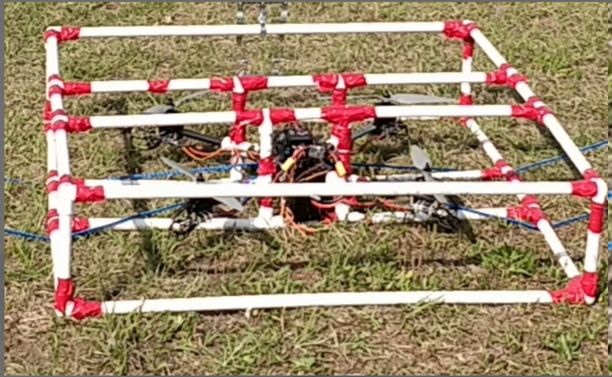
Robocopters Quadcopter Competition for Lockheed Martin
BLUE TEAM




Group 4:
Justin Bates
Kelsey Cameron
Jacob Hazelbaker
Trisha Singh

LOCKHEED MARTIN

What is a RoboCopter?

- Autonomously search for other “enemy” drones
- Intentionally collide with them in a game-like environment to score points.
- Allow us to take out existing drone threats before they become a problem.
- Explore object detection algorithms and how they can be used within military applications.
- Deploy drones to save human lives in threatening conditions.

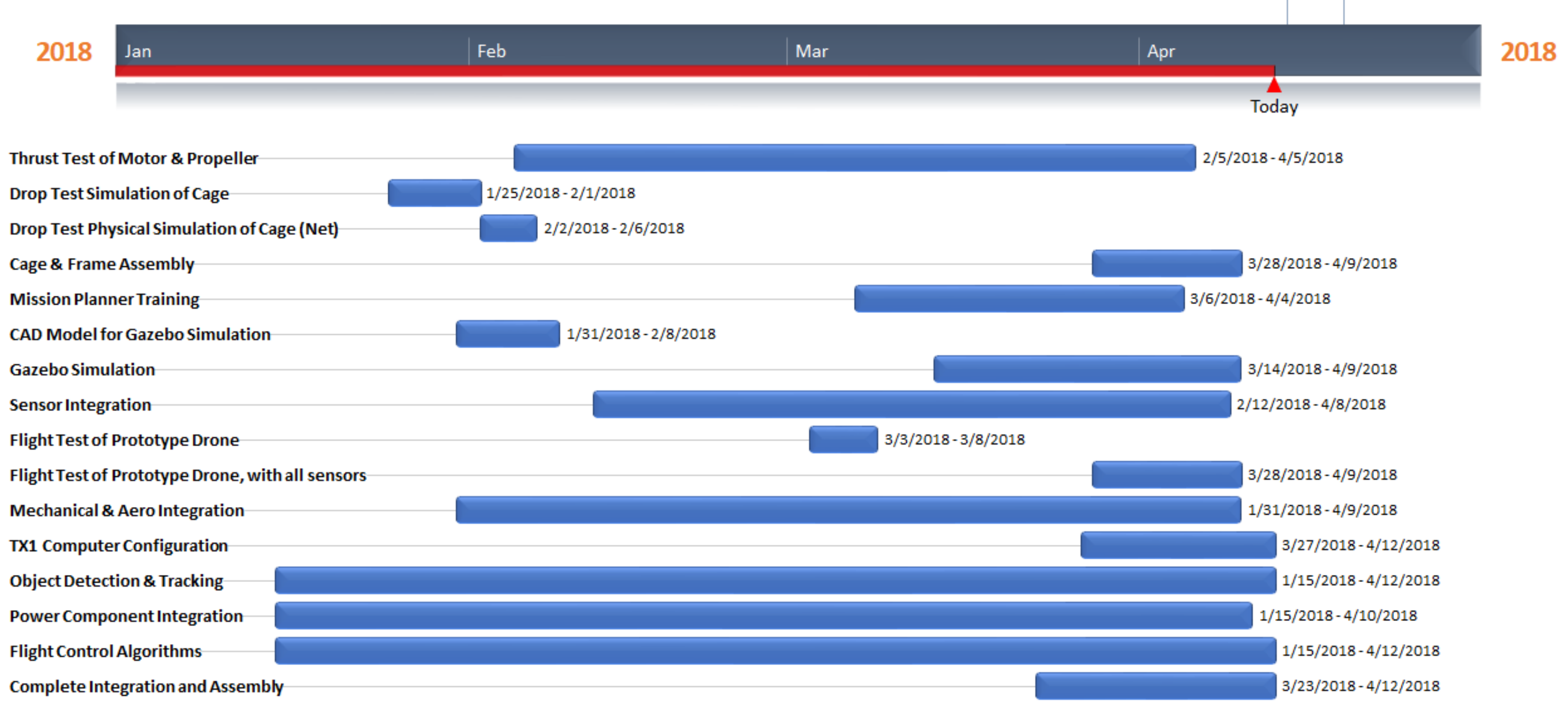
Measurable Specifications

1.0	The quadcopter must not fly beyond the 30ft x 30ft x 30ft field.
1.1	The quadcopter must support at least at least 2 pounds of electronics.
1.2	The quadcopter must fly at least 10 miles per hour.
1.3	The quadcopter must fit within a 4 ft x 4 ft x 4ft box, including the dimensions of the cage.
1.4	The quadcopter must include a processor that is at least 400 MHz and quad-core, in order to effectively use the object detection algorithms.
1.5	The power supply must support voltages in the range of 11 volts to 14 volts.
1.6	The printed circuit board must be capable of delivering 3 to 5 Volts.

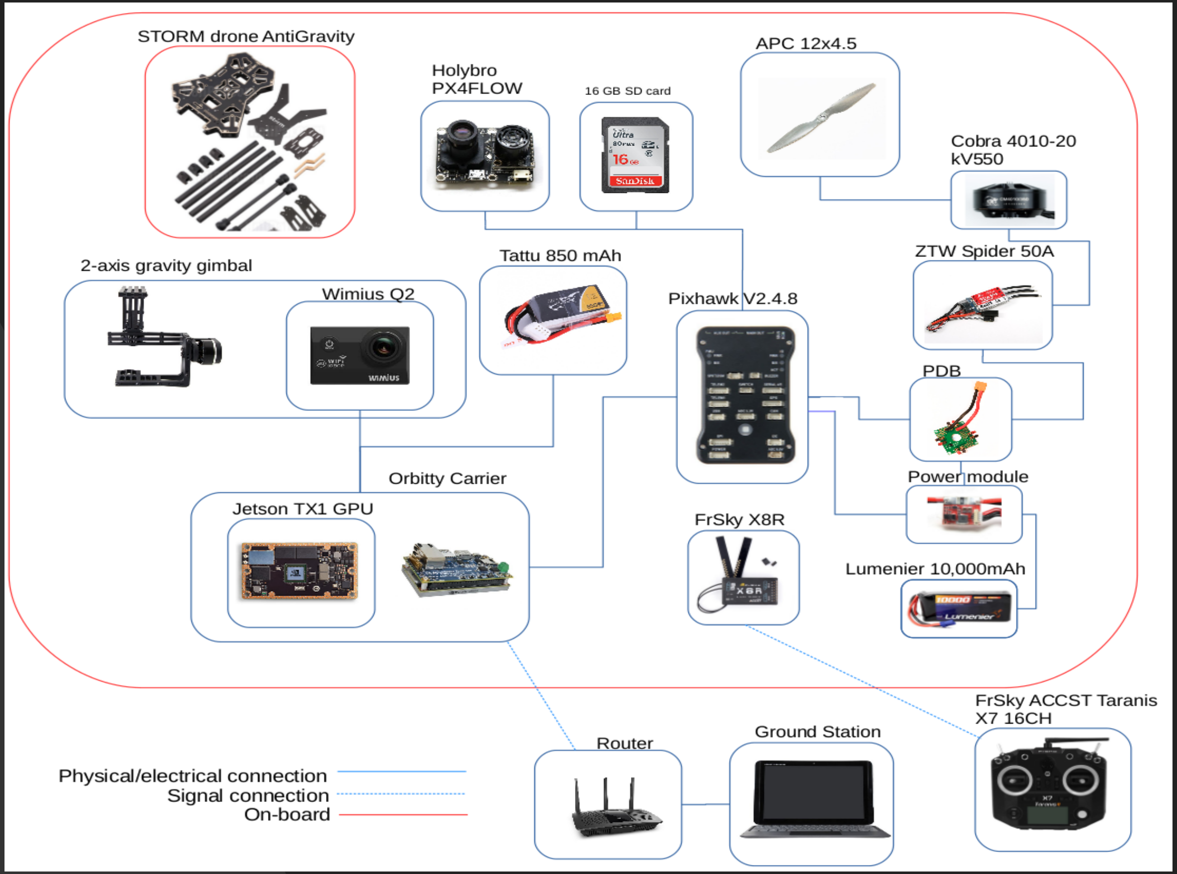
Money Management

	Total allowed (as demonstrated model)	$ 1,500.00
	Components:	
1	Main battery (Lumenier 10000mAh 6s 25c Lipo)	$ 170.00
2	Battery (TATTU 850 mAh)	$ 15.00
3	Frame (STORM Drone AntiGravity GPS Flying Platform)	$ 160.00
4	Cage	$ 55.00
5	Lidar Range Finder	$ 41.00
6	Pxflow (Holybro PX4FLOW Kit v1.31)	$ 105.00
7	Camera (WIMIUS Q2 Action Camera HD)	$ 32.00
8	Pixhawk	$ 127.00
9	Motors (Cobra CM-4010, Kv=550)	$ 260.00
10	Propellers (APC 12x4.5)	$ 24.00
11	Motor controllers (ZTW SPIDER 50A)	$ 80.00
12	Orbitty Carrier Board	$ 188.00
13	TX1 GPU	$ 210.00
14	Gimbal	...
	As demontrated model:	$ 1,467.00

BLUE TEAM SPRING SEMESTER PLAN



Hardware Selection



Final Drone versus Prototype

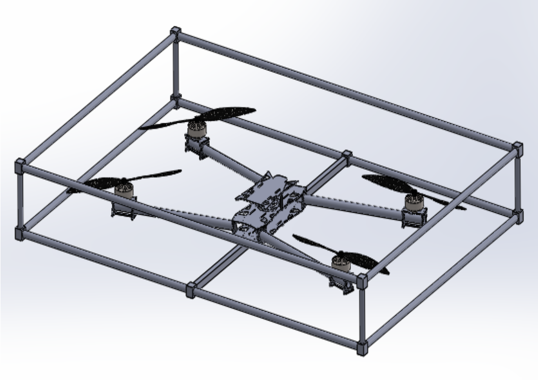
Shape: Cuboid

Weight: 6.5 lbs

Volume: 3ft x 3ft x 0.75ft

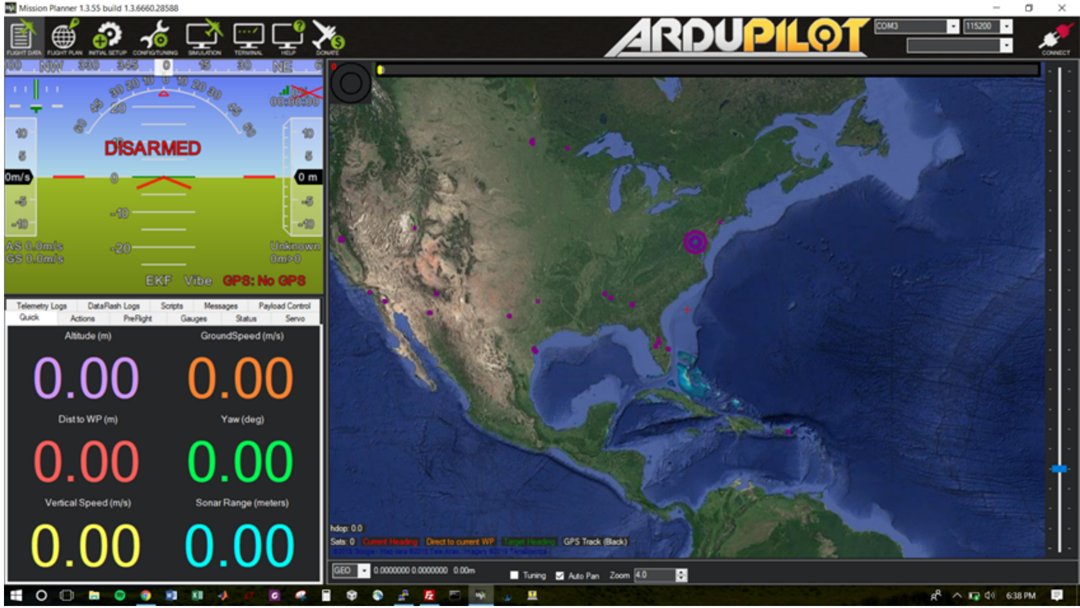
Materials:

- ½” PVC Pipes
- ½” PVC Fittings
- Plastic Net
- Velcro & Zip Ties Ties



Mission Planner

- The software that corresponds with Pixhawk firmware
- Thousands of parameters.
- Diagnosing Errors.
- Programming flight modes.
- Logging graphs for specific sensors.
- Programmable switches

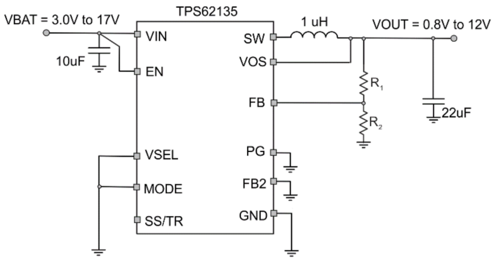


Overcoming Many Challenges

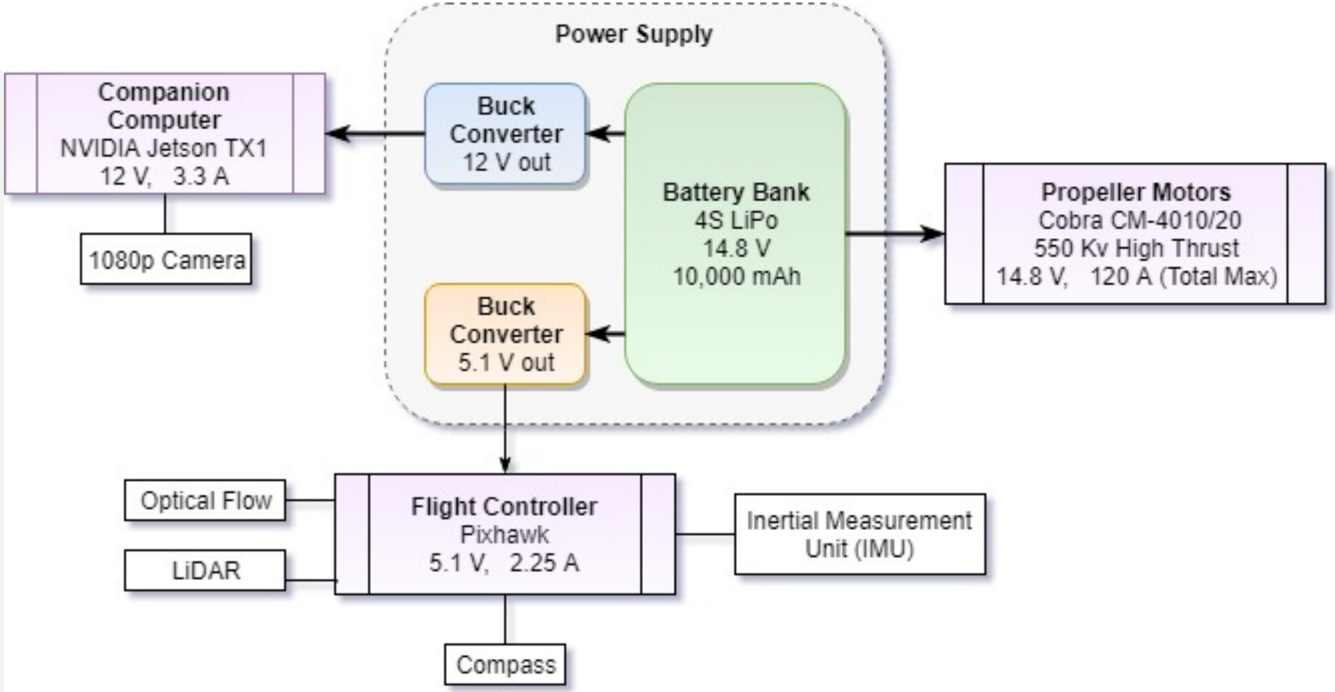
- Changing Requirements
- Lack of a reliable GPS signal near buildings and competition area
- Soldering connection breakage
- Mission Planner Weird Errors
- Wiring Issues (The motor connections)
- Flight Crashes
- Communication between Jetson TX1 and Pixhawk
- Image Processing Algorithm Issues
- Uploading / Downloading 26,000 Images
- Battery connection current fusing the battery together. (Last minute)
- TX1 wiping flight mode settings on competition day.
- Inability to test drone on campus (5 Million Dollar Fine)

TPS62135 Printed Circuit Board – Buck Converter

- We decided to create a printed circuit board to manage power across the internal embedded systems.
- We needed to power the Pixhawk with approximately 5 volts and up to 4 amps, because the extra power was required for our sensors like Lidar.
- The motors used to power our quadcopter have a battery independent of the PCB, to prevent system failure and allow for steady flight.
- The purpose of the printed circuit board is to facilitate varying voltage levels across the system. The printed circuit board (PCB) will also stabilize voltage spikes, ensuring that components don't get damaged.
- Our PCB uses a buck converter to allow for a wide range of voltages.



Sensor Collaboration



DroneKit

- Open-source software development framework which allows developers to get info about vehicle state, guide the vehicle to a specific destination or program a mission
- Developers can create object detection and path planning applications which require a lot of processing power
- Python scripts are loaded on to a companion computer such as a NVIDIA Jetson TX1
- TX1 and Pixhawk transfer messages using a communication protocol called MAVLink

Overview

- Ardupilot running on Pixhawk calibrates the machine, controls the motors, stabilizes flight, and transfers sensor data

- DroneKit running on the TX1 analyzes sensor and image data and changes the path of the drone accordingly

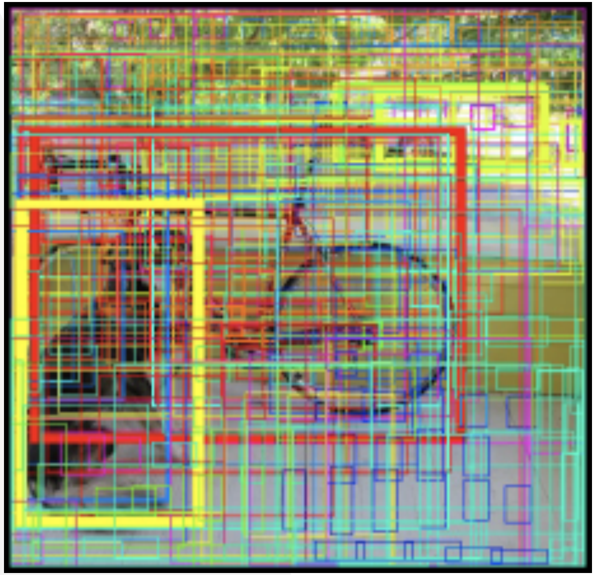
Object Detection/Tracking Software

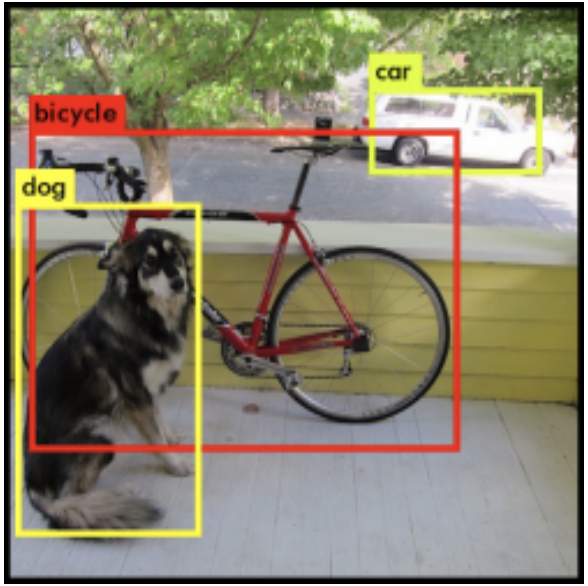
- **Tensorflow -** Google’s open-source library for machine learning and deep learning
  - Trains networks relatively fast and accurately
- **Keras -** user friendly neural network framework which runs on top of Tensorflow
  - Easier to use and understand compared to Tensorflow
  - Code is written in Python and can run on both a CPU and GPU
- **OpenCV** – computer vision library
  - We used it to show detected prey drones and calculate center points and pixel distances which helped in designing the flight algorithms
  - Helped to convert video to frames for object detection training

You Only Look Once (YOLO)

- Quick and precise object detection system
- Convolutional Neural Network (CNN) is applied to an image once
  - Image is split into several sections and network makes bounding box predictions over all sections
  - Bounding boxes with high probabilities are seen as successful detections
- YOLO runs faster compared to other object detection systems as they run the neural network several times across an image and may need many evaluations
- We used Tiny YOLO which has a CNN with less layers but is quicker compared to Full Yolo
- 20,000 images were annotated to be used to train the algorithm

You Only Look Once (YOLO)

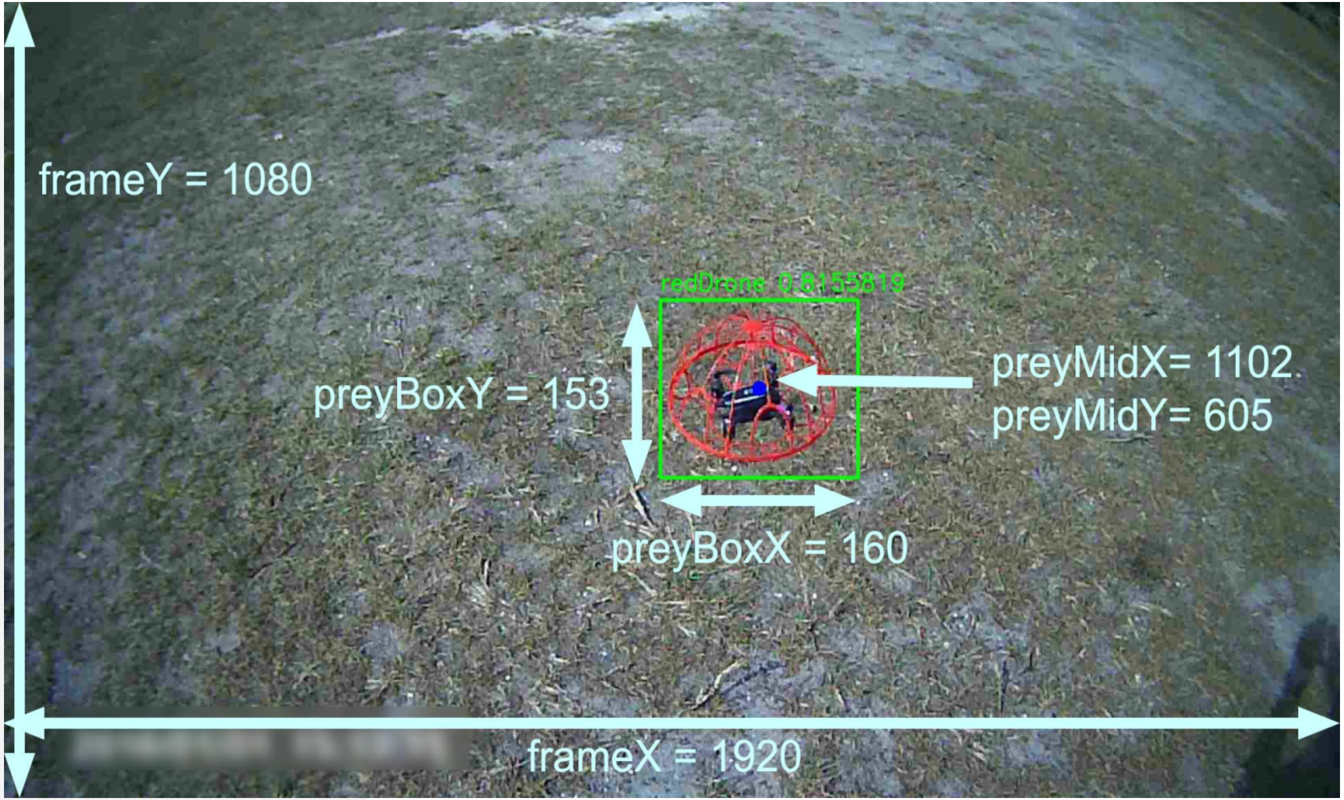




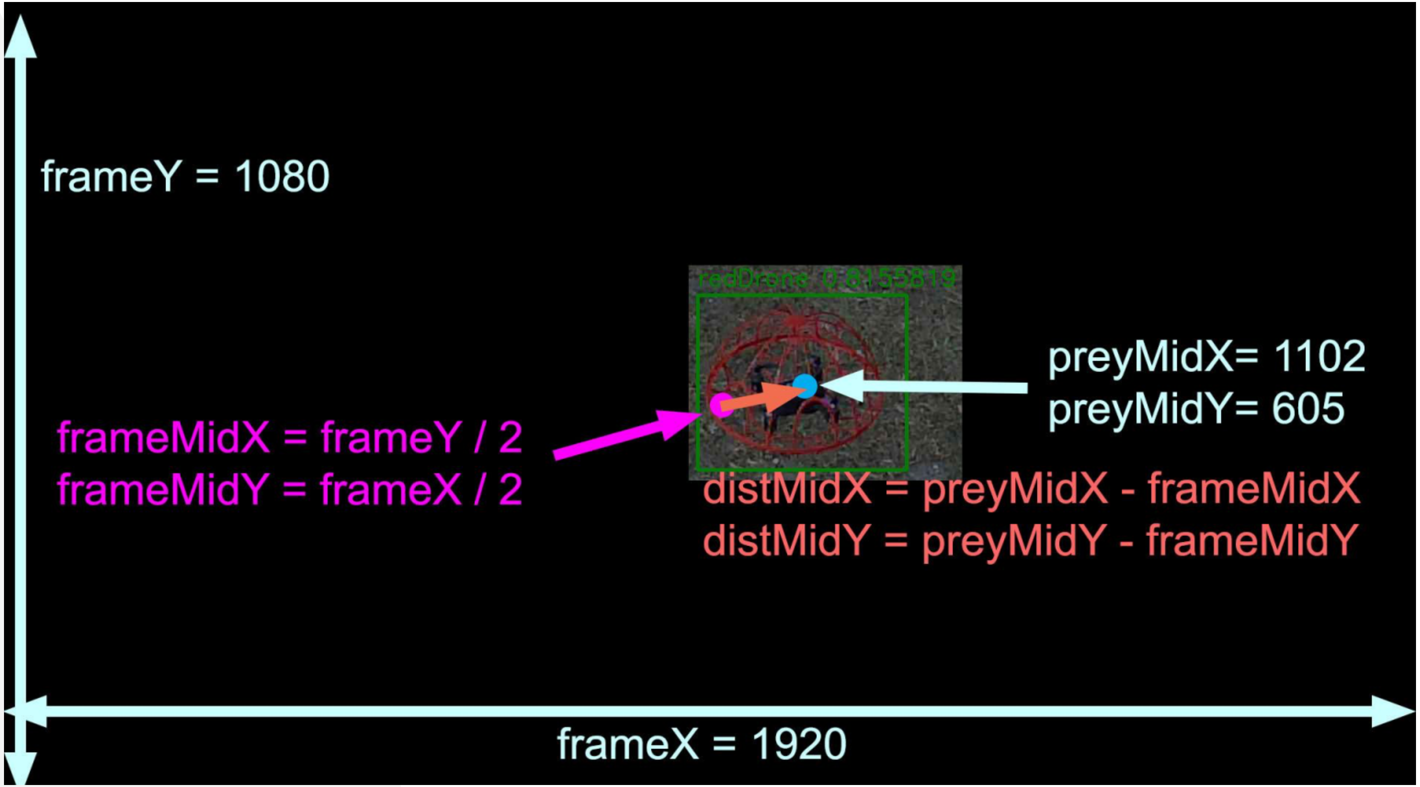
Autonomous Flight Software Strategies

- Two modes: ***Search*** Mode and ***Pursue*** Mode
- In *search* mode the quadcopter hovers in place and rotates 10 degrees/sec
- In *pursue* mode the quadcopter attempts to keep the prey drone in the center of the frame
  - It calculates the angle between itself and the prey drone and then yaws (rotates) to that angle
  - 3D Velocity vectors are used to direct the quadcopter towards the prey drone
    - ❑ Speeds are changed based on the relative distance between the center of the bounding box detected in the frame and the center of the frame itself
- If two prey drones are detected the quadcopter moves towards the drone with the larger bounding box as that implies that it is closer

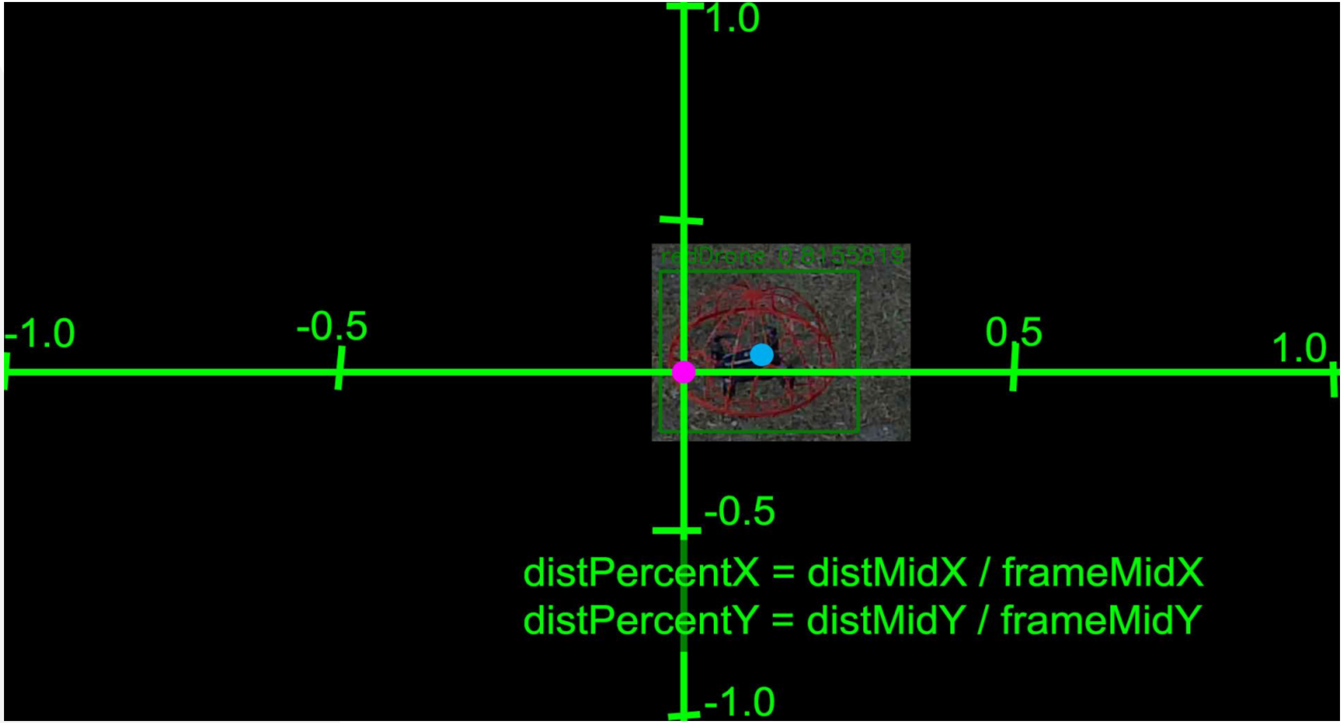
Pursuit Mode - Calculating Yaw and Vertical Velocity



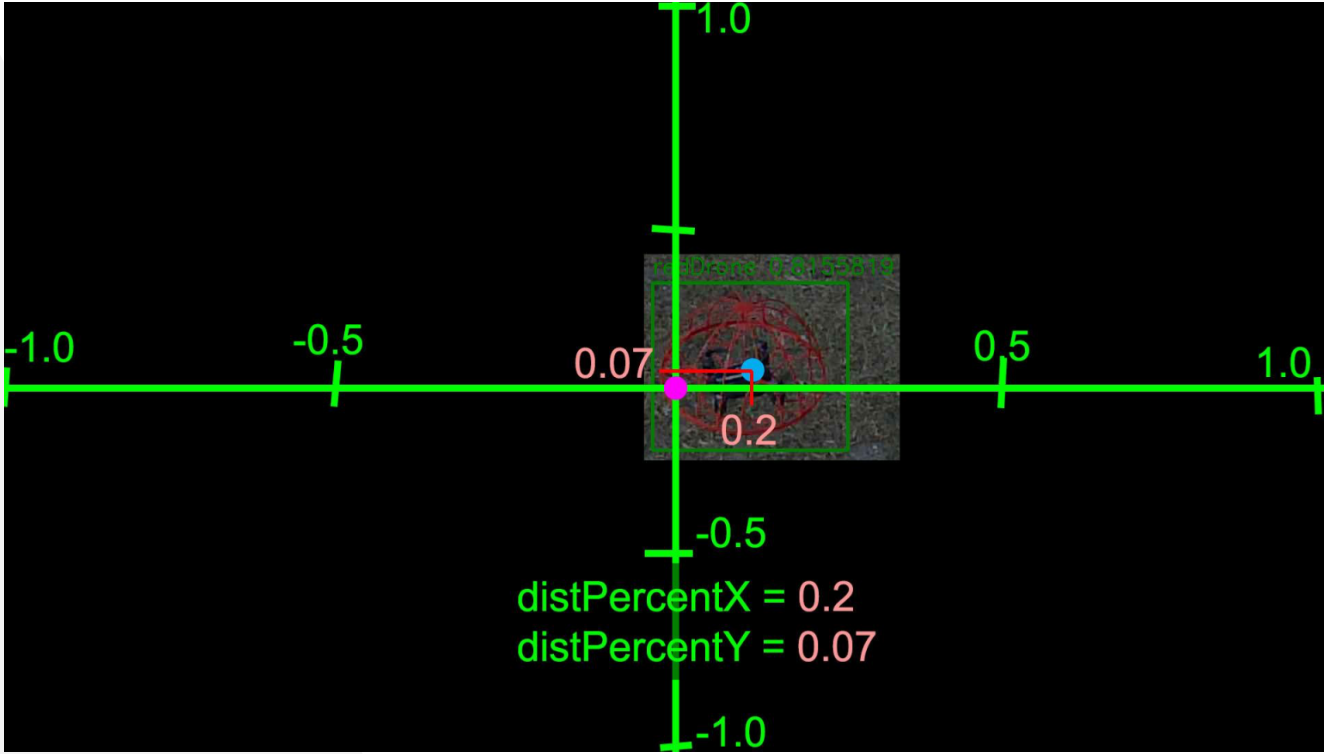
Pursuit Mode - Calculating Yaw and Vertical Velocity



Pursuit Mode - Calculating Yaw and Vertical Velocity

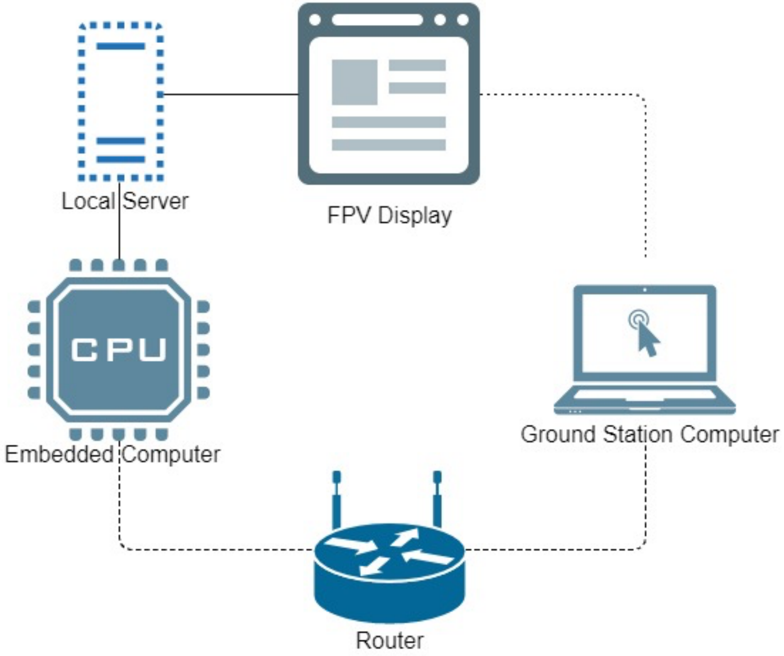


Pursuit Mode - Calculating Yaw and Vertical Velocity



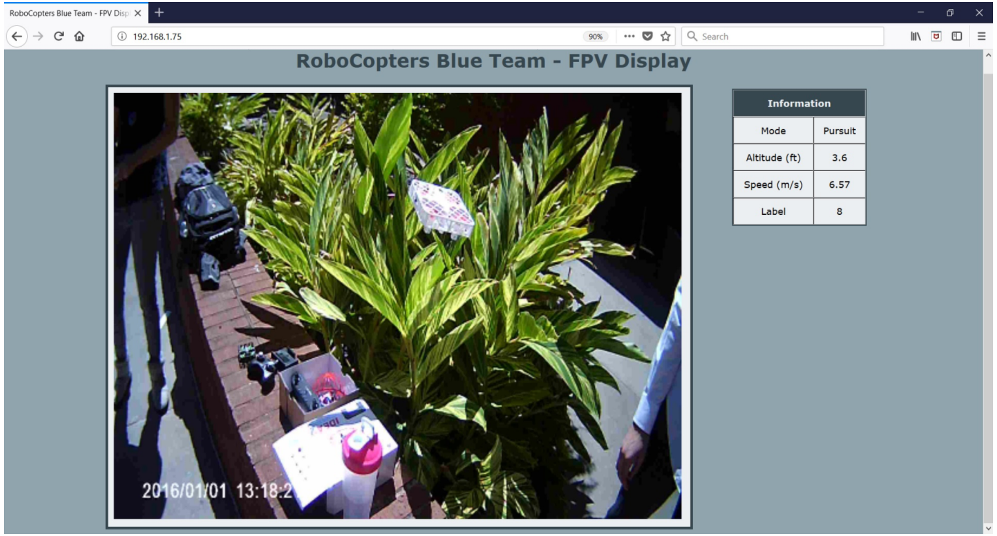
FPV - HTML Streaming Web Display

- Alternative to minimOSD display
- Display overlays of detected/pursued prey drones along with drone information from flight controller/telemetry
- Run local server on embedded computer with HTML website, accessed over local network from other computer

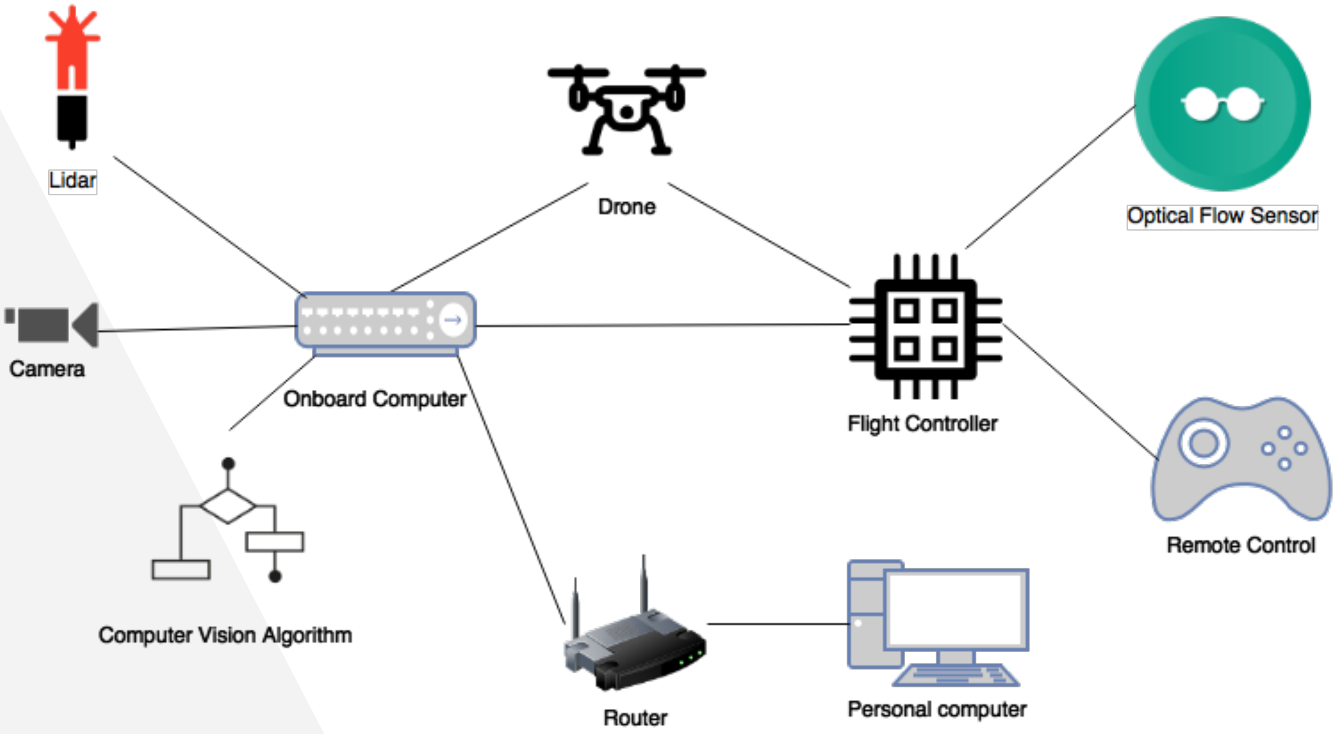


FPV Web Display Design

- HTML and CSS used to create and design Web Display site, Javascript used for updating video and information functionality
- Table displays information from flight controller/telemetry, updates every second
- Video/Image display updates around 12-14 frames per second

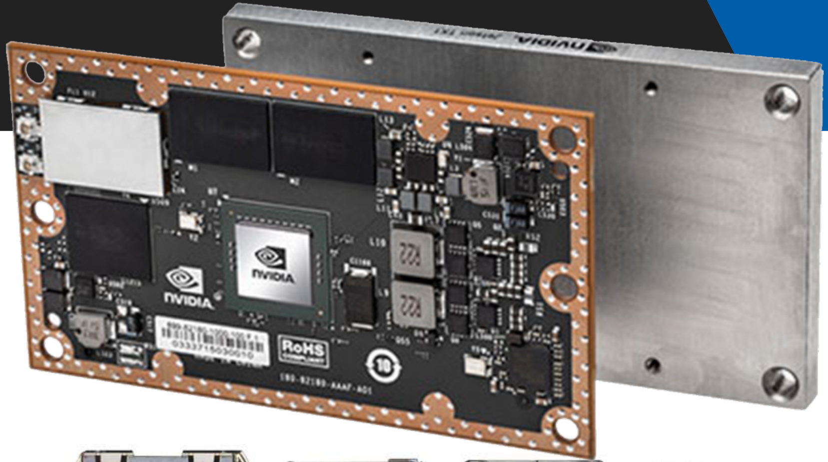


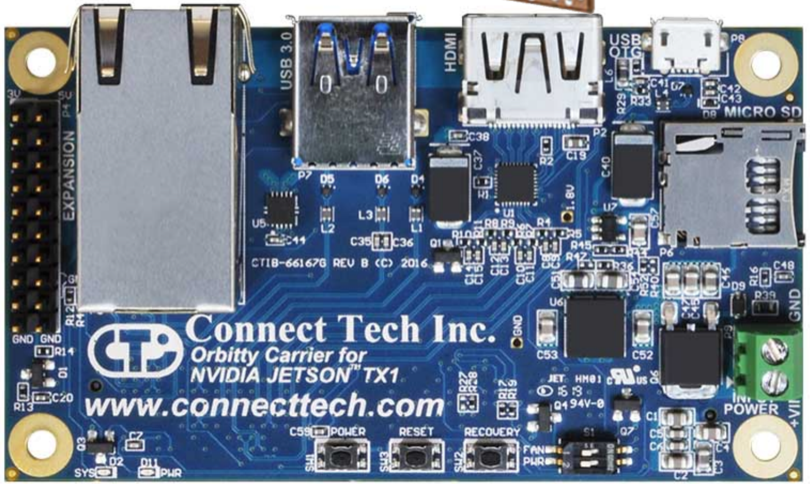
Project Block Diagram



Onboard Computer

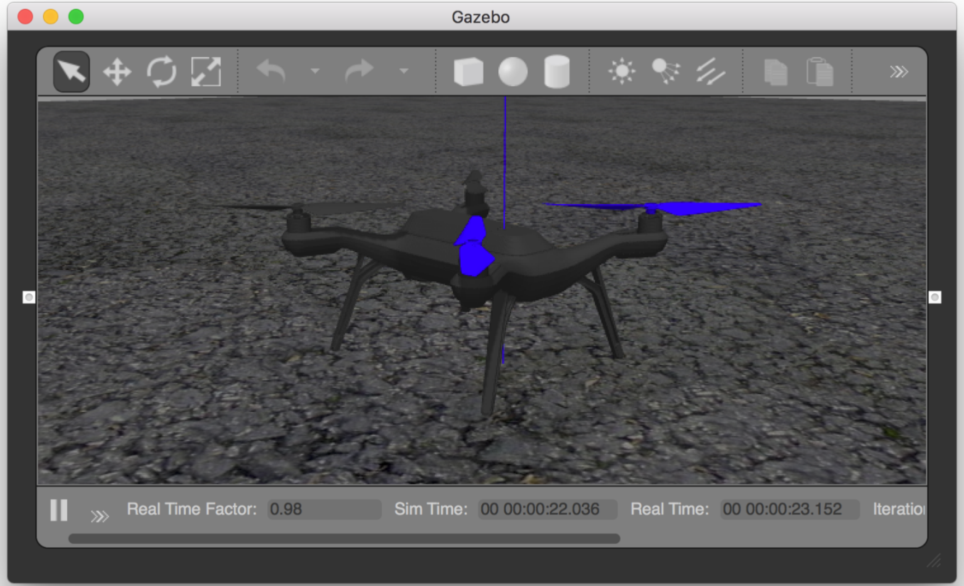
- Jetson TX1
- Carrier Board





Simulation for DroneKit Programming

- Gazebo
  - Less risk of damage
  - Better workflow
  - Ability to do multiple tests rapidly
  - Allow us to see how the drone actually responds to the environment.
  - Not loose our drone
  - Testing neural network confidence levels.

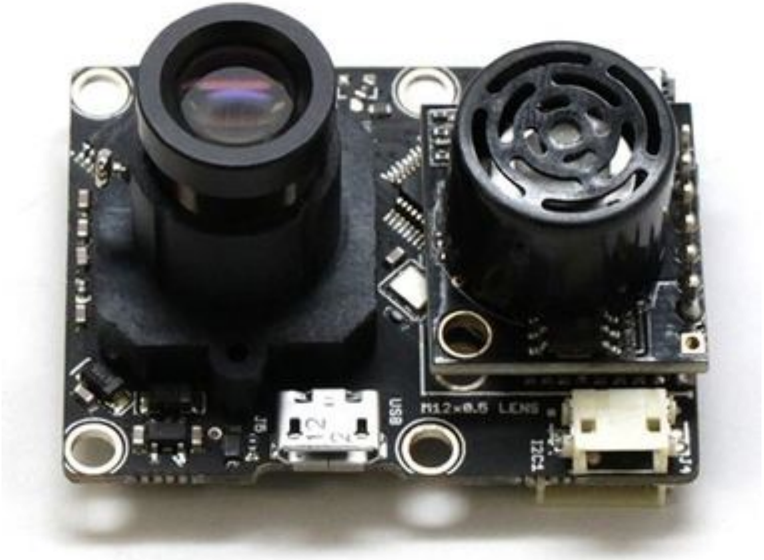


Sensors

- Lidar
- Optical Flow Sensor
- Camera
- Safety Switch
- Buck Converter Power Module
- Power Distribution Board
- ESCs and motors

Flight Controller

- PX4Flow
- 752x480 MT9V034 image sensor
- Works via relative motion of objects and surfaces
- 168 MHz Cortex M4F CPU
- L3GD20 3D Gyro
- 115mA / 5v



Range Finder (Lidar)

- Benewake TFmini lidar
- Small, low cost, low power consumption
- Necessary for use of optical flow sensor
- 100 measurements per second
- 0.3-12m
- UART protocol
- 0.12W average power consumption

Flight Controller

- Pixhawk
- Arm Cortex-M7
- 2MB memory
- 512KB RAM

Difficulties

Successes

- Active acoustic localization failed
- No experience in the field
- No one else has done it yet
- Getting gazebo to run properly
- No flight code written
- Budget ran out
- Lack of experience with GPU Microcontroller
- UCF Shipping Delays

- Drone components and layout done
- Know what we need to do to succeed
- Good communication
- Fantastic Team Spirit
- Hardware Improvisation (Gimbal)
- Motivation
- Prototype drone
- Embedded Software knowledge allowed us to grow as engineers

Acknowledgements

We would like to thank the following people for all of their support within this project:

- Samuel Richie, for being our wonderful advisor and giving us advice on hardware and software, and making sure we all stay on track as successful engineers.
- Lei Wei, for giving us the drive and motivation to stretch our project to new levels of difficulty.
- Nate Enos, for providing invaluable mentorship regarding PCBs and electronics learning.
- Leu Glaros, for giving us good advice and the power to continue without giving up hope. Lockheed Martin for providing us with funding.
- Marcella and Chip Cameron, for letting us use their house as our build space and feeding us.
- And all of you, for taking time out of your busy schedules to come see our senior design project!

Sources

- https://www.softwebsolutions.com/resources/tensorflow-googles-artificial-intelligence-system.html
- https://keras.io/
- http://machinethink.net/blog/object-detection-with-yolo/
- https://pjreddie.com/darknet/yolov2/
- http://python.dronekit.io/about/overview.html

Questions?