Design Packet

Raytheon Autonomous Vehicle Competition

Jadon Reichmuth, Max Youngson, Zach Reiber, Jake Hahn, Janani Ganesh, Maria Raju, Tadeh Shanazari, LaiLa Huang, Phillip Doan, Nathaniel Mendez

March 17th, 2024

Contents

1 Executive Summary

The Raytheon Autonomous Vehicle Competition (AVC) is an annual engineering event sponsored by Raytheon. The purpose of the AVC is to allow students to use their creativity, innovation, and problem-solving skills to research, develop, integrate, and test Unmanned Vehicle hardware and software components to solve existing real-world problems. This collaborative project is an opportunity for participants to work in an environment like industry professionals, practicing project management and problem-solving skills on open-ended problems.

The AVC challenges students with the integration of various technologies such as system integration advanced manufacturing, computer vision, cloud and edge processing, wireless technologies, command and control, system integration, chip design, autonomous and avoidance algorithms, artificial intelligence, imagery, and sensing, to name a few.

The AVC challenges includes two vehicles provided by each team: a Unmanned Air Vehicle (UAV) and Unmanned Ground Vehicle (UGV) which can be observed in Figure 1. A single team's UAV attempts to seek, identify, and deliver a water blast to all rival schools' UGV's while avoiding their own. For each challenge run, a single school's UAV operates against all school's UGVs. Each team has designed, developed, and implemented both a UGV and UAV. In all scenarios, the UAV and UGV are to behave autonomously.

Throughout this report, the final concepts for each subsystem are outlined. It should be noted that successful water detection and delivery contribute to the scoring, so the better detection and accuracy, the better the team performs against other schools. The UGV used off the shelf water senors for water detection, which can easily be integrated to play sound and lights when hit (part of the scoring). For the UAV, a DIY pressurized water gun was incorporated. Multiple nozzle diameters have been purchased to provide a range of exit velocities and flow rates, since the dynamic effects of rotor downwash on the UAV are currently unknown. All UGVs have a unique ArUco marker on top to allow for each identification. That said, to detect ArUco markers (UGVs), the UAV needs a visual detection system- a Raspberry Pi 5 with a PiCam 3 camera module. As for the autonomous requirement, the MAVSDK C++ library in combination with Ardupilot's Mission Planner allows for full customizability and straightforward implementation of search, move, and drop algorithms for the UAV.

Given the complexity of the AVC, it is imperative that risk be managed. Special attention has been placed on injury safety, with two members becoming FAA Certified. Every good engineering project runs into unforeseen problems, and time spent anticipating said problems could save hours if not days of troubleshooting. In line with this risk management, various attempted prototype plans are outlined, explaining why certain concepts were or were not worth pursuing. The $5000 budget constraint is also considered, with a detailed breakdown of costs to get both the UAV and UGV competition-ready.

UAV UGV

Figure 1: Final Built Vehicles

2 Introduction/Background

There are several emerging use cases that use Unmanned Air Vehicles (UAVs) to autonomously identify targets and deliver payloads to them. Package delivery is one of the most publicized use cases. This year's AVC requires the autonomous delivery of water from the UAV onto various (UGVs) belonging to different teams.

Both vehicles must adhere to the competition Rules of Engagement (RE), and the UAV must also comply with current Federal Aviation Administration (FAA) UAV regulations. The operation of UAVs falls under FAA Code of Federal Regulations which states an airborne UAV must have an FAA certified remote pilot in command [1]. Therefore two team members have completed FAA certification and passed the required exams. Without certification and compliance, our team cannot compete.

To autonomously navigate, the UAV needs at least a primary microprocessor, an accelerometer, a gyroscope, a magnetometer, and a barometer [2]. Without these basic sensors, autonomous flight can't happen. The data these sensors collect must be processed by an onboard microcontroller. Two options exist: building a custom board that would have a smaller form factor and save weight, or the quicker and cheaper option: purchasing one. In addition to this microcontroller, flight pathing software is necessary. Three software packages were found on the open market that we have integrated into our autonomous systems: Robot Operating System (ROS), MAVsdk (C++ library), and Matlab. Ardupilot is the most common manual interface for drone operation, and is used in tandem with the autonomous portion.

Also necessary for autonomous navigation is some form of locating technology. Global Positioning System (GPS) is the most widely used form of location technology around. Real-Time Kinematic (RTK) GPS allows for centimeter-level accuracy, which is significantly more precise than GPS, which offers meter-level accuracy. Raytheon's Rules of Engagement state the UGV must not vary more than ± 0.4m in the y direction when following paths, so that rules out GPS and leaves RTK GPS as a viable option. There are two ways to interface with RTK GPS systems: radio and Wifi. A pro of radio is accuracy, while a con is there can be lots of noise and signal interference. A pro of wifi is cost (significantly cheaper than radio), while a con is connection(may require a hotspot which is risky and may drop connected devices).

Originally created for augmented reality and robot localization, ArUco markers allow for camera pose detection [3]. An RGB camera can gather all the information needed to triangulate its position relative to the marker. This means only a camera is needed to find payload targets. That being said, Light Detection and Ranging (Lidar) sensors may still be important for crash prevention [4]. Both sensing options are explored and have their own tradeoffs, in the end we went along with a camera.

Given the relatively low budget and steep UAV requirements, the max payload capacity is an important factor to consider. Furthermore, the industry standard of a 2:1 thrust-to-weight ratio must be adhered to. The average price for prefabricated drones with a payload capacity of 3.5kg is 2500 dollars, while DIY kit drones on the market exist for just over 1000 dollars and can carry payloads of up to nearly 9kg [5][6][7][8][9]. All subsystems should be well within weight to ensure the 2:1 thrust:weight is respected.

Overall, many subsystems need to work together for a shot at competing, let alone winning the AVC.

3 Solution Overview

The four main engineering challenges of this project are water detection, water delivery, object detection, and autonomous mobility. Each of these challenges is tackled with a specific solution that integrates seamlessly into the overall system. Detailed descriptions of each subsystem will be provided in their respective sections, but an overview is presented here.

3.1 Water Detection

For water detection, the team has chosen an off-the-shelf water sensor due to its ease of use and robustness. Dropped water hits the top outermost plate of the UGV and be channeled into encircling gutters towards the lowest corner. A cup-like water sensing system catches and measures the collected water level to trigger alarm subsystems when certain conditions are met.

3.2 Water Delivery

Water delivery is achieved using an adapted beer growler with custom plumbing and an electronic release mechanism. This system is pressurized using small 8g CO2 canisters, allowing it to generate a steady stream of water downwards when activated by the GPIO output pins of the Raspberry Pi.

3.3 Object Detection

UGVs are be detected by their ArUco markers, using a camera attached to a Raspberry Pi running OpenCV. This system enables precise identification and tracking of targets.

3.4 Autonomous Mobility

Both the UGV and UAV completed autonomous missions with a Pixhawk flight controller running Ardupilot, in combination with an RTK GPS system. The UGV followed its waypoint missions using Ardupilot's built-in Mission Planner, while the UAV used use Mavsdk on a Raspberry Pi connected to the Pixhawk for its autonomous operations.

4 Description of the Design: UAV

4.1 Needs and Engineering Characteristics

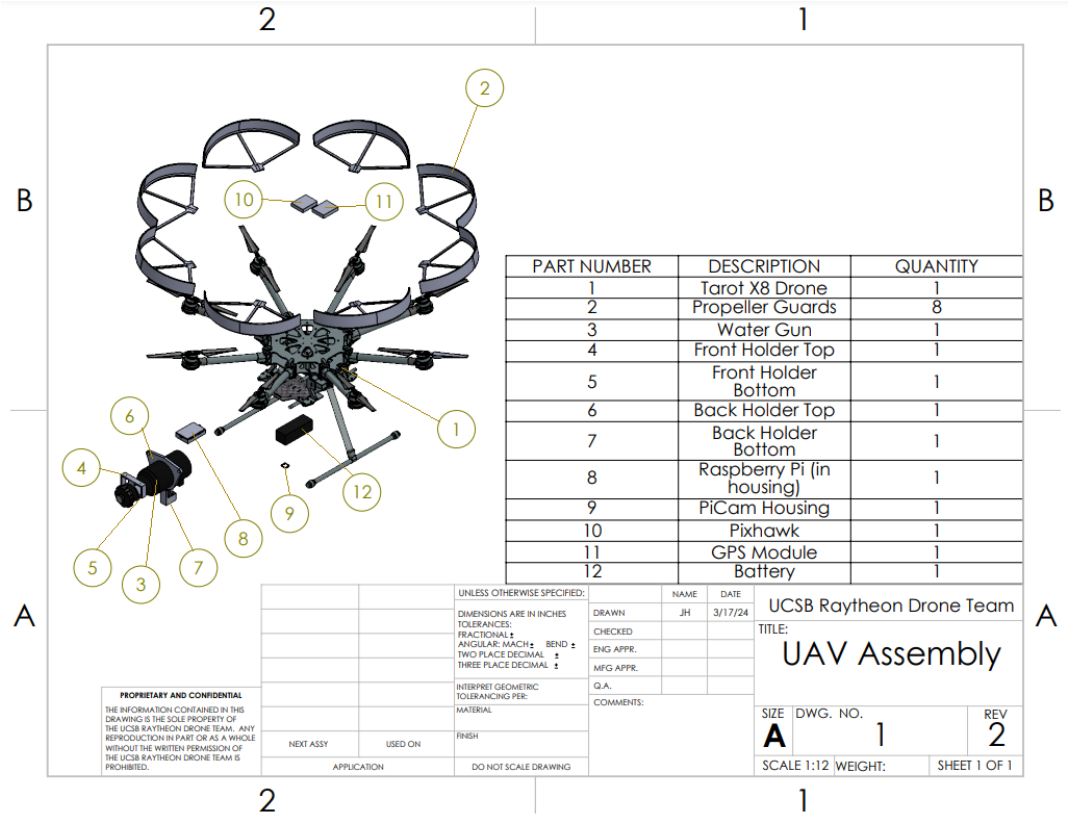
Needs	Engineering Characteristics	Target Spec (Min)	Target Spec (Ideal)
Detect Markers	Accuracy (%)	50(%)	100(%)
Water Delivery Volume	Volume per drop (ml)	20mL	80mL
Water Capacity	Volume(ml)	40mL	1800mL
Water Delivery Speed	speed(mph)	1mph	4.25mph
Payload Capacity	Mass(kg)	4kg	8.2kg
Repeatable Water Delivery	Accuracy(%)	50(%)	100(%)
Height Delivery Range	Meter(m)	1.3m	12.192m

Table 1: Target Specifications

Table 1 outlines the UAV's requirements and engineering characteristics. The primary function of the UAV is marker detection, as all other subsystems depend on the successful identification of ground vehicles marked with ArUco markers. Water cannot be delivered to a target if it cannot be autonomously identified. The minimum acceptable detection rate is 50%, though the ideal goal is 100%, ensuring markers are detected without fail. To understand how the detection rate was determined, the minimum water capacity must be explained. According to AVC guidelines, the UAV must deliver at least 20 mLof water to each of the 2 UGVs, resulting in a total minimum volume of 40 ml. The team decided to include a safety factor of 2, allowing for two shots per UGV. This means the UAV must carry at least 80 mLof water, ensuring a detection rate of 50%. The ideal volume per drop is 80 ml, calculated as the amount of water needed to cover an area with a 2-foot radius at a density of 20 mLper square foot. This allows the UAV to potentially hit a UGV indirectly (using a spray and pray method) and still deliver 20 mLwith the specified density. The finalized water capacity is set at 1800 ml, based on the size of the Growler. The minimum speed for water delivery is 1 mph; however, the UAV hovers over the UGV and deliver water at a speed of 4.25 mph. Similar to the detection rate, the UAV should maintain an accuracy of 50% when delivering water to avoid running out mid-challenge. The minimum payload capacity was determined by adding the weight of the minimum water capacity to the estimated weight of all other onboard systems, resulting in a requirement of 1 kg. The finalized payload capacity of the UAV is 8.2 kg. Lastly, the detection distance must be considered. The UAV can deliver water from the camera's maximum fly height of 12.192 m, with a minimum distance of 1.3 m to ensure it is not too close to the UGVs.

4.2 UAV Design

When considering the design parameters for the UAV, the following main engineering characteristics were taken into account: water delivery, object detection, and autonomous mobility. The key aspects prioritized in the selection of parts and design parameters were environmental suitability, functionality, and ease of user interface.



PART NUMBER	DESCRIPTION	QUANTITY
1	Tarot X8 Drone	1
2	Propeller Guards	8
3	Water Gun	1
4	Front Holder Top	1
5	Front Holder Bottom	1
6	Back Holder Top	1
7	Back Holder Bottom	1
8	Raspberry Pi (in housing)	1
9	PiCam Housing	1
10	Pixhawk	1
11	GPS Module	1
12	Battery	1

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY | USED ON
APPLICATION | DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	JH	3/17/24
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF THE UCSB RAYTHEON DRONE TEAM. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF THE UCSB RAYTHEON DRONE TEAM IS PROHIBITED.

UCSB Raytheon Drone Team
TITLE:
UAV Assembly
SIZE A | DWG. NO. 1 | REV 2
SCALE 1:12 | WEIGHT: | SHEET 1 OF 1

Figure 2: UAV Bill of Materials

Integration of the UAV centers around a Raspberry Pi single-board computer. This component serves as the interface, facilitating communication with the Pi Camera via OpenCV for ArUco marker detection, as well as interfacing with the Pixhawk flight controller and the water-dropping mechanism. Additionally, various peripherals are integrated with the flight controller to ensure optimal functionality. These peripherals include the Electronic Speed Controller (ESC) for motor control, as well as the gyroscope, barometer, and accelerometer for precise navigation and environmental sensing. The connection diagram visually represents these integrations:

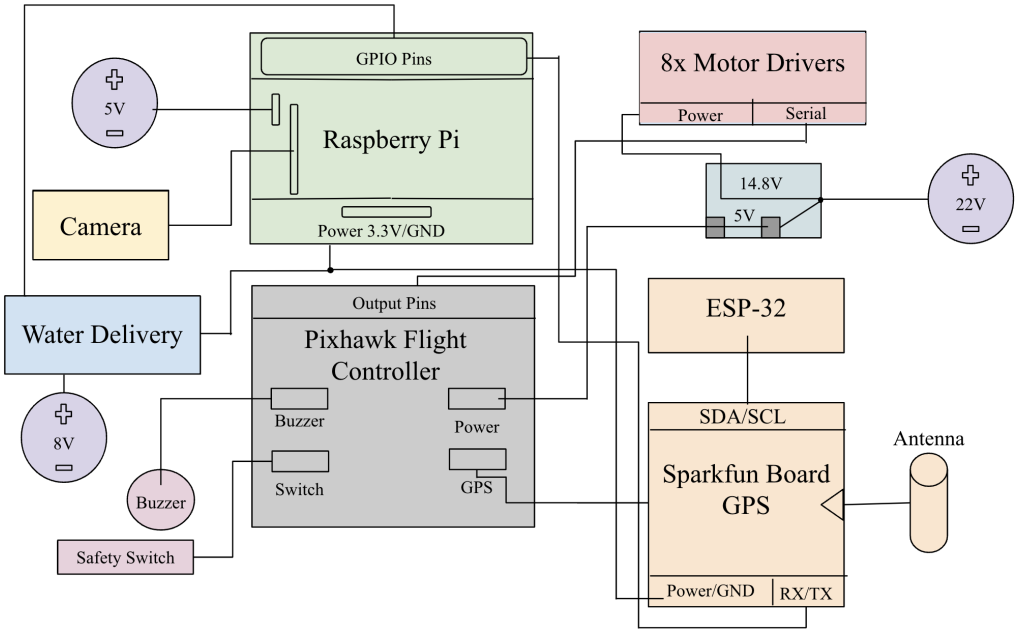


Figure 3: UAV Connections Diagram

4.2.1 Tarot X8 Drone

The Tarot X8 is an octocopter drone kit, equipped with eight rotors, propellers, and ESCs. Its body is constructed from carbon fiber, aluminum, and 3D printed ABS components to minimize weight and enhance strength. An internal power distribution board is attached to one of the two carbon fiber plates that form the core of the drone. The drone features eight aluminum arms housing the ESCs and motor mounts. Additionally, there are two carbon fiber bottom rails for attaching various components. The landing gear consists of two electronically actuated legs. Stable flight under varying payload conditions is crucial for the UAV's effectiveness. The Pixhawk module includes multiple PID controllers for roll, pitch, yaw, and elevation control. These tunable parameters ensure the drone can compensate for changes due to wind, payload, ground downwash, etc.

Figure 4: Tarot X8 Drone Kit

An octocopter has the highest number of rotors among commercially available drones. While only two rotors are needed for stable flight, as seen in helicopters, the additional rotors in an octocopter provide redundancy, reducing the thrust required from each rotor. This configuration is ideal for the competition, which involves extensive takeoffs and flights. The drone can remain stable even if one or more rotors fail. Prioritizing safety, our team chose this configuration for its redundancy. The primary reason for selecting this drone kit is its capacity to carry the projected payload while maintaining high speed. The industry standard for drones is a 2:1 thrust ratio, which is factored into our calculations.

Each motor can support a minimum of 1.77 kg and nominally 2.15 kg. With eight motors, the drone can support a minimum of 14.16 kg with a thrust ratio of 1.72:1 and nominally 17.20 kg with a thrust ratio of 2.1:1. During calibration flights, we must ensure the motors provide nominal thrust to maintain the 2:1 thrust ratio. The entire UAV and its subsystems are powered by a ZEEE 10000mAh 22.2V 6-cell battery. The Tarot X8 is equipped with a power distribution board that supplies power to the eight ESCs and three auxiliary XT60 power connections. One auxiliary connection is reserved for the Pixhawk module, one for the water gun circuit, and the third is unused. The water gun switching circuit uses a variable step-down transformer to supply 8.1V to the BJT.

Below is a table of our onboard payload and their masses:

Component	Mass (kg)
Tarot X8 + ESCs + Motors	3.9
Water Delivery System (12% capacity)	3.0
Marker Detection System	0.065
Battery (10000mAh)	1.15
GPS Module	0.05
Autonomy System	0.02
Total	$\tilde{8}.2$

Table 2: Weights of each onboard system.

Each motor can support 1.77 kg of weight at the minimum and 2.15 kg nominally. So with 8 motors, we can support 14.16 kg at the minimum with a thrust ratio of 1.72 and 17.20 kg nominally with a thrust ratio of 2.1:1. This means that we must ensure that the motors supply the nominal thrust during our calibration flights to preserve the 2:1 thrust ratio.

The entire UAV and subsystems are powered by a ZEEE 10000mah 22.2V 6-cell battery. The Tarot X8 drone is fitted with a power distribution board to supply power to the 8 ESCs and three auxiliary XT60 power connections. One auxiliary power connection is reserved for the Pixhawk module, one is for the water gun circuit, and the other is unused. The water gun switching circuit uses a variable step-down transformer to supply 8.1 V to the BJT.

4.3 Water Delivery Design

The water delivery system is designed in a gun-based format, akin to a Super Soaker. A central water tank stores all the water, with the air inside pressurized using CO2 canisters. This system is mounted to two rails and the bottom plate of the drone using two mounts: the back holder and the front holder. The back holder is a two-piece assembly with dovetails for a press-fit and glued connection. The bottom section of the back holder positions the water gun nozzle straight down to minimize the distance the water stream travels. The front holder is also a two-piece assembly, secured together with two nuts and bolts, and mounted to the bottom plate with additional nuts and bolts.

The water tank represents the heaviest payload the drone must carry, making it essential to place it near the drone's center of gravity to avoid excessive inertia and substantial torque from the water's weight.

The entire water gun is mounted beneath the drone using custom-made 3D printed PLA plastic pieces. Each end of the water gun is secured by a holder to restrict all six degrees of freedom. The back holder consists of two dovetailed pieces designed for a sliding fit, bonded together using cyanoacrylate glue ("super glue"). The top piece slides onto the bottom rails of the drone, while the bottom piece snugly holds the gun nozzle to ensure a level firing stream.

Figure 5: Water Gun Mount Assembly

The front holder is a two-piece assembly fastened with M2-0.4x8mm steel pan head corrosion-resistant Phillips screws and matching M2-0.4 hex nuts. Additional screws and nuts secure the front holder to the drone's bottom plate. Below are 3D models of the front holder, back holder, and the bottom plate:

Figure 6: Front Holder

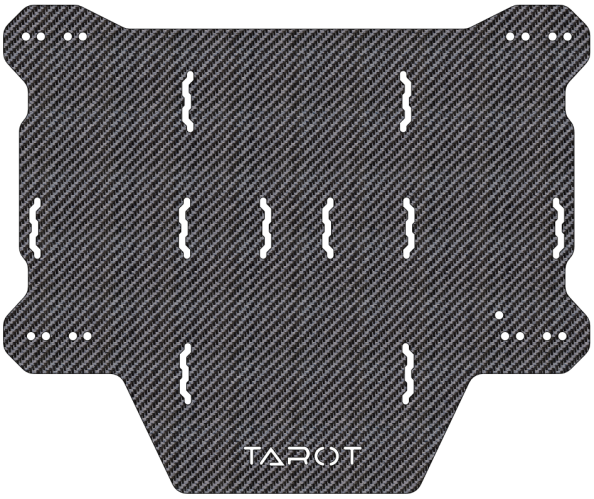


Figure 7: Carbon fiber bottom plate with mounting holes

Figure 8: Back Holder, Exploded View

4.3.1 Water Delivery and Storage

The water delivery system consists of a CO2-pressurized water blaster actuated with a solenoid valve. The solenoid valve is to be actuated by the Raspberry Pi when it detects an ArUco marker. A solenoid was chosen because it is electronically controlled and all moving parts are contained inside of it. Water is stored in the water tank with a relief valve set to activate at about 30 psi so as to avoid pressurizing the tank past its rated limit. A hand-operated valve allows pressure to escape from the CO2 cartridges placed in the tank lid. Barbed fittings are used to connect the tube to the solenoid and water tank. A 1/4 NPT nozzle is fitted to the end of the solenoid to supply a solid fluid spray. The capacity of the water tank is about 64 oz. (1892.71 ml). We expect to be at about 10 percent water capacity when competing (2 enemy UGVs, 40 mLper UGV, FOS of 3.0). The total weight of the water gun on competition day was about 0.7kg.

Below is a table of the materials used in the construction of the water gun system.

Component	Vendor	Rated Pressure
Craft Master 64oz CO2 Pressurized Growler (with pressure meter, relief valve, and CO2 adapter)	Amazon	30 psi
5/16” ID Clear PVC Tubing	Amazon	30 psi
1/4” NPT 12V Solenoid Valve	Amazon	100 psi
2x 1/4” NPT Barbed Fittings	UCSB	100 psi
CO2 Canisters	Amazon	generates 900 psi
0.203” Solid-Stream Spray Nozzle	McMaster-Carr	100 psi
Custom BJT switching circuit	[custom-made]	N/A
AITRIP Mini MP1584EN DC-DC Buck Converter Adjustable Power Supply Module 24V	Amazon	N/A

Table 3

Below is a flow diagram of the water gun with all relevant components called out with the proper symbols.

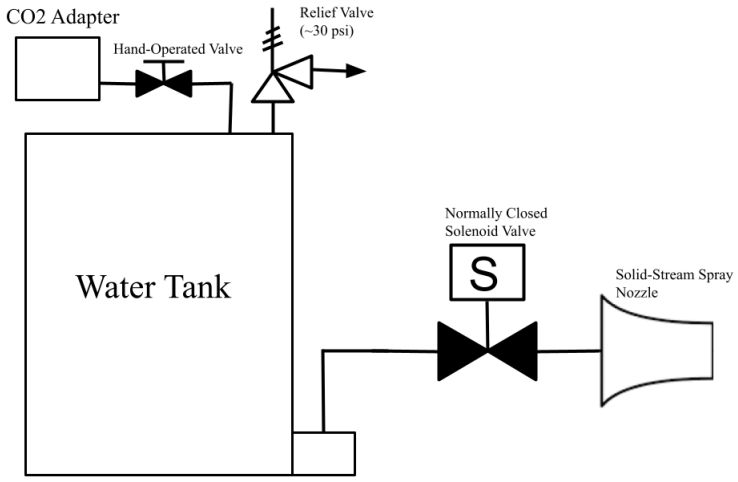


Figure 9: Flow diagram of water gun system.

A circuit was built to step-down the 22.2V supplied by the battery to below the 12V and 6.5W the solenoid is rated for. This power is then fed into a switching circuit, whose input comes from the GPIO pins on the Raspberry Pi which supply 3.3V to the BJT to activate the solenoid valve. The switching circuit is controlled by the voltage input from the Raspberry Pi into the base of the BJT. The BJT in this case essentially acts like a switch. When a high voltage (3.3V) is applied to the base, the switch is closed, and the current flows from the collector to the emitter of the transistor. This, in turn, opens the solenoid valve, allowing water flow. When the input voltage to

the base turns low, the switch is open, and no current flows through, thereby closing the solenoid valve. The Schottky diode, placed in parallel with the solenoid valve, protects the valve from the inductive voltage spikes that occur when the solenoid releases energy. The purpose of the circuit is to control the water delivery on the ground vehicles only when detected. Below is a schematic of this whole circuit:

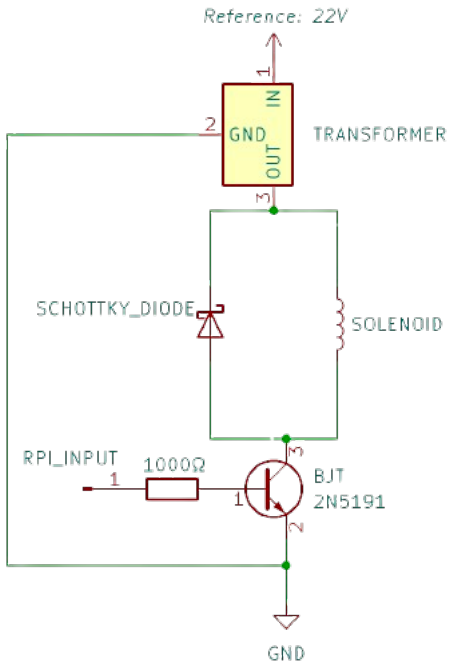


Figure 10: Switching Circuit Schematic

4.4 Object Detection Design

The main components of the object detection mechanism are a Raspberry Pi camera and its 3D-Printed housing. The key aspects considered for the UAV design include: the location of the Pi-camera relative to the water delivery mechanism, the orientation of the Pi-camera relative to the ground, and the location intent of electronic components to avoid interference with flight actuation and marker detection. Moreover, the orientation of the Pi-camera is placed directly towards the ground for a wide field of view.

The main C++ libraries that were used in the object detection algorithm were libgpiod, GeographicLib, MAVSDK and OpenCv. The libgpiod library helps interface with the kernel to turn on gpio pins, GeographicLib is used for calculating gps locations precisely using WGS84 earth approximation, MAVSDK which sends commands to the Pixhawk over MAVLink (a serial communication protocol) and OpenCV is used for marker detection.

Testing with the Picam V3 yielded promising results, a maximum detection distance of 65 feet without movement, and a maximum distance of 25 feet when moving at 4.8 miles per hour (with an 85 percent detection rate). This was tested by recording the time it took to move 15 feet while keeping the camera pointed at the marker. The limiting factor in these tests is glare, large amounts of glare blur the edges of the marker which results in OpenCV rejecting the marker entirely. In order to reduce this glare we are utilizing a polarized lens which, when at the correct angle, removes the sun glare entirely.

The case of the Raspberry Pi camera was a cheap off-the-shelf ABS case that was bought with time-saving considerations. It encloses the camera via a snap-fit mechanism and features an extruded cut out for visibility of the camera lens. Figure 11 shows the product that was used.

The Raspberry Pi camera is placed next to the outlet nozzle tube of the back holder. Doing so, allows for less calibration and distortion between the object detection and water delivery subsystems and thus better integration. In order to achieve this, the design choice was to use sticky back velcro. This was placed on the bottom base of the horizontal-lying water growler mount. Making the Pi camera detachable allowed for simpler testing means and ease of access when testing the integrated drone, as compared to testing individual code for the camera.

Figure 11: Snap Fit Pi-Cam Case

4.5 Autonomous Mobility Design

We installed PX4 firmware on the Pixhawk for autonomous mobility on our UAV. This firmware was selected for its compatibility with MAVSDK libraries in C++, which provide a high-level API for communicating with MAVLink-compatible drones. PX4 and MAVSDK communicate through the MAVLink protocol, a lightweight messaging system designed for drones and ground control stations. MAVSDK connects to the PX4 autopilot using communication links like UDP or serial. Once connected, they exchange MAVLink messages: PX4 sends telemetry data (such as position, velocity, and battery status) to MAVSDK.

Figure 12: MAVSDK

In addition, MAVSDK sends control commands (such as arm, takeoff, and set position) to PX4. MAVSDK offers high-level API functions that handle these MAVLink messages internally, making drone control easier. For example, using the command drone.action.arm() in MAVSDK sends an arming command to PX4. MAVSDK allowed us to develop our search algorithm by allowing the drone to scan the field in a vertical and horizontal direction as shown in the flight path below. Once it detects the UGV in that path, it flies toward it and drops down close to the UGV to shoot water. Then it continues scanning for the next one until it lands.

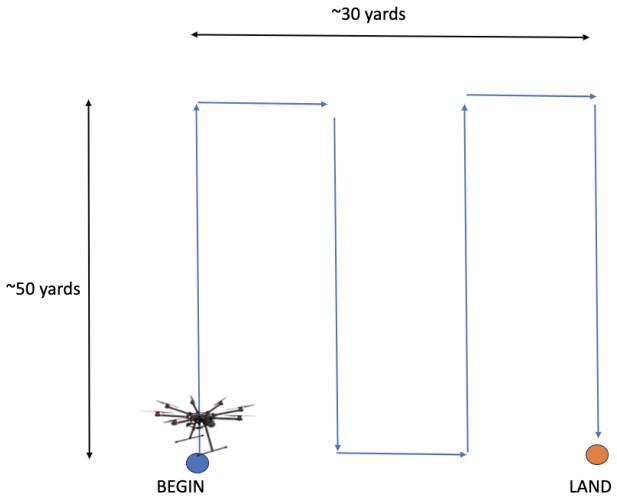


Figure 13: Flight Path

The search algorithm starts by taking the corners of the field and converting them into GPS coordinates, which the drone then moves to. The OpenCV thread runs continuously, searching for Aruco markers using specific functions. If a marker is found and it's not the intended one, a notification is sent to the MAVSDK thread using condition variables in C++. In the search state, the MAVSDK thread waits for this notification, and if it doesn't receive it within about 5 seconds, it commands the drone to move to the next part of the search algorithm, then resumes waiting

for the notification. Once the notification is received, a mutex is taken to prevent the OpenCV thread from accessing shared global variables simultaneously. The OpenCV thread calculates the GPS location of the marker using the current drone GPS position and the translation vector. This GPS location is stored in a shared hashmap, and upon notification, MAVSDK retrieves the GPS location and directs the drone there, bringing it within approximately 1 meter of the UGV. The OpenCV thread continues to calculate these vectors while MAVSDK directs the drone to each new GPS location. Eventually, the drone positions itself directly over the UGV, descends, shoots, and then resumes the search.

For autonomous mobility on the Unmanned Ground Vehicle (UGV), ArduPilot is utilized for navigation and control. Using waypoint missions, operators can define routes for UGVs to follow autonomously. This is achieved through the Mission Planner interface, as illustrated in the figure below.

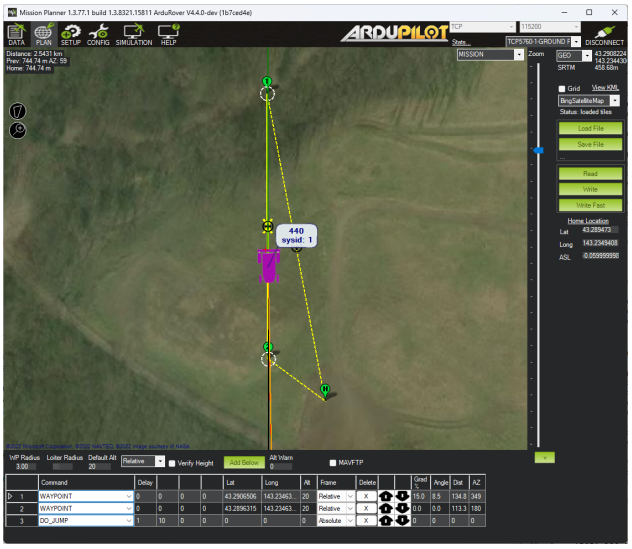


Figure 14: Flight Path

The Mission Planner software allows users to set multiple waypoints that the UGV sequentially navigates. Each waypoint is defined by its latitude, longitude, and altitude coordinates, along with specific actions the vehicle should take upon reaching each waypoint. These actions can include stopping, changing speed, or executing custom scripts.

Given that our project plan focuses on completing Challenge Two, only a controlled straight-line path is required. This stipulation simplifies our waypoint mission setup, as we only need to define a start and end point for the UGV to follow a straight trajectory. This approach ensures we meet the challenge requirements without the complexity of more intricate navigation tasks.

4.5.1 OpenCV Integration

The Raspberry Pi utilizes the OpenCV Aruco library which contains an Aruco detector class with several useful member functions, mainly the detectMarkers function, and the estimatePose function. The detectMarkers function scans the video feed for any markers that fit your parameters, and returns the pixel location and marker ID. The estimatePose function takes in a pre-calculated distortion coefficient matrix and your marker size; this function returns vector with the distance from your camera to the marker.

4.5.2 UAV State Machine

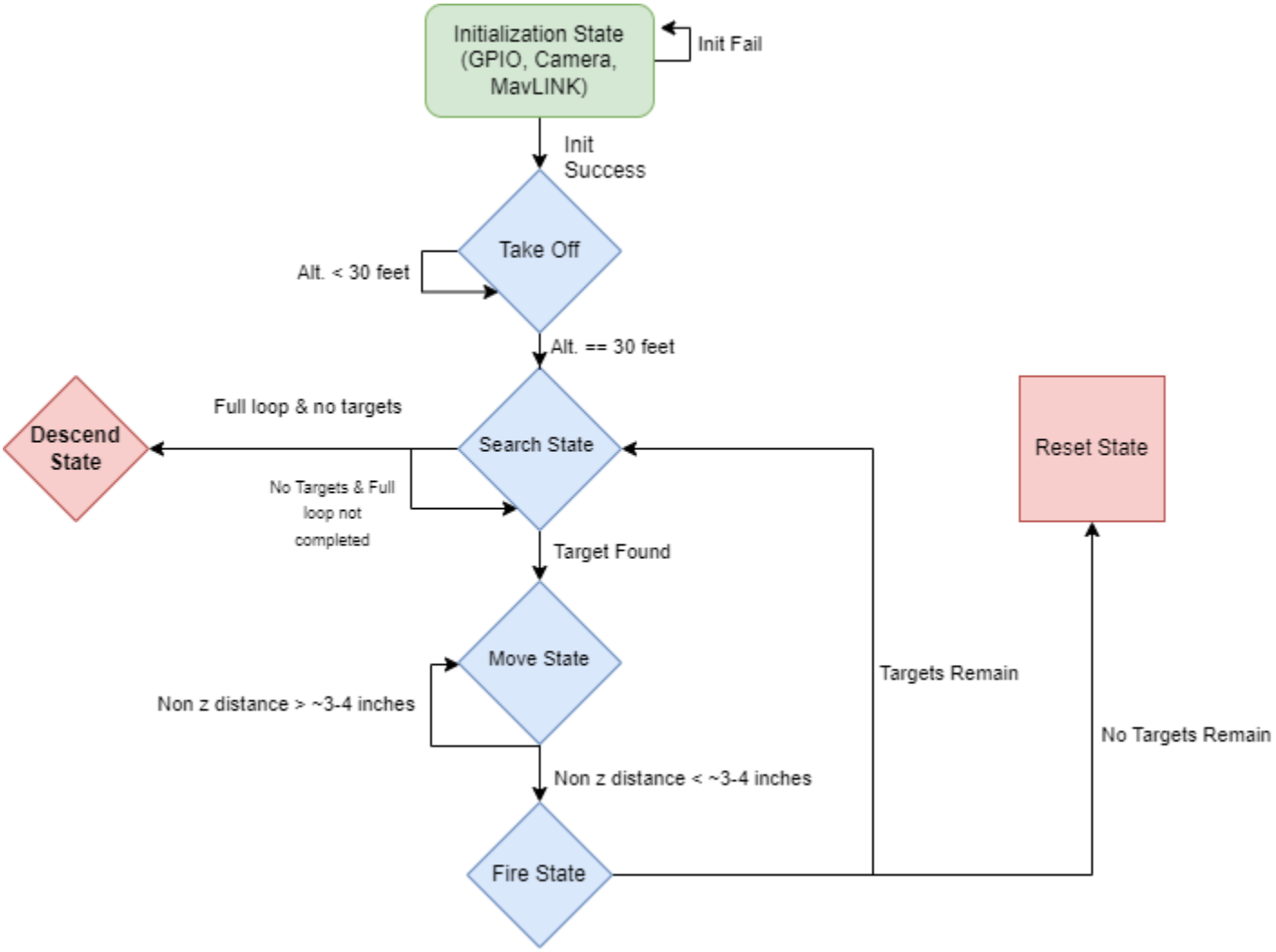


Figure 15: UAV State Machine

Autonomy is achieved through a super-loop state machine. The functionality of each state is described below.

- Initialization:
  This state initializes GPIO ports, checks that the camera feed is live, and uses MavSDK to establish a connection link to the Pixhawk. If all initialization functions complete succesfully, the state machine transitions to the Take Off state. If one or more of the initialization functions fails, they are re-attempted.

- Take Off:
  This state utilizes the MavSDK local ascend function to rise 30 feet above the ground. If the drone has succesfully ascended to 30 feet above the ground, the state machine transitions to the Search State.

- Search State:
  The Search State utilizes the waypoint functionality of MavSDK. The drone flies around the field in a set search pattern whilst the OpenCV marker detection function scans the camera feed. If other teams' Aruco markers are for whatever reason difficult to detect, the drone is programmed to descend if no targets are found in a full search pattern loop. Once a target has been detected, the search pattern is interrupted, and the state machine transitions to the Move State.

- Descend State:
  State that simply descends a set distance if no targets are found within one full search pattern. Returns to Search State.

- Move State:
  The Move State uses the pose estimation function from OpenCV to obtain a translation vector from our camera to the Aruco marker. This vector is fed to the MavSDK local move function, moving our drone towards the target UGV. Once we are at an acceptable distance we move to the Fire State.

- Fire State:
  This state simply turns on the GPIO port that is connected to our switching circuit. When the GPIO port is activated, the switching circuit turns on, allowing power from the battery to flow into the solenoid that activates our water gun. If the second phase of the competition is happening, the Move State and Fire State are combined, meaning pose vectors are constantly fed to MavSDK whilst the water gun is firing, this ensures that we can properly dispense enough water to a moving target.

- Reset State:
  This state is entered after all targets have had water deployed on them, or if a manual reset is ever issued. This state flies the drone back to a pre-set GPS waypoint and lands.

4.5.3 UAV Test Flights

Before testing the UAV flight, there are multiple steps that must be verified. The motors must be in the correct orientation according to an Ardupilot chart. The GPS and compasses must be calibrated. The transmitter and receiver must be connected with each other, and the Pixhawk's failsafes deactivated, that being the safety switch, low battery voltage detection, and the remote ON button.

On Saturday, March 16th at 1:00pm, the first initial tuning flight was run. This test was run in STABILIZE mode, meaning the pilot has full control over the drone controls. No oscillations were detected in the roll, pitch, and yaw. However, significant horizontal drift was detected with little to no wind towards the aft direction.

In the following months up until the competition, more than 50 flight tests were conducted. The PID tuning of the octocopter drone utilized the autotune feature of the flight software Ardupilot. Although this feature enables automatic tuning for the respective controllers, it was a hassle due to our limitations on battery life at the time. It was through this process that our team collectively decided to buy more batteries and implement more safety limits for prolonged battery life. Note that most if not all the tuning was done just with the drone frame and its sub-components (e.g. Water Growler not attached). After attaching the subsystem components, we qualitatively observed that the correctional stabilization values (found from the tuning) demonstrated sufficient inertial stability even with the extra weight. After the PID calibrations were complete, the autonomous flight functions were tested, and the speed settings were calibrated through Mission Planner. However, once the search algorithm was implemented, we had to switch from Missionplanner to Q Ground Control for PX4 firmware on the Pixhawk, since the Ardupilot firmware was not compatible with the MAVSDK Search Program. The search algorithm was then tested in conjunction with the Open CV water detection thread to ensure that all UAV functionalities worked together. After numerous tests, the drone could detect and fly reliably within +/- 1 meter of the ArUco marker with more than 50 percent certainty. Final flight tests were conducted at SBCC a day before the competition. Final flight results are discussed in later sections.

5 Description of the Design: UGV

5.1 Needs and Engineering Characteristics

Needs	Engineering Characteristics	Target Spec (Min)	Target Spec (Ideal)
Long Battery Life	Time (min)	30 min	300 min
Water Resistance	Volume (mL)	300 mL	3000 mL
Sensitive to Water	Volume (mL)	20 mL	1 mL
Play a Sound When Hit	Volume (decibels)	60 decibels	70-80 decibels
Flash a Light When hit	Brightness (lumens)	100 lumens	1300 lumens
Meet Required Marker Size	Area (ft^2)	16 x 16 in^2	16 x 16 in^2
Meet Speed Requirements	Speed (mph)	0.17 mph	0.49 mph
Obstacle Detection Distance	Distance (cm)	30 cm	100 cm
Vehicle Size	Area (ft^2)	less than 1 x 1 ft^2	less than 1 x 1 ft^2

Table 4: Target Specifications

The needs listed above are in order of importance to the project. The first 7 are crucial to this project since they are all involved in the scoring process of the competition. The next 2 are for optimized performance.

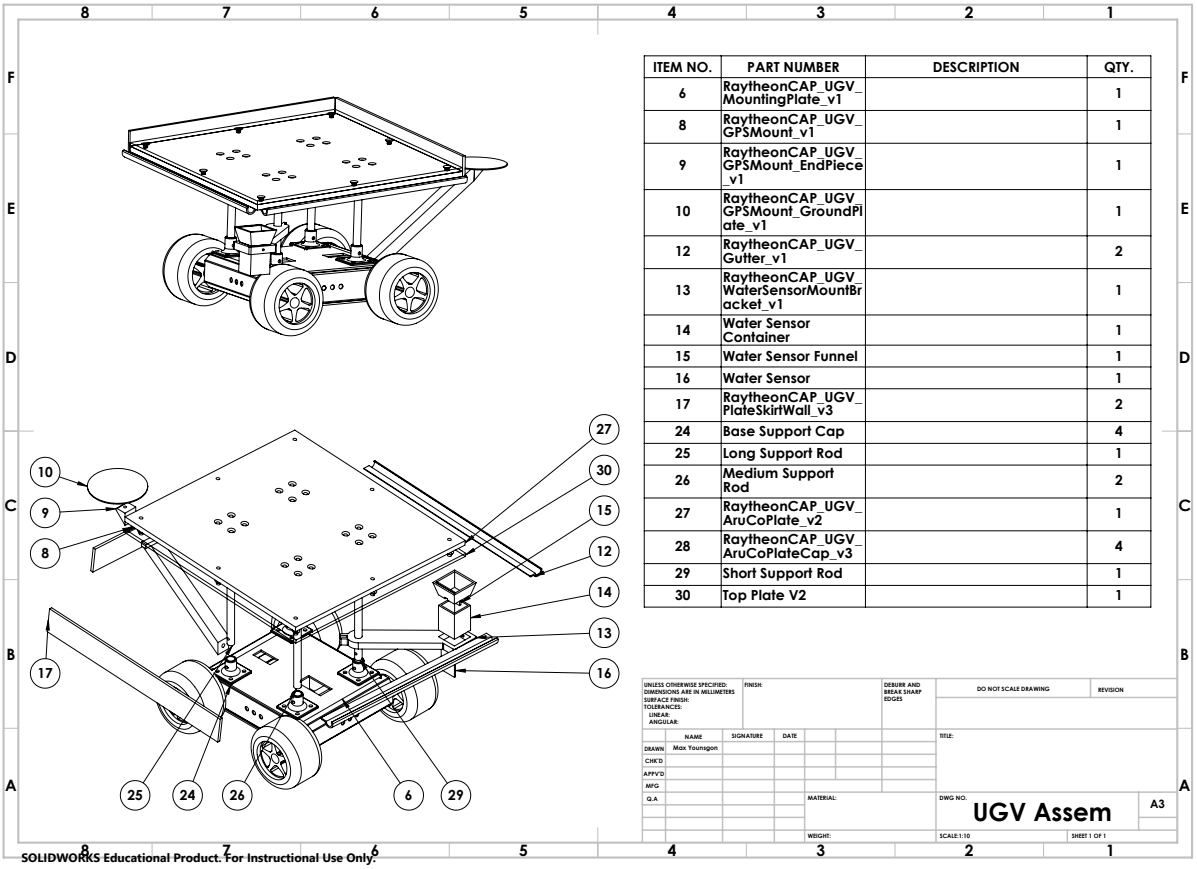
A minimum battery life of 30 mins (which is the duration of each challenge for the UGVs) is necessary. An ideal battery life of 300 mins is the goal, since there are 5 total challenges with breaks in between. The UGV must also be water resistant since it could have been splashed with water. A minimum resistance to a value of 300mL is necessary since that is the amount of water that may get on the UGV, assuming every UAV hits it once throughout the 5 challenges. Ideally, a splash proof UGV would be most beneficial, which is why the large resistance value of 3000mL was chosen. According to the Rule of Engagement, the ArUco marker must be sensitive to at least 20mL of water, but since points are given for detecting water, ideally, if even 1mL can be detected, the most points are earned. Capacitive sensors can be used to achieve this since they are very sensitive to water. The UGV must also play a sound and flash a light once water has been detected. A value of 60 decibels (approximately the volume of a human's speaking voice) is the minimum value that can be heard effectively. Ideally, a value closer to 65-120 decibels (the loudness of a fire alarm) is better since the UGV was on a field 50 yards away from all of the observers [16]. The brightness of an Iphone flashlight is 50 lumens, and it is slightly visible in the daylight [18]. The light should be at least double that to ensure that it can be seen, ideally closer to 1000-3000 lumens (brightness of emergency vehicles) to be seen from farther away [17]. The maker size must be 1ft by 1ft, which is another requirement listed in the rules. The minimum and ideal speeds are 0.17 mph and 0.49 mph respectively. These specific values come from the Rule of Engagement document requirements for each challenge.

The last two needs from the table that were not mentioned previously would help improve the performance, but are not mandatory for scoring purposes. The minimum and ideal obstacle detection distances are 30cm and 1m respectively. With a lidar sensing circuit, if the vehicle can detect objects within 1 vehicle length (30cm) it should have enough time to stop. Lastly, a vehicle that is less than 1 square ft is ideal in order to limit the area that can get hit by water, limiting potential damages.

,

5.2 UGV Design

In this section, design parameters regarding the subsystem functionalities of each engineering challenge for the UGV are elaborated.



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
6	RaytheonCAP_UGV_MountingPlate_v1		1
8	RaytheonCAP_UGV_GPSMount_v1		1
9	RaytheonCAP_UGV_GPSMount_EndPiece_v1		1
10	RaytheonCAP_UGV_GPSMount_GroundPlate_v1		1
12	RaytheonCAP_UGV_Gutter_v1		2
13	RaytheonCAP_UGV_WaterSensorMountBracket_v1		1
14	Water Sensor Container		1
15	Water Sensor Funnel		1
16	Water Sensor		1
17	RaytheonCAP_UGV_PlateSkirtWall_v3		2
24	Base Support Cap		4
25	Long Support Rod		1
26	Medium Support Rod		2
27	RaytheonCAP_UGV_AruCoPlate_v2		1
28	RaytheonCAP_UGV_AruCoPlateCap_v3		4
29	Short Support Rod		1
30	Top Plate V2		1

Figure 16: UGV B.O.M

5.2.1 Lynxmotion Aluminum A4WD1 Rover Kit

The Lynxmotion Aluminum A4WD1 Rover Kit was chosen for quick structural foundation of our UGV concept. It came with an aluminum chassis frame, two plates for the top and bottom of the chassis, four 12.0V dc 30:1 gear head motors and 4.75" tires and wheels and has plenty of attachment points for miscellaneous hardware needing to be attached.

This UGV kit was chosen over our previous tank kit because of the reduced friction between the ground and wheels compared to the tank treads. The 4 geared DC motors also provide more torque compared to the dual DC motors of the tank kit.

Figure 17: UGV Rendering

Figure 18: Lynxmotion Aluminum A4WD1 Rover Kit

5.2.2 ArUco Mount

As the ArUco marker is what differentiates our UGV from the UGVS of the other schools, having the ArUco marker visible is very important. As it is the main target for the other schools, making the marker water-resistant is also very important. To solve these issues, the ArUco marker is sandwiched between two clear acrylic plates. They are held together with screws and nuts, with the screws specifically having o-rings under the head that prevented water from seeping in through

the screw holes. Flange caps are screwed into the bottom acrylic plate, which are put on top of support rods. Those caps are attached to the rods using quick-release pins. These rods are also similarly attached to the UGV, with the rods being kept in place of flange caps that are screwed into the mounting plate of the UGV using quick-release pins.

5.2.3 GPS Mount

It is required to have positional accuracy of less than 1ft for proper operation and adherence to RTX guidelines for the competition. Given the UGV is not as constrained regarding weight savings, a bulkier, more precise, GNSS L1/L2 band antenna was selected. For best results, a mount was designed and fabricated to allow the antennae the most unrestricted aerial view that was deemed allowable (i.e height and distance allowance from the UGV base itself). A 1ft long aluminum square tubing extrusion was selected to be angled 45 degrees from the ground plane. Instead of cutting angles into the extrusion itself, 3D printed angle adapter mounts were designed and printed to achieve the same result, in less time. This was done given limited precut fixturing dimensions on the UGV kit itself which would of made cutting the extrusion and kit more time intensive. Printed models attach to both ends of the extrusion to complete the connection. Finally, a steel plate received alongside the Sparkfun "GNSS L1/L2 Multi-Band Magnetic Mount Antenna - 5m (SMA)" antenna is fixed atop the mount to complete the subassembly. Referencing Figure 9, This sub assembly consists of parts 8, 9, and 10.

Figure 19: GPS Mount View

5.2.4 Electronics Housing

An electronics housing is required to protect the onboard electronics from wet environmental conditions with considerations to be be made regarding heat dissipation of some components such as the motor driver shields that could potentially pose significant risk to neighboring components and the vehicle itself. The onboard components, particularly motor drivers, are still being tested to prove efficacy for sake of competition. The housing was made with considerations for the motors we utilized, the power ratings they were equipped to handle, and the additional electronics and their corresponding heat dissipation.

Figure 20: UGV FlowChart

5.2.5 Power and Propulsion:

The UGV is propelled by two DC-geared motors connected to each wheel. These motors are controlled by two motor controllers, one controlling the front two wheels and one controlling the back two wheels. The motor controllers and motors are powered by a single 14.8V 6500mAh LiPo battery.

5.2.6 Sensory Components:

The standard sensory requirements of the UGV is to put off both sound and LED illumination when hit by enemy UAV water strikes. The water detection was provided by a rain/water sensor, where all the water was channeled towards it by using a hydrophobic coating on the top of the marker. When the marker gets hit, water sensor detects the water and send signals to the onboard microcontroller which then illuminates the LED and sound the alarm of a hit status. The LED and sound package were apparent from the sidelines of the competition and were thus appropriately powerful. Given the UGV has far fewer components relative to the UAV, a simple Arduino microcontroller board should be sufficient for controlling the respective hardware.

5.2.7 Electrical Component/Subsystem Descriptions

The UGV's operational framework consists of an Arduino micro-controller interfacing with a Pixhawk flight controller. This is alongside additional peripherals such as the rain/water sensor used for water detection, two motor drivers, and ultrasonic sensors designated for object avoidance. Moreover, the Pixhawk flight controller is integrated with the Real-Time Kinematic (RTK) GPS system, enabling the UGV's autonomous navigation capabilities. To show this visually, the connection diagram is as follows:

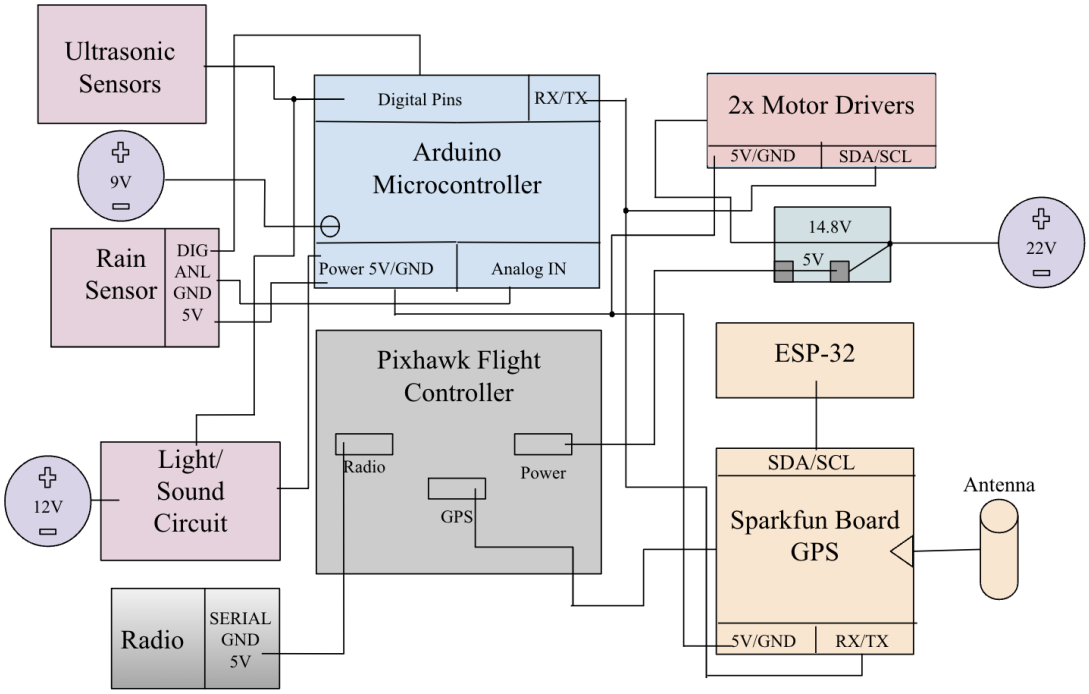


Figure 21: UGV Connections Diagram

5.2.8 Pixhawk Flight Controller

Both the UAV and the UGV utilize Pixhawk flight controllers. These flight controllers are interfaced through the MavSDK library. The Pixhawk controllers were selected because of their open source firmware and impressive autonomous capabilities.

6 UGV Drawings

See appendix for UGV drawing portfolio.

7 Analysis and Modeling

7.0.1 Water Gun Analysis and Modeling

The most relevant engineering characteristics of the water gun are its firing speed, volumetric flow rate, and rated pressure. 40 ml/sec was an appropriate flow rate goal as the drone is required to deliver 20 mLof water to the target, the water was delivered in a 1 second window, and a factor of safety of 2.0 was chosen to account for gun inaccuracy. Keeping flow rate high while shooting at the fastest possible velocity is difficult because we cannot operate at above 30 psi without exceeding the rated pressure of the water gun and aerodynamic factors necessitate high velocities to limit flow deflection.

A flexible 7/16" OD, 5/16" ID clear PVC tube is used along with two barbed fittings. A quantified resistance number was not measured for these parts as the resistance of the nozzle and solenoid valve is much more significant. Our solenoid constricts to a diameter of about 1 mm and contains three 90 degree bends. This almost certainly creates the most flow resistance.

To model the water gun system, a resistance analogue equation was used, where pressure is analogous to voltage and flow rate is analogous to current. Bernoulli's equation with head loss was not used because the exit nozzle of the system is at atmospheric pressure. What this means is that the information gained by measuring the head loss of the fluid is not anymore important than the resistance, R, that we calculate with the equation below.

$$\dot{V} = \frac{\Delta P}{R} \tag{1}$$

To determine the exit velocity, we use the mass flow rate equation.

$$\dot{m} = \rho v A = \rho \dot{V} \tag{2}$$

Where velocity is v and the nozzle diameter is A. We can rearrange this equation to calculate velocity.

$$v = \frac{\dot{V}}{A} \tag{3}$$

This analysis was performed for two nozzle diameters because our current water gun system allows the size of the exit nozzle to be switched out. Currently, 0.203" and 0.1" diameter nozzles are available. Testing was done to determine the specifications of the system for different nozzle sizes and pressures. Below is a table of all of the measured specs for the two nozzles.

Nozzle Size	Pressure (psi)	Flow Rate (ml/sec)	Velocity (m/s)	Resistance (psi/ml/sec)
0.203"	25	48.0	0.56	0.52
0.1"	25	32.8	1.8	0.76

Table 5: Water Gun Testing Results, Winter Quarter

We found that the resistance of the system increases as we decrease nozzle size. Inversely, exit velocity increases at the same time. This means that our system is not perfectly linear, meaning we do not get two times the velocity when we use a hole that has two times less the area. Further testing determined the need for an even smaller nozzle if a higher exit velocity is needed, however the modular design of our water gun means that components like the tubing and nozzle can be

swapped out very easily and leakage has not been as significant issue when swapping parts during the course of our testing.

When using the 0.1" nozzle, we fall below our target specification of 40.0 ml/sec. What this means is that we must shoot for a longer window of time in order to deliver the 20 mLof water with a factor of safety of 2.0 as specified by the Rules of Engagement. However, we are able to shoot at a much higher velocity.

Wind and rotor downwash was a significant aerodynamic factor on our water stream and determined the accuracy of the water delivery system. Further testing with the water gun fully mounted to the drone determined if the current configuration can fire at a fast enough velocity to overcome these effects and shoot straight down at the target, which it was able to do.

7.0.2 UAV Motor Analysis

The motors of the UAV are critical components for flight. To select the correct motors, thrust generation must be taken into account to overcome the payload mass. Furthermore, control and maneuverability are key characteristics of motors, as they allow for precise control of the drone's movements. The drone chosen for this purpose was the TL8X000 X8 Octocopter Pro Combo. An octocopter, due to its number of arms, provides greater stability and control over flight.

In total, the UAV had a payload of 8.2 kg, including the weight of the drone. The industry standard for a motor is a 2:1 thrust-to-weight ratio.

$$T = 2W = 2 \times 8.2\,kg = 16.4\,kg \tag{4}$$

Essentially, the above shows that the octocopter should be able to provide 16.4 kg of thrust. Since there are 8 motors on an octocopter, we find the following:

$$T = 16.4\,kg \quad \Rightarrow \quad \frac{16.4\,kg}{8} = 2.05\,kg \tag{5}$$

Therefore, we found that we need at least 2.05 kg of thrust per motor on the octocopter. The motor chosen was the Tarot 4114/320 KV Brushless Motor. According to the data sheet, this motor can provide 2.15 kg of thrust.

$$T = 2.15\,kg \times 8 = 17.2\,kg \tag{6}$$

With these motors, the drone can provide a total thrust of 17.2 kg, which is greater than the total payload we need to carry.

7.0.3 UAV Power/Battery Analysis

The power supply is a critical point of failure in every system. In this system, the drone must be able to fly for a certain amount of time without failure while also supplying power to the peripherals, including the Raspberry Pi, Pi Camera, Pixhawk, and the water delivery system.

The battery we decided to use is a 10000mAh 22.2V 222Wh battery. We had a total of 30 minutes to fly for each challenge and could replace the battery as many times as needed in between.

$$Power = (10000, mAh) \times (22.2, V) = 222, Wh \times 0.80 = 177.6, Wh \tag{7}$$

Say our average flying time is about 11 minutes per round, which is 0.1833 hours per round. Using the usable capacity and flying time, we find:

$$Power = \frac{UsableCapacity}{FlyingTime} = \frac{177.6, Wh}{0.1833, h} \approx 968, W \tag{8}$$

The above calculation shows that the battery is capable of providing 968 W of power for a flight time of 11 minutes at 80 percent efficiency. A Raspberry Pi and Pi camera usually draw a maximum of 12W/h, so in 11 minutes, they would draw approximately 2.5W. The Pixhawk consumes a maximum of 2.5W of power. To actuate the solenoid valve, a current of 300mA and 12V is needed. As the system would only be on for a few minutes, it would require around 5W of maximum power. The motor power draw would be the greatest, with a current draw of 14.5A and a voltage pull of 22.2V. The total power draw for the 11 minutes would be approximately 400W. To account for the extra power needed to compensate for the payload weight, about 800W of power is required.

$$TotalPower = 2.5, W + 2.5, W + 5, W + 321.9, W \approx 400, W \rightarrow 400, W \times 2 = 800, W \tag{9}$$

At 80 percent operational efficiency, the battery can account for this. We do not run the battery to its maximum potential to avoid oxidizing them and getting poor performance. This battery is able to handle the drone for a short period of time, so multiple batteries were bought to replace each one after 10-15 minutes of flight time.

8 Minimum Performance Specs for Components/Subsystems

Table 6: Minimum Performance Specifications for Pixhawk Autopilot System

Performance Parameter	**Minimum Specification**
Processor	ARM Cortex-M4F, 168 MHz
IMU	3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer
Flight Modes	Stabilize, Altitude Hold, Loiter, Auto, RTL (Return-to-Land), etc.
GPS	Ublox NEO-M8N or compatible GPS
Telemetry Communication	2.4 GHz telemetry radio with a range of at least 500 meters
Input Channels	At least 8 PWM input channels for RC receiver connectivity
Output Channels	At least 6 PWM output channels for servo and motor control
Compatibility	Compatible with Octocopter
Operating Temperature	-10°C to 55°C (14°F to 131°F)
Power Input	5V DC

Table 7: Minimum Performance Specifications for Raspberry Pi 4

Performance Parameter	**Minimum Specification**
Processor	Quad-core ARM Cortex-A72 CPU, 1.5 GHz
Memory	2 GB, 4 GB, or 8 GB LPDDR4 SDRAM
Storage	MicroSD card slot (minimum recommended: Class 10)
USB Ports	2 × USB 3.0, 2 × USB 2.0
Ethernet	Gigabit Ethernet port
Wireless Connectivity	802.11ac Wi-Fi, Bluetooth 5.0
Video Output	2 × micro HDMI ports (up to 4Kp60 supported)
Audio Output	3.5mm analogue audio-video jack, HDMI
GPIO Pins	40-pin GPIO header
Operating Temperature	0°C to 50°C (32°F to 122°F)
Power Input	3.3V DC via USB-C connector
Dimensions	85.60mm × 56mm × 21mm

Table 8: Minimum Performance Specifications for SparkFun: GPS-RTK-SMA ZED-F9P

Performance Parameter	**Minimum Specification**
Receiver Chipset	Ublox NEO-M8N
Satellite Systems	GPS, GLONASS
Receiver Channels	32
Update Rate	Up to 5 Hz
Position Accuracy	4.0 meters CEP (Circular Error Probable)
Velocity Accuracy	0.1 meters/second
Startup Time	¡ 50 seconds
Operating Voltage	3.3V - 5V
Interface	Serial UART (TTL)

Table 9: Minimum Performance Specifications for Raspberry Pi 4 Camera Module

Performance Parameter	**Minimum Specification**
Resolution	8 megapixels (3264 x 2448 pixels)
Sensor	Sony IMX219
Sensor Size	1/4 inch
Pixel Size	1.12 μm x 1.12 μm
Frame Rate	1080p @ 30 fps, 720p @ 60 fps
Lens	Fixed focus
Field of View	62.2 degrees diagonal
Video Output	RAW Bayer format, YUV420 format
Interface	CSI-2 (Camera Serial Interface)
Dimensions	25mm x 23mm x 9mm (excluding camera connector)
Weight	3 grams
Compatibility	Raspberry Pi 4

Table 10: Minimum Performance Specifications for ESC Motors:XRotor Pro 40A Opto

Performance Parameter	**Minimum Specification**
Voltage Input	22.2v
Max Current(A)	14.5A
Max thrust(Kg)	1.77Kg
Weight(g)	32g
Compatibility	Brushless motors

9 Risk Analysis

The risk analysis for the UAV and UGV was considered by evaluating different test case/scenarios in which our drone would fail to accomplish its objective. As a result, the following aspects are considered: functionality of the kill switch, actuator and sensor functionalities, fracture of physical systems/components, software bugs, power supply functionality, water leaks on hardware, water delivery payload failure, GPS malfunctioning, and the ability of our drone(s) to carry out the objectives in a timely manner. Presented respectively on the following page (in the first, second, and third columns) are tables describing:

The **Severity** of the scenario (based on a scale from 1 to 5, with 5 being the highest severity): If Said Scenario Happens, How Bad Can it Get?

The **Probability** (based on a scale from 1 to 5, with 5 being the most probable): How Likely Said Scenario Would Happen With Proper Preparation.

The **Priority** (based on a scale from 1 to 25, with 25 being the highest priority) of Emphasizing Risk Concern (with Comments)

Risk	Severity	Probability	Priority
Kill Switch Process Malfunctions (UGV & UAV)	A non-working kill switch on the autonomous drone could be potentially dangerous and would violate FAA guidelines. 5	A working kill switch is one of the requirements to participate in the competition. 1	This is a risk that is prioritized for the competition and was tested in many cycles, prior to entering the competition 5
Actuator Malfunctions (UGV & UAV)	Actuator malfunctions would lessen our chances of accomplishing our payload objective and winning the competition. 3	Through a series of testing, this risk can be mitigated and made improbable. 2	Our team conducted rounds of testing to individual and joint functionalities of the actuators. 6
Sensor Malfunctions (UGV & UAV)	Sensor malfunctions can distort the way our drone interprets the 3D space, causing inaccurate actuations. 3	Similarly to actuator probability: this risk can be mitigated through proactivity. 2	Our team conducted rounds of testing to analyze the output of the sensors in environments similar to the competition setting. 6
Physical Component of System Fractures (UGV & UAV)	With other subsystems functioning properly, the risk this scenario produces refers to unexpected environmental inputs (e.g. wind causing inertial instability $\rightarrow$ possibly leading to fracture of drone rotor if it collides with any other part of drone) 2	By analyzing material binding and integration properties, the loads our drone can handle with the fracture point of materials used can be quantifiably compared. 1	When designing our integrated system of subcomponents, factor of safety was considered for main functionalities. 2
Software Bugs (resulting in: e.g., nonautonomy) (UGV & UAV)	Software bugs can result in the drone doing an unintended task when it is placed in a foreign environment. 4	With organized code blocks, multiple code reviewers, and rounds of testing, this risk becomes improbable. 1	Addressing software bugs is a proactive task that different team members can do on the software. 4
Power Supply Output Disturbed (e.g., runs out of battery) (UGV & UAV)	Shortage of power output to the system would result in the drone unexpectedly turning off 5	Through a series of testing and preparation, this scenario can be made highly improbable. 1	Address the issue by: testing power supply before competition, back-up power supply perhaps, use a voltmeter 5

Risk	Severity	Probability	Priority
Water Leaks on Hardware (Either by rain, UAV: through the water stored UGV: water dropped on) (UGV & UAV)	If water gets into the hardware, it would essentially make our drone inoperable 5	Designing proper housing for the hardware to make it water-proof makes this scenario improbable. 1	Similar to "code reviewing", A "design review" can be conducted with our team, professors, and other resources. 5
Water Delivery Process Failure (payload misses) (UAV)	Water delivery failure would at worst lose us a couple points for the competition aspect. 1	There are numerous factors to account for depending on delivery mechanism. As such, they have their strengths and weaknesses in certain environmental conditions. 3	Address this risk case through analysis of projectile dynamics in: math, simulation and real-world testing. 3
GPS System doesn't work as intended (e.g. delay in communication) (UAV)	The GPS malfunctioning may cause our drone to depict its boundaries wrong. 3	Not knowing the location of the competition reduces our ability to fully comprehend and test our drone's capabilities with real-time kinematic. (RTK station location) 2	Need at least three people among the team to be proficient and knowledgeable about the implementation and functionality of the RTK system. 6
Drone is Too Slow in Carrying Out Objective (UAV)	If the drone is too slow, then it may not be able to drop the payload on each of the enemy UGVs. 2	Testing can be done to mitigate this risk, the objective task must be emulated by setting up models of enemy UGVs when testing. 2	This risk is attributed to many factors including efficiency of both mechanical and electrical systems and their dependencies upon one another. (different system aspects affect this risk e.g. slow rotor speed, invalid object detection, inefficiency of autonomy: all would slow whole operation) 4

Table 11: Risk Analysis for UAV and UGV

10 Final Prototypes and Designs

10.1 Prototype 1: Water Delivery/Dropping Mechanism

10.1.1 Concept: Water Gun

This concept is based on a water gun concept by Long Lam at MIT in 2010[15]. A lever with a rubber stopper covers an air-bleed hole that can be opened to allow water to flow in through the one-way valve next to it. A torsional spring keeps that stopper closed when not filling. The one-way air valve on the end end is used to then pressurize the water chamber. This can be done with CO_2 cartridges. When a shot is signaled, the servo can turn the ball valve open and closed to let out a shot of water.

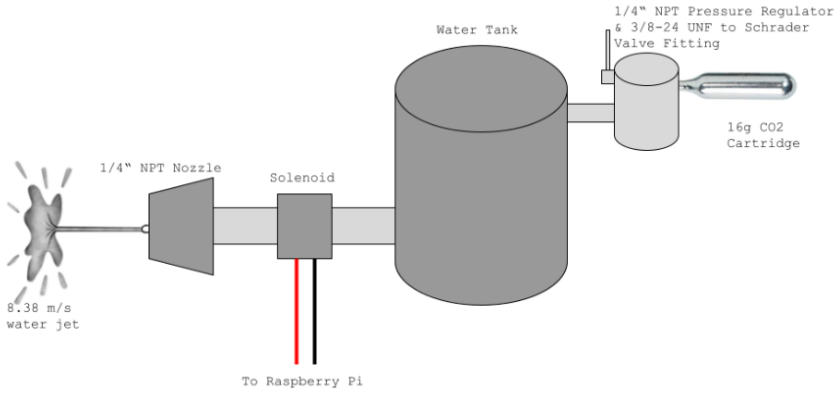


Figure 22: Concept 2

10.1.2 Description:

One of the most important design decisions during this project was the mode of water delivery as it is the only way that the UAV can score points for our team and the quality of our mechanism is highly dependent on the conditions on the day of the competition and on the interactions between our different subsystems. Given that the Rules of Engagement is vague with regards to water delivery ("The Water Blast delivery method is be determined by design team according to the vehicle architecture"), many ideas were brainstormed and placed into two different categories: bomb-type and projectile-type. Bomb-type mechanisms drop a payload with a set mass and store all these payloads on-board. Projectile-types use compressed carbon dioxide to launch projectiles like water squirts at the UGV's, as opposed to just dropped them.

10.1.3 Purpose

The purpose of this prototype was to test the viability of each delivery mechanism. Each were evaluated based on factors such as accuracy, precision, volume delivery per "shot", firing and reload time, aerodynamic drag effects, and weight added. This helped determine which mechanism performed the best and whether or not two firing mechanisms could be implemented.

10.1.4 Define level of approximation:

This prototype was used to test one subsystem, though it had some effect on the others. The chosen delivery mechanism added load to the drone itself, which had to be compensated by the rotors, and if the rotors were compensating for a greater load, it drew more power from the battery whose power could be used on other subsystems. This made choosing a delivery mechanism difficult, so this system was isolated and all of its properties were tested before fully implementing it into the entire drone system..

10.1.5 Experimental plan:

Mock-ups were built of the delivery method with simplified designs and cheap, easy-to-make parts. This allowed the simulation to replicate the dynamics of each design without having to create robust, accurate, and costly prototypes. The prototypes dropped their projectiles on a 1' x 1' target, and their precision and accuracy were measured. The payload size was also measured and used to calculate how much load the aircraft had onboard

Raw Material	Tools
Variety of screws and fasteners	Screwdriver
Growler	Bucket of water
Solenoid Valve	
CO2 Canister	

Table 12: Bill of Materials, Prototype 1

10.1.6 Results:

The water delivery system was completed successfully, and the gun mechanism proved to be an excellent water blaster for the competition. After numerous trials, the water blaster achieved a flow rate of 32.8 ml/sec and an exit velocity of 1.8 m/s. Overall, the system could deliver 20 mLof water in 0.61 seconds, enabling many successful hits on the ground vehicle.

Nozzle Size	Flow Rate (ml/s)	Velocity (m/s)	Firing Time (for¿20ml)
0.1"	32.8 ml/s	1.8m/s	0.61 sec

Table 13: Water Gun Testing Results, Spring Quarter

10.2 Prototype 2: Object Detection Utilizing Camera System

10.2.1 Description/Purpose:

Object detection was another essential part of our project for the Aerial Vehicle. The competition required our Aerial Vehicle to autonomously detect an ArUco marker on the ground vehicle. To solve this challenge, our team integrated a camera with the Raspberry Pi and utilized an OpenCV library for ArUco marker detection

10.2.2 Define level of approximation:

The system as a whole, integrated with the Aerial Vehicle, was the end goal. However, during that quarter, the goal was to find a camera system that had high enough resolution and camera speed to capture the ArUco markers. This system needed to be tested separately from the majority of the components as it was the most essential component for identifying the ground vehicles. After getting all of those parts working separately, they were integrated into the drone with the Raspberry Pi as well as the flight controller. This allowed decisions to be made in real time and autonomously.

10.2.3 Experimental plan and Results:

OpenCV and C++ programming were utilized to program the camera to detect the ArUco marker. Different algorithms were created to detect the opposing team markers, navigate to their location, and deploy water on them once detected. Additionally, algorithms were developed to center the marker in the frame of the camera for the best shot. The power draw was accounted for, and a 22V battery was used to power the drone and its peripherals, including the camera, which was connected to the Raspberry Pi. The camera was able to detect markers from a height of 35 feet and had a maximum field of view of 12.5 yards by 8 yards, which determined our flight pathing.

Electronic Components
Raspberry Pi 4 Model B
Raspberry Pi Camera Module V2-8 Megapixel,1080p (RPI-CAM-V2)
Breadboard

Table 14: Bill of Materials, Prototype 2

10.3 Prototype 3: RTK GPS System

10.3.1 Description/Purpose:

An RTK system or Real-Time Kinematic system was the GPS system used during our competition. An RTK system allowed for high-precision positioning. It consisted of a base station, rover, and internet access. Utilizing data sent from the Global Navigation Satellite System (GNSS) as well as a base station in an RTK system, it compared the two data sets, took the time difference, and sent that information to the rover. This provided us with centimeter-level accuracy versus a regular GPS, which could only give meter-level accuracy. For the purpose of this project, it allowed us to create waypoints, which are reference points to determine where the drone and the ground vehicle should be moving. There were free base stations throughout California that could be used for the corrections, which were sent to the NTRIP network and then passed onto the phone and rover.

10.3.2 Define level of approximation:

This system was first tested standalone to determine its accuracy. It was then integrated with the Pixhawk flight controllers and an ESP-32 microcontroller over WiFi to get correctional data, ensuring the GPS system's accuracy was on the centimeter level. This system was integrated into both the ground and aerial vehicles.

10.3.3 Experimental plan and Results:

During the competition, the GPS system was integrated into both the ground and aerial vehicles. It was combined with the Pixhawk and interfaced with the MAVSDK library and Mission Planner software. This setup aided the autonomous system in triangulating the positions of the ground vehicles and mapping out the geofence of the competition area.

Electronic Components
Raspberry Pi Camera Module V2-8 Megapixel,1080p (RPI-CAM-V2)
Breadboard
ESP32
Sparkfun GNSS SMA Breakout Board F9P

Table 15: Bill of Materials, Prototype 2

10.4 Prototype 4: Water Detection System

10.4.1 Description/Purpose:

The water detection system was designed to meet the RTX rules of engagement for the competition, specifically the requirement to detect 20mLof water. To achieve this, we employed a superhydrophobic coating from Rust-Oleum, which is shown in Figure 23. This coating promises contact angles up to 160 degrees, classifying it as superhydrophobic. The coating performed well, allowing us to tilt our ArUco marker only 5 degrees from parallel to the ground. This minimal tilt was crucial to avoid interference with other teams' ability to detect our marker, ensuring fair play.

10.4.2 Testing:

In our testing phase, we verified the effectiveness of the superhydrophobic coating and the water detection system. We started by applying the coating to the surfaces involved in water detection and observed its performance in repelling water. After confirming the coating's efficiency, we proceeded to test the entire water detection setup, including the 3D printed PLA plastic gutters and the containment vessel. Water was dropped from heights of 2 feet, 4 feet, and 6 feet, with water being successfully collected in our containment device each time.

10.4.3 Results:

The water detection system successfully detected the required 20mLof water. The superhydrophobic coating allowed water to slide off with only a 5-degree tilt, directing it into the 3D printed PLA plastic gutters. These gutters funneled the water into the lowest corner, where it was collected by a PLA printed containment vessel. This vessel housed an off-the-shelf water sensor. The interior geometry of the vessel was carefully calculated so that 20mLof water nearly reached the top of the sensor, utilizing the full analog resistive range of our water level sensor for more precise measurement in theory.

Figure 23: Rust-Oleum Superhydrophobic Coating

10.5 Prototype 5: Propeller Guards

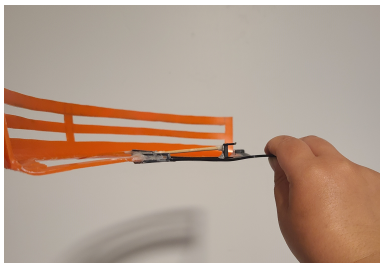
10.5.1 Description/Purpose:

Propeller guards were required as one of the qualification tests. They serve as both protection for the drones' propellers and the people and objects around the drone. As the drone kit we bought did not come with propeller guards, we had to design our own.

10.5.2 Design and Results:

Because the drone kit wasn't designed with propeller guards in mind, it was difficult to find a way to attach them. We decided to attach them to the motor mounts in the back where the motor mount attaches to the arms of the drone. While the dimensions of the propeller guards could work out, the first version did not actually work as intended. The part of the guards that wrapped around the propellers sagged down due to its own weight, as seen in subfigure a below. As the arm portion was made of 3d printed PLA, some of the sagging was mitigated by heating up the arms and curving them slightly up to counter the sagging. A support was also added to the center arm to provide more rigidity, as seen in subfigure b below. While this dampened some of the possible vertical oscillations that may occur when the drone was flying, it did not appear to prevent any sort of twisting oscillations. In the end, we decided that it was safer to not fly with them attached, as we did not get an opportunity to fly with them on before the competition and did not know how well the drone would be able to fly. If we had a second chance, we would probably

(a) Initial Propeller Guard Design



(b) Propeller Guard with Added Supports

Figure 24: Propeller Guard Designs

Figure 25: Drone with Propeller Guards On

11 Final Test Results

Our final test results were our vehicles's performance on the competition day with CSU Long beach and Cal Poly SLO at SBCC on June 8th 2024. Challenge 1 was completed however since CSULB and Cal Poly's drone's didn't fly, challenge 2 was done with some modifications (only checking if UGV's are moving in a pre-planned path and if water detection was working properly).

11.1 UAV

There were 11 flight attempts at Challenge 1 (UGV's are stationary). The individual components of the system performed well together: the marker was detected 9 out of 11 times, and water was delivered successfully on all 11 attempts. During most of the water delivery, the drone missed the UGV by less than an inch or only hit the corner of the ArUco marker meaning it was able to fly reliably within +/- 1 meter, with 50 percent certainty. This was due to inconsistencies in picking up GPS locations by the RTK GPS system. While the overall accuracy needed improvement, the precision of our operations was consistently good.

11.2 UGV

During challenge 1, our UGV was stationary while our drone searched for other UGV's to drop water. When water detection was tested on our UGV it was able to detect 20mL of water 100 percent of the time, setting off the both motorcycle horn and lights.

During challenge 2, the UGV's were tested on their movement and pre-planned paths. The goal of the UGV was to move in a straight line at a speed of 0.17 mph. However, our UGV ended up moving in a straight line but at a much faster speed as Mission Planner was not responding to the speed changes. When moving at that faster speed when water was detected, the UGV stopped, played sound, and set off the light.

11.3 Competition Results

After each round, the points were tallied based on the RTX scoring rubric. UCSB won first place, CSULB won second place and Cal Poly SLO won third.

12 Broader Impacts

12.1 Global and Cultural Setting

The Raytheon Drone Competition (RDC) provides a platform where students from various universities come together to develop and test advanced unmanned vehicle systems. This competition fosters global collaboration, innovation, and cultural exchange among students from diverse backgrounds. By promoting the integration of technologies such as autonomous navigation, computer vision, and artificial intelligence, the RDC prepares students to tackle global challenges and contribute to the advancement of technology on an international scale.

12.2 Implications if Continued

If this project continues, it could lead to significant advancements in autonomous vehicle technology, which has broad applications in areas such as disaster response, environmental monitoring, and logistics. For instance, drones equipped with autonomous navigation and target identification capabilities can be used for delivering medical supplies in remote areas, conducting search and rescue operations, and monitoring wildlife populations. These applications can save lives, protect the environment, and improve the efficiency of various operations globally.

12.3 Potential for Good

The technologies developed through the RDC have the potential to be used for humanitarian and environmental causes. For example, drones could be deployed to deliver essential supplies to disaster-stricken areas, thus providing timely aid and support. Additionally, these autonomous systems can be used for environmental conservation efforts, such as monitoring deforestation, tracking endangered species, and assessing the health of ecosystems.

12.4 Social Factors

The project promotes STEM education and encourages students to pursue careers in science and technology. By involving students in real-world problem-solving and innovation, the RDC helps to cultivate a new generation of engineers and scientists who are equipped to address future challenges. Moreover, the competition encourages teamwork, communication, and project management skills, which are essential in any professional setting. The inclusion of diverse teams from different universities also fosters an inclusive environment where ideas can be exchanged freely, promoting cultural understanding and cooperation.

12.5 Summary

In summary, the broader impacts of the Raytheon Drone Competition extend beyond the immediate technical achievements. The project supports global innovation, humanitarian efforts, environmental conservation, and the development of future leaders in science and technology.

13 Task Distribution

There are 13 people associated with our Team - 10 students, 2 advisors, and 1 sponsor Below is a chart of how our team is organized and how communication flows.

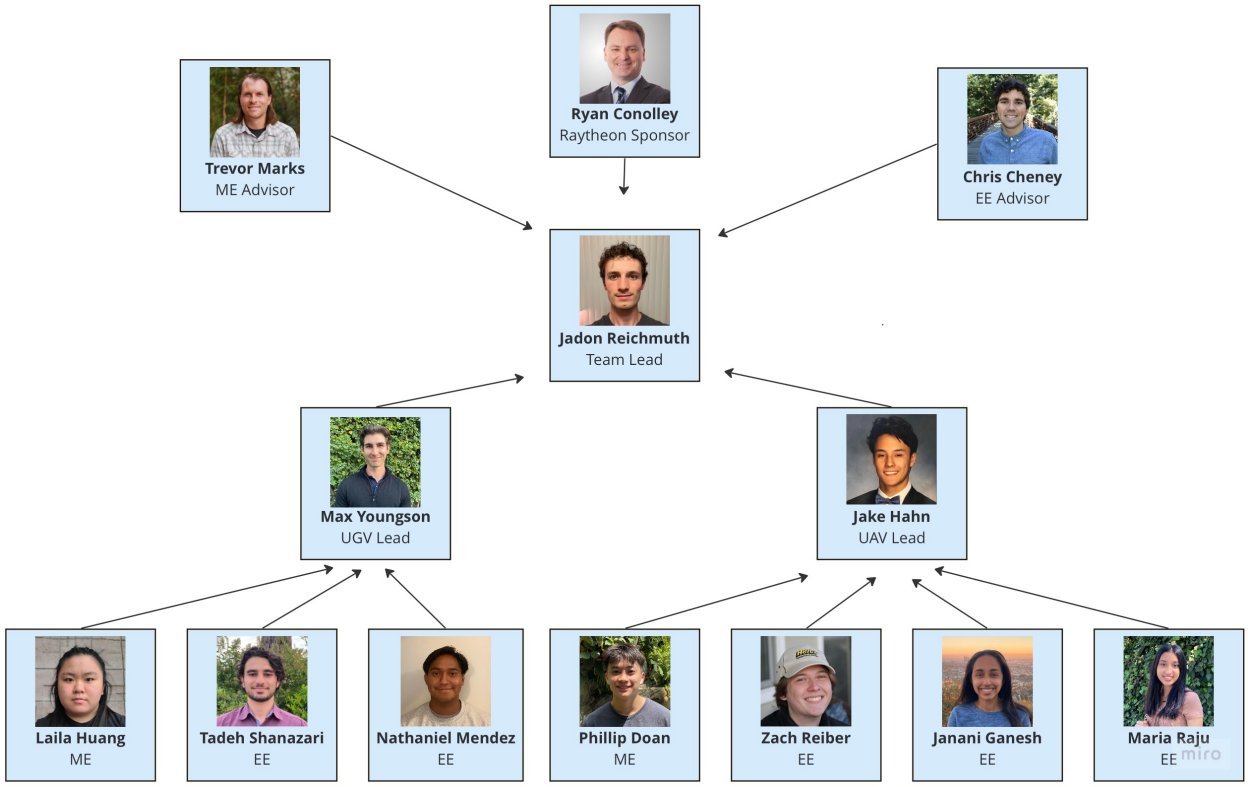


Figure 26: Organizational Chart

Each member has written a brief description of his or her role and skills applicable to this project below:

- **Ryan Conolley** is the Raytheon contact for our project.
- **Trevor Marks** is our Mechanical Engineering advisor for our project. All ME Students met with him every Wednesday from 1-2 to discuss our progress.
- **Chris Cheney** is our Electrical Engineering advisor for our project. All EE Students met with him every Wednesday from 1-2 to discuss our progress.
- **Jadon Reichmuth** is our team lead. He oversaw each subsystem, ensure all deadlines are met, and maintain good communication between all team members and Advisors/Sponsors.
- **Jake Hahn** is our UAV lead. He oversaw all work done on the drone, ensure the functionality of all subsystems, and foster a collaborative environment between all the UAV team members.
- **Max Youngson** is our UGV lead. He oversaw all work done on the ground vehicle, ensure the functionality of all subsystems, and foster a collaborative environment between all the UGV team members.

- **LaiLa Huang** is under the UAV team. Her strengths are in CAD and concept generation. She worked on the UAV's water delivery system.
- **Janani Ganesh** is under the UAV team. Her strengths are in electrical circuit design and control systems. She worked on object detection and avoidance.
- **Maria Raju** is under the UAV team. Her strengths are in computer vision, object-oriented programming, and sensor integration. She worked on object detection and flight control.
- **Phillip Doan** is under the UAV team. His specializations include computer vision, python programming, and CAD. His interests in this project are in object detection, housing design, and systems integration.
- **Nathaniel Mendez** is on the UGV team. His strengths are in sensor peripheral design and communication systems. His interest in this project are in designing sensor interfaces with the sound and LED illumination on the UGV.
- **Zach Reiber** is under the UAV team. His strengths are in embedded software and computer architecture. His interests in this project are helping design the embedded software for both the ground and air vehicle.
- **Tadeh Shanazari** is under the UGV team. His strengths are in circuit and sensory design. His interests in this project are helping design the motion paths of the ground vehicle as well as the computer vision aspect of the air vehicle.

14 Works Cited

[**1**] "14 CFR Part 107 Subpart D – Operations Over Human Beings." Accessed: Oct. 26, 2023. [Online]. Available: https://www.ecfr.gov/current/title-14/part-107/subpart-D

[**2**] "Gyro sensors - How they work and what's ahead — about Gyro sensor — Technical Information — Information." Accessed: Oct. 26, 2023. [Online]. Available: https://www5.epsondevice.com/en/information/technicalinfo/gyro/

[**3**] A. Babinec, L. Jurišica, P. Hubinský, and F. Duchoň, "Visual Localization of Mobile Robot Using Artificial Markers," Procedia Eng., vol. 96, pp. 1–9, Jan. 2014, doi: 10.1016/j.proeng.2014.12.091.

[**4**] M. G. Ocando, N. Certad, S. Alvarado, and Á. Terrones, "Autonomous 2D SLAM and 3D mapping of an environment using a single 2D LIDAR and ROS," in 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Nov. 2017, pp. 1–6. doi: 10.1109/SBR-LARS-R.2017.8215333.

[**5**]"DJI Inspire 3 - Specs - DJI," DJI Official. Accessed: Nov. 15, 2023. [Online]. Available: https://www.dji.com/inspire-3/specs

[**6**]"DJI Mavic 3 Pro - Specs - DJI," DJI Official. Accessed: Nov. 15, 2023. [Online]. Available: https://www.dji.com/mavic-3-pro/specs

[**7**]"Fisherman MAX (FD2) Heavy Lift Fishing Drone," SwellPro Store. Accessed: Nov. 15, 2023. [Online]. Available: https://store.swellpro.com/products/fisherman-max

[**8**]J. Y. W. he´s not writing about tech, you may find J. flying his M. Mini, and re-watching B. G. or enjoying a cup of java, "How Much Weight Can A Drone Carry? Real Cases (lbs Kg) - DroneGuru." Accessed: Nov. 15, 2023. [Online]. Available: https://www.droneguru.net/weight-a-drone-carry/

[**9**]"SplashDrone 4 Fishing Edition," SwellPro Store. Accessed: Nov. 15, 2023. [Online]. Available: https://store.swellpro.com/products/splashdrone-4-fishing-bundle

[**10**]"Amazon.com: Top Race Drone Clip Remote Control Object Launcher - Release and Drop Drones Delivery (180 Feet) - Drone Drop Release: Toys Games." Accessed: Nov. 15, 2023. [Online]. Available: https://www.amazon.com/Top-Race-Launcher-Delivery-PATENTED/dp/B07521DGVD

[**11**]R. M. Fitzpatrick, M. T. Mayberry, B. L. Nakayama, and E. C. Burt, "Ammunition magazine," US8839543B2, Sep. 23, 2014 Accessed: Nov. 15, 2023. [Online]. Available: https://patents.google.com/patent/US8839543B2/en

[**12**]"DIY Autonomous Intelligent drone sprayer - Parts List, Schematics, Build Details - YouTube." Accessed: Nov. 15, 2023. [Online]. Available: https://www.youtube.com/watch?v=Fflbc$_y$2IGQ

[**13**]U. Oskar, "Pocket article dispensing container," US2620061A, Dec. 02, 1952 Accessed: Nov. 15, 2023. [Online]. Available: https://patents.google.com/patent/US2620061/en

[**14**]RCLifeOn, "Remote Control Drop Mechanism," Instructables. Accessed: Nov. 15, 2023. [Online]. Available: https://www.instructables.com/Remote-Control-Drop-Mechanism/

[**15**] L. Lam, "The continuous refill, short-burst, hand- powered water toy".

[**16**] "Decibels dB — What are they?," Soundstop.co.uk. Accessed: Nov. 16, 2023. [Online]. Available: https://soundstop.co.uk/pages/what-are-decibels

[**17**] "Bontrager Ion Pro RT Front Bike Light - Trek Bikes." Accessed: Nov. 16, 2023. [Online]. Available: https://www.trekbikes.com/us/en$_US/equipment/bike-accessories/bike-lights/bike-front-lights/bontrager-ion-pro-rt-front-bike-light/p/22466/$

[**18**] "6 Hacks For Your iPhone Flashlight You Never Knew Existed," Fenix Store. Accessed: Nov. 16, 2023. [Online]. Available: https://fenix-store.com/blogs/news/blog-6-hacks-for-your-iphone-flashlight-you-never-knew-existed

15 Appendix

15.1 Drawings

The following pages include manufacturing drawings for all of the parts to be fabricated.