StratoCats

Wisconsin Space Grant Consortium Regional Rocket Competition

Flight Readiness Report

Jeffrey Allen
Kevin Eliason
Jacob Chesley
Nick Fagan
Brandon Horne

University of Cincinnati, Cincinnati, Ohio

Faculty Advisor – Grant Schaffner, Ph.D., University of Cincinnati

Special thanks to:
Curt Fox, Senior Research Associate, University of Cincinnati
Professor Kelly Cohen, University of Cincinnati
Professor Elad Kivelevitch, University of Cincinnati
Wei Wei, Graduate Student, University of Cincinnati
Team Ohio Rocketry Club

`

Table of Contents

`

1.0 Executive Summary

The goal of this competition is to design, construct, and fly a one-stage high-powered rocket that will accurately achieve an apogee of 3000 feet and be recovered safely and in flyable condition within the competition parameters. The competition parameters are as follows: The rocket motor to be used is the Cesaroni I540-16, and the dimensions of the rocket cannot exceed 4" in diameter, 72" in height, and must weigh no more than 7.5 lbs without the motor. The rocket must reach an apogee as close to 3000 ft as possible, but not going below 2500 ft, or exceeding 3500 ft.

The team from the University of Cincinnati designed and constructed a rocket (*Icarus*) to meet these parameters. The rockets dimensions are 69" in height, 4" in diameter, and weighs approximately 5.8 lbs (7.2 lbs with motor). The rocket was based off of a kit from LOC Precision, the Fantom 438 EXL, with most parts coming from the provided material in the kit. Fasteners and other hardware were purchased from McMaster Carr, and custom parts were 3D printed with a Solidoodle 3D printer. Electronics and circuitry items were purchased from Sparkfun.com and Radio Shack.

2.0 Introduction

This competition is an initiative of the Wisconsin Space Grant Consortium (WSGC) in collaboration with the University of Wisconsin. The competition is open to groups of four to five students from universities in the Midwest Region. There are 15 teams from the region will be competing in the competition.

3.0 Design

3.1 Design Approach

The goal of the competition is to launch a rocket as accurately as possible to an apogee of 3000 ft. There are many ways to design a rocket to achieve this goal. A rocket can be designed so that it is heavy enough to reduce the apogee to 3000 feet, but as the accuracy of the launch can be heavily influenced by weather and other launch conditions the method is not very accurate. A more precise way to reach 3000 feet consistently is with an active control system guiding the rocket to the correct altitude, be it with air brakes, retro-rockets, or other ways of inducing drag or downward force to slow the rocket down. The active control system will need to be able to calculate what the expected apogee is and control the method of slowing the rocket down enough to reach the goal apogee.

In order to ensure the rocket reaches an accurate apogee of 3000 feet, a mixture of both methods listed above was employed. First, the rocket was designed to be heavy enough to not exceed the ceiling apogee limit of 3500 ft. This ensures the rocket will always be within the competition bounds, as well as reducing the power necessary to accurately reach the goal apogee of 3000 ft consistently with the active control system. The control system designed was a geared air brake system that rotates out perpendicular to the flow across the body during flight. This increases the cross sectional area of the rocket exposed to the flow increasing the drag and slowing the rocket down with variable cross-sectional profiles.

As the goal apogee is high enough to cause a large amount of drift when a single parachute deployment is employed, the rocket was designed for dual deployment. The dual deployment will be controlled by an electronic match (e-match) from the flight computer/altimeter, with the motor ejection used as a back-up to eject the drogue chute of the black powder charge fails to ignite.

In addition, other design concepts have been considered. Many high performance rockets are big, ungainly and hard to transport and store. In order to mitigate these issues, the rocket was designed to be modular. Each module has its own purpose, even if it's just an empty body tube used to lengthen the rocket between other sections. See section *3.3 Layout* for a description of each module. In addition to modules, a removable fin design was considered for easy storage as well as for variance in aerodynamics.

3.2 Materials

The majority of the rocket was built from a couple Loc Precision Fantom 438 EXL rocket kits. The kits include body tubes, couplers, bulkheads, centering rings, a motor mount tube, an electronics bay, a nose cone, two shock chords, a drogue parachute and a main parachute. The kit also came with wooden fins, but the team decided to make custom fins from Divinycell Foam and Carbon Fiber. The rocket was glued together with Loctite Quickset epoxy (resin and hardener) as well as Loctite Heavy Duty epoxy. For the fins, the carbon fiber was applied to the Foam and cured with Fibre Glast 2000 epoxy resin and 2060 60 minute epoxy cure. The airbrake components were printed using ABS High Impact plastic.

3.3 Layout

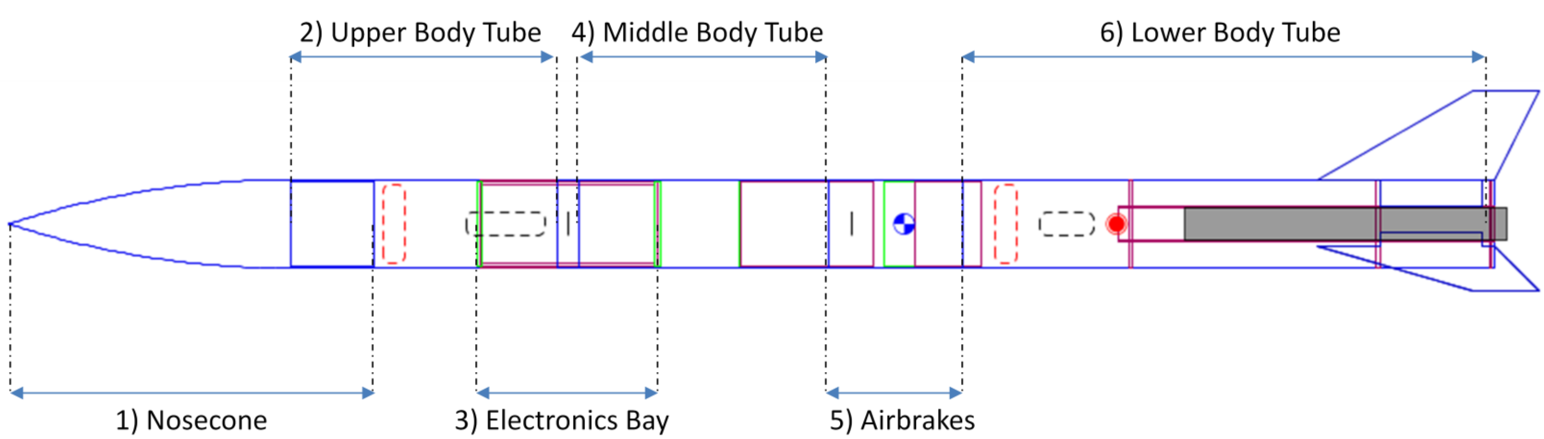


Figure 1: ***Icarus*** **Rocket Layout**

`

The Rocket "Icarus" has the following overall dimensions:

I. Length: 69" from tip of nosecone to end of fins (under the competition maximum 72")
II. Mass: 5.82 lbs less motor (under the competition maximum of 7.5 lbs)
III. Static Margin: 1.595
    a. CG: 40.38" from tip of nosecone
    b. CP: 49.953" from tip of nosecone

The rocket was designed to be modular and includes the 6 modules in figure 1. A thorough description of each module is as follows:

1) Ogive Nosecone
   The Ogive Nosecone is from the Loc kit and has an outer diameter (OD) of 4" and a transition length of 12.75" with a wall thickness of 0.125". The shoulder of the Nosecone that inserts into the Upper Body Tube has an OD of 3.875", a length of 3.75" and a wall thickness of 0.125". The Nosecone is made from Polystyrene PS. Two 0.5" diameter holes were drilled into the bottom surface of the Nosecone to thread and secure the Main parachute shock chord through it.

2) Upper Body Tube
   The Upper Body Tube has an OD of 4", a length of 12", and a wall thickness of 0.05". It is made of Cardboard/Paper. The Upper Body Tube stores the Main parachute and shock chord. For the Main event, the Nosecone will be ejected off of the Upper Body Tube.

3) Electronics Bay
   The Electronics Bay is a thick coupler (3.83" OD, 7.75" long, 0.12" wall thickness made from Cardboard/Paper) that sits between the Upper Body Tube and the Middle Body Tube. The Electronics Bay houses a wooden electronics sled that the Competition Flight Recorder, a backup altimeter (Perfect Flight Stratologger SL100) and power source is mounted to. The Bay is be contained by two wooden bulkheads, held together by two 8.5" threaded steel rods and wing nuts. The bulkhead facing the nosecone has a steel eyebolt epoxied through the center for the main shock chord to attach to. The lower bulkhead is epoxied to the coupler. Four holes of at least 0.09" diameter are drilled through the bay per the manual of the SL100 altimeter to take accurate pressure readings. The Upper Body Tube and Middle Body Tube are attached to the Electronics Bay with a set of metal screws to keep them from separating during flight.

4) Middle Body Tube
   The Middle Body Tube has an OD of 4", a length of 11.25" and a wall thickness of 0.05". It is also made of Cardboard/Paper. This module acts as a spacer between the Electronics Bay and the Air Brakes. It houses a bulkhead epoxied to the Tube about 4" from the bottom to create a smaller enclosed surface for the Air Brake altimeter to take more accurate pressure readings. It also houses a Push Button switch from Radio Shack, above the bulkhead

mounted through the wall of the body tube, which arms the rocket when pressed on the launch pad. The wires are routed through a center hole in the bulkheads at both ends of the Middle Body Tube (to the Electronics Bay and Air Brakes Modules), and sealed off with electrical tape to keep the bays enclosed structures for pressure readings..

5) Air Brakes
The Air Brakes module has an OD of 4” and a length of 7”. The module contains the assembly of the Air Brakes with the lower bulkhead, shown in detail in section 4.1 below. The air brakes consist of two main sections: the mechanical and electrical. The electrical section is a 3D printed bracket that all of the electronics are mounted to. The bracket is then mounted to the top bulkhead of the mechanical part. The electronics used to control the airbrakes are: an Arduino Mega 2560 R3, an SL100 Stratologger Altimeter, an ADXL345 Triple Axis Accelerometer, and a ProTek 100T servomotor. The mechanical portion of the air brakes contains the bulkheads, central gear, and air brake flaps that will extend out of the body tube to provide the braking power. When extended the air brakes have 0.475 in^2 of surface area each (total of 1.425 in^2 exposed surface area) perpendicular to the flow.

6) Lower Body Tube
The Lower Body Tube has an OD of 4” and a length of 24” (excluding fins). The 38mm Motor Mount Tube is mounted to the body tube by the centering rings. The Centering Rings were epoxied to the Motor Mount Tube, inserted into the Body Tube, and then epoxied into place. The Lower Body Tube assembly contains the drogue parachute and shock chord, motor, and fins. The shock chord is attached to an eyebolt glued to the upper centering ring. The fins are through-the-wall, and are held to the Motor Mount Tube by a row of 1/8”diameter neodymium magnets. The motor is inserted into the Motor Mount Tube and secured by a Slimline Motor Retainer, epoxied to the bottom of the Motor Mount Tube.

7) Recovery
The rocket is designed for dual deployment, with the Main parachute ejecting at 500 ft. The drogue parachute will be stored between the Lower Body Tube and the Air Brake modules, with the shock chord connecting them. The main parachute will be stored between the Upper Body Tube and the Nosecone, with the shock chord connecting them. Both areas of separation will be held together with 1-2 shear pins on the launch pad to keep the rocket from separating too early.

The drogue parachute is 18” in diameter, with the shock chord about 4’ long. The main parachute is 36” in diameter, with the shock chord about 3’ long. The shock chords are long enough so when the parachutes are deployed, the modules won’t impact one another as they are suspended from the parachutes. All shock chords are connected to bulkheads (or the Nosecone) with steel eyebolts. The coupler that is exposed to the shock chord from the Air Brakes module is coated with a layer of epoxy on the inner diameter to help it resist zippering. The parachutes will be protected with “dog barf” where the ejection charges are (both sides for the drogue, the bottom side for the main).

3.4 Motor Used

The motor that we are using for the competition is a reloadable I540-16 made by the manufacturer Cesaroni Technology Inc. The average thrust is 537.77 Newtons with a peak thrust of 626.00 Newtons, and has a delay of 16 seconds before the ejection charge ignites. With our rocket weighing in at 7.14 lbs this gives us a thrust to weight ratio of 19.71. The motor fits into a Pro38-5G motor case and is ignited using an e-match. The motor without the casing weighs 598.2 grams, 328.8 of which is the propellant. The length of the motor is 367mm with a diameter of 38mm. It has a total impulse of 634.58 Ns with a thrust duration of 1.18 seconds. We will be using a Slimline motor retainer attached to the bottom to contain the motor. The thrust curve for this motor is shown in Figure 2.

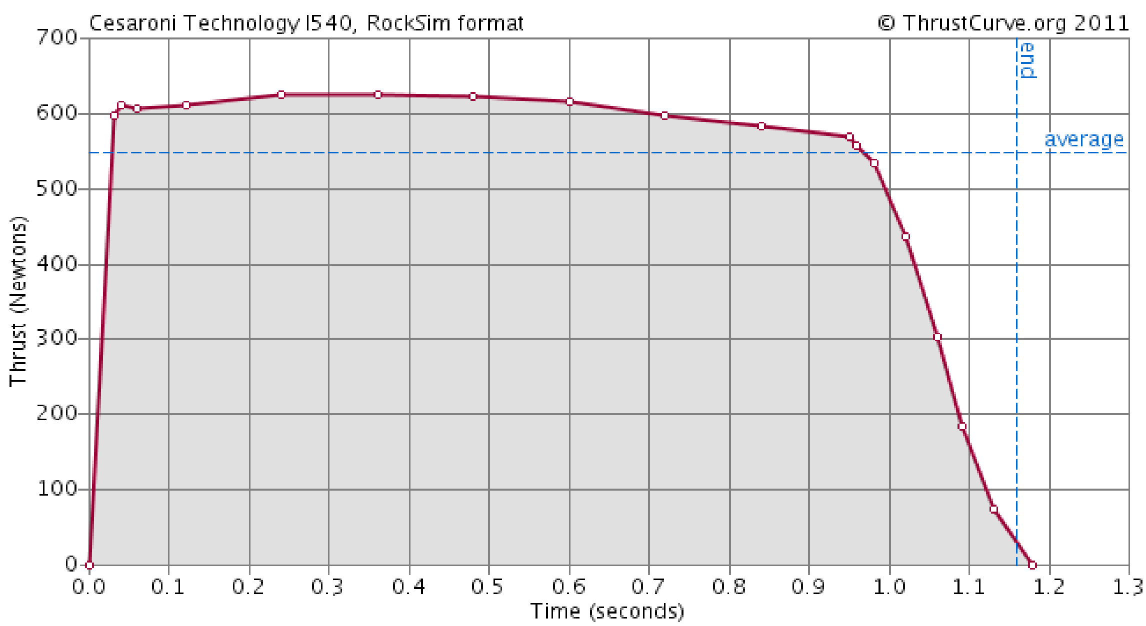


Figure 2: Cesaroni I1540-16 Thrust Curve

3.5 Rocket Finish

The rocket will have a finish on the body of a vinyl wrap. This will reduce some drag, and make the rocket look aesthetically pleasing. The Fins will have a smooth resin/carbon fiber finish. The Nosecone will have the stock plastic finish, but shaved with a razor blade to remove excess material from the mold.

4.0 Construction

4.1 Air Brakes

4.1.1 Construction

The airbrakes were made out of wooden bulkheads, custom 3D printed parts, and fasteners from McMaster Carr. Three wooden bulkheads were purchased from Loc Precision to ensure the dimensions were accurate to fit inside the rocket. Two of the bulkheads served as the lower support bulkhead assembly which was epoxied to the inner diameter of the body tube. A truss head screw was then threaded through the middle of the bulkheads and a steel eye nut screwed onto the threads to attach the drogue parachute shock chord to. A custom 3D printed plastic bulkhead was then glued on top of the wooden bulkheads to serve as a form-fitting mounting area for the 3 rotating flap assemblies. Each flap rotated about a set of 2 ball bearings with a machine screw holding the assembly together. The 3d printed central gear and flaps are mounted on top of the plastic bulkhead, with the machine screw for the ball bearing assembly protruding out the bottom of the lower bulkhead assembly. This allows the flap assemblies to be fastened to the lower bulkhead assembly via aluminum wing nuts. Then, the upper bulkhead assembly was fashioned by another plastic bulkhead epoxied to the bottom of one wooden bulkhead. This upper bulkhead assembly was placed on top of the flaps and gear and fastened to the lower bulkhead assembly by screwing machine screws through the top to weld nuts epoxied to the bottom bulkhead assembly. A stencil was 3d printed for the body tube of the air brakes assembly to cut the correct hole size for the air brake flaps to extend out of. Once the air brake module was constructed, a coupler was glued to the body tube and bottom of the lower bulkhead so it could fit on top of the Lower Body Tube. The inside of the coupler was coated with a layer of epoxy to help resist zippering of the thin material during recovery.

`

4.1.2 CAD model

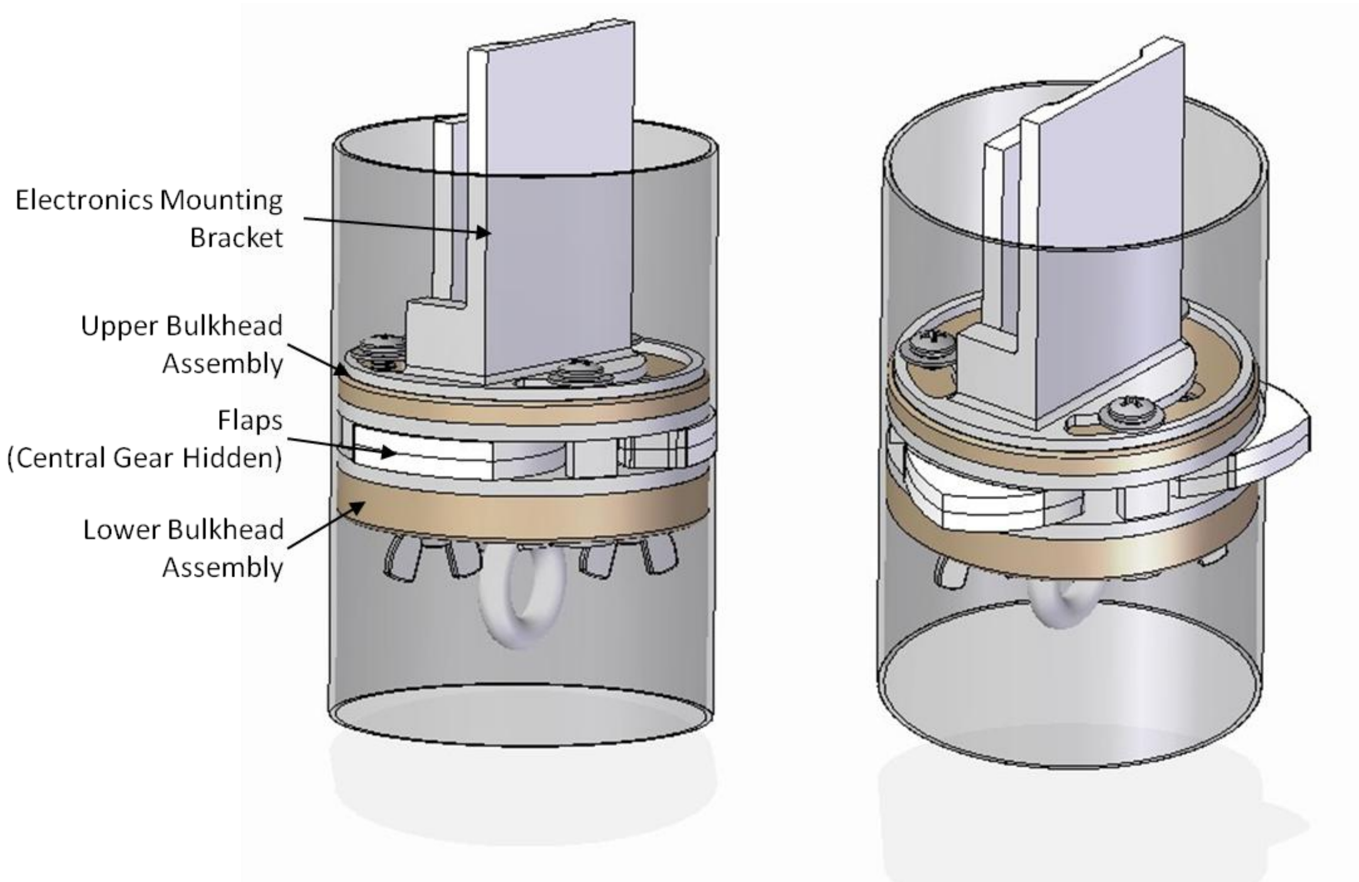


Figure 3: Overall Air Brakes Module, Closed (L) and Open (R)

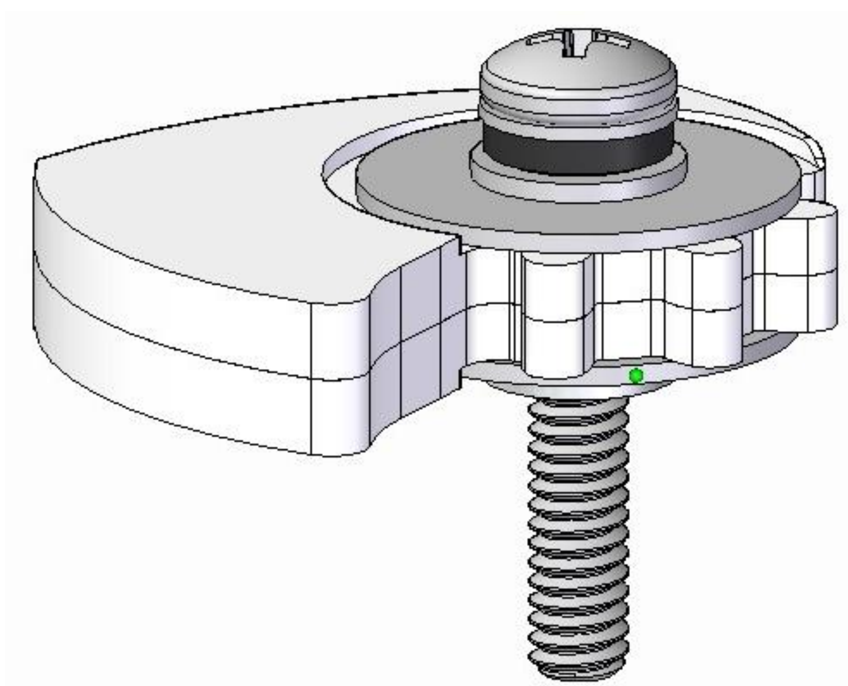


Figure 4: Air Brakes Flap Assembly

`

4.1.3 Control System

The air brakes are controlled by an active control system. This was accomplished by writing an Arduino code reading in real time data to calculate the expected apogee from coasting distance and adjust the air brakes accordingly. It's a simple system that will calculate the expected apogee based on velocity and altitude at the data point, and if the expected apogee is greater than 3000 ft, the air brakes will deploy, slowing the rocket down. If the expected apogee is at or below 3000 ft, the air brakes will close. This process will run in a continuous loop, but data will be taken every 100 milliseconds to help reduce error from noisy readings.

The instantaneous velocity is derived by taking the integral of the acceleration curve over the time the control system is running (starting at acceleration of 0 ft/s^s and starting time of 0 s). The integral is numerically calculated by taking the area under the acceleration curve, calculated and constantly updating in the control system loop, so the instantaneous velocity can always be calculated throughout the flight. The velocity is read from the telemetry data output from the SL100 altimeter. The time is calculated through an internal function in the Arduino as soon as it's turned on, and a time delta is calculated each time the loop completes.

Using the altitude, time delta and calculated velocity inputs, the expected apogee can be calculated. It is important to note that the control system will have a delay of over 1s after ignition to 1) reduce the chance of the air brakes deploying during powered flight, increasing chances for instability and failure, and 2) to make calculations of expected apogee easier, needing only to calculate coasting altitude and adding it to the current altitude to result in the expected apogee. To calculate the coasting altitude, rocket equations were used:

$$y_{apogee} = h + y_{coast} \tag{1}$$

$$y_{coast} = \frac{m}{2k}\ln\left(\frac{\mathrm{mg+kv^2}}{\mathrm{mg}}\right) \tag{2}$$

$$k = \frac{1}{2}\rho_{ISA} C_D A \tag{3}$$

$$\rho_{ISA} = \rho_0 \left(\frac{\frac{T_0}{\lambda}+h}{\frac{T_0}{\lambda}}\right)^{-\left(\frac{g}{\lambda R}+1\right)} \tag{4}$$

The control system inputs are noted as "h – altitude" and "v – velocity" (velocity is calculated using acceleration and the time delta). The end result is the variable y_{apogee} (equation 1), calculated from the height input (h) and the coast height (y_{coast}, equation 2). For increasing accuracy in the calculations, density was calculated using the International Standard Atmosphere (ISA) model (equation 4) which uses the height input to calculate the density.

The known constants of the above equations are as follows:

m	5.82 lb	Mass of rocket with empty motor.
g	9.8 m/s^2	Gravitational acceleration
C_D*	0.41	Coefficient of Drag for rocket (closed AB)
	0.70	Coefficient of Drag for rocket (open AB)
A	50.27 in^2	Cross Sectional area of rocket (closed AB)
	51.70 in^2	Cross Sectional area of rocket (open AB)
ρ_0	1.225 kg/m^3	Initial air density at ground level
T_0	288.16 K	Initial air temperature at ground level
λ	-0.0065 K/m	Constant for ISA density model
R	287.058 J/kgK	Specific Gas Constant

* The drag coefficients are estimates from simulation software. They are expected to be validated before competition

Below are schematics of the code and electronic wiring of the air brakes system.

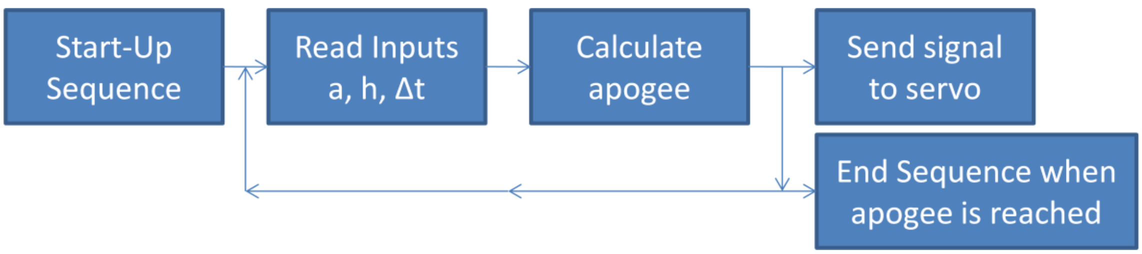


Figure 5: Air Brake Control System, Code Schematic

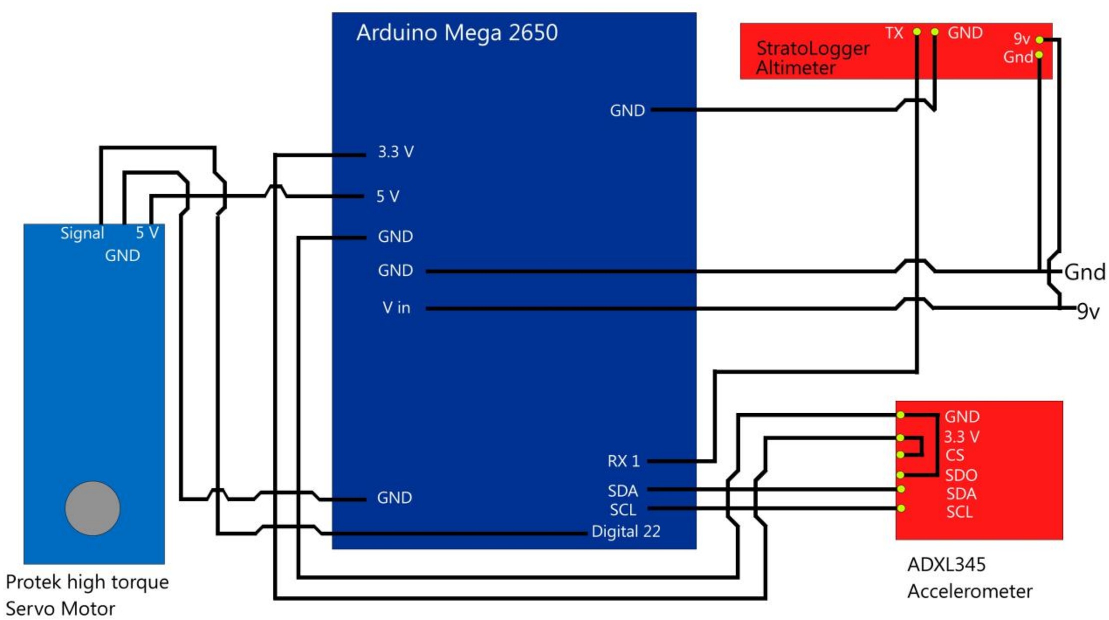


Figure 6: Air Brake Control System, Electronics Wiring Schematic

`

4.2 Fins

4.2.1 Carbon Fiber

The Fins were constructed from Divinycell Foam sandwiched between two layers of carbon fabric. An Aeropoxy resin and hardener was first applied equally to an area of plastic sheeting. Then, the carbon fabric sheets sandwiching the foam (cut into the shape of the fins) was laid between the plastic. Then, all air bubbles were removed from the resin/hardener mixture between the fabric and plastic. The edges were then sealed with AT-200Y Sealant tape and the sandwich vacuum sealed for 24 hours to allow the resin to cure with a smooth surface. After curing, the fins were then removed from the vacuum sealed plastic and the edges of carbon fiber trimmed to be smooth with the pre-cut foam shapes.

4.2.2 Magnet Application

The removable fin design considered was magnetic fins. The fins are through-the-wall mounted fins, but instead of epoxy to hold the fins to the body tube, a row of strong rare-earth neodymium magnets hold the fins to the motor mount tube. Each magnet has a pull of about 5 pounds each, so as more magnets are added, the pull of each magnet compounds.

The 3/8" disc magnets were epoxied in a row on both the motor mount tube and directly to the carbon-fiber fin tabs, making sure the polarities of each magnet matched up. Then, the fins could be inserted into the body tube fin slits and removed (after considerable force was applied).

5.0 Analysis

5.1 Flight Simulation

5.1.1 Flight Characteristics

Using OpenRocket for simulations since it is a free and fairly accurate rocket simulation software, we ran our rocket design to get a good idea of how the rocket would behave during flight (without air brakes). From the simulation we gathered the following data:

- Maximum Velocity: 606 ft/s (Mach 0.54)
- Maximum Acceleration: 615 ft/s (19 Gs)
- Time off Launch Rail: 0.12675 s
- Velocity of Rail: 65 ft/s
- Predicted Apogee (w/o airbrakes): 3370 ft
- Time to Apogee: 13.591 s
- Ground Hit Velocity: 28.6 ft/s

`

5.1.2 Recovery Characteristics

The expected recovery time is around 2 minutes after apogee.

5.1.2.1 Drogue Parachute (1st Event at Apogee)

The drogue parachute is designed to deploy at apogee, using the SL100 altimeter to control the detonation of the black powder charge. The descent rate is expected to be around 30 ft/s. To eject the drogue parachute, a 3g black power charge will be used, and the motor ejection (16s delay after launch) will be used as a backup.

5.1.2.2 Main Parachute (2nd Event at 500 ft)

The main parachute is designed to deploy at 500 ft as opposed to at apogee to minimize drift during recovery. The SL100 altimeter will control this black powder charge as well. There will be two altimeters used to ensure redundancy so the second event occurs for a safe recovery. To eject the main parachute, a 2-3g black powder charge will be used.

5.2 Air Brake Flap Analysis

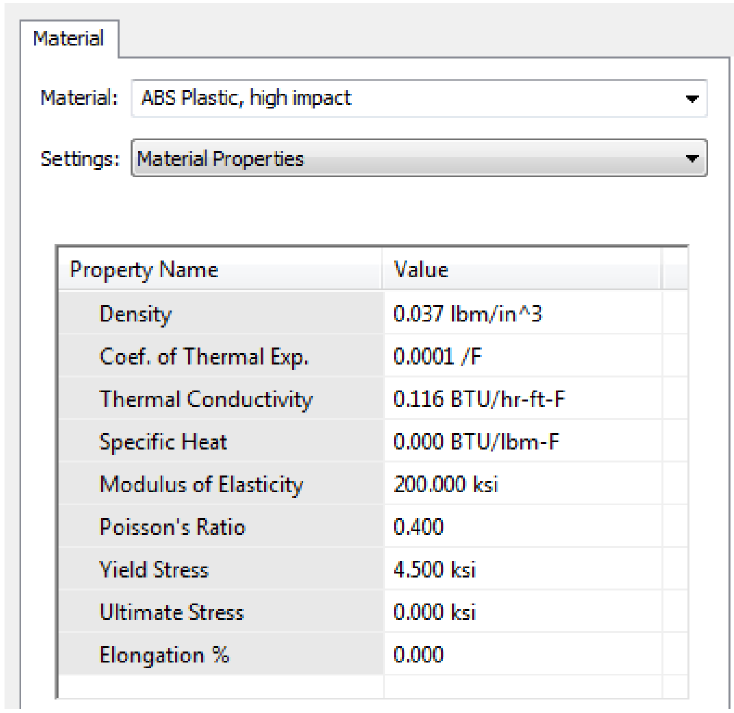
As most of the air brake components are 3D printed using high Impact ABS plastic, Finite Element Analysis was done to verify the part will withstand the loads seen on the part during flight. The most conservative load case was used, and only the flaps were analyzed. This is because the flaps will be the only parts seeing direct load. The most conservative load case used was assuming the flaps are open at maximum velocity and maximum acceleration. The maximum velocity directly relates to the maximum air pressure seen on the flaps, and the maximum acceleration relates to the maximum amount of force the flaps weight will impart on itself at the edges of the body tube where the flap is no longer supported. Solid Edge ST3 NASTRAN solver software was used since the model was made directly in Solid Edge to begin with. The pressure applied to the model was the dynamic pressure corresponding to the max velocity, and the acceleration applied was the max acceleration (19G's) seen during flight. The model was fixed at the washer circles where the ball bearing assembly clamps to the flaps, and movement in the +Z direction was restricted on the bottom surface where it is supported by the body tube and lower bulkhead assembly. The pressure was applied to the face that is exposed when the flap is fully open.

Dynamic Pressure Calculation:

$$q = \frac{1}{2}\rho v^2 = \frac{1}{2}\left(1.225\frac{kg}{m^3}\right)\left(606\frac{ft}{s}\right)^2 = 3.031\ psi \quad (5)$$

Looking at the results that follow, the 3D printed flaps are well within margin and will not fail structurally during flight. The maximum Equivalent Stress seen at boundary conditions is around 500 psi, and ABS High Impact has an allowable of 4.5 ksi. This gives the flap a margin of 8.

Here are the boundary conditions and loads applied for the analysis:



Material

Material: ABS Plastic, high impact

Settings: Material Properties

Property Name	Value
Density	0.037 lbm/in^3
Coef. of Thermal Exp.	0.0001 /F
Thermal Conductivity	0.116 BTU/hr-ft-F
Specific Heat	0.000 BTU/lbm-F
Modulus of Elasticity	200.000 ksi
Poisson's Ratio	0.400
Yield Stress	4.500 ksi
Ultimate Stress	0.000 ksi
Elongation %	0.000

Figure 7: ABS High Impact Material Properties from Solid Edge ST3

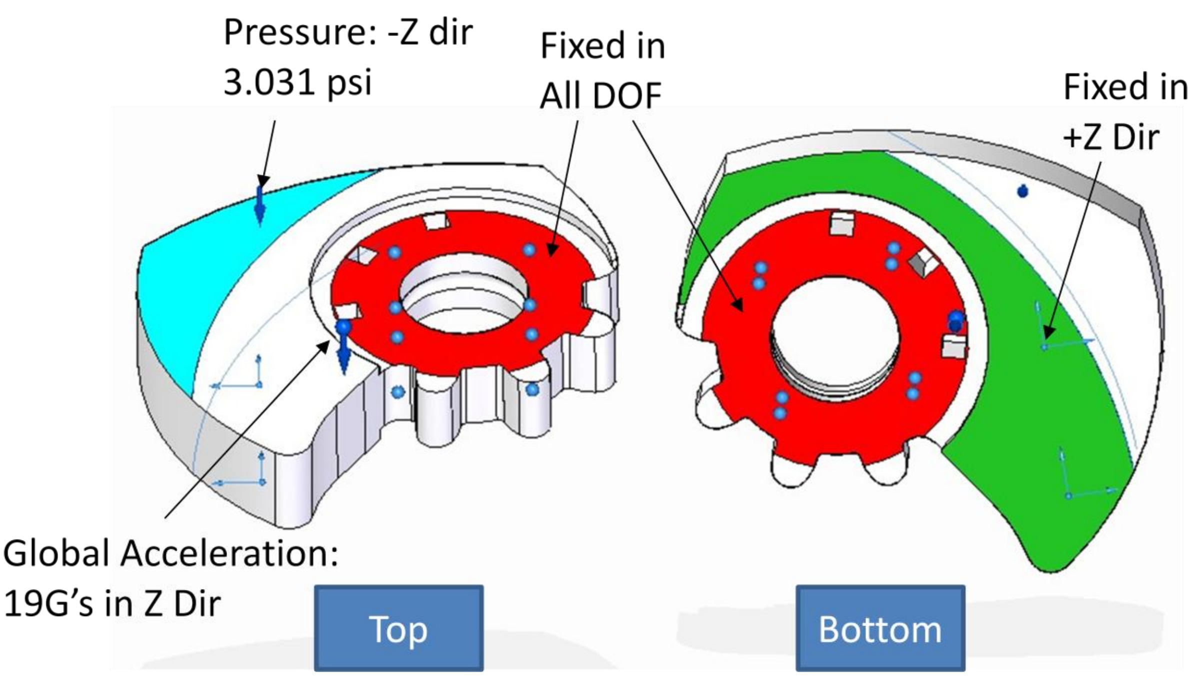


Figure 8: Boundary Conditions and Loads for Analysis

`

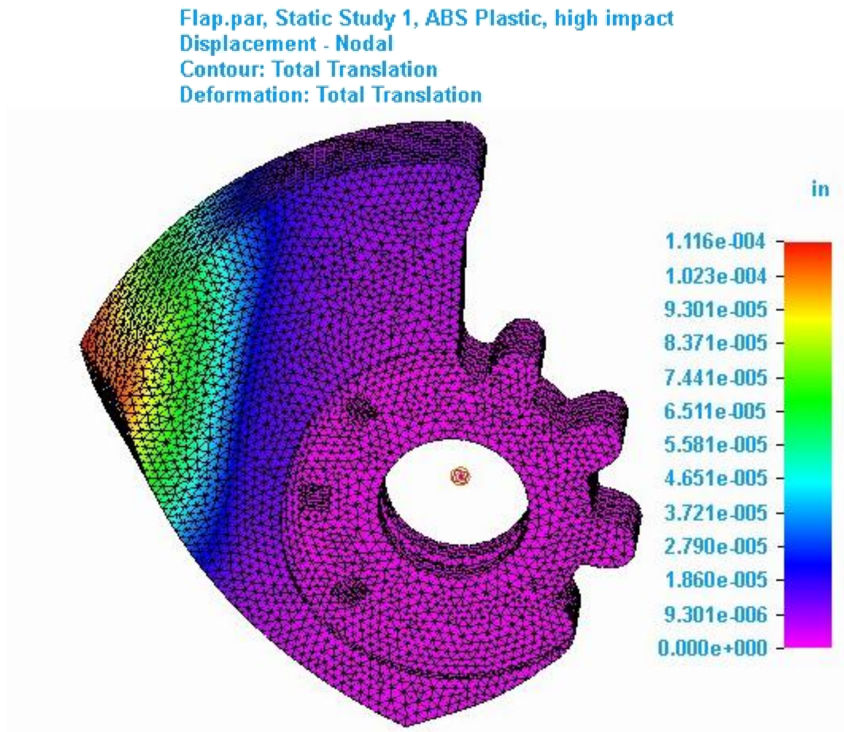


Figure 9: Total Deflection, Max @ Flap Tip, 0.1116 mils in -Z direction

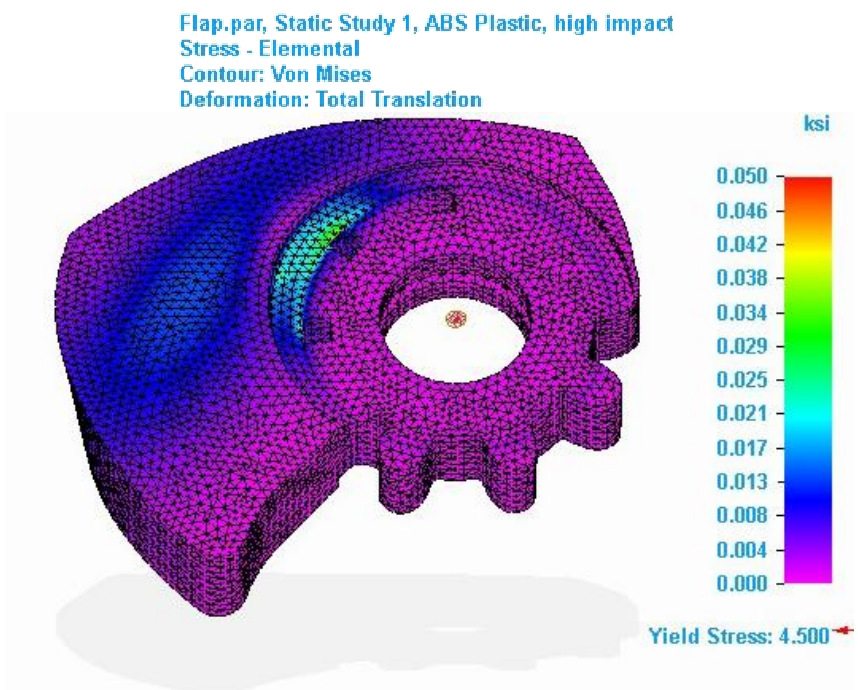


Figure 10: Equivalent Stress on Top Side of Flap

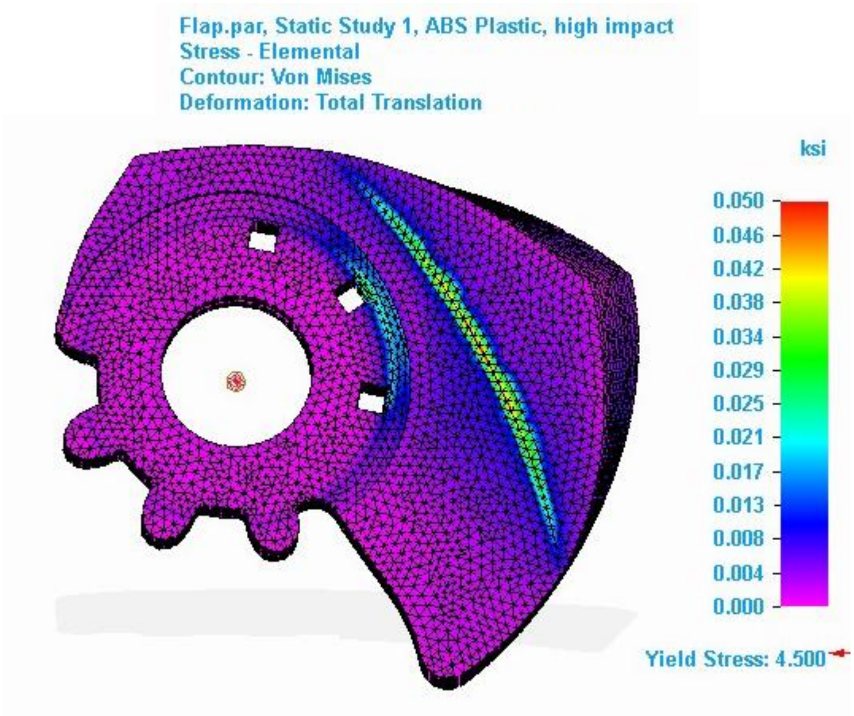


Figure 11: Equivalent Stress on Bottom Side, Max of 0.05 ksi at BCs.

6.0 Test Flight

The rocket was test flown on 4/13. It was a cold, 40°F day with winds around 15mph. All systems were operational besides the airbrakes, as the goal of the first flight was to verify the simulations and ensure the rocket was reaching an apogee between 3000 and 3500 ft. Therefore, the rocket was launched without any air brake control system, though the mechanics were included to simulate weight.

The magnetic fins were being used, and though the RSO questioned them slightly, the rocket was deemed safe and was put on the launch pad. When the rocket was launched, all went well during burnout, but then a gust of wind hit the rocket pretty hard and then suddenly the lower body tube ripped apart and the ejection charges went off (assuming due to safety reasons for not reaching apogee nor 500 ft for either events to occur).

Upon inspection of the rocket, it was evident that the magnetic fins were a cause for the destruction. However, the motor retainer (solid 6061 Aluminum) was shredded, and that is still being looked into. The reason the lower body tube was destroyed, was the magnetic fins were not restricted in tangential movement. The magnets did a perfect job keeping the fins from moving in the radial driection, and the bulkheads kept them from moving axially. However, when the gust hit the rocket, the fins were instantly torn sideways and ripped off the lower end of the body tube, as there was nothing restricting that movement but the thin body tube.

`

Lessons learned from the flight: when using removable fins, ensure movement in all directions is restricted and reinforced. Fins are the most important part of the rocket and when they fail, everything fails. The lower portion of the rocket is being re-built, but the control system may go untested for flight before the competition as there are no more launches we can attend before it.

7.0 Conclusions

In conclusion, the team is confident that the rocket will be able to perform within the competition parameters. The accuracy of the rocket is still untested however. After the failed test launch, the team feels more comfortable of what to expect come launch day. All of the modular designs implemented in the rocket are expected to help speed up assembly time to have more time to assess issues if they arise. All in all, the team is confident the rocket will perform largely as expected and would like to thank the Wisconsin Space Gran Consortium for the opportunity to participate in the competition.

`

8.0 Appendix

8.1 Launch Check-Off List

Assembly Check Off

1) Shock chord attachments are firm and the excess from the knot is taped
2) Routing between the switch and electronics is correct.
3) Test the electronics for continuity before e-match and black powder ejection charges are used.
4) Verify altimeter preset is 2, Main ejection at 500 feet, Telemetry is always on for the air brakes.
5) Verify all connections to the Arduino are good.
6) Verify Upper Body Tube, E-Bay, Middle Body Tube and Air Brakes are all connected properly with screws.
7) Fill PVC ejection cases with 2-3g of black powder after e-match is threaded through. Pack with "dog-barf" and taped.
8) Pack separation points with "dog-barf".
9) Ensure parachutes are folded properly and excess shock chord is wrapped and lightly tapes.
10) Use shear pins at Nosecone and Lower Body Tube.

On Launch Pad

1) Press switch to arm rocket
2) Verify the altimeter start-up sequence is correct
3) Verify Air Brake start-up sequence worked and are ready to go.
4) Insert e-match and secure

8.2 Arduino Code

```
/* --------------------------------------------------------------------------
  Icarus Flight Control System
  UC Stratocats 2013

  Accelerometer ADXL345     Arduino pin
  GND                        GND
  VCC                        3.3 V
  CS                        3.3V
  SDO                        GND
  SDA                        SDA (20 on Communication)
  SCL                        SCL (21 on communication)

  Servo PROTEK RC
  black                      GND
  red                        5 v
  White                      digital (22 on digital)
```

`

```
   PerfectFlite Stratologger
   pin 4 (TX)              RX1 (19)
   pin 5 (GND)               GND

---------------------------------------------------------------------------*/

#include <Wire.h>
#include <Servo.h>
#include <math.h>

/* -------------------------------------------------------------------------
                  ADXL345 Data Variables
 https://github.com/jenschr/Arduino-
libraries/blob/master/ADXL345/examples/ADXL345_no_library/BareBones_ADXL345.pde
---------------------------------------------------------------------------*/

#define DEVICE (0x53) // Device address as specified in data sheet - ADXL345

byte _buff[6];

char POWER_CTL = 0x2D;  //Power Control Register
char DATA_FORMAT = 0x31;
char DATAX0 = 0x32;         //X-Axis Data 0
char DATAX1 = 0x33;         //X-Axis Data 1
char DATAY0 = 0x34;         //Y-Axis Data 0
char DATAY1 = 0x35;         //Y-Axis Data 1
char DATAZ0 = 0x36;         //Z-Axis Data 0
char DATAZ1 = 0x37;         //Z-Axis Data 1
/*---------------------------------------------------------------------------*/

/* -------------------------------------------------------------------------
                  Arduino Variables
---------------------------------------------------------------------------*/
/* edit these variables */
int maxAltitude = 4000;  // the rocket WILL NOT go above this value
int rotateDegree = 45;   // degree amount for servo rotation
float mass  = 1.9553;    // mass of rocket in kilograms
float brakeDrag = .7;    // coefficant of drag with brakes
float noBrakeDrag = 0.4; // coefficant of drag without brakes
/*------------------------*/

float   altitude;      // measured in feet
int lastGoodAltitude;  // last altitude between 0 and specifed max feet
float expectedApoge = 0; // gets updated later

Servo motor;           // the servo motor
```

`

```
int pos = 90;            // position of the servo

int openPos = 135;
int closePos = 45;
boolean brakesOpen = false;  // brakes open/close

unsigned long accelStart = 0;
unsigned long accelEnd = 0;
unsigned long deltaT = 0;
long int iteration = 0;

float velocity = 0;
float acceleration = 0;  // measured in g's
float lastAcceleration = 0;

/* CONSTANTS */
float gravity = 9.80665;  // acceleration of gravity
float area = 0.008107;
float rho0 = 1.225;
float temp = 288.16;  // kelvin
float lambda = -0.0065; // k/m
float R = 287.06;
/*-------------------------------------------------------------------------*/
void setup()
{
  Wire.begin();       // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output. Make sure you set your Serial Monitor to the same!
  Serial1.begin(9600);

  motor.attach(22);

  motor.write(closePos);
  delay(1000);
  motor.write(openPos);
  delay(3000);
  motor.write(closePos);
  delay(3000);

  //Put the ADXL345 into +/- 16G range by writing the value 0x0B to the DATA_FORMAT register.
  writeTo(DATA_FORMAT, 0x0B);
  //Put the ADXL345 into Measurement Mode by writing 0x08 to the POWER_CTL register.
  writeTo(POWER_CTL, 0x08);
}

void loop()
{
```

```
 altitude = readAltitude();
 /*
 Serial.println("Altitude");
 Serial.println(altitude);

 Serial.println("    ");
 */
 accelEnd = millis();
 if(iteration > 0){
  deltaT = (accelEnd - accelStart);
  /*
  Serial.print("Delta T  ");
  Serial.print(deltaT);
  Serial.println("  ");
  */
 }
 accelStart = millis();

 lastAcceleration = acceleration;
 acceleration = readAcceleration();
 /*
 Serial.print("acceleration   ");
 Serial.print(acceleration);
 Serial.print("   ");
 */
 updateVelocity();
 /*
  Serial.println("velocity");
  */
 updateExpectedApoge();
  Serial.println(expectedApoge);

  if(expectedApoge > 3000 && brakesOpen == false){
    openBrakes();
    brakesOpen = true;
  }

  if(expectedApoge < 3000 && brakesOpen == true){
    closeBrakes();
    brakesOpen = false;

  }

 delay(100);
 iteration +=1;
}
```

`

```
/* ==========================================================================================
                           Read Acceleration Data
==========================================================================================*/

float readAcceleration() {
  uint8_t howManyBytesToRead = 6;
  readFrom( DATAX0, howManyBytesToRead, _buff); //read the acceleration data from the ADXL345

  // each axis reading comes in 10 bit resolution, ie 2 bytes.  Least Significat Byte first!!
  // thus we are converting both bytes in to one int
  float x = (((int)_buff[1]) << 8) | _buff[0];
  float y = (((int)_buff[3]) << 8) | _buff[2];
  float z = (((int)_buff[5]) << 8) | _buff[4];

  // convert the raw data to float's representing g's
  x = x * 0.0039;
  y = y * 0.0039;
  z = z * 0.0039;

  /* return the axis you want! */
  return y * gravity;
}

void writeTo(byte address, byte val) {
  Wire.beginTransmission(DEVICE); // start transmission to device
  Wire.write(address);            // send register address
  Wire.write(val);                // send value to write
  Wire.endTransmission();         // end transmission
}

// Reads num bytes starting from address register on device in to _buff array
void readFrom(byte address, int num, byte _buff[]) {
  Wire.beginTransmission(DEVICE); // start transmission to device
  Wire.write(address);            // sends address to read from
  Wire.endTransmission();         // end transmission

  Wire.beginTransmission(DEVICE); // start transmission to device
  Wire.requestFrom(DEVICE, num);    // request 6 bytes from device

  int i = 0;
  while(Wire.available())         // device may send less than requested (abnormal)
  {
    _buff[i] = Wire.read();    // receive a byte
    i++;
  }
  Wire.endTransmission();         // end transmission
```

```
}
/* ================================================================================
*/

/* ================================================================================
                              Read Altitude Data
==================================================================================*/

int readAltitude(){
int value = 0;

 /* wait to get to get to the end of a data transmission */
   while(Serial1.read() == -1){
 }

 while(Serial1.read() != 10 ){
 }
 /* wait until a full data set comes at us */


 /* read our actual data */
 while(1){

   char byteBuffer = Serial1.read();
   if(byteBuffer > -1){
         if(byteBuffer >= '0' && byteBuffer <= '9') //Is the character a digit?
      /*Yes, shift left 1 place (in decimal), and add integer value of character (ASCII value - 48)*/
            value = (value * 10) + (byteBuffer - '0');
         else
            /*No, stop*/
          break;
   }
 }
 // return a valid altitude
 if(value >= 0 && value <= maxAltitude){
   lastGoodAltitude = value;
   return value;
 }
 // If height is altitude, return last valid altitude
 return lastGoodAltitude;
}
/* ================================================================================
*/
```

```
/* ==========================================================================================
                          Servo Motor Functions
============================================================================================*/

void openBrakes(){
   motor.write(openPos);
}

void closeBrakes(){
   motor.write(closePos);
}

/* ==========================================================================================
*/

/* ==========================================================================================
                          Calculations
============================================================================================*/

void updateVelocity(){
  velocity = velocity + (0.5 * (deltaT*0.001) * (acceleration - lastAcceleration));
}

void updateExpectedApoge(){
   float h_m = altitude/3.281;
   float v_m = velocity/3.281;
   float density = pow((rho0*(((temp/lambda)+h_m)/(temp/lambda))),((gravity/(lambda*R))+1));

   float drag;
   if(brakesOpen == true){
     drag = brakeDrag;
   }
   else{
     drag = noBrakeDrag;
   }
   float k = 0.5*density*drag*area;
   float y = (mass/(2*k))*log(((mass*gravity)+(k*v_m*v_m))/(mass*gravity)) + h_m;
   expectedApoge = y*3.281;

}
```