of the drone
- Buzzer will indicate when drone is too close to an object
- Proximity data can be relayed back to the Jetson Nano and used to assist autonomous flight



Hardware/PCB design



Component Power Draw

Component	Voltage	Current	Power
Jetson Nano	5V	2-4A	10-20W
Intel RealSense Depth Camera	1.8V	83mA	150mW
Readytosky PixHawk	0.3-3V	0.83-8.3A	0.249-24.9W
Cobra CM-2206/17 2400kV Motors	6-8V	0.97-1.06A	5.82-8.48W
Spedix GS30 32bit DShot 1200 30A ESC	8-25V	30A	240W
Hereflow Optical Flow and Lidar Sensor	5V	800mA	4W
Speed’s ReSpeaker Mic Array v2.0	5V	180mA	0.9W
PCB with Distance/Proximity Sensors	5V	1mA	5mW

Motors

Cobra CM-2206-17 Motor Test Data, Kv=2400

Data Collected at 14.8 volts with HQ 5x4 Prop						
Throttle Setting	Motor Amps	Input Watts	Prop RPM	Thrust (Grams)	Thrust (Ounces)	Efficiency Grams/W
10%	0.64	9.46	6,756	49.9	1.76	5.28
20%	1.47	21.70	10,228	117.6	4.15	5.42
30%	2.62	38.70	13,038	192.1	6.78	4.96
40%	3.82	56.55	15,223	263.8	9.31	4.66
50%	5.07	75.07	17,084	332.8	11.74	4.43
60%	6.61	97.77	18,970	411.0	14.50	4.20
70%	8.75	129.44	21,200	512.2	18.07	3.96
80%	11.77	174.12	23,768	638.7	22.53	3.67
90%	15.59	230.70	26,185	779.3	27.49	3.38
100%	20.80	307.80	29,016	969.4	34.19	3.15

Prop Manf.	Prop Size	Li-Po Cells	Input Voltage	Motor Amps	Input Watts	Prop RPM	Pitch Speed in MPH	Thrust Grams	Thrust Ounces	Thrust Eff. Grams/W
HQ	4x4	4	14.8	15.29	226.3	30,758	116.5	650.9	22.96	2.88
HQ	4x4x3	4	14.8	19.18	283.9	29,469	111.6	776.5	27.39	2.73
HQ	4x4.5	4	14.8	11.62	172.0	31,938	136.1	475.0	16.75	2.76
HQ	4x4.5-BN	4	14.8	21.43	317.1	28,782	122.7	780.6	27.53	2.46
HQ	5x3	4	14.8	14.99	221.9	30,998	88.1	757.0	26.70	3.41
HQ	5x4	4	14.8	20.80	307.8	29,016	109.9	969.4	34.19	3.15
HQ	5x4x3	4	14.8	27.23	403.0	27,052	102.5	1146.8	40.45	2.85
HQ	5x4.5	4	14.8	23.17	342.9	28,320	120.7	978.2	34.50	2.85
HQ	6x3	4	14.8	22.46	332.5	28,513	81.0	1118.0	39.44	3.36

Cobra CM-2206/17 2400 kV

Approx diameter	Prop Size	Recommended stator size	Lowest kv	Highest kv
150-250mm	4”	1806	2600	2800
190-220mm	5”	2204-2206	2300	2600
220-270mm	6”	2204-2208	1960	2300
350mm	7”	2206-2210	1450	1600

Electronic Speed Controllers

Old ESC

New ESC

- ESC and motor are a combination package and are directly compatible with each other.
- 30A maximum current draw
  - the 5x4 propellers draw 20A at max throttle
  - the 5x3 propellers draw 15A at max throttle

Batteries

Battery	Venom Fly	Dynamite Reaction	Tattu
Capacity (mAh)	3200	5000	10000
Voltage (V)	14.8	11.1	22.2
Discharge Rate (C)	30	50	25
No. Cells (S)	4	3	6
Weight (g)	330	204	1400
Price (USD)	59.99	74.49	185.00

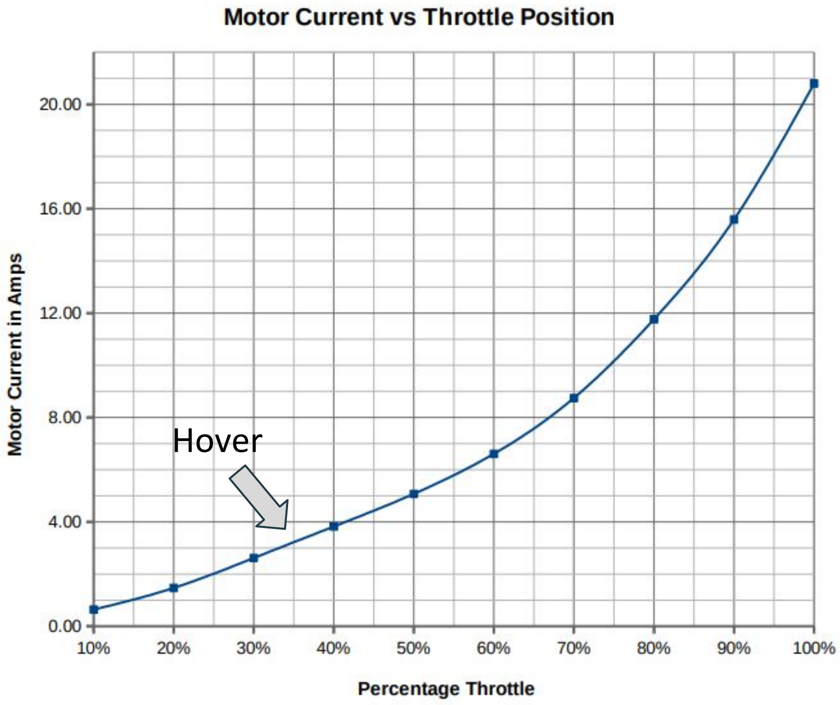
Power Supply

Max Current Draw Allowed :

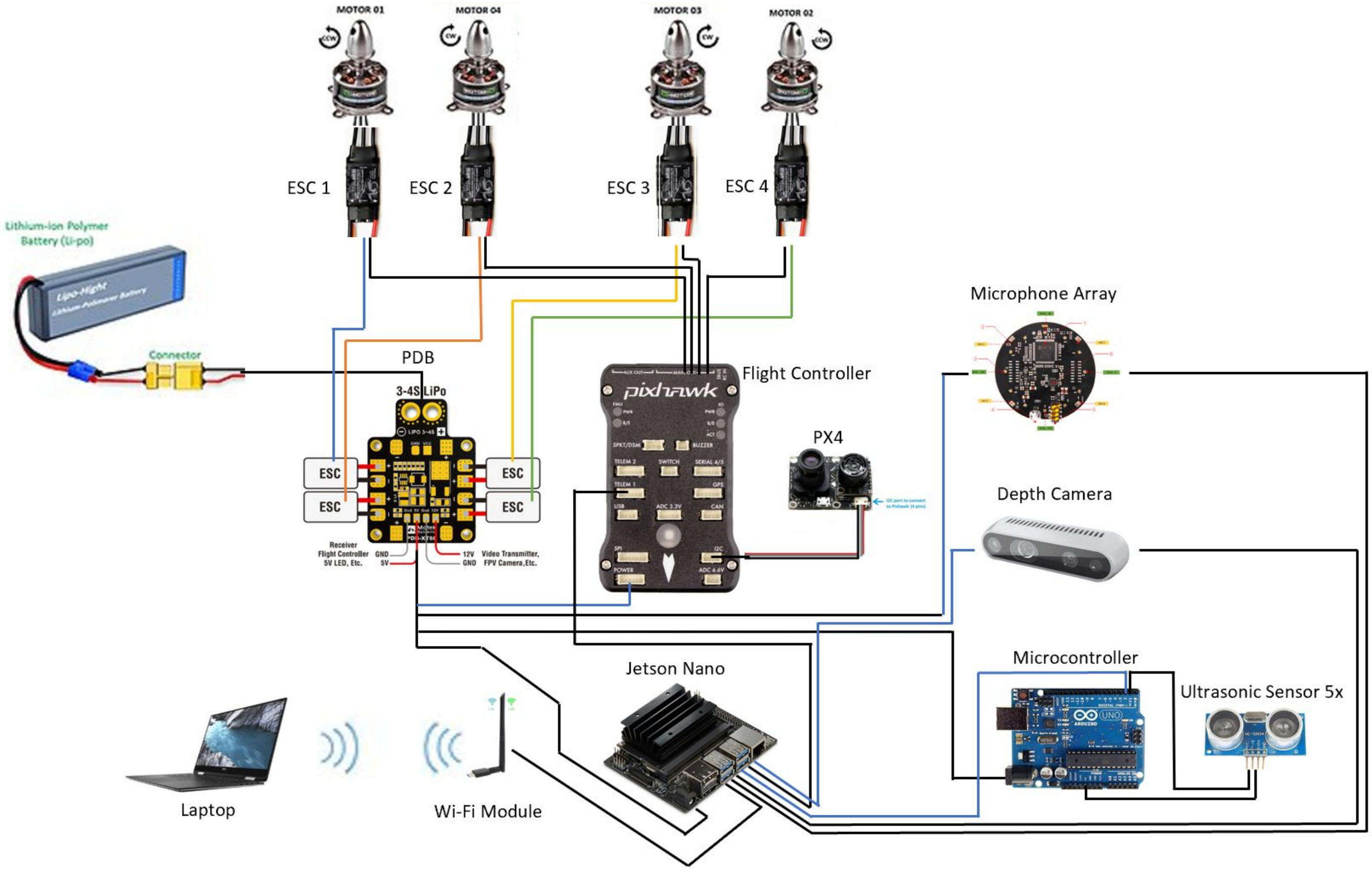
Capacity x C-Rating = 3.2 x 30 = 96A

Flight Time for 85% Discharge :

$$\frac{(mah\ Battery/1000)\ x\ (.85)}{(21\ Average\ Amps)}(60) = 7.7\ minutes$$

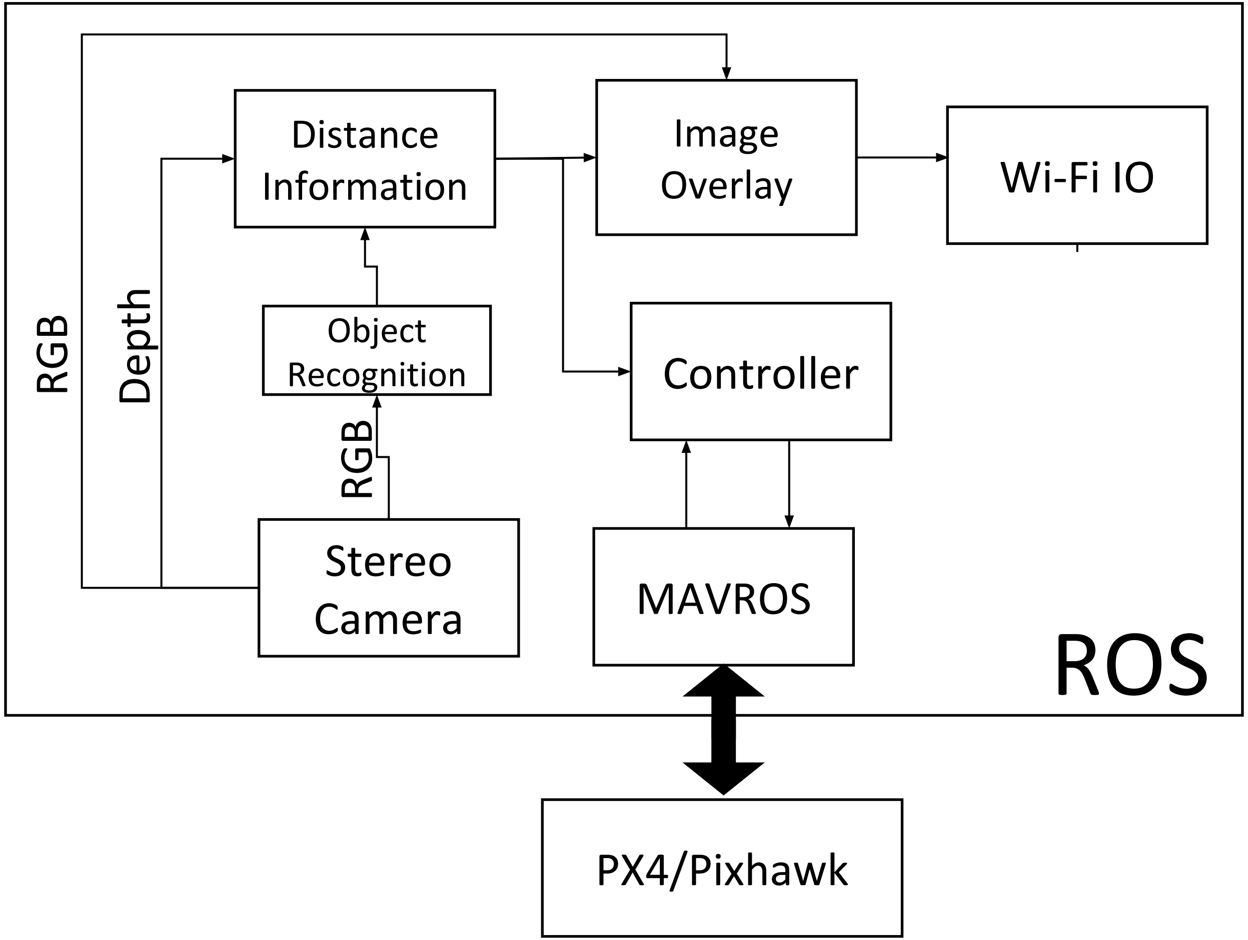


Component Connection Overview



Software Design/Integration

Drone Software Diagram



Obstacle Recognition Node

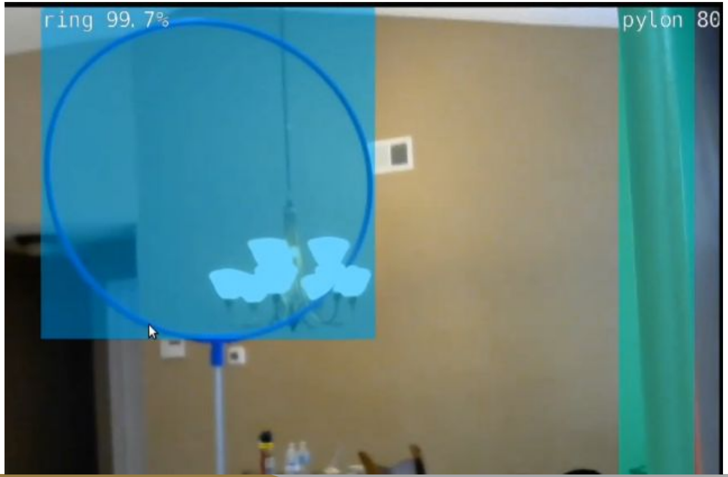
A computer object recognition algorithm running on the drone computer takes an RGB image from a computer and use this information to create bounding boxes around rings and pylons in view.

Based on benchmarks for object recognition models and software running on our hardware, we chose to use a single shot detector (SSD) that uses a Mobilenet V2 backend as the image recognition model.

Our object recognition node is called ros_deep_learning and was designed by Nvidia to run on the Jetson series of computers.

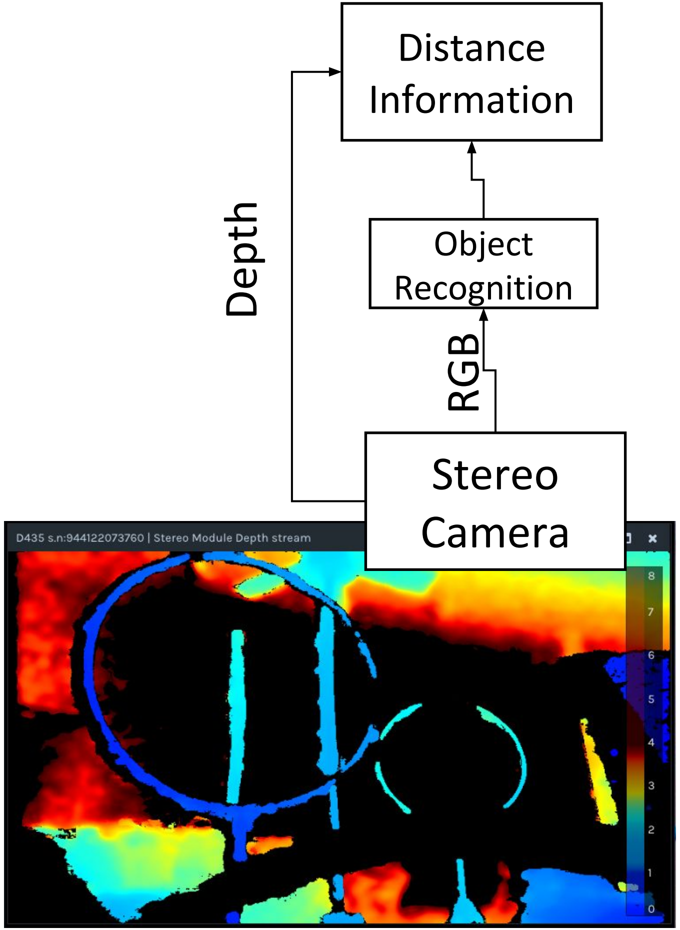
Using Default Coco Dataset

Using Custom Dataset to detect Pylons and Rings



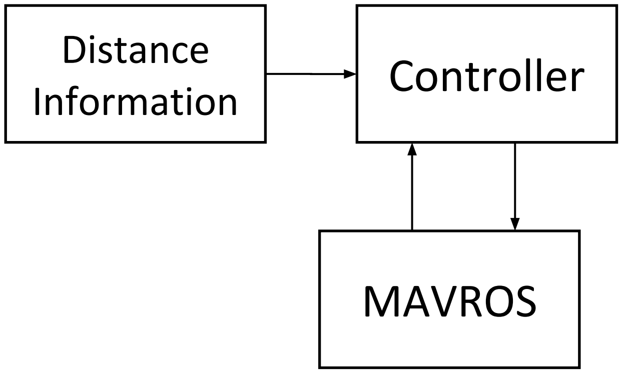
Distance Estimation Node

- This node would be given the bounding boxes of each of the objects in the FOV of the RGB camera
- The bounding box would be translated from the RGB image to the depth image from the depth camera
- Based on the type of image, the node determines distance
- Data on the closest obstacle would sent to the controller node and image overlay node



Drone Controller Node

- When in Autonomous Operation mode:
  - Would controls what submode the drone is currently in based on received data
    - Most of the time in the AutoNav submode navigating to an obstacle
    - When positioned in front of obstacle, would enter Auto Maneuver submode to navigate obstacle
    - When 0.5-1 kHz audio signal is picked up, would enter Take-Off/Land submode
- When in Manual Operation mode:
  - Flight controller commands flight based on radio signals sent from drone controller.



Software Operation

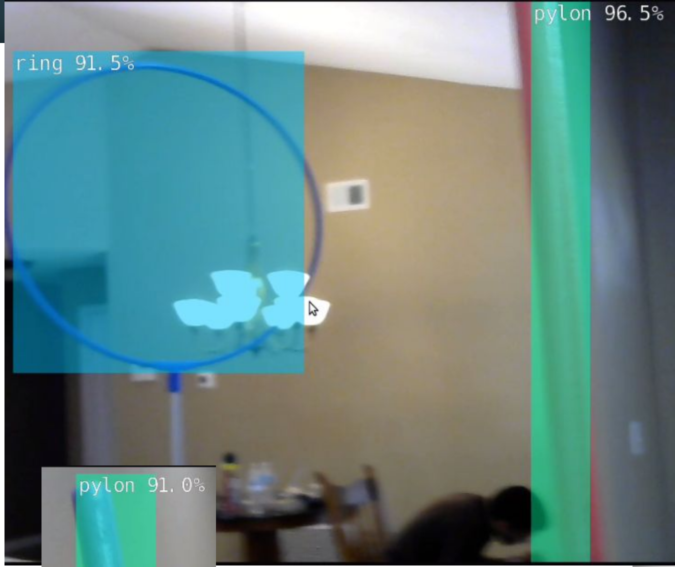
- Software startup is accomplished by using SSH to access the shell of the Jetson Nano
  - Once into the shell, programs can be started up individually in this order:
    - Camera Node
    - Computer Vision Node
    - MAVROS
    - Drone Controller Node

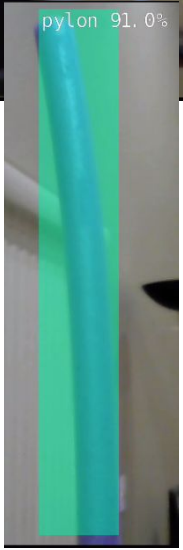
Project Results

Final Build

Obstacle Detection

- Model was trained on one of our members computers using Ubuntu 18.04, Tensorflow 14.0 and a 980ti GPU using set of 800 images labeled using imgLabel
- Initial training without GPU optimizations took about 3 days. With GPU optimizations training took about 8 hours.
- Frozen model was moved from training computer to Jetson Nano where a TensorRT tool was used to convert the model to a UFF format
- TensorRT was then used to convert the UFF file into a file easily interpreted by the GPU for inference



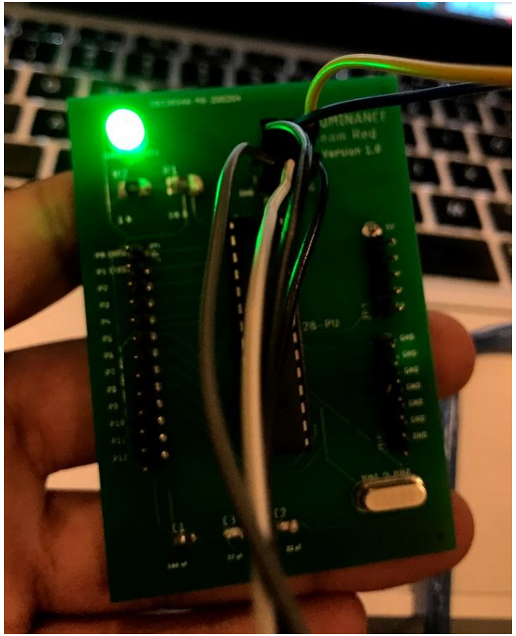


Consistently reaches 40+ FPS

PCB/Microcontroller

- Successfully implemented circuit on breadboard
- Able to communicate sensor measurements via UART
- Fabricated PCB had some signal interference
    - Difficult to diagnose without lab tools

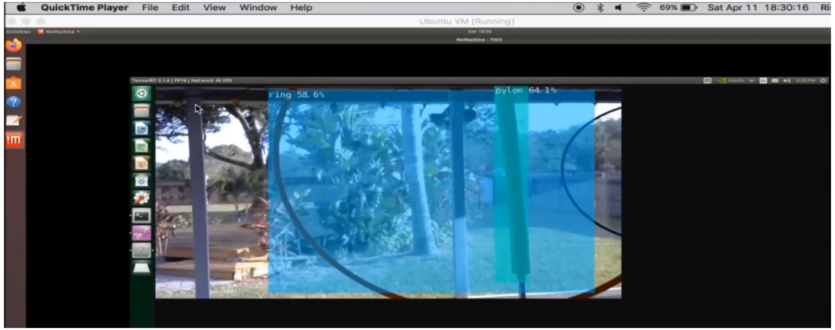
```
21:31:15.438 -> S1: 3; S2: 6; S3: 6; S4: 3; S5: 0
21:31:16.911 -> S1: 3; S2: 12; S3: 13; S4: 3; S5: 0
21:31:18.329 -> S1: 3; S2: 4; S3: 7; S4: 3; S5: 0
21:31:19.789 -> S1: 3; S2: 4; S3: 6; S4: 3; S5: 0
21:31:21.229 -> S1: 4; S2: 4; S3: 6; S4: 3; S5: 0
21:31:22.681 -> S1: 3; S2: 4; S3: 6; S4: 3; S5: 0
21:31:24.111 -> S1: 3; S2: 4; S3: 5; S4: 3; S5: 0
21:31:25.572 -> S1: 7; S2: 4; S3: 5; S4: 3; S5: 0
21:31:27.028 -> S1: 5; S2: 4; S3: 5; S4: 3; S5: 0
21:31:28.474 -> S1: 242; S2: 4; S3: 5; S4: 3; S5: 0
21:31:29.941 -> S1: 239; S2: 4; S3: 5; S4: 3; S5: 0
21:31:31.406 -> S1: 237; S2: 4; S3: 5; S4: 3; S5: 0
21:31:32.834 -> S1: 202; S2: 4; S3: 5; S4: 3; S5: 0
21:31:34.306 -> S1: 238; S2: 4; S3: 5; S4: 3; S5: 0
```



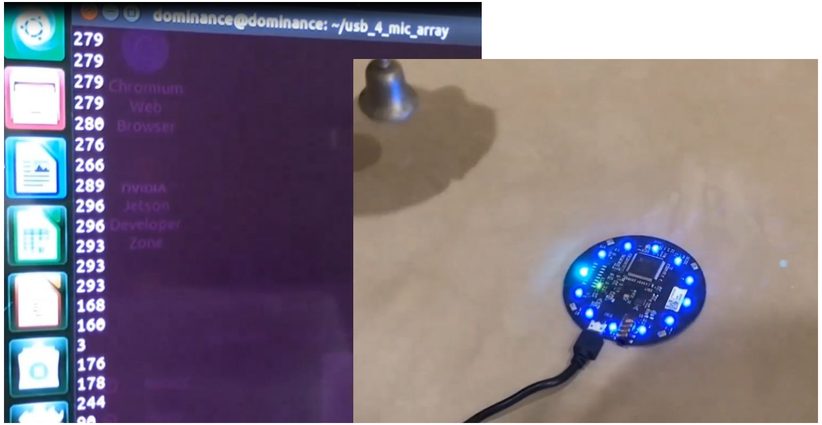
Remote Control and Sound Detection

- Remote Control of the Drone was established by an SSH connection into the Jetson Nano via wifi
  - However, running the Object Recognition lead to too much power draw from the CPU resulting in a shutdown

- Sound localization was achieved with the mic array
  - Determines the angle to the sound
  - Limits frequency to 180+Hz to filter sound



SSH into Jetson Nano while tethered



Mic Array displaying angle to high frequency bell

Power System

- All components on drone were able to stay powered during idle
- Increasing the computer CPU load caused intermittent power failures
- Unreliable power system made drone unsafe to fly autonomously

Autonomous Flight

Autonomous Flight was not achieved for two main reasons:

- Drone is unable to self stabilize and hold position sufficiently
  - Sudden jerks in directions, cutting of motors
  - Could be due to flow camera issues
    - PX4Flow (Optical Flow Sensor/Ultrasonic Sensor) was putting out obviously incorrect data
    - HereFlow (Optical Flow/Lidar Sensor) was bought to replace, but still was unable to hold a position; similar issues to PX4Flow
  - Could be due to issues with the optical flow sensors measuring height
  - Could be that our flight controller was too out of date to integrate with these sensors.
- Power not sustainable enough to run the necessary software

Administrative Content

Division of Work

Electrical and Computer Engineering	Mechanical and Aerospace Engineering
• Ground Station Communication • CPU Integration • Object Recognition and Mapping Algorithm • Sound Recognition and Filtering • Data Processing • Power System • PCB/Microcontroller Development • Sensors	• Drone Frame Design • Sensor Mounts • Flight Controller • Electronic Speed Controllers • Motors • Propellers • Balancing

Division of Responsibilities

Component	Responsibility	Assist
Object recognition and mapping	Caleb	Hamza, Rishi, Ryan
Wireless communication, noise filtering, ground station data transfer	Hamza	Caleb, Rishi, Ryan
Power system, PDB, battery	Ryan	Rishi, Caleb, Hamza
PCB, Microcontroller, sensors	Rishi	Ryan, Caleb, Hamza

Assembled Drone Cost

Component	Name	Unit Cost	Quantity	Cost
Jetson Power Cable	Adafruit 5V 4A Supply	$14.95	1	$14.95
Battery	Venom 4s 30c 3200mah14.8V LiPo battery	$59.99	1	$59.99
Drone Wifi Module	Geekworm NVIDIA Jetson Nano Wi-Fi Adapter	$16.79	1	$16.79
Companion Computer	NVIDIA Jetson Nano Developer Kit	$99.00	1	$99.00
Flight Controller	ReadyToSky Pixhawk	$74.99	1	$74.99
Optical Flow/Lidar	HereFlow Optical Flow/Lidar Sensor	$49.99	1	$49.99
Power Distribution Board	PDB XT60 Matek Power Distribution Board	$8.49	1	$8.49
ESCs	Spedix GS30 32bit DShot 1200 30A ESC	$12.95	4	$51.80
Propellers	HQ Prop 5x3 Propellor (Black) (4)	$0.49	4	$1.96
PCB	Custom Microcontroller BOM	$9.40	1	$9.40
Ultrasonic Sensors	HC-SR04	$3.95	4	$15.80
Motors	CM-2206/17-V2 MULTIROTOR MOTOR KV=2400	$22.99	4	$91.96
Microphone	ReSpeaker Mic Array v2.0	$64.00	1	$64.00
Mounting Tape	Double Sided Tape, M3 Nuts/Bolts/Standoffs, Cable Ties, Battery Clip	$26.71	1	$26.71
SD Card	MicoSDXC	$19.49	1	$19.49
Depth Camera	Intel RealSense D435	$177	1	$177.00
			Total:	$782.32

Prototyping Cost

Component	Name	Unit Cost	Quantity	Cost
Optical flow camera	PX4 Flow (Donated)	$0	1	$0
Electronic Speed Controller	Cobra MR30	$27.99	4	$111.96
Miscellaneous PCB Components	PCB Prototype and Mouser Components	$27.78	1	$27.78
PPM Encoder	ShareGoo 8CH PPM Encoder & I2C Splitter	$13.99	1	$13.99
			Total:	$153.73

Project Summary

Project Successes

We were able to present a competent design for an autonomous drone:

- Software implemented on the Jetson Nano is able get live feed video from the Intel Realsense D435 and pass it to the object detection node
- The object detection node is able to output relatively accurate bounding boxes 40+ times per second.
- Drone is able to be manually flown by an operator.
- Can remotely communicate with the drone to send software commands and receive video feed.

Project Challenges

- Delays in receiving parts due to MAE department ordering process
- Integration of parts to create basic drone was time consuming
- Unable to test drone at Lockheed Martin's drone facility
  - Very few public places are available to fly experimental drones
- Inability to utilize senior design lab for second half of semester
- Unable to congregate as team due to pandemic stay-at-home orders
  - Project had many subsystems, and required subsystem leads to be present for drone operation

Suggestions for Future Projects

- For future drone projects:
  - If you have sponsor money, don't be afraid to spend it
  - Order parts early if using sponsor/school funds
  - Source parts from vendors with high inventory and fast shipping (in case anything breaks)
  - Very software heavy; make sure you have at least 3 students familiar with writing software
- For remote Senior Design semesters:
  - For instructors:
    - Provide basic lab tools for checkout (Multimeters, soldering irons, etc)
  - For students:
    - Choose a project that will allow for multiple people to work on it without splitting up your hardware between too many group members

Thank you for viewing our presentation

More information on our project can be found in our demo video:

https://drive.google.com/file/d/1mfKd8VyEHFhHomFzC-8NLHOd8mco6xrR/view