"DESIGN AND FORCE ANALYSIS OF A WIND TUNNEL"

Due: 9 January 2016

Michael Kim	321821829
Adam Lee	322271750
Daniel Pekar	322210469
Kaarthic Pulogarajah	322210675
Gordon Winch	342212701

This page intentionally left blank

Table of Contents

1. Overview

This document will detail the components used, the mechanical and the electrical design, the testing procedures, and the building procedures of a wind tunnel analyzer. This wind tunnel is designed to measure the lift and drag forces of a given airfoil piece. An open air wind tunnel with a venturi section will be used to obtain the required similitudes in order to scale the forces and flow characteristics to an accurate representation of a full scale wing. The wind tunnel will be made of Bristol board laser cut into the desired shape and will be supported by laser cut wooden struts. The airfoils being tested will be 3D printed for maximum precision and to have minimal flaws in the design. The wind tunnel is to have a wide opening and converge to a narrow tube to increase the speed, as described by Bernoulli's principle.

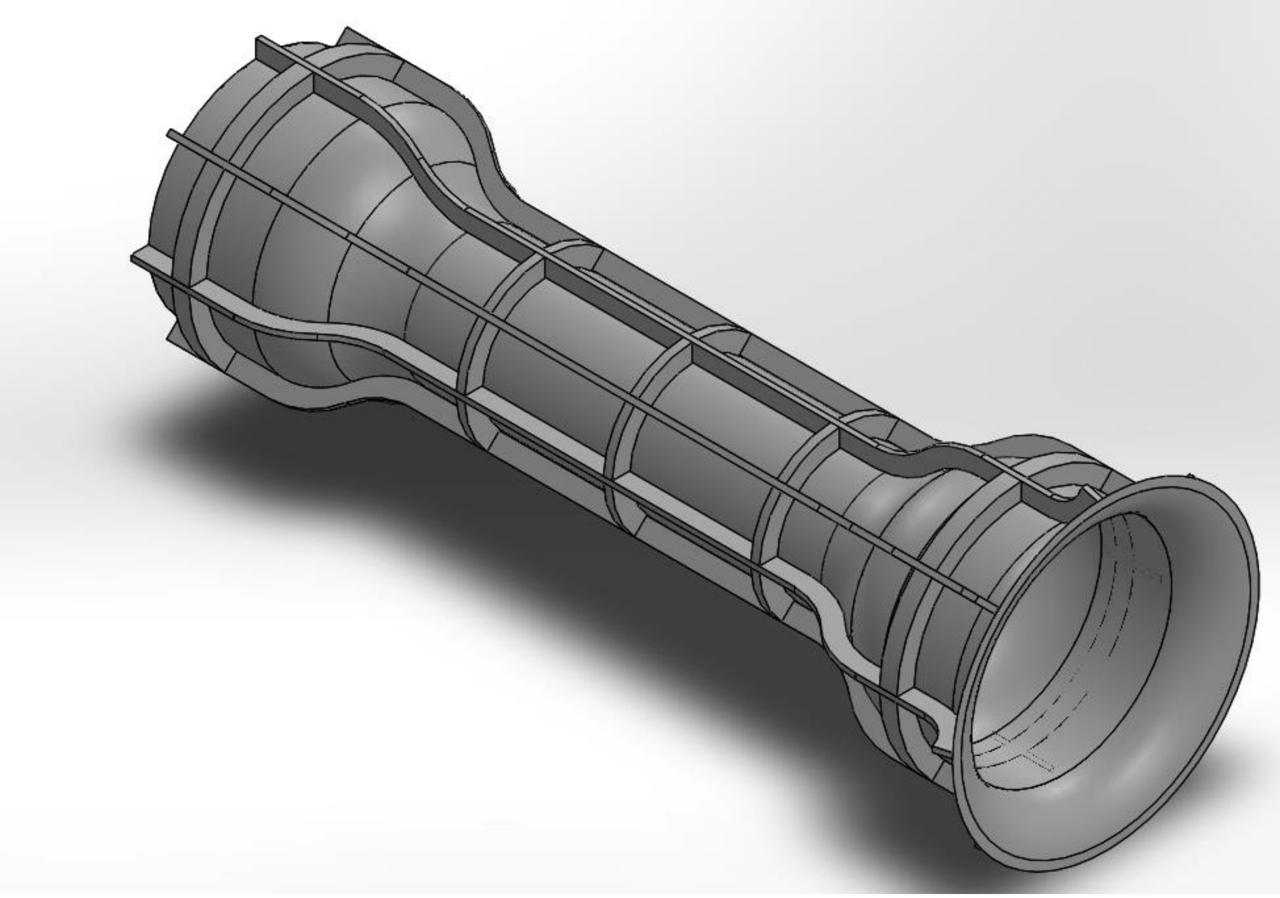
Several sensors will be used for determine the wind speed and forces on the airfoil inside the tunnel. A differential pressure sensor will be used to calculate the speed of the air and load cells will be used to determine lift and drag on the wing. This assembly will be controlled by an Arduino microcontroller that will run software to control the speed of the fan used to generate the forces. The graphical user interface will be able to store data, control wind speed, and switch between the readings that will displayed on 7 segment LED chips. Through this model, wings of small, subsonic airplanes can be scaled down and tested to determine the forces on it at a realistic wind speed without having to build a full size model.

2. Mechanical Components

2.1 Wind Tunnel Design

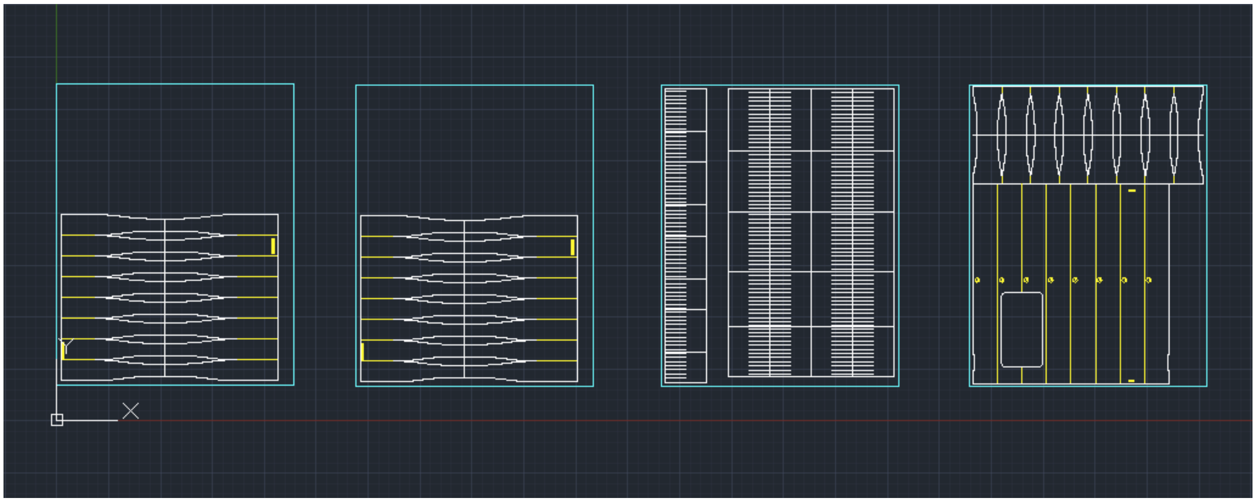
The wind tunnel will be divided into three main parts. The front section starts with a horn to allow a large volume of air to smoothly transition into the wind tunnel. The horn was modelled after a trumpet as trumpet horns are designed to expel air in a smooth and wide arc such that few vortices occur. This will make it easier to straighten the flow later on. The air then flows into a straight section with diameter of 25.4cm. This is done to make this end of the wind tunnel easier to support with wooden struts. It then narrows into the testing section, a venturi of diameter 15cm. The edges of the Bristol board curve into the smaller section in two sections of circular arcs. It curves such that the average change of degrees along the horizontal to the change in degrees of the circular arc is 1:1, therefore halfway through the curved section the angle is exactly 45°. This ratio of 1:1 was deemed acceptable after extensive testing of various ratios (3:2, 2:1, 1:2, 2:3) in flow simulation software. This will accelerate the air into desirable speeds of approximately 10ms^{-1}. The beginning of the venture contains a 10cm long flow straightener. The flow straightener is a 10cm rectangular grid with spacing of 1 cm. While hexagonal/honeycomb type straighteners are more effective, the rectangular grid was chosen, as it is much easier to construct and the increased effect of the hexagonal grid is unnoticeable at such low speeds.

This leads to the airfoil and force sensor section. After a 10cm interval, the anemometer is placed. It was found that 10cm is sufficient to settle the disturbances in air flow from the airfoil after flow simulation testing. Finally, the wind tunnel curves out, in the same way it curved in with a 1:1 ratio, into the small section containing the motor and the 9in propeller which suck the air into the tunnel. Along the wind tunnel, there are 5 struts which hold the cylindrical shape of the tunnel and 8 that hold the lengthwise shape. 3 of the cylindrical struts are in the test section, equally spaced apart. The other two are located at the front and back section of the tunnel. Each of the 8 lengthwise struts follows the curvature of the tunnel they are spread apart equally, 45° apart from each other.

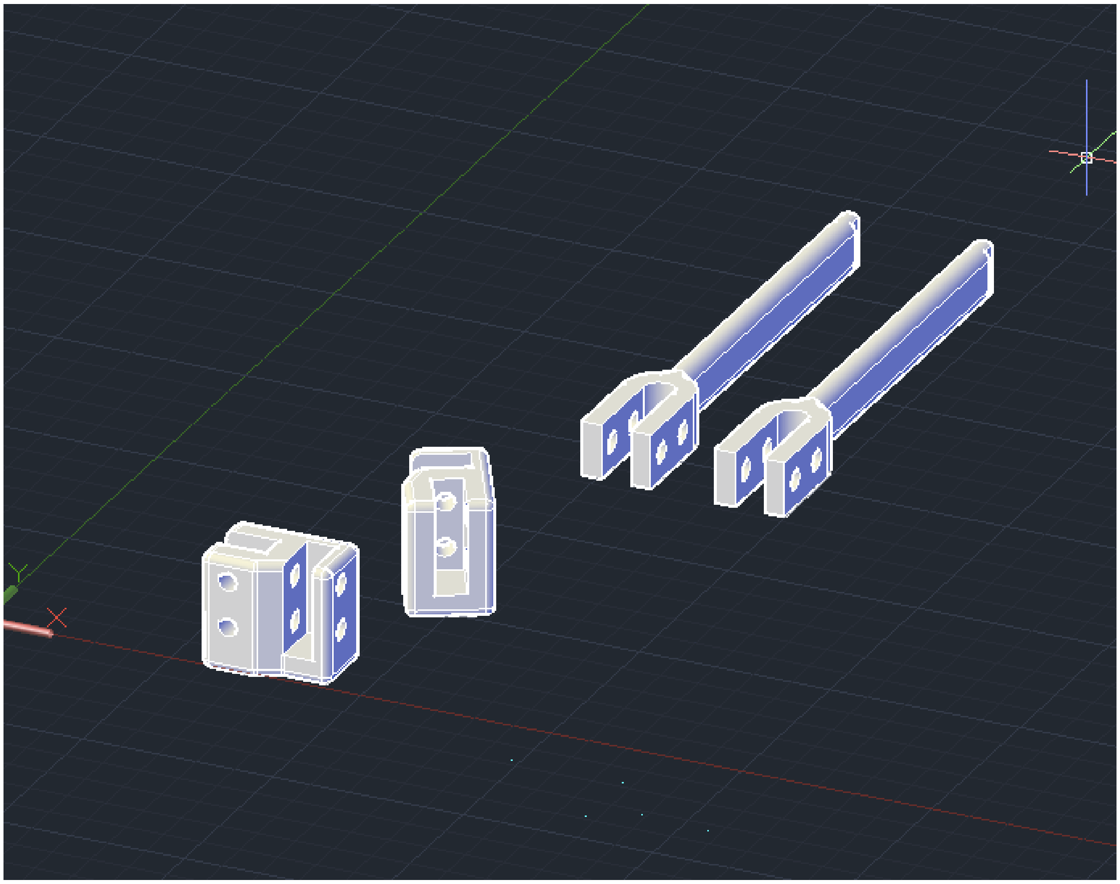


This image shows a 3D representation of the wind tunnel exterior. The basic shape of the Bristol board walls as well as the wooden supports can be seen. The front end of the wind tunnel is shown on the bottom right corner. There are eight struts that run length wise along the tunnel and 5 rings that these struts attach to. This 3D representation was made with the Solidworks software.

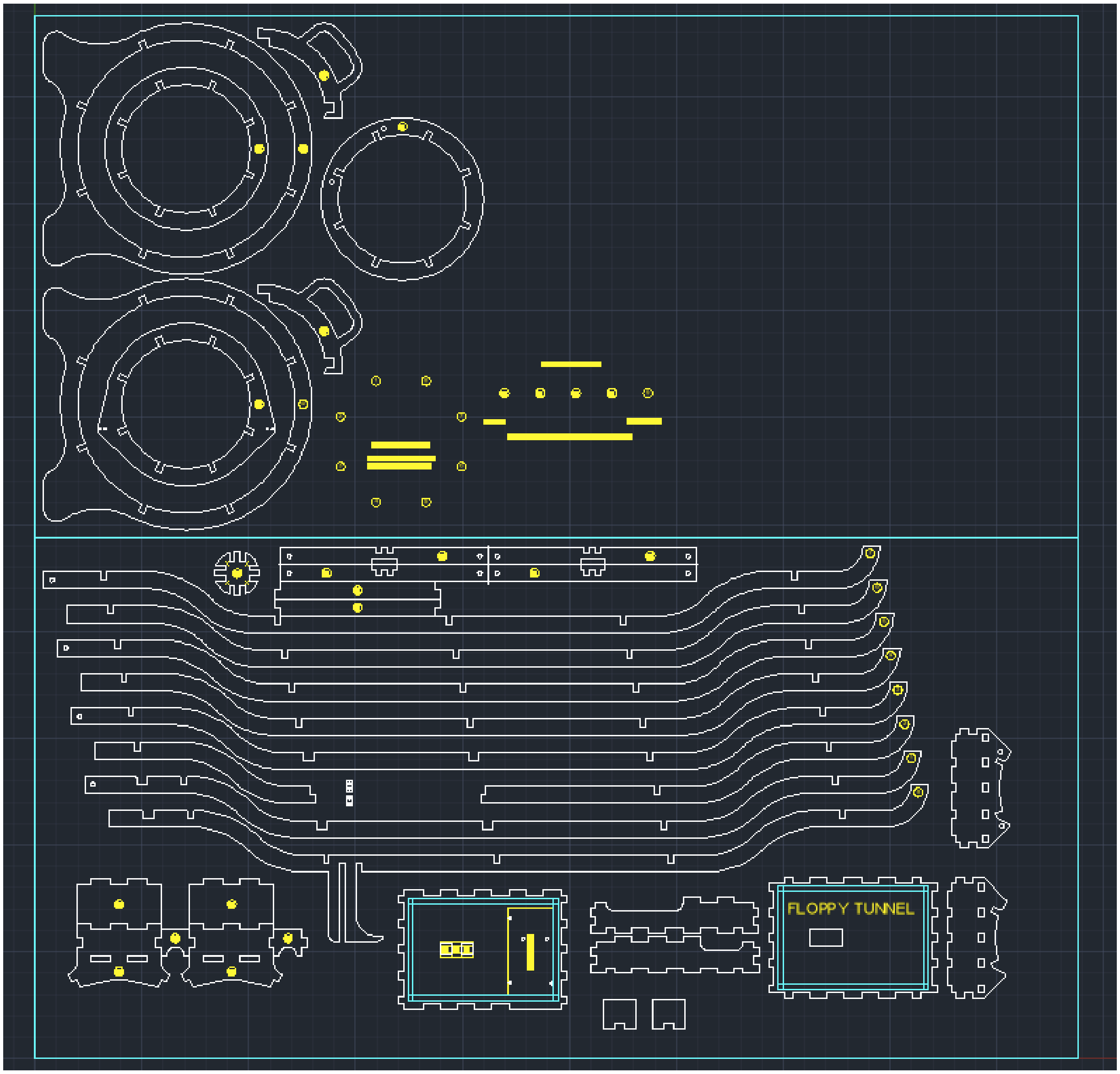
The final cross-sectional shape of the wind tunnel required much consideration and deliberation. There were three basic shapes considered for the cross section of the wind tunnel: a square, a circle, or a many sided polygon (anywhere from octagon up to a 26 sided polygon). The advantage of a square cross section is that it is easy to design and build. However, boundary layer effects near the corners severely affect the flow characteristics, so that shape was shown to be a bad choice. The advantage of a circle is that it provides optimal flow characteristics, but is harder to design and build. The many sided polygon is an in-between solution with smaller boundary layer affects than the square, but not quite as good as the circle. The circle was chosen as the best design and found ways to keep the structure stable and intact as that would optimize readings from the force sensor and help keep incorrect assumptions from being made about the airfoil during the testing stage.



The four sheets of Bristol board used to cut the pieces of the wind tunnel. The two sheets on the left contain the 16 slices used to create the circular venture. The shape of these curves is sinusoidal in nature. Middle right are the interlocking pieces used to make the flow straightener. Far right are the 16 sections for the trumpet, created in a similar manner to the Venturi, as well as the test section of the wind tunnel.



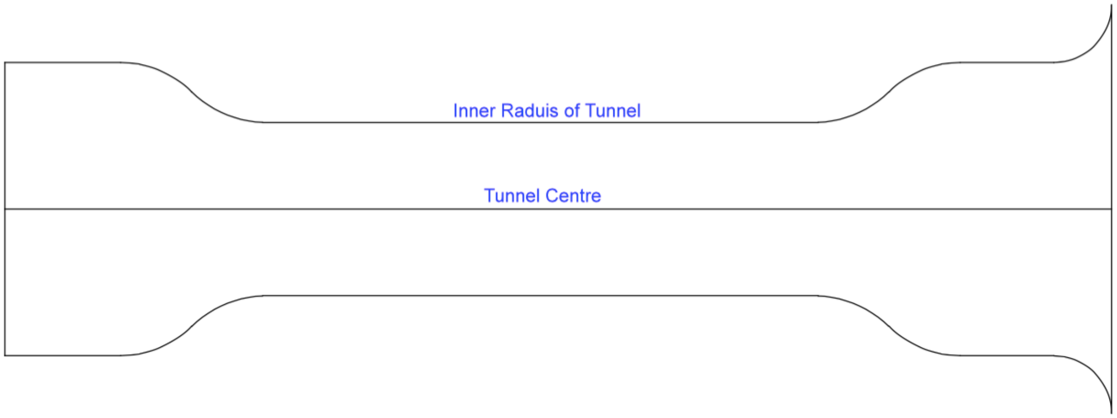
The four 3D printed pieces used in the load cell assembly. On the left are the two connector pieces that join the load cells at a right angle and on the right are the two spikes that lead from the load cell to the wing inside the wind tunnel.



The two sheets of Particle Board used to create the structure of the wind tunnel. On the top are the five ring pieces that hold the circular shape of the tunnel, on the bottom are the eight long that form the changing shape of the wind tunnel. Above the long pieces on the lower sheet are the parts of the motor stage, and below are the pieces of the electronic control box and ESC module. The foot-shaped protrusion on the bottom long piece is the stand for the anemometer.

2.2 Bristol Board Venturi Design

Two important components in the mechanical layout of this wind tunnel are the fore and aft venturi. These sections control the change in velocity and pressure that occurs as air enters and leaves the test section of the wind tunnel. As such, the smoothness and continuity of their design is paramount to achieving stable, laminar airflow inside the test section. Having opted to construct a circular wind tunnel with a skin made from Bristol board, it is necessary to create a smooth, continuous surface which transitions from the large radius of the front of the wind tunnel to the small radius of the test section, and back out again. This smooth change in radius also occurs in the trumpet at the front of the tunnel. The profile of this radial change is shown below.



Inner profile of wind tunnel. As can be seen from the diagram, curvature occurs during both venturi, as well as at the trumpet on the front of the tunnel.

Since Bristol board does not stretch or deform around the rounded shapes inside the tunnel to form a smooth surface, the only realistic solution is to approximate smoothness with many small strips of Bristol board with small angles between them. In this regard, there are two possible solutions.

First, a series of arcs of Bristol board can be created so that when their ends are taped together, they form bands with an inclined surface, as if a small horizontal cross-section were cut out of a traffic cone. By varying the length and curvature of these arcs, the smooth shape shown above can be approximated. The major advantage of this design is that it would provide a smooth transition between the curved and straight sections. The major disadvantage is that it presents many edges perpendicular to the direction of flow, edges that may or may not be entirely smooth. This creates a rougher boundary layer, which is more likely to cause turbulence.

Another solution is to could cut long strips of paper that run the length of the radial transition, whose width is a fixed percentage of the diameter of the circle. Rather than a circle, this creates a many-sided polygon that closely approximates the shape of the circle. The advantage of this setup is that the joints between pieces of paper run parallel to the direction of flow, minimizing disturbance. The disadvantage is that it makes the transition to steady radius more difficult.

Given these two options, the second one was selected as preferential. The need for a smooth boundary layer was seen as more important that creating a perfect transition between changing and static radius. With enough strips, the polygon will so closely approximate a circle that this transition will be relatively straightforward. In addition to this, the transition with the greatest potential to affect airflow on the wing (the entrance into the Test Section) occurs in the middle of the flow straightener, so any turbulent effects would be negated. Thus, lengthwise strips were chosen as the optimal method of transitioning between radii. The next challenge was to model shape of these strips to an optimal design the strips that constitute the venturi must be modelled first. First, the number of strips to be used must be chosen. It was reasoned that, given the 8 struts that run the length of the tunnel, a multiple of 8 would fit best. 8 itself provides a very rough octagonal shape, and makes transitioning smoothly to the round wind tunnel that much more difficult. However, increasing the number of strips to 24 or 32 would make the strips entering the tunnel 1.96 and 1.47 cm wide, respectively. These widths were deemed too small to be reliable assembled. Thus, 16 strips were chosen.

In order to create the strip, width as a function of distance along the border of the tunnel must be found. Both of these values can be assessed as functions of theta, the angle along the curve of the wind tunnel. Distance along a circle is given by

$$d = \theta r$$

Where,
θ = Angle (degrees)
r = Radius of the curve

Next, the radius of the wind tunnel (denoted as R) must be found as a function of θ, where R_0 is the radius at the center of the arc. This is given by:

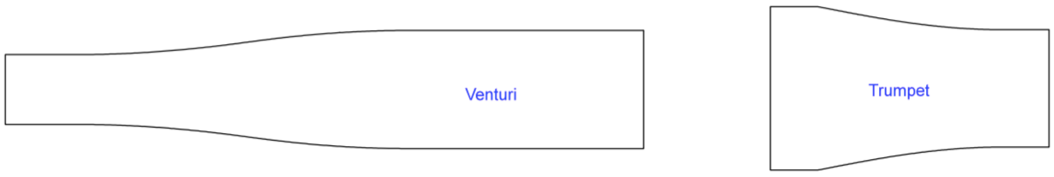
$$R = r \sin\theta + R_0$$

Where,
θ = Angle (degrees)
r = Radius of the curve

Using these two equations, we can find the radius of the tunnel as a function of distance along the tunnel. To find the width of the strip, we must first remember that the strips are flat. This means that half a strip forms a right-angled triangle with the centre of the tunnel. The angle of the corner of this triangle touching the centre of the wind tunnel is half of one sixteenth of one full revolution, or 11.25°. Thus, the width (w) of half of a strip is given by:

$$\frac{1}{2}w = R\sin 11.25°$$

Using this, we can plot the shape of the strips in 2D and create the following drawings:

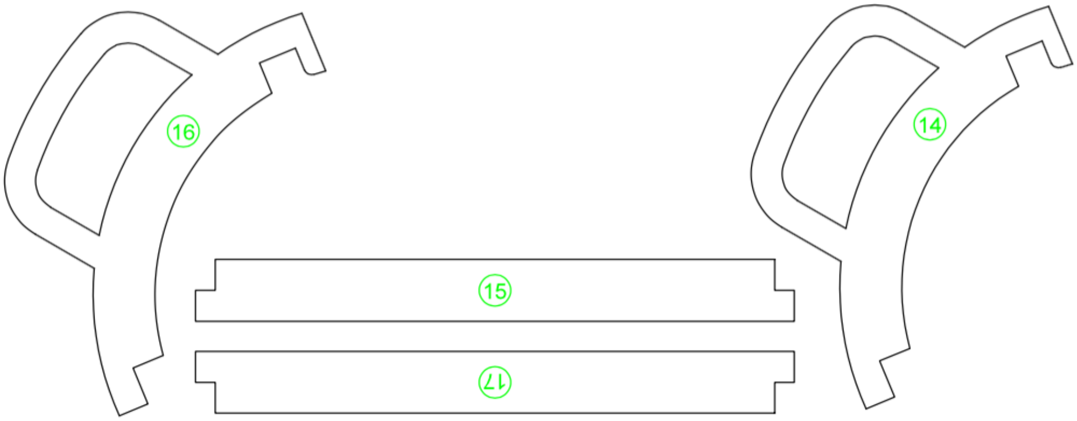


This image shows the shape of the strips used to create variable radii in the wind tunnel. On the left, the radius goes from large to small. On the right, the rectangular section on the left hand side of the trumpet actually folds over the outside of the tunnel. The curved part of the trumpet then begins by pointing straight in to the centre of the tunnel, and then slowly curves out to a flat cylinder.

As seen above, the equations used create shapes that are to be expected. The precision of the laser cutter to be used to cut these pieces will eliminate human error in cutting the curve, so the proper shape of the venturi should be preserved.

2.3 Viewing Port Design

An essential requirement in wind tunnel design is access to the inside of the tunnel. This allows switch out different airfoils, as well as construct, inspect, and repair the inside of the wind tunnel. Another necessary feature is a viewing port into the wind tunnel. This allows the operator of the wind tunnel to easily judge what is happening inside the tunnel from a mechanical standpoint (for instance, erroneous readings from an improperly mounted airfoil). Viewing ports would also enable the assessment of smoke tests, or other such common methods of visualizing the airflow moving around the wing. There is one downside that both of these features share: they both require breaking the smooth skin of the wind tunnel to be implemented. Breaking the skin disturbs airflow and can affect the output of the wind tunnel, so it should be kept to a minimum. Given this, it was decided that these two features would be combined, i.e. a window inside a door. This leaves only one interface to perturb the smooth flow of air, minimizing the effect on the conditions inside the tunnel.



This image shows the four structural pieces of the viewing port. Parts 14 and 16 run parallel to the supporting hoops, and parts 15 and 17 run flat along the wind tunnel.

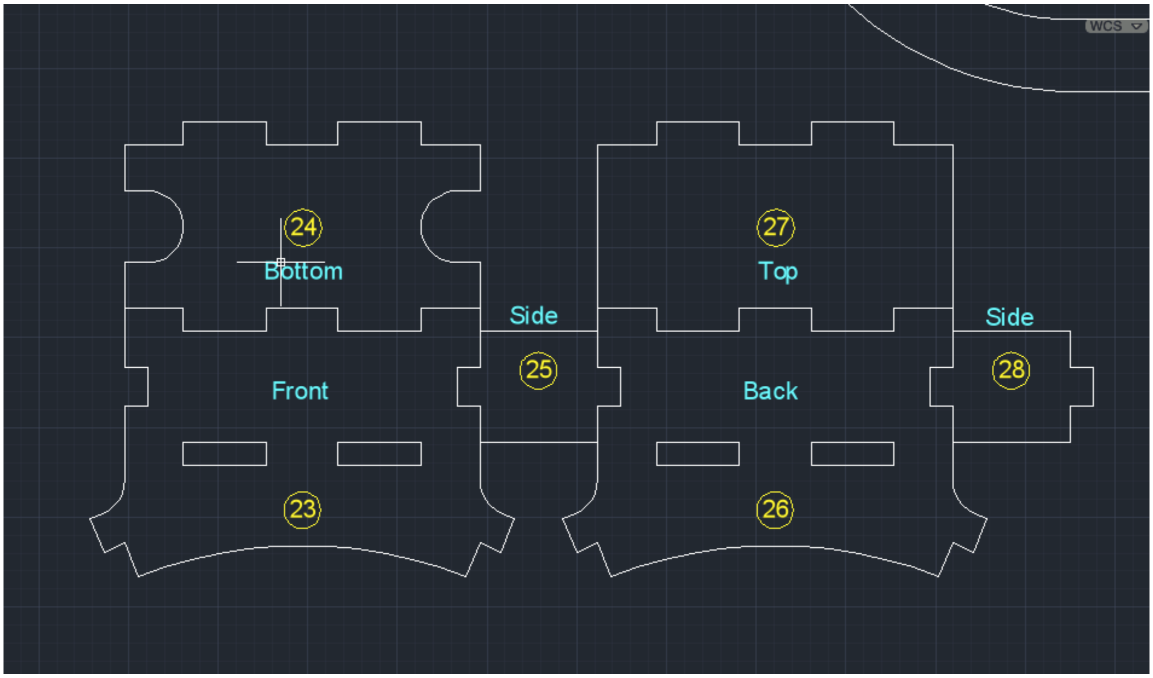
In the above image, the design of the four viewing ports can be seen. This module fits between the middle and rear hoops in the test section, hanging from the uppermost lengthways beam. To assemble the viewing port, parts 15 and 17 are glued into the inner sides of the notches in 14 and 16, creating a rectangular shape. This inner surface is then furnished with a rectangle of acetate, creating a transparent and removable viewing piece. The handles on pieces 14 and 16 allow for easy removal and replacement of this component. The cut-out in the Bristol board skin of the wind tunnel is slightly smaller than the footprint of the viewing port, allowing for some overlap of Bristol board and acetate. This creates a more airtight and aerodynamic junction between the two.

The viewing port on the wind tunnel is innovative in its dual function: it gives both a view of the airfoil and mechanical access to the inside of the test section. This is an essential component of the mechanical design of the wind tunnel.

2.4 ESC Module Design

Several factors were taken into consideration in the design of the container for the batteries and Electronic Supply Control. First, the length of the wires between the motor and the ESC mandated that they be no more than 20 cm apart. While purchasing additional wire was a possibility, it was considered a needless complication in an already complicated project. Thus, the box for the ESC and batteries would have to be located in the aft section of the wind tunnel.

Next, from the principles of modular design, it is important to have a removable box that contains the batteries and ESC would better facilitate testing and changing of the components inside. By containing them in a box, the concern of running wiring again each time the circuitry must be changed is reduced. Thus, a removable box was designed.



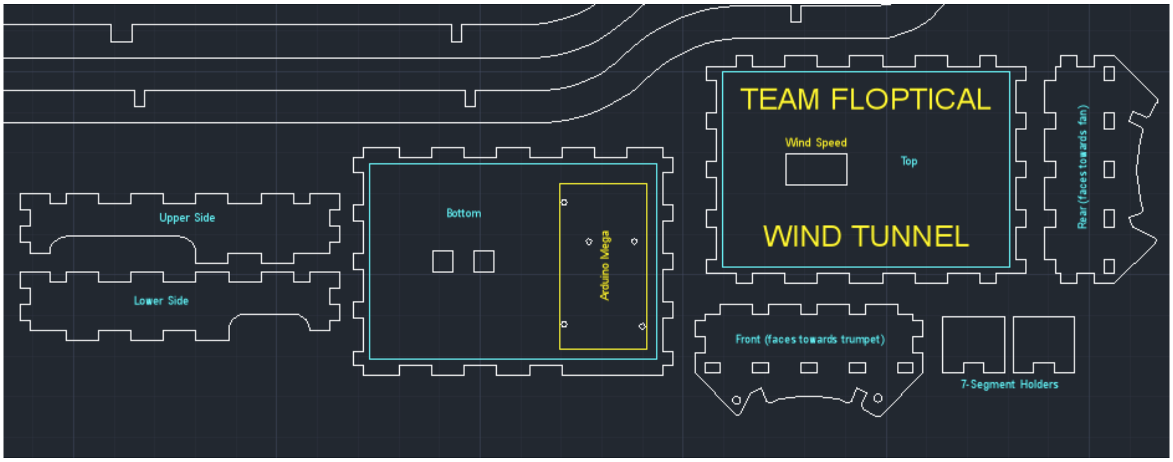
ESC Module. Note the curved base, fitted to mount on top of the circular wind tunnel. Holes are cut in the bottom piece to facilitate discreet exit of control wires.

This container had to be of sufficient size to hold both the battery pack and ESC, and had to mount firmly on the existing wind tunnel structure. It was decided that the optimal way to do this would be to create a base for the box that mimicked the shape of the upper sections of the wind tunnel hoops, fitting easily in line with existing parts of the wind tunnel. The bottom of the box was designed with holes big enough to fit the necessary wiring through, and the box was given a removable top that can be pried off to repair or replace the components inside.

2.5 Electronics Module Design

As with the ESC module, the assembly of electronics in a modular container was considered to yield some significant advantages. First, they could be installed and tested in their final environment with relative ease, as the box is small and easy to transport. Second, this installation would not take up time with the wind tunnel structure, thus precluding simultaneous testing of other components. Third, the modular container makes it relatively simple to remove, test, repair, and replace electronic components in the highly complex circuit. Finally, by grouping all electronics together in a protected box, they can be neatly kept out of sight and out of harm's way, and save running large bundles of wire around our wind tunnel connecting various components.

The first step in designing the electronics control box is deciding where to place it on the wind tunnel. Several factors contributed to this decision. The box would carry the 7-Segment display, so it was important that it be mounted in a place of convenience to the user. Placing the box in a central location on the wind tunnel made logistical sense, as this requires less extensive wiring to connect sensors and actuators to the central control station. Lastly, the box was designed to be removable, so a position that facilitated that ability was considered favourable.



This image shows the 8 components of the electronics box. Note that white lines are cuts, yellow lines are going to be written on the material and blue are simply for viewing purposes and will not be printed on the final material.

Given these criteria, it was decided that the electronics box should be attached to the outside of the test section, to the right of the window. This is a central location, and would provide all relevant information (i.e. the inside of the wind tunnel and the wind speed readout) directly to the viewer.

This electronics box will contain the majority of all the electronic components involved in the project, i.e. the Arduino Mega, the 7-segment display and encoder chips, and the load cell amplifiers. The differential pressure sensor used to measure wind speed will also be located inside the box, with vinyl tubing running from the anemometer stage inside the wind tunnel through two holes in the upper side of the box and onto the leads on the sensor. There are five holes cut in the bottom of the box, corresponding to the 5 holes found in an Arduino Mega.

These will be used to bolt down the Arduino. Other holes will later be drilled to secure the Load Cell Amplifiers. These holes cannot be pre-cut as the LCA supplier does not supply mechanical drawings of the component, and the exact dimensions are therefore unknown. There are two openings for wires in the box: one on the bottom side for the USB plug connecting to the Arduino, and one on the bottom of the box close to the upper edge, for wires leading to the Load Cells and ESC. The 7-segment displays will be mounted on the top of two pieces of particle board planted in the floor of the box, so that they poke through the lid and lie flush with it. There may be some sanding required to fit the displays across the ½ inch combined thickness of the two sheets, depending on the uniformity of the underside of the chips.

Testing of electronics will occur inside this box, with load cells attached to the Load Cell Testing Stage nearby. When the time comes to attach it to the wind tunnel, the box slides into notches in the long struts on the upper half of the wind tunnel, next to the window. It is then bolted to the front hoop in the test section for added security. Once this is fastened, the load cells will be bolted in place and the pressure sensor and ESC will be connected.

Accessing the box is relatively simple; the top cover is simply pried off of the top of the box. This will be the only part of the box structure that is not glue together. In order to disconnect the box from the tunnel, the load cells must also be removed as they are soldered directly to the LCA chips contained inside the box. The leads from the ESC go directly to the Arduino and consequently are not soldered, so they can be removed with relative ease. The anemometer tubes can be pulled off the differential pressure sensor inputs and pulled out of the side of the box. The bolts can be undone and the box removed.

The Electronics Box is a major piece of the operational wind tunnel and will play a central role in its function. It houses the central circuitry of the project protecting it and providing a removable stage on which to build the circuitry.

2.6 Airfoil Designs

Four airfoils will be used to observe the capabilities of the wind tunnel. To test the abilities of the force sensor layout, airfoils with altering lift and drag coefficients will be used. This will ensure the load sensor can detect forces in various orders of magnitude with high fidelity. The lift and drag forces will be used to calculate the coefficients of lift and drag. Those will then be used to calculate the coefficients of lift and drag. The coefficients will be compared to any known sourced values so that the abilities of the wind tunnel may be gauged. To calculate the coefficient of drag, the following force equation will be used; most of the values are known:

$$F_D = \frac{1}{2}\rho u^2 C_D A$$

Where:

F_D = Force of Drag (N)
ρ = Density of air (kg m^{-3})
u = Flow velocity (ms^{-1})
C_D = Coefficient of Drag
A = Reference Area (m)

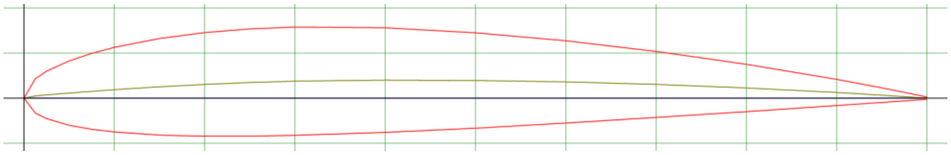
Since the density of air at room temperature (20 degrees Celsius) is 1.225 kgm^{-3}**,** the reference area will be known per airfoil, flow velocity will be calculated with the anemometer, and the force will be measured, then the coefficient of drag can be calculated and compared to sourced values if they exist. A similar process will be used for the coefficient of lift, where the following equation will be used:

$$L = \frac{1}{2}\rho v^2 A C_L$$

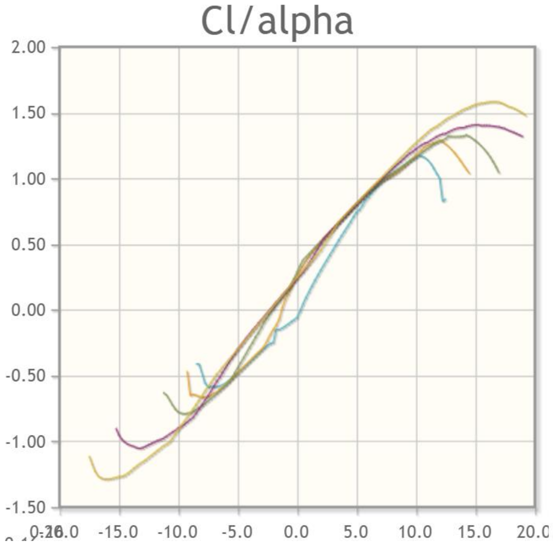
Where:

L = Force of lift (N)
ρ = Density of air (kg m^{-3})
v = Flow velocity (ms^{-1})
C_L = Coefficient of Drag
A = Reference Area (m)

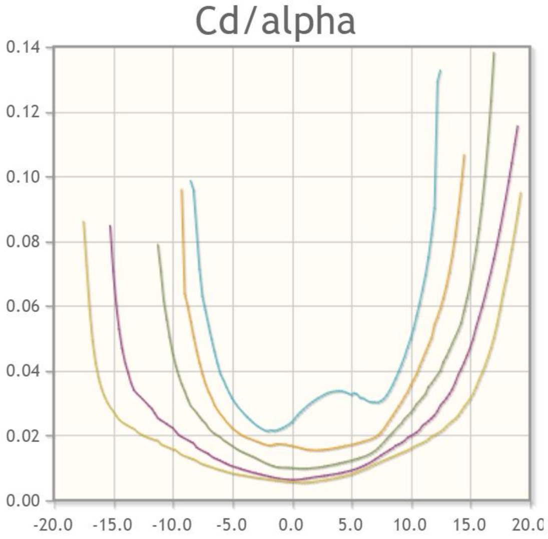
The density of air, velocity, reference area, and lift force will be sourced or measured in the same way they were measured for the drag coefficient. Comparing these two coefficients will allow for insight as to how accurate the wind tunnel is. The first shape will be that of a basic geometric shape, extruded. A short cylinder was chosen, as that has a very high coefficient of drag of 1.15. It will be symmetric across the horizontal axis and will therefore not create lift. The second shape will be a NACA 2412 airfoil extruded. This specific airfoil will be used as it from the well-known Cessna 172 civilian airplane. This plane flies at comparatively slow speed between 100 and 200 kmh^{-1}. It has extensively tested flight characteristics, most importantly lift and drag coefficients according to angle of attack (α). This will give us a high coefficient of lift as the shape is designed to do that, and a low coefficient of drag.



This diagram shows the cross section for the NACA 2412 airfoil. The line in the centre of the airfoil is the middle of each vertical slice while the black line is the chord length. Reference areas for drag and lift will be calculated computationally with the assistance of the 3D CAD program Solidworks.



This graph demonstrates the coefficient of lift in response to changes in angle of attack at different Reynolds numbers. At an angle of 0, the coefficient is around 0.4.



This graph demonstrates the coefficient of drag in response to changes in angle of attack at different Reynolds numbers. At an angle of 0, the coefficient is around 0.1.

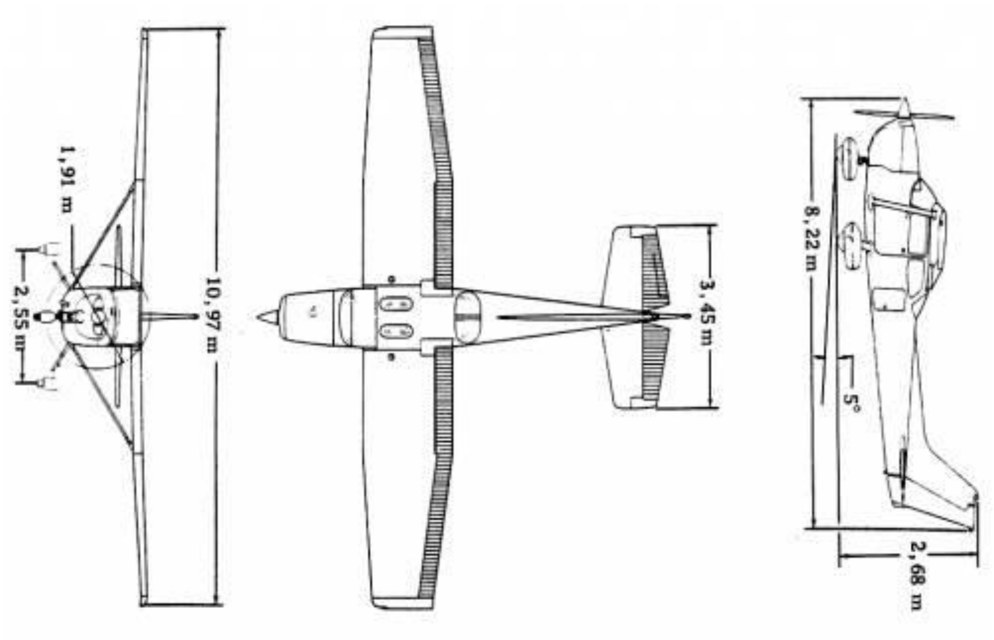
To create the third airfoil, the NACA 2412 will be stretched in the vertical by a factor of three. This should increase the reference area for drag, but not lift. This will produce a higher coefficient of drag. Finally, a model airfoil will be taken from a Cessna 172 and used to provide an airfoil with a varied profile along two axis. This will allow for 3D measurements of lift and drag.

2.6.1 Attaching Wings to Load Cells

The load cells must somehow be connected to the airfoil in order to detect force. Using a cylindrical dowel or string will not work as the dowel will induce drag not found on the airfoil and the string will only be able to support the airfoil from above, not below. Thus, two blades will be used. They will originate from each respective load cell holder and reach the airfoil. There will be a hole in the top of the wing and a slit running from the back of the wing such that they may be slid onto the blade. Since the blade is rigid, it does not bend and fully supports the airfoil. As the blade is very thin it does not noticeable affect drag. Rounding the pegs will allow the wing to be rotated.

2.7 Similitude

To ensure the wind tunnel scales up, three types of similarities will be used. The first is geometric similarity. Geometric similarity is when the model is scaled version of the original. Since the NACA 2412 airfoils are being scaled via AutoCAD to acceptable sizes, the geometric similarity for three of the airfoils is guaranteed. The real size of the wing is 4.66m x 1.63m x 0.47m, which is a 9.91:3.47:1 ratio and the size of our models will be 10cm x 4cm x 1cm so the geometric scaling is accurate. As there is no original large rectangular prism, there is no scaling factor for it.



This schematic shows the dimensions of a Cessna 172 which uses a NACA 2412 airfoil.

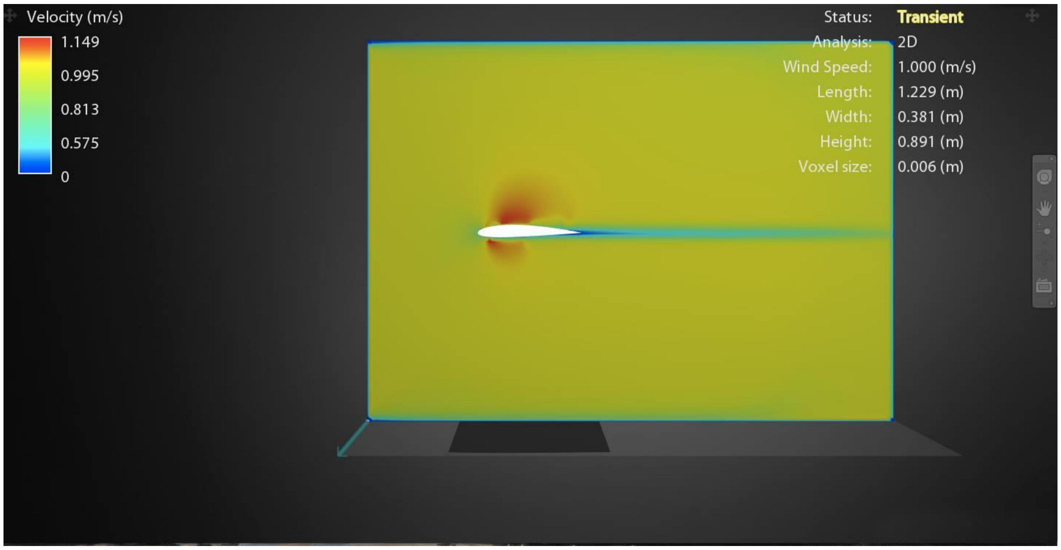
The second type of similarity is kinematic similarity. This means that the fluid flow of the model undergoes similar rates of change as the real application. In other words, the fluid streamlines are similar. The Reynolds number was used to determine that our model is kinematically similar to the life sized wings. The Reynolds number dictates the flow characteristics of a given fluid. It is a measure of inertial forces divided by viscous forces. The equation is:

$$\mathrm{Re} = \frac{\text{inertial forces}}{\text{viscous forces}} = \frac{\rho \mathbf{v} L}{\mu} = \frac{\mathbf{v} L}{\nu}$$

Where:

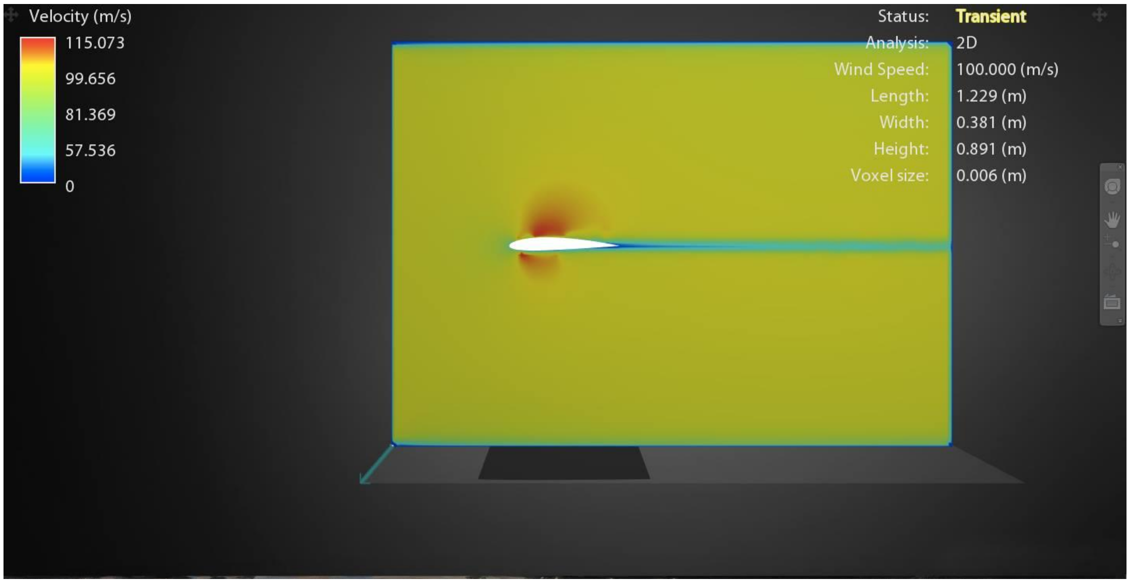
$\mathbf{v}$ = velocity of the object relative to the fluid (ms^{-1})
L = characteristic length (m)
ν = kinematic viscosity of the fluid (m^{2}s^{-1}) which is given by the dynamic viscosity divided by the density

The larger the Reynolds number, the more turbulent the flow is. Between approximately 40 and 10^{3} the flow is mostly laminar. Above 10^{4} the flow is almost entirely turbulent. In between the flow is called transitional and it displays characteristics of both types of flow. The lowest sourced values for the Reynolds number of the NACA 2412 airfoil is 5x10^{4}**)** however at an average cruise speed of 226 kmh^{-1} (62.7 ms^{-1}) the Reynolds number is upwards of 5x10^{6}. The maximum our wind tunnel can manage is 12ms^{-1}, with a chord length of 0.12m and a kinematic viscosity of 1.51x10^{-5}, the Reynolds number of our model is 9.5x10^{4}. While the difference of two orders of magnitude to the cruise speed may seem extreme, in actuality this produces only miniscule differences in air stream flow. This is due to the diminishing returns of the Reynolds number. This can be observed in a computational flow simulator. Below two images are seen with the same sized airfoil, one where the airspeed is 10ms^{-1}, the other 1000ms^{-1} (the difference needed to obtain the Reynolds number of 5x10^{6}).



Status: Transient
Analysis: 2D
Wind Speed: 1.000 (m/s)
Length: 1.229 (m)
Width: 0.381 (m)
Height: 0.891 (m)
Voxel size: 0.006 (m)

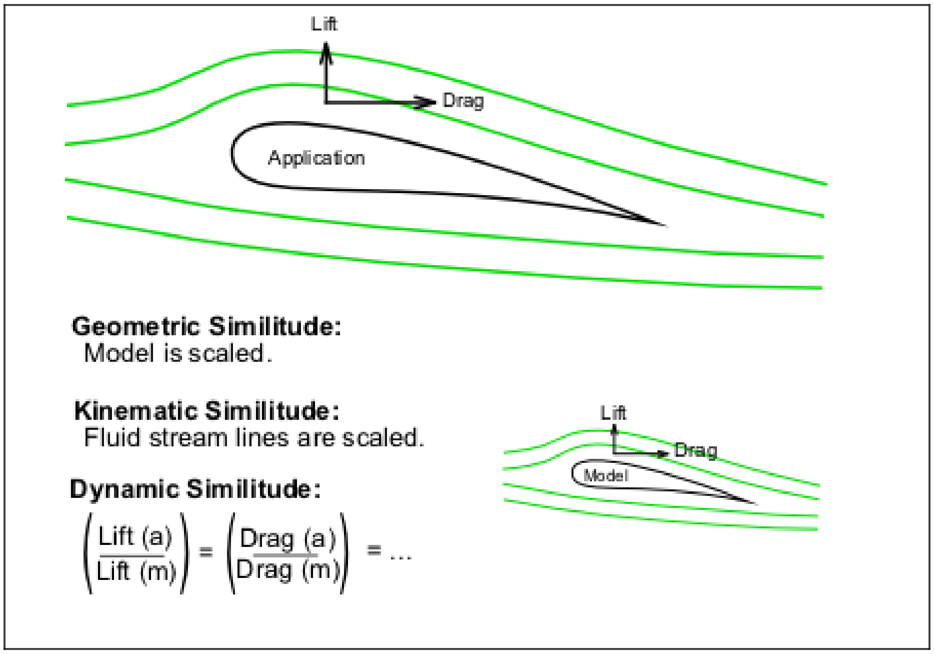
Here Autodesk Flow Design is used to simulate the flow characteristics of a 6cm chord length airfoil where the air is flowing at 1ms^{-1}. Here the Reynolds number is about 4x10^{3}. Just above and below areas of higher velocity can be seen and behind is a trail of air going at lower velocity.



Status: Transient
Analysis: 2D
Wind Speed: 100.000 (m/s)
Length: 1.229 (m)
Width: 0.381 (m)
Height: 0.891 (m)
Voxel size: 0.006 (m)

Here, Autodesk Flow Design is used to simulate the flow characteristics of a 6cm chord length airfoil where the air is flowing at 100ms^{-1}. Here the Reynolds number is about 4x10^{5}. No noticeable differences can be seen in the flow of the air between 10ms$_{-1}$ and 100ms^{-1} even though the Re are two orders of magnitude apart. That is because both are in highly turbulent flow.

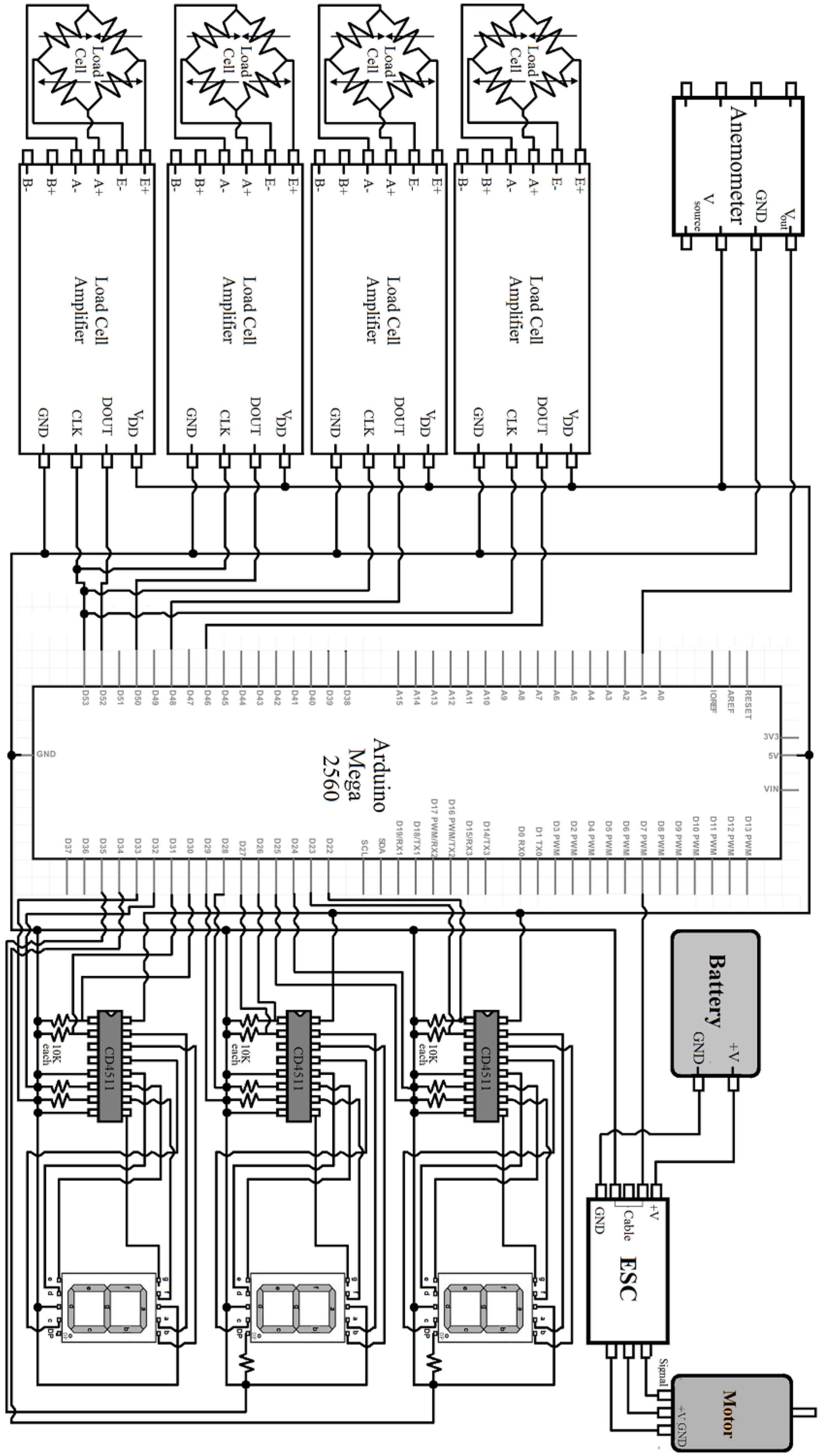
Finally, to ensure similitude dynamic similarity must be observed. This means that the forces are scaled properly. To do this drag to lift forces can be compared, or, more accurately, coefficient of lift to coefficient of drag. This will be done during testing to ensure similitude. However, dynamic similitude implies kinematic and geometric similarity and kinematic and geometric similarity imply dynamic similarity. So if one is true, the other is true. Therefore it can be assumed that this should scale correctly.


Lift
Drag
Application
Geometric Similitude:
Model is scaled.
Kinematic Similitude:
Fluid stream lines are scaled.
Dynamic Similitude:
$\left(\frac{\text{Lift (a)}}{\text{Lift (m)}}\right) = \left(\frac{\text{Drag (a)}}{\text{Drag (m)}}\right) = \ldots$
Lift
Drag
Model

As is shown in this image, there are three similarities which must be achieved in order to satisfy similitude. Geometric scaling ensures the model is scaled. Kinematic similitude ensures scaled flow characteristics. And dynamic similitude ensures equivalent force ratios.

3. Electrical Components

The following schematic diagram includes every electrical component of the wind tunnel and its wiring diagram. These components include the anemometer, load cells, load amps, motor, motor ESC, motor battery, seven segment displays, and seven segment display decoder chips. The components are powered and grounded to the Arduino Mega, with the exception of the motor battery pack used to power the motor. Due to the size of the schematic, it is placed on the following page:


Load Cell
Load Cell
Load Cell
Load Cell
B-
B+
A-
A+
E-
E+
Load Cell Amplifier
Load Cell Amplifier
Load Cell Amplifier
Load Cell Amplifier
GND
CLK
DOUT
V_{DD}
Anemometer
V_{out}
GND
V_{source}
Arduino Mega 2560
GND
3V3
5V
VIN
RESET
AREF
IOREF
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
D38
D39
D40
D41
D42
D43
D44
D45
D46
D47
D48
D49
D50
D51
D52
D53
D13 PWM
D12 PWM
D11 PWM
D10 PWM
D9 PWM
D8 PWM
D7 PWM
D6 PWM
D5 PWM
D4 PWM
D3 PWM
D2 PWM
D1 TX0
D0 RX0
D14/TX3
D15/RX3
D16 PWM/TX2
D17 PWM/RX2
D18/TX1
D19/RX1
SDA
SCL
D22
D23
D24
D25
D26
D27
D28
D29
D30
D31
D32
D33
D34
D35
D36
D37
Battery
+V
GND
CD4511
CD4511
CD4511
10K each
10K each
10K each
a
b
c
d
e
f
g
DP
ESC
+V
Cable
GND
Signal
Motor
+V
GND

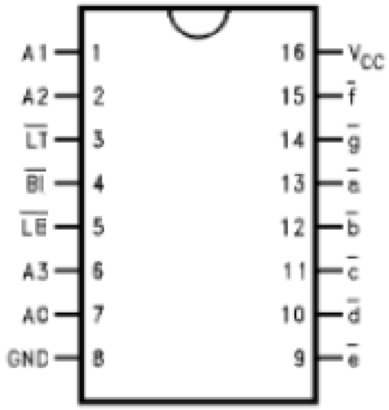
3.1 Seven Segment LED Display

The purpose of the seven segment displays is to display the wind speed. A simpler and more cost effective method to display wind speed would be to show it in the UI, but the requirements of this project state that seven segment displays driven with a decoder chip must be incorporated in the design. Three seven segment displays will be used to display wind speed. This number was chosen because the wind speed sensor (anemometer) will only be able to measure wind speed to this degree of precision. There are two types of sevens segment displays available for use: common anode and common cathode. The common pins (#3 & 8) of a common anode display are connected to +V, and the display lights up when the input pins are grounded. In a common cathode display, the common pins need to be grounded, and the display lights up when the input pins are connected to +V. Each type of seven segment display requires a specific decoder chip since they require different inputs (ground or +V). A common anode display can be wired to a 74LS47 decoder, and a common cathode display can be wired to a CD4511 decoder.

The seven segment display chosen will be a common cathode display made by the CHINA YOUNG SUN LED TECHNOLOGY CO, model No.: YSD-160AR4B-8. This display has a operating current range of 10mA to 20mA, voltage range from 1.8 to 2.2V, and an operating temp of -25 to 85°C. The operating voltage makes it mandatory to use resistors in seven segment display circuit since the power supplied from the Arduino is +5V.

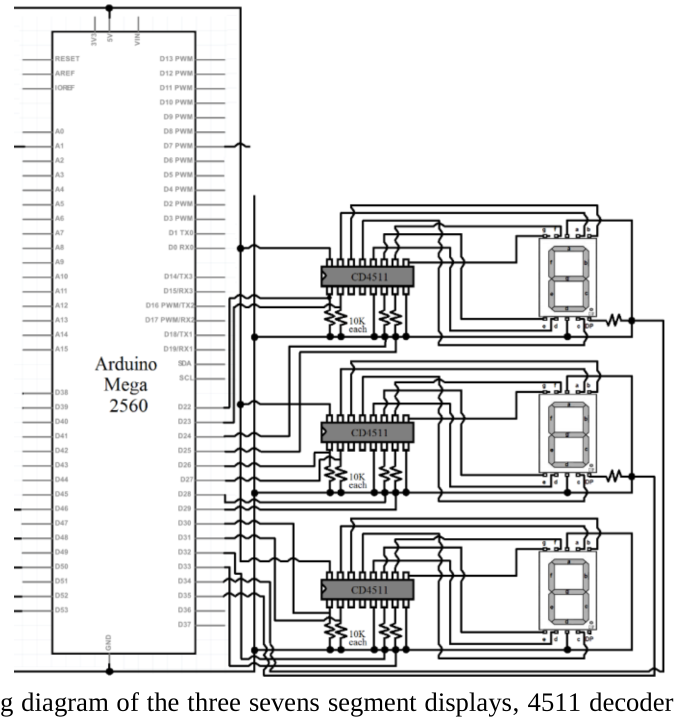
A decoder chip is used to decrease the amount of Arduino output pins required to operate the seven segment displays. Each of the eight LED’s in the display can be switched on and off by connecting power or ground to the eight corresponding input pins, depending on the type of seven segment displays used. Without decoder chips, the eight input pins on the display must be connected to eight output pins on the Arduino. With three displays, this would occupy 26 output pins on the Arduino (24 for digits and 2 for decimal points). To reduce the amount of output pins needed, each display should be wired to a decoder chip. A decoder chip designed specifically to drive seven segment displays take binary inputs via four pins from the Arduino and and *decodes* the binary inputs into output signals sent through eight pins that correspond to eight pins on the display. Using three decoder chips for the three displays, only 12 pins output pins would need to be occupied on the Arduino, plus an additional 2 pins for decimal places.

Comparing the 74LS47 and CD4511 decoder chips, they perform the task of driving seven segment displays, but they are selected based on the type of display being powered. As mentioned earlier, the 74LS47 is designed for a common anode display, and the CD4511 is designed for use with a common cathode display. Therefore, the 74LS47 has its input pins wired to ground, and output pins wired to Vcc. The CD4511 has opposite input and output wiring: input is connected to Vcc and output pins to ground. Since the Arduino output pins send a +V pulse, the CD4511 decoder will be chosen since it accepts +V inputs. If a 74LS47 decoder were used, a transistor must be placed before each input pin to convert the +V signals from the Arduino into -V inputs to the decoder. This is an extra cost and also adds unnecessary complexity to the seven segment display circuit.



Connection diagram for the CD4511 decoder. Pins 1,2,6, and 7 are input pins. These receive binary inputs that will be decoded into output signals that go to the display. The output signals are spread across eight pins (for the eight LED's in the display) through pins 9-15. Pins 3 and 4 are used to test the functionality of the connected seven segment display. Pin 3 of the chip is used as a lamp test, which turns on every digit in the display, producing an "8". Pin 4 is the blackening input, and functions to blank the display, shutting off all segments. This pin when activated overrides all the other input pins, including the lamp test. Pin 5 is a latch that allows for the output signals to be locked. Whenever the pin is activated, the chip operates in latch function and maintains its output based on the binary input it received when this pin was activated. Pin 8 is connected to ground and pin 16 is connected to +V that can be 3V-15V according to the manufacturer, Texas Instruments.

3.1.1 Seven Segment Display Design



Wiring diagram of the three sevens segment displays, 4511 decoder chips and connection to the Arduino control pins. Not visible due to resolution of this scaled down image is the specification of the resistors which is 10KΩ.

An important note for the wiring of the CD4511 decoder chip, the input pins must have pull-down resistors wired to them. A pull down resistor ensures that a logical zero will exist on the input pins when no signal from the Arduino is sent. Due to electrical noise in the circuit, the input pins of the decoder could detect a false signal even if no source signal is being sent from the Arduino. Thus, a logical zero does not exist and the decoder will drive the seven segment display to display the wrong digits. For the CD4511, a 10KΩ resistor will be used to connect each input pin to ground. While other pins and LED's of the display are active, the pull down resistor will prevail and bring the electrical noise down through ground. When a legitimate logic signal from the Arduino is sent, it will override the pull down resistor and make sure a logical input is detected by the decoder.

3.1.2 Testing Procedure:

Before any circuits are built, it is a good idea to test all components individually and then attach them all together on a breadboard. Power can be drawn from the Arduino (+5V) and connected to their respective rails on the breadboard. First, the seven segment displays should be tested individually. To do this, put two resistors in series between the ground pin on the Arduino and the ground rail on the breadboard. Then connect pins 3 and 8 of each display to ground. To test each segment in the display, connect power (+V) to pins 1,2,4,5,6,7,9, and 10 one at a time. This will light up each segment of the display individually. In a perfectly working display, connecting +V to the pins will light up only one segment at a time. The following is a helpful testing chart to keep track of which pins work:

	g	f	a	b	c	d	e	DP
Display A								
Display B								
Display C								

Once each display has been tested, the CD4511 decoder chip should be wired. Before inserting the decoder chip onto the breadboard, disconnect ground and power or shut off the Arduino. Connect the output pins on the decoder to the corresponding pin on pin on the display. The pin letters of the chip match with the letters on the display. Then wire the 10KΩ resistors from the input pins to ground. Next, connect the input pins to the Arduino output pins #22-33, and connect ground and power to pins 8 and 16 respectively. Finally, wire two 10KΩ push down resistors on pin 5 (decimal pin) of the hundreds and tens digit displays to the ground rail of the breadboard. Then connect the hundreds digit and tens digit decimal point to Arduino output pins #34 and 35 respectively.

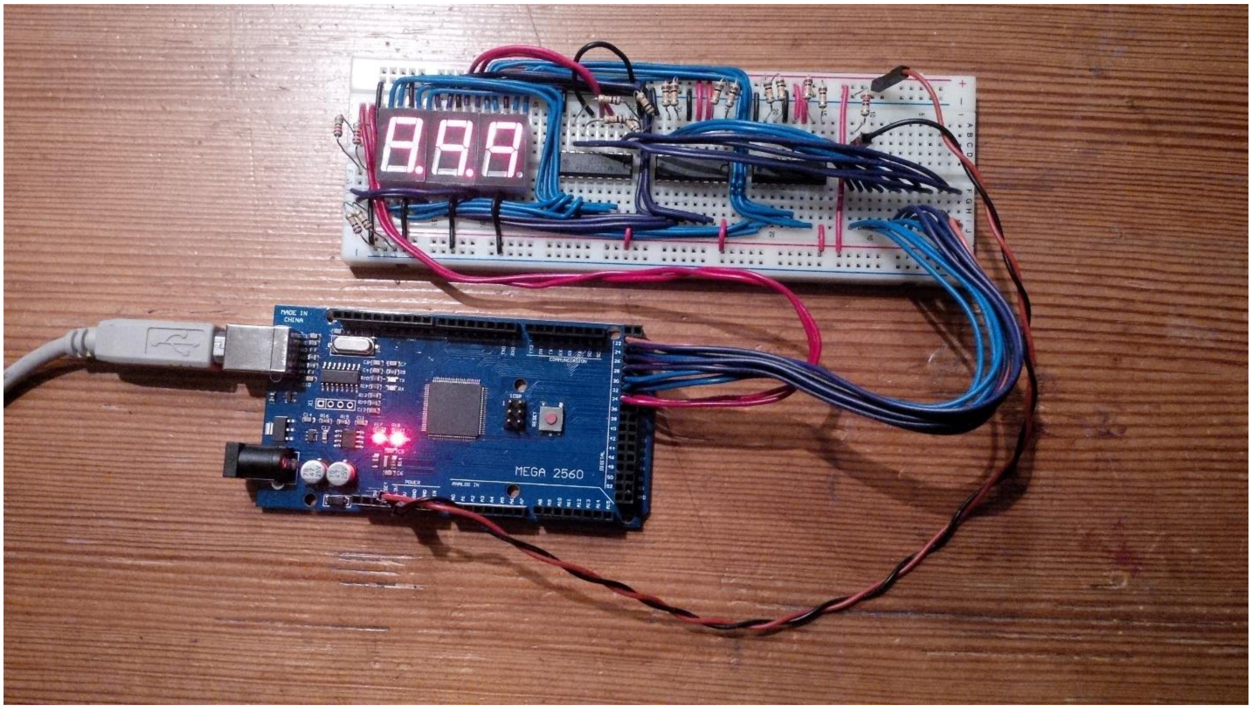
When the circuit is powered on with no input coming from the Arduino, the seven segments should display three zeroes with no decimal. If six segments are lit but a zero is not formed, then it is likely that wires have been switched between the chip and the display. Double check that the chip output pin is wired to the corresponding seven segment display pin by letter. If no segments light up, immediately unplug the Arduino in case of a short circuit. This could permanently damage the seven segments, decoder chip, or the Arduino. If the displays flicker, this indicates that the push down resistors are not properly wired. Electrical noise in the decoder prevents logical zeros even when no input is sent to the decoders, causing the chip to send output signals to the display that cause segments to flicker on and off. To fully test the functionality of the decoder chips, run the following Arduino code:

```
//Automatic Testing of Seven Segment Display
//Loops through all numbers and decimal points
int pins[3][4] = {
      {22, 23, 24, 25}, //Pins for MSD
      {26, 27, 28, 29},
      {30, 31, 32, 33} //Pins for LSD
};
int decPins[] = {34, 35}; //Pins for decimal point

void outDigit(int pin, int digit){ //Outputs single digit as 4 bit
digit
  for(int i=3;i>=0;--i){
    digitalWrite(pins[pin][i], digit%2);
    digit /= 2;
  }
}

void setup(){
  for(int i=22;i<=35;++i) pinMode(i, OUTPUT); //Set pins to output
}

void loop(){
  for(int i=0;i<=9;++i){
    outDigit(0,i);
    outDigit(1,i);
    outDigit(2,i);
    digitalWrite(decPins[0], i%2);
    digitalWrite(decPins[0], i%2);
    delay(1000);
  }
}
```

When this program is run on the Arduino, the numbers on the three displays should be identical and ascending from 1 to 9. The numbers should also change at the same time once every second. A decimal point will also be displayed for all odd numbers on the hundreds and tens digit. Here is an example of the output:

```
0.0.0
1.1.1
2.2.2
3.3.3
4.4.4
5.5.5
6.6.6
7.7.7
8.8.8
9.9.9
 LOOP
```

If the segments light up and change but some of the digits do not form actual numbers, that indicates the input pins of the chip are not correctly wired to the Arduino output pins that the testing program is using. Refer back to the circuit diagram to double check the wiring.



Sample testing breadboard for three seven segment displays, three decoder chips, and the Arduino testing program.

3.1.3 Construction

In the final project, the seven segment displays and decoder chips will be placed in a module that on the upper part of the venturi section of the tunnel. This display module will contain the three displays, three decoder chips, and the Arduino. A 6 ft USB cable running from the display module to the computer will connect the Arduino to the computer. Additional cords will exit the module to connect the force sensors, anemometer, and motor ESC to the Arduino control pins. The shape of the module is a rectangular box with a display panel of dimensions 6cm x 21cm and a depth of 3cm. Three seven segment displays will be mounted on the display panel which occupies the upper face of the box. This panel will flip open to allow access to the decoder chips, Arduino, and wiring underneath. The construction of the box will be made out of leftover particle board. No ventilation is needed as the Arduino, decoder chips, and seven segment displays do not create enough heat to be a concern.

3.1.4 Assembly Procedure

Building and soldering of the final circuit should take place in the same order as the testing, with routine checks to isolate cold joints or other circuit defects. The circuit should be wired and soldered for one seven segment display and decoder at a time. When soldering the seven segment display to the CD4511 chip, small amounts of solder should be used to prevent soldering two pins at once. Since the pins of the seven segment display and 4511 decoder are spaced next to each other, it is a good idea to place a folded sheet of aluminium foil or wax paper four layers thick between pins when soldering. This will prevent two pins from being soldered together by accident. It is a good idea to solder a 16 pin IC socket to the seven segment display instead of the CD4511 decoder. In case the decoder chip fails, it will be easy to take it out of the IC socket and replace it with a new chip. The first test should verify that soldering was done correctly on the seven segment display. Ground pins 3 and 8 and individually apply 2V to pins 1,2,4,5,6,7,9, and 10. Each pin should light up the corresponding segment. Once the IC socket has been soldered, test the connection by applying 2V to the pins connected to the seven segment display. Next solder on power, ground, and input wires to the IC socket. To test these connections, place the CD4511 decoder in the socket and connect the socket to the appropriate pins on the Arduino. Do not forget to ground/power as appropriate for pins 3,4 and 5 of the decoder. Then solder the pull down resistors between pins 1,2,6, and 7 and ground. When the circuit is powered, the displays should show three zeros. If not, follow the troubleshooting procedures found in the "Testing Procedure". If the soldering and wiring checks up, repeat this again for displays #1 and 2, and their corresponding decoder.

Once all three seven segment displays and decoder chips have been wired and checked, connect the pull down resistors and input wires to the Arduino in the same fashion used in the "Testing Procedure". To check that this was done correctly, run the testing program again to see if the correct outputs are displayed. Once the entire circuit is soldered, connect it to the Arduino and place neatly in the display module. Then mount the three seven segments on the display panel and place the module on its appropriate location above the tunnel.

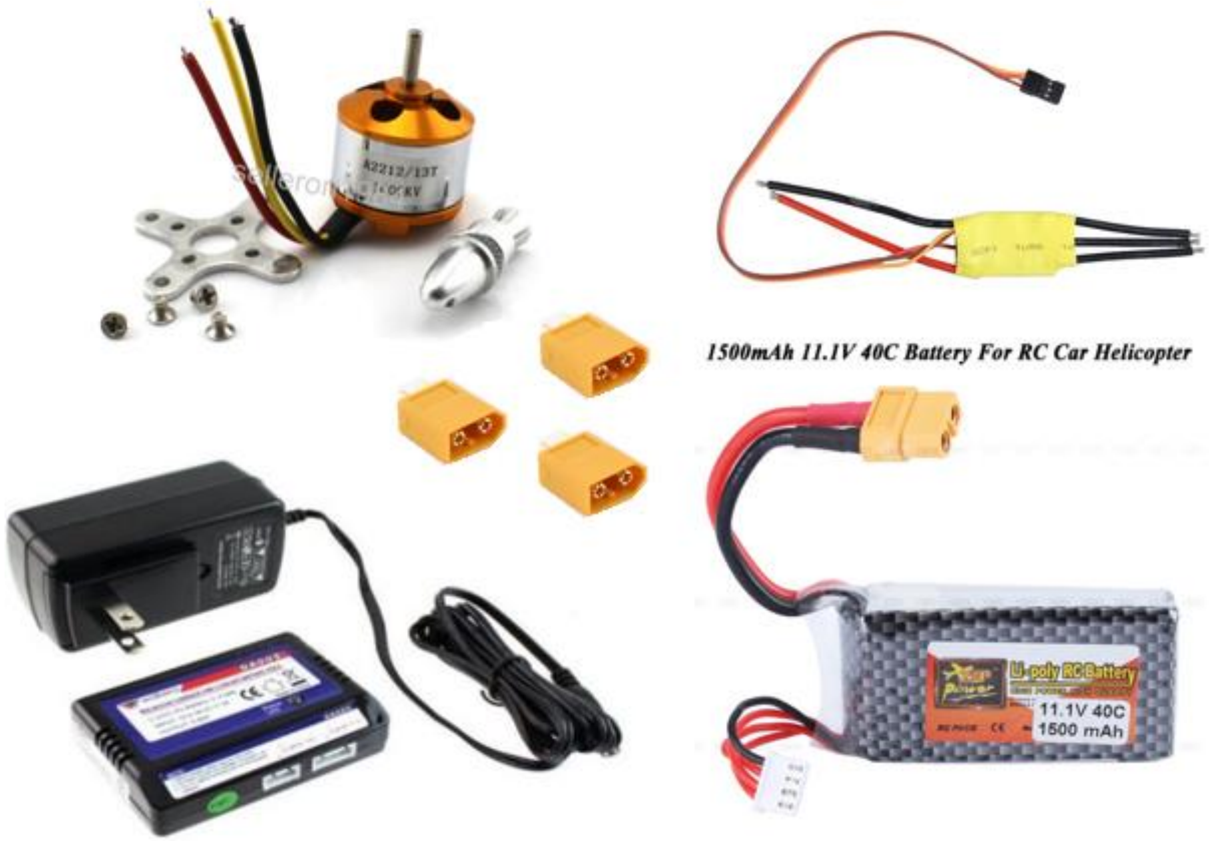
3.2 Wind Generation Fan

The wind tunnel must use a fan to generate airflow around the airfoil being tested. To achieve uniform airflow over the airfoil, the fan will have to be placed at the back of the wind tunnel. This would 'suck' air through the tunnel and across the airfoil, a method that produces less turbulence than placing a fan at the front of the tunnel to 'push' air through. Due to the scaling choices made with the airfoil, a wind speed of 8-11 ms^{-1} will be required in the wind tunnel.

To generate enough air flow in the tunnel, a 9" (20.32) cm prop will be mounted on an RC aircraft motor. This will provide sufficient wind speed in the tunnel and is a cost effective motor + propeller combination, totalling $10. Alternatives to using RC aircraft parts are to use computer case fans. While some computer case fans can be larger than the aircraft part combination being used for this project, they are designed with quietness in mind. These fans spin at a slower RPM, and their CFM is too low for this project. CFM is the amount of air in cubic feet that the fan pushes each minute. Due to the high RPM of the motor, the propeller will generate higher turbulence. By placing the fan behind the airfoil at the back of the wind tunnel, turbulence is reduced at the airfoil, as it would have to travel upstream. This will ensure that the airfoil around the airfoil is kept uniform.

An overview of the parts needed for the entire fan component of the tunnel are:

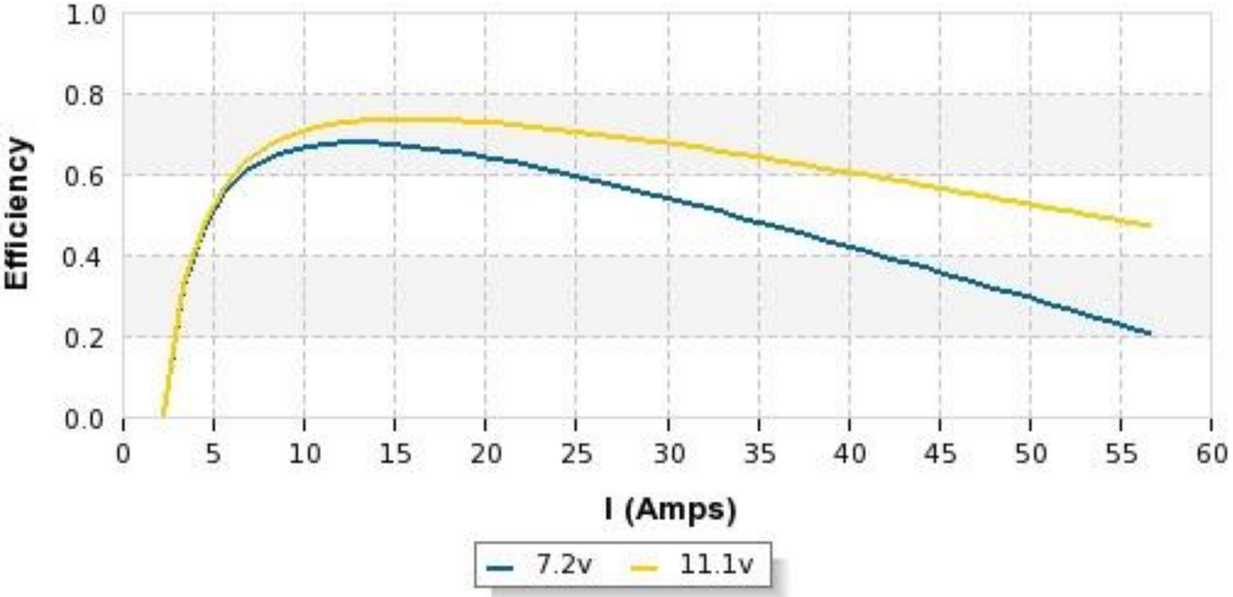
- A2212 1000KV brushless motor
- Motor mounting cross and screws
- Set screw with nose cone
- 9x4 propeller
- 40A ESC
- 1500 mAh, 3S, 11.1 V LiPo Battery for RC vehicles.
- Walkera QR X350 Battery charger and balancer
- Three (3) XT60 Plugs



This is an image of all the components needed to make the fan assembly. Starting in the top left corner and moving clockwise is the A2212 motor with set screw, mounting plate, and mounting screws. Next is the 40A ESC. The three black wires connect to the motor. The short black and red wires on the left connect to the battery, and the thin wires connect to the Arduino control pins. On the bottom right is the 1500 mAh, 3 cell, 11.1 V LiPo battery used to power the motor. The cords extending upward connect to the ESC, and the cords extending down are used for charging. In the last corner is the battery charger. The battery will plug into the rightmost white port, and the charger will be plugged into an AC wall socket. At the centre are three XT60 plugs that connect the ESC to the battery, and the motor to the ESC. This plug allows the all components to be separated from the ESC. The battery must be removed from the circuit while charging, and the motor may be removed to fit on a new propeller, swap in a cooler motor, or do structural checkouts up the tunnel.

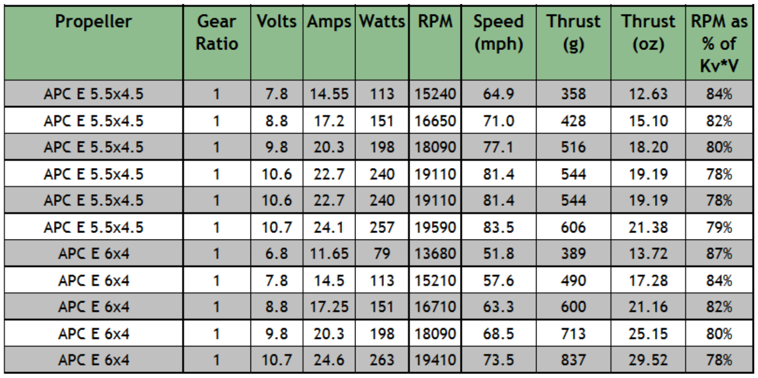
For logistical purposes, it is important to mention that the set screw, nose cone, metal mounts, and mounting screws are included as a package with the motor. This package can be found for $3.99 CAD on [2], but realize that it takes over a month to ship to Canada, the U.S, or Europe. The same is true when ordering the 40A ESC and LiPo battery from cheaper online vendors.

The specifications of the A2212 motor show that it generates 2200 RPM/V, has a current of 1.4 A @ 10V, and has a maximum efficiency of 75%. This is while the motor operates under no load. With a 10V power supply and 2.25A current, the motor is specified to spin at 22650 RPM with no load. The motor is able to run with voltages ranging from 7.2 to 11.1 V. Specifications were taken from [14]. It is recommended that the motor drive a 7-8” prop, although there were examples online of the motor driving a 10” prop, as seen in [3].



The graph above is an efficiency graph for the motor based on voltage and current supplied to the motor. This testing data was taken from [27], when there was no load on the motor. The blue line plots data for the motor at it's lowest voltage rating, 7.2V. The yellow line plots data for the motor running at its highest voltage rating, 11.1V. Maximum motor efficiency is achieved when the current is near 15A. The test data shows that the motor is operational up to around 50A. This current would be necessary to power to motor if a large propeller (10" +) were used.

Mounted on the motor will be a 9"x4 propeller. The first number (9) specifies the diameter of the propeller, and the last number (4) specifies the pitch of the blades. To get some data on the type of performance expected from this combination with the motor, the following table was consulted:

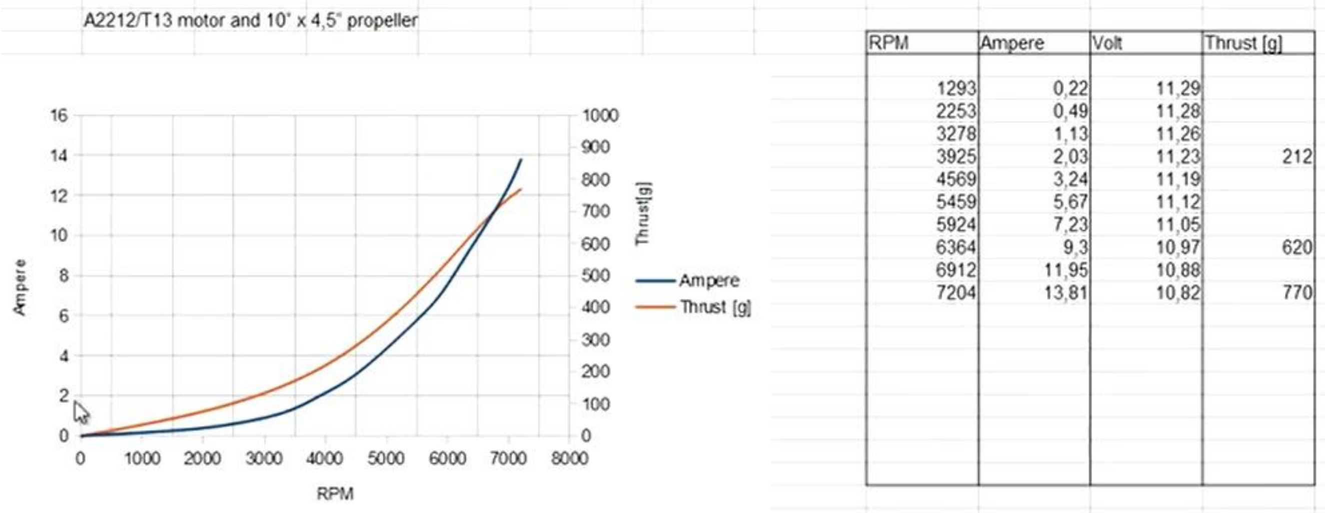


Propeller	Gear Ratio	Volts	Amps	Watts	RPM	Speed (mph)	Thrust (g)	Thrust (oz)	RPM as % of Kv*V
APC E 5.5x4.5	1	7.8	14.55	113	15240	64.9	358	12.63	84%
APC E 5.5x4.5	1	8.8	17.2	151	16650	71.0	428	15.10	82%
APC E 5.5x4.5	1	9.8	20.3	198	18090	77.1	516	18.20	80%
APC E 5.5x4.5	1	10.6	22.7	240	19110	81.4	544	19.19	78%
APC E 5.5x4.5	1	10.6	22.7	240	19110	81.4	544	19.19	78%
APC E 5.5x4.5	1	10.7	24.1	257	19590	83.5	606	21.38	79%
APC E 6x4	1	6.8	11.65	79	13680	51.8	389	13.72	87%
APC E 6x4	1	7.8	14.5	113	15210	57.6	490	17.28	84%
APC E 6x4	1	8.8	17.25	151	16710	63.3	600	21.16	82%
APC E 6x4	1	9.8	20.3	198	18090	68.5	713	25.15	80%
APC E 6x4	1	10.7	24.6	263	19410	73.5	837	29.52	78%

This table shows testing data for the A2212 motor running various propellers, ranging from 5.5 to 6". Voltage, current, power, RPM, speed, thrust, and Kv*V were recorded.

The table shows that up to 837g of thrust can be produced with a 6x4 prop running at 19410 RPM. To do this, 10.7V and 24.6A were sent to the motor. Using a larger prop, more air could be pushed (higher CFM), but the propeller would spin slower due to its extra size.

Source [3] is a video test of the A2212 motor with a 10x4 prop mounted on. Test data on the RPM, amperage, and thrust were taken and are shown on the following graph:



RPM	Ampere	Volt	Thrust [g]
1293	0,22	11,29	
2253	0,49	11,28	
3278	1,13	11,26	
3925	2,03	11,23	212
4569	3,24	11,19	
5459	5,67	11,12	
5924	7,23	11,05	
6364	9,3	10,97	620
6912	11,95	10,88	
7204	13,81	10,82	770

The relation between current of the motor and thrust produced is seen on the graph to the left. With a 9x4 propeller, 800g of thrust can be expected at maximum power. Even though the correct unit for thrust is the Newton, grams were measured due to their practicality and the equipment used by the testers.

3.2.1. Powering the Fan Motor

The requirement for the wind tunnel limits any use of potential greater than 12V (without using a CSA approved controller). To power the fan, a battery pack that works with the motor must be used. The motor specifications call for a three cell, 11.1V LiPo battery. The connection between the batteries and fan will have two switches: One in the UI and a physical switch between the battery and ESC. The double redundancy here is a safeguard to ensure that the high RPM fan does not start operation unexpectedly. This will avoid possible injury to those around the wind tunnel and destruction of the wind tunnel itself.

Due to the nature of batteries and their capacities, the potential supplied by the batteries will diminish over time. This would limit the maximum wind speed that could be attained in the tunnel based on the diminishing capacitance in the batteries. At lower speeds, the UI software will receive RPM signals from the motor and wind speed measurements from the anemometer. Then potential through the fan motor could be adjusted/boosted through the Arduino’s PWM. This solution would only work for situations when maximum wind speed (motor running at 11.1V) is not needed.

If there are sufficient funds leftover near the end of the project, it will be possible to implement a power supply monitoring system. This would record the potential and current of the batteries and display these properties in the UI.

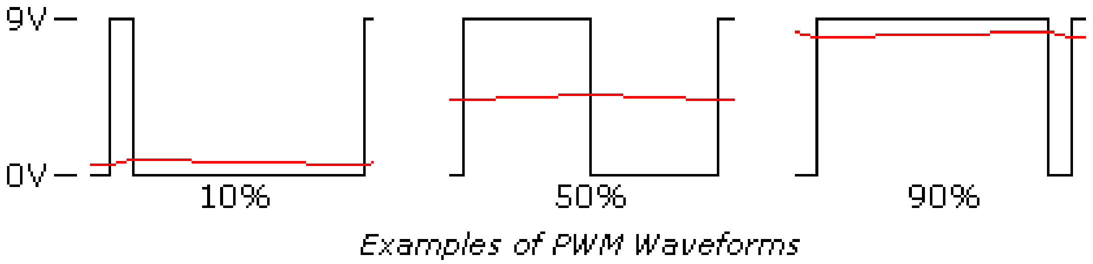
The LiPo battery also has specific charging requirements. A certain type of charger needs to be used that has the right port/interface between the battery and charging pins. For the three cell battery, a port that accommodates the 4 pin battery charging cable on the battery is required. The charger must also have a in-built balancing function. This ensures that all cells of the battery are balanced, preventing sudden voltage drop offs if one cell dies as the motor draws power from the battery. The charger used is the Walkera QR X350 which is designed specifically for charging and balancing 3 cell 11.1V batteries.

3.2.2 Controlling the Fan

The maximum potential that can be used on any component in this project is 12V. Therefore, whatever wind speed is generated by the fan and motor running at 12V will represent the maximum airspeed the wing flies at in real life: ~280 kmh-1. In the wing performance analysis, it is a good idea to put the wing under a run profile: meaning a series of force and drag measurements are made for a range of wind speeds. This means that the fan speed will have to change during the operation of the wind tunnel's run profile.

The easiest way to control wind speed in the tunnel is to adjust the RPM of the fan. This can be achieved by under volting the fan motor to get it to spin slower. It would be preferable if this method was controlled by the microcontroller (Arduino/computer). This would allow for precise control of the fan, and perhaps user input to control the fan. A desired wind speed could be inputted into the UI, and the microcontroller would adjust the fan to reach the specified wind speed. It is difficult to have a hard-wired potentiometer or similar component that could vary the potential going to the fan motor while being controlled by the Arduino. Instead, another feature will have to be used.

The most viable option on the Arduino is the ability to use analog outputs to switch output power on/off rapidly. This method is formally known as Pulse Width Modulation (PWM), where the power source is switched ON and OFF so rapidly that the electrical load is neither on nor off, but somewhere in between. Modulation of these pulses or switches will affect the total power supplied to the load. The proportion of 'on' time to the regular interval or 'period' of time is described by the term *"duty cycle"*. A low duty cycle corresponds to low power because the power is off for most of the time. Duty cycle is expressed in percent, with 100% being fully on. The Arduino has the capability for us to choose between a 0% to 100% duty cycle, thus providing full control of the fan and wind speed in the tunnel



Examples of duty cycles. The average power that the load (motor) experiences is represented by the red line.

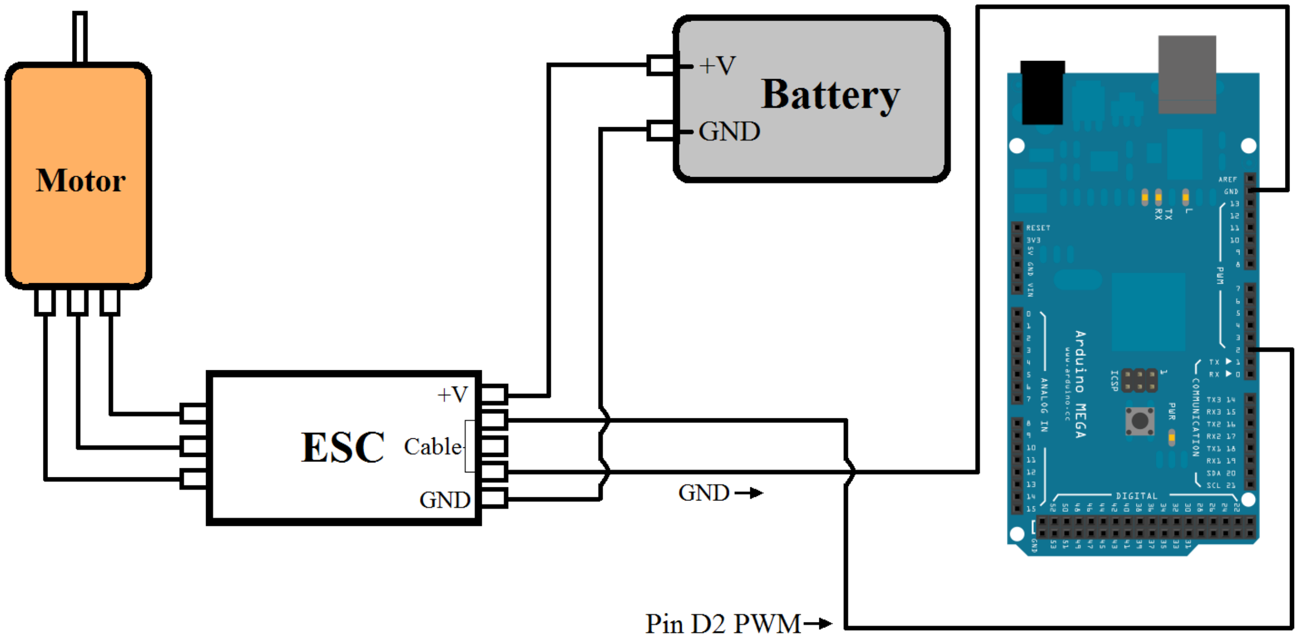
Using PWM to control the fan is not the only practical option. It is possible to use a potentiometer (variable resistor), but there are drawbacks to this. Since a motor is a varying electrical load (especially at start up), the potentiometer may not provide sufficient power to get the fan blades turning, given the extra weight and high rotational inertia of the custom fan design used. With PWM, extra power can be supplied to the motor if needed to help the motor get the fan blades spinning from rest. If a potentiometer were to be used, the increasing power requirements of the motor at start up would exceed the power rating on the potentiometer, causing it to heat up and fail permanently. Lastly, assuming the potentiometer would be sufficient in controlling the fan, the resistors inside would waste excess power as heat. This wasted energy is of specific concern given that the fan motor is to be powered by batteries.

PWM will be implemented along with an electronic speed controller, detailed in the following section.

3.2.3 Using an Electronic Speed Controller (ESC):

For the wind tunnel, a more powerful motor will be used which will allow for a greater range of wind speed: a brushless DC (BLDC) motor. The difference between a brushless motor and a brushed motor like one explained above is, as the name suggests, brushes. Brushes are connected to opposite poles of the power source, and physically contact the commutator, a rotating split-ring device that periodically changes the direction of the current. The brushes charge the commutator in the opposing charge of the permanent magnet in the motor, which causes the rotation. On the other hand, a brushless motor are designed so no brushes are required. Instead, its permanent magnets are located along the perimeter of the rotor.

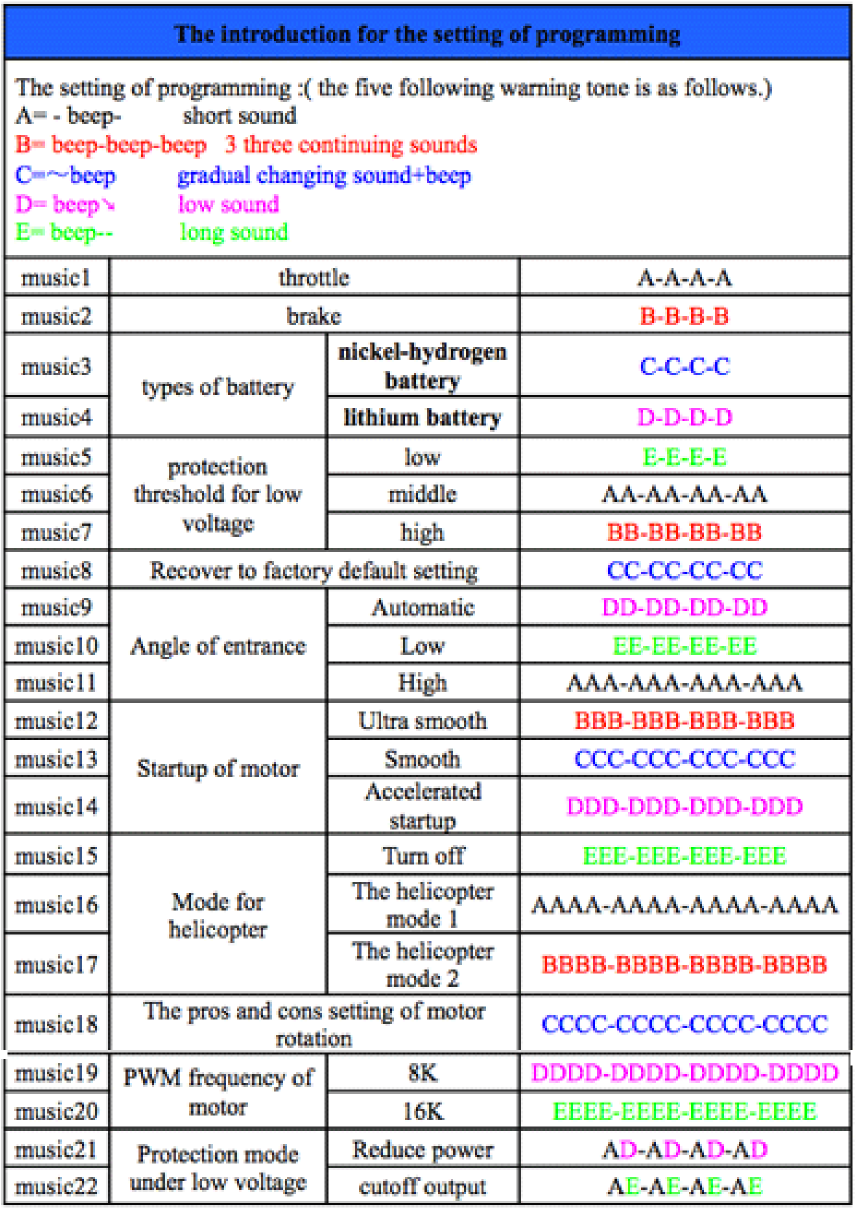
The benefit to using a BLDC motor is that there is less maintenance required as there are no connections via brushes. The motor is also quieter and smoother, dissipates heat faster, and has a higher speed range. In addition, the lack of brushes makes the motor more efficient, as there is a potential drop across the brushes. However, in contrast to the simple brushed motors which can be controlled using only two wires from the motor, the more complicated BLDC motor requires an electronic speed controller (ESC) to accurately create the pulses needed to run the motor. The BLDC motor takes three inputs which control each section of the motor. These wires are connected to the ESC. Supplying the power is a battery connected to the ESC using two leads. Finally, a servo connection is wired to the Arduino, specifically any PWM pin and ground.



A diagram of how the ESC connects with the Arduino and the motor. The ESC acts like an adapter for the Arduino to the BLDC motor so it can be controlled with a PWM signal like a brushed motor. The ESC also connects to the battery, which is a 3 cell LiPo battery specifically designed for RC aircraft engines.

An ESC is synonymous to the brushes in a brushed motor. It takes a DC input and converts it into a rotating electric current for the BLDC motor. The frequency of the motor can be adjusted using the PWM input to the ESC, which is supplied by the Arduino. The range of the ESC can be armed when it is first turned on. After supplying a high PWM signal to the ESC control pin, the ESC must be connected. It will then emit a series of tones of different frequency to indicate the upper limit has been set. Following immediately after is to send the low throttle signal, and the ESC will respond with another series of tones. The lower limit has been set and the ESC is armed.

Similar to how an Arduino can be programmed, an ESC can be similarly programmed too. Sending a high signal for five seconds will trigger a special tone, activating the program mode. In program mode, the ESC will cycle several unique series of tones, each indicating an setting to select. A low PWM signal will select the current setting and will cycle through the tones of the options for that setting. To finish the setting change, a high signal will select the current option and the ESC will return from program mode.



The introduction for the setting of programming

The setting of programming :(the five following warning tone is as follows.)
A= - beep- short sound
B= beep-beep-beep 3 three continuing sounds
C=~beep gradual changing sound+beep
D= beep↘ low sound
E= beep-- long sound

music1	throttle		A-A-A-A
music2	brake		B-B-B-B
music3	types of battery	**nickel-hydrogen battery**	C-C-C-C
music4		**lithium battery**	D-D-D-D
music5	protection threshold for low voltage	low	E-E-E-E
music6		middle	AA-AA-AA-AA
music7		high	BB-BB-BB-BB
music8	Recover to factory default setting		CC-CC-CC-CC
music9	Angle of entrance	Automatic	DD-DD-DD-DD
music10		Low	EE-EE-EE-EE
music11		High	AAA-AAA-AAA-AAA
music12	Startup of motor	Ultra smooth	BBB-BBB-BBB-BBB
music13		Smooth	CCC-CCC-CCC-CCC
music14		Accelerated startup	DDD-DDD-DDD-DDD
music15	Mode for helicopter	Turn off	EEE-EEE-EEE-EEE
music16		The helicopter mode 1	AAAA-AAAA-AAAA-AAAA
music17		The helicopter mode 2	BBBB-BBBB-BBBB-BBBB
music18	The pros and cons setting of motor rotation		CCCC-CCCC-CCCC-CCCC
music19	PWM frequency of motor	8K	DDDD-DDDD-DDDD-DDDD
music20		16K	EEEE-EEEE-EEEE-EEEE
music21	Protection mode under low voltage	Reduce power	AD-AD-AD-AD
music22		cutoff output	AE-AE-AE-AE

This table indicates the possible settings and options that are available when a ESC is in program mode. Five different tones can be produced, such as a single short tone, and three quick consecutive tones. Here, 22 settings are available to choose in the beginning. For several of these settings, there are multiple options to choose from, such as which type of battery is to be used with the ESC.

3.2.4 Testing of the Charger/Balancer + Battery

Use the charger to fully charge the battery. When this is done, remove the battery and measure the voltage difference across the power and ground pins. This should be 11.0-11.1V, indicating a functional battery and charger. Given the capacity of the battery and type of charger, it is typical for a full charge to take 30 mins.

3.2.5 Testing of the ESC

Wire the ESC as if a motor were to be used. Solder the XT60 plug to the battery leads of the ESC. Once this is done, the ESC and battery will be able to connect. This connection has to be removable with the plugs to enable the battery’s removal and charging. Instead of having the motor connected, use a multimeter to check voltage and current as the Arduino cycles PWM from low to high. The ESC control pin should be connected to output pin #2 of the Arduino. The ESC should respond with varying voltage ranging from 7.2 - 11.1 V, and current ranging from 1.5A to 20A. The Arduino code for this test and the motor test is below:

```
//Testing PWM control to motor via ESC

int pin = 2; //ESC Control pin

void setup(){
  pinMode(pin, OUTPUT); //Set pin to output
}

void loop(){
  for(int i=0;i<256;++i){ //Gradually increase to max RPM
    analogWrite(pin, i);
    delay(200);
  }
  for(int i=255;i>=0;--i){ //Gradually decrease to stop
    analogWrite(pin, i);
    delay(200);
  }
}
```

3.2.6 Testing of the Motor

Once the ESC, battery, and charger are approved for use, the motor can be tested. Mount the motor to a clamped down piece of wood to prevent it from moving during testing. The motor axle should be pointing straight up. Screw the mounting plate onto the motor and wood using the M3 screws that came with the motor. Connect the motor to the ESC and run the motor testing program (refer to ESC testing). Once again, connect the signal wire of the ESC to control pin #2 of the Arduino. This program will cycle between low and high voltage, testing the motor throughout its operating range. A video of what the motor’s performance should be like can be seen at [7]. If the motor performs in the same way it did for the video, proceed to test with the propeller. Mount the propeller and set screws on the axle and tighten until the propeller is firmly mounted on. Double check this or else the blade may fall off during testing. For this test, the motor should be mounted horizontal, meaning the propeller is oriented as if flying on an RC plane. Eye protection is mandatory for this test, and people should be a safe distance away from the motor-propeller assembly. Run the motor testing program again for one cycle and monitor the motor’s performance. After the cycle is complete and the propeller has come to rest, check the temperature of the motor. The outer case should not be warmer than 50°C, even though it is certified to reach 65°C [14]. To test motor performance, mount the motor and propeller assembly vertically so the propeller faces upwards. Place the assembly on a scale of proper sensitivity and zero it. Then run the motor and record the weight measured by the scale - this is the thrust produced by the motor - propeller combination. As seen from testing data, the thrust should range from 200g to about 830g.

3.2.7 Assembly Instructions for the Fan

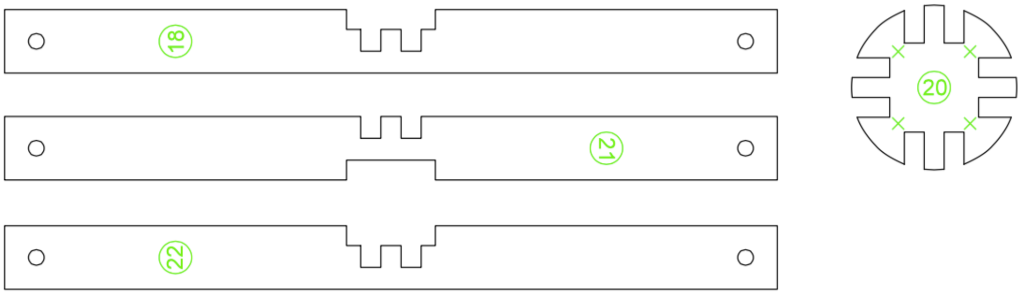
After testing has been done, the motor-propeller assembly may commence. First, secure the motor-fan assembly to the four support struts at the back of the wind tunnel. Do this by screwing the cross plate onto the struts. If there are any structural defects on the struts, do not mount the motor. The struts are rated to withstand 1.1kg of thrust from the motor. Due to the fan's location at the back, a structural failure of the struts during operation will likely result in the detachment of a 5000+ RPM motor and propeller. This is a very scary thing. As it flies off the back of the tunnel, it will cause serious damage to the surrounding area. To reduce damage if such an event were to occur, the fan is placed inside the tunnel, blowing air out. In a detachment scenario, the thrust of the fan should direct the propeller forwards, towards the narrow part of the wind tunnel. This should stop the rogue fan, but the wind tunnel may be destroyed at the cost of keeping the environment outside safe.

Once the motor and propeller has been securely mounted, components can be placed in the fan module. This module is located at the bottom rear section of the wind tunnel. The box shaped module should have dimensions 3.1x3.28x8.62 cm. It houses the ESC and removable battery. The battery is removable so that fresh batteries can be swapped, and also because batteries must be charged separately with the charger/balancer outside the wind tunnel. To further modularize the design, the motor mounting and ESC will be removable after the removal of some mounting screws.

Due to the nature of the high RPM motor, a few safety designs must be added. A safety grill/net should be used on both sides of the fan. Even though this will decrease the CFM and wind speed in the tunnel, it is necessary to include. Leaving the fan unshielded is negligence to safety for the clients. An inner net is necessary to prevent damage to the fan in a freak incident where the airfoil breaks apart/off and is sucked into the fan. An impact of the airfoil with a high RPM motor would destroy the airfoil, propeller, motor and possible wind tunnel structure. A net would catch the airfoil and prevent such a disaster. Another safety feature to be programed in the UI is a master off button. This would begin a motor power down sequence (through the Arduino) to quickly and safely shut off the motor in case something is not right. This could be a loose airfoil, structural failure, propeller wobble, etc. The point of this is to prevent a failure from progressing into something catastrophic.

3.2.8 Mounting the Motor

A basic element of every wind tunnel is the method by which air is propelled through the testing stage. In the case of this tunnel, a propeller attached to an exceedingly powerful motor was used to pull air through the tunnel. As such, a sturdy mount was required to hold this motor in place. It was also required that the mount be removable to facilitate the installation, testing, repair, and possibly replacement of the motor. As such, a mount was designed consisting of four interlocking bars that cross the end of the wind tunnel and bolt on to either side of each of four of the long struts that run the length of the tunnel. At the intersection of the four pieces that make up the motor mount, a round circular piece is fit into place. The entire assembly is glue together and bolted to the motor before being bolted to the back of the wind tunnel.



This drawing shows the mechanism of the motor mount. Parts 19 and 21 are inserted into parts 18 and 22, and part 20 is inserted on top. This creates a strong base on which to mount the extremely powerful motor.

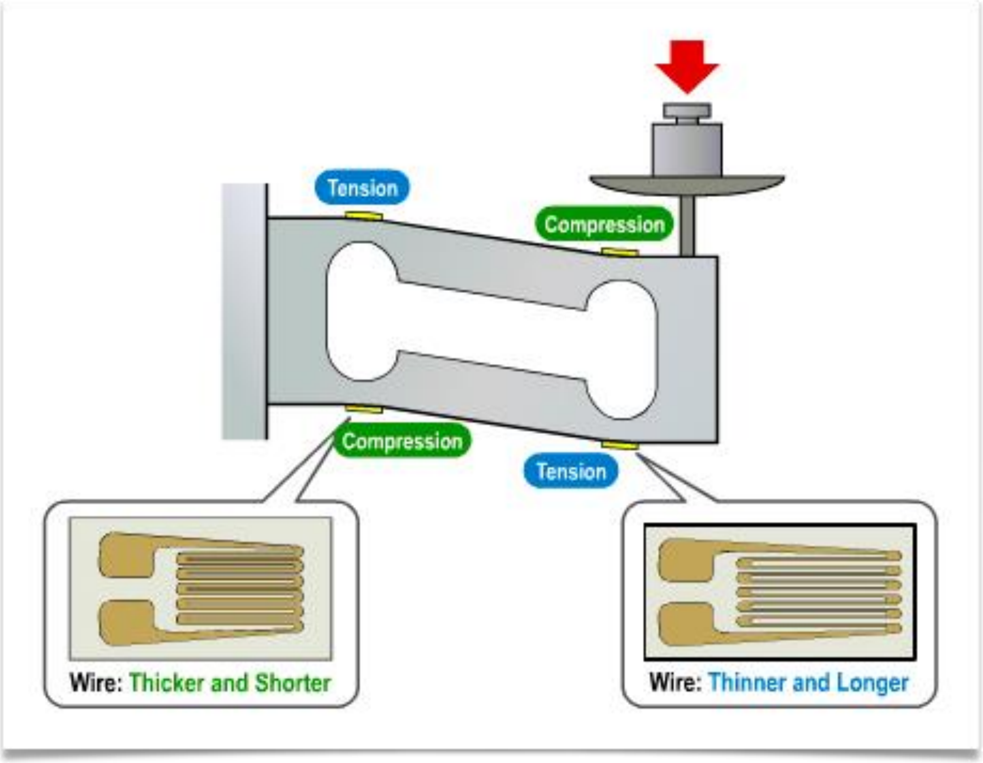
3.3 Force sensors

3.3.1 Load Cells

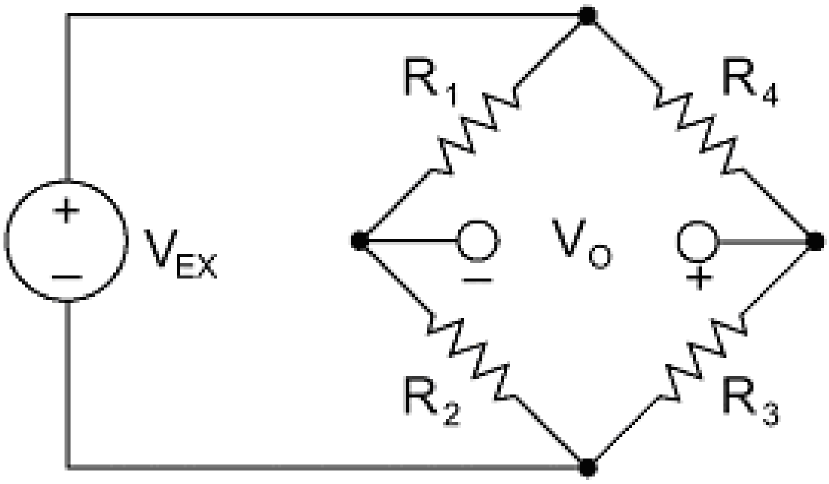
An integral part of the design of our wind tunnel was creating a mechanism for measuring the forces of lift and drag acting on the wing. Several possibilities to this effect were considered, including Force Sensitive Resistors, strain gauges, and load cells. FSR's, while simple, operate with a resolution of between 10 and 50 grams [1], quite insufficient for our applications. Strain gauges on their own do not measure force, and must be attached to a load cell whose strain is proportional to the force applied. Preliminary research revealed that the most sensitive of these Load Cell-Strain Gauge combinations (henceforth referred to simply as "Load Cells") are able to resolve forces on the order of tenths of millinewtons, so this was chosen as the method of force measurement.

3.3.1.1 Operation of Load Cells

Single-point parallelogram-based load cells are the most cost effective of the load cells sensitive enough for our applications. They operate using a rectangular metal structure, with four strain gauges placed in specific points (see figure below). As load is applied to one end of the load cell, the structure bends at the thin points in the metal and either stretches or compresses each of the four strain gauges. The figure below also shows the operation of the strain gauges. As the unit is stretched, the many wires inside become slightly thinner, increasing resistance. The converse is true for compression, which decreases the resistance. These four resistors are set up in a Wheatstone Bridge formation, so that the two compressing and two stretching strain gauges are opposite each other. Thus, when a load is applied to the load cell, the voltage Vo across the sides of the Wheatstone Bridge will increase or decrease proportional to the load applied. The rated output of a load cell is defined as the ratio of the voltage across Vo to the applied voltage Vex at maximum load. For most load cells, this is on the order of 1 mV V-1.



Rectangular load cell with load applied. Note the placement of strain gauges at each strain point. In the bottom left and right, diagrams of stretched and compressed strain gauges display the thickening effect that compression has on the wire, changing the resistance of the load cell.



Circuit diagram of Load Cell. R1 and R3 are compressed, and R2 and R4 are stretched (or vice versa).

3.3.1.2 Selecting a Load Cell

Several factors went into the selection of load cells for the wind tunnel. First, rough estimates of the forces of lift and drag in the wind tunnel placed them between 0 and 0.5 N [2]. The load cell would thus have to resolve several significant digits of data within this range. Second, the price of the load cell would have to be reasonable within the budget constraints of our project. Given the importance of accurate force measurements, an absolute maximum of $20 for each of four load cell was allotted. However, further reductions in price from this point were considered highly favourable.

Given the above restrictions, the Phidgets 3139_0 Micro Load Cell was selected. This load cell has a maximum reading of 100 g (1 N) and a rated output of 0.6 mV V^{-1}, giving it an error of approximately 0.1 g (1 mN) [3]. This will enable us to measure our forces of lift and drag with between 2 and 3 significant digits. This load cell costs $8.75 per unit from the Canadian distributor Roboshop [4], which is well within our budget of $20. With this equipment, the forces of lift and drag on the wing can be measured. However, the output range of the load cell powered at 5V is 3 mV. This analog signal must be amplified and converted to digital before being read by the Arduino.

3.3.1.3 Load Cell Testing

The assembly of the force-sensing section of the wind tunnel begins with testing the individual components. Five load cells and load cell amplifiers will be purchased and tested, in case one of the components is damaged or broken in delivery or assembly. First, load cells why by tested individually. This would be accomplished by mounting the load cells on a testing apparatus and measuring the response to known weights using a voltmeter.

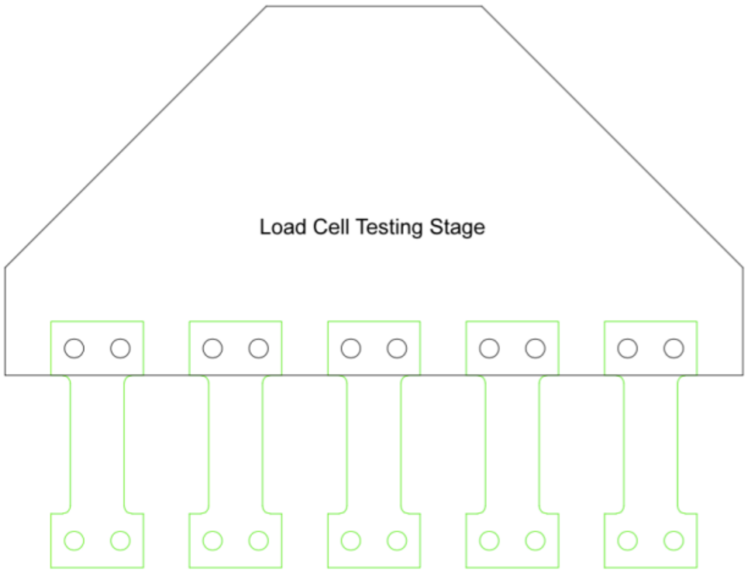


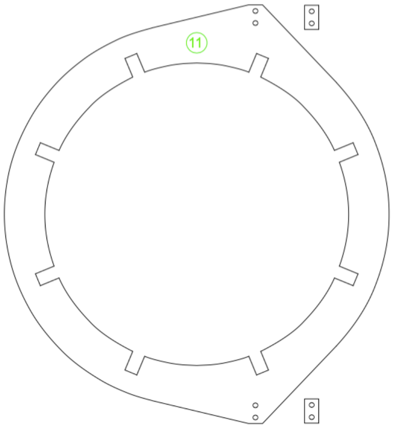
Diagram of the stage used to test load cells during construction. One extra load cell is tested in case of failure of a load cell.

The above testing stage will be laser-cut from MDF. It will be placed on the edge of a table with a counterweight to hold it in place, and the five load cells will be bolted to its edge. Each cell will be connected to the 5V power source from the Arduino, and will be individually loaded with weights up to 100g. The precise mass of these weights will be measured using a high-precision scale to ensure that the values are accurate. The Equus 4320 voltmeter will be used to measure the raw voltages. This instrument has a precision of 100 uV, so this would be able to roughly resolve the signal coming from the load cells. This will be used to verify that each of the load cells is producing a signal, and that it is roughly proportional to the weight applied to it. However, precision testing of the load cells will be left until they are assembled with the load cell amplifiers, as these instruments have a far higher resolution than the voltmeter.

3.3.1.4 Load Cell Mounting

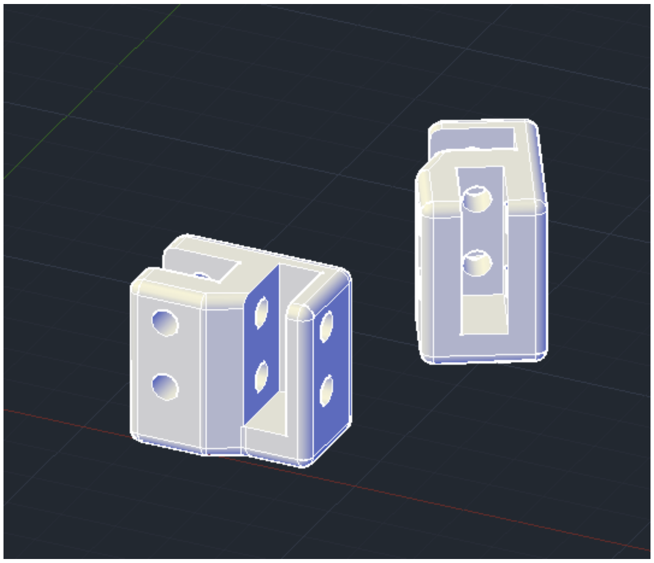
The basic goal in most wind tunnels is the measurement of forces acting on test objects. Force sensors of one type or another effect these measurements. In this case, these measurements are made using tiny bar-type load cells. These load cells measure just 3.2 cm long, but are theoretically capable of producing data resolution in the hundredths of grams. Given this accuracy, it is essential that these load cells be mounted in such a way as to maximize the force applied directly to the sensor and minimize stress experienced by the joints and mounts used to hold the load cells in place.

In this particular wind tunnel, there are four load cells, two at each end of the wing. Of these, one measures lift and one measures drag. Since this type of load cell measures force perpendicular to its length, the drag load cell will be perpendicular to airflow and the lift load cell will be parallel to airflow. For convenience of mounting, the load cell measuring drag is chosen to attach to the structure of the wind tunnel. This is done with a ¼ inch spacer to ensure the load cell does not fowl the strut behind it and to provide space for the junction piece connecting it with the lift load cell. These spacers, and the modified hoop to which they attach, are shown below.



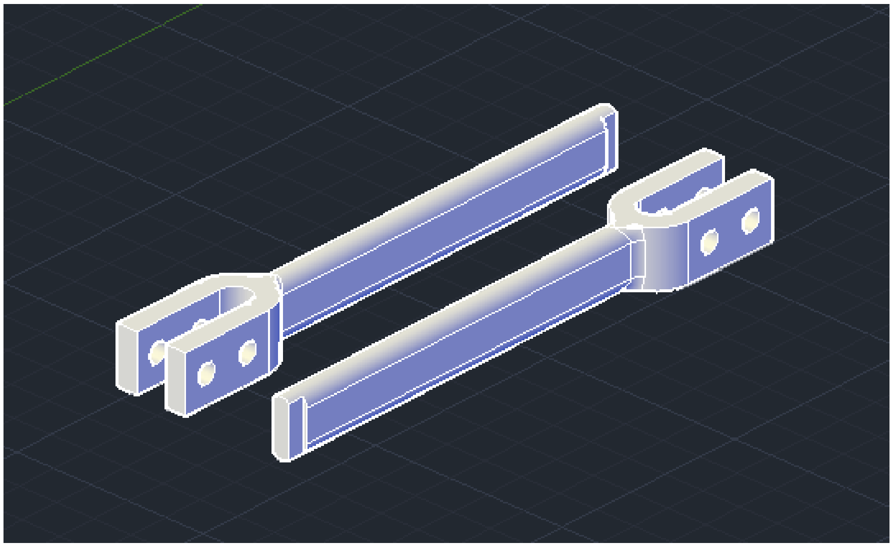
Modified Structur [illegible] load cells. Once attached, the end of the load cell lies over the centre of the tunnel.

Next, the load cells measuring lift are attached to the ones measuring drag. This is done with four bolts and a custom 3D printed junction to hold the cells in place. The design of the junction is such that all force directed backwards is applied directly to the load cell measuring drag. A rendering of this custom part is shown below.



Rendering of a custom 3D printed part that holds the two load cells together. M2.5 bolts are used to secure the piece to the load cells. Here, the drag-measuring load cell would go into the slot in the bottom of the "y" shape, and the lift-measuring cell would go in the top.

The lift load sensor is finally connected to the wing via a second 3D printed part. This part is thin and flat, in order to reduce any drag effects it might experience. This part is shown below.



Rendering of another custom 3D printed part that connects the force sensor measuring lift to the airfoil. These pieces are made small and thin so as to have minimal effect on drag measurements.

3.3.2 Load Cell Amp

3.3.2.1 Load Cell Amp Selection

Several factors went into the selection of load cells for the wind tunnel. First, rough estimates of the forces of lift and drag in the wind tunnel placed them between 0 and 0.5 N [2]. The load cell would thus have to resolve several significant digits of data within this range. Second, the price of the load cell would have to be reasonable within the budget constraints of our project. Given the importance of accurate force measurements, an absolute maximum of $20 for each of four load cell was allotted. However, further reductions in price from this point were considered highly favourable.

Given the above restrictions, the Phidgets 3139_0 Micro Load Cell was selected. This load cell has a maximum reading of 100 g (1 N) and a rated output of 0.6 mV V^{-1}, giving it an error of approximately 0.1 g (1 mN) [3]. This will enable us to measure our forces of lift and drag with between 2 and 3 significant digits. This load cell costs $8.75 per unit from the Canadian distributor Roboshop [4], which is well within our budget of $20.

With this equipment, the forces of lift and drag on the wing can be measured. However, the output range of the load cell powered at 5V is 3 mV. This analog signal must be amplified and converted to digital before being read by the Arduino.

3.3.2.2 Load Cell Amp Electrical Requirements

Although most of the capabilities required for signal processing are contained within the HX711 chip, there are some considerations necessary for connecting it to the Load Cell. With a gain of 128, the ±2.5 V range of the ADC is scaled down to ±20 mV. This is much larger than the Load Cell output of 3 mV. Thus, it must be decided whether a second gain amplifier is required to boost the signal before passing it to the Load Cell Amplifier.

At a gain of 128, the LCA has an input error of 50 nV (root mean squared). Converting this RMS error to peak-to-peak error, the value is 330 nV or 0.33 uV. Given that at 100 g of load, the load cell will output a signal of 3 mV, then this 330 nV peak-peak error roughly equates to an input error of 11 mg. This is well below the intrinsic error of the load cell, which is on the order of 100 mg. Thus, amplifying the signal from the Load Cell would do little to nothing to resolve the signal going into the Load Cell Amplifier.

Without the need for a second amplifier, the setup of the LCA is quite simple. It has inputs for power and ground, and outputs of same to the Load Cell. The designated module also has dual-channel input, with Channel A having a selectable gain of 128 and 64 and Channel B having a gain of 32. To achieve maximum resolution, Channel A will be used for a gain of 128. The output leads from the Load Cell will connect directly to these pins. For data output, there is the Power Down Serial Clock input (PD_SCK) which is pulsed by the Arduino to mediate data output. In order to read the 24-bit digital signal, this pin is pulsed 25 times by the Arduino. The four LCA's in the final project can all be run from the same clock signal, so they will all be connected to the same clock pin in the Arduino. After each pulse, the Serial Data pin (DOUT) changes low or high to represent each binary digit. These values are read by the Arduino, and sent to the computer for data processing.

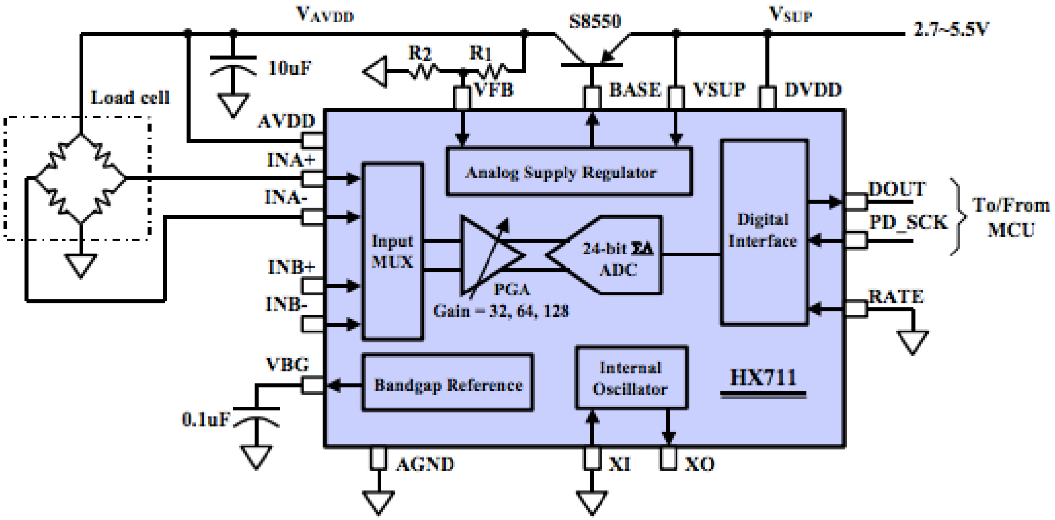
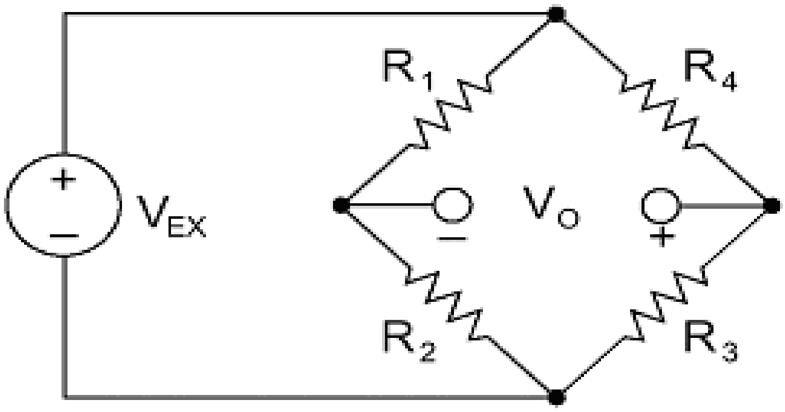


Diagram of Load Cell Amplifier HX711 chip (blue) with surrounding breakout board and load cell.

3.3.2.3 Load Cell Amp Testing

Load Cell Amplifiers take a small analog signal, amplify it, and convert it to digital. To test the load cell amplifiers individually, a known input must be compared to the output from the load cell. While output can be measured reliably using the Arduino, creating a reliable input to serve as a test load cell amp proves difficult. The possibility of constructing a Wheatstone Bridge out of known resistors was entertained. This would require four resistors forming the Wheatstone Bridge, with two additional resistors added to opposite branches, as shown below.



A circuit diagram of the Wheatstone bridge to be used for testing. Vo leads to the load cell amplifier. R1 and R3 will be the large resistor, while R2 and R4 will be a large resistor in series with a small resistor.

KVL and KCL gives the following equation for the voltage across the LCA:

$$V_{LCA} = \frac{10R}{2R_0 + R}$$

Where:

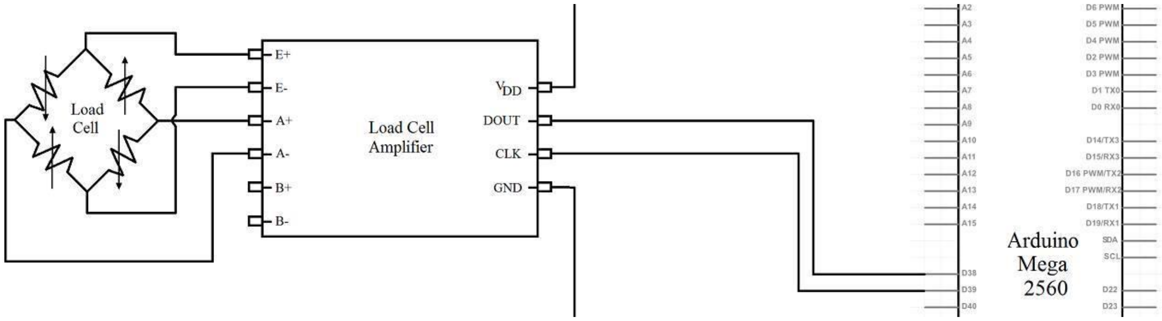
V_{LCA}= the voltage across the LCA input
R_0 = the resistance on each of the four sides of the Wheatstone Bridge (1 MΩ).
R = the resistance of the smaller resistor, used to vary voltage into the LCA (0-600 Ω).

In order to achieve a V_{LCA} value on the order of 3 mV, R_0 value at least 10,000 times greater than the value of R is necessary. This means that in order for our results to be a meaningful estimate of the accuracy of the load cell amplifier, a R_0 tolerance of less than 0.001% is required. While these resistors exist, they are typically priced on the order of $25 and above [1], which is out of our price range for such an unimportant component. It is a better solution to connect the load cell to the load cell amplifier, and testing the assembly with known weights. This will allow for verification of both components at once. If one pair is producing faulty readings, the either the load cell or the LCA may be switched with an alternate component to isolate the error. Given the listed accuracy of the load cell, this method would be an even more precise evaluation of the Load Cell Amplifier's accuracy than even the 0.001% tolerance resistors. It would also allow us to isolate issues with the Load Cells to a greater degree than by using a multimeter. Thus, individual testing of the load cell amplifiers will be exempted in favor of simultaneous testing with the Load Cells.

3.3.3 Connecting Load Cells and Amps to Arduino

Before soldering the Load Cell and Load Cell Amps together, they must be assembled and tested. This will also serve to test the load cell amps, to make up for the lack of individual testing. The method for this will be as follows: first, the five load cells will be mounted on the testing stage described in the Load Cell Testing Section. The power, ground, and output wires of the load cells will be connected to the breadboard. The LCA's will also be connected to the breadboard using male-male header pins. Next, the clock pins of each LCA will be connected to the designated clock output pin on the Arduino. The digital outputs from each of the five LCA's will be connected to the Arduino as well. This setup can be seen below. This concludes the assembly of the testing circuit.

The Arduino will also need to be connected to the computer in order to display the forces created. This requires that basic elements of the GUI be created before load cell testing is done.



This image shows the load cell amplifier connected to the load cell and the pins of the Arduino microcontroller. Each load cell needs its own amp, so this setup would be multiplied 5 times to represent the number of pins and sensors in use for testing purposes. In the wind tunnel, 4 sensors will be used to measure lift and drag.

In setting up force input into the Arduino program, each force sensor will need to be calibrated. This will is because there is significant uncertainty in the rated output of the load cells (0.60±.15 mV V-1). Whatever value of rated output each load cell actually has, this value will be consistent throughout the operation of the load cell, so the data values will still be proportional to the force applied. Thus, through calibration, a conversion constant can be established for each of the load cells between the ADC output value and the force applied. However, before the conversion constant is measured, the load cell amplifiers must be working properly. In testing the LCA's, the load cells can also be measured with a higher degree of precision than previous tests using the voltmeter.

To test the LCA's and load cells, known masses of 0, 50, and 100g will be placed on each of the five load cells. Each Load Cell Amplifier will be tested in each of these 15 conditions, and the results will be compared. It is expected that the 0g reading (testing without any load) will produce a consistent, non-zero offset. As previously discussed, the error of the load cell is an order of magnitude greater than that of the load cell, so any difference is likely to be generated by the load cell. Since the same weight measurements will be in each set of tests, the non-linearity error of the load cell can be eliminated. By constraining our experimental setup to add the weights in the same order, hysteresis error can also be eliminated. Thus, the expected error between the readings of different LCA's will only be the repeatability error of the load cell, which is 0.05% of the maximum reading (when the load cell is loaded with 100g). If the readings of any LCA deviates from the mean consistently by more than the expected error, it will be excluded from assembly of the project and kept as an extra component. If all components perform optimally, one will be chosen at random.

Next, the data from the above tests will be analyzed to find the load cell with the greatest deviation in readings. However, the construction of the load cells is such that they have a Zero Balance Ratio (the sensor output when no load is applied) of ±.1 mV V-1 and an uncertainty in the rated output of ±.15 mV V-1. As a result, significant deviation is expected between the readings of different load cells. In order to compare effectiveness, the load cells must first be calibrated.

Calibration of the load cells will consist of taking a reading at no load and a reading at full load (100 g), and using these to generate a linear function for force with respect to the output of the load cell amplifier. The expected values during this conversion are shown in Table 1. From the datasheet and +5V incident potential, the output of the load cell with no load and at full load can be calculated. With the LCA's 128 times gain, the output of the PGA (Programmable Gain Amplifier) to the ADC can also be calculated. GIven the ADC's input range of -2.5 to +2.5 V [1], and its 24 bits of accuracy, the binary readout from the ADC can also be estimated. This value will be read by the computer and converted into numerical values, also shown on the next page.

Table 1: Expected signal values at no load and full load

Mass	0	100	g
Equivalent force	0	0.98	N
LC Rated Output	(0.0±.1)	(0.6±.2)	mV V^{-1}
LC Output at 5V Vcc	(0.0±.5)	(3.0±.8)	mV
Signal with Gain of 128	(0±60)	(380±100)	mV
Binary Mantissa	0b0000 0000 0000 0000 0000 0000	0b0001 0011 0000 0000 0000 0000	
Binary Uncertainty	0b0000 0011 0000 0000 0000 0000	0b0000 0100 0000 0000 0000 0000	
Numerical Value	(0±200,000)	(1,200,000±300,000)	

From these values, an expression for force applied as a function of the numerical output of the LCA can be found. This would be calculated using the following formula:

$$F = \frac{0.98(x - x_0)}{x_{100} - x_0}\text{ N} \qquad [1]$$

Where

F = the force applied
x = the numerical output of the load cell amplifier
x_0 = the numerical output when no load is applied
x_{100} = the numerical output when 100 g (9.8 N) mass is applied

Given the expected values of N_0 and N_{100}, this equation is expected to be

$$F = (8\pm2)\text{E-7} * x - (0.0\pm.2)\text{ N} \qquad [2]$$

Where

F = the force applied
x = the numerical output of the load cell amplifier

These functions will be found for each load cell. Then, intermediate weights of 25, 50, and 75 g will be tested on the load cells and their conformity to their respective force function will be assessed. Expected tolerances are a combination of repeatability, non-linearity, and hysteresis errors, so the expected tolerances are the magnitude of all three, 0.09% of the maximum load. This equates to 0.9 mN or 0.09 g. The load cell with the greatest error will be selected as an extra component, and the remaining four load cells will be used to construct the wind tunnel.

3.3.4 Soldering Load Cells to Amps

Once the individual components are all tested, the load cells and load cell amplifiers will be soldered together and plugged into the Arduino. Sufficient length of wire will be left between load cells, load cell amplifiers, and the Arduino (these lengths are specified in the Mechanical Design section). The load cell and load cell amplifier chosen to be left out of the project will not be soldered, as they are unlikely to be used and thus soldering them in advance would likely be a waste of time. Once soldering is complete, the load cells will again be mounted on the Load Cell Testing Stage, and their responses to known masses will be assessed. As before, expected tolerances are 0.9 mN or 0.09 g. If one load cell is not performing as expected, the soldering joints will be re-assessed. If necessary, the voltages of the load cells may be checked with a voltmeter in the manner described in the Load Cell Testing section. The wires leading from the Load Cell Amplifier will not be soldered to the Arduino, but will instead be pushed into the input sockets already soldered to the Arduino board. The reason for this is twofold: first, soldering would require the removal of these sockets, and second, by leaving the wires unsoldered the load cells can be easily replaced in the event of a failure. With these wires in place, the electrical assembly of the load cells is complete. All that remains is the installation of these components. This will be described in the Mechanical Assembly Section.

3.4 Anemometer

3.4.1 Placement of Sensor

The speed of the airflow through the wind tunnel must be measured to find the speed at which the airfoil travels through the air. This can be extrapolated to determine the realistic wind speed of a full size wing. The challenge with measuring wind speed is devising a way to accurately and precisely find the speed of the fluid without affecting the airflow to the wing, which may lead to unwanted turbulent effects and incorrect force readings. To avoid these effects, the sensor will be placed 10 cm behind the airfoil. To determine if the airfoil would disrupt the airflow to the sensor, fluid modelling software was used. This software showed that 10 cm is an acceptable distance from the airfoil. Any turbulent effects from the wing will have diminished at that distance, so a smooth flow of air can be detected by the sensor.

3.4.2 Description of Method

The wind speed in the wind tunnel will be measured in accordance with Bernoulli's principle of the venturi effect. This principle is used for incompressible fluids, and at low speeds (subsonic) gases are considered incompressible, so this effect is applicable to the wind tunnel. This principle describes the relationship between moving air and the pressure of the fluid. Bernoulli's principle states:

$$p_1 + \frac{1}{2}\rho {V_1}^2 + \rho g h_1 = p_2 + \frac{1}{2}\rho {V_2}^2 + \rho g h_2$$

Where:

p_1 and p_2 are pressures in different sections of an airstream
$V_1$and V_2 are speeds of the airstream, corresponding to p_1 and p_2
ρ is the density of the fluid
g is acceleration due to gravity
h_1 and h_2 are the heights of measurement, corresponding to p_1 and p_2

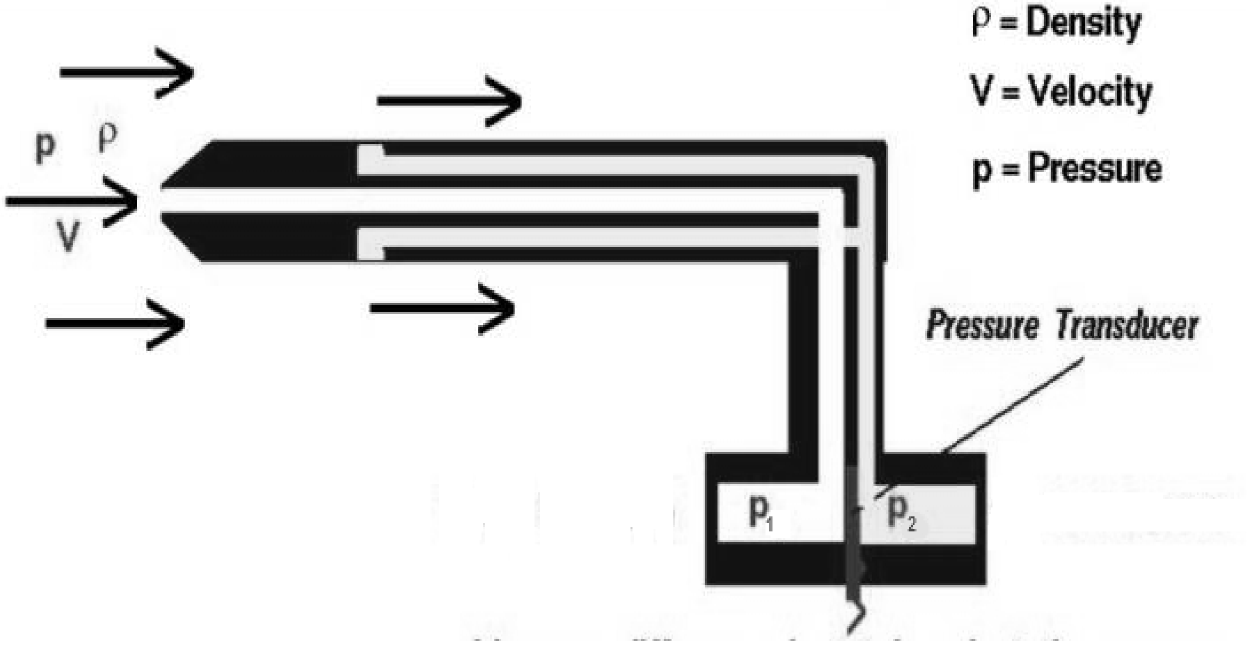
This is an extension of Conservation of Energy: the work done by the pressure, kinetic energy, and potential energy of one section of the tunnel must be equal to the work done in another section of the tunnel.This property of the venturi effect is used in an instrument known as a pitot tube.The pitot tube consists of two tubes, one facing into the airflow and the other perpendicular to the airflow. Due to the incompressibility of air, the local speed of the air at the entrance of the tube facing the airflow is 0. In addition, if the measured heights are the same, Bernoulli's equation can be further simplified:

$$p_1 = p_2 + \frac{1}{2}\rho V^2$$

This can be rearranged to find the speed of the fluid:

$$V = \sqrt{\frac{2(p_1 - p_2)}{\rho}}$$

On the following page is an image of a mechanical pitot tube showing how it can used to measure wind speed:

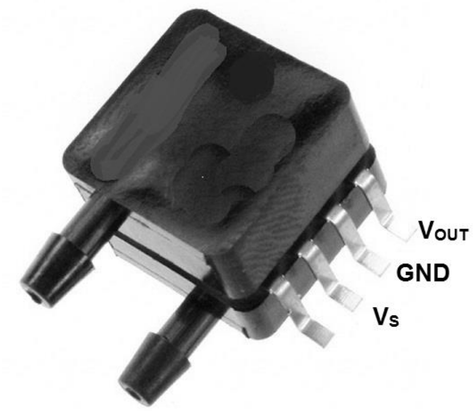


This image shows the mechanism of a pitot tube, a highly precise instrument that can be used to measure wind speed. One section points into the air while the other is perpendicular to the wind. The tube measures the difference of the pressures in these sections, which can be used to find the speed using Bernoulli’s equation.

3.4.3 Sensor to find Wind Speed

Several methods were considered to find the speed of the wind through the tunnel. One possible method was to have the wind spin an impeller. A laser would be shone through the blades of the fan to a photoresistor or photodiode located on the other side of the fan. Theoretically, the moderation between light and dark of the diode or resistor would indicate the rpm of the fan, and consequently, the wind speed. However, it was found that neither a photoresistor nor a photodiode had a response time fast enough to register the number of times a blade passed through the light. As the fan must spin to maintain a wind speed of around 8 ms^{-1}, the rotational speed of the blades is far too great to detect any motion using a light-based device. As a result, this idea was rejected for more favourable sensors. The reliability and accuracy of pitot tube makes it a logical choice for a wind speed sensor. A differential pressure sensor was decided to be more valuable than an absolute pressure sensor because it indicates the difference in pressures, instead of the pressure in one section. Since the difference is used in the calculations for wind speed, it is a much better choice than a sensor of another type.

The sensor chosen to find the speed of the wind is the commonly used MPXV5004DP differential pressure sensor. This sensor is an electronic pitot tube that measures the difference in pressures between the sections as in a mechanical pitot tube. With respect to the diagram above, it acts as the pressure transducer and can supply varying voltage depending on the speed of the wind. This sensor is compatible with the Arduino microcontroller and is easily connected to the microcontroller.



This image shows an electronic differential pressure sensor that can be connected to a microcontroller and will output voltage based on the difference in pressures and is used to find wind speed in the tunnel.

As seen in the diagram, only three pins must be connected to the Arduino to calculate wind speed measurements. V_S will be the 5V provided by the Arduino. This sensor has an acceptable voltage range from 4.75V to 5.25V, so the output voltage of the microcontroller is perfect, and more importantly, the voltage used to power the motor of the fan cannot be used to power this device, as the fan's voltage will not be in the acceptable range for the pressure sensor. The ground pin will be connected to the common ground necessary for electric flow through the sensor. Tubes will be attached to the nozzles of the sensor. One tube will point into the airflow, while the other is perpendicular to it, to mimic the conditions of the pitot tube. The V_{OUT} pin will be connected to analog input on the Arduino so the varying voltages can be read and interpreted to give the speed of the moving air.

3.4.4 Testing and Calibration

Although this sensor is designed to be very accurate (within ±1.5%), it still needs to be tested and calibrated before use of the sensor. To test the pressure sensor and its wind speed readings, it will be necessary to first entirely code the Arduino and be able to display numbers on a 7-segment LEDs. To compare the readings to the actual speed of the wind, the setup will be held outside the window of a car. When the car starts to move, the speed on the speedometer will be compared to the reading on the 7-segment chips by taking simultaneous data points of speedometer and sensor readings over a long period of time and range of speeds. This will allow for accurate comparisons and may show that changes to the setup or the code must be changed. This process will be repeated until a precise and accurate reading is displayed by the LEDs. To calibrate the sensor, the wind speed in the tunnel will be measured when air is not flowing. The wind speed detected by the sensor will be deemed the offset value and will be subtracted from any further calculations.

4. Software Components

The GUI will be the interface that is used to monitor and control the wind tunnel. It calculates and records all the data taken from the wind tunnel such as scaled and true wind speed, and forces on the airfoil, displaying them in diagrams to easily visualize the forces. The GUI (computer) will be directly connected to the Arduino, which drives the wind tunnel and its sensors. However, the Arduino will not perform any calculations on its own; it merely acts as a messenger between the GUI and the wind tunnel.

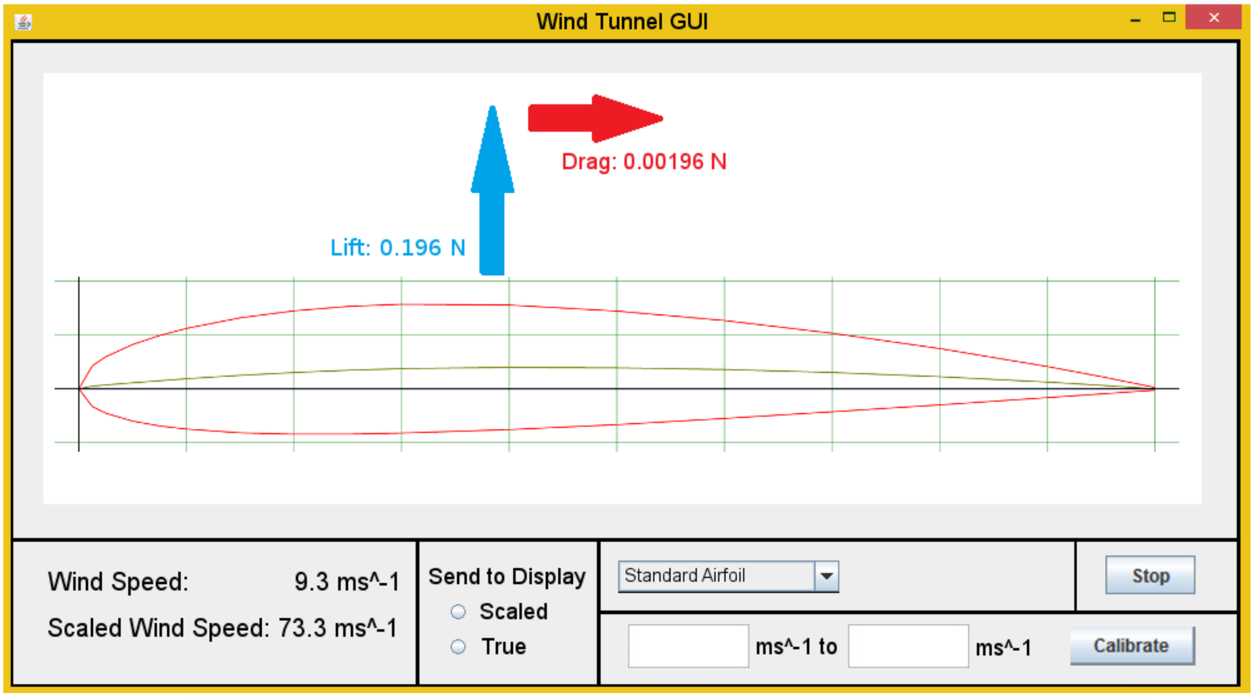
The GUI will show a real time visualisation of the lift and drag forces on the airfoil using 4 different airfoil schematics which can be chosen depending on the airfoil chosen. Forces will be shown using arrows on the airfoil varying in size and length to demonstrate the magnitude of the force. The wind speed will be controlled using PWM analog output to apply varying amounts power to the fan motor. This can be changed manually using the GUI to observe the changes in forces with different wind speeds. The wind speed can be switched between the value in the wind tunnel and the true value as if the airfoil was scaled to its proper size in reality. This can be used to demonstrate the relationship between the model and a full-scale wing. In addition to the GUI, the wind speed will also be displayed on the seven segment displays at three significant digits.

In addition to the basic functionality laid out above, there are several optional features that may be implemented with time remaining. These features are not explicitly mentioned on the wind tunnel requirements, but will increase the utility of the GUI. A master switch can be added to use to turn on and off all power to the wind tunnel as a safety switch when switching airfoils in the tunnel. The fan can be controlled using a pre-programmed routine. This will allow for an automated analysis of the lift and drag forces based on different wind speed without manually varying it. Finally, the data collected by the sensors can be plotted to easily visualize trends between the multiple forces and wind speed. The plot will be displayed regardless of how the data was retrieved, manual or automatic. Also, the data can be exported to a .csv file to save the measurements and for further analyses regarding further relationships between data points.

Among the numerous programming languages that are compatible with Arduino’s serial ports, the GUI will be written in Java for several reasons. Firstly, Java is a popular high-level language that many can understand. Hence, anyone with sufficient experience in Java can modify the program to fit their needs or to add further functionality to improve the program.

Also, creating a graphical interface is exceptionally simple without requiring external libraries or extensive code. Finally, Java is platform independent; a Java executable can be run on any system that has Java pre-installed. This makes the GUI very accessible to all users than having to recompile the source code from scratch.

For Java to communicate with the Arduino using the serial ports, it uses an external library named Ardulink. All the code that runs the GUI is written in Java and run on the computer. A protocol sketch is uploaded to the Arduino that handles all the requests sent from the GUI via a serial port. Ardulink converts all outgoing messages from the GUI into strings sent through serial communication. The protocol on the Arduino then receives the message, parses it, and performs the requested instruction such as reading a value from a pin or outputting a signal to a pin.



Wind Speed: 9.3 ms^-1

Scaled Wind Speed: 73.3 ms^-1

Send to Display

○ Scaled

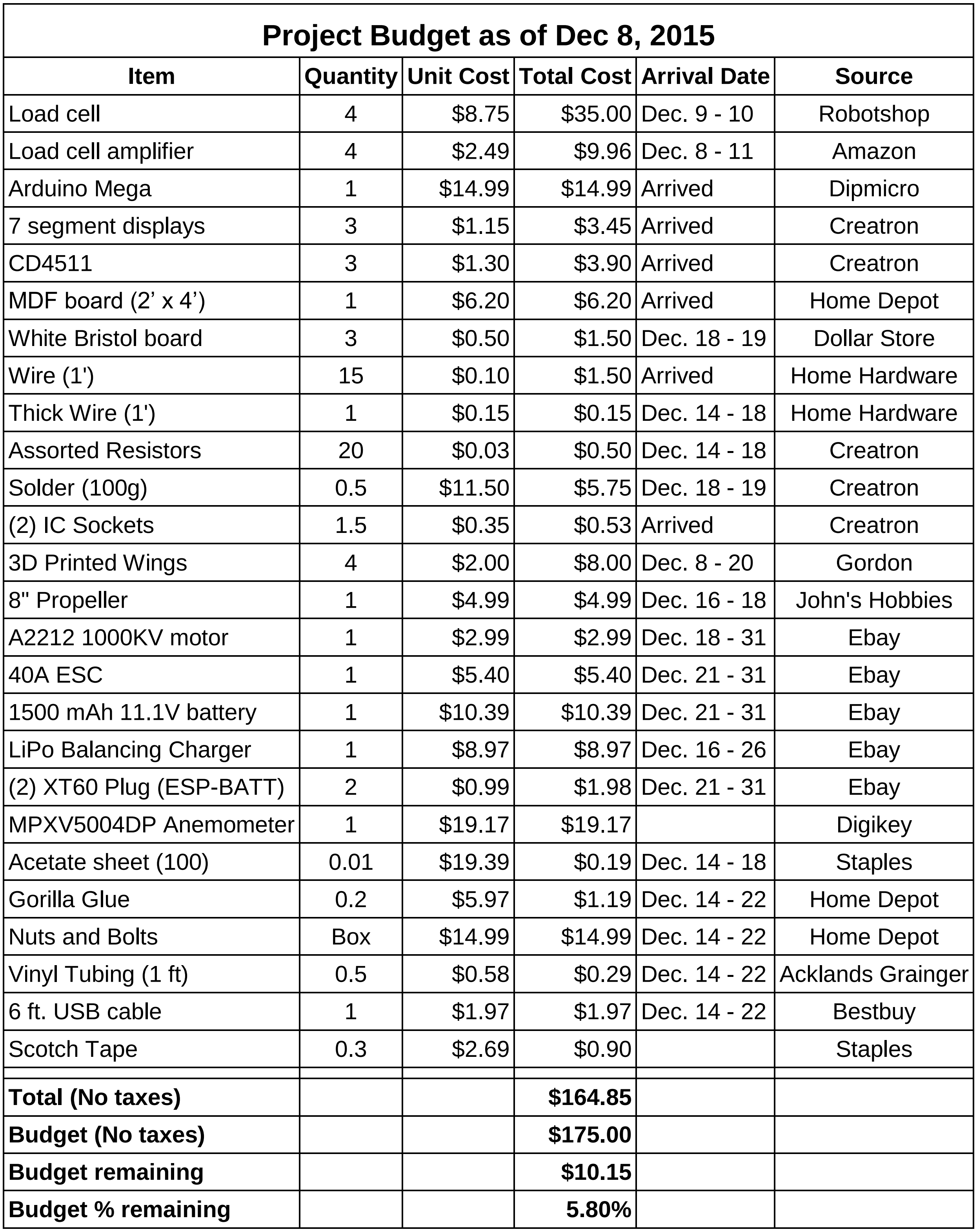
○ True

ms^-1 to ms^-1

An approximate model of the GUI. This mock-up shows a basic layout of the GUI with all of the required components, such as the diagram with the forces, and the wind speed switch. However, this is more for illustrating functionality rather than the layout as these parts are not true Java components but are manually drawn copies. In addition, the implementation of any optional features will modify the layout heavily in order to display them accordingly.

5. Budget

The budget is a list of the different parts needed to build the wind tunnel, complete with the price and quantity of each item. The limit is $175.00 for all supplies that are used. This is an accurate depiction of the cost of the wind tunnel, were this design to be recreated by another party. The budget also shows the expected arrival date of parts to aid in project construction planning. Due to the lack of available mechanical drawings for several components, a block estimate has been used for all metal fasteners. Exact parts and prices may vary slightly from this, but are expected to be below the threshold of $15.



Project Budget as of Dec 8, 2015

Item	Quantity	Unit Cost	Total Cost	Arrival Date	Source
Load cell	4	$8.75	$35.00	Dec. 9 - 10	Robotshop
Load cell amplifier	4	$2.49	$9.96	Dec. 8 - 11	Amazon
Arduino Mega	1	$14.99	$14.99	Arrived	Dipmicro
7 segment displays	3	$1.15	$3.45	Arrived	Creatron
CD4511	3	$1.30	$3.90	Arrived	Creatron
MDF board (2’ x 4’)	1	$6.20	$6.20	Arrived	Home Depot
White Bristol board	3	$0.50	$1.50	Dec. 18 - 19	Dollar Store
Wire (1')	15	$0.10	$1.50	Arrived	Home Hardware
Thick Wire (1')	1	$0.15	$0.15	Dec. 14 - 18	Home Hardware
Assorted Resistors	20	$0.03	$0.50	Dec. 14 - 18	Creatron
Solder (100g)	0.5	$11.50	$5.75	Dec. 18 - 19	Creatron
(2) IC Sockets	1.5	$0.35	$0.53	Arrived	Creatron
3D Printed Wings	4	$2.00	$8.00	Dec. 8 - 20	Gordon
8" Propeller	1	$4.99	$4.99	Dec. 16 - 18	John's Hobbies
A2212 1000KV motor	1	$2.99	$2.99	Dec. 18 - 31	Ebay
40A ESC	1	$5.40	$5.40	Dec. 21 - 31	Ebay
1500 mAh 11.1V battery	1	$10.39	$10.39	Dec. 21 - 31	Ebay
LiPo Balancing Charger	1	$8.97	$8.97	Dec. 16 - 26	Ebay
(2) XT60 Plug (ESP-BATT)	2	$0.99	$1.98	Dec. 21 - 31	Ebay
MPXV5004DP Anemometer	1	$19.17	$19.17		Digikey
Acetate sheet (100)	0.01	$19.39	$0.19	Dec. 14 - 18	Staples
Gorilla Glue	0.2	$5.97	$1.19	Dec. 14 - 22	Home Depot
Nuts and Bolts	Box	$14.99	$14.99	Dec. 14 - 22	Home Depot
Vinyl Tubing (1 ft)	0.5	$0.58	$0.29	Dec. 14 - 22	Acklands Grainger
6 ft. USB cable	1	$1.97	$1.97	Dec. 14 - 22	Bestbuy
Scotch Tape	0.3	$2.69	$0.90		Staples
Total (No taxes)			**$164.85**		
Budget (No taxes)			**$175.00**		
Budget remaining			**$10.15**		
Budget % remaining			**5.80%**		

6. Production Schedule

A production schedule was produced to solidify deadlines to create the wind tunnel before the Jan 9 due date. Most of the scheduling decisions were based on the expected arrival date of components. For example, the fan assembly will be one of the last modular sections built because the motor, ESC, and battery will arrive in late December. But for the seven segment displays where the parts have already arrived, production can begin in mid-December. The production schedule is spread over two pages immediately following the budget.

DATE		Fan/Motor	GUI	Arduino	Assembly
18-Dec			Java Arduino Input		Laser Cut
19-Dec		PWM, TP31, Diode	"		Sanding
20-Dec		Wait for ESC	Output to 7 Seg/Fan		Assemble wood
21-Dec					Assemble Bristol Board
22-Dec			Data Processing		
23-Dec			Data output to 7seg		
24-Dec		ESC Testing – Multimeter, Arduino Cycles through PWM Test Motor with Prop -> 8.11V, 18.23A	Basic GUI		
25-Dec		Assemble set screw, prop, motor	"		
26-Dec		Test with Arduino	"		
27-Dec		"	GUI Functionality – Set/convert speed		
28-Dec		"	Force Diagram		
29-Dec			Force Diagram	Organize wires into Arduino Box	
30-Dec			Data Storage, Master On/Off, Routine	"	
31-Dec			Opt. Features	"	
1-Jan			Opt. Features	"	

2-Jan		**ASSEMBLY**	**ASSEMBLY**	**ASSEMBLY**	**ASSEMBLY**
3-Jan		Overflow	Overflow	Overflow	Overflow

DATE		<u>Load Cell</u>	<u>7 Segment</u>	<u>Anemometer</u>	<u>Airfoil</u>
18-Dec		Testing	Solder 7 seg chips, test frequently		3D Print LCA Ports
19-Dec		“	“		Sand
20-Dec		“	“ Test entire 7 seg		
21-Dec		Test LCA with LC, Arduino	Create module/cord bundle		
22-Dec		“	“	Test sensor with code Wire/solder sensor	
23-Dec		“	Place inside module	Test with car	
24-Dec		Solder LC to LCA		“	
25-Dec		Test LCA Assembly			
26-Dec		“		Set up in tunnel, test	
27-Dec		“		“	
28-Dec		Assemble Load Cell Module Test/troubleshoot		“	
29-Dec		“			
30-Dec		Sand LCA part LCM testing			
31-Dec		LCM testing			
1-Jan					
2-Jan		**ASSEMBLY**	**ASSEMBLY**	**ASSEMBLY**	**ASSEMBLY**
3-Jan		Overflow	Overflow	Overflow	Overflow

7. Sources

1. 24-Bit Analog-To-Digital Converter (ADC) For Weigh Scales. 1st ed. Avia Semiconductor, 2015. Web. 8 Dec. 2015.
2. A2212 1000Kv Brushless Outrunner Motor For Airplane Aircraft Quadcopter Useful. (n.d.). Retrieved December 8, 2015, from http://www.ebay.com/itm/A2212-1000Kv-Brushless-Outrunner-Motor-For-Airplane-Aircraft-Quadcopter-Useful-/151722253490?hash=item235359dcb2:g:TRAAAOSwT6pVh~p-
3. A2212 Motor and 10x4,5 Propeller on the Test Stand; Thrust, Current and Speed Measurement. (n.d.). Retrieved December 8, 2015, from https://www.youtube.com/watch?v=D-eBnXrWJYw
4. Analog Devices : Analog Dialogue : Weigh Scale Design. (n.d.). Retrieved December 8, 2015, from http://www.analog.com/library/analogDialogue/archives/39-12/weigh_scale.html
5. Ardulink. (n.d.). Retrieved December 8, 2015, from http://www.ardulink.org/
6. Bernoulli's Equation. (n.d.). Retrieved December 8, 2015, from https://www.princeton.edu/~asmits/Bicycle_web/Bernoulli.html
7. Brushless Motor A2212/13T 1000KV review. (n.d.). Retrieved December 8, 2015, from https://www.youtube.com/watch?v=NVGkuwHi7ek
8. Cessna 172 Skyhawk. (n.d.). Retrieved December 8, 2015, from http://www.the-blueprints.com/blueprints/modernplanes/cessna/18078/view/cessna_172_skyhawk
9. Controlling an ESC. (n.d.). Retrieved December 8, 2015, from http://forum.arduino.cc/index.php?topic=270309.0
10. Drag Coefficient. (n.d.). Retrieved December 8, 2015, from https://en.wikipedia.org/wiki/Drag_coefficient
11. Digital Counting Scale. 1st ed. 2015. Web. 8 Dec. 2015.
12. ESC Programming on Arduino (Hobbyking ESC). (n.d.). Retrieved December 8, 2015, from http://www.instructables.com/id/ESC-Programming-on-Arduino-Hobbyking-ESC/
13. Force Sensing Resistor. (n.d.). Retrieved December 8, 2015, from https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pd
14. Hot Deals (503)708-2214. (n.d.). Retrieved December 8, 2015, from http://www.batteryheatedclothing.com/pages/A2212-6T-Technical-Data.html
15. How Is Temperature Affecting Your Strain Measurement Accuracy? (n.d.). Retrieved December 8, 2015, from http://www.ni.com/white-paper/3432/en/
16. How to connect MPXV5004DP (pressure sensor) to Arduino Uno. (n.d.). Retrieved December 8, 2015, from http://forum.arduino.cc/index.php?topic=226279.0
17. How to control a brushless motor through a ESC with Arduino. (2012, November 24). Retrieved December 8, 2015, from https://dronesandrovs.wordpress.com/2012/11/24/how-to-control-a-brushless-motor-esc-with-arduino/
18. Lecture #5. (n.d.). Retrieved December 8, 2015, from http://web.stanford.edu/class/me220/data/lectures/lect05/lect_9.html
19. Load Cell Amplifier HX711 Breakout Hookup Guide. (n.d.). Retrieved December 8, 2015, from https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide

20. MPXx5004: 0 to 3.92kPa, Differential and Gauge, Integrated Pressure Sensor . (n.d.). Retrieved December 8, 2015, from http://www.nxp.com/products/sensors/pressure-sensors/differential-gauge-up-to-10-kpa/0-to-3.92kpa-differential-and-gauge-integrated-pressure-sensor:MPXx5004
21. MPXV5004DP. (n.d.). Retrieved December 8, 2015, from http://www.digikey.ca/product-detail/en/MPXV5004DP/MPXV5004DP-ND/1168429
22. NACA 2412 (naca2412-il). (n.d.). Retrieved December 8, 2015, from http://airfoiltools.com/airfoil/details?airfoil=naca2412-il
23. Pitot Tubes. (n.d.). Retrieved December 8, 2015, from http://www.engineeringtoolbox.com/pitot-tubes-d_612.html
24. Phidgets Inc. - 3139_0 - Micro Load Cell (0-100g) - CZL639HD. (n.d.). Retrieved December 8, 2015, from http://www.phidgets.com/products.php?product_id=3139
25. Photoresistor - LDR (Light Dependent Resistor) - Electronics Area. (2014, September 1). Retrieved December 8, 2015, from http://electronicsarea.com/photoresistor-ldr/
26. Similitude Conditions. 2015. Web. 8 Dec. 2015.
27. Suppo - A2212-6 Gold. (n.d.). Retrieved December 8, 2015, from http://www.flybrushless.com/motor/view/50
28. Suppo Speed Controller Programming Instructions. 1st ed. Altitude Hobbies, 2015. Web. 8 Dec. 2015.
29. Venturi effect and Pitot tubes. (n.d.). Retrieved December 8, 2015, from https://www.khanacademy.org/science/physics/fluids/fluid-dynamics/v/venturi-effect-and-pitot-tubes
30. Weighing Scale Design. (n.d.). Retrieved December 8, 2015, from http://www.cypress.com/documentation/technical-articles/weighing-scale-design