University of Texas at Austin

Unmanned Aerial Vehicle Team

AUVSI SUAS 2017 Technical Design Paper

Team Members

Blake Younger	James Bell	Israel Hernandez
Bradley Bridges	Hari Kosuru	Akshat Ramadurai
Rahul Butani	Clayton Lawrence	Evan Wilde

Contents

1 Systems Engineering Approach

1.1 Mission Requirement Analysis 2

1.1.1 UAS Derived Requirements Overview:

The most important mission requirements are autonomous flight, obstacle avoidance, object detection/classification/localization, and air delivery. The system is designed with these requirements in mind, while also considering the team's previous experience, and budget constraints. Other restrictions include a maximum takeoff weight of 55lbs, and a maximum airspeed of 70 KIAS, and standard, low risk fuel or batteries.

1. The UAS shall be capable of autonomous flight
2. The UAS shall accurately fly to each waypoint while remaining inside the flight boundaries
3. The UAS shall avoid stationary and moving obstacles, whose locations are received from the Interoperability system.
4. The UAS shall take images/video of approximately 0.1 square miles in under 40 minutes to identify target characteristics and location.
5. The UAS shall accurately deliver a standard 8oz water bottle to a drop location.

1.1.2 Ground Station Derived Requirements Overview:

1. The ground station shall display a map showing flight boundaries, UAS position, other competition elements, UAS speed, and altitude for the competition judges
2. The ground station shall send mission commands to the UAS, and receive imagery from the UAS.
3. The team shall have personal protective equipment (PPE) which includes, at minimum, proper tools, gloves, eye protection, and hearing protection when appropriate. Safety risk mitigation shall also be implemented, which includes team training, checklists, and radios for communication. The team shall have equipment (first aid kit, fire extinguisher) to respond quickly to emergencies.
4. The ground station shall receive mission details and submit mission deliverables using the interoperability system.
5. The ground station shall be portable and possible to set up in less than 20 minutes.

1.2 Design Rationale

1.2.1 Environmental Factors

The team began the Fall 2016 semester with a budget of $8300 for system design and travel expenses to Maryland. Six juniors and seniors from the 2015-2016 academic year remain active on the team this year. The rest of the UAV team consists of 15 freshmen and sophomores. To determine the system design approach, the Program Manager and Chief Engineer analyzed the team’s abilities. The team has experience in CAD modeling, aerodynamic analysis, electronic circuit design, computer vision, computer networks, and full stack development. With this wide range of skills, the UAV team is confident it can achieve all tasks at the competition.

1.2.3 Design Selection Criteria

The UAV team only considered fixed-wing aircraft because of the team’s previous experience with fixed wings. To select the ideal platform for the UAS, weighted criteria were made to score possible baseline aircraft.

<u>Low Wing Loading - 60%</u>
Low wing loading was given a significant weight because it has numerous performance benefits. Some of the most notable are lower takeoff and landing speed, lower stall speed, increased climb rate, increased efficiency during cruise, and tighter turns. These characteristics make an ideal search aircraft for the mission. A lower speed means each target spends more time in the camera’s field of view. Cruise efficiency allows the plane to stay in the air longer if necessary. A tight turning radius is useful for ideal searching patterns.

Usable Volume - 40%

Higher usable volume makes it easier to install system components, organize wiring, and change the location of components with design iterations. Based on University of Texas' aircraft design classes and UAV team data, we chose not to consider planes that had less usable volume than past, usable concepts. Experience taught us that planes with low usable volume are difficult to assemble and repair.

We analyzed 5 RC kit planes: the Spitfire, Theory Type W, Darkwing FPV Drone, Skywalker and the Hugin II using this selection criteria formula.

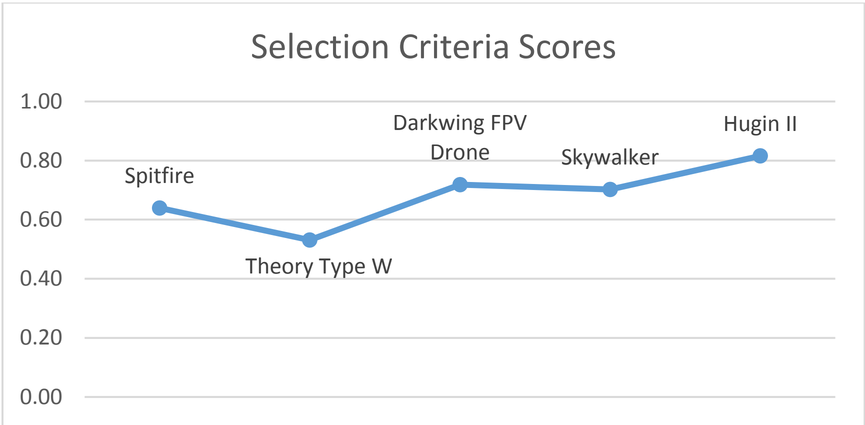

Figure 1: Comparison of Criteria Scores of the 5 RC Kits

	Spitfire	Theory Type W	Darkwing FPV Drone	Skywalker	Hugin II
Wing Loading (lbs/ft^2)	0.41	0.54	0.59	0.23	0.55
Usable Volume (in^3)	380	200	700	410	1080
Wing Loading Score	0.49	0.45	0.44	0.54	0.45
Volume Score	0.15	0.08	0.28	0.16	0.37
Total Score	0.64	0.53	0.72	0.70	0.82

Table 1: Depth Analysis of the Criteria Scores Comparison

The Hugin II achieved the highest selection score of 0.82.

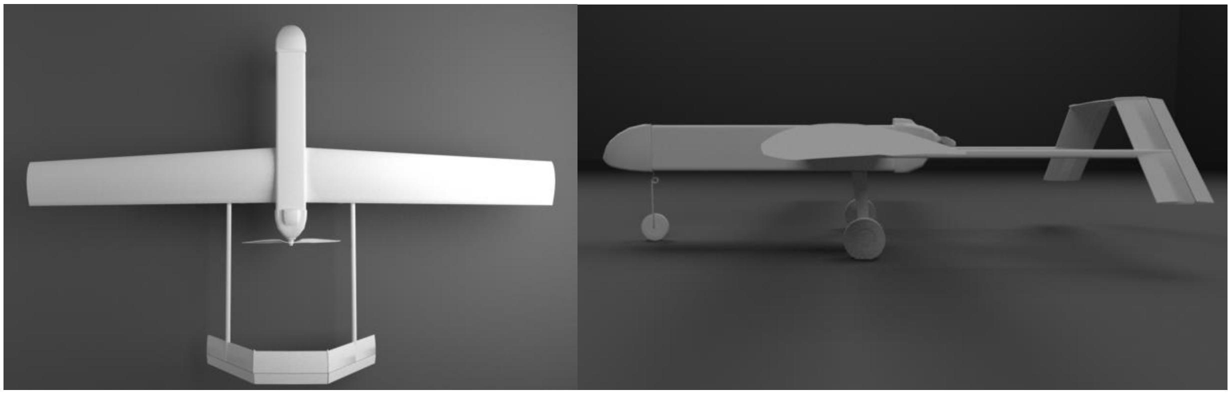

Figure 2: 3D model of Hugin II

1.3 Programmatic Risks and Mitigations

Risk	Description	Mitigation	Fall Back Plan
Flight test crash	Plane crashes or lands in condition where it is no longer capable of completing mission tasks	Perform flight checks and familiarize team with flight behavior and performance	Assess damage, cause of crash analysis, rebuild any components that require rebuilding
Safety and legal issues	The team must not design an aircraft which violates legal regulations. Operating the aircraft can also cause personal injury.	Appoint a safety officer who is knowledgeable about unmanned aerial vehicle laws.	Program Manager will be supervising the safety officer and will assume his job if necessary
Falling behind program schedule	Missing deadlines in Gantt chart	Work ahead of schedule when possible to create flexible margins	Appoint members from other sub-teams to assist with development
Failure to Meet AUVSI Deliverables	Deadline not met for deliverables	Program Manager understands rules and makes planes to meet deadlines ahead of schedule	Submit deliverables A.S.A.P., contact AUVSI judges for further instructions.
Insufficient ground crew training	Due to schedule delays, the ground crew may not have enough time to fully develop and practice their jobs during the mission.	Have frequent flight tests. Practice the mission exactly like it will be done at the competition.	Perform a “dry run” of the mission at the competition.

Table 2: Programmatic Risks and Mitigations

2 System Design

2.1 Aircraft

The Hugin II is a pusher configuration aircraft. It has a fuselage length of 42.5 inches and an effective wingspan of 102.25 inches The design gives a large amount of space to add components in, which in turn allows for the center of mass to be adjusted as needed. Additionally, the large space allows for easy accessibility to internal components, allowing them to be swapped when necessary with minimal effort. The main body consists of lightweight materials including a plywood fuselage frame with balsa paneling, carbon fiber beams supporting a rear H-wing configuration, and a *MonoKote* skin to provide an aerodynamic outer coating. The inner structure employs a plywood truss design with specific attention towards bearing the lift of the wings (translated through a central aluminum beam) and support points from the wheels, as well as the horizontal thrust and vibrations from the rear propeller. This truss layout translates applied forces evenly throughout the body to ensure stability during flight and flight maneuvers such as takeoff or landing. All internal components and their respective wires are secured to the rigid framework with adhesive material, screws, or Velcro in order to fully mitigate unstable movements during turbulence.

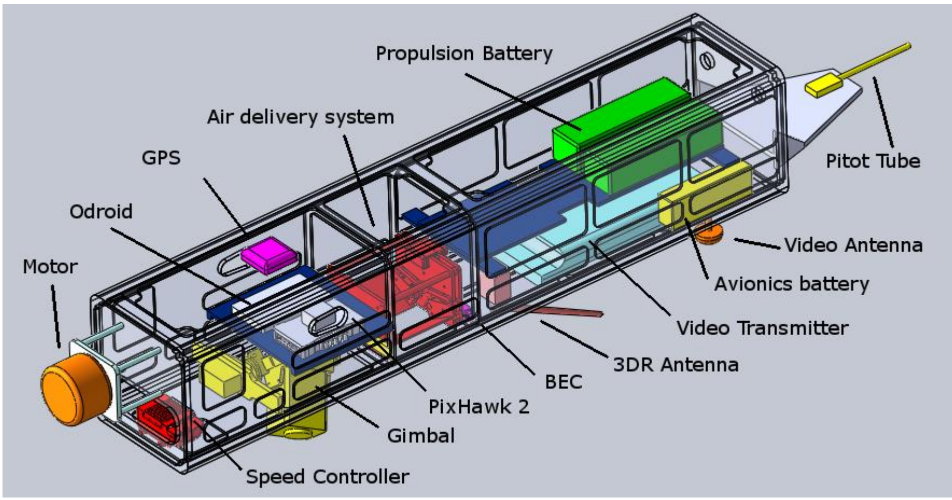

Figure 3: Internal Configuration of the Fuselage

The payload of the entire aircraft is divided into four compartments. The nose compartment houses a digital airspeed sensor and a servo which provides operation of the front wheel. The compartment directly posterior to the nose holds the main and supplemental batteries. They are secured with Velcro so that they can be taken out intermittently for recharging. The main compartment houses the cargo release mechanism, camera gimbal, antennae, command sensors, and the remainder of the electrical components. The final, most posterior compartment of the main body houses the motor for the rear propeller, securely held to the plywood firewall to prevent vibrations or precession during operation. An electronic speed control allow users or programs to modulate its effective thrust as needed and records voltage, current, and temperature.

The following tables 3-5 include the relevant metrics as measured for the plane.

Mass	Fuselage Length	Fuselage Volume	Flight Time (Endurance)	Tail Boom Length	Climb Rate
6.8 kg	1.1303 m	.70 m^2	40 minutes	1.22 m	3.3 m/s

Table 3: Plane's overall flight metrics

Wing Airfoil Type	Wing Planform Area	Wing Span	Taper Ratio	Leading Edge Sweep	Aspect ratio	Wing Loading
Clark-Y	.841 m^2	2.6 m	.6721	6.2 deg	8.0196	8.085 kg/m^2

Table 4: Wing Dimensions

Stabilizer Airfoil Type	Vertical Tail Area	Horizontal Tail Area	Vertical Volume Coefficient	Horizontal Volume Coefficient
NACA 0011	.046 m^2	.116 m^2	.05	.31

Table 5: Metrics relevant to empennage

The motor is a rear-running Scorpion SII 520kv with a 14x7 propeller. The propeller and motor combination were based on a data chart provided by the manufacturer listing the thrust efficiencies of various propellers given a 6s lithium polymer battery. The propeller is also sized to not hit the ground on takeoff, in part due to the rear skids on the tail.

Wing selection was based on projected weight and velocity, an estimate that stems from experience in previous years. A sample of the table we built is provided below in Table 6, which shows that given a C_{Lmax} of 1.2, stall speed of 12 m/s, and mass of 9 kg, the plane would need a planform area of about .8 m^2. However, once the plane was fully assembled, we found that the mass fell to 6.8 kg, which led to a stall speed of 9 m/s, matching the stall speed we recorded in testing of 8.5 m/s. The difference is probably due to underestimating the C_{Lmax} of the wings, since a Clark-Y airfoil has a C_{Lmax} of 1.3-1.4.

	Mass (kg)	6	6.5	7	7.5	8	8.5	9.5	9.5
Wing	0.825	9.46	9.84	10.21	10.57	10.92	11.25	11.90	11.90
Area	0.85	9.32	9.70	10.06	10.42	10.76	11.09	11.72	11.72
(meters)	0.875	9.18	9.56	9.92	10.27	10.60	10.93	11.55	11.55

Table 6: Stall speed in m/s for a given mass and wing area

Finally, the batteries were chosen based on the needs of the system, our projected flight time, and battery testing previously recorded by the Air Systems Lab here at UT Austin. In the end, we chose 2 2200 mAh, 3s ThunderPower (TP) Lipos for the primary avionics and a 1350 mAh, 3s TP Lipo for the secondary avionics. The propulsion system runs on 2 5000 mAh, 6s TP Lipos, which is the limiting factor for endurance at 40 minutes. This was intentional, the avionics should always outlive the propulsion system in case a gliding landing is required.

2.2 Autopilot and Avionics Overview

The plane uses a Pixhawk 2 Flight Controller and runs Arduplane 3.8. It has autonomous takeoff, flight, and landing capabilities, and allows for manual and automatic tuning of PIDs to create a more stable aircraft. The Pixhawk 2 was chosen to replace the APM 2.6 board from previous year, since the APM board did not have enough storage to fully run the newer versions of Arduplane, meaning that important features like antenna tracking and gimbal stabilization were not kept. The Pixhawk 2 also lets us both send and receive commands over multiple telemetry connections, such as both telemetry ports and the USB port, whereas the Pixhawk 1 and APM board do not. Finally, the Pixhawk 2 allows for Intel Edison integration, which could be useful in the future for obstacle avoidance and other algorithms.

For GPS, we chose to use the 3DR GPS and Magnetometer module, since both sensors benefit from being mounted on top of the plane. The I2C pitot tube is mounted through the nose where it samples undistrubed airflow, since the propeller is back-mounted. The Pixhawk 2 requires two power sources, one for the servo rail and one for the board itself. We use redundant power for each, with both the primary avionics and propulsion batteries providing power to the Pixhawk 2, and the primary and secondary avionics powering the servo rail. The secondary avionics battery is used exclusively for the servo rail and the RC reciever, meaning that if a short occurs elsewhere in the avionics circuit, the plane can still be landed by the safety pilot. Also, powering the Pixhawk 2 via the primary avionics and propulsion batteries by using power bricks offers us the ability to measure voltage and Amperage used, which lets the team determine when the plane needs to land.

Below is a wiring diagram (fig. 4) of the avionics system of our UAV. It should be noted that power, ground, and signal wires are not distinguished, and many of the pins are not readable as they are not relevant to wiring the plane. The numbers in the table correspond to components in the image.

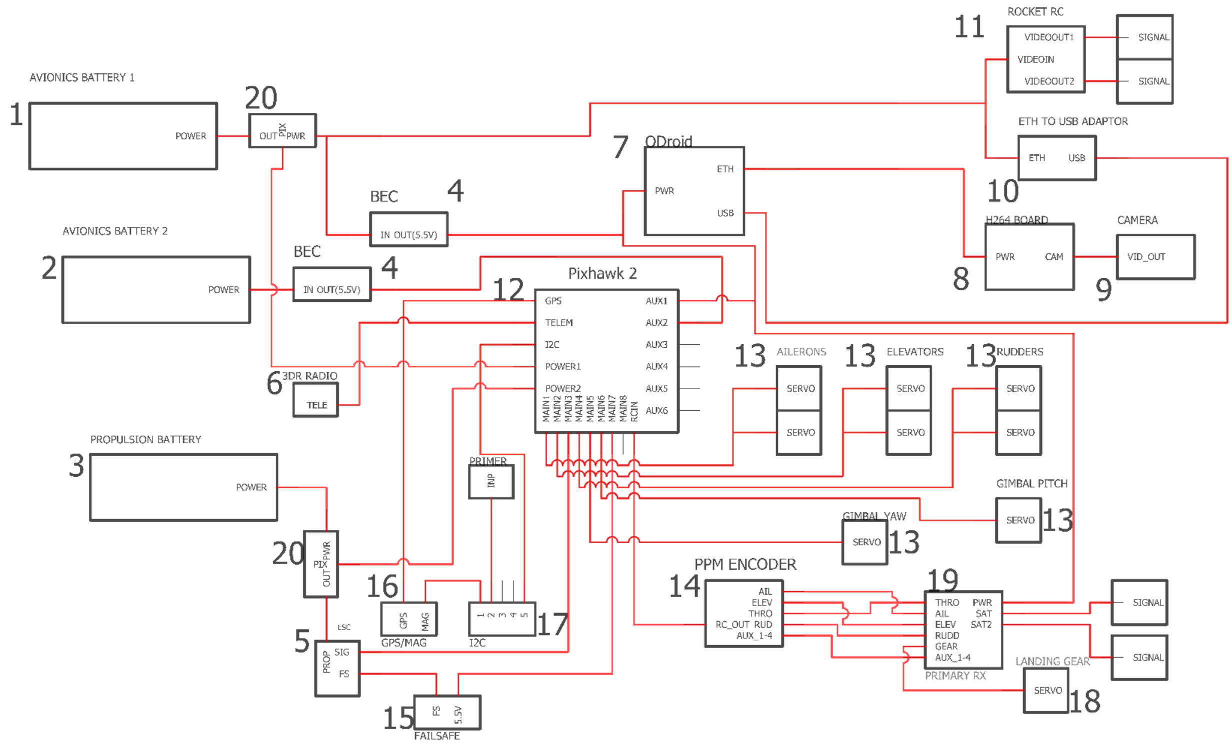

Figure 4: Wiring diagram for the avionics system

1. Primary Avionics battery	5. Electronic Speed Control	9. Camera	13. Servos 14. PPM Encoder	17. I2C Splitter
2. Secondary Avionics battery	6. 3DR Radio	10. Ethernet to USB adapter	14. PPM Encoder	18. Landing Gear
3. Propulsion battery	7. Odroid-C2	11. Rocket AC	15. Throttle Failsafe	19. Primary Rx
4. Avionics BEC	8. H2.64 Board	12. Pixhawk 2	16. GPS & Magnetometer	20. Power Brick

Figure 5: Wiring diagram reference table

For our ground station, we have two windows. The first is Mission Planner, seen in Figure 6 on left, which we use for general flight purposes, such as flight planning, parameter tuning, and uploading new firmware. The second is called Flight View, shown in Figure 6 on right, which contains the information required for the judges, allowing the Mission Planner pilot to focus on their job. The window displays waypoints, obstacles, no-fly zones, and altitude, airspeed, map, and plane location. Flight View is served up from an in-house server and also has view for image overlay and flight history.

Figure 6, left: Mission Planner during flight. Right: Flight View for a mission from the competition server

2.3 Imaging System

The camera is a Sony EV7100 block camera with an H2.64 converter board. It has 10x optical zoom and auto-focus/auto-exposure. This means that we can get quality images without dependence on the altitude or lighting conditions. Furthermore, it outputs 1080p video at 30 Hz with a shutter speed of 1/250th of a second, which is the maximum shutter speed that still allows in enough light for color images. Given that the lettering will be at least one inch in size, the minimum pixels per square inch for manual recognition becomes .62 for diagonal letters. This ensures that the partial color is carried through the diagonals and that full color is captured on the straight sections. For automatic image recognition, there should be at least 1 pixel per square inch of diagonal letter, or a 1 to 1 ratio to ensure that the full color is carried throughout the image without breaks in the letter. The below graph gives the field of view vs altitude line, where all points on or below the blue line enable manual image recognition, and all points below the orange line enable automatic image recognition.

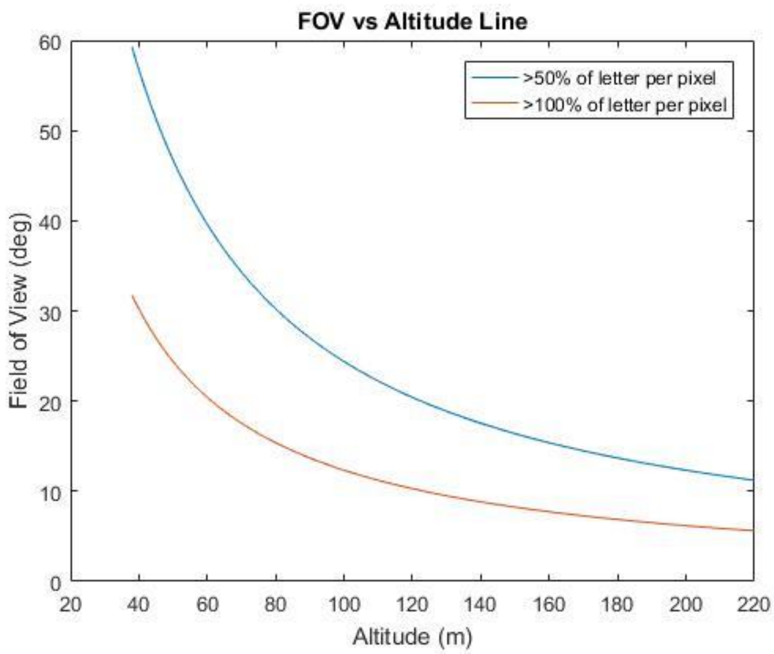

Figure 7: Field of View vs Altitude lines for the EV7100

The high shutter speed eliminates most blur, since only 2 meters, or .04% of the image changes per frame captured. To avoid losing data or blurry video, images are frame-grabbed by the ODROID-C2 onboard computer, time stamped, and queried by the ground station. If communications are lost during flight, the images can be retrieved via the Ethernet port on the side of the plane and processed normally.

The camera is held in the plane by a gyroscopically stabilized, two-axis, pitch and roll gimbal, pictured in Figure 8. The gimbal allows up to 45 degrees of rotation from center in all directions, and is entirely 3D printed. The main assembly of the gimbal, including the camera, is a self-contained unit that can be removed from the aircraft by 2 bolts, allowing for easy maintenance. We chose to use a pitch and roll gimbal to counteract the airplane’s movement on those axes and stabilize the camera. Yaw is in gimbal lock when pointing vertically down out of an airplane, making it less effective at stabilization.

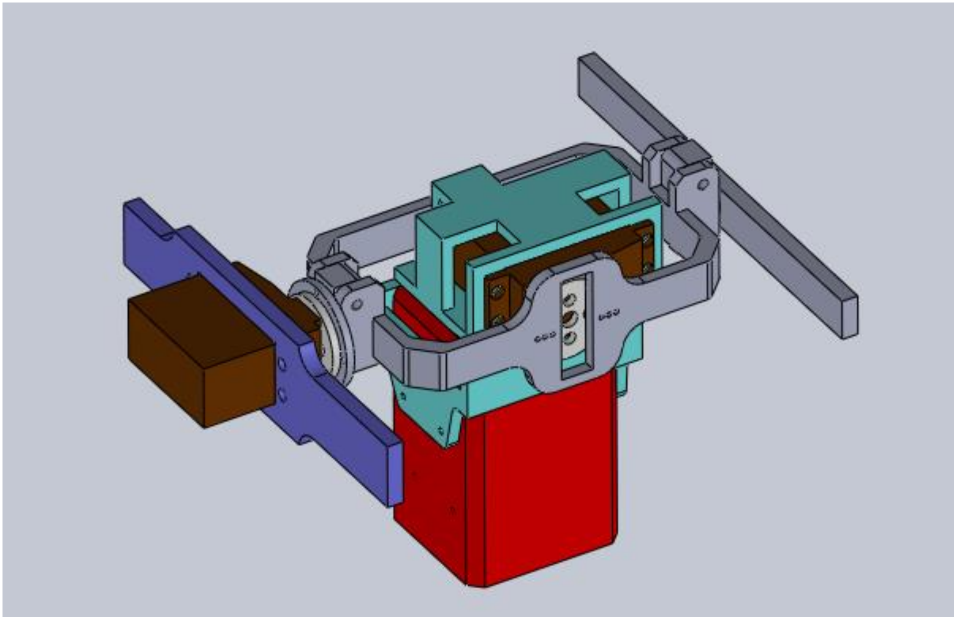

Figure 8: Gimbal and camera in Solidworks. Grey and light blue parts were printed in ABS.

2.4 Communications and Full Stack

Both the secondary flight screen and the manual target recognition screens are bundled into a program we call Flight View. Our goal is to build a ground control station from scratch, and currently we can view the aircraft's location and flight plan, adjust waypoints, and upload targets to the competition server. Multiple instances of this program can be opened on different computers and can sync together to form one ground station working over several computer screens. This allows for only one connection to the plane and the competition server, but for many people to use it at once. For our current purpose, we will be using both the flight screen and the manual image recognition screen on separate computers. The following diagram, Figure 9, shows a flow chart for the entire system.

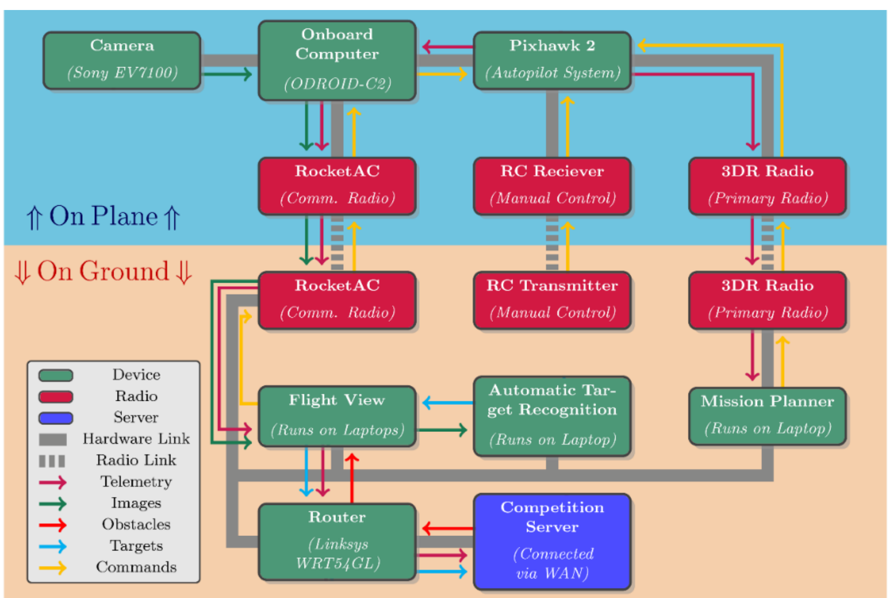

Figure 9: Flow chart for communications

Flight View is built on Electron by GitHub, which allows for HTML / CSS / JavaScript development for desktop applications. This web design approach allows for faster development for GUIs as well as in the backend. The program also uses a NoSQL document-based database to store telemetry over time to cross-reference it with an image taken at a given time.

The automatic image recognition script running on the ground and the Python image-storing scripts running on the plane both interact with Flight View through its REST API. Instead of using sockets, which can cause many unplanned issues such as handling messages incorrectly and dropped connections, this API exposes endpoints such adding images to the ground, retrieving images and telemetry, and positing targets to the competition server in a way that is safe even with dropped connections.

The python script running on the onboard computer stores images from the camera and requests telemetry from Flight View so that it can warp the image into true shape, and then sends both the original and warped images to it. The automatic image recognition script requests unprocessed images, and then sends back the targets it finds so Flight View can forward those to the competition server.

2.5 Obstacle Avoidance

Due to development time constraints, we will not be using a fully automatic obstacle avoidance system. Instead, the mission path will be modified manually to prevent the plane from hitting the obstacles.

In the previous year, we had attempted to update the waypoints on Mission Planner as the plane was flying. The obstacles were shown on a separate screen, and so there was some difficulty in updating the waypoints since the obstacle positions had to be referenced from another screen. This put a high load on the person flying the plane on Mission Planner, as they had other duties to fulfill, and the objective could not have been met in this way.

To resolve this conflict, this year the waypoints can also be moved on our additional GUI which also displays the obstacles. The person controlling the Flight View flight screen will be adjusting the path by moving individual waypoints, letting the Mission Planner pilot focus on the overall mission.

2.6 Object Detection, Classification, and Localization

Before images are processed they are warped to true shape if possible using the telemetry at the time the image was taken. During normal operation, the camera points downwards in flight. However, due to variations in the orientation it becomes helpful to warp them to make target identification and location easier. We use a bicubic interpolation in OpenCV onboard the plane to accomplish this with transparency in the corners after. North in every warped picture also points straight up in the new image. Images will not be warped if the horizon if visible. In most cases this should not be a concern, unless a heavy angle is needed for the off-axis target, then the original image will simply be used.

Figure 10, left: Original stabilized image. Right: Rotated and stretched image

Preprocessing:

Any arbitrary RGB image is processed through a color edge detector where the strength of each edge is determined by the maximum color gradient of adjacent pixels. A canny edge detection is also performed and used in combination to account for potentially blurry images. In general, this serves as a better edge detector than a sole canny edge detector as it takes advantage of the color differentiation of the targets. If the resulting number of connected neighboring pixel sets (referred to as blobs) exceed a preset upper bound, the edges are further refined at a varied threshold to remove excessive noise. The resulting blobs are considered potential targets, and are each standardized through a combination of region filling, dilation, and further noise filtering. Finally, the preprocessing is complete when each blob is assigned a bounding box and readied for classification.

Classification:

Shape detection is performed with a retrained version of Google's deep neural network model Inception-v3 using TensorFlow. The model consists of several computational layers, such as convolution, max pooling, and softmax. The original model's final classification layer is retrained with our dataset of training images, consisting of at least 500 scraped images and 1000 synthetic images for each of the nine shapes: circles, crosses, quarter circles, rectangles, semi circles, squares, stars, trapezoids, and triangles. After cross-validation, the resulting training accuracy is around 93%.

Each blob, defined by its corresponding bounding box, is classified with this pre-trained model, which outputs a shape classification and confidence score. If the confidence exceeds a given threshold, the shape's pertinent information is appropriately saved and the blob proceeds to alphanumeric detection. If it is a square, it is first redirected to QR detection, implemented using a standard open-source library. Alphanumeric detection (OCR) is performed using another retrained inception-v3 model. Its training set consists almost entirely of synthetic images up to 500 for each character, which are derived from 3 commonly used fonts. All alphanumeric training images are standardized with no rotation. Given a confirmed shape, the alphanumeric detection attempts to rotate the blob with fixed increments, and determines the highest confidence among all rotations, thereby assigning the alphanumeric value and the corresponding rotation of the shape. Lastly, the colors of the shape and the alphanumeric are determined by analyzing the HSV color histogram of the bounded box associated with the shape. The two most significant off-green colors are chosen as the colors of the target and alphanumeric.

In the case where automatic image recognition is not possible, such as images with low quality or taken at heavy angles, or if the algorithms do not catch a target, the manual recognition target system in Flight View is used.

The manual recognition screen consists of warped images overlaid on top of each other on an interactive map. To identify an image, the user simply right click and drags over the region where the image is, and then inputs the data needed to identify it. Then it is sent to the competition server.

Automatic recognition cross-references existing targets that have been previously submitted to avoid duplicates, and will not override images submitted manually.

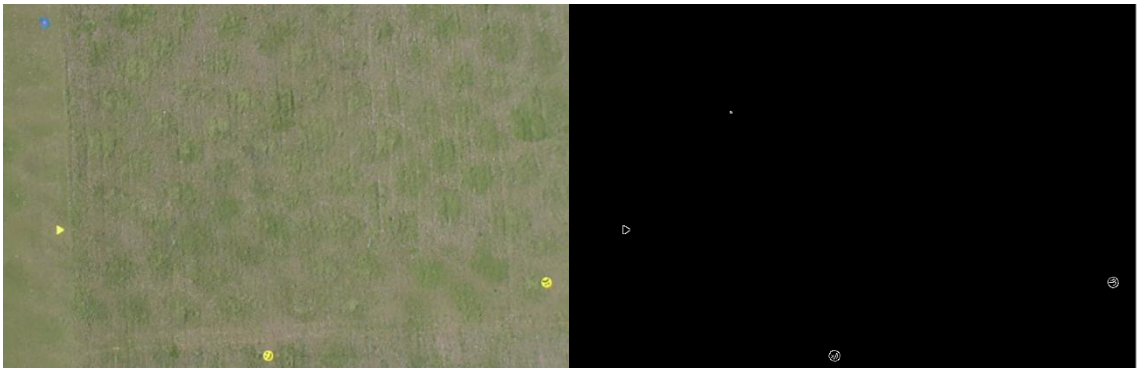

Figure 11: Original image (left); Color gradient edge detection (right)

2.7 Air Delivery

The air delivery system is a self-contained unit that is mounted to the plane for flight. The unit consists of the doors, a frame, and the servo that actuates the system. The system can be easily removed for maintenance and is attached to the plane by 4 bolts, 2 at the front and back that connect to small plywood tabs that are glued to bulkheads in the fuselage. The mechanism itself consist of 2 doors that slide out from under the package while rotating to face the walls of the aircraft, with no part of the mechanism protruding from the plane at any point in its operation. The system is powered by one micro servo. The primary reasons for creating a mechanism that folds the doors to the side is to eliminate the need for the servo to be powered to keep the doors closed, in case of a servo failure, as well as eliminate any strain on the servo during flight. The mechanism has minimal resistance when operating and naturally returns to the closed state when unpowered. A tradeoff of the system is that the package would be lifted upwards a small amount while the system is operating, requiring a small amount of free space to accommodate this upward movement of the package.

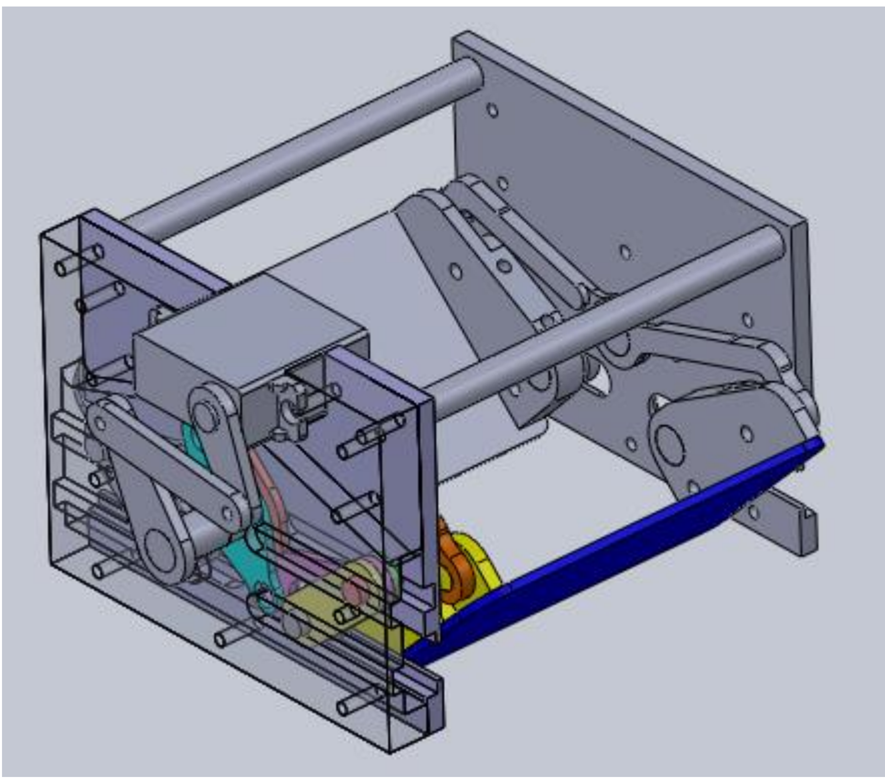

Figure 12: Air Delivery Mechanism

2.8 Cyber Security

The benefits to hacking, jamming, or misdirecting drones grow as the role of drones in everyday life increases. As a result, drone manufacturers and operators must take measures to both educate themselves on potential security threats and mitigate those threats.

The main vulnerabilities for any system are the communications endpoints. Simply put, every receiver has a frequency and protocol which can be taken advantage of, either by reading data, sending false commands/data, or jamming to block communication. For our plane, we have a GPS receiver, 3DR receiver/transmitter for autopilot communications, RC receiver, and WiFi receiver/transmitter for imagery plus autopilot redundancy. On the ground station, the only receivers are for WiFi and 3DR. Table 7, below, details the components, risks, mitigations, and fallback plans for the main security risks.

Component	Risk	Mitigation	Fallback Plan
GPS Receiver	Spoofing	uBlox receivers offer spoofing detection, which can alert the plane and tell it enter failsafe mode (RTL or Loiter).	Safety pilot takes over and lands the plane.
GPS Receiver	Jamming	GPS is oriented upwards and sits on a metal plate to prevent interference.	Plane enters failsafe mode when GPS is lost. Safety Pilot can take over if condition persists.
3DR Link	Monitoring	AES encryption, FHSS	None
3DR Link	Hacking	AES encryption, FHSS	Safety pilot attempts to land and, failing that, turns off transmitter to force failsafe, resulting in RTL and eventual crashing.
3DR Link	Jamming	900 MHz is regulated, so only malicious sources can jam.	Use WiFi link and Flight View for primary telemetry communications until link is regained.
RC Receiver	Hacking	RC transmitter and receiver are paired before competition, and FHSS prevents others from hijacking the link without knowing the initial seed.	Attempt to change the flight modes via Mission Planner so that the autopilot is controlled via the telemetry link alone, then land.
RC Receiver	Jamming	RC receivers are placed on either wing to decrease ambient noise and provide redundancy.	Plane will return to land after 30 seconds and terminate flight after 3 minutes.
WiFi Link	Monitoring	WPA2 security and encryption	None
WiFi Link	Hacking	WPA2 security and encryption	Safety pilot attempts to land. If this fails, the safety pilot then turns off transmitter to force failsafe, resulting in RTL and eventual crashing. All computers on network are secured with passwords.
WiFi Link	Jamming	Rocket AC automatically switches to lowest noise frequency in 5.8 GHz band. Also using 20 MHz bandwidth instead of 40/80 MHz to decrease jamming chances.	Imagery is saved to Raspberry Pi onboard and can be accessed after flight. Telemetry is routed through MavProxy to Flight View

Table 7: List of main security risks for the plane and ground station

Particularly important to note from the above list are GPS spoofing, 3DR hacking, and WiFi hacking. GPS spoofing, while difficult, has been demonstrated just recently at the Kremlin, where Uber passengers found themselves charged airport fares when sightseeing. However, spoofing can be more malicious. It could result in a plane registering that it is 500 meters too high and crashing into a civilian building. As a result, the team plans to spend more time focusing on this in the future. As for 3DR hacking, until recently there was no encryption on for telemetry, which resulted in an extremely vulnerable system. Now, however, the link is even more secure than a WiFi link, since there are not any real backdoors. Finally, if the WiFi were to be hacked, the hacker would only have access to data being transmitted, since the computers themselves are all secured.

3 Testing & Evaluation Plan

3.1 Developmental Testing

Developmental testing was performed for all new subsystems, since those were the most important to test before designing the other subsystems around them. The antennas, Rocket AC's, blob detection algorithm, and TensorFlow all necessitated testing prior to use.

3.1.1 Antenna Testing

Due to previous video transmission challenges, we decided to perform extensive antenna testing to determine the optimal antenna configuration. Clover-leaf, helical, patch, and dipole antennas were all tested. Antennas were tested in pairings on the Rocket AC adapters at ten feet in four configurations: optimal directionality, worst-case directionality for each individually, and both in worst-case. The two clover leaf antennas paired with either themselves or the helical antenna had better signal strength in almost every configuration than anything with the dipole or patch antenna from the previous year. As a result, we chose to use two clover-leaf antennas on the plane and a clover leaf and helical on the ground station, where the helical can be pointed towards the plane with an antenna tracker.

3.1.2 Rocket AC

The Rocket AC's and antennas were tested by moving the transmitter and camera system to various distances with line-of-sight from the receiver. The two antennas were pointed towards each other, and video and transmission rate were monitored to ensure no drops in quality. The important parameter is the power received, since one would typically expect for -90 dBm to be the noise floor in a noisy environment, such as UT Austin.

Distance (ft)	10	60	200	325	400	800
Power Received (dBm)	-24	-50	-55	-60	-70	-75

Table 8: Distance vs power test data for Rocket AC

Given that the mission could result in the plane flying as far away as 2400 feet, the tests showed that the doubling rule for distance would end up with the transmission system being on the edge of working at the edges of the field. As a result, we decided to add in the ability to pull images off the computer easily on landing by installing an ethernet port into the fuselage.

3.1.3 Blob Detection and TensorFlow

Image testing was primarily done using video feed captured from the current and previous year's flight tests, past competitions, and aerial pictures of targets from buildings. From a sample of 20 frames taken from this year's flight tests, we detect 5 out of 6 unique targets, with 4 of them correctly classified. Alphanumeric detection was only successful in recovering one correct letter, and orientation detection was unsuccessful since it relied on the accuracy of the alphanumeric detection. However, up to 4 unique objects such as airstrips and trees were falsely detected and classified as shapes. To remedy this issue, we employ a false classification filter that analyzes the color distribution and removes any bounded regions with a near uniform color composition.

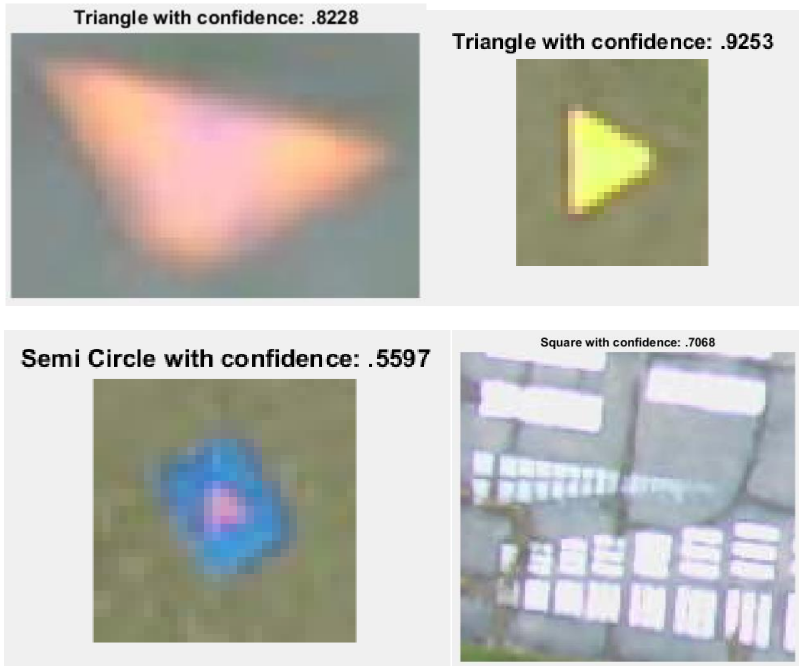

Figure 13: Shape classification outputs on tested images (left to right): 1) Correct classification, 2) Correct classification, 3) Incorrect classification, 4) False detection.

3.2 Individual Component Testing

3.2.1 Autonomous Flight

Autonomous flight was tested in the lab by flying SITL with both Mission Planner and Flight View. In Mission Planner, the goal was to setup a mission based on a set of predetermined waypoints and fly it, including takeoff and landing, using the latest Arduplane firmware. The mission was a clear success, with the plane appearing to have hit all waypoints in order. In Flight View, the goal was to ensure that the simulated plane hit the waypoints as read by the server, where it was successful. Unfortunately, the GUI has not been tested with the server, since the methods to rotate images are not currently written.

3.2.2 Imaging System

The imaging system was tested to determine that it could store video at a rate of 2 Hz and transmit it at a rate of 30 Hz. The second is helpful for manual image recognition and determining where to fly next, as well as being a metric for a data rate that is far higher than the system requires. In flight testing, the plane achieved both, with no signal drop at up to 1500 feet, furthest we can fly legally at our field, and consistent image storage on the Odroid-C2.

3.2.3 Plane Communications

The plane must be able to transmit and receive on both the 900 MHz and 5.8 GHz bands throughout the whole mission, plus receive only on the 2.4 GHz band. To test this, we had two senior design teams power on all of their systems right next to our ground station with our plane out on the runway. While we saw no drop in the lower two bands, the 5.8 GHz signal dropped completely. To remedy this, the signal bandwidth was reduced from 80 MHz to 20 MHz, meaning that the noise generated by the analog transmitters could be avoided more easily by switching to another frequency in the 5.8 GHz band. A direct result of this is a loss in data rate, but there was no visible drop in video quality. As a result, we believe that we are ready for the noisy environment out at Webster Field. As a side note, if everyone is on WiFi then we should not have a problem at 80 MHz, since WiFi is regulated to avoid jamming itself whereas analog video maintains no such qualms. On the same token, analog video systems will probably have lots of problems out at the field.

3.2.4 Object Detection/Classification/Localization

Object detection requires edge detection sensitive enough to create a tight bounding box. However, our tests produced significant noise and also detected unwanted objects such as runways or trees. Our solution has been to dynamically remove noise depending on the number of blobs detected, and also filter any blobs that are too big to be potential shapes given the zoom level. To account for varying conditions, our training images are randomized with different brightness, crops, scaling, rotations, gaussian blurs, and hues. The model is trained with a 80-20 train-test split, and has a validation accuracy of 93%. Localization was tested by ensuring that each frame is correctly associated with a corresponding time stamp, which in turn is associated with GPS coordinates. Taking the center of the image as the plane's location, we interpolate the location of the shape from its distance to the center. The hardest task is to vary the distance scales with respect to the camera's zoom and the plane's altitude. From our tests, a stable altitude and a constant zoom level have had the most accurate localization results.

3.2.5 Server Communications

All endpoints exposed by the competition are tested in Flight View, with both successful and unsuccessful requests to mitigate any unexpected behavior at the competition field. We used the docker image provided with an automated testing environment on TravisCI to ensure the communication library was not broken between commits.

3.2.6 Air Delivery

To determine how we would ensure safe delivery of the payload to the target after it departs the plane, we conducted several drop tests from the top of a 90 foot tall building. These tests were conducted late at night with several team members clearing the area where the bottle would land to mitigate risk. We began with only the water bottle and found that the top unscrewed on impact. For the second test, we added 0.25 inch thick foam padding to the bottom with a stream attached to the top for stabilization and visibility, which was successful. Subsequent flight testing in this configuration saw that the bottle both fit into the delivery mechanism (see Figure 12, Section 2.7) and did not break on impact. Future testing will be focused around refining accuracy in different wind speeds and testing with other brands of water bottle.

3.3 Mission Testing

Mission testing is currently underway, and systems are tested on a rolling schedule as they become ready. We were limited at the beginning of the year due to previous year's plane breaking in transit. With the new plane, we have successfully tested autonomous waypoint capture, air delivery, and the imaging system. Each test includes the previous ones, meaning that we progress towards performing the full mission every time. Future tests are as follows:

Friday, Apr 21 – Test server telemetry upload/download, obstacle download and display

Friday, Apr 28 – Test image display screen for manual recognition with server upload

Friday, May 5 – Test automatic image recognition in conjunction with manual image recognition

Following tests during dead week and exam week will be full mission tests with the whole team, at which point the FRR will be recorded. Any systems which are found to be unreliable will continue to be improved and tested until the plane is shipped in June, since multiple students will be remaining in Austin throughout the summer.

4 Safety, Risks and Mitigation

First and foremost for safety and risk mitigation is to develop a plan and checklists. For plane design and development, a Gantt chart must be used to make sure that critical deadlines are not missed. For flight tests, checklists are maintained and followed for flight preparation, packing, transportation, setup, and takedown. A flight plan is created for each flight to make sure all objectives are met for the flight. If objectives are not met, the reasons must be recorded so that they can be fixed before the next flight.

4.1 Developmental Risks & Mitigations

Table 9 gives the consequences of the encountering the risks listed in Table 10, which describes risks related making poor design decisions and how they will affect development and schedule.

Consequence	Detail
Catastrophic	Extremely long schedule setback (2+ weeks); Permanent damage to equipment; Personnel or third-person injury
Critical	Long Schedule setback (1-2 weeks) ; Structural damage that does not allow for immediate flight; low-medium degree of personnel injury
Small	Schedule setback (3-6 days); small equipment damage (still able to fly); low degree of personnel injury
Marginal	Relatively short schedule setback (0-2 days); very low-degree equipment damage (can be fixed immediately)

Table 9: Explanation of "Consequence" columns in the following table

Risk	Description	Consequence	Mitigation	Fall Back Plan
Not enough internal volume to install and access components	A minimum volume is needed to fit all the internal components and their necessary wiring.	Critical	When selecting a fuselage, we estimated the internal volume required and applied 50% safety margin.	Change to components which take less internal volume.
Aircraft's weight is too high	The aircraft's weight exceeds estimations, which negatively effects its flight performance	Critical	A mass properties Excel sheet tracked the mass and location of all components and their wiring.	Optimize the motor and propeller size to fly at a higher speed.
Center of mass (COM) is too far away from the quarter-chord of the wing	The aircraft's center of mass (COM) must be close to the quarter-chord of the wing. If the COM is behind the center of lift (COL), the plane will turn uncontrollably.	Small	A mass properties Excel sheet tracked the mass and location of all components and their wiring.	Rearrange the internal components to balance the aircraft. Add a ballast as a last resort.
Electrical malfunction/ damage	Exposing electronics to excessive voltage or current can damage them.	Critical	Read each components' documentation to ensure the correct voltage is supplied. Follow a circuit diagram for installation. Ensure all wires are properly insulated.	If damage occurs to any electronic component, identify the cause of the problem, and ensure it does not happen again.
Not enough funds to finance the project	Cannot buy critical components later in year	Critical	Track expenses in an Excel document	Fundraise for more money

Table 10: Developmental Risks and Mitigations

4.2 Mission Risks & Mitigations

Table 11 describes risks to the mission

Risk	Description	Mitigation	Fall Back Plan

Loss of Manual Control	The ground crew may lose control of the plane by exceeding its max range or depleting the batteries. This may result in the aircraft crashing	Range tests before flight. All battery voltages will be tested before flight. Ground crew will plan to land with a 20% battery reserve.	Safety Pilot will take control of plane. If safety pilot is unable to connect, plane will return back to home. If no connection is made within 30 seconds, the plane will enter fail safe mode.
Pitot Tube Failure	Pitot tube does not return accurate value for airspeed.	Pitot tube will be calibrated and tested before flight.	Change setting in the flight control software to rely on GPS for groundspeed.
Loss of WiFi Link	WiFi link is for telemetry and imagery	Use antenna tracker and ensure clear line of sight	Use Mavproxy to send telemetry data to Flight View. Download images from plane after flight for processing
Loss of 3DR Link	The primary link between Mission Planner and the autopilot	Maintain clear lines of sight with plane and use USB current booster	Use Flight View to tell plane to return to home and wait until the primary link is regained. Try to reroute Mavlink to Mission Planner through the 5.8 GHz link with Mavproxy
Failing to meet the mission time limits	Failing to meet the 20 minute setup or 45 minute mission time limit.	Mission will be repeatedly rehearsed before the competition. Ground station will record the amount of time left for the setup and mission.	If the 45 minute mission time limit is about to be reached, cease all mission objectives land the plane immediately.
Wind Interference	High winds will change the plane's flight path and may cause it to miss waypoints or violate the no fly zone boundary.	The ground station flight planner will monitor wind direction and speed. He will adjust the flight plan to ensure the plane does not violate the no fly zone.	If the no fly zone boundary is violated, then return inside the boundary as soon as possible. If a required waypoint is missed, adjust the flight plan to hit the waypoint, correcting for the high wind.
Air Delivery Mechanism (ADM) Malfunction	The air delivery mechanism (ADM) may malfunction, wherein the servo burns out or water bottle gets caught.	The ADM will be tested before flight to ensure that any ribbon does not get caught in the mechanism	Perform the rest of the mission as usual.

Table 11: Mission Risks and Mitigations

4.3 Operational Risks & Mitigations

Table 12 describes the risks involved in operating an aircraft

Risk	Description	Mitigation	Fall Back Plan
Injury from spinning propeller	The spinning propeller can cause serious cuts.	Anyone having contact with the plane during setup will wear gloves and safety glasses. Checklist manager verifies mitigation compliance.	First aid kit will be easily accessible. Phone emergency responders if necessary.
Injury/property damage from plane crash	The plane may fall out of the sky and land at an unpredictable location	Range test to verify connection strength. Batteries tested to be at full capacity before flight. Safety Pilot will take control if autonomous control is lost.	First aid kit will be easily accessible. Phone emergency responders if necessary.

Injury from electric shock	The batteries can potentially deliver a painful electric shock and burns. These effects may be amplified if exposed wires combine the voltage of the batteries.	Check for exposed wires and battery damage before the flight. Do not use electrical tape for long-term fixes.	First aid kit will be easily accessible. Phone emergency responders if necessary.
Injury/property damage from fire	Batteries can potentially explode and there may be a short in the electronics. These can cause a fire.	Check for exposed wires and battery damage before the flight. Use a power monitor to verify the current for the avionics system is drawing less than 20 amps under load, and propulsion system is drawing less than an amp while motor is off. Checklist manager verifies mitigation compliance.	Fire Extinguisher. First aid kit will be easily accessible.

Table 12: Operational Risks and Mitigations