Glider: *The Monarch*

Eliott Lacomme Ruiz

EE 399 - Final Report

Introduction

The NASA University Student Launch Initiative (NASA USLI) competition spans over nine months, challenging students nationwide to design, build, and test high-powered rockets carrying a payload. In this year's competition, the design team settled on a 39-inch foldable glider with a 49.25-inch wingspan that fits within a 6-inch diameter section of the rocket. According to NASA's regulations, the glider must be deployed below an altitude of 400 feet and land while meeting predefined human-safe acceleration standards.

Over the past eight months, my contributions to our glider, The Monarch, have covered a variety of fields, including the design of the control surfaces, data collection from sensors, code for the microcontroller, design of the retention system, autonomous and manual control, and integration of the electronics.

The design went through multiple iterations, and as a result, the time allotted for the electronics integration and testing was severely reduced, causing major redesigns to occur in order to meet the strict timelines. This document will dive into each of my contributions, how they changed from the original design, and their success in the mission.

Control Surfaces

A rectangular winged glider has an aspect ratio based on the mathematical formula:

$$AS = \frac{s}{c} \quad \textbf{(1)}$$

where *AS* is the aspect ratio, which describes the relationship between the wingspan, *s,* to the chord length, *c*. One main challenge is that typical high-efficiency gliders have an aspect ratio around 20 to 30, while the glider, due to its limited size, could only reach an aspect ratio of 7.822. However, the simulated lift coefficient is approximately 1.1439 with a drag coefficient of 0.223, meaning that the glider could theoretically produce enough lift to stabilize itself but would descend rapidly after ejection.

As a result of these calculations, it became evident the glider would need as much control as possible. Ailerons, elevator, and rudder were chosen to control the glider by using five 9-gram servos. The ailerons use two servos that move in opposite directions to produce roll, the elevators use two servos that move in the same direction to produce pitch, and the rudder uses one servo to produce yaw.

Pitch and roll were considered to be the most important axes of freedom to maximize. Thus, I chose the ailerons to be 38% of the wing's chord length, spanning 36% of the wing's length. The elevators were also maximized to take up the majority of the available surface area, using 91% of the horizontal stabilizer length and 37% of the chord length. Figure 1 and Figure 2 show the final design of both control surfaces. The rudder was also optimized; however, it was later removed due to the insufficient deflection angle after the full rear stabilizer integration. Since the rudder was made before the final iteration of the retention system and sat in between the horizontal stabilizers, as shown in Figure 3, the rudder was only able to move 10° in both directions. Only 1 inch of rudder was exposed to free stream air and was thus deemed to be insignificant and removed.

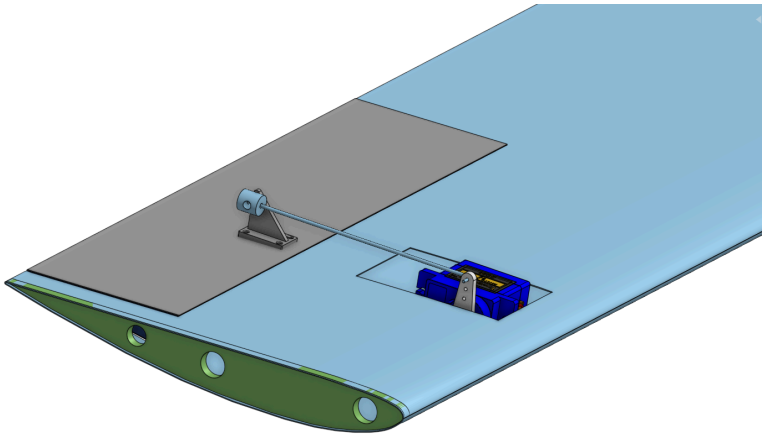

Figure 1: Aileron control surface

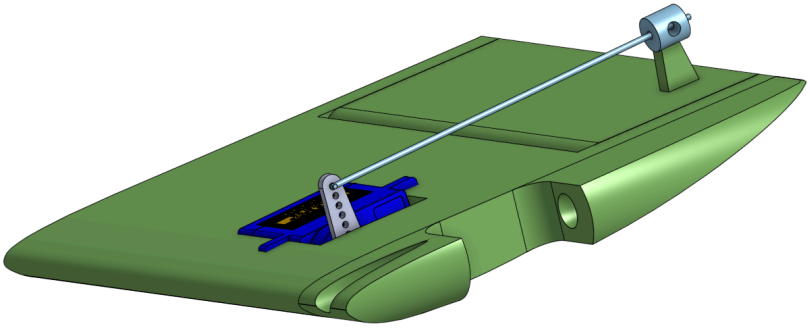

Figure 2: Elevator control surface

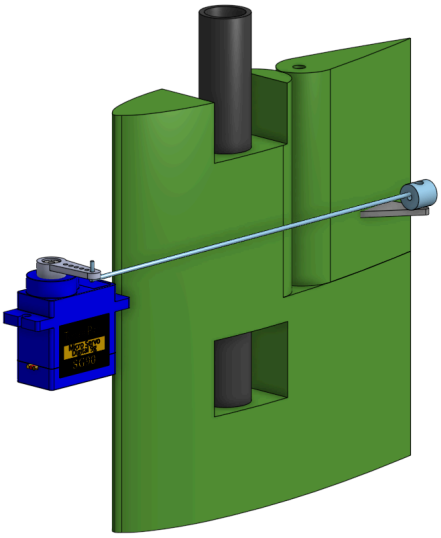

Figure 3: Rudder control surface

Data Collection

A suite of sensors was used to keep track of the glider's position and evaluate mission success. The sensor suite consisted of the following sensors: ADXL375 accelerometer, BMP390 altimeter, BNO055 absolute orientation sensor, and the Ultimate GPS Breakout v3. The GPS was later deemed unnecessary for mission success and was removed from the final schematic.

All data was processed through the Teensy 4.1 microcontroller, chosen for its small size, built-in SD card slot, and compatibility with Arduino, which allowed for simple integration. To ensure smooth integration of all the sensors, I tested each individual sensor and made sure the code snippet could log, read, and interpret the data as necessary. The I2C protocol was used to allow multiple sensors to be connected to the same pins of the Teensy 4.1. All of these tests were documented in a master document on GitHub, with each sensor having comments describing how to wire it, install all the necessary libraries, relevant examples, and post-testing notes.

(GitHub - Sensor Tests)

Retention

Under Federal Aviation Administration (FAA) regulations, no Unmanned Aircraft Systems (UAS) are allowed to be piloted above 400 feet. During the competition, the Range Safety Officer (RSO) would clear the pilot to manually arm the ejection system around 400 feet, but the altimeter must activate the retention servo only at exactly 400 feet.

This poses a challenge since the glider had to be ejected from the launch vehicle after the deployment of the main parachute, meaning the glider needed to withstand a maximum shock force of 532.63 lbf without being prematurely released from the launch vehicle.

My design moved the original retention system from being attached to the launch vehicle, where it would need its own electronics to release the glider, to the rear of the glider, where it would be held in place by a servo. As shown in Figure 4, this was further iterated upon by using a pin-rod system coupled with a 35-kg servo. The new retention system was able to withstand the load of the main parachute deployment, and the 35-kg servo was strong enough to pull out the rod from the pin with no issue. In addition, the new system did not require any additional secondary electronics and could use the data being sent to the Teensy to activate the servo. The retention servo would not arm until it was manually activated by a switch on the RadioLink AT10 and was below the required altitude.

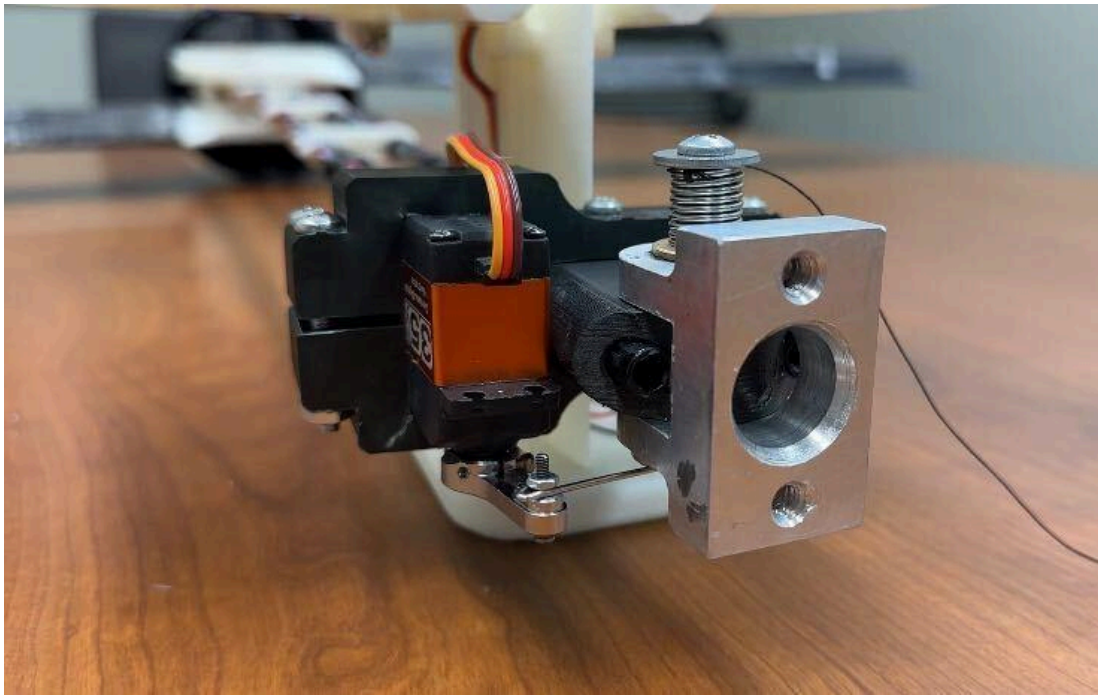

Figure 4:

(GitHub - Ejection Code)

Autonomous and Manual Control

At the beginning of the competition, the original idea was to design our own autonomous code that would rely on the absolute orientation sensor and the relative altitude of the glider. As seen in Figure 5, I designed the orientation code to take in information from the sensors, filter it, and correct the ailerons based on orientation and the elevators based on the glider's current height and pitch. The pitch would be constantly adjusted based on a fixed angle per foot of altitude decrease. This output would be sent to the Teensy, where it would be distributed across the servos.

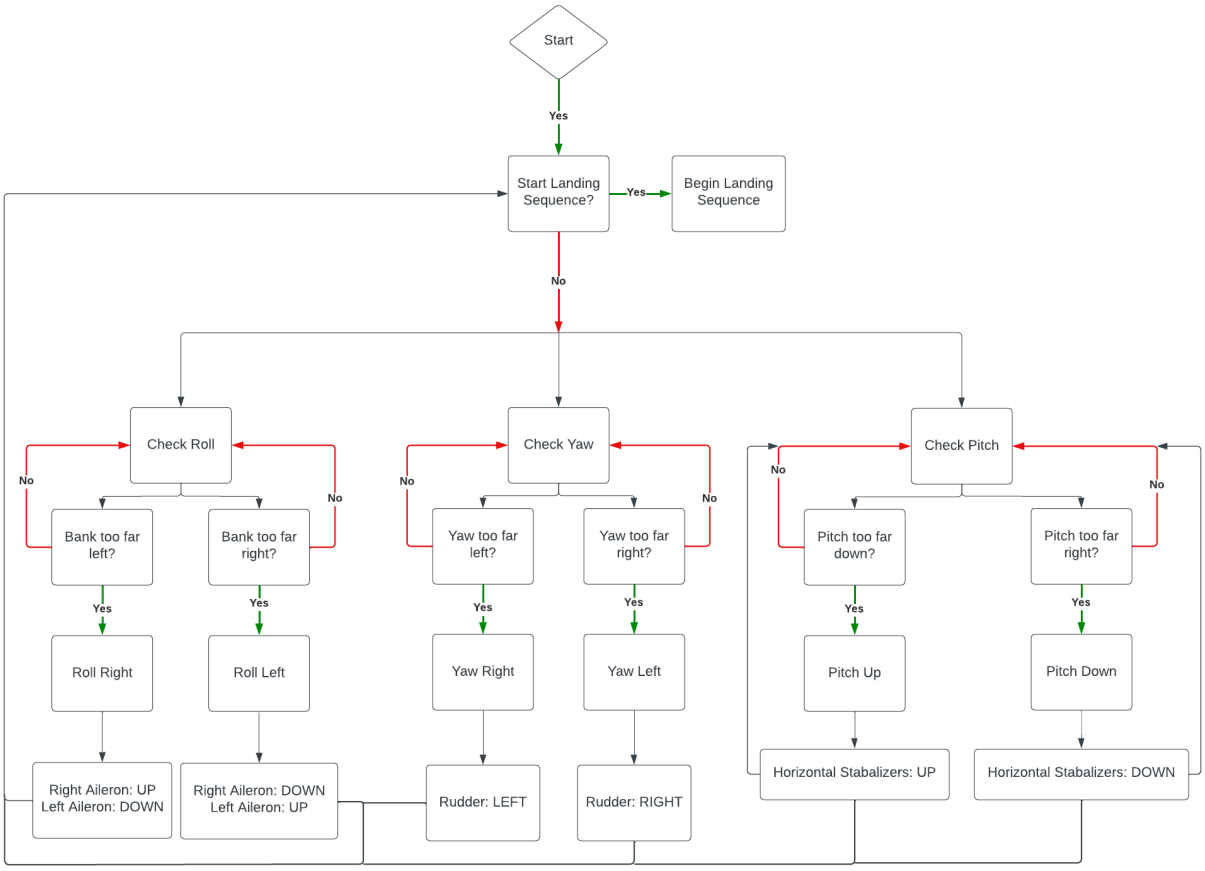

Figure 5:

However, it quickly became apparent that with minimal time for testing with the actual glider, creating our own autonomous code from scratch without a method of tuning would mitigate the benefits of using custom code. To solve this problem, we decided to switch to the Mini Pix, a prebuilt autopilot circuit that uses a software called ArduPilot to train and program the Mini Pix for optimal autonomous flight. The Mini Pix came with pre-trained fixed-wing models for faster tuning. I redesigned the architecture for the electronics, as seen in Figure 6, to work with the Mini Pix. Unfortunately, significant challenges arose with the Mini Pix. Due to a lack of research, the Mini Pix was bought without realizing it was no longer supported by ArduPilot. As a result, critical features, such as GPS mission planning and proper in-flight

autonomous control, were not fully accessible. Additionally, the GPS module compatible with the Mini Pix was discontinued and could not be found online for a reasonable price. These issues meant that the Mini Pix could no longer be used for its prebuilt autonomous control, only for manual control.

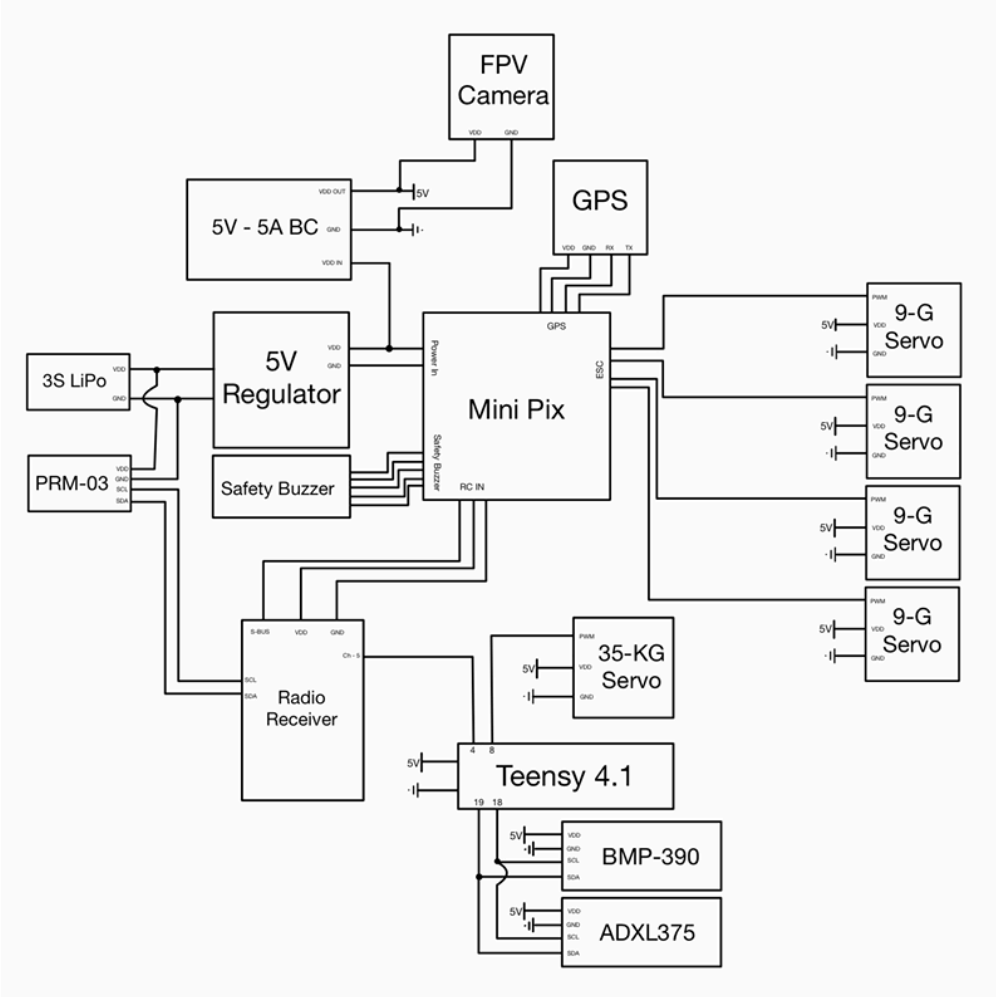

Figure 6:

Since the main functionality of the Mini Pix was gone, I decided to remove it. This reduced the electronic size and complexity by half. Manual control could be achieved through the RadioLink R12DS radio receiver and trimmed through the RadioLink AT10 radio transmitter. To allow the receiver to output the same input to two servos, two pairs of channels were configured as master-slave pairs, so the slave would mirror or invert the input of the master for simultaneous control. The electronics were rewired in the configuration shown in Figure 7. The switch to this final configuration was successful and functioned as intended in all tests. It was confirmed during flight testing that the new configuration successfully deployed the glider and retained it after main parachute deployment.

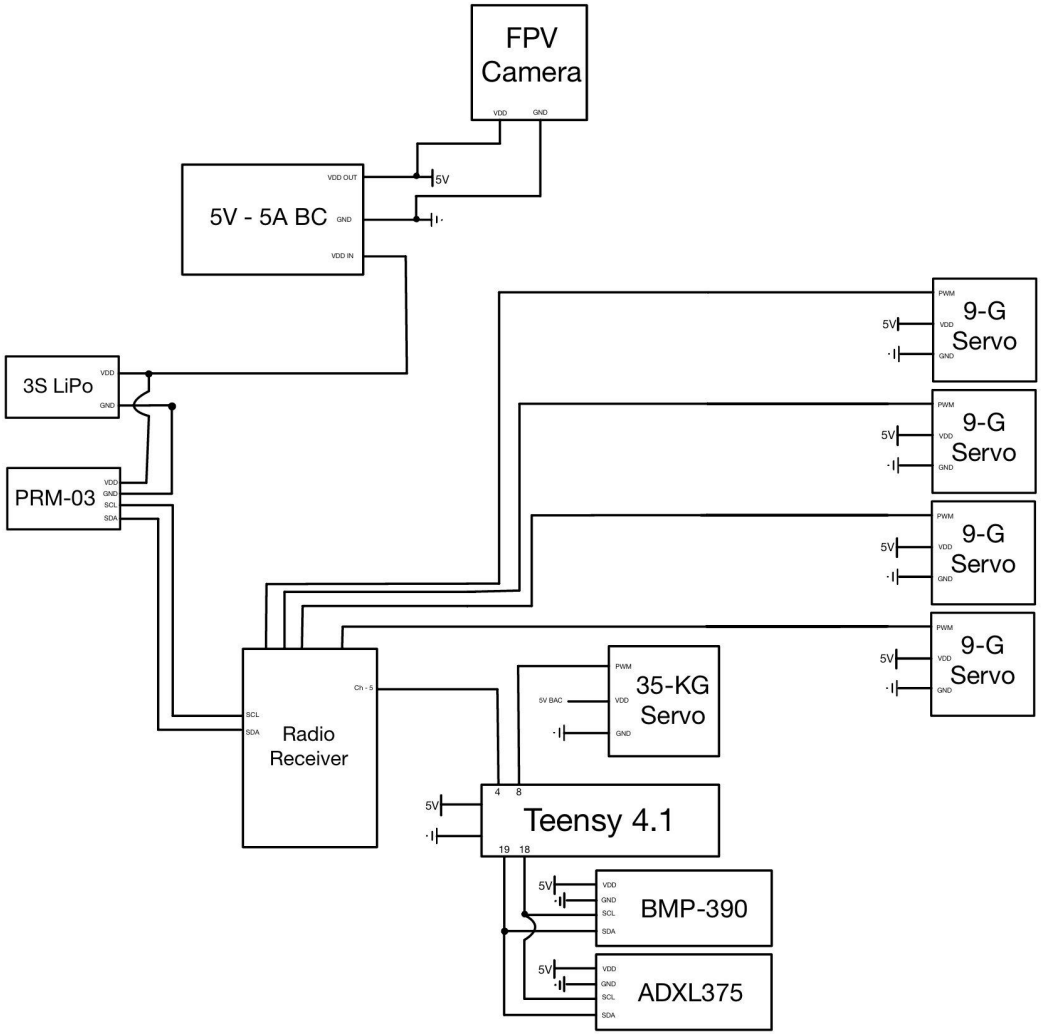
FPV Camera
VDD
GND
5V - 5A BC
VDD OUT
GND
VDD IN
5V
3S LiPo
VDD
GND
PRM-03
VDD
GND
SCL
SDA
9-G Servo
PWM
VDD
GND
5V
9-G Servo
PWM
VDD
GND
5V
9-G Servo
PWM
VDD
GND
5V
9-G Servo
PWM
VDD
GND
5V
Radio Receiver
Ch - 5
SCL
SDA
35-KG Servo
PWM
5V BAC
VDD
GND
Teensy 4.1
4
8
19
18
5V
BMP-390
VDD
GND
SCL
SDA
5V
ADXL375
VDD
GND
SCL
SDA
5V

Figure 7:

Electronics

As a result of the changes in the autonomous and manual code, the electronics went through various iterations. Figures 6, 8, and 12 show the changes in the schematics.

The first version of the electronics, Figure 8, did not contain the Mini Pix, but had the rudimentary electronics needed to control the glider and receive data from altitude, orientation, and voltage telemetry. It used a voltage divider to output the proper voltage to each of the servos, however, it was quickly realized this would be an ineffective constant voltage source due to the load of the servos.

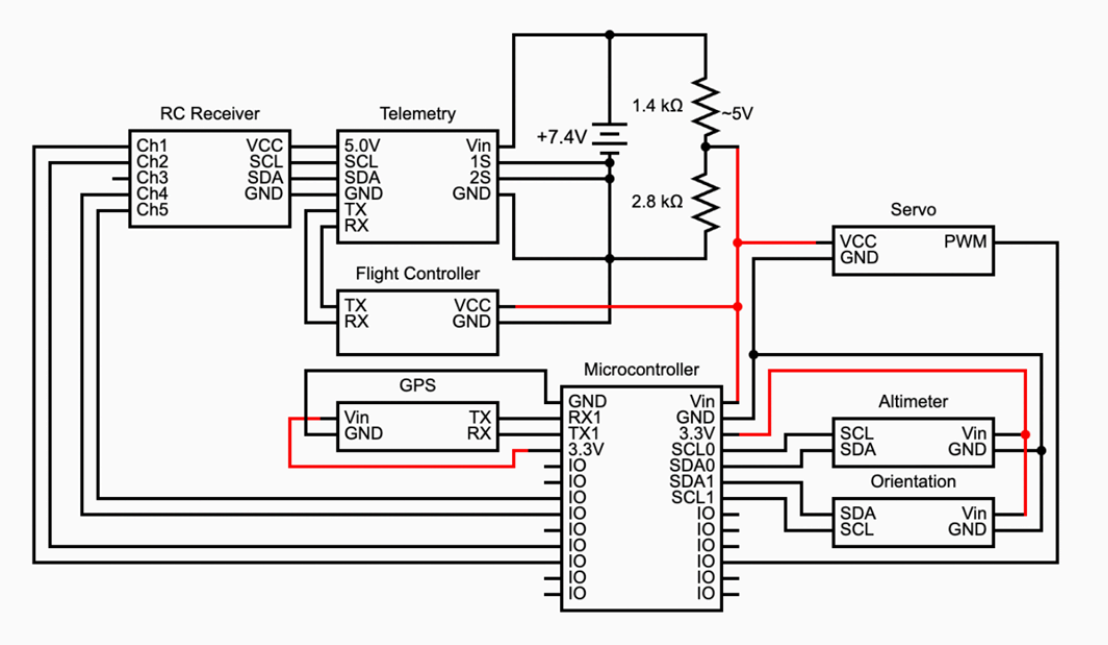

Figure 8: First iteration of the electronics

The second iteration of the electronics, Figure 6, came as a result of the switch to the Mini Pix. This circuit integrated the Mini Pix and was successfully able to move the control surfaces autonomously. In addition, the Teensy 4.1 now controlled the retention servo and an extra IO pin was used to receive the arming signal to activate the servo from the R12DS receiver. To make flying the glider easier, I added a First Person View (FPV) camera to the circuit, so the pilot could more easily control the glider by receiving a wireless video signal from the camera that would be displayed on a pair of wearable goggles. The FPV goggles can be seen in Figure 9 and the camera in Figure 10. The output from the goggles is displayed in Figure 11. The camera proved help, in theory, but its small antenna caused frequent video dropouts when testing on the field. In addition, one of the limitations of this circuit was that the 5V regulator that came with the Mini Pix was not able to supply enough current to the FPV camera, Teensy, and retention servo. Thus, a 5V-5A step down Buck Converter was used with the 3S LiPo battery to provide enough power to all of the components.

Figure 9: FPV Goggles

Figure 10: Micro FPV camera

The final design of the electronics, Figure 7, came after the switch from autonomous to full manual control. The complexity of the circuit was greatly reduced and was able to fit within a smaller space. One of the main differences between the final and second version is the servos

are now controlled by the R12DS receiver, insead of the Mini Pix. The circuit was successfully able to control the retention servo and all control surfaces, while displaying live video to the pilot. The newest circuit integration is shown in Figure 12.

Figure 11: FPV Goggle output

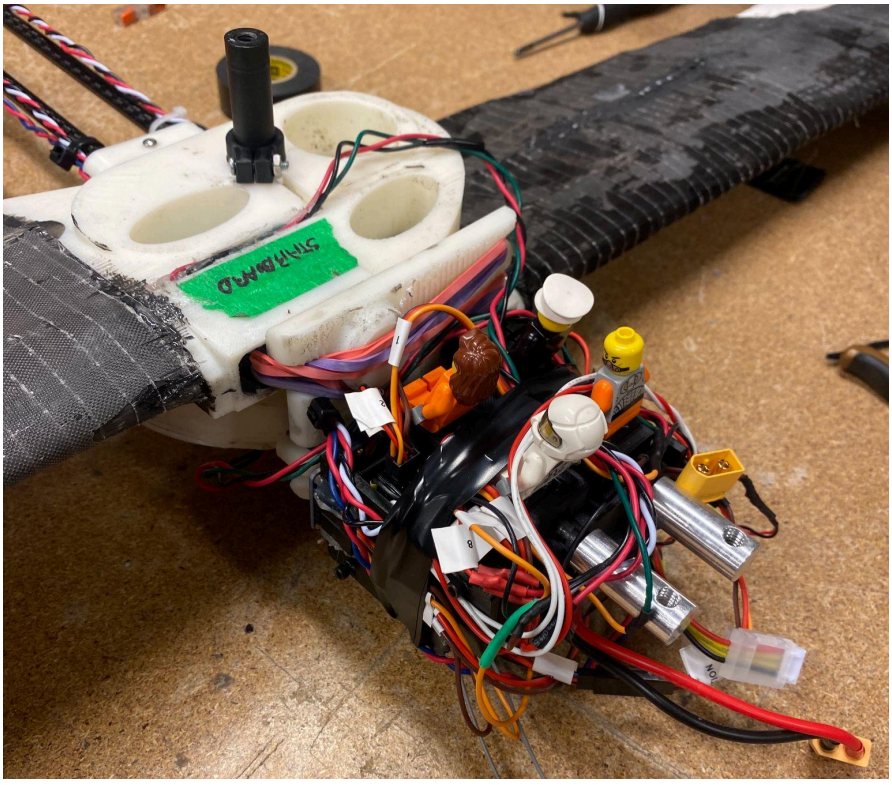

Figure 12: Final electronics integration on The Monarch

Future Development

For the future, I would like to switch to the PixHawk, an updated version of the Mini Pix that is more powerful, well-documented, and still supported by ArduPilot. Given enough time, the PixHawk could execute a plan based on GPS coordinates and train the glider mid-flight.

One main issue with the design was trying to constrain the electronics to the size of the nose cone. Due to my lack of experience in making printed circuit boards (PCB), I was unable to create an efficient design and, as a result, had to solder the components directly to a protoboard, making them irremovable for any future testing. A custom PCB would significantly reduce the number of disconnected wires and simplify integration.

In addition, a better camera setup would increase the pilot's ability to control the glider. Our camera struggled to transmit signals due to the monopole antenna. A better alternative would be using a cloverleaf antenna or a longer antenna. Also, the current camera uses analog video transmission, while a digital camera could transmit longer distances and output higher resolution video. However, it would require specialized FPV goggles that can receive the digital signal.

Bill of Materials

The Bill of Materials can be accessed here: Bill of Materials