MIAMI UNIVERSITY

COLLEGE OF ENGINEERING AND COMPUTING

ECE 448/449

Solar Powered Vertical Take-Off Airplane

Capstone Paper

Ryan Gracia-Rodriguez, Evan Johnson, Thomas Warrick, and Jacob Gero

Department of Computer and Electrical Engineering
College of Engineering and Computing, Miami University, Oxford, OH, 45056

Advisor: Mr. James Leonard

6 May 2021

Table of Contents

Intro/Problem Definition

The team, composed of Jacob Gero, Ryan Gracia-Rodriguez, Evan Johnson, and Thomas Warrick and advised by Mr. James Leonard, was to design and assemble and fly an aircraft at the 2021 Midwest Regional UAS Competition sponsored by NASA. This aircraft was required to have vertical take-off and landing (VTOL) capability, utilize a solar battery charger, fly via autopilot, drop glow sticks on a visible target from an altitude of 200 feet, and a video camera to aim the glow sticks at a ground target.

Project Background

A Solar Powered VTOL aircraft such as the one the team is building has many applications: Military usage, Search and Rescue operations, and different types of general aircraft usage. The military could benefit from having a VTOL aircraft that can keep itself charged and flying for large amounts of time. Scouting and surveillance missions can use an unmanned aircraft/drone, rather than sending a soldier. This would eliminate the risk of a casualty and allow the aircraft to not be limited by human factors. An aircraft designed to charge in-flight using solar panels would have an extended range compared to current aircraft of comparable size. Search and Rescue teams could use an aircraft like this to survey a large area over a long time

Designing the New Aircraft

At the end of the last semester the team knew they needed to build a larger airframe to meet all of the design criteria. They had originally planned to develop a flying wing as the large wing area and short wingspan would provide ample area to mount solar panels while maintaining a size of aircraft that could be easily transported. After researching RC flying wings the team decided to use the PW1211 airfoil. Stability simulations showed that the aircraft as a flying wing alone had little yaw stability so a NACA 0009 airfoil was chosen to act as a vertical stabilizer.

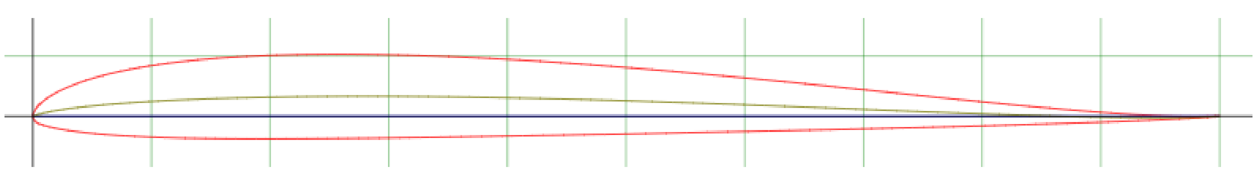

Figure 1: PW1211 Airfoil

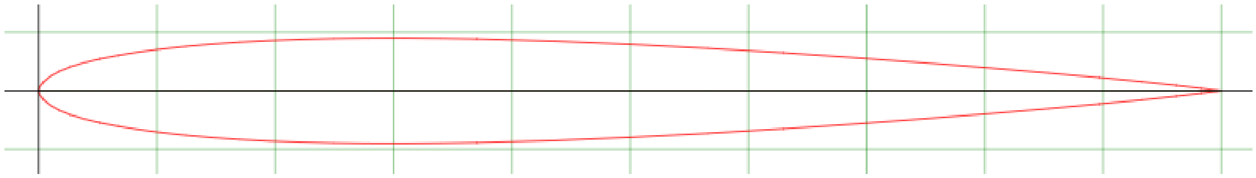

Figure 2: NACA 0009 Airfoil

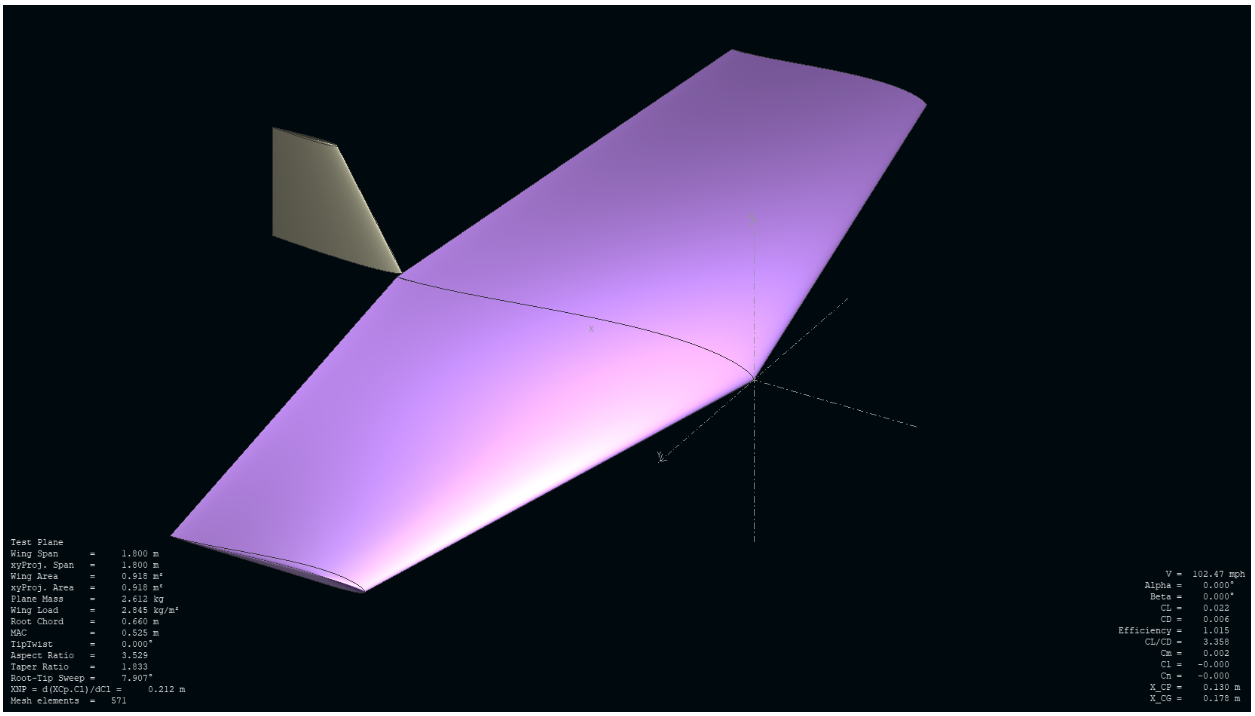

Figure 3: Flying Wing Design

Since the team had almost no aircraft design experience they decided to reach out to Peter Sripol, a professional RC aircraft builder, for advice. He informed the team that they correctly decided to build a larger aircraft but he suggested that they design a high aspect ratio aircraft rather than a flying wing as it would have better performance. Following this advice, the team began to design a high aspect ratio aircraft.

The aircraft would need to have a high lift to drag ratio so that it can cruise at a low power setting. A minimum wing area of 1400 sq in put the wing loading of the aircraft in the glider category which would provide the efficiency the team needed. They then began to research popular airfoils for gliders in this size range. The AG25 was used on many FPV gliders so the team decided to build the aircraft around this airfoil. Unfortunately when designing the physical components the AG25 was too short to provide the necessary strength. Consequently the team needed to change the airfoil to the AG24, a thicker airfoil with similar aerodynamic parameters. The NACA 0009 airfoil was once again chosen to act as the base airfoil for the horizontal and vertical stabilizers.

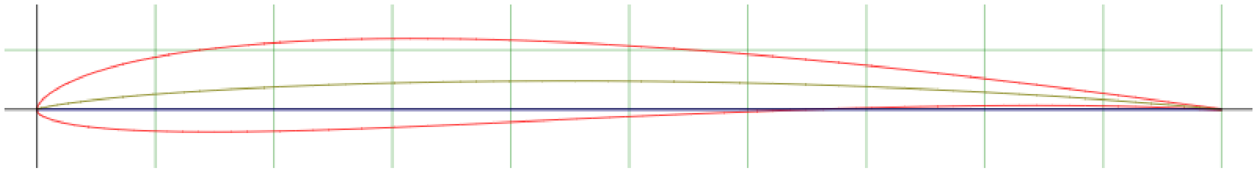

Figure 4: AG25 Airfoil

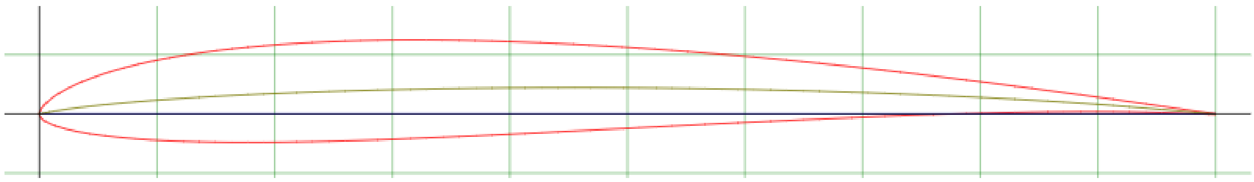

Figure 5: AG24 Airfoil

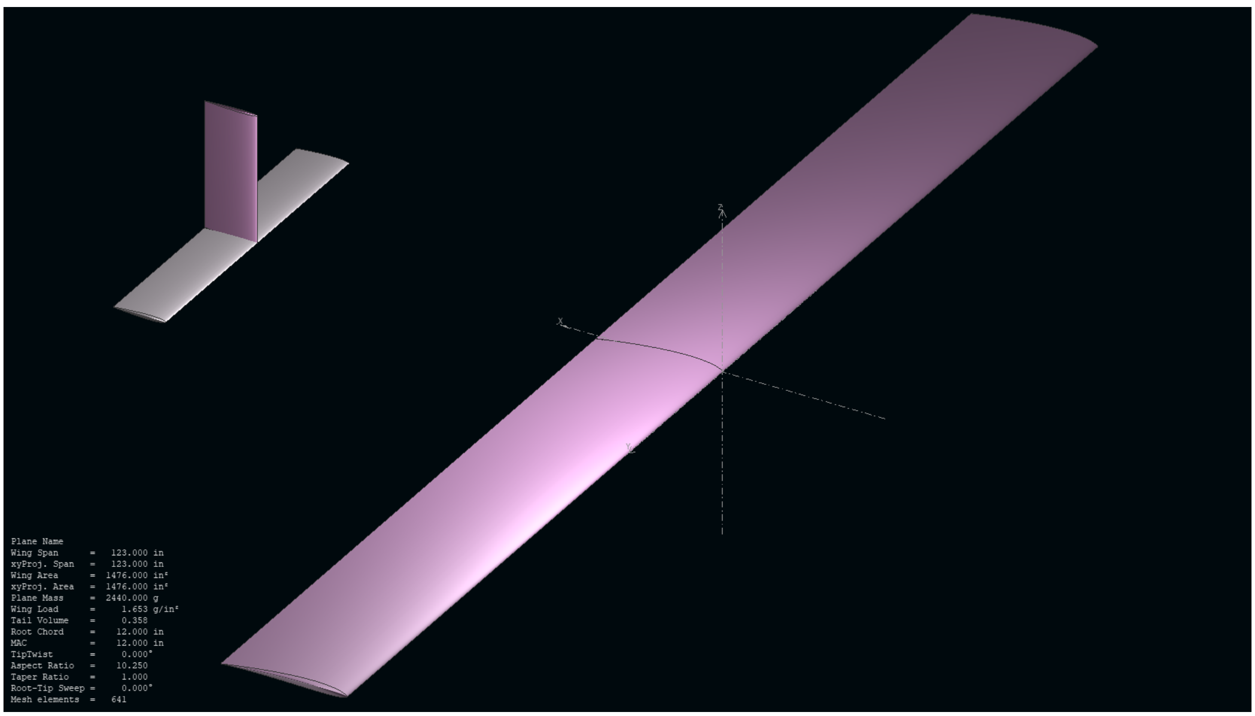

Figure 6: High Aspect Ratio Aircraft Design

A performance and stability analysis was performed on the designed aircraft to determine the proper location of the center of gravity, the distance from the leading edge to the empennage, and the estimated stall speed of the aircraft at different angles of attack. The team found that when the empennage was 44 in from the leading edge, a center of gravity located at roughly 4 in behind the leading edge provided the best stability performance.

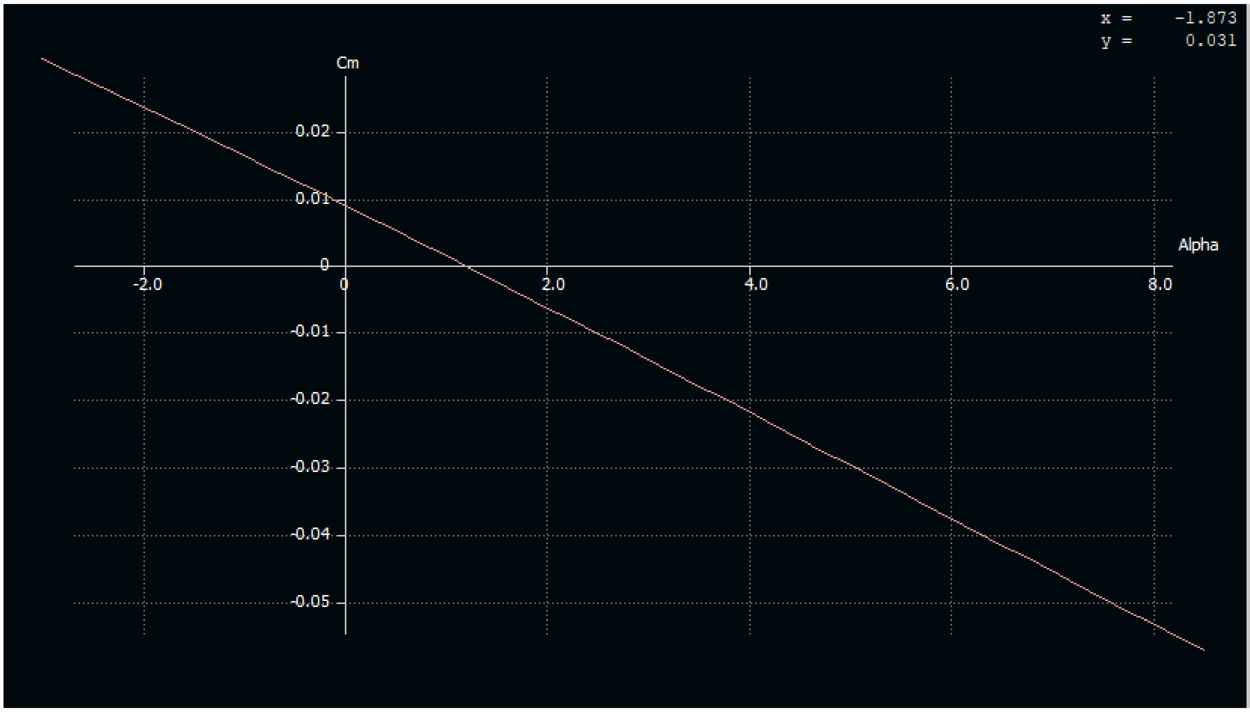

Figure 7: Pitching Moment vs Angle of Attack

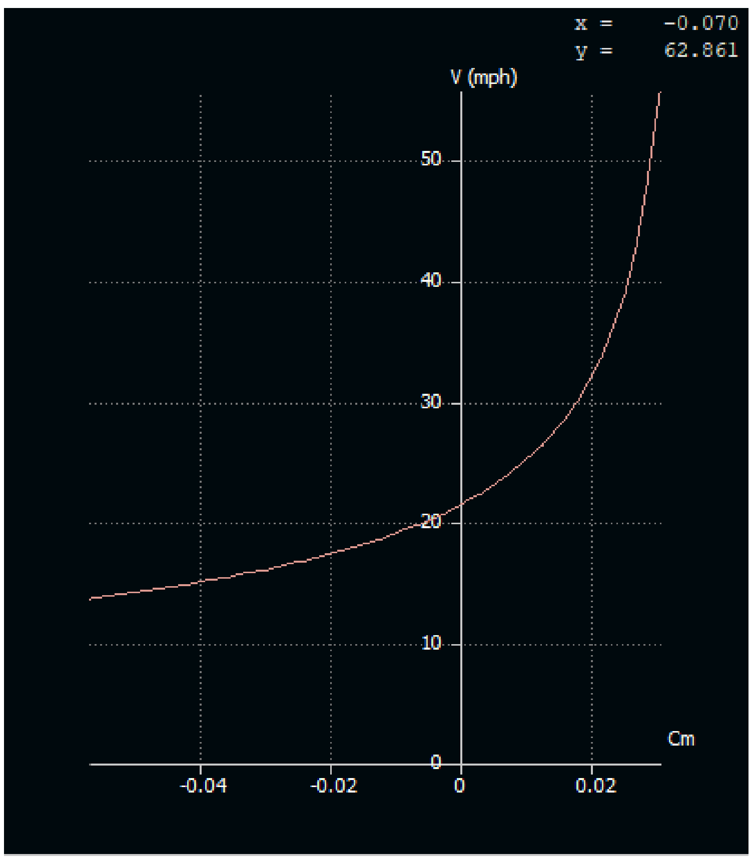

Figure 8: 21 mph Glide Speed Estimation

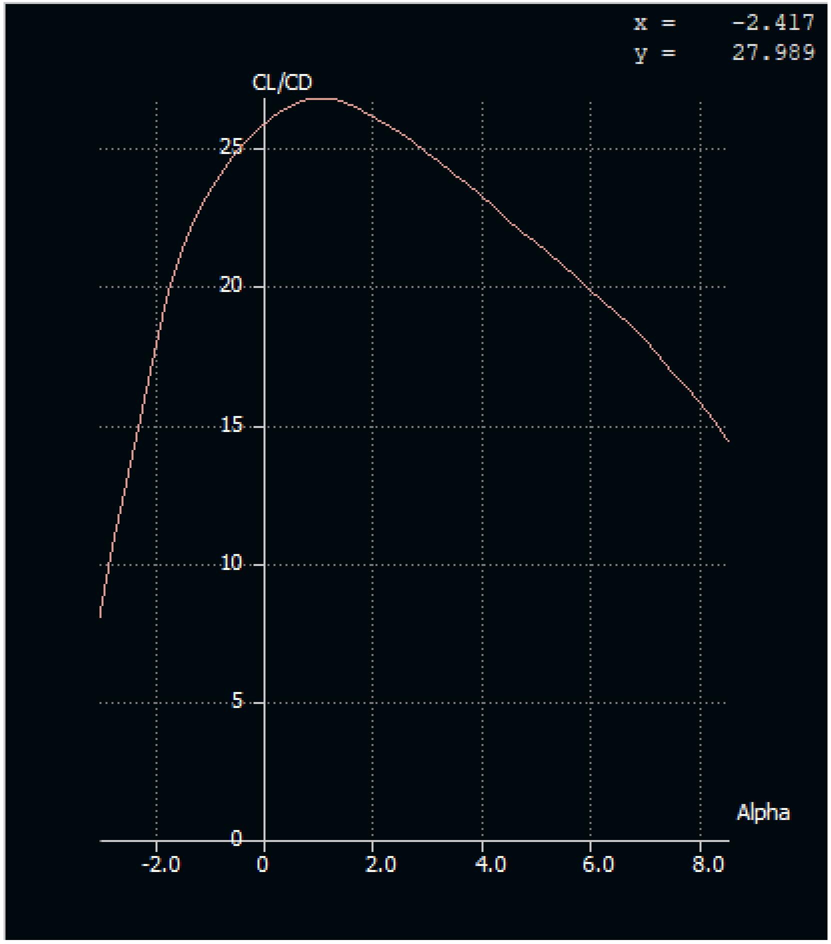

Figure 9: Coefficient of Lift to Drag Ratio vs Angle of Attack

After simulating the aircraft the team felt comfortable moving forward with the physical design and assembly of the aircraft. A 4 in aft center of gravity would be used and a transition speed of 27 mph, glide speed plus a 25% error margin, would be set in the aircraft parameters.

Modeling and Construction

After Ryan Gracia-Rodriguez finished designing the airfoil and running simulations, the next step was to make the design manufacturable. The team had to evaluate the carbon fiber tubes they were considering to use to verify that they met the strength requirements for the design loads. During this time they discovered that they needed an 18 mm carbon fiber rod for the center wing section spar and 16 mm carbon fiber rods to support the outer wing sections. The original airfoil design was too narrow to allow for adequate strength when a hole was cut to allow the 18 mm rod to pass through. As a result of this inconvenience they needed to use a different airfoil from the original design. Using CAD software allowed them to catch these issues in advance and as a result it helped to expedite the design process by reducing unexpected delays.

Figure 10: AG25 Airfoil designed in Fusion 360

The CAD software they used to design the components for the wing and the tail is called Fusion 360 and is a component of the Autodesk suite. To begin the design they imported a 2D model from the Airfoil Database for both of the required airfoils. In order to reduce weight they designed holes to make the most efficient use of materials. After the 2D modeling was complete they extruded the two dimensional drawing into a three dimensional part. From that 3D component they created a model representation of the skeleton of the aircraft. This allowed them to make sure that all of the parts will fit the way that they intended. This also allowed them to get an accurate count of the number of parts that they need to match the dimensions of the final assembly.

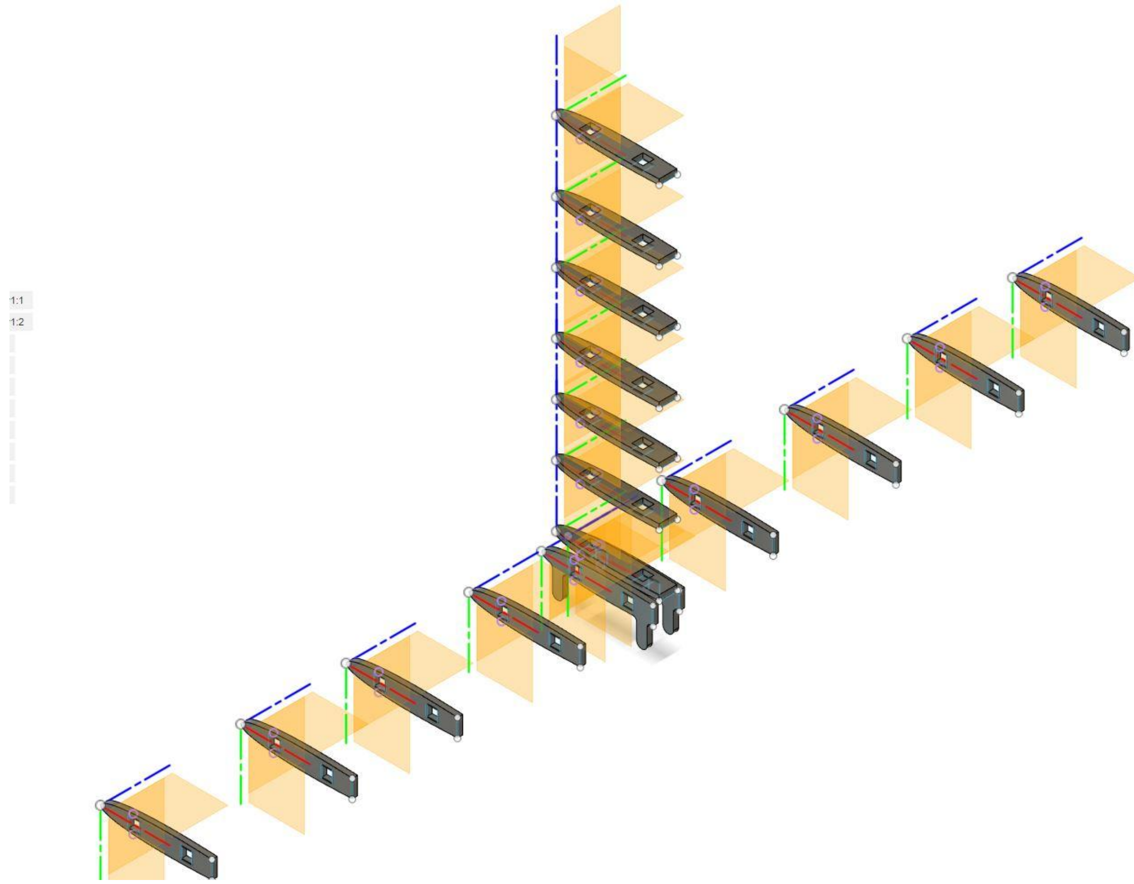

Figure 11: Fusion 360 Model of the Empennage

The file was then exported as a DXF; this file type is compatible with the CAM (lightburn) software that generates the G-code used to run the laser cutter. The G-code tells the microcontroller in the laser how fast to move the laser head and in which direction as well as the power setting that needs to be used. The laser may not cut the entire way through the material in 1 pass. If that is the case the number of passes can be increased to reach the desired cut depth.

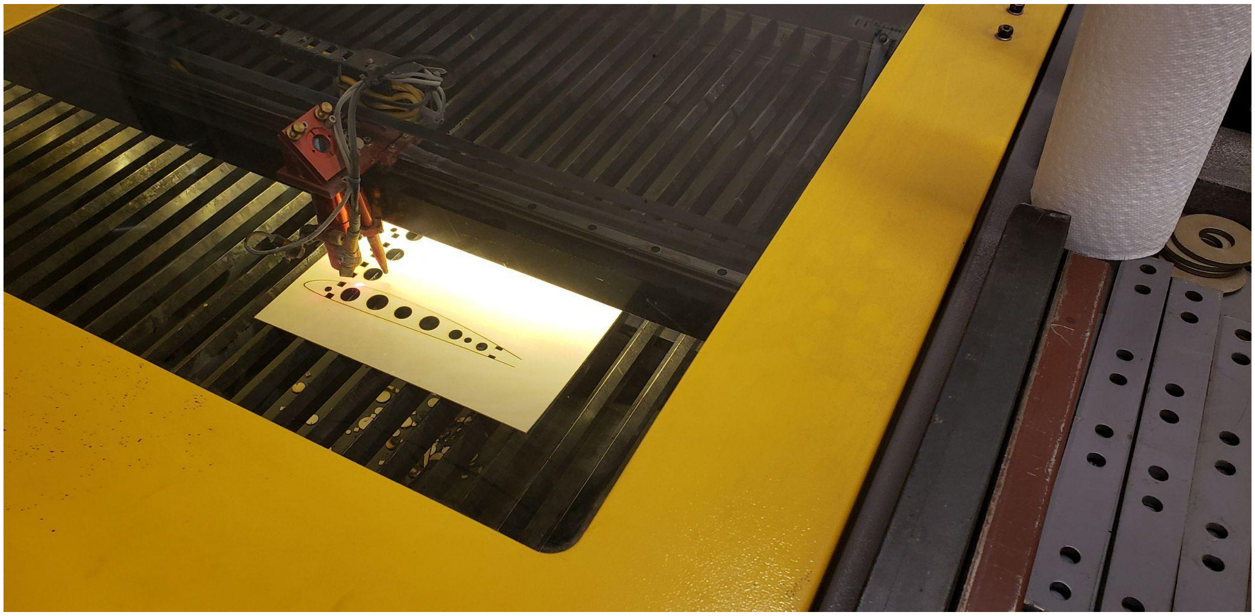

Figure 12: Test Cut on the Laser Cutter

Assembly

After the parts were cut out the team started to assemble the aircraft and realized that they needed to create an assembly fixture to allow consistent spacing and to keep the wing ribs normal to the spar. The ribs were also supported below to keep a constant angle of incidence along the span of the wing. This allowed the team to insure aircraft stability and decrease the manufacturing time. As a result of spending the time to refine the manufacturing procedures they could assemble a new wing in as little as 1 day in the event of a crash.

Figure 13: Aircraft Skeleton Suspended by Outer Wing Spar

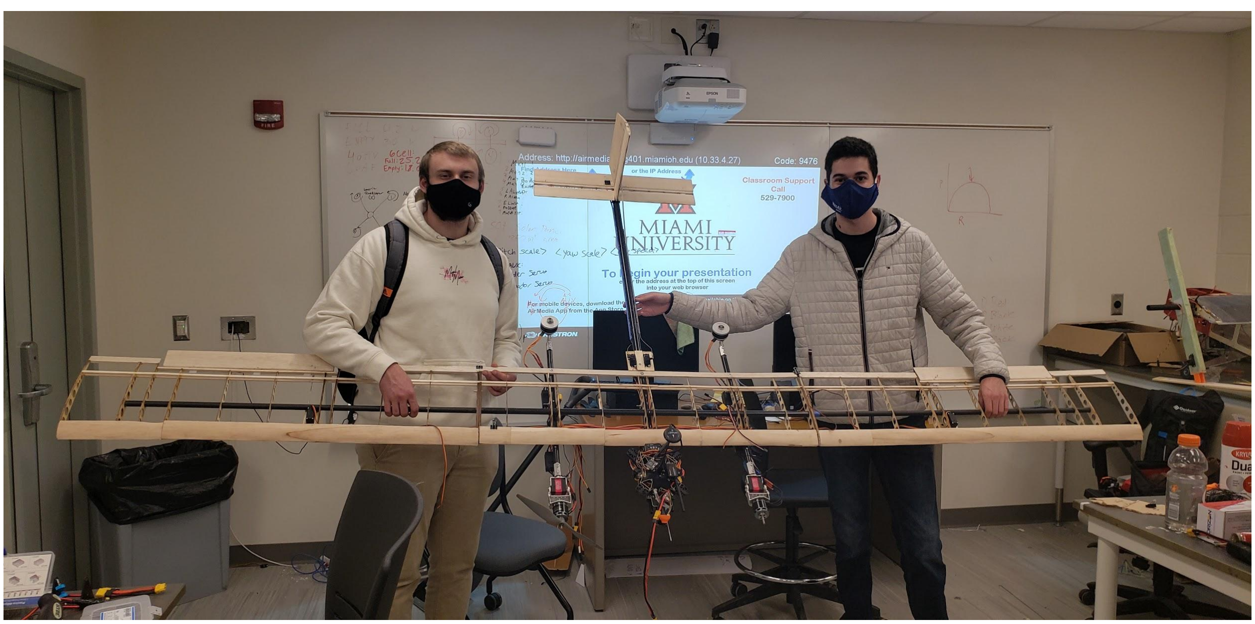

Figure 14: Assembled Aircraft Skeleton

Electronic Speed Controller Calibration

The electronic speed controller (ESC) is a circuit used to regulate the speed of an electric motor. In order for a quadrotor to maintain stability all of the ESCs must have the same behavior. It is possible to program each ESC to insure it will not have any unwanted behavior. The aircraft uses AeroStar Wifi ESCs which, as the name suggests, can be programmed over wifi. Using their app, the team was able to control many operating parameters of the ESC such as the PWM Frequency, Battery Type, Brake Disable, and Motor Timing. Figure 6 shows the parameters for the team's ESCs.

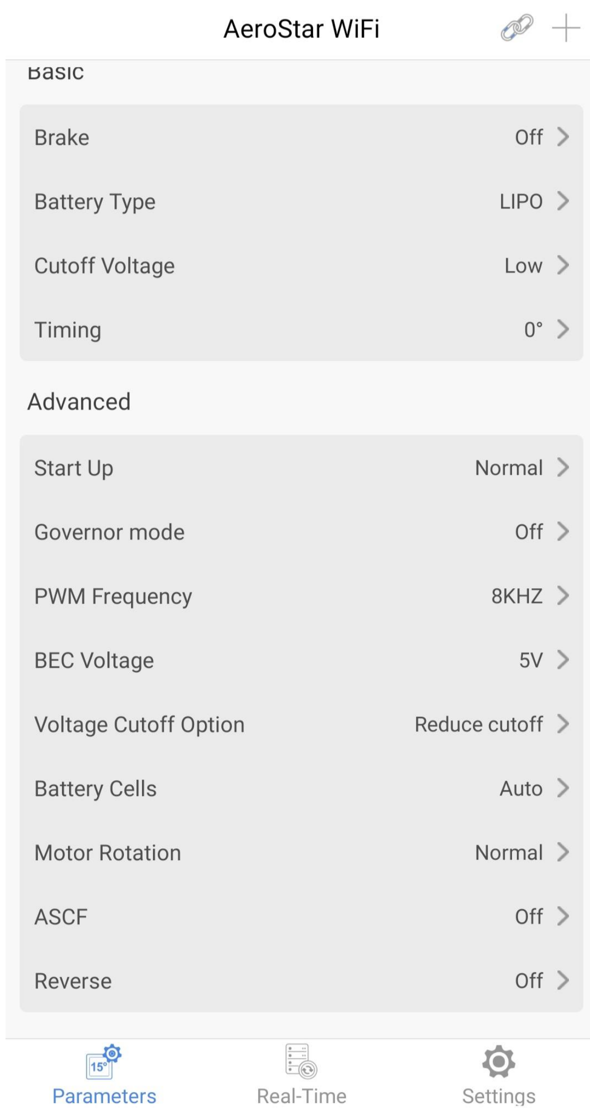

Figure 15: ESC Parameters

Programming the Final Aircraft

In order to fly the final aircraft, the team needed to create custom firmware compatible with QGroundControl. This required them to look into how a custom Pixhawk airframe is programmed. There are two different files that are important: the main airframe setup file and the mixer file. The main file is where the team sets the default airframe parameters values if they are different from the default airframe group values, the mixer files the airframe will use, and the group of the airframe (e.g. Fixed Wing, Quadrotor, Quadrotor VTOL, Tiltrotor VTOL, etc.). Setting the category can be accomplished by setting the "MAV_TYPE"; MAV_TYPE 21 correspons to a Tiltrotor VTOL. The motors will be set to their corresponding outputs in this file.

The mixer files define the behavior of the motors as well as the different actuators on the aircraft. In the figure below a single mixer from a mixer file can be seen.

```
M: 2
O:      10000  10000      0 -10000  10000
S: 0 0  -6000  -6000      0 -10000  10000
S: 0 1   6500   6500      0 -10000  10000
```

Figure 16: Mixer file

The first line indicates that there are 2 controls inputs being mixed. The second line indicates the output scaling. This line can be omitted if the default scaling for the airframe chosen will be used. The last two lines are used for the control input scalars. The first number represents the control group of the input. The second line indicates the exact input from that control group. The numbers after define the range of the PWM signal output.

Since the final aircraft is a Tiltrotor VTOL the team did not need to define a new aircraft group. The team used "VTOL_CONVERGENCE" airframe as a starting point for their program since it was the closest airframe to their custom design. If a mixer is going to be used on the auxiliary board it must be set as a "MIXER_AUX" in the main file and saved as a "file_name.aux.mixer" in the mixer folder. All mixer files and the main file must be declared in the "CMakeLists.txt" file in the respective mixer and main folders. The figures below show some of the code used to define the airframe.

```
param set-default SENS_BOARD_ROT 8

param set-default VT_B_TRANS_DUR  1
param set-default VT_F_TRANS_DUR  1.2
param set-default VT_F_TR_OL_TM   4
param set-default VT_FW_DIFTHR_EN 1
param set-default VT_FW_DIFTHR_SC 0.17
param set-default VT_FW_MOT_OFFID 24
param set-default VT_FW_PERM_STAB 0
param set-default VT_IDLE_PWM_MC 1200
param set-default VT_MOT_ID 1234
param set-default VT_TILT_FW  1
param set-default VT_TILT_MC  0
param set-default VT_TILT_TRANS   0.45
param set-default VT_TRANS_MIN_TM 1.2
param set-default VT_TRANS_P2_DUR 1.3
param set-default VT_ELEV_MC_LOCK 0
param set-default VT_TYPE 1
set MAV_TYPE 21

set MIXER quad_x
set MIXER_AUX vtol_TTAAERD

set PWM_OUT 1234
```

Figure 17: The Main File That Defines Parameters MAV_TYPE and Mixer Files.

```
# Generic quadplane tiltrotor servo mixer

# Tilt mechanism servo mixer
---------------------------
# front left up:2000 down:1000
M: 1
S: 1 4      0 -20000  10000 -10000  10000

# front right up:1000 down:2000
M: 1
S: 1 4      0  20000 -10000 -10000  10000


# Aileron mixer
# --------------------------------
M: 1
S: 1 0  10000  10000      0 -10000  10000

M: 1
S: 1 0  10000  10000      0 -10000  10000


# Elevator mixer
# ------------
M: 1
S: 1 1  -10000  -10000    0 -10000  10000


# Rudder mixer
# ------------
M: 1
S: 1 2  10000  10000      0 -10000  10000
```

Figure 18: Mixer File for Tilt Servos and Flight Controls

```
R: 4x

AUX1 Passthrough
M: 1
S: 3 5  10000  10000      0 -10000  10000

AUX2 Passthrough
M: 1
S: 3 6  10000  10000      0 -10000  10000

Failsafe outputs
The following outputs are set to their disarmed value
during normal operation and to their failsafe falue in case
of flight termination.
Z:
Z:
```

Figure 19: Mixer File for Vertical Flight Mode

```
# [13000, 13999] VTOL
13000_generic_vtol_standard
13001_caipirinha_vtol
13002_firefly6
13003_quad_tailsitter
13004_quad+_tailsitter
13005_vtol_AAERT_quad
13006_vtol_standard_delta
13007_vtol_AAVVT_quad
13008_QuadRanger
13009_vtol_spt_ranger
13010_claire
13012_convergence
13013_deltaquad
13014_vtol_babyshark
13015_convergence_edit
13030_generic_vtol_quad_tiltrotor
13050_generic_vtol_octo
13200_generic_vtol_tailsitter
```

Figure 20: CMakeLists.txt VTOL Section

QGroundControl

QGroundControl is a program used in PixHawk and ArduPilot powered vehicles. It provides full flight control and vehicle setup for many different kinds of airframes. The team chose to use QGroundControl was the ability to design flight routes for our aircraft to autonomously navigate. This removed the human error which caused the majority of crashes the team faced and would allow operators such as search and rescue teams to devote more of their attention to their objective rather than solely on manipulating the aircraft.

Using QGroundControl the team was able to change a wide variety of parameters from the airframe style, to manual controller sensitivity, to the airspeed thresholds for transitions. Using this program, they were able to fine tune the aircraft and improve efficiency, as well as decreasing the chances of failure. They were also able to monitor how much power the aircraft was using, how much voltage the battery had remaining, the airspeed of the aircraft with a pitot tube, ground speed of the aircraft using GPS, and so much more.

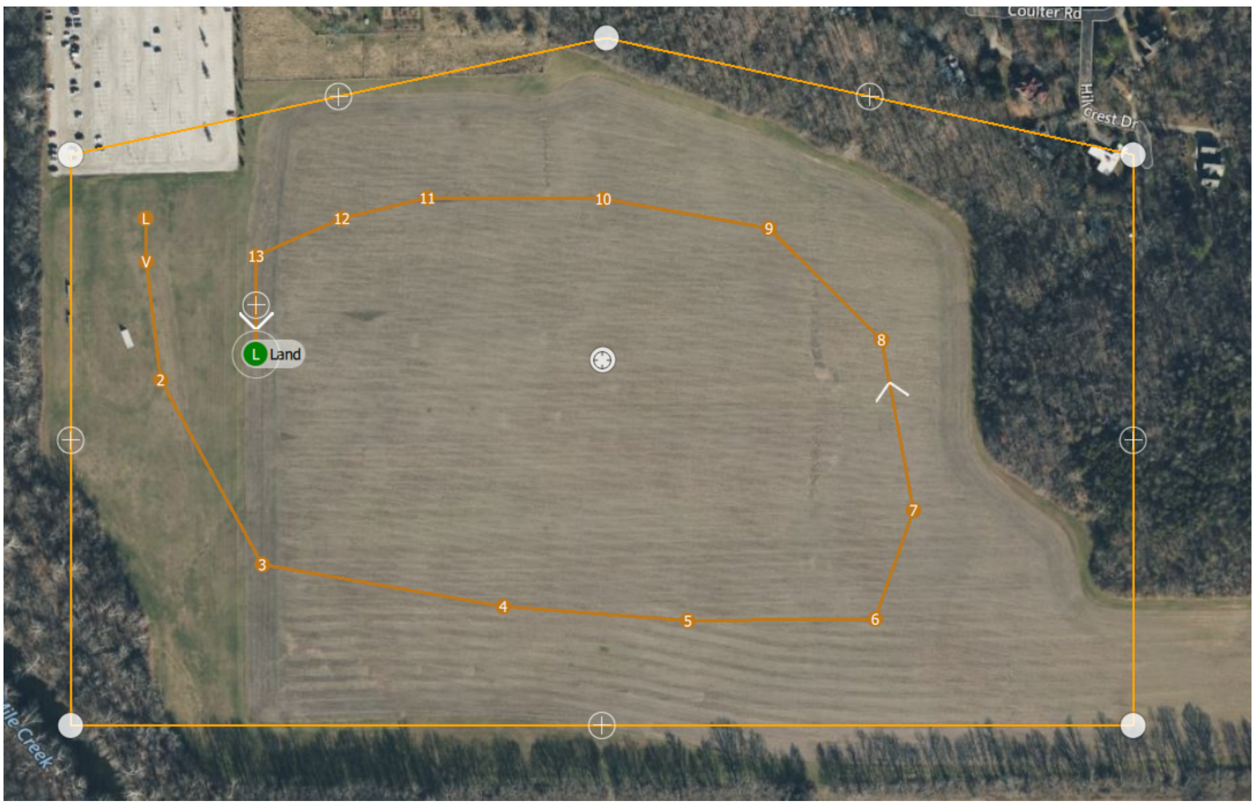

Figure 21: The Route Planning Screen with a set flight route.

The route planning feature gives the operator full control over the flight plan of the aircraft. They can have the aircraft follow a predetermined route or order it to navigate to a GPS fix mid-flight. Figure 21 shows a flight path built in a field. In accordance with this flight plan, the aircraft will take off to a specified height, transition from quadcopter to forward flight, fly the route at a specified speed, and land at the landing point. Another route planning feature is, “Return to Home.” When this feature is activated the aircraft will return to the “Home Location,” defined as the takeoff location unless otherwise specified, then perform the specific action (e.g. Land, Loiter, Wait, Circle, etc.).

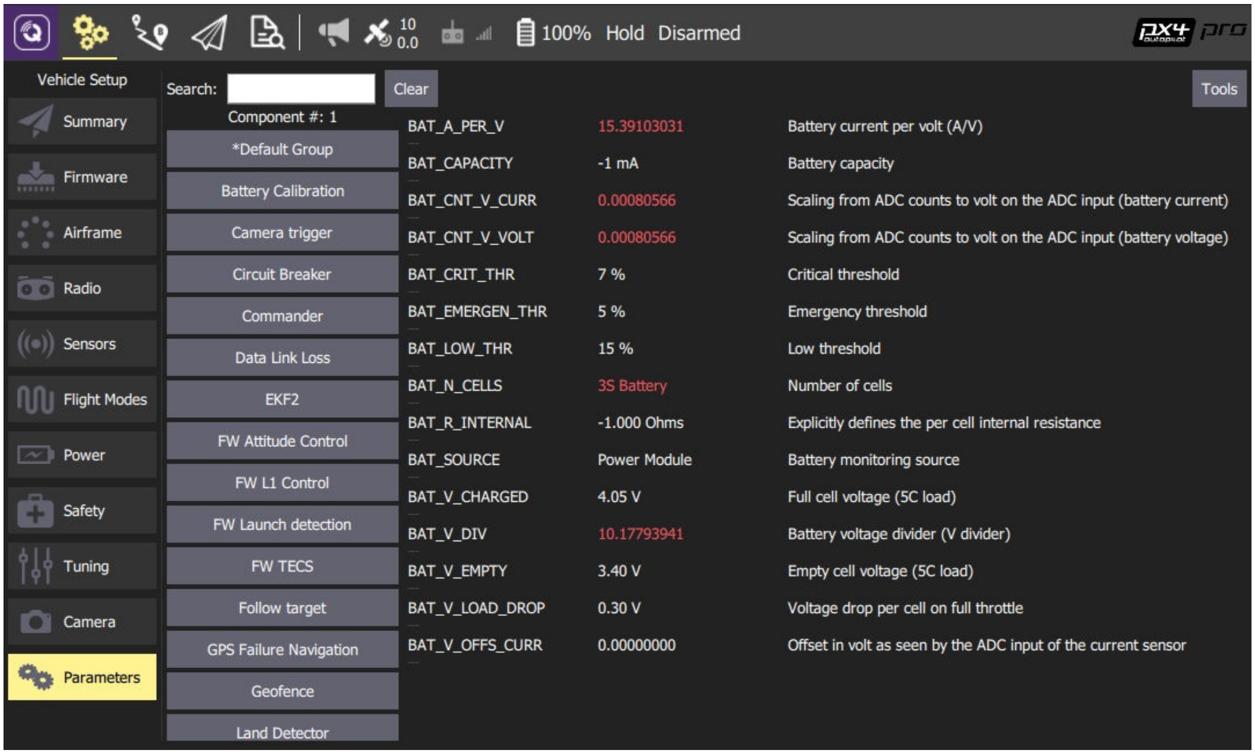

Figure 22: Parameters Window

There were many different parameters the team changed when testing different airframes. For testing the quadcopter, they needed to disable the airspeed sensor, and change the voltage thresholds for accurate battery levels. For the manually flying of the Explorer, they disabled the airspeed sensor and changed the sensitivity of the controls. Sensitivity adjustments can also be made in the Tuning window. For the final aircraft, they needed to invert a few PWM signals, set the minimum transition airspeed and the transition minimum and maximum duration in order to make sure the transition was completed safely, while retaining the ability to automatically abort if the situation became unsafe.

Another section in QGroundControl is the Safety Setup window. This allowed the team to set many different safety behaviors if a situation that could jeopardize the safety of flight occurred. The first setting they adjusted was the "Low Battery Failsafe Trigger" which would force the aircraft to safely return/land when the battery voltage hit a certain threshold. For quadcopter testing, The team set a battery warning at 40% charge, told the aircraft to force a return to land at 20%, and if the battery level ever reached 10% the aircraft would immediately perform an emergency landing. Another important trigger was the "Geofence Failsafe Trigger" which would tell the autopilot what to do if the aircraft left the defined geofence. The team set the aircraft to hold position but options such as Land Immediately, Return to Home, and Warn Operator are also available.

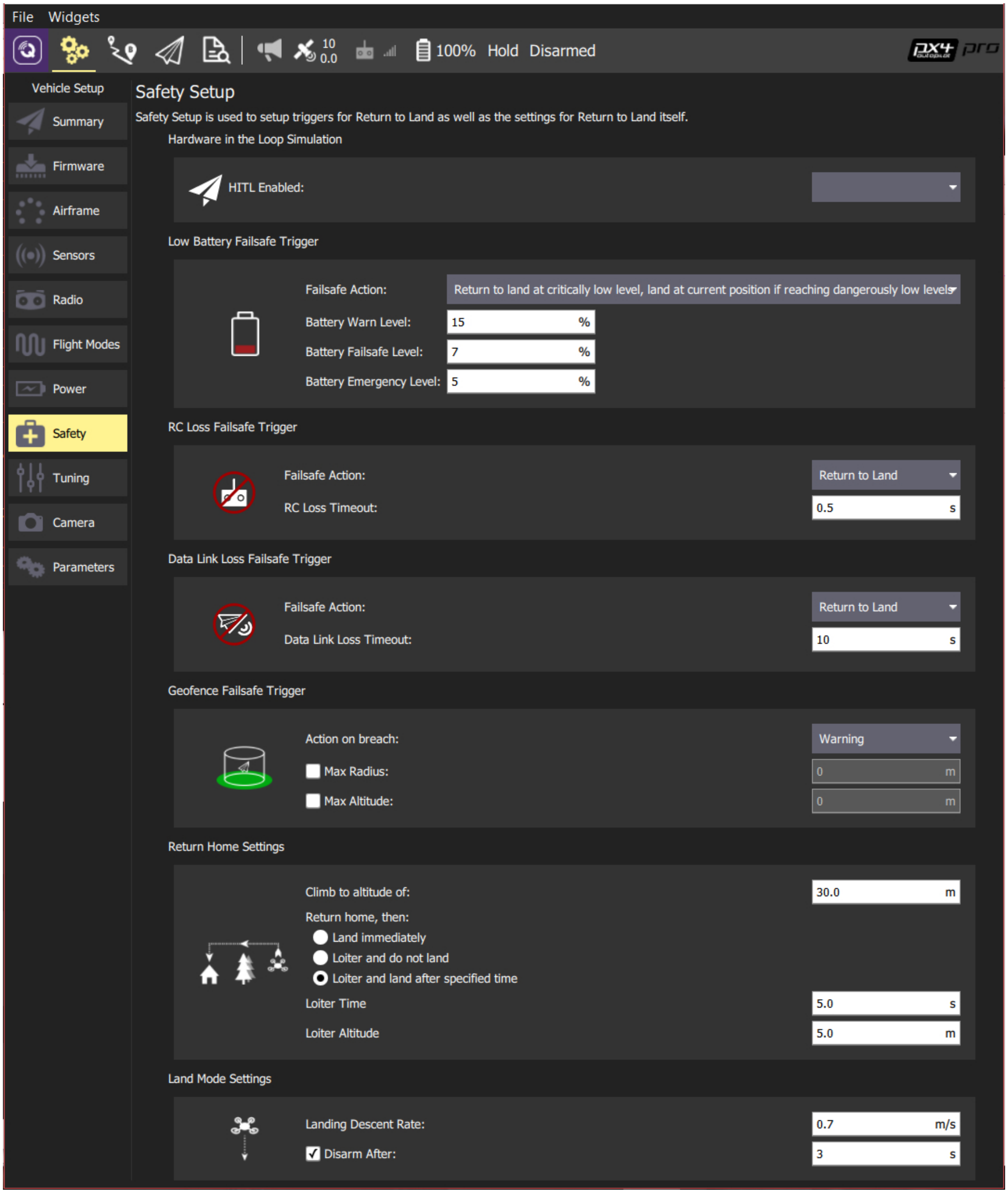

Figure 23: Safety Window

The "Motors" tab allowed the team to test each motor and know if it was functioning correctly. They tested for two main things: the correct motor activates for each channel, and spins in the correct direction. Each airframe has a set direction for each motor, and if the motor is not spinning in the correct direction, then the aircraft will not fly correctly. Usually, if there is a motor spinning in the wrong direction, then two of the motor leads need to be swapped, or PWM signal under the "Parameters" tab can be inverted.

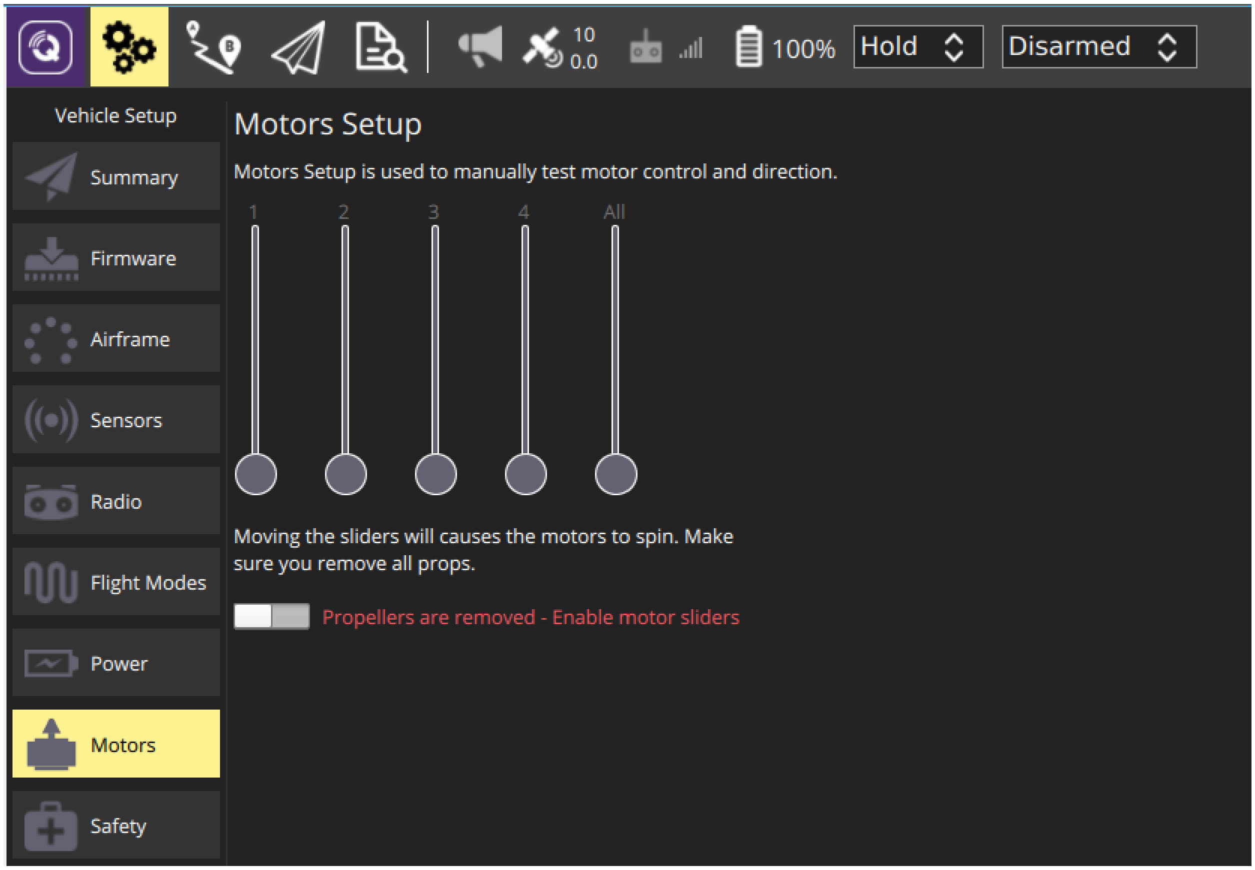

Figure 24: Motor Setup Window

All of the aircraft the team used, except the final aircraft, used a 3 cell, 12.6 V LiPo battery. QGroundControl requires you to define the battery specifications to prevent damage to the battery, even in manual flight mode. The number of cells, the full voltage of a single cell, and the empty voltage of a single cell must be defined. For the battery the team utilized, the full voltage was 4.2 V per cell, and the empty cell voltage was 3.0 V. Then, the team needed to calculate the voltage division ratio in order for QGroundControl to display the correct battery percentage while flying, otherwise it is impossible to know how long the aircraft can safely fly.

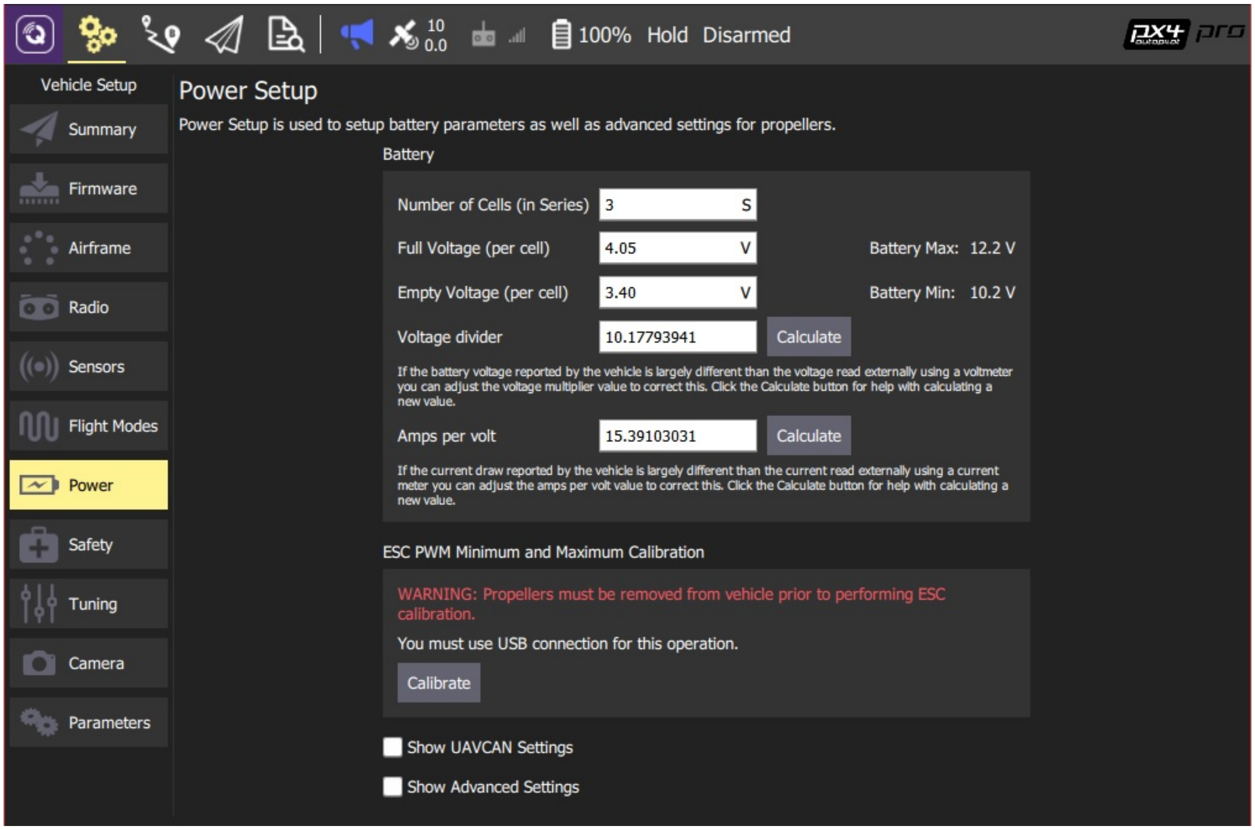

Figure 25: Power Setup Window

For manual flight, the controller must be set up through QGroundControl. The calibration process is straightforward with clear onscreen instructions. As inputs are made, the corresponding channel should move on screen.

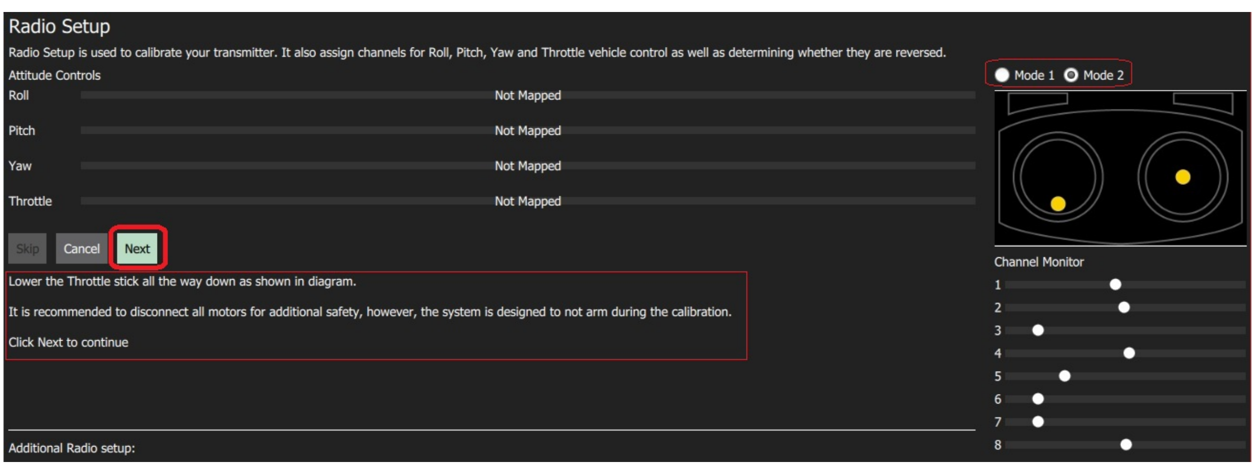

Figure 26: The Radio Setup Window

Solar power

The team began the semester by estimating the number of solar panels necessary to charge the battery in flight. The estimated takeoff power requirements for the aircraft was about 600W, and required cruise power was between 150 to 200 W. The team then began testing the solar panels to determine their characteristics. They were able to determine that maximum output voltage was between 0.5 and 0.6 V for each cell. This is defined as the performance of individual cells given optimum conditions. Jacob Gero assembled a test array of 21 panels to determine the real world performance for multiple cells measured over various resistance values.

Figure 27: Solar Array Voltage Test

From the test, they found that peak output power of our solar panels was about 30 W. Using this data they designed a wing that would be large enough to hold 40 solar panels.

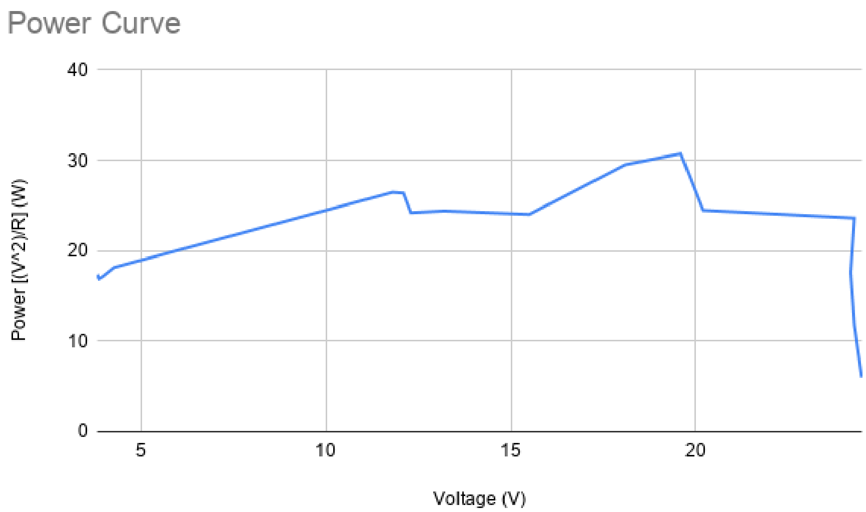

Figure 28: Power Curve

Resistance	Voltage (V)	Power
100	24.5	6.0025
50	24.3	11.8098
33.3	24.2	17.58678679
25	24.3	23.6196
16.67	20.2	24.4775045
12.5	19.6	30.7328
11.1	18.1	29.51441441
10	15.5	24.025
7.14	13.2	24.40336134
6.25	12.3	24.2064
5.55	12.1	26.38018018
5.26	11.8	26.47148289
1	4.26	18.1476
0.91	3.95	17.1456044
0.877	3.85	16.9013683
0.86	3.83	17.05686047
0.847	3.82	17.2283353
0.84	3.8	17.19047619

Figure 29: Power Data

In order to charge the battery the team used the Genasun Solar Charge Controller for lipo batteries. The solar charge controller takes the input from the solar panels and boostes the output voltage to maintain a safe charging current for the battery until it is completely charged.

Figure 300: Genasun Charge Controller Connected to Solar Panels

The team calculated the flight time of the aircraft using battery power alone using this equation: $\frac{Battery\ Capacity\ (Ah) * 0.8}{Current\ Draw\ (A)} * 60\ (\frac{min}{hr})$. The team used a 6000 mAh battery and in cruise the motors were drawing 6 A. Therefore, the team estimated that the plane would be able to fly for 43 minutes using only the battery. During solar panel testing, the team found each panel produced a little over 1.4 W of power. Since some of the solar panels were damaged before the test was performed, the team decided to use a reasonable estimate of output power that would be provided by their final array: 2.5 W per panel. The team then calculated how long the aircraft could fly with the benefit of solar panels. First the team calculated the maximum charge current output by the solar charge controller using this equation: $\frac{Power\ Provided\ by\ Array}{Battery\ Charge\ Voltage}$. Since there are 40 panels producing about 2.5 W, the array will produce 100 W and the maximum voltage of the battery is 24 V. This means that the charge controller can output a current of 4.17 A if it was operating at 100% efficiency. Unfortunately, this cannot happen so the team estimated a system loss of 20% in conjunction with the 95% efficiency of an MPPT charge controller: $4.17\ A\ *\ (1\ -\ 20\%)\ *\ 95\%\ =\ 3.17\ A$. The team then used the flight time equation from

before and substituted the current draw from the motors by the difference between that current draw and the current provided by the change controller:
$\frac{Battery\ Capacity\ (Ah) * 0.8}{[Current\ Draw - Charge\ Controller\ Current]\ (A)} * 60\ (\frac{min}{hr}) = \frac{4.8\ Ah}{6\ A - 3.17\ A} * 60\ \frac{min}{hr} = 102\ min.$Thus, with the benefit of solar power, the aircraft should be able to fly almost 2.5 times longer with a flight time of 1 hr and 42 min.

Video Transmission

Video transmission from an unmanned aircraft allows a ground team to get a bird's eye view of the area below the aircraft as well as provides the pilot another source of information about the attitude of the aircraft. The camera the team is using is the Split Mini 2. It can take a 30 fps video at 1080p and has an option to choose between two different aspect ratios, 4:3 and 16:9. The camera has its own dedicated 3-cell 2200 mAh battery and transmits on a frequency of 5.917 GHz. The team can view the video feed on a portable monitor. The camera was mounted to the nose of the aircraft and angled towards the ground so the team would be able to see the glow stick drop target.

Figure 20: *Split Mini 2*

Future Work

In the future, the team will attach the solar panels to the wings as well as a glow stick dropper to the fuselage. The team will then need to conduct several flight tests including a ground transition test to ensure the aircraft will only complete a transition above its stall speed. Finally, The team will perform a complete contest trial run before the contest in August. If the airframe proves to be a robust platform other applications might be considered for future work such as a flying cell tower, a night vision camera, or other payloads.

Conclusion

This project has helped the team “learn by doing.” There were so many failures over the course of this project that the team was forced to learn new methods of problem solving as well as the tenacity to continue a project despite growing frustrations. After countless nights working in the lab until the early hours of the morning hoping the next fix would be the one gave the team an overwhelming sense of accomplishment once they found something that worked. The team hopes to take the lessons learned in this project: the importance of documentation for future engineers, strong communication skills, strength to overcome obstacles, and the ability to continue working even when the task seems impossible, into their future careers so that they can produce quality products.

Works Cited

"PX4 Autopilot User Guide (Master)." PX4 User Guide, 2020, docs.px4.io/master/en/.

Raymer, Daniel P. Aircraft Design: A Conceptual Approach. 1992.