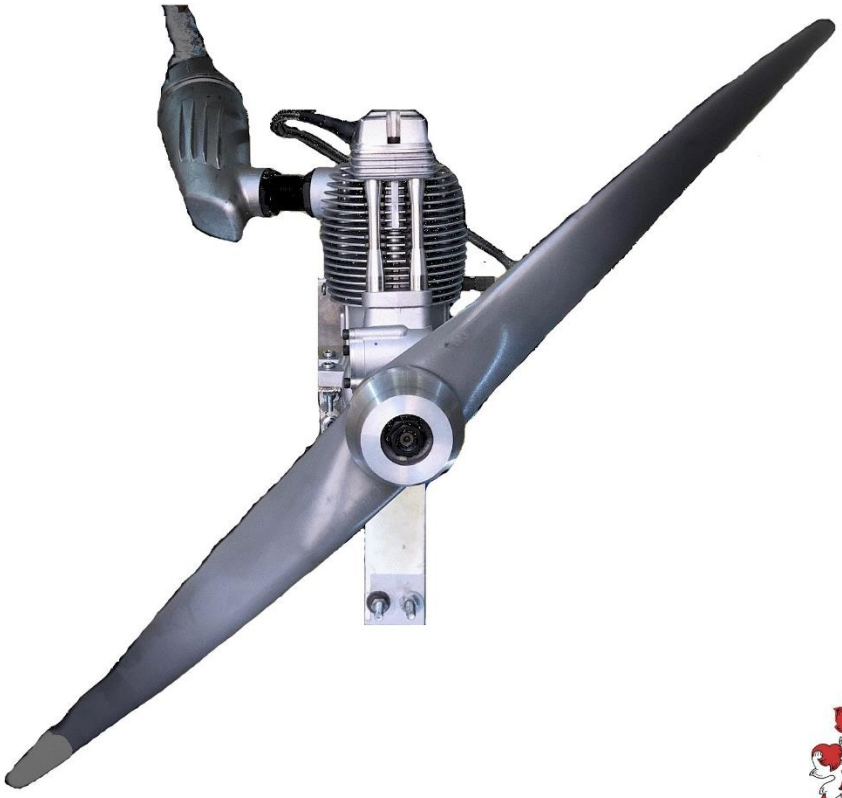
Development and Test of an Arduino DAQ System to Ease Propeller Testing

Rohit Vadlamudi

Supervised by Professor Andy J Keane

Word count: 8200



This report is submitted in partial fulfilment of the requirements for the MEng Aeronautics and Astronautics, Faculty of Engineering and the Environment, University of Southampton

Declaration

I, Rohit Vadlamudi declare that this project and the work presented in it are my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a degree at this University;
2. Where any part of the project has previously been submitted for any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. Except for such quotations, this project is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the project is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. None of this work has been published before submission.

Acknowledgements

Many thanks to my supervisor, Professor Andy J Keane, for his support and guidance during this project.

Thanks also to Mr. Gus Gillam for his assistance in the wind tunnel.

Contents

Abstract

This report details the research and development done to develop an Arduino based data acquisition system to ease propeller testing in the wind tunnel located in room 1025 of building 9 at the University of Southampton. The current technique used to measure a propeller's performance in the wind tunnel is not automated and doesn't always yield accurate results. The IC engine/ DC motor and the wind tunnel are run at various speed settings and the values shown on the different sensor readouts installed in the wind tunnel are photographed or written down for measuring the performance of a propeller. This process is further complicated when a propeller's RPM and the wind tunnel's inlet velocity must be measured as these readouts are not in the same plane as the other sensors. The goal of this project is to develop an easy to use data acquisition system that can data log all the values displayed on the sensor readouts and measure the performance of a propeller and the motor attached to it.

Nomenclature

C	Coulomb
C_T	Thrust coefficient
C_P	Power coefficient
D	Effective duct inner diameter
η	Aerodynamic efficiency
Hz	Hertz
J	Advance ratio
kV	Motor velocity constant
T	Total Thrust
mAh	Milli-Ampere-hours
V	Volt
n	Rotations per sec
N	Newton
Q	Torque
$P_{electrical}$	Electrical power consumed
P	Propeller power output
A	Amps
Kg	Kilogram

Abbreviations

DAQ	Data Acquisition
CPU	Central Processing Unit
ICE	Internal Combustion Engine
DC	Direct Current
AC	Alternating Current
LCD	Liquid Crystal Display
ADC	Analog-to-Digital Converter
RPM	Rotations Per Minute
HMI	Human Machine Interface
KB	KiloByte
ISP	In-System Programming
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
MIPS	Millions of Instructions Per Second
SPI	Serial Peripheral Interface
IIC	Inter-Integrated Circuit
RS-232	Recommended Standard 232
RS-232	Recommended Standard 485
GUI	Graphic User Interface
LiPo	Lithium Polymer Battery
EMI	Electromagnetic Interference
RFI	Radio Frequency Interference
TFT	Thin-Film Transistor
UAV	Unmanned Aerial Vehicle
UIUC	University of Illinois at Urbana-Champaign

Chapter 1

Introduction

1.1 Project Overview

This report details the research and development done to develop an Arduino based data acquisition system to ease propeller testing in the wind tunnel located in room 1025 of building 9 at the University of Southampton.

Figure 1: Wind tunnel.

As seen in Figure1, this wind tunnel forms an open circuit and allows for a positive airflow to be forced past the engine/propeller to simulate flight speeds for various flight conditions such as take-off, cruise and landing. This wind tunnel's test section is a meter long and has a cross-sectional area of 0.48 x 0.475 meters. A honeycomb flow straightener installed at the inlet straightens the air flow in the wind tunnel by minimizing velocity components caused by swirling of the flow during entry. The maximum working airspeed of the tunnel, when a motor under test, is 20m/s. The actual testing of an IC engine or a DC motor with propeller is performed behind the polycarbonate armoured surround shown in Figure1 to protect the operator from moving parts. The maximum size of a propeller that can be tested in the wind tunnel is 24 inches in diameter. The wind tunnel also incorporates a dedicated exhaust extraction line for IC engine testing.

The current technique used to measure a propeller's performance in the wind tunnel is not automated and doesn't always yield accurate results. The IC engine/ DC motor and the wind tunnel are run at various speed settings and the values shown on the multiple sensor readouts installed in the wind tunnel are photographed or written down for measuring the performance of a propeller. This process is further complicated when propeller's RPM and the wind tunnel's inlet velocity must be measured as these readouts are not in the same plane as the other sensor readouts. The goal of this project is to develop an easy to use data acquisition system that can data log all the values displayed on the sensor readouts and measure the performance of a propeller with the motor attached to it.

1.2 Existing Instrumentation

The test bench has two load cells that measure the Thrust and Torque generated by a propeller. An L-shaped beam connects the Torque loadcell with the motor and the Thrust loadcell directly connects to the motor as illustrated in *Figure2.* These loadcells are connected to weighing scale indicators that display the force applied to them in Kgs on small LCD readouts. The airspeed in the wind tunnel is measured by an L-shaped pitot tube that connects to a Manometer. The pitot tube is fixed to a circular acrylic disk that sits down from the top side of the polycarbonate armor as illustrated in *Figure2*. This type of arrangement allows measurement of airspeed at different points in the wind tunnel. The propeller's RPM is calculated by pointing a Laser/IR tachometer onto the rotating propeller.

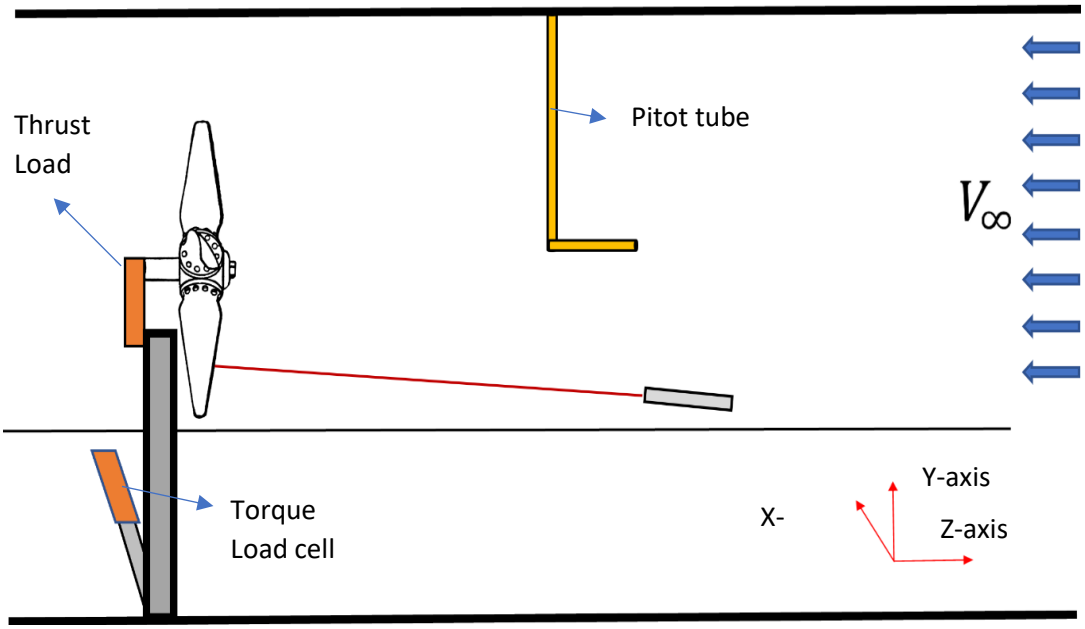


Figure 2: Illustration of Wind tunnel's Test Section.

Figure 3(a), 3(b) & 3(c): Fuel Sensor readout (left); Weighing Scale readout (middle); Extech Manometer(right).

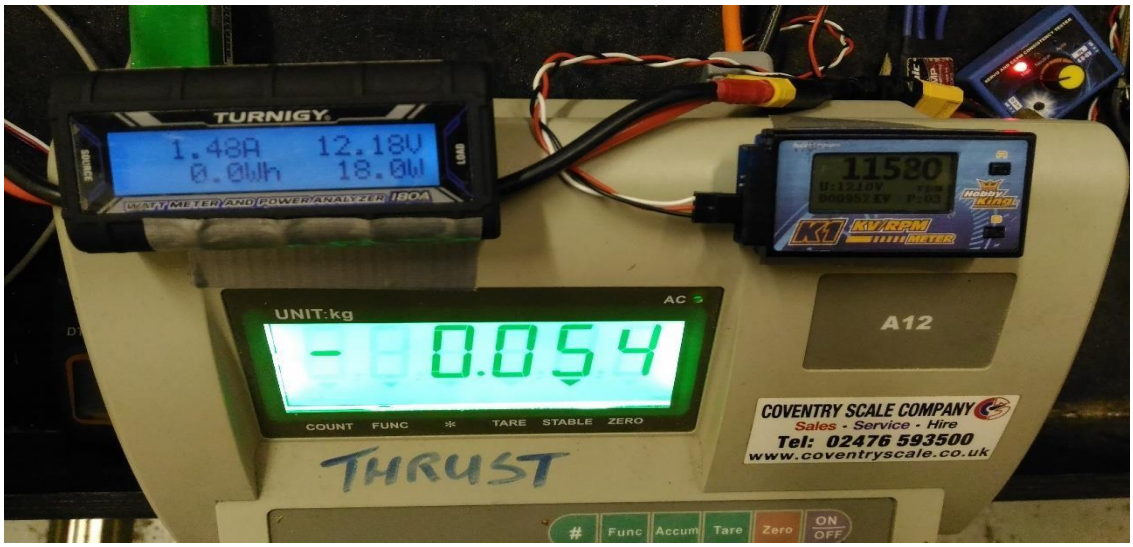


Figure 4: Thrust Weighing Scale readout with Power Analyzer for DC motor testing [1].

The 2 single-point load cells from the test bench are connected to weigh scale readouts like the ones shown in *Figures3(b)&4*. These readouts have an Analog-to-Digital converter inside them that amplifies and conditions the signal coming from a loadcell to measure the force applied. The Extech Manometer shown in *Figure3(c)* measures the differential pressure from the L-type pitot tube and calculates the velocity in the wind tunnel's test section. If an internal combustion engine is under test, the fuel is sent to the engine through a turbine hall effect flow sensor and the fuel flow rate is displayed on the fuel sensor readout shown in *Figure3(a)*. Channel A of the fuel flow sensor readout shows the fuel flow rate and Channel B of the readout gives the total fuel consumed. Students testing a DC motor with propeller use a power analyzer like the one shown in top-left corner of *Figure4* to measure the voltage and current drawn from an energy storage device.

1.3 Aims and Objectives

The purpose of this project is to develop an Arduino-based DAQ system that eases the process of testing IC engines and DC motors with propellers in the wind tunnel. The DAQ system must read all the performance parameters such as Thrust, Torque, RPM, Airspeed, Electric Power and Fuel flow rate from various sensors and transducers and log them on to a storage device.

There are 4 main objectives for this project are to

- Develop a simple DAQ system that logs enough data such as Thrust, Torque, RPM and Airspeed to measure just a propeller's performance.
- Further develop the system to log electrical power consumed by DC motors and Fuel flow rate consumed by IC engines.
- Install an HMI touch screen device to enable users to calibrate the sensors easily.
- Test and validate the DAQ system.

1.4 Report Structure

This report is divided into five main parts. Firstly, the layout of the wind tunnel and the current method for testing propellers with motors along with its implications are discussed in Chapter 1. Chapter 2 provides detail on the research undergone for developing the Arduino DAQ system while providing analytical knowledge on how the instruments installed in the test bench work. Chapter 3 provides detail on the development of the Arduino DAQ system to log enough performance parameters to benchmark just the propellers. Chapter 4 discusses on further developing the system to data log motor performance parameters such as Voltage and Current drawn or Fuel flow rate depending on the motor type. Both these chapters provide detail on how the Arduino system is developed programmatically and on how it is interfaced with the sensors.

Chapter five focuses on the HMI touch screen device implementation and provides detail on its GUI design. The final part is the validation of this system and it demonstrates the system's ability to data log all the performance parameters and compares the data recorded from testing two propellers against the data from the UIUC propeller database to validate its functionality. The codes implemented in the Arduinos can be found in the ArduinoDAQ GitHub repository [2].

Chapter 2

Literature review

2.1 Load cells

"Load cells are a type of transducers that change the force applied to them into an Analog voltage signal. The voltage signal generated by a load cell is directly proportional to the force applied to them."[4] These are various load cell types and we will be dealing with strain gauge type load cells in this project. The load cells used in this project come with a four-wire configuration as seen in *Figures5(a)&5(b).* This configuration is called a Wheatstone bridge formation and the resistance of these wires change if they are under compression or tension. [3] This formation has 4 output wires as seen in *Figure5(a),* the red and black wires to the wheat-stone bridge in the load cell are for the excitation voltage and the yellow and green wires are the signal. The voltage signal from the load cell is sent to a load cell amplifier that conditions and amplifies it so that it can be sent to LCD readouts or computers. Every Load cell has a sensitivity and excitation voltage rating. Suppose, a load cell has a sensitivity of 6mV per V and an excitation voltage of 10V. This means at full load capacity the load cell would output 60mV when excited with 10V. These ratings are crucial as a load cell must be matched with a suitable ADC. [4]

Figures 5(a) & 5(b): Wheatstone bridge arrangement [4] (left); Wires coming from a load cell [4] (right).

2.2 Analog-to-Digital Converters

ADCs used in this project convert the analog signal coming from the loadcells to a digital number. The resolution of an ADC determines the number of binary digits that correspond to the digital number that represents the analog value. This is only an approximation of the real analog signal value as the digital voltage can only be represented in discrete steps. The resolution of the ADC indicates the number of discrete values it can produce over the range of analog values.[5] The A12

weighing scale indicators used in this project have a 20bit ADC inside them. This means the input analog signal from the load cell could be resolved into 0.00476 mV/bit. We use a different ADC with a resolution of 24bits to read the force applied to the load cells with an Arduino. The resolution per bit is given by the *Equation 2.1*. [5]

$$resolution = \frac{Excitation\ Voltage}{2^{number\ of\ bits}} \quad (2.1)$$

2.3 Pitot tube & Differential Manometer

Using an anemometer in this small wind tunnel to read the flow velocity would not be ideal because of its rotating cups. It is unsafe to have moving parts inside the test section when a propeller is under test. The best solution would be to use an L-type pitot tube that measures both the static pressure (pressure of the flow when travelling along with it) and stagnation pressure (pressure of the flow when it is brought to rest). Differential pressure is defined as the difference between stagnation and static pressure. [6]

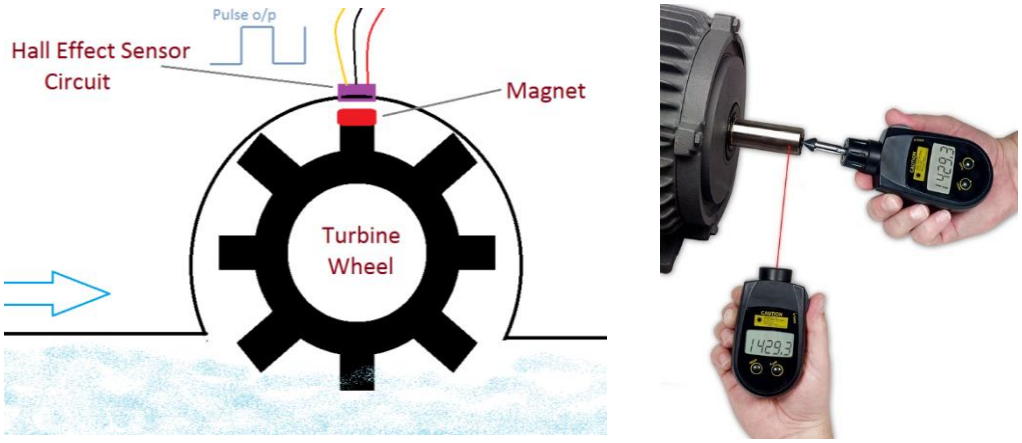
The pitot tube is connected to a pressure transducer that gives a signal that is linearly proportional to the differential pressure measured in the wind tunnel. The velocity of the flow can be calculated by applying Bernoulli's Principle shown in *Equation 2.2*. [7] where p is the pressure, ρ is the density, V is the flow velocity, g is the acceleration due to the gravity of earth, and h is the height from a reference plane. The equation can be rearranged to get the flow velocity as shown in *Equation 2.3* where Δp is the differential pressure. [7]

$$p + \frac{1}{2}\rho V^2 + \rho g h = constant \quad (2.2)$$

$$V = \sqrt{\frac{2(\Delta p)}{\rho}} \quad (2.3)$$

2.4 Hall effect sensors and Laser Tachometers

" Hall effect-based sensors output a voltage signal that is directly proportional to the magnetic field measured by it".[8] Hall effect sensors have many applications and, in this project, we use a hall effect sensor to measure the volume flow rate of the fuel to the engine. The number of pulses given out by this sensor can be used to determine the fuel flow rate. The working principle of a hall effect sensor can be seen in *Figure 6(a)*. [8]



Figures 6(a) & 6(b): Working of Flow Sensor [9] (left); Laser tachometer [10] (right).

Laser/IR tachometers also work on a similar principle and they can be used from a distance by beaming the laser/IR light towards the spinning object. An IR/laser detector generates a pulse signal by measuring the rate at which the beamed light is obstructed or reflected depending on the arrangement *[Figure 6(b)]*. The tachometer used in this project uses a Laser instead of an IR light because most IR detectors are limited by their detection range.

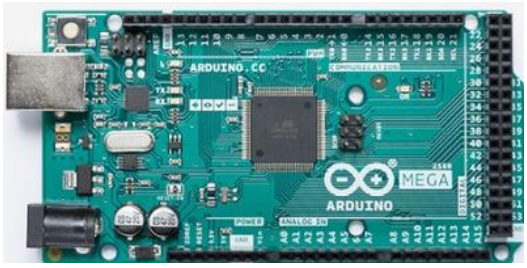
2.5 Arduino

"Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and kits for building digital devices and interactive objects that can sense and control both physically and digitally. The boards are equipped with sets of digital and analog input and output pins". [11]

2.5.1 Arduino Mega vs Nano

The Arduino Mega 2560 shown in *Figure7(a)* is a 101.52 mm X 53.3 mm microcontroller board based on the 8-bit ATmega2560 Microchip. It comes with a 16MHz oscillating crystal, 256KB ISP flash memory for storing code, 8KB SRAM, 4KB EEPROM. This board has more than 50 digital input/output pins,16 analog inputs, 4 hardware serial ports. It can be powered with an external power supply of 4.5 to 12 volts and connects to a computer with a USB cable. Both the Arduino Mega and Nano approach a throughput of 1MIPS per MHz [12].

The Arduino Nano shown in *Figure7(b)* is a smaller microcontroller based on the same ATmega328 microchip. It has the same functionality of the Arduino Mega. The ATmega328 has only 32 KB of ISP flash memory for storing code 2 KB of SRAM and 1 KB of EEPROM. The board has 8 analog inputs, and 14 digital input/output pins and only one hardware serial port. [12]



Figures 7(a) & 7(b): Arduino Mega [13] (left); Arduino Nano [14] (right).

Both the Arduino Mega and Nano support Serial and IIC communication and this is discussed in *Chapter 2.5.2*. They also have a voltage regulator that maintains a constant voltage of 5 volts and a 10bit Analog-to-Digital Converter. This means they provide a resolution of 4.88mv per bit. This resolution is pretty much useless in precisely reading the force applied on the load cells as they only produce 10mV at full load capacity. These microcontrollers are programmed with the open-source Arduino integrated development environment software. [12]

2.5.2 Serial and IIC Communication

Serial communication is an easy and flexible way for Arduino to interface with computers and other Arduino boards. It is the process of sending data one bit at a time, sequentially, over a channel. This way of exchanging data in the form of serial digital binary is cheap and maintains signal integrity over long distance cabling. Some of the well-known interfaces used for the data exchange are RS-232, RS-485, IIC, SPI etc. We use IIC and RS-232 interfaces to exchange data in this project. The load cells are connected to the Analog-to-Digital converters through RS-232 cables. [15] [17]

Serial communication can take many forms depending on the type of transmission mode and data transfer. In this project, we use the full-duplex transmission mode and the wiring for this mode is shown in *Figure 7(b)*. This means both the sender and receiver can transmit and receive at the same time. We can also define how fast data can be transferred over a channel by changing the baud rate. The load cell amplifiers operate at a baud rate of 9600 bits per second. Other standard baud rates are 1200, 2400, 4800, 19200, 38400, 57600, and 115200 bits per second. [15] [17]

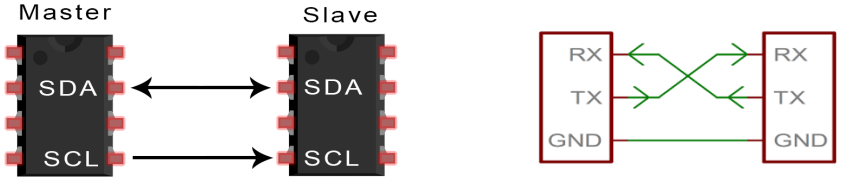


Figure 7(a) & 7(b): IIC communication wiring [16] (left); Serial Communication wiring(right) [17].

“IIC is a type of serial communication protocol, in which the bits are sent sequentially along a single wire (the SDA line)., I2C communication is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master Arduino”.[16] The wiring for IIC is shown in *Figure 7(a).*

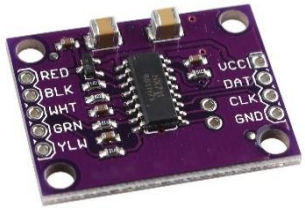
The main advantage of using IIC over Serial communication is that the IIC is a bus which supports many devices, each with own address unlike Serial communication, which is a point to point connection. IIC has better speed than Serial communication, but Serial communication supports longer wires. We are using IIC in this project to interface a 24bit Analog-to-Digital converter to the Arduino. [16]

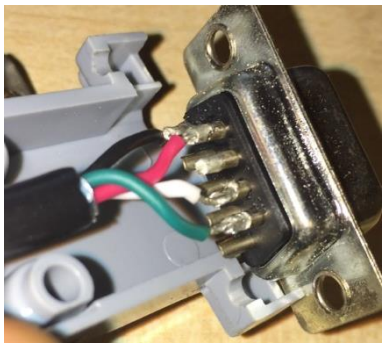
Chapter 3

Development of DAQ System Part 1

3.1 Reading Thrust and Torque with an Arduino

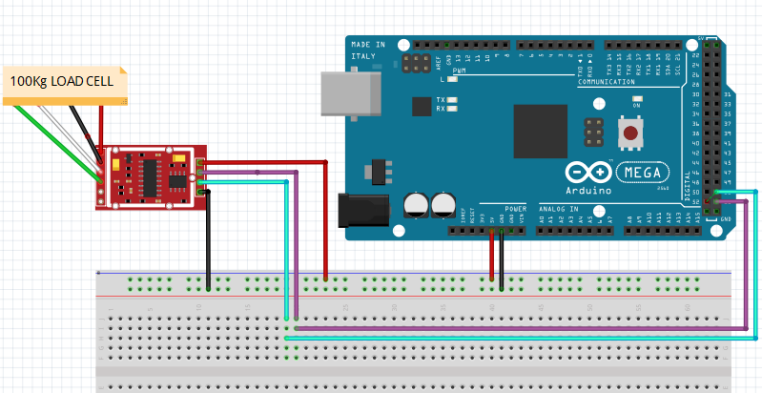
The loadcells installed in the wind tunnel are SPO Series Single Point Load Cells [18] with a maximum load capacity of 20 Kgs. These load cells require a minimum excitation of 5 volts and their full-scale output is 2mV/V +/- 10%. [18] This means under full load capacity at 5 volts excitation the signal has a potential difference of 10 mV+/- 10%. The Arduino's 10-bit ADC doesn't provide excellent resolution for reading these small voltages. Thus, two high precision 24-bit ADC breakout boards called HX711[19] shown in *Figure8(a*) were purchased to amplify the signals from the load cells. When 5 volts from the Arduino is supplied to the HX711 sensor, the gain of 128 corresponds to a full-scale differential voltage sensing of ± 20 mV. The red and black wires are for excitation and the white and green wires are the signal coming from the loadcell shown in *Figure8(c).* [4]

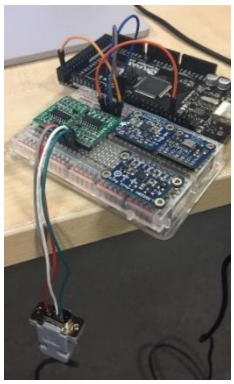




Figures 8(a), 8(b) & 8(c): HX711breakout board [19] (left); Load cell Wires(middle).; 100Kg Load cell(right).

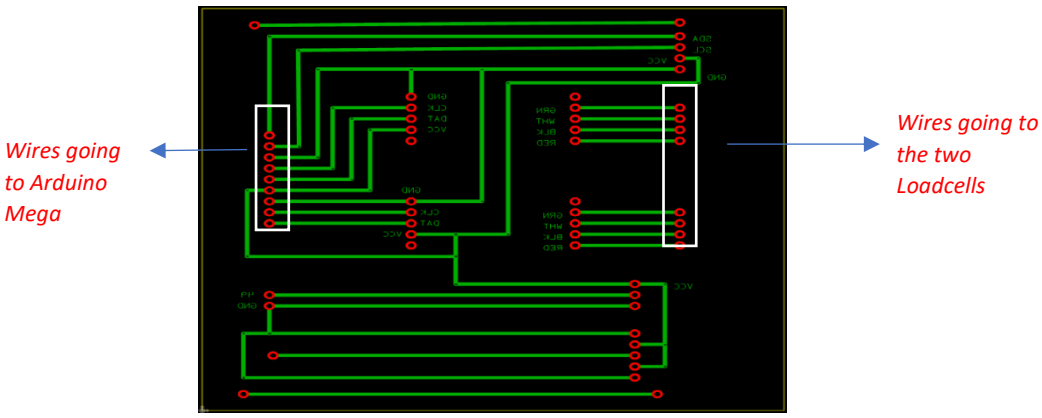
Before interfacing the loadcells installed in the wind tunnel with an Arduino, a 100Kg load cell shown in *Figure8(c)* was provided by Professor Andy Keane for initial testing.





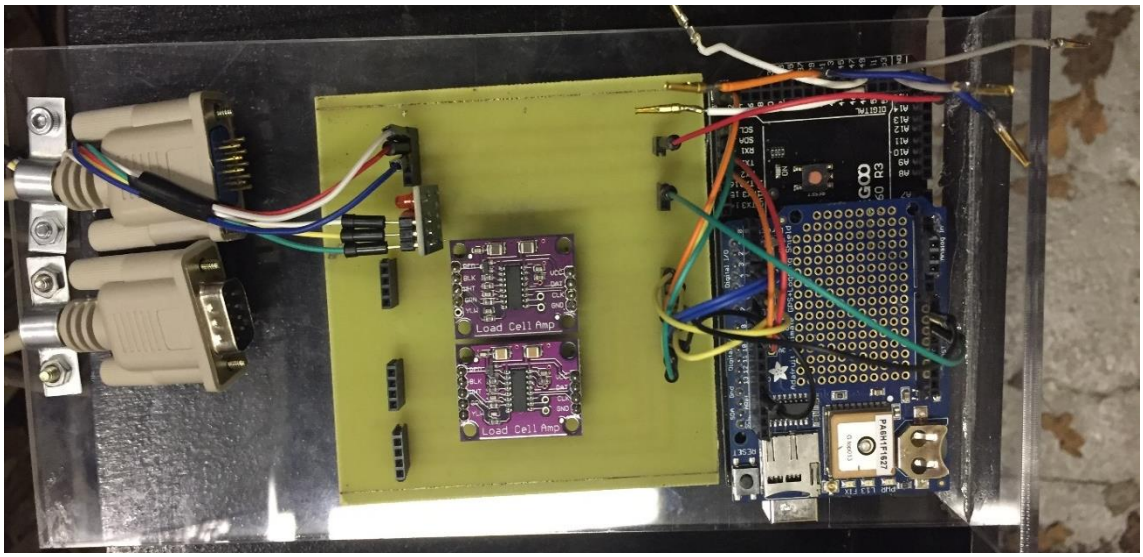
Figures 9(a) & 9(b): HX711 Circuit Diagram (left); 100Kg Load cell testing(right).

The white and green wires are the signal from the load cell going to HX711 board as illustrated in *Figure 9(a)*. The cyan and purple wires are the serial communication wires to the Arduino Mega. All the Arduino circuit diagrams shown in this report are designed in an Open-source software called Fritzing [20]. After successfully interfacing the 100Kg load cell with the Arduino Mega a custom printed circuit board shown in *Figure10* was designed and manufactured by the electronics workshop located in building 13 at University of Southampton. This PCB can hold two HX711 breakout boards with female headers.



Figures 10: Initial PCB design.

The Arduino Mega and the PCB shown in *Figure11* were attached to the acrylic base using heavy-duty stick-on tape and then taken to test bench for interacting the 20Kg loadcells. The Arduino Mega was successful in reading the force from the two load cells at the same time and displaying in on its serial monitor.



Figures 11: Circuitry Version 1.

The loadcells must be calibrated as the readings shown with the Arduino can be off due to things like temperature, creep and electrical interferences. The HX711 ADC doesn't require any calibration but, the load cells must be calibrated periodically. [4]

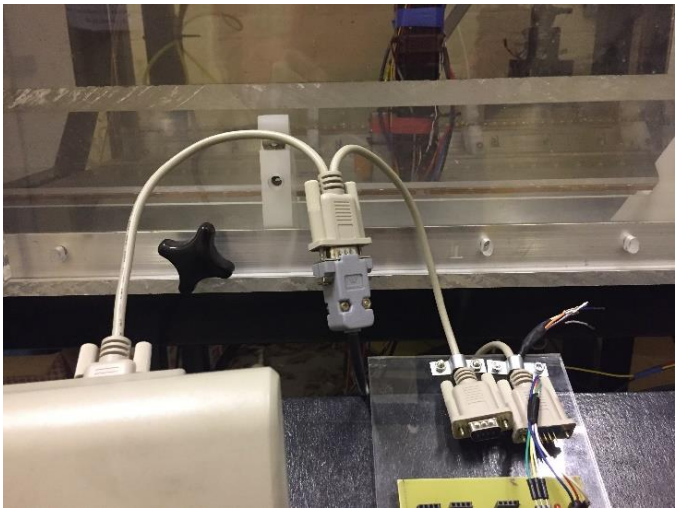


Figure 11: RS-232 cable splitter.

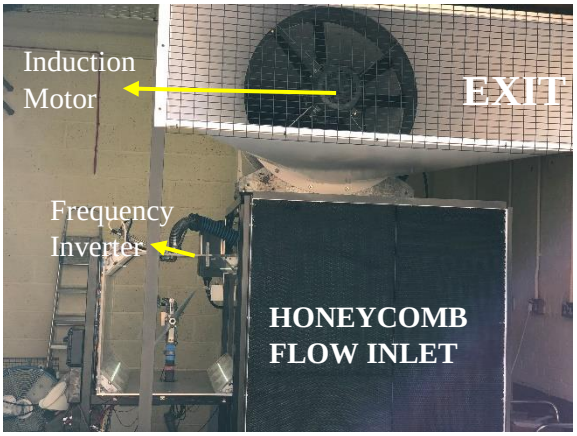
The serial signal coming from the loadcells was split using an RS-232 splitter as shown in *Figure11* and sent to weigh scale readouts and to the Arduino Mega at the same time. This allowed for adjusting the calibration factors of the two load cells until they showed accurate force readings. The readings were verified by applying a constant force on the loadcells with a spring balance. The code implemented in the Arduino allows for easy calibration and more detail on this provided in Chapter 4.

3.2 Reading Airspeed with an Arduino.

The airspeed in the wind tunnel is currently measured by a 12” L-type pitot tube connected to the Extech Manometer shown in *Figure12(c).*

Figures 12(a), 12(b)&12(c): Pitot Tube (left); Differential Manometer(middle); Pitot tube with Manometer (right) [21].

This 12” pitot tube sits down from the top side of test section as shown *in Figure12(a).* The inlet velocity of the wind tunnel is set by adjusting the speed of a 3-phase induction motor that pulls air into the tunnel through the honeycomb shown in *Figure13(a).*



Figures 13(a) & 13(b): Wind Tunnel Overview (left); Eurotherm 584S Frequency Inverter

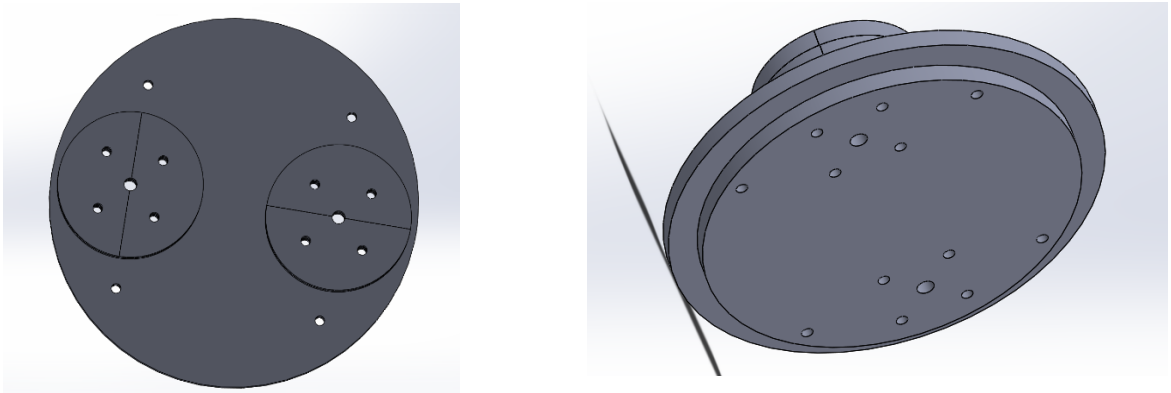
The speed of the induction motor is set by the frequency inverter shown in *Figure 13(b).* The cold

weather during the testing limited the test velocity inside the wind tunnel to nearly 16 meters per second as the density of air changes with variation in temperature. Hence, the propellers couldn't be tested at wind tunnel's maximum working speed of 20 meters per second.

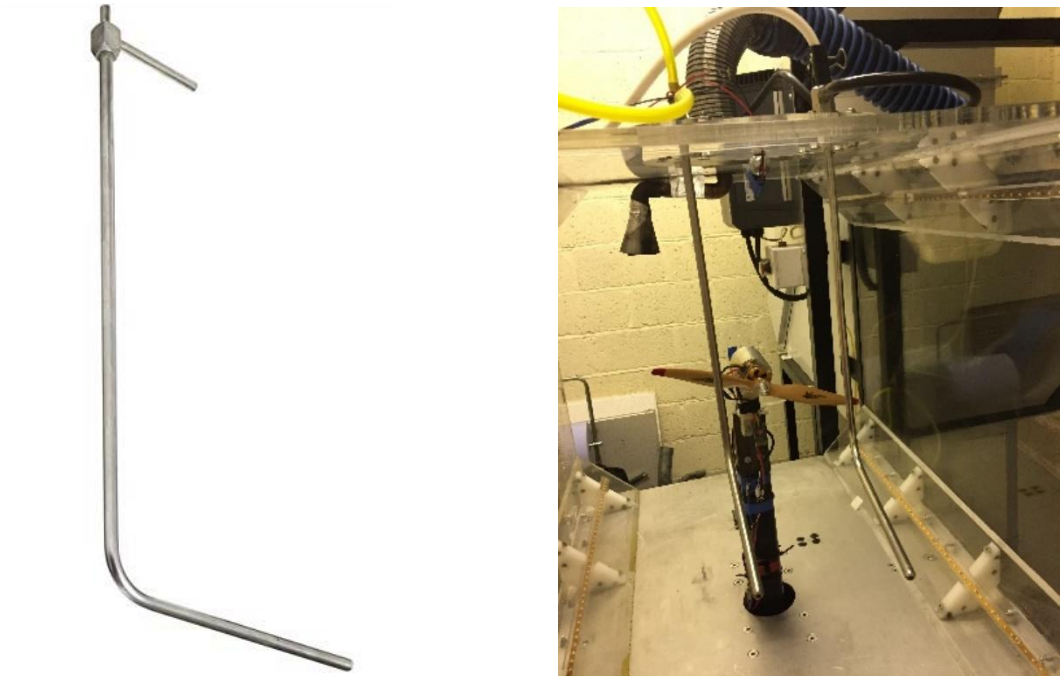
To read the flow velocity in the wind tunnel with an Arduino, the idea of attaching a small UAV pitot tube like the one shown in *Figure14(b)* to the existing pitot tube was proposed. This idea is illustrated in *Figure 14(a).*

Figures 14(a)& 14(b): Proposed Idea (left); UAV Pitot tube (right) [22].

It was later decided that it would be ideal to use another 12" pitot tube which is like the existing one in the wind tunnel to read the flow velocity with the Arduino. A Fluke PT12 12" Pitot Tube [23] was purchased and installed in the wind tunnel. A new pitot tube holder was that can accommodate two pitot tubes at the same time was designed in Solidworks and then manufactured with 8mm acrylic sheets by the Engineering Design and Manufacturing Centre at the University of Southampton. The 3D models of the holder can be seen in Figures *15(a)&(b).*

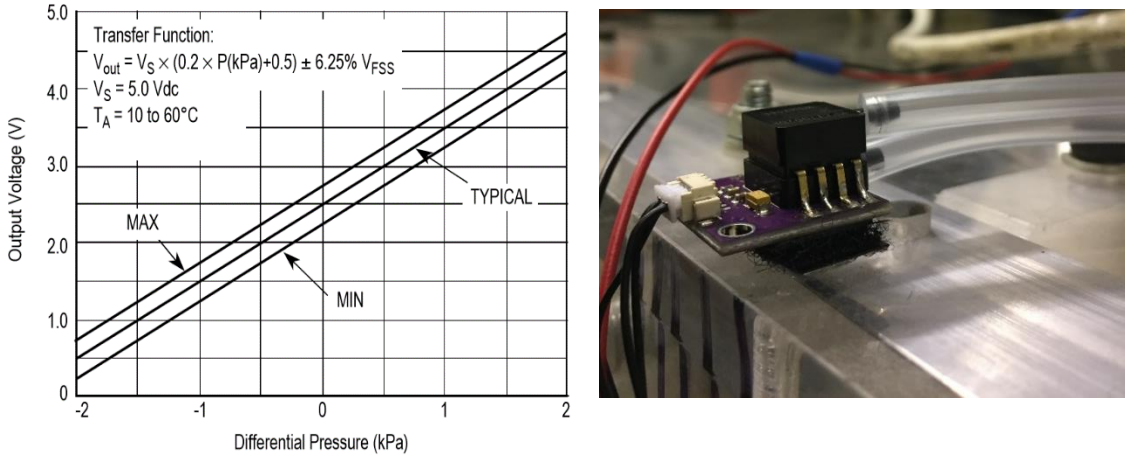


Figures 15(a) and 15(b): Top view Pitot tube holder (left); Bottom view Pitot tube holder (right).



Figures 16(a) & 16(b): Fluke 12" Pitot Tube [23] (left); New Pitot tube holder sitting from the top side(right).

The new pitot tube holder can be seen in *Figure16(b)* holding both the pitot tubes at the same time. This arrangement made the validation of velocity readings from the Arduino against the values displayed on Extech Manometer easy. The new pitot tube holder was successful in holding both the pitot tubes stationary under full operational flow speed. The new pitot tube was connected to a pressure transducer called MPXV7002DP [24] shown in *Figure17(b).* This transducer gives out an analog voltage signal directly proportional to the dynamic pressure measured by the pitot tube. This pressure transducer can measure airspeeds greater than 200 meters per second.



Figures 17(a) and 17(b): Output vs Differential Pressure from MPXV7002DP [25] (left); MPXV7002DP (right).

It can be seen from the *Figure17(a)* that the sensor gives out 4.5 volts at full load capacity. The maximum achievable flow velocity in the wind tunnel is 20 meters per second and this corresponds to a differential pressure of 245 Pa from Bernoulli's Principle [7]. At 245 Pa the MPXV7002DP pressure transducers gives an analog signal with a potential difference of 250 mV. The Arduino's 10-bit ADC when provided with an excitation of 5 volts has a resolution of 4.88mv per bit. This resolution is good enough to read the velocity in the wind tunnel without losing much detail. In the later part of the project, this signal was sent to a 16-bit ADC to allow for precise velocity monitoring. This is discussed in Chapter 4. An acrylic enclosure was later manufactured for the MPX7002DP sensor and this is shown in Chapter 5. [25]

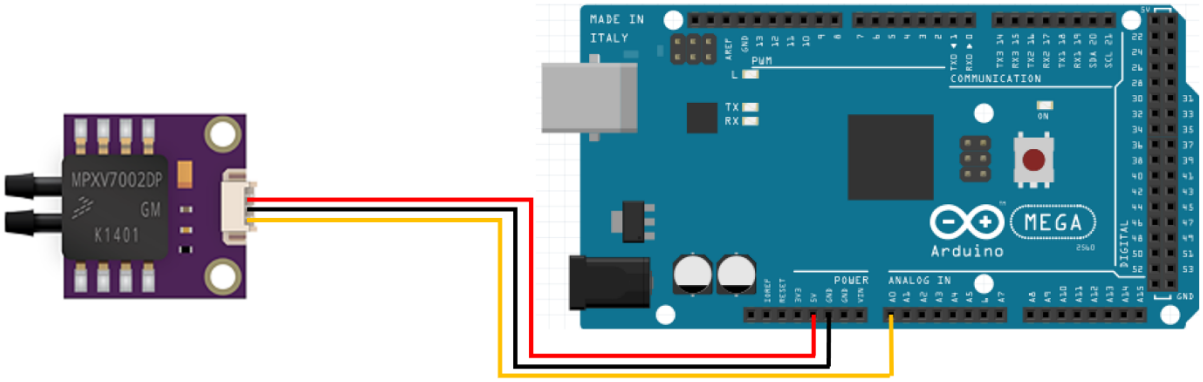


Figure 18: MPXV7002DP wiring

The wiring diagram for this pressure sensor is shown in *Figure18.* The red and black wires are the excitation and the yellow wire is the signal.

3.3 Reading RPM with Arduino

The gasoline engines tested in the wind tunnel usually have a hall effect sensor attached to the crankshaft, as shown in *Figure19*, giving out a pulse signal that can be used to calculate the RPM of the propeller.

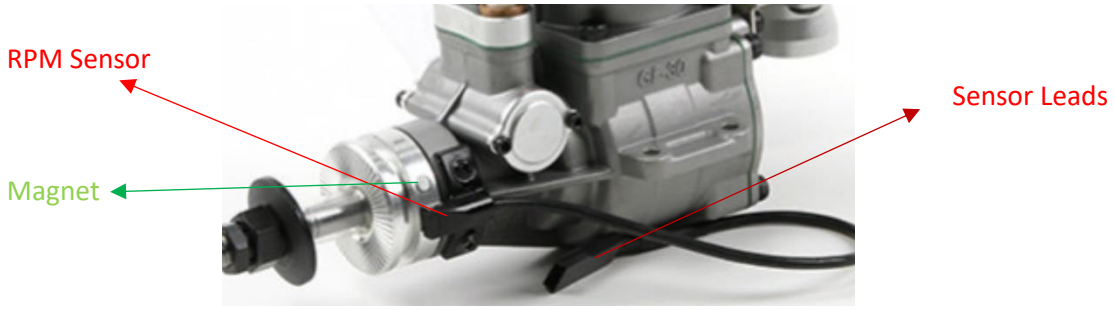


Figure 19: Hall effect sensor on a GF30 engine from Ogawa Shigeo [26].

The battery-operated propellers do not have RPM signal output. Students hold a laser tachometer pointed onto the reflective sticker attached to one of the propeller's blades to measure the RPM for battery-powered propellers. This creates inconvenience as they might have to record other data at the same time. Thus, a simple laser tachometer as illustrated in *Figure20* was developed and installed in the wind tunnel.

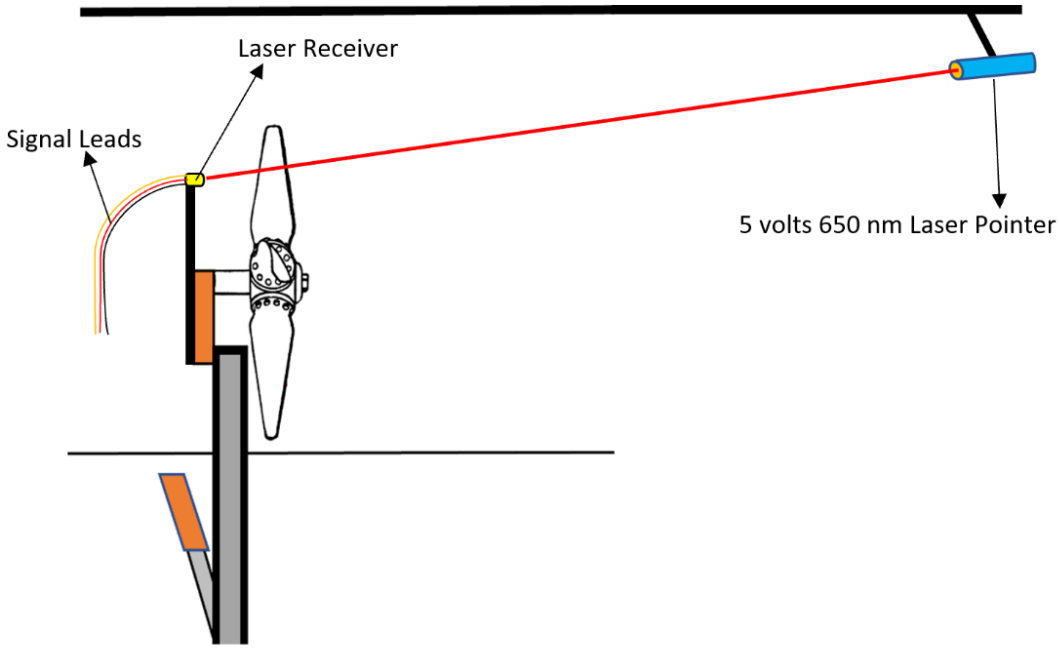


Figure 20: Overview of the installed tachometer system.

A 5V 650 nm laser pointer is pointed onto a laser receiver module through the rotating propeller. This laser tachometer can be interfaced with the Arduino system along with the other sensors. The laser receiver generates a high-level digital signal whenever the laser beam hits it. The digital level signal is read by the Arduino system on a specific pin, as shown in the *Figure21*, to give the frequency. The frequency measured must be divided by the number of propeller's blades to calculate the RPM.

The code implemented in the Arduino system can read frequencies very accurately from 0.1 to 1000Hz. No amplification for this signal is needed as the input signal is not a sine wave or a tiny AC signal. The signal measured by the Arduino was very clean that there was no need for a low pass filter.

This code uses a timer interrupt for the frequency measurement. The other programs that measure Thrust, Torque and Windspeed often delayed the response of this timer. This often led to counting more cycles in one gate interval. To avoid inaccuracies in measuring a propeller's RPM, an Arduino Nano was used to run the timer interrupt to calculate the frequency and sent the data to the Arduino Mega via Serial Communication. The red and black wires are for the excitation voltage of 5V.

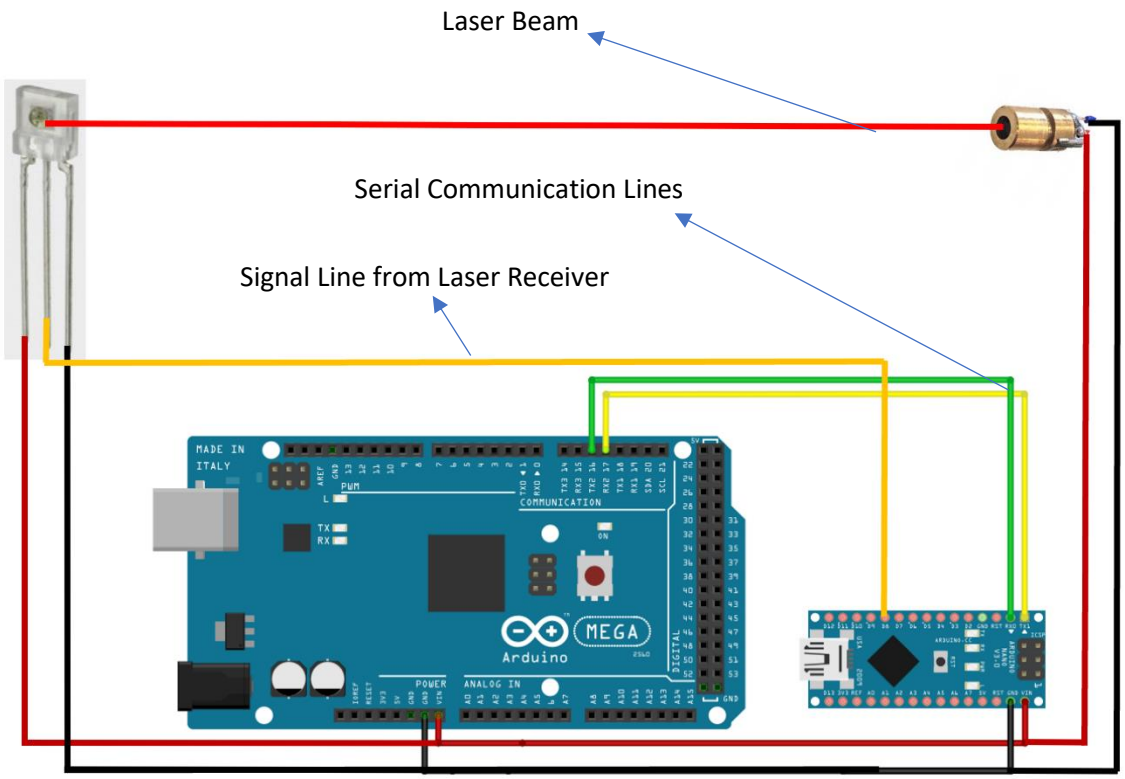
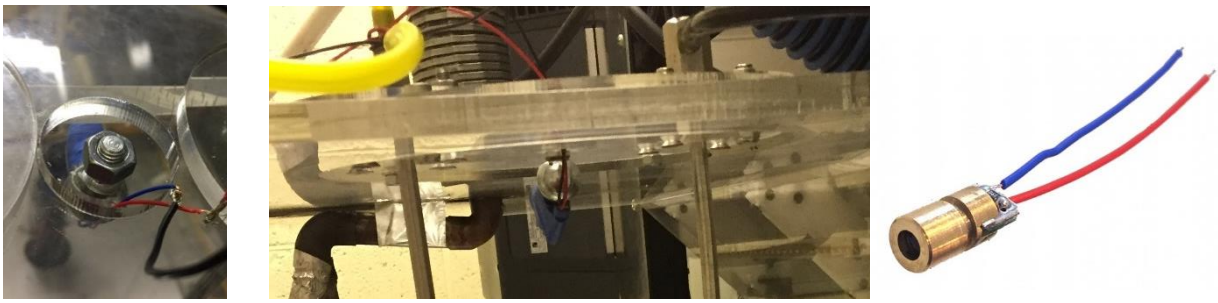
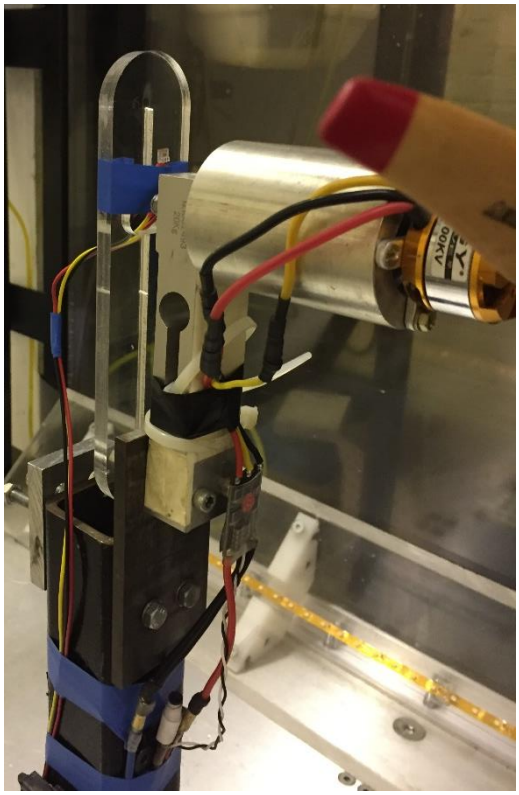


Figure 21: Overview of the installed tachometer system wiring. [27] [28]

The red laser pointer shown in *Figure22(c)* is secured to an L-type ball and socket joint screwed to the new acrylic pitot tube holder as shown in *Figures22(a)&(b).* The laser receiver diode is attached to the custom acrylic holder that screws to the test bench behind the load cell as shown in *Figure23.* The ball and socket joint that holds the laser pointer can move up to 18 degrees vertically and 360 degrees horizontally. This configuration allows for testing various propeller sizes.



Figures 22(a), 22(b) 22(c): Top View Laser holder (left); Laser Holder with Pitot tubes(middle). Laser Pointer [27] (right).



Figures 23: Laser receiver taped to its holder.

The laser diode is attached to its holder with tape so that the position of it can be changed for other propellers if needed.

3.4 Datalogging

The Arduino system was successful in reading data from all the sensors and displaying it on its serial monitor at the same time.

Figure 24: Overview of the Adafruit GPS+Logging Shield [24].

The GPS Logging Shield shown in *Figure24* was inserted into the Arduino Mega as shown in *Figure25* to add datalogging functionality to the existing system. A simple SD card module with just datalogging functionality could have been purchased but, the GPS+Logging module shown in the above picture was already available. This module also communicates with the Arduino system via Serial communication as seen in *Figure25*.

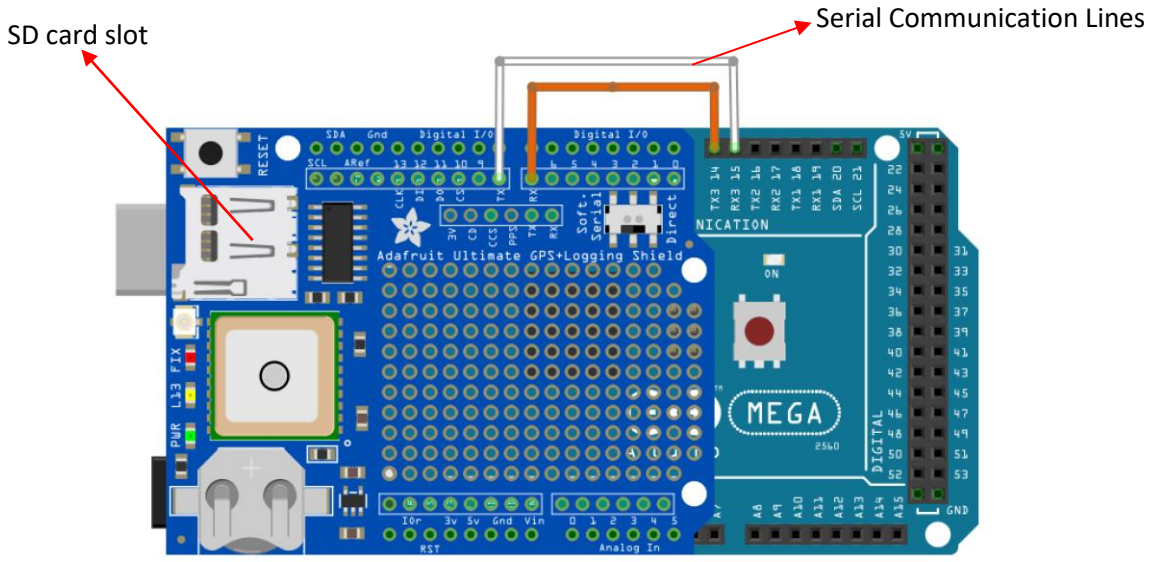


Figure 25: Circuit wiring diagram for SD module

A comma-separated values (.csv) file like the one shown in *Table1* is created on the microSD card and data from all the sensors is stored in it. Pushing the reset button on the Arduino Mega or restarting the entire device creates a new .csv file to record data. The process of recording data from testing a propeller in the wind tunnel is now much easier compared to what it was before. The DAQ system is further developed to log Electric Power consumed by DC motors and Fuel flow rate consumed by IC engines. More detail on this is provided in Chapter 4.

THRUST	TORQUE	WIND SPEED	RPM
0.22	0.04	0	59
0.22	0.04	0	59
0.22	0.04	0	59
0.22	0.04	0	59
0.22	0.07	0	59
0.22	0.08	0	59
0.22	0.07	0	59
0.22	0.07	0	59
0.22	0.08	0	59
0.22	0.08	0	59
0.14	0.05	0	5
0.21	0.04	0	423
0.15	0.03	0	301
0.19	0.04	0	264
0.2	0.04	0	306
0.21	0.04	0	273
0.19	0.04	0	215
0.24	0.04	0	264
0.26	0.04	0	334

Table 1: Spreadsheet created by the DAQ system.

The force readings in the above *Table1* are in Newtons. The wind speed is shown in meters per second. The Torque readings shown in the above table is just force experienced by the Torque load cell. To get the actual torque generated by a propeller the readings in the Torque column must be multiplied by the vertical distance from the Thrust loadcell to the base over the horizontal distance from the Torque load cell to the Thrust loadcell. This Chapter just details the progress made from October 1, 2018 to January 1, 2019. The Arduino system is later developed to allow the users to enter the relevant distances to measure the actual Torque. More detail on this is provided in Chapter 5.

The Arduino system's circuitry developed until January 1st, 2019 is shown below in *Figure26.*

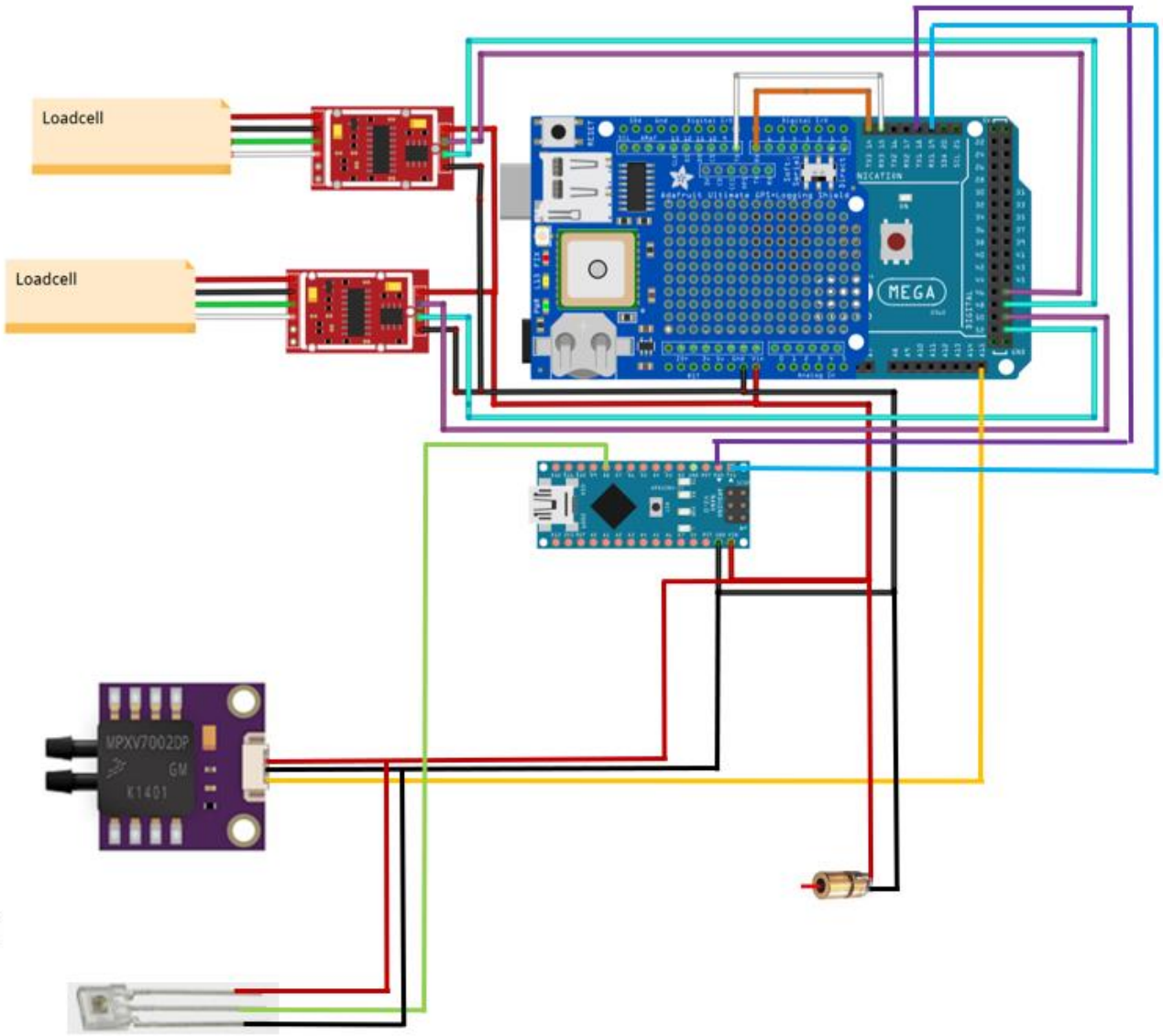


Figure 26: Circuit Diagram of the DAQ system. [27] [28]

As mentioned earlier the DAQ system records enough data to measure the performance of a propeller but not the motor attached with it. The next chapter provides detail on developing the system to measure a motor's performance.

Chapter 4

Development of DAQ System Part 2

4.1 Reading Current drawn by a DC motor.

The DC motors attached to the propellers usually draw power from a LiPo battery pack. A lithium polymer battery is a rechargeable battery that has very high specific energy than other battery types and is usually used in RC aircrafts, where weight and size are a critical design feature. The propeller motors come with an ESC that must be matched to an appropriate battery pack based on its Voltage and Current rating. If an ESC is rated 6-14V, this means that we can use a 2S or a 3S LiPo battery pack. A single LiPo cell has a voltage of 3.7 volts. If the battery is rated 2S it means it has 2 cells in series. The battery capacity and 'C' rating of a battery pack must meet the motor's current draw and the run time requirements. If a motor is drawing 15 amps of current and you need to run the test for 20 minutes, then a LiPo with at least 5Ah capacity is needed. This can be calculated from the *Equation 4.1*. [30]

$$Capacity(Ah) = Current\ draw(A) * \frac{runtime(minutes)}{60} \qquad (4.1)$$

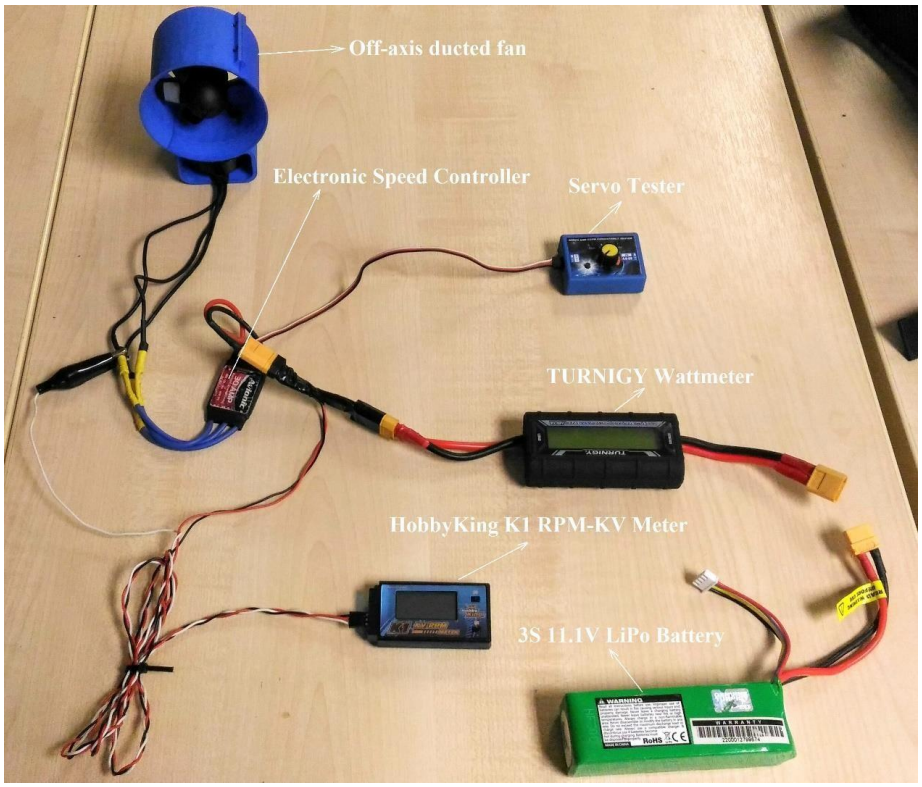


Figure 27: Typical Motor Circuit Layout [1].

As shown in *Figure27,* the motor draws power through a Power Analyzer, shown in *Figure 28,* which measures the Voltage and Current drawn from the LiPo battery pack.

Figure 28: Power Analyser.

To read the Current drawn by the motor with an Arduino, experimenting with a current sensing module called ACS712 was done. The sensor gives an analog signal with potential difference directly proportional to the current detected. This module can only detect AC/DC currents up to 30A. But the DC motors that can be tested in the test rig can sometimes draw in excess of 250A of current for brief amounts of time. Every motor has a burst current rating and is defined as the maximum current a motor can handle for something like 20 seconds followed by a cool down session. No Arduino sensor was readily available in the market that can handle this much current. [31]

Figure 29: Current Shunt Resistor. [32]

So, a 300A DC shunt resistor shown in *Figure29* was purchased to read the current drawn with an Arduino. The current shunt was connected in parallel to the motor circuit. This current shunt gives out an analog signal with a potential difference of 75mV at full load capacity. It works just like the ACS712 but can handle much higher currents. The Arduino's 10-bit ADC has a resolution of 4.88mV per bit and would lose a lot of detail if the current shunt resistor is directly interfaced with it. To increase the resolution, a 16-bit ADC breakout board module called ADS1115[33] was purchased. This ADC provided a resolution of 0.0078125mV per bit at gain set to 16 and it can measure up to plus or minus 256mV. The voltage signal from the current shunt is sent to this ADC for amplification and conditioning. The circuit diagram for interfacing the current shunt is shown

in *Figure30.*

The ADC has 4 channels and the current shunt requires two channels since the signal is differential. The signal coming from the differential pressure sensor (MPXV7002DP) for airflow velocity measurement is single ended and requires only one channel. The Arduino's ADC provided decent resolution, but this signal is now sent to this ADC for further amplification. This means we now have a much better resolution for measuring the wind speed.

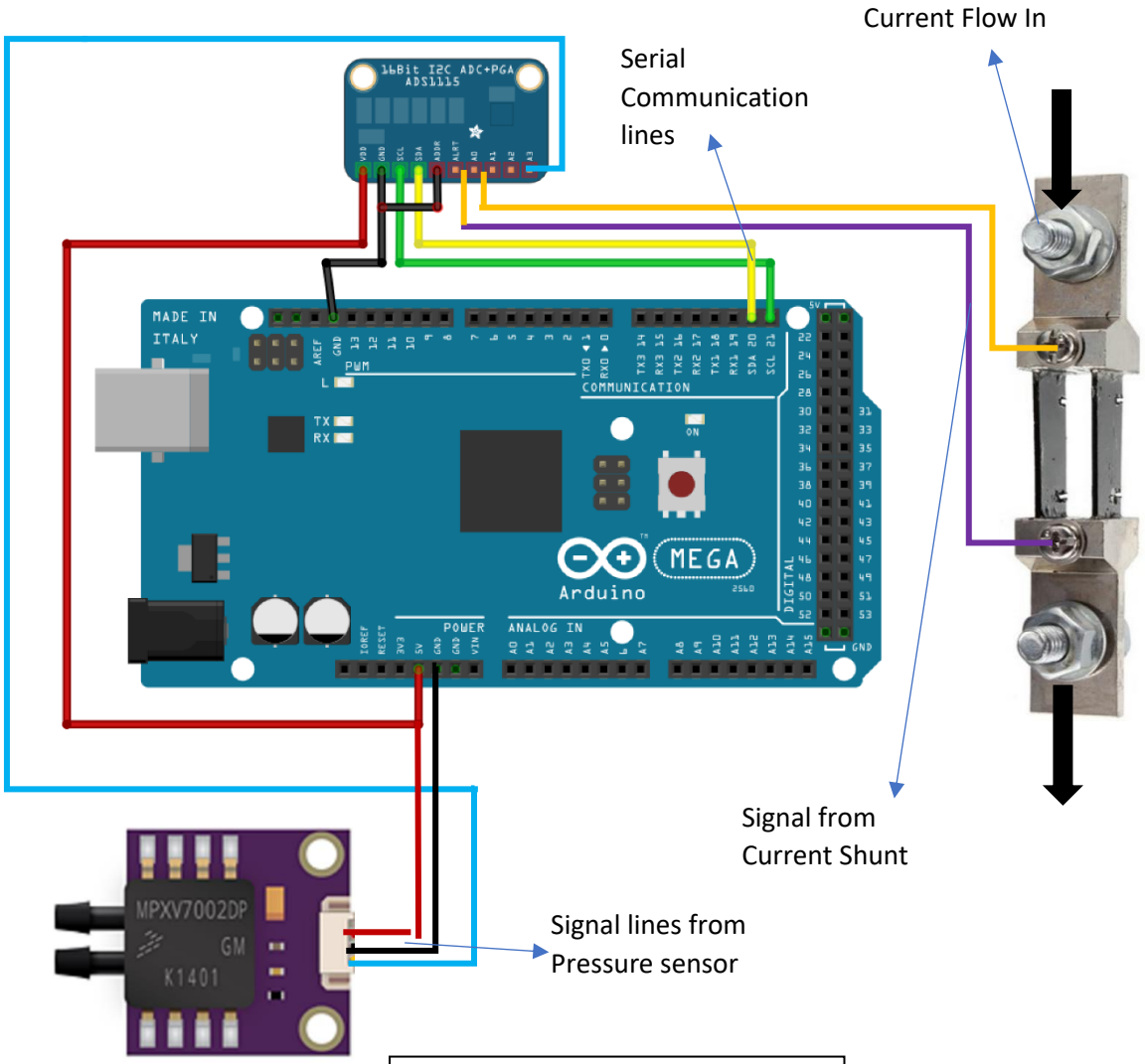


Figure 30: Current shunt Wiring [32].

This resistor is enclosed in an acrylic box as shown in *Figure31* which is manufactured at the EDMC in University of Southampton. This acrylic case has holes for all the four wires that attach the current shunt. The Electronic Workshop in the University had wires only capable of handling currents up to 30A. A copper wire that is 177.3mm^2 in diameter is required to carry currents up to 310A [34]. Car battery jumper cables were purchased to carry these high currents. These cables can carry currents up to 600A and it was cheaper than purchasing 2 meters of 300kcmil wire from a

retail store.

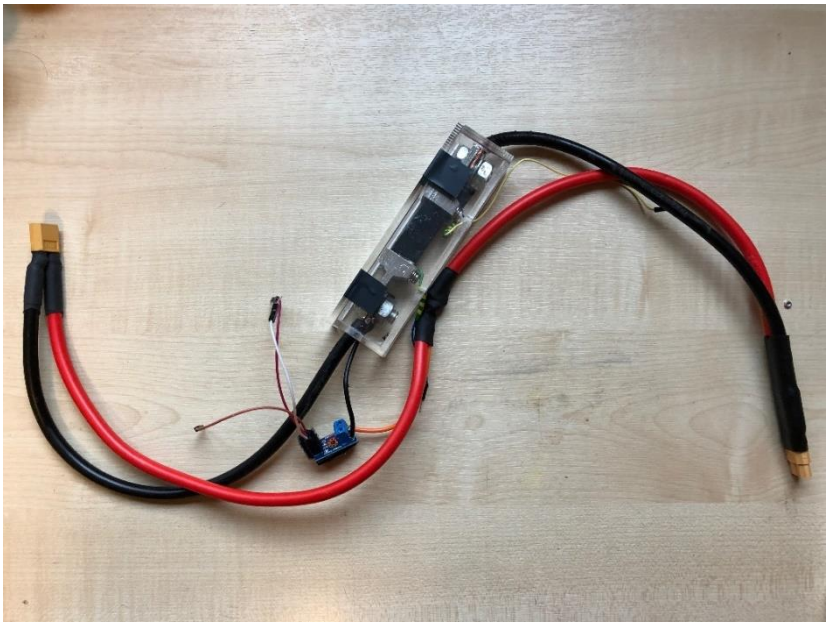


Figure 31: Current shunt resistor with XT60 connectors.

XT60 male and female connectors are soldered to the wire ends, as shown in *Figure27*, as most LiPo batteries come with them. The entire circuit shown in the above is now insulated and safe to go into an Aluminum enclosure box. The current shunt's case is made from gluing 4mm acrylic sheets. The top sheet of the holder is just taped to the rest of the holder so that the resistor can be swapped easily with a new one if it goes bad.

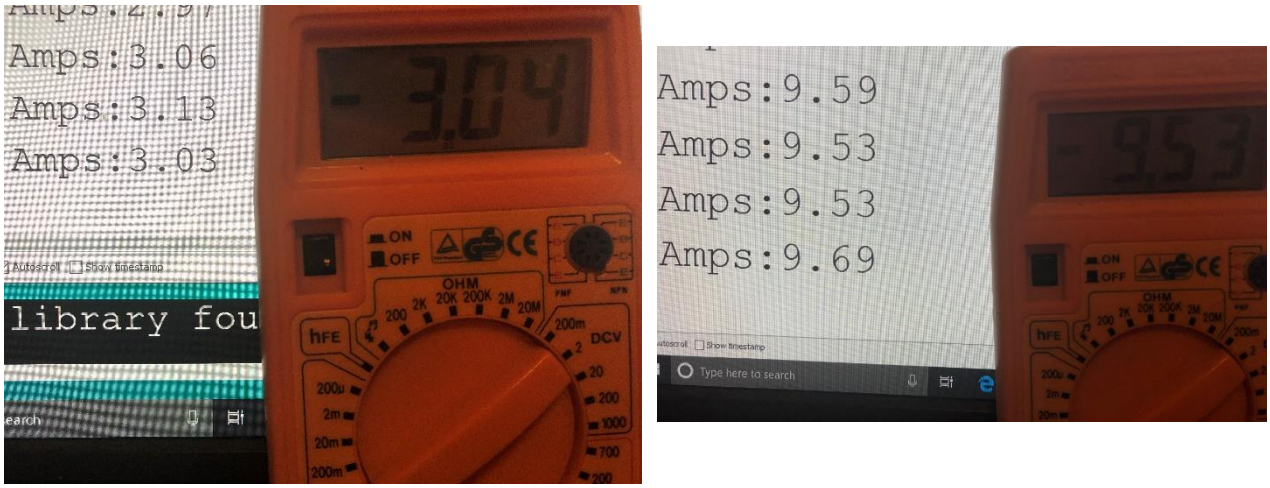


Figure 32: Comparison of Current draw readings with a multimeter.

The readings shown in the Arduino's serial monitor *(Figure32)* were refreshed faster than the values shown on the multimeter. This explains why the readings on the serial monitor are off by a small value.

4.2 Reading Voltage with Arduino

The Arduino's analog signal input is limited only to 5.5 volts. Using just the Arduino to measure the LiPO's voltage would limit the testing to just 1S batteries. To measure higher voltages, we need some other means. In this project we use a five: one voltage divider module connected in series as shown in Figure33(a) and this means the maximum voltage we can measure now is 27.5 V. This limits the testing from 1S to 7S batteries. The voltage divider can be easily swapped with a different one or the signal can be sent to the ADC to measure much higher voltages. The Voltage divider's wiring diagram can be seen in *Figure33(b).*

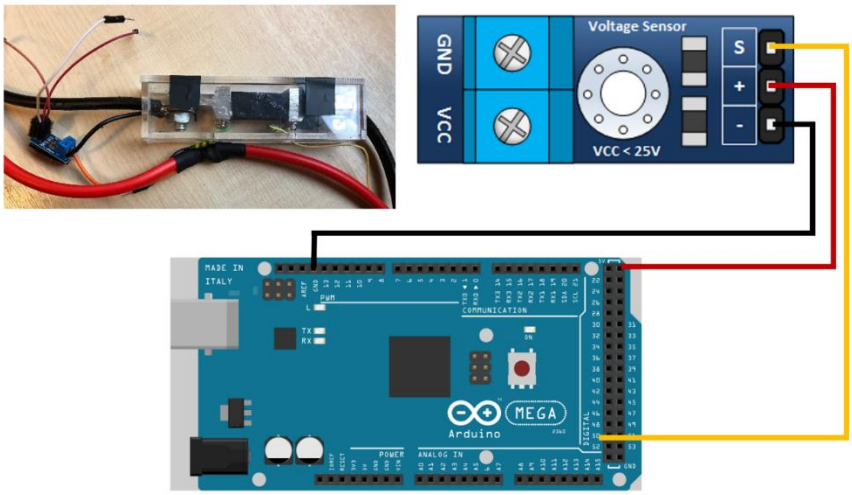


Figure 33(a) & (b): Voltage Divider connected in Series (top left); Voltage Divider wiring (right).

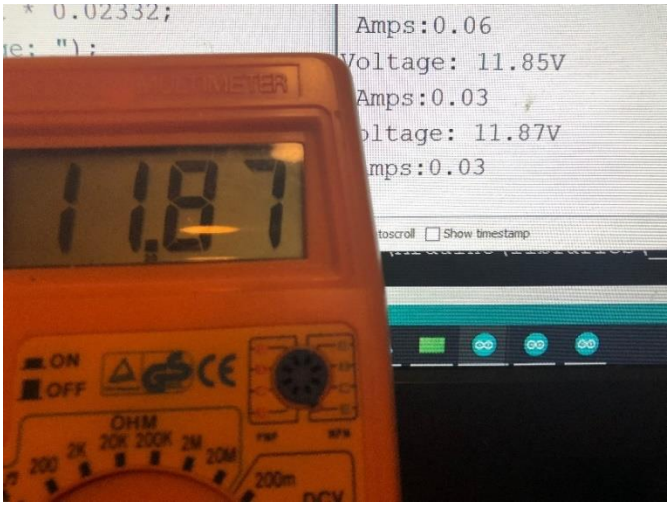


Figure 34: Comparison of Voltage readings.

Figure34 compares the voltage readings shown on the Arduino's serial monitor with the values shown on a multimeter.

4.3 Reading fuel flow rate with Arduino.

The fuel flow consumed by an IC engine is currently measured by a turbine hall effect flow sensor shown in *Figure35(b)*. The sensor installed in the wind tunnel gives out 14000 pulses for every liter of fluid passed through it. Before interfacing this sensor with the Arduino, a cheap water flow sensor was purchased and interfaced with the Arduino system for the initial testing. This cheaper sensor shown in *Figure35(a)* gave 540 pulses per liter of fluid passed through it. [35]

Figure 35(a)&(b): Water flow sensor(left) [35]; Fuel flow sensor in the wind tunnel (right).

If the sensor from *Figure35(b)* sensor gives out 7000 pulses in a second during a test, this means the IC engine consumed 500mL of fuel in a second. The wiring for the fuel flow sensor is similar to the water flow sensor's wiring seen in *Figure36.*

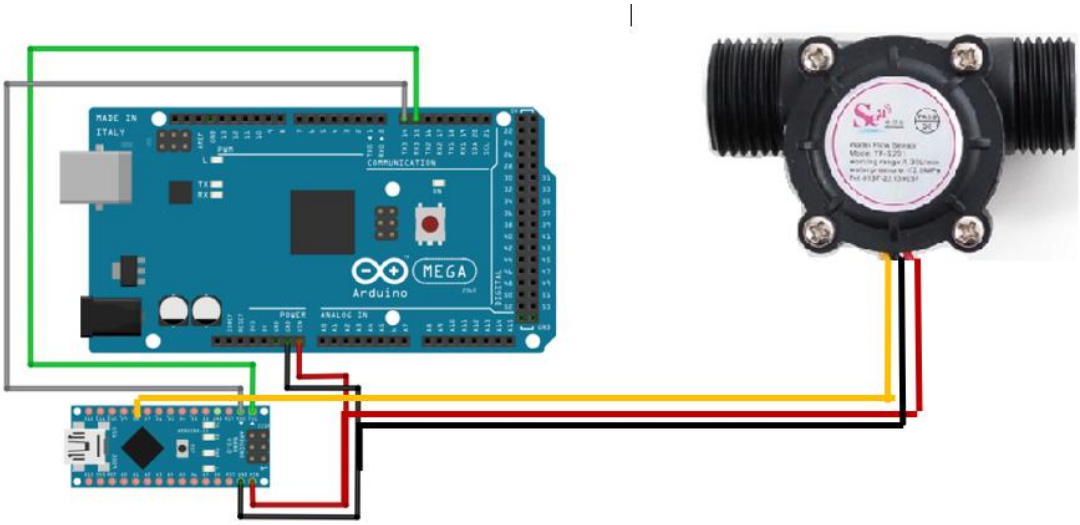


Figure 36: Circuit Schematic for Interfacing the Flow sensor. [35]

This code also uses a timer interrupt for frequency measurement. To avoid latency and inaccuracies in measurements, this code was implemented in another Arduino Nano and the data was sent over Serial Communication as seen in *Figure36*.

Chapter 5

Development of DAQ System Part 3

5.1 Nextion Display

The 3.2” TFT touch screen display is added to the system to calibrate the sensors easily. This display communicates with the Arduino system via Serial communication at a 115200 baud rate. The wiring diagram for the Nextion display can be seen in *Figure37*.

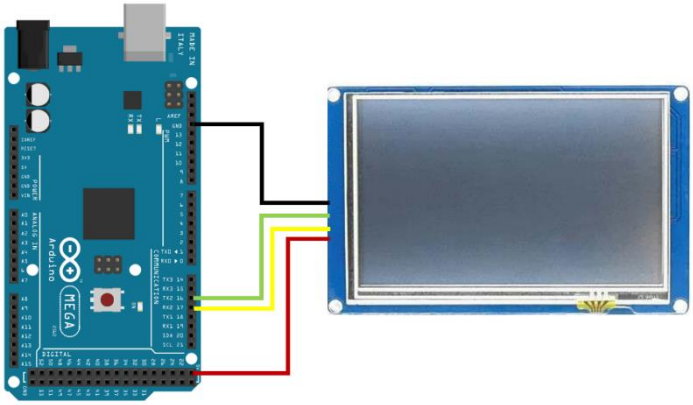


Figure 37: Nextion Enhanced 3.2” Display. [36]

5.2 Graphic User Interface

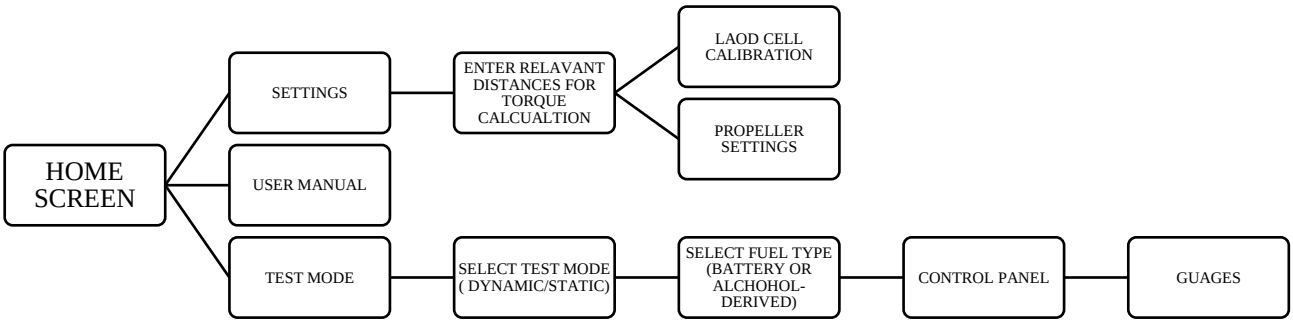
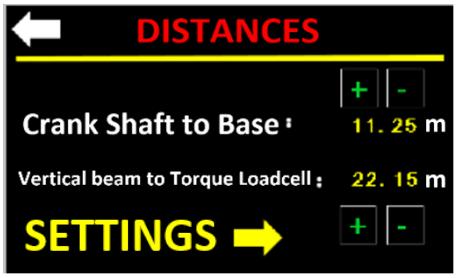


Figure 38: Flow chart showing the page order hierarchy.

Figure38 gives an overview of the designed GUI and the page order hierarchy. The GUI of this device was designed using a software called the Nextion Editor. This software comes with mass components such as icons, buttons, progress bars and sliders. Custom gauges and control panels are designed using these components to enrich the data monitoring experience. The device lets the user

to calibrate the loadcells, define the test type (i.e. dynamic or static) and enter data relevant to a test such as propeller's blade count and diameter.

Figure 39: Page-0: Home screen.[37][38][39]



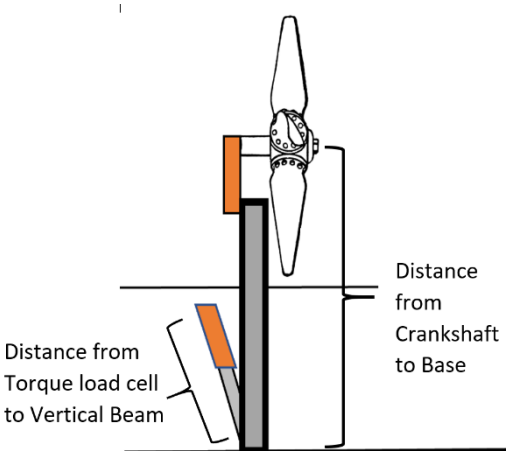


Figure 40(a) & (b): Page-1: Distances page; Visual Representation of relevant distances (right)

Pushing the settings button on the home screen shown in *Figure39* takes the user to the distances page shown in *Figure40(a)* before taking him/her to the actual settings page. Here the user must enter relevant distances for Torque calculation. Since the torque sensor in this test bench is not directly attached to the crank shaft of the engines, the user must enter the distance from the crankshaft to the base and the distance from the torque load cell to the vertical beam and these distances are illustrated in *Figure 40(b).*

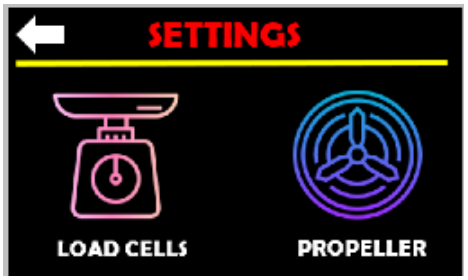
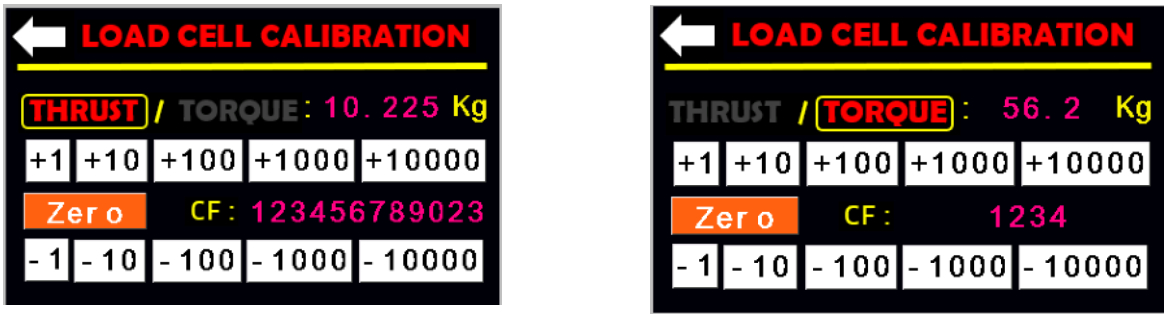


Figure 41: Page-2: Settings page. [40][41]

The yellow settings button on page-1 shown in *Figure40(a)* takes the user to actual settings page and this page has two buttons as seen in *Figure41*.

In this page the user can either choose to calibrate the load cells or enter relevant propeller data. The two 20Kg loadcells used in this project are calibrated using this HMI device. The calibration factor might have to be changed periodically.

Pushing the loadcells button on page-2 shown in *Figure41* takes the user to the calibration page and here he/she can change the calibration factor for both the load cells. The Thrust load cell is highlighted and selected by default as seen in *Figure42(a)* and pushing the Thrust/Torque button selects the other load cell for calibration. The orange zero button resets the displayed weight from the selected loadcell to zero. Pushing a white colored numbered button adds/subtracts the calibration factor for the highlighted loadcell with the value shown on it.



Figures 42(a)&(b): Thrust Loadcell Calibration (left); Torque Load cell calibration (right).

The Propeller button on page-2 shown in *Figure41* leads to the Propeller settings page seen in *Figure43* where the user must enter the propeller's blade count and diameter. The blade count is essential for RPM calculation and the diameter is used for calculating the advance ratio and efficiency for dynamic test modes (i.e. when the inlet velocity of the wind tunnel is nonzero).

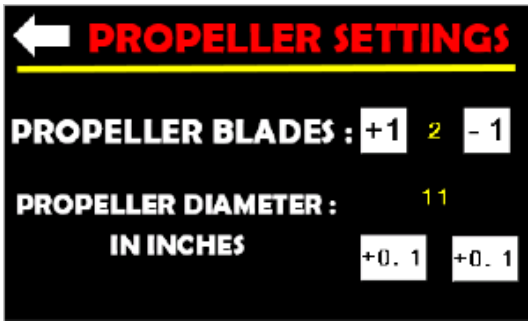


Figure 43: Page-4: Propeller settings page.

"All propellers are usually described by two numbers, such as 18 X 9. Those numbers are the diameter and pitch. Diameter is the overall size of the propeller and the pitch is defined as the distance the propeller would travel in one revolution if it were travelling through a fluid that doesn't allow slip".[53] All these distances are measured in inches.

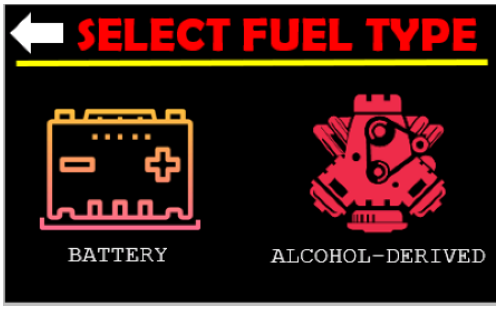


Figure 44: Page-5: Fuel type page.[42][43]

Pushing the test mode button from the home screen shown in *Figure39* takes the user to Page 5 where he/she must define the motor type (i.e. Battery powered, or Alcohol-Derived fuel powered). This lets the Arduino know if it should monitor the electrical power consumption of a DC motor or the fuel flow rate consumed by an IC engine.

After selecting the fuel type the user is taken to Page 6 shown in *Figure45* where the propeller's test environment is selected.

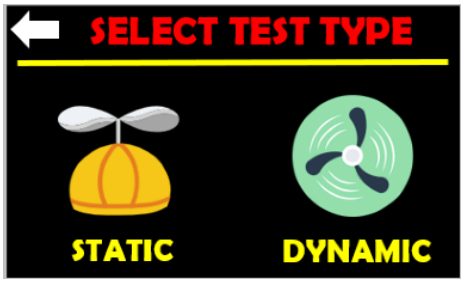
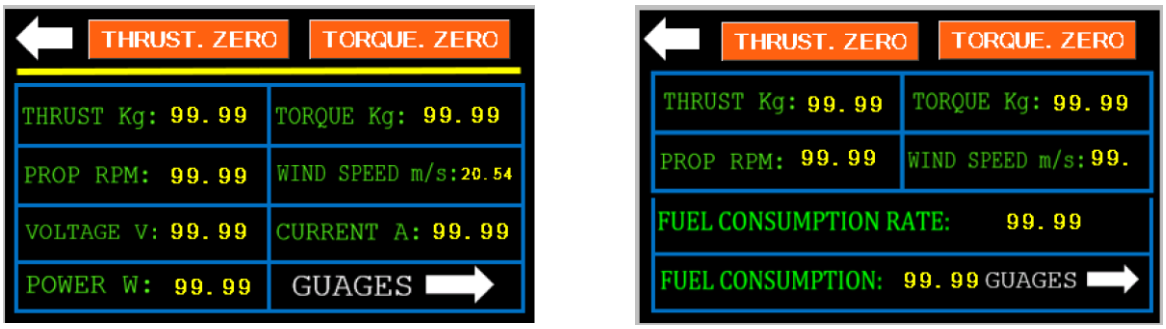


Figure 45: Page-6: Test type. [44][45]

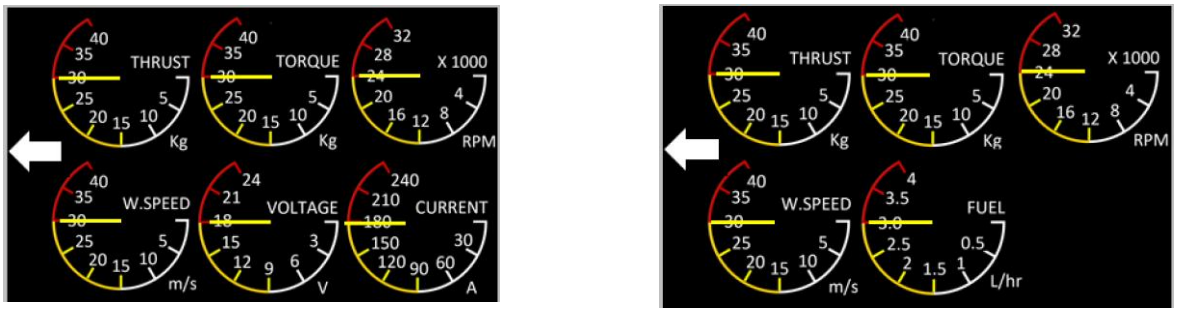
Depending on the motor type selected in page-5 the DAQ system takes the user to an instrumentation panel. The two panels designed in the Nextion editor can be seen *in Figures 46(a) &(b).*



Figures 46(a) & (b): Instrumentation panel for DC motors (left); Instrumentation panel for IC engines (right).

These instrumentation panels make data monitoring and validation of the readings against the existing sensor readouts easy. The buttons THRUST.ZERO and TORQUE.ZERO from *Figures 46(a)&(b)* reset the respective loadcell force reading to zero. The Thrust and Torque readings in these pages are shown in Kgs for verification against the values shown in the A12 readouts.

The GAUGES button at the bottom right of data monitoring instrumentation panels from *Figures46(a)&(b)* takes the user to the relevant gauge page *(Figure47(a)&(b))* shown below.



Figures 47(a)&(b): Gauges for DC motor testing (left); Gauges for IC engine testing (right).

5.3 Enclosure

The entire circuit diagram can be seen below in *Figure48.*

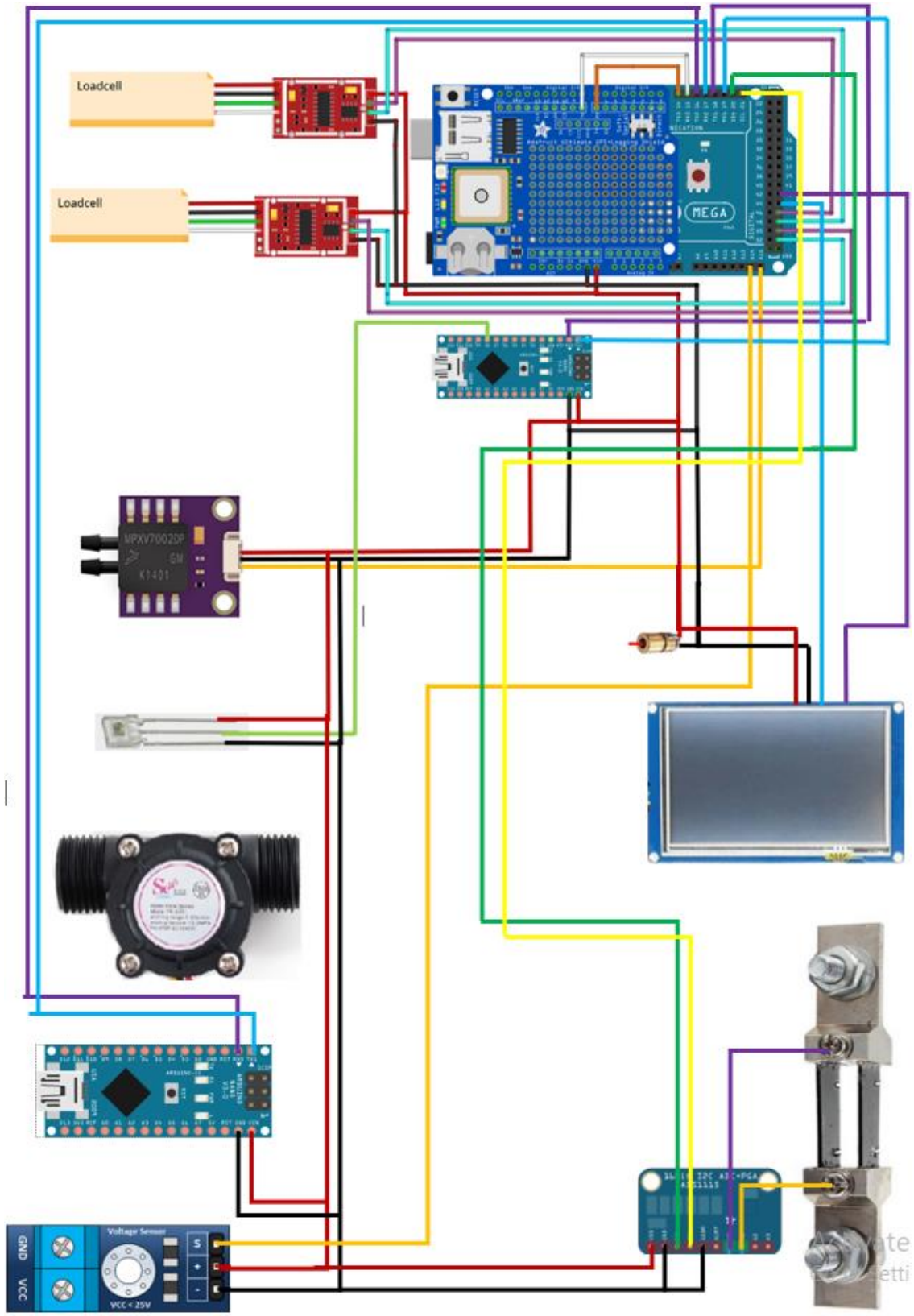


Figure 48: DAQ System Circuit Diagram. [27][28][32][35][36]

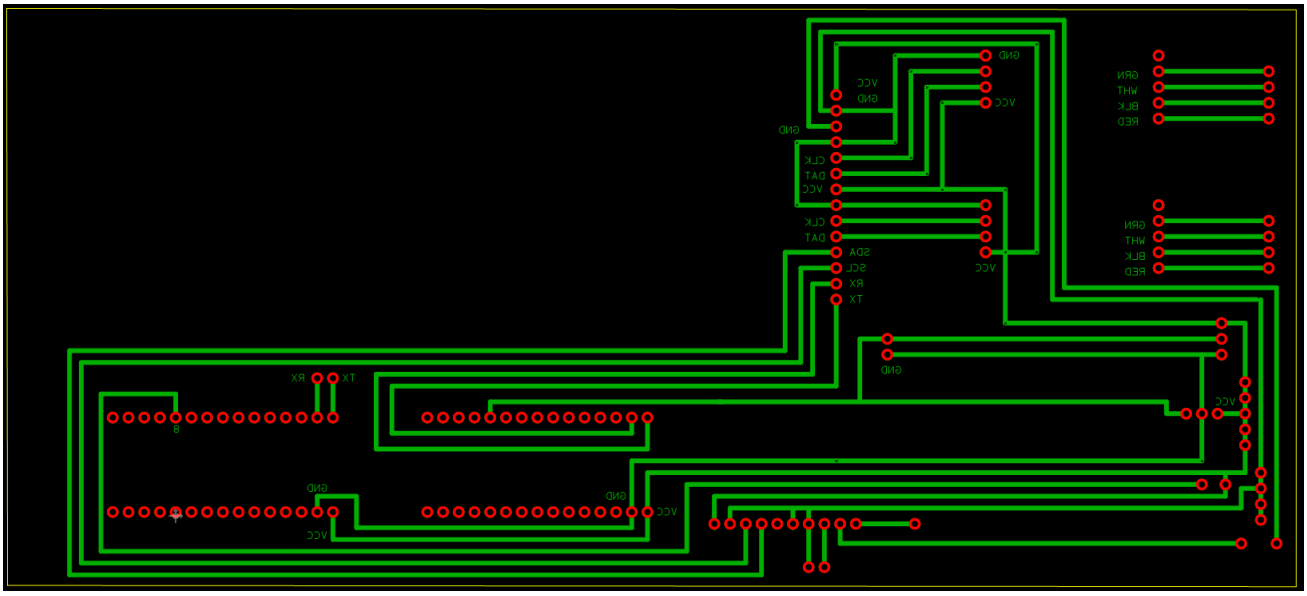


Figure 49: Custom PCB

The PCB shown above in *Figure49* was designed and manufactured to hold all the components together and reduce cabling in the enclosure box.

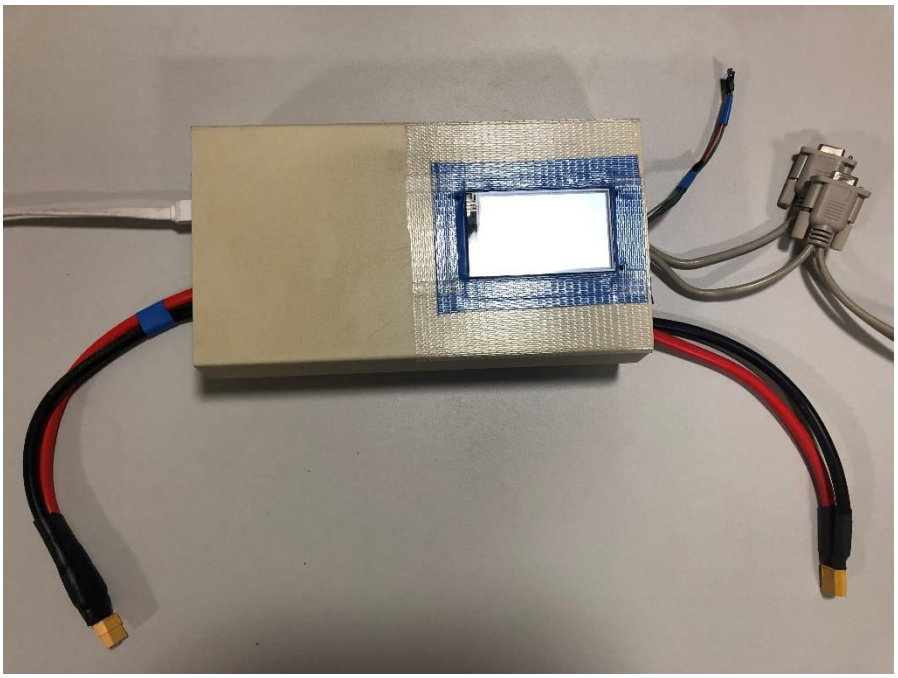


Figure 50: Aluminium Enclosure Box.

All the Arduinos, sensors and the PCB go into the aluminum enclosure box shown in *Figure50* and this box protects the components from EMI, high temperatures and RFI. The HMI touch screen

device is fixed to the enclosure as shown above.

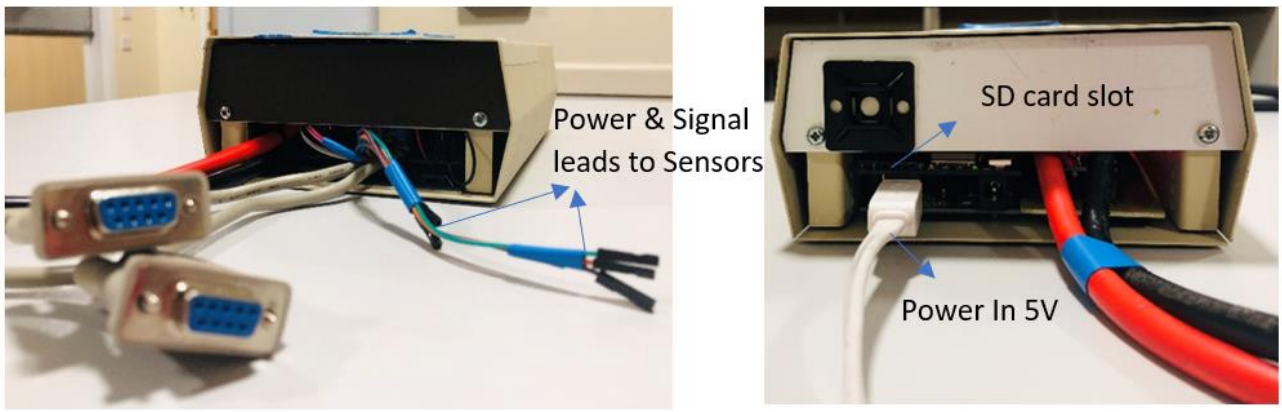


Figure 51(a)&(b): Right view of the enclosure(left).; Left view of the enclosure(right).

As shown in *Figure51(b)*, the SD card can be easily inserted/removed from the Arduino. The white cable shown in *Figure51(b)* powers the entire system and a constant voltage of 5V and a minimum current of 1.5A should always be supplied.

Chapter 6

Testing and Validation

6.1 Testing a DC Motor with Propeller

A 11 x 5 propeller is attached to a 1100KV Brushless Motor and connected to 30A ESC as shown in *Figure52.*

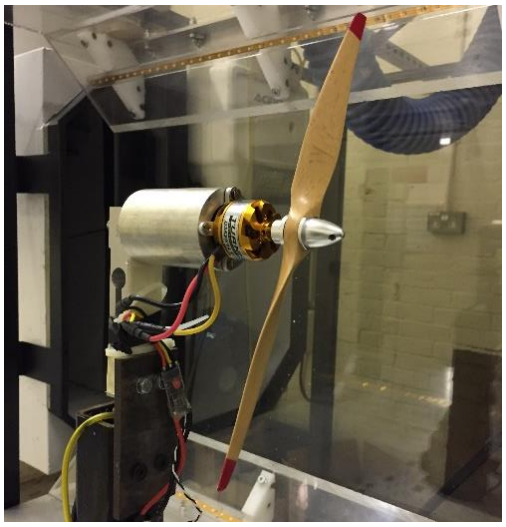


Figure 52: Overview of the mounted propeller.

The motor is powered by a 3S 5Ah LiPo battery pack. The speed of the propeller is supposed to be controlled by the servo consistency tester (*Figure53)* that sends a pulse signal to the ESC but, unfortunately the servo consistency tester in the wind tunnel stopped working.

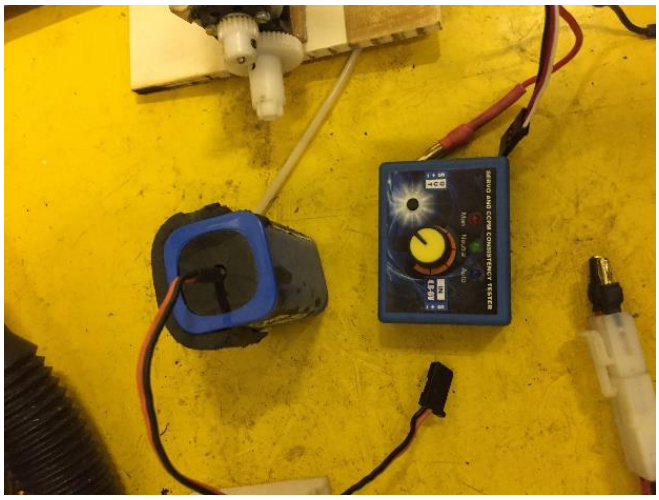


Figure 53: Servo Consistency tester.

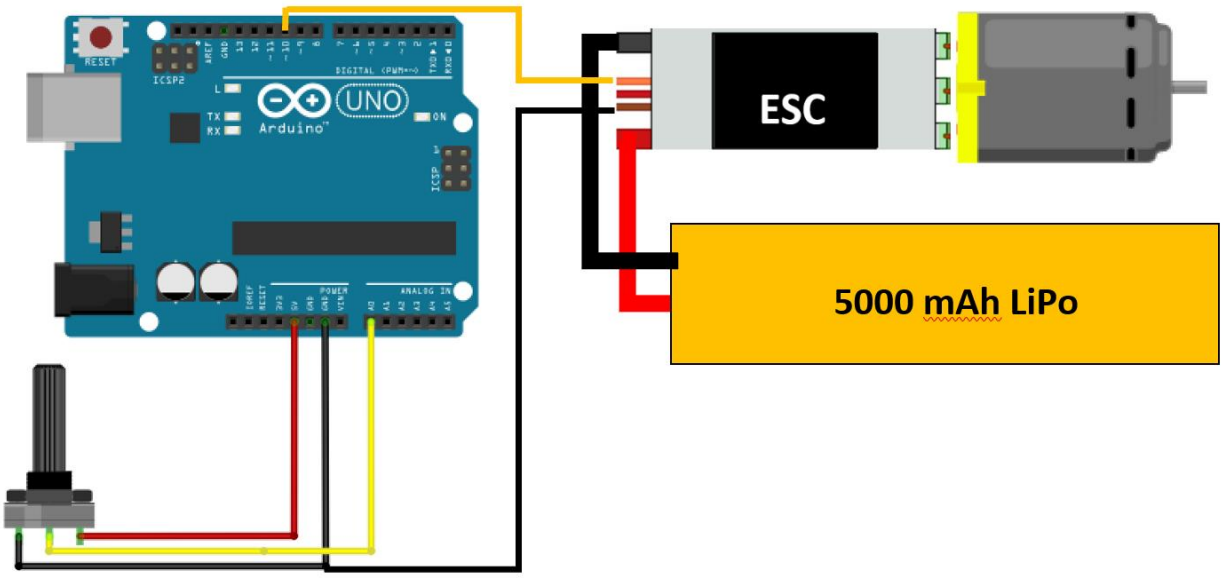


Figure 54: Overview of the Arduino ESC speed controller.

Hence, a potentiometer and an Arduino Uno were used to control the motor's speed as shown in *Figure54*. The analog signal given by the potentiometer is read by the Arduino and this signal dictates the PWM signal going from Pin-10 t to the ESC.

6.1.1 Static Testing

Thrust (N)	Torque (N-m)	RPM	rev/sec	Propeller Power (Watts)	Coefficient of Power	Coefficient of Thrust	Voltage (V)	Current (A)	Electrical Power (W)	Motor Efficiency	Propeller Mech. Efficiency	Overall Efficiency
0.5886	0.017658	2065	34.42	3.818846893	0.044898471	0.066551344	12.24	0.41	5.0184	0.760969013	0.015711549	0.011956002
0.5886	0.017658	2065	34.42	3.818846893	0.044898471	0.066551344	12.27	0.37	4.5399	0.841174231	0.015711549	0.01321615
0.5886	0.017658	2065	34.42	3.818846893	0.044898471	0.066551344	12.29	0.44	5.4076	0.706199958	0.015711549	0.011095495
0.5886	0.017658	2065	34.42	3.818846893	0.044898471	0.066551344	12.29	0.41	5.0389	0.757873126	0.015711549	0.011907361
0.5886	0.017658	2076	34.595	3.838262879	0.044445379	0.065879742	12.29	0.44	5.4076	0.709790458	0.015632072	0.011095495
0.5886	0.017658	2076	34.595	3.838262879	0.044445379	0.065879742	12.29	0.41	5.0389	0.761726345	0.015632072	0.011907361
0.5886	0.017658	2076	34.595	3.838262879	0.044445379	0.065879742	12.27	0.44	5.3988	0.71094741	0.015632072	0.011113581
0.5886	0.017658	2076	34.595	3.838262879	0.044445379	0.065879742	12.27	0.44	5.3988	0.71094741	0.015632072	0.011113581
0.5886	0.017658	2076	34.595	3.838262879	0.044445379	0.065879742	12.24	0.44	5.3856	0.712689928	0.015632072	0.01114082
0.5886	0.017658	2114	35.24	3.909824652	0.042833295	0.06349021	12.24	0.41	5.0184	0.77909785	0.015345957	0.011956002
0.5886	0.017658	2114	35.24	3.909824652	0.042833295	0.06349021	12.29	0.63	7.7427	0.504969152	0.015345957	0.007749235
0.5886	0.017658	2114	35.24	3.909824652	0.042833295	0.06349021	12.27	0.53	6.5031	0.601224747	0.015345957	0.009226369
0.5886	0.017658	2114	35.24	3.909824652	0.042833295	0.06349021	12.2	0.5	6.1	0.640954861	0.015345957	0.009836066
0.5886	0.017658	2117	35.29	3.915372076	0.042712006	0.063310427	12.27	0.56	6.8712	0.569823623	0.015324214	0.008732099
0.5886	0.017658	2135	35.59	3.948656622	0.041994973	0.062247596	12.29	0.56	6.8824	0.573732509	0.015195041	0.008717889
0.5886	0.017658	2174	36.235	4.020218396	0.040513219	0.060051247	12.22	0.44	5.3768	0.747697217	0.014924562	0.011159054
0.5886	0.017658	2174	36.235	4.020218396	0.040513219	0.060051247	12.22	0.41	5.0102	0.802406769	0.014924562	0.01197557
0.5886	0.017658	2174	36.235	4.020218396	0.040513219	0.060051247	12.22	0.47	5.7434	0.699971863	0.014924562	0.010446774
0.5886	0.017658	2176	36.26	4.022992108	0.040457373	0.059968469	12.27	0.41	5.0307	0.799688335	0.014914272	0.01192677
0.5886	0.017658	2176	36.26	4.022992108	0.040457373	0.059968469	12.2	0.44	5.368	0.749439662	0.014914272	0.011177347
0.5886	0.017658	2176	36.26	4.022992108	0.040457373	0.059968469	12.24	0.37	4.5288	0.888313043	0.014914272	0.013248543
0.5886	0.017658	2176	36.26	4.022992108	0.040457373	0.059968469	12.24	0.53	6.4872	0.620143068	0.014914272	0.009248983
0.6867	0.017658	2184	36.405	4.039079638	0.040135734	0.069407001	12.29	0.47	5.7763	0.699250323	0.017330681	0.012118484
0.6867	0.017658	2187	36.445	4.043517578	0.040047681	0.06925473	12.24	0.5	6.12	0.660705487	0.01731166	0.011437908
0.6867	0.017658	2187	36.445	4.043517578	0.040047681	0.06925473	12.29	0.53	6.5137	0.620771233	0.01731166	0.01074658
0.6867	0.017658	2187	36.445	4.043517578	0.040047681	0.06925473	12.27	0.5	6.135	0.65909007	0.01731166	0.011409943
0.6867	0.017658	2187	36.445	4.043517578	0.040047681	0.06925473	12.27	0.53	6.5031	0.621783085	0.01731166	0.010764097

Table 2: Data log file from Static testing a DC Motor with Propeller.

The .csv file created by the Arduino system from static testing a DC motor with propeller is shown in *Table2*. The Arduino system calculates all the parameters shown in the above table and stores them on the micro SD card. All the motor testing in this project is done just to validate the functionality of the system but not to measure a propeller's performance. More than 2500 data points are collected for each test.

In *Table2*, the Propeller Power is given by the *Equation6.1.1*. [47]

$$P = 2\pi nQ \tag{6.1.1}$$

Coefficient of Power is given by the *Equation6.1*.2. [46]

$$\boldsymbol{C_P = \frac{P}{\rho n^3 D^5}} \tag{6.1.2}$$

Coefficient of Thrust is given by the *Equation6.1.3*.[46]

$$C_T = \frac{T}{\rho n^2 D^4} \tag{6.1.3}$$

Motor Efficiency is given by the *Equation6.1.4*.[48]

$$Motor\ Efficiency = \frac{P}{P_{electrical}} \qquad (6.1.4)$$

Propeller Mech. Efficiency is given by the *Equation6.1.5*. [48]

$$Propeller\ Mech.\ Efficiency = \frac{Thrust\ in\ Kg}{P} \qquad (6.1.5)$$

Overall System Efficiency is given by the *Equation6.1.6*. [48]

$$Overall\ System\ Efficiency = Motor\ Efficiency * Propeller\ Mech\ Efficiency \qquad (6.1.6)$$

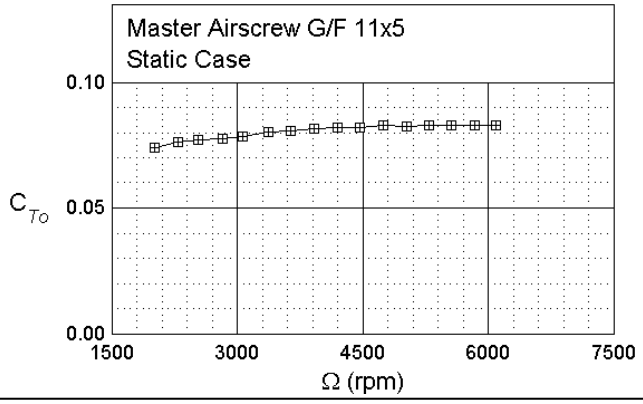


Figure 55: Coefficient of Thrust vs RPM curve of a 11x5 Propeller taken from UIUC database. [49]

Figure55 shows the Coefficient of Thrust vs RPM curve of a 11 X 5 propeller taken from the UIUC database. *Figure56* shows the Coefficient of Thrust vs RPM curve of a wooden propeller of the same size and pitch (11X5) tested in the wind tunnel. The graph shown in *Figure56* is plotted from the data collected by the Arduino DAQ system.

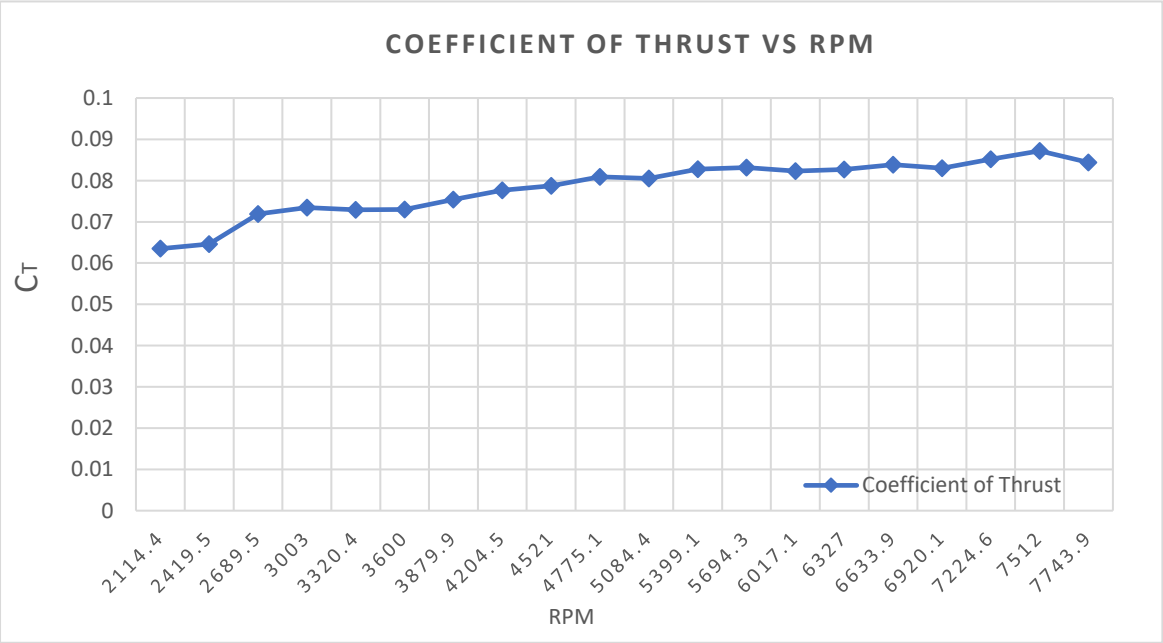


Figure 56: Coefficient of Thrust vs RPM curve of the 11x5 prop tested with the DAQ system.

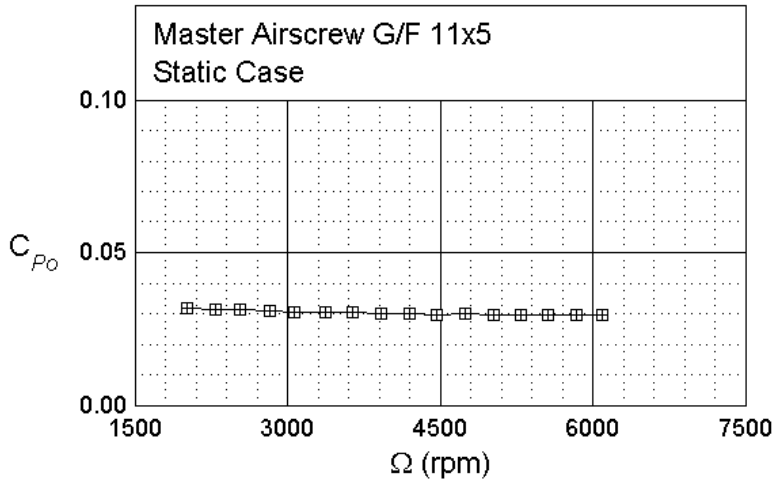


Figure 57: Coefficient of Power vs RPM curve of a 11x5 Propeller taken from UIUC database. [50]

Figure57 shows the Coefficient of Power vs RPM curve of a 11 X 5 propeller under static testing taken from the UIUC database. *Figure58* shows the Coefficient of Power vs RPM curve of the same wooden propeller (11X5) tested in static conditions in the wind tunnel. The graph shown in *Figure58* is plotted from the data collected by the Arduino DAQ system.

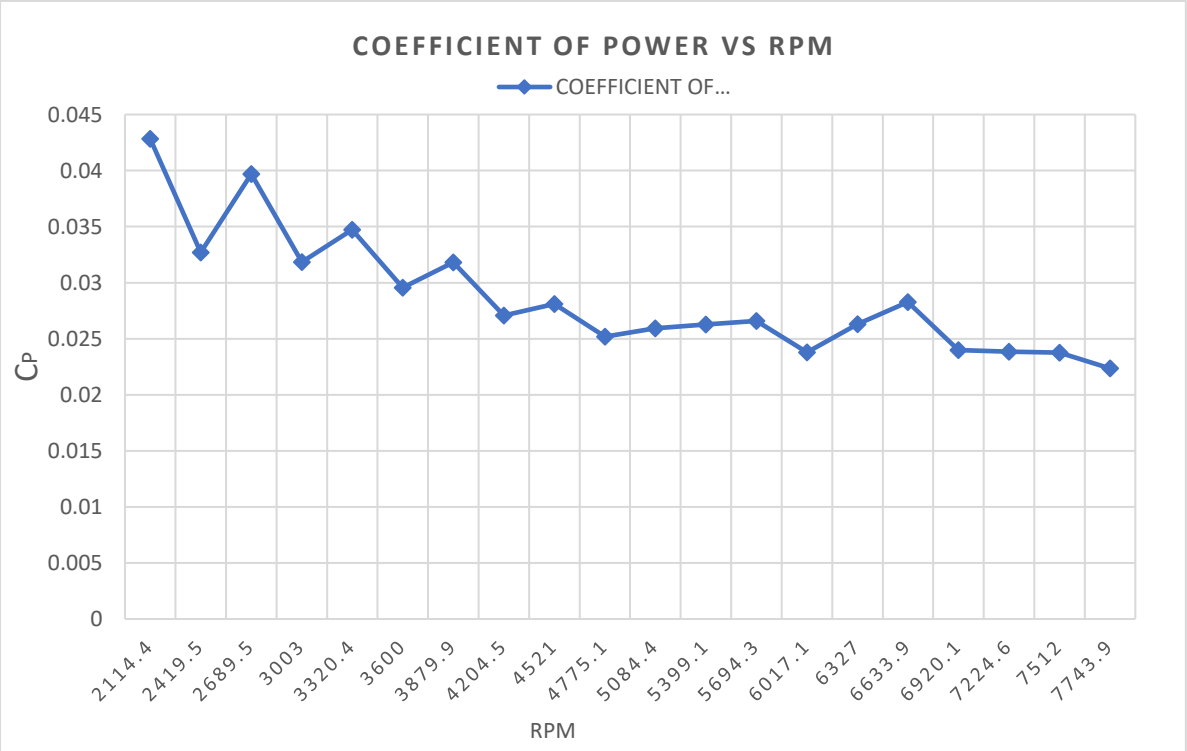


Figure 58: Coefficient of Power vs RPM curve of the 11x5 prop tested with the DAQ system.

6.1.2 Dynamic Testing

Thrust (N)	Torque (N-m)	RPM	rev/sec	Propeller P	Coefficient of Power	Coefficient of Thrust	Voltage (V)	Current (A)	Electrical P	Motor Efficiency	Propeller Efficiency	WINDSPEED (m/s)	ADVANCE RATIO (J)	EFFICIENCY
8.5347	0.123606	6964.8	116.08	90.1523	0.02763354	0.084846077	11.61	15.16	176.0076	0.51220687	0.009650336	4.04	0.124565439	0.382466
8.5347	0.123606	6977.1	116.285	90.31151	0.027536195	0.084547188	11.52	15.22	175.3344	0.515081541	0.009633323	5.71	0.175746227	0.539612
8.5347	0.123606	6986.7	116.445	90.43578	0.027460575	0.084315006	11.33	15.28	173.1224	0.522380554	0.009620087	5.71	0.175504744	0.53887
8.5347	0.132435	7002	116.7	97.10766	0.029293606	0.083946937	11.57	15.31	177.1367	0.548207472	0.008959128	4.04	0.123903652	0.355072
8.4366	0.132435	7018.5	116.975	97.33649	0.029156033	0.082592318	11.38	15.81	179.9178	0.541005358	0.00883533	4.04	0.123612363	0.350165
8.5347	0.132435	7017.6	116.96	97.32401	0.029163512	0.083574126	11.59	15.56	180.3404	0.539668383	0.008939212	4.04	0.123628216	0.354282
8.5347	0.132435	7020.3	117.005	97.36146	0.029141084	0.083509854	11.47	15.56	178.4732	0.545524242	0.008935774	4.04	0.123580669	0.354146
8.5347	0.132435	690	11.5	9.569307	3.016607913	8.644719051	11.96	15.44	184.6624	0.051820549	0.090915675	4.04	1.257352712	3.603206
8.5347	0.132435	7059.3	117.655	97.90233	0.028819986	0.082589681	11.57	15.53	179.6821	0.54486413	0.008886407	0	0	0
7.848	0.132435	7051.5	117.525	97.79416	0.02888378	0.076112639	11.64	15.47	180.0708	0.543087254	0.008180448	4.04	0.123033875	0.324211
1.3734	0.123606	690	11.5	8.931353	2.815500719	1.391104215	11.4	15.47	176.358	0.050643312	0.015675116	4.04	1.257352712	0.621242
8.7309	0.123606	7055.7	117.595	91.32891	0.02692611	0.084574532	11.54	15.84	182.7936	0.499628601	0.009744997	4.04	0.122960638	0.386218
8.6328	0.132435	7085.7	118.095	98.26846	0.028605631	0.082917646	11.57	15.84	183.2688	0.536198529	0.00895506	4.04	0.122440037	0.354911
8.7309	0.132435	7097.7	118.295	98.43488	0.028508986	0.083576569	11.47	15.84	181.6848	0.541789317	0.00904151	4.04	0.122233029	0.358337
8.6328	0.132435	7102.2	118.37	98.49729	0.02847287	0.082532821	11.61	15.84	183.9024	0.53559547	0.008934256	4.04	0.122155582	0.354086
8.7309	0.132435	7096.2	118.27	98.41408	0.02852104	0.083611906	12.1	15.91	192.511	0.511212767	0.009043421	4.04	0.122258867	0.358412
8.7309	0.123606	7110.3	118.505	92.03565	0.026514166	0.083280624	11.57	15.78	182.5746	0.504098886	0.009670166	4.04	0.122016423	0.383252
8.7309	0.123606	7103.4	118.39	91.94634	0.026565701	0.083442494	11.19	15.75	176.2425	0.521703556	0.009679559	4.04	0.122134945	0.383624
8.7309	0.132435	7098.3	118.305	98.4432	0.028504167	0.083562441	11.61	15.88	184.3668	0.533952994	0.009040746	4.04	0.122222697	0.358306
1.3734	0.132435	7087.8	118.13	98.29759	0.028588682	0.013183628	11.38	15.91	181.0558	0.542913207	0.001424247	4.04	0.12240376	0.056446
1.3734	35.254197	690	11.5	2547.35	803.0210264	1.391104215	11.71	15.88	185.9548	13.69875639	5.49591E-05	5.71	1.777099997	0.003079
8.6328	0.123606	7086.3	118.105	91.725	0.026694068	0.082903605	11.36	15.81	179.6016	0.510713695	0.009593895	4.04	0.12242967	0.380229
8.7309	0.114777	7083.6	118.06	85.14076	0.024806248	0.083909621	11.5	15.78	181.47	0.469172639	0.010453278	4.04	0.122476336	0.414288
8.7309	0.114777	7079.7	117.995	85.09388	0.024833586	0.084002093	11.45	15.78	180.681	0.470961989	0.010459036	4.04	0.122543804	0.414517
8.6328	0.114777	7083.3	118.055	85.13715	0.024808349	0.082973844	11.43	15.84	181.0512	0.470237993	0.010336263	4.04	0.122481523	0.409651
8.6328	0.114777	7073.4	117.89	85.01816	0.024877842	0.083206269	11.5	15.84	182.16	0.466722446	0.01035073	5.71	0.17335355	0.579797
8.6328	0.114777	7071	117.85	84.98931	0.024894733	0.083262761	11.38	15.78	179.5764	0.473276634	0.010354243	0	0	0
8.6328	2.198421	7070.1	117.835	1627.665	0.476951284	0.083283961	11.36	15.69	178.2384	9.131955015	0.000540652	4.04	0.122710198	0.021427
8.6328	0.132435	7070.1	117.835	98.05211	0.028732005	0.083283961	11.26	15.81	178.0206	0.550790816	0.008974819	4.04	0.122710198	0.355694
8.6328	0.132435	7066.8	117.78	98.00635	0.028758845	0.083361762	12.01	15.88	190.7188	0.513878785	0.00897901	4.04	0.1227675	0.35586
8.6328	0.141264	7070.4	117.84	104.5934	0.030644871	0.083276893	11.59	15.84	183.5856	0.569725277	0.008413536	4.04	0.122704991	0.333449
8.6328	0.141264	7072.2	117.87	104.62	0.030629274	0.083234508	11.47	15.78	180.9966	0.578021822	0.008411395	4.04	0.122673761	0.333364
8.6328	0.132435	7069.8	117.83	98.04795	0.028734444	0.083291029	11.57	15.84	183.2688	0.534995323	0.0089752	4.04	0.122715405	0.355709

Table 3: Data log file from Dynamic testing a DC Motor with Propeller.

Table3 shows the .csv file created by the Arduino from testing the same 11X5 wooden propeller under dynamic flight conditions. In dynamic test mode, the Arduino calculates and logs the airspeed and Advance ratio as shown above. For dynamic tests, the mean drag was found by removing the propeller from the motor and running the wind tunnel at various speeds. The force readings shown in the thrust readouts from the drag test were subtracted from the thrust readings shown when testing with the propeller to get the Thrust generated by just the propeller.

The Advance Ratio is defined by the *Equation6.1.7.* [46]

$$J = \frac{V}{nD} \tag{6.1.7}$$

The Efficiency in the last column of *Table3* is given by the equation 6.1.8 [46]

$$\eta = \frac{C_T J}{C_P} \tag{6.1.7}$$

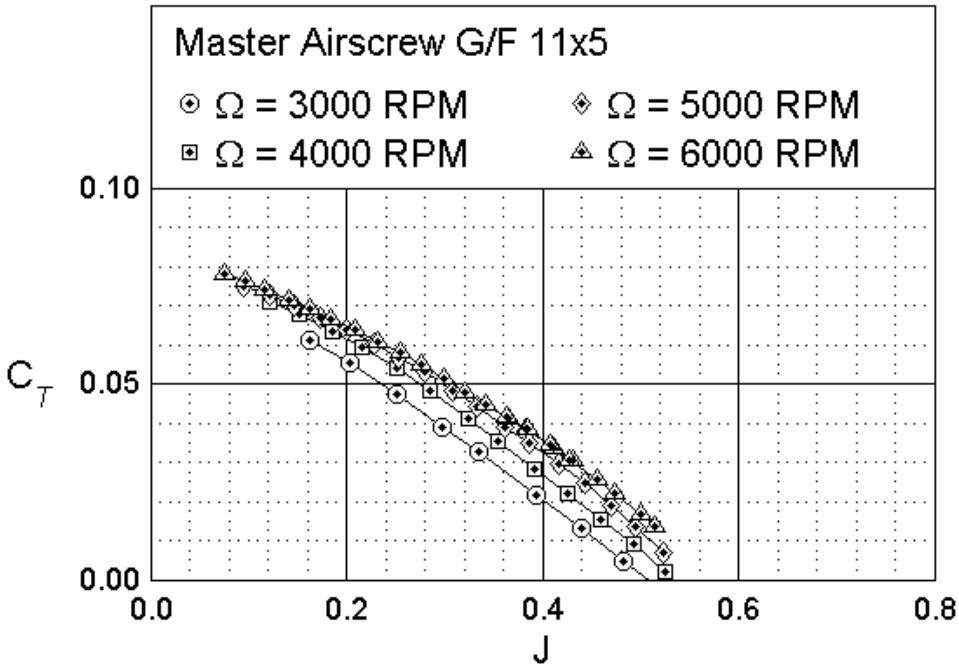


Figure 59: Coefficient of Thrust vs J curve of a 11x5 Propeller taken from UIUC database. [51]

Figure59 shows the Coefficient of Thrust vs Advance Ratio curve of a 11 X 5 propeller under dynamic test conditions taken from the UIUC database. *Figure60* shows the Coefficient of Thrust vs Advance Ratio curve of the same wooden propeller (11X5) tested under dynamic conditions in the wind tunnel. The graph shown in the *Figure60* is plotted from the data recorded by the Arduino DAQ system. Higher advance ratios couldn't be achieved in the testing as the wind tunnel couldn't move air at velocities greater than 16m/s.

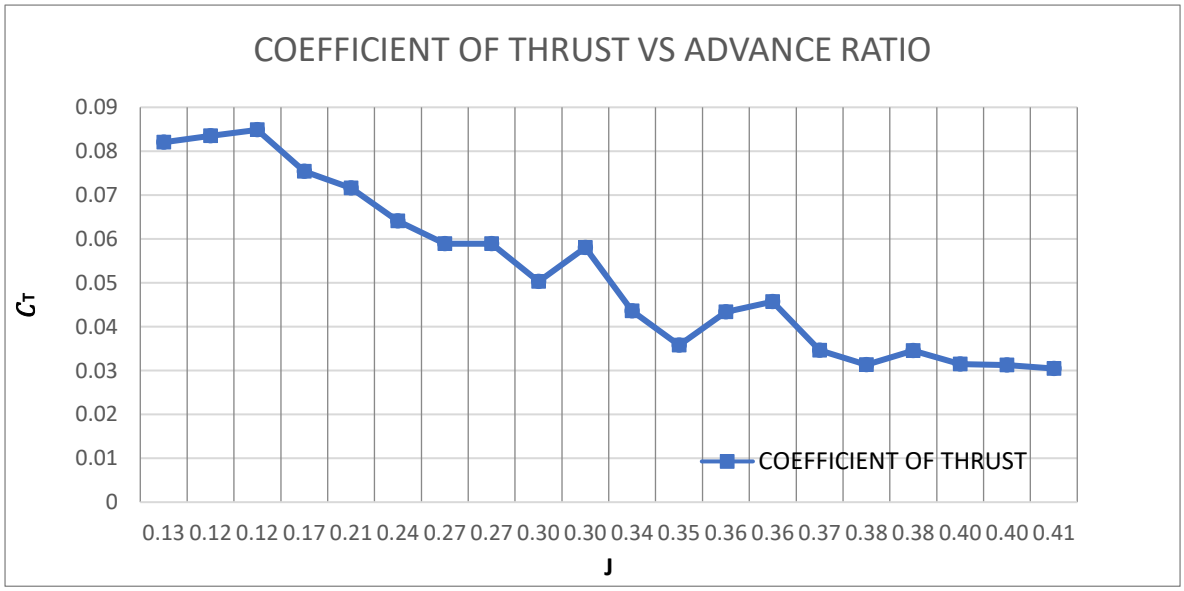


Figure 60: Coefficient of Thrust vs J curve of the 11x5 prop tested with the DAQ system.

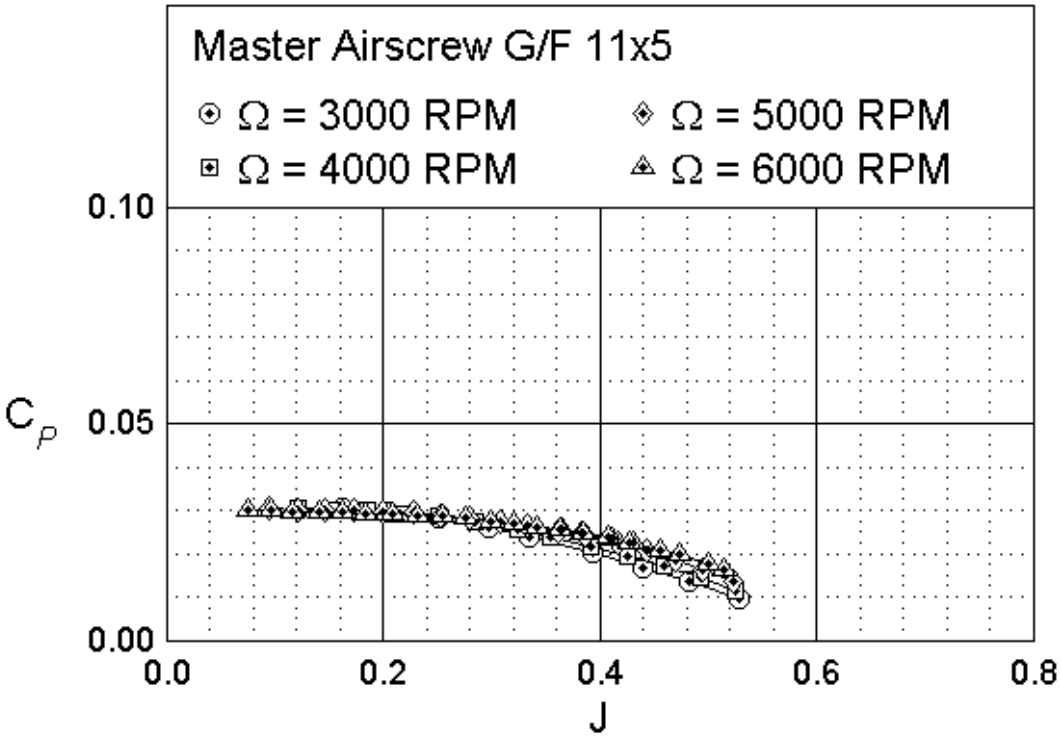


Figure 61: Coefficient of Power vs J curve of a 11x5 Propeller taken from UIUC database. [52]

Figure61 shows the Coefficient of Power vs Advance Ratio curve of a 11 X 5 propeller under dynamic test conditions taken from the UIUC database. *Figure62* shows the Coefficient of Thrust vs Advance Ratio curve of the same wooden propeller (11X5) tested under dynamic conditions in the wind tunnel. The graph shown in the *Figure62* is plotted from the data recorded by the Arduino DAQ system.

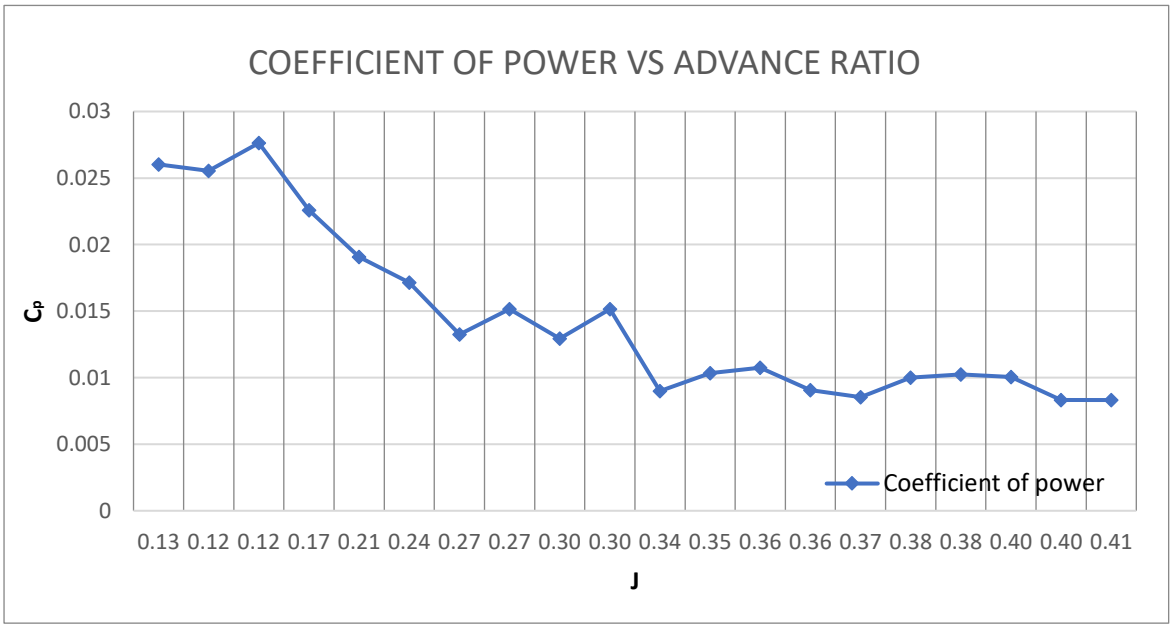


Figure 62: Coefficient of power vs J curve of the 11x5 prop tested with the DAQ system.

6.2 Testing an IC engine with Propeller

An 18 X 8 propeller is attached to an OS GF30 petrol engine[26] as shown in *Figure63*.

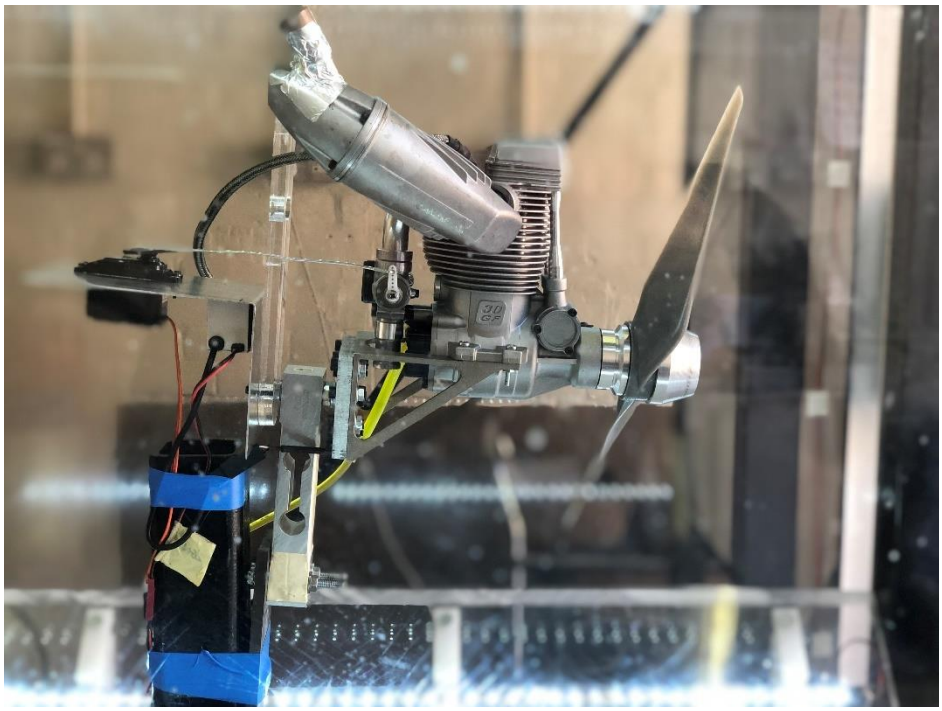
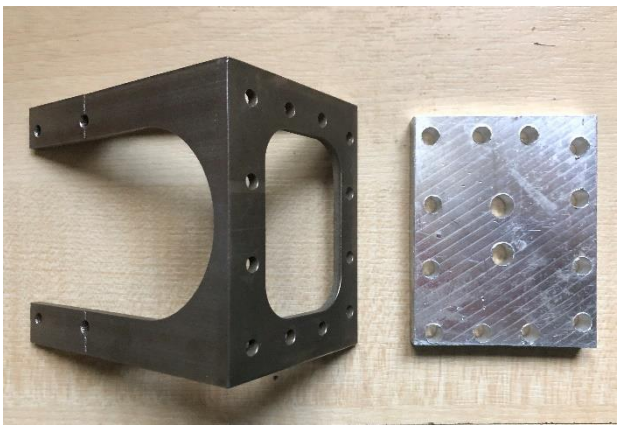


Figure 63: Overview of the mounted engine with propeller.

Various mounting fixtures such as the servo holder *(Figure63)*, load cell to engine adapter(*Figure64(a)*) and the spinner (*Figure64(b)*) had to be manufactured in the Engineering Design and Manufacturing Centre at the University of Southampton to test the engine.



Figures 64(a)&(b): Loadcell adapter (left); Propeller spinner cone(right)

6.2.1 Static Testing

Thrust Loadcell Reading (Kg)	Torque Loadcell Reading (Kg)	Thrust (N)	Torque (N-m)	RPM	rev/sec	Propeller Power (Watts)	Coefficient of Power	Coefficient of Thrust	Propeller Mechanical Efficiency (Kg force/ Watts)	Fuel Consumption Rate (mL/min)
-2.07	0.8	20.3067	0.70632	4213	70.21667	311.6173148	0.036781866	0.07694811	0.006642763	12.25714286
-2.11	0.72	20.6991	0.635688	4213	70.21667	280.4555833	0.033103679	0.07843503	0.007523473	12.25714286
-2.17	0.78	21.2877	0.688662	4213	70.21667	303.8268819	0.035862319	0.08066541	0.007142225	17.22857143
-2.2	0.82	21.582	0.723978	4213	70.21667	319.4077476	0.037701413	0.0817806	0.006887748	17.27142857
-2.33	0.94	22.8573	0.829926	5145	85.75	447.1501363	0.028979026	0.058075866	0.005210778	12.3
-2.38	0.81	23.3478	0.715149	5145	85.75	385.3102239	0.024971288	0.059322129	0.006176841	12.17142857
-2.45	0.85	24.0345	0.750465	5145	85.75	404.3378892	0.026204438	0.061066898	0.006059289	17.35714286
-2.55	0.83	25.0155	0.732807	4914	81.9	377.0972622	0.028050104	0.06967555	0.006762181	12.21428571
-2.64	0.9	25.8984	0.79461	4914	81.9	408.9006457	0.030415775	0.072134688	0.006456336	12.25714286
-2.73	0.98	26.7813	0.865242	4914	81.9	445.2473698	0.0331194	0.074593825	0.006131423	12.21428571
-2.74	1	26.8794	0.8829	5747	95.78333	531.3507916	0.024708381	0.054736711	0.005156669	39.98571429
-2.71	0.94	26.5851	0.829926	5747	95.78333	499.4697441	0.023225878	0.054137404	0.005425754	23.18571429
-2.64	0.92	25.8984	0.812268	5747	95.78333	488.8427283	0.022731711	0.052739021	0.00540051	23.31428571
-2.72	0.89	26.6832	0.785781	4956	82.6	407.8133506	0.029570189	0.073066254	0.006669718	12.21428571
-2.77	0.97	27.1737	0.856413	4956	82.6	444.4707304	0.032228183	0.074409383	0.006232131	23.14285714
-2.79	1.06	27.3699	0.935874	4956	82.6	485.7102827	0.035218427	0.074946635	0.005744165	17.27142857
-2.82	1.06	27.6642	0.935874	4956	82.6	485.7102827	0.035218427	0.075752513	0.00580593	17.35714286
-2.81	1.07	27.5661	0.944703	5064	84.4	500.9767944	0.034050467	0.072298529	0.005609042	12.25714286
-2.84	1.03	27.8604	0.909387	5064	84.4	482.2486899	0.032777552	0.073070399	0.005889078	12.21428571
-2.83	0.91	27.7623	0.803439	5064	84.4	426.0643765	0.028958808	0.072813109	0.006642189	23.22857143
0.14	0.86	1.3734	0.759294	5064	84.4	402.654246	0.027367665	0.003602062	0.000347693	12.3
-2.76	0.87	27.0756	0.768123	4993	83.21667	401.6251987	0.028478874	0.073046007	0.006872079	12.25714286
-2.79	0.96	27.3699	0.847584	5297	88.28333	470.1553049	0.02792145	0.065607697	0.005934209	39.81428571
-2.77	0.92	27.1737	0.812268	5297	88.28333	450.5655006	0.026758057	0.06513739	0.00614783	12.21428571
0.14	0.95	1.3734	0.838755	5297	88.28333	465.2578538	0.027630602	0.003292142	0.000300908	12.25714286
-2.89	1.1	28.3509	0.97119	5297	88.28333	538.7196202	0.031993329	0.067959227	0.005364572	12.21428571
-2.96	1.04	29.0376	0.918216	5094	84.9	489.8153767	0.032707108	0.075263492	0.006043093	23.31428571
-3.02	1	29.6262	0.8829	5134	85.56667	474.6746066	0.030960999	0.075597206	0.006362253	23.31428571
-3.03	1	29.7243	0.8829	5134	85.56667	474.6746066	0.030960999	0.075847528	0.00638332	12.17142857
-3.04	1.05	29.8224	0.927045	5134	85.56667	498.4083369	0.032509049	0.07609785	0.006099416	12.25714286
-3	1.04	29.43	0.918216	5185	86.41667	498.5655139	0.031569122	0.073626519	0.006017263	12.3
-2.92	0.96	28.6452	0.847584	5185	86.41667	460.2143206	0.029140728	0.071663145	0.00634487	12.25714286
-2.92	1.02	28.6452	0.900558	5185	86.41667	488.9777156	0.030962024	0.071663145	0.005971642	12.21428571
0.14	1.04	1.3734	0.918216	5185	86.41667	498.5655139	0.031569122	0.003435904	0.000280806	12.25714286
-3.07	1.04	30.1167	0.918216	5237	87.28333	503.5655924	0.030945313	0.073855656	0.006096525	23.27142857
-3.06	1.06	30.0186	0.935874	5314	88.56667	520.7958923	0.030632995	0.071497171	0.005875622	12.25714286
-3.01	0.99	29.5281	0.874071	5314	88.56667	486.4037107	0.028610061	0.070328917	0.006188275	12.17142857
-2.99	1.11	29.3319	0.980019	5314	88.56667	545.3617363	0.032077947	0.069861615	0.0054826	644.7857143
-3	1.13	29.43	0.997677	5314	88.56667	555.1880739	0.032655928	0.070095266	0.005403574	30.47142857
-3.06	1.08	30.0186	0.953532	5264	87.73333	525.629548	0.031806705	0.072861851	0.005821591	12.17142857
-3.03	1.11	29.7243	0.980019	5264	87.73333	540.2303688	0.032690225	0.072147519	0.005608718	23.22857143
-3.01	1.07	29.5281	0.944703	5264	87.73333	520.7626077	0.031512199	0.071671298	0.005779985	12.21428571
-3.04	1.08	29.8224	0.953532	5252	87.53333	524.4313044	0.031952218	0.072716787	0.005796755	89.78571429

Table 4: Data log file from Static testing an IC engine with Propeller.

Table4 shows the .csv file created by the Arduino from testing an 18 X 8 propeller attached to an IC engine under static flight conditions. For IC engine testing, the Arduino also calculates and logs the fuel flow rate in the last column of the log file as shown in *Table4.*

Further testing and plotting various graphs as shown in *Chapters 6.1.1&6.1.2* was not performed for this IC engine as there was no data available for an 18 X 8 propeller to compare with the data gathered by the Arduino. The main objective of this static testing is to validate the Arduino system's fuel flow rate sensing capability, and this can be seen in *Table4.* There was some noise in logging the fuel flow rate from the sensor and this explains some of the random numbers logged in the last column of the above table.

6.3 Margin of Error

The margin of errors in the Arduino DAQ system's measurements are

Load cell readings – 0%

RPM – 0%

Airspeed – 5%

Voltage - 1%

Current – 5%

Fuel flow rate – 7%

The airspeed can be a off by a small value as the density of air keeps changing from time to time. This margin of error can be reduced by adding a temperature and pressure sensor to the system and this will allow for real time density monitoring. The current shunt resistor used in this project can be swapped for a higher quality one to reduce the error in reading the current drawn by a DC motor. The margin error in reading the fuel flow rate went down to zero when the flow rate is high. It is found that the margin of error increases with decrease in the fuel flow rate to the IC engine.

Chapter 7

Conclusion

This project has met all the aims and objectives mentioned in *Chapter1.4.* A simple DAQ Arduino system was first developed to log enough parameters to test just the propellers and this provided valuable insights onto how microcontrollers work. The DAQ model was then further developed to measure the motor's performance by calculating the Electrical Power or Fuel flow rate depending on the motor type. Eelevant testing was performed in the final part of the project to validate the system's datalogging capability. This project has provided me with a strong basis on UAV propulsion system such as gasoline engines and DC motors and on microcontrollers.

7.1 Future Work

Fully Automated Propeller Testing

This project eliminates a lot of human effort required to test a motor with its propeller. Future work could implement a PID controller that controls the wind tunnel's induction motor speed and the propellers RPM. This controller then can test a propeller under static conditions at various RPMs and log the data onto the SD card. For dynamic testing, the system must also control the inlet airflow velocity of the wind tunnel to change the Advance ratio during tests. This would mean all the data required to measure the performance of a propeller along with its motor would be gathered by just switching on the DAQ system and entering information relevant to a test.

Bibliography

[1] A. Mukherjee, "Design and Test of an Off-Axis Ducted Fan", Masters, University of Southampton, 2019.

[12] Rohit Vadlamudi. ArduinoDAQ. https://github.com/rohitvadlamudi/ArduinoDAQ

[3] "Load cell", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Load_cell. [Accessed: 26- Apr- 2019].

[4] S. Al-Mutlaq and A. The Giant, "Load Cell Amplifier HX711 Breakout Hookup Guide - learn.sparkfun.com", *Learn.sparkfun.com*, 2019. [Online]. Available: https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide/hardware-hookup-. [Accessed: 26- Apr- 2019].

[5] Measurement Computing, "Analog to Digital Conversion", Measurement Computing, Norton, 2019.

[6]"Pressure measurement", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Pressure_measurement#Static_and_dynamic_pressure. [Accessed: 06- May- 2019].

[7]"Bernoulli's principle", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Bernoulli%27s_principle. [Accessed: 03- May- 2019].

[8] "Hall effect sensor", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Hall_effect_sensor. [Accessed: 26- Apr- 2019].

[9] "Water Flow Sensor YF-S201 Arduino Interface", *theoryCIRCUIT - Do It Yourself Electronics Projects*, 2019. [Online]. Available: http://www.theorycircuit.com/water-flow-sensor-yf-s201-arduino-interface/. [Accessed: 06- May- 2019].

[10]"Checkline PLT-5000, Combination Contact and Non-Contact Laser Tachometer", *Omnicontrols.com*, 2019. [Online]. Available: https://www.omnicontrols.com/product/plt-5000-laser-tachometer/. [Accessed: 06- May- 2019].

[11] "Arduino", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Arduino. [Accessed: 26-Apr- 2019].

[12]"Compare board specs", 2019. [Online]. Available: https://www.arduino.cc/en/products.compare. [Accessed: 03- May- 2019].

[13]"Arduino Mega 2560 Rev3", *Store.arduino.cc*, 2019. [Online]. Available: https://store.arduino.cc/mega-2560-r3. [Accessed: 06- May- 2019].

[14]"Arduino Nano", *Store.arduino.cc*, 2019. [Online]. Available: https://store.arduino.cc/arduino-nano. [Accessed: 06- May- 2019].

[15]"Serial communication", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Serial_communication. [Accessed: 06- May- 2019].

[16] C. Basics, "Basics of the I2C Communication Protocol", *Circuit Basics*, 2019. [Online]. Available:

http://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/. [Accessed: 03- May- 2019].

[17]"Serial Communication - learn.sparkfun.com", *Learn.sparkfun.com*, 2019. [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication/all. [Accessed: 06- May- 2019].

[18] *SPO Series Single Point Load Cell* [Online] Available: http://www.coventryscale.co.uk/scale-type/dini-argeo/dini-argeo-load-cells/spo-series-single- point-load-cell/

[19] *HALIJA HX711 Load cell amplifier* [Online] Available:https://www.amazon.co.uk/HALJIA-High-precision-Electronic-Dual-channel-Raspberry/dp/B01MS9RV9V/ref=asc_df_B01MS9RV9V/?tag=googshopuk-21&linkCode=df0&hvadid=205210909091&hvpos=1o4&hvnetw=g&hvrand=4368760331939649945&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9045801&hvtargid=pla-422609072814&psc=1

[20]"Fritzing", *Fritzing.org*, 2019. [Online]. Available: http://fritzing.org/home/. [Accessed: 06- May- 2019].

[21]"Extech HD350 Pitot Tube Anemometer + Differential Manometer", *Tester.co.uk*, 2019. [Online]. Available: https://www.tester.co.uk/extech-hd350-pitot-tube-anemometer-differential-manometer. [Accessed: 06- May- 2019].

[22]"UAV Factory – Unmanned Platforms and Subsystems", *Uavfactory.com*, 2019. [Online]. Available: http://www.uavfactory.com/product/12. [Accessed: 06- May- 2019].

[23] *Fluke 12" Pitot Tube* [Online] Available: https://uk.rs-online.com/web/p/digital-pressure-meter-accessories/8507675/

[24] MPXV7002DP Pressure Transducer https://coolcomponents.co.uk/products/pressure-sensor-mpxv7002dp?variant=45222884430&gclid=EAIaIQobChMIjbuR9PWH4gIVFuDtCh35gAgLEAYYASABEgLRmfD_BwE

[25] *Nxp.com*, 2019. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MPXV7002.pdf?fromsite=ja. [Accessed: 03- May- 2019].

[26]"NGH GF30 30cc Gas 4 Stroke Engine", *Hobbyking*, 2019. [Online]. Available:

https://hobbyking.com/en_us/gf30-engine.html?countrycode=GB&gclid=EAIaIQobChMIiO2dy_mH4gIVqrXtCh0GEgDqEAYYASABEgJbZvD_BwE&gclsrc=aw.ds&___store=en_us. [Accessed: 06- May- 2019]

[27]"Laser Diode 650nm 3v 5mw with Copper Head - Red", *PotentialLabs*, 2019. [Online]. Available: https://potentiallabs.com/cart/buy-laser-diode-online-hyderabad-india. [Accessed: 06- May- 2019].

[28]"Laser Receiver", *Embsysstore.com*, 2019. [Online]. Available: https://www.embsysstore.com/laser-receiver-detector-module. [Accessed: 06- May- 2019].

[29]*Cdn-learn.adafruit.com*, 2019. [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ultimate-gps-logger-shield.pdf. [Accessed: 06- May- 2019].

[30]"Lithium polymer battery", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Lithium_polymer_battery. [Accessed: 07- May- 2019].

[31]"What is burst current?", *4 in 1 ESC*, 2019. [Online]. Available: https://4in1esc.com/articles/what-burst-current. [Accessed: 07- May- 2019].

[32]2. Meter, "200A 75mV DC AC Current Shunt Resistor For Digital Ampere Analog Meter", *BazaarGadgets.com*, 2019. [Online]. Available: https://www.bazaargadgets.com/200a-75mv-dc-ac-current-shunt-resistor-for-digital-ampere-analog-meter.html. [Accessed: 07- May- 2019].

[33]A. [ADA1085], "Adafruit ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier [ADA1085]", *The Pi Hut*, 2019. [Online]. Available: https://thepihut.com/products/adafruit-ads1115-16-bit-adc-4-channel-with-programmable-gain-amplifier?variant=27739204689&gclid=EAIaIQobChMI1ZyrqJeI4gIVzr3tCh3iEQ4eEAQYASABEgI6dfD_BwE. [Accessed: 07- May- 2019].

[34]"American Wire Gauge Chart and AWG Electrical Current Load Limits table with ampacities, wire sizes, skin depth frequencies and wire breaking strength", *Powerstream.com*, 2019. [Online]. Available: https://www.powerstream.com/Wire_Size.htm. [Accessed: 07- May- 2019].

[35]"How to Interface an Arduino With a Flow Rate Sensor to Measure Liquid | Arduino", *Maker Pro*, 2019. [Online]. Available: https://maker.pro/arduino/tutorial/how-to-interface-arduino-with-flow-rate-sensor-to-measure-liquid. [Accessed: 07- May- 2019].

[36]"3.2 Inch Nextion HMI Intelligent Smart Usart UART Serial Touch TFT LCD Screen MO for sale online | eBay", *eBay*, 2019. [Online]. Available: https://www.ebay.co.uk/p/3-2-Inch-Nextion-HMI-Intelligent-Smart-Usart-UART-Serial-Touch-TFT-LCD-Screen-MO/5013933278?iid=172841676596. [Accessed: 07- May- 2019].

[37]V. Design, "'Apple Apps' by Vicons Design", *Iconfinder*, 2019. [Online]. Available: https://www.iconfinder.com/icons/2697651/apple_configuration_control_gear_preferences_setting_settings_icon. [Accessed: 07- May- 2019].

[38]"Rocket Clip Art #40810 - Free Icons and PNG Backgrounds", *freeiconspng.com*, 2019. [Online]. Available: https://www.freeiconspng.com/img/40810. [Accessed: 07- May- 2019].

[39]"Book free vector icons designed by Smashicons", *Flaticon*, 2019. [Online]. Available: https://www.flaticon.com/free-icon/book_660419. [Accessed: 07- May- 2019].

[40]"Scale free vector icons designed by smalllikeart", *Flaticon*, 2019. [Online]. Available: https://www.flaticon.com/free-icon/scale_1498717. [Accessed: 07- May- 2019].

[41]"Propeller free vector icons designed by Eucalyp", *Flaticon*, 2019. [Online]. Available: https://www.flaticon.com/free-icon/propeller_1287417#term=propeller&page=2&position=81. [Accessed: 07- May- 2019].

[42]"Battery free vector icons designed by Linector", *Flaticon*, 2019. [Online]. Available: https://www.flaticon.com/free-icon/battery_968127#term=car battery&page=3&position=79. [Accessed: 07- May- 2019].

[43]"Engine free vector icons designed by smalllikeart", *Flaticon*, 2019. [Online]. Available: https://www.flaticon.com/free-icon/engine_1682945. [Accessed: 07- May- 2019].

[44]"Hat free vector icons designed by Smashicons", *Flaticon*, 2019. [Online]. Available: https://www.flaticon.com/free-icon/hat_135040. [Accessed: 07- May- 2019].

[45]"Propeller free vector icons designed by Vectors Market", *Flaticon*, 2019. [Online]. Available: https://www.flaticon.com/free-icon/propeller_234762. [Accessed: 07- May- 2019].

[46]"UIUC Propeller Data Site", *M-selig.ae.illinois.edu*, 2019. [Online]. Available: https://m-selig.ae.illinois.edu/props/propDB.html. [Accessed: 07- May- 2019].

[47]R. W. Deters and M. S. Selig, "Static Testing of Micro Propellers", *M-selig.ae.illinois.edu*, 2019. [Online]. Available: https://m-selig.ae.illinois.edu/pubs/DetersSelig-2008-AIAA-2008-6246-MicroProps.pdf. [Accessed: 07- May- 2019].

[48]"How to Measure Brushless Motor and Propeller Efficiency · GitBook", *Rcbenchmark.gitlab.io*, 2019. [Online]. Available: https://rcbenchmark.gitlab.io/docs/en/dynamometer/theory/how-to-measure-brushless-motor-and-propeller-efficiency.html. [Accessed: 07- May- 2019].

[49]*M-selig.ae.illinois.edu*, 2019. [Online]. Available: https://m-selig.ae.illinois.edu/props/volume-1/plots/magf_11x5_static_ct.png. [Accessed: 07- May- 2019].

[50]*M-selig.ae.illinois.edu*, 2019. [Online]. Available: https://m-selig.ae.illinois.edu/props/volume-1/plots/magf_11x5_static_cp.png. [Accessed: 07- May- 2019].

[51]*M-selig.ae.illinois.edu*, 2019. [Online]. Available: https://m-selig.ae.illinois.edu/props/volume-1/plots/magf_11x5_ct.png. [Accessed: 07- May- 2019].

[52]*M-selig.ae.illinois.edu*, 2019. [Online]. Available: https://m-selig.ae.illinois.edu/props/volume-1/plots/magf_11x5_cp.png. [Accessed: 07- May- 2019].

[53]"Consent Form | Boating Magazine", Boatingmag.com, 2019. [Online]. Available: https://www.boatingmag.com/mercury-enertia-eco-pro#page-5. [Accessed: 07- May- 2019].