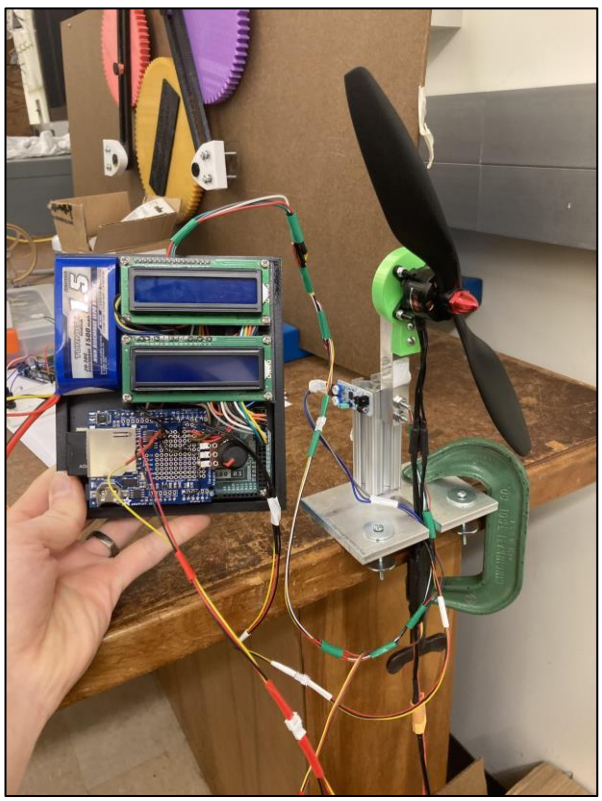
Thrust Measurement System User Manual

Daniel Noel[1], Dr. Michael Davis[1]

[1]University of Southern Maine



Author Note

This report was generated as part of the 2022-23 Undergraduate Research Opportunities Program and was funded by the Maine Space Grant Consortium.

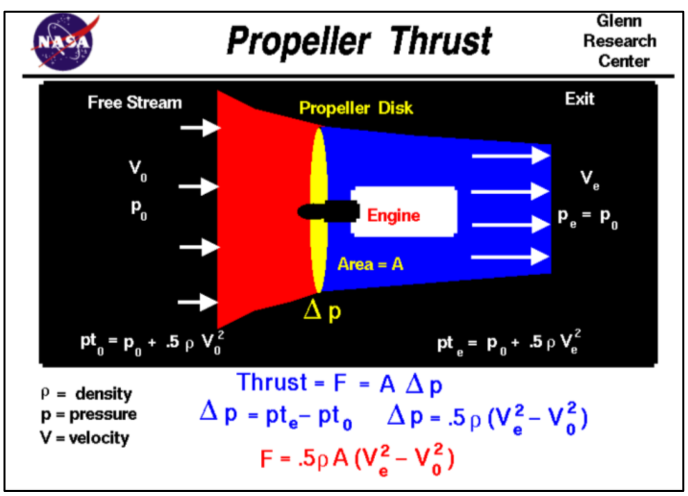
Table of Contents

Purpose

The Thrust Measurement System (TMS) was conceived to collect reliable propeller performance data in a pedagogical environment. An iterative design review process brought the thrust measurement device from an idea to a product. It applies mechatronic principles to the exploration and quantification of basic fluid dynamics properties, providing a way for the user to accurately measure the thrust generated by a propeller at a variable angular velocity. The TMS is driven by an Arduino microcontroller that collects data from a load cell and an inferred sensor to calculate the thrust and RPM of an attached propellor. LCD screens display both the thrust and RPMs in real time. The propellor's angular velocity is determined by a potentiometer that limits the power supplied to a DC motor, allowing the user to adjust the rate at which the propellor is turning. This data is recorded and stored on an onboard SD card for further analysis. To maximize the audience and application of the TMS, its construction is designed to be repeatable exportable to other educators.

Brief Aerodynamics Overview

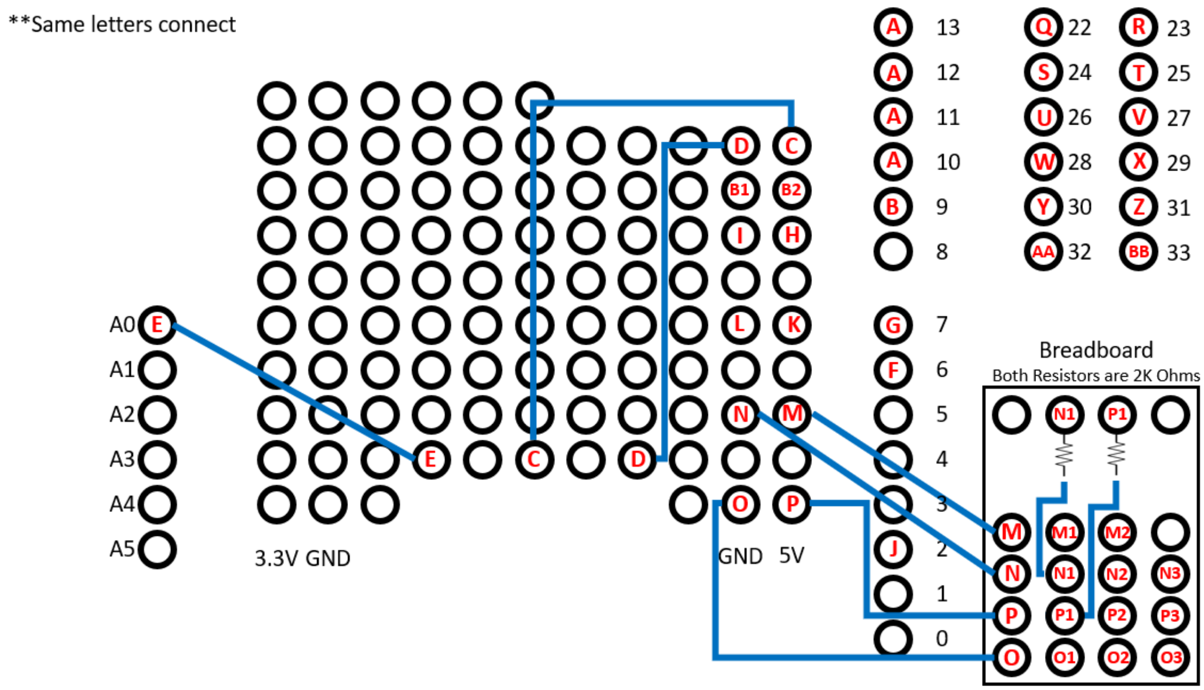
Thrust is explained by Newton's third law which states that for every action there is an equal and opposite reaction. The TMS measures the reaction force generated by the propellor.



This is seen in the figure where the thrust is equal to the product of the propeller's area and the pressure difference that it creates. This pressure differential is the result of energy conservation and is proportional to the air's change in velocity as it passes through the propeller.

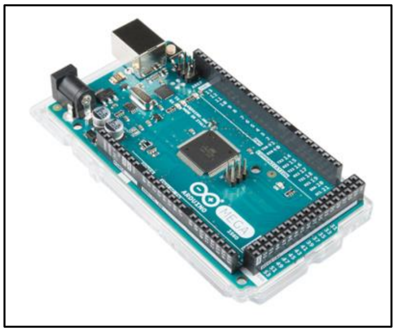
Components and Layout

The below diagram and table correspond to the circuit layout for the Arduino Mega board and the attached Data Logging Shield. An overview of each component is provided as well.



Letter	Description	Letter	Description	Letter	Description
A	SD Card Writer (Reserved Spots)	M1	LCD 1 – VDD	Q	LCD 1 – RS
B	PWM Output / ESC Input	M2	LCD 1 – A	R	LCD 1 – E
B1	PWM Ground	N1	LCD 1 – VO	S	LCD 1 – D4
B2	PWM Power	N2	LCD 1 – RW	T	LCD 1 – D5
C	Potentiometer Power	N3	LCD 1 – K	U	LCD 1 – D6
D	Potentiometer Ground	O1	LCD 2 – VDD	V	LCD 1 – D7
E	Potentiometer Input	O2	LCD 2 – A	W	LCD 2 – RS
F	Load Cell SCK	P1	LCD 2 – VO	X	LCD 2 – E
G	Load Cell DT	P2	LCD 2 – RW	Y	LCD 2 – D4
H	Load Cell Power VCC	P3	LCD 2 – K	Z	LCD 2 – D5
I	Load Cell Ground GND			AA	LCD 2 – D6
J	Tachometer Sensor (out)			BB	LCD 2 – D7
K	Tachometer Power				
L	Tachometer Ground				
M	Power for LCD Screen 1				
N	Ground for LCD Screen 1				
O	Power for LCD Screen 2				
P	Ground for LCD Screen 2				

Arduino Mega 2560 R3

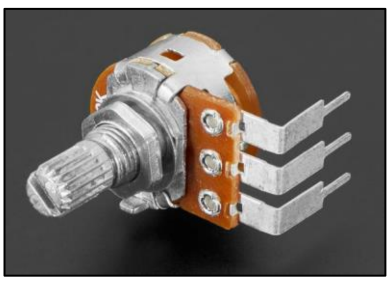


The Arduino Mega is a microcontroller board. It has 54 digital input/output pins, 16 analog inputs, 4 hardware serial ports, a 16 MHz crystal oscillator, a USB connection, a power jack, and a reset button. All components in the TMS are controlled by and connected to this device.

DC Turbine

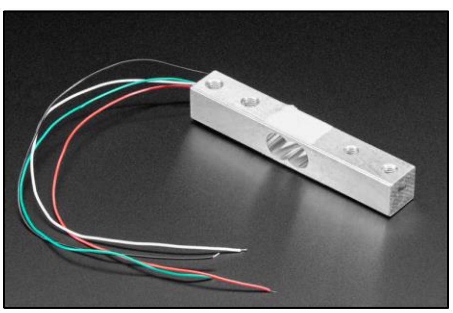
The EMAX Multi copter motor MT2213-935KV was used as the DC turbine. This system allowed for pulse width modulation control which allows the user to adjust the RPMs of the propellor. This device can rotate both clockwise and counterclockwise.

Potentiometer



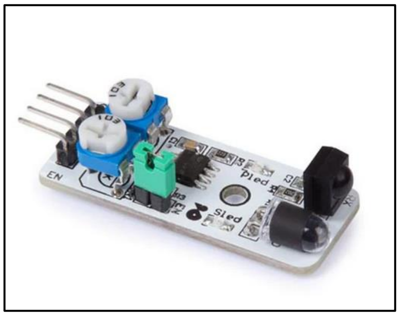
The chosen potentiometer was the Panel Mount Right Angle Potentiometer. It contains a standard linear taper of 10K Ohms, and it also comes equipped with an on/off switch at the beginning of the shaft's rotation.

Load Cell



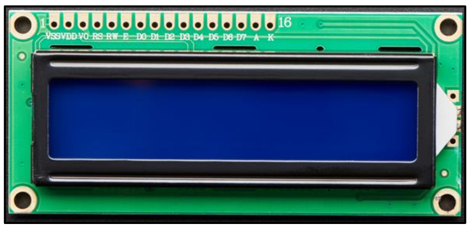
The load cell chosen was from Adafruit and has a 5kg max limit. It is a device that converts a force, such as weight, pressure, or torque, into an electrical signal that can be measured and standardized.

Tachometer



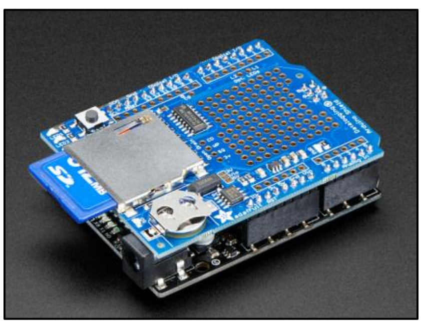
The Velleman Obstacle Detection IR Sensor was used as a tachometer by recognizing every time a propellor blade passes in front of its field of view. It can interface directly with Arduino, and it also has variable frequency and intensity.

LCD Screen



The two screens used in the system were HD44780 LCDs. They display 16 characters wide and 2 rows deep. One was used to display thrust and the other RPMs.

Data Logging Shield



The data logging shield used was Adafruit Assembled Data Logging Shield for Arduino. It reads data from the system outputs and writes them to a removable SD card. This shield mounts directly on top of the Arduino Mega board.

3 Cell 11.1V 1500mAh Battery

The type of battery chosen for this project was a 3 Cell 11.1V, 1.5A, 1500mAh battery. A more or less powerful battery could be substituted if a different range of thrust is desired.

Arduino Code

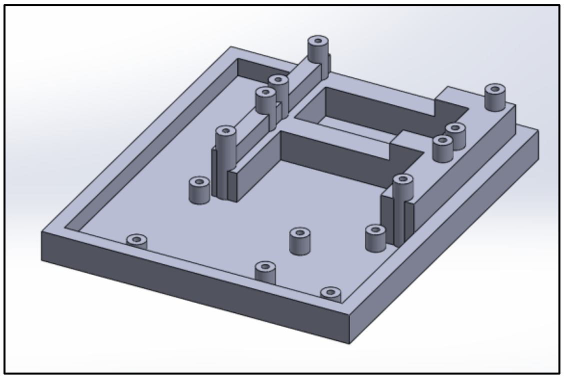
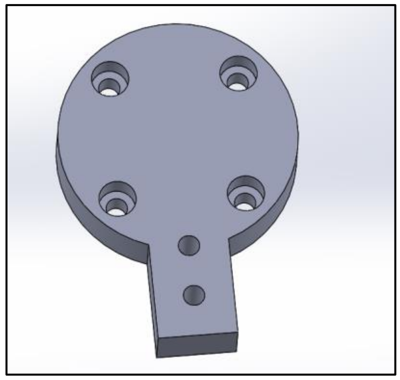
The code that needs to be uploaded to the assembled Arduino Mega can be found on TMS's GitHub page: https://github.com/danielthomasnoel/Thrust-Measurement-Device. The code is annotated, but the two main items that may need to be changed for a different setup are lines 28 and 115. Line 28 is the calibration factor for the load cell which is explained further in the calibration section. Line 115 needs to be modified depending on how many blades are on the propellor.

Case Design and Assembly

This system was designed specifically to meet the size requirements of USM's AeroLab wind tunnel. The board case will not need to be resized, however for different applications it is recommended to design all other components to meet specific requirements. The CAD files for the board case and DC motor mount can be found on the TMS's GitHub page: https://github.com/danielthomasnoel/Thrust-Measurement-Device.

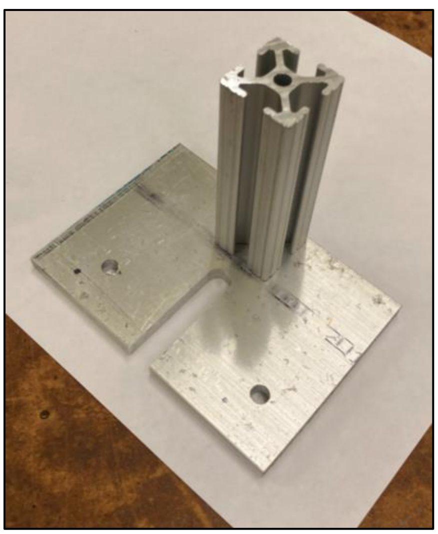
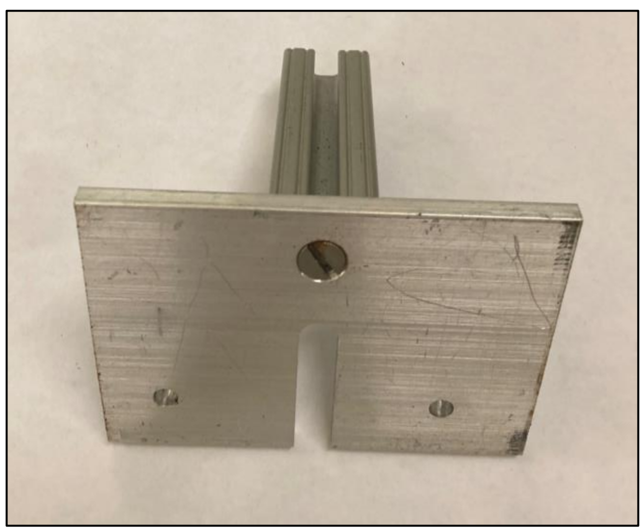
The current design requires a soldering station as about 20 connections need to be soldered directly to the data logging shield. This number can be increased or decreased depending on the specific specifications of components procured for this build.

Board Case and DC Motor Mount

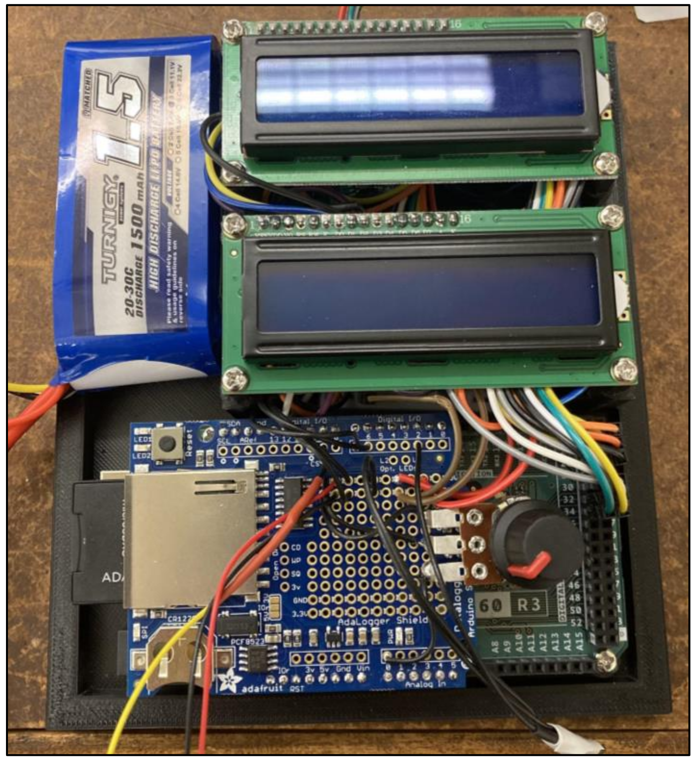
The board case (left) was designed to fit two LCD screens, 1500 mA battery pack, Arduino Mega, and data logging shield. All fasteners are sized to have slight interference with M3 fasteners. The DC motor mount (right) was designed to attach the EMAX DC Turbine to the load cell. As stated earlier, the specific dimensions of this piece may vary depending on the system's size requirements.

Thrust Stand Mount

The aluminum base and 1" 8020 post were sized to fit USM's AeroLab wind tunnel.

Full Setup



The fully assembled Thrust Stand Mount (left) is shown with the EMAX DC motor, DC motor mount, load cell, and tachometer attached. The fully assembled control system is shown fastened to the Board case. It contains two LCD screens, battery, Arduino Mega, data writing shield, and potentiometer.

Operation

The first step to operation is to upload the Arduino code from the TMS GitHub to the fully assembled Arduino Mega and components. It then needs to be calibrated and can be operated dynamically or by storing data as described below.

Calibration

This system is designed to not needed to be calibrated regularly, generally only after construction or after making modifications to the system. To calibrate, a known mass is attached axially in line with the center of the DC motor. This can be done many ways, but for this design it was done by rotating the entire system 90 degrees and attaching the mass with fishing line to the DC motor. The system is then calibrated iteratively by adjusting line 28 in the Arduino code. This number needs to be adjusted until the value displayed as thrust on the LCD screen matches the force applied to the DC motor. Assuming the mass applied is in grams, the mass only needs to be multiplied by gravity ($9.81 m/s^2$) to convert to millinewtons. This line of code is shown below, and the calibration factor s currently set to 40, calibrated with a 100g mass or 981mN force.

```
24  // Loadcell Stuff
25  #define LOADCELL_DOUT_PIN  7 //SCK
26  #define LOADCELL_SCK_PIN  6 //DT
27  HX711 scale;
28  float calibration_factor = 40; //calibration from hanging weight on propellor tip (~981 mN Force)
```

These readings should also be checked against manufacturer's data if possible. For this specific build we were using the EMAX 1045 propellor with 1.5A. To compare the readings to expected values, the user must interpolate the expected thrust at an known RPM and then compare the thrust reading to this value. Manufacturer data for this system is shown below.

The voltage (V)	Paddle size	current (A)	thrust (G)	power (W)	efficiency (G/W)	speed (RPM)	Working temperature (° C)
11	EMAX8045	1	110	11	10.0	3650	
		2	200	22	9.1	4740	
		3	270	33	8.2	5540	
		4	330	44	7.5	6200	
		5	390	55	7.1	6700	
		6	440	66	6.7	7150	
		7.1	490	78.1	6.3	7400	36
	EMAX1045	1	130	11	11.8	2940	
		2	220	22	10.0	3860	
		3	290	33	8.8	4400	
		4	370	44	8.4	4940	
		5	430	55	7.8	5340	
		6	480	66	7.3	5720	
		7	540	77	7.0	5980	
		8	590	88	6.7	6170	
		9	640	99	6.5	6410	
		9.6	670	106	6.3	6530	43

Dynamic Readings

Once calibrated, the TMS only needs to be reset before every use because this tares the load cell. It is powered directly by the battery contained in the system. Once the power is on, the top LCD screen will read either “SD Card Ready” or “SD Card Failure” indicating if the data storage is operating. After this the knob on the potentiometer is turned to increase the rate the propellor spins. In this specific system, the propellor needs some manual support to get started, so the potentiometer should be turned to about 60 degrees clockwise and then the user should carefully give the propellor a spin. The two LCD screens will now read dynamic thrust in millinewtons and RPMs. To stop the TMS turn the potentiometer knob counterclockwise until an audible click is heard.

Data Storage and Retrieval

To retrieve thrust and RPM data over time, the same process as the dynamic readings is performed with two minor changes. Before starting the test, the SD card needs to be removed from the system and cleared. Then at the end of the test, you simply remove the SD card, and it will contain about 1 data point for every .75 seconds elapsed.

Bill of Materials

The approximate bill of materials is shown in the table below with a total of $165.23. Of note, some materials used for this build were available for free use at our facility so were not included in this list. These include Thrust Stand aluminum pieces (plate and 8020), 3D printer and filament, extension wires, soldering equipment and materials, and most hardware.

Item Number	Supplier Name	Catalog Nur	Product Description	Quantity	Unit of Me	Price
1	AdaFruit	4722	1.25mm Pitch 4-pin Cable Matching Pair - 40cm long - Molex PicoBlade Compatible	2	Pair of Wi	$1.90
2	AdaFruit	4721	1.25mm Pitch 3-pin Cable Matching Pair - 40cm long - Molex PicoBlade Compatible	3	Pack	$2.85
3	AdaFruit	4685	Black Nylon Machine Screw and Stand-off Set – M3 Thread	1	Box	$16.95
4	AdaFruit	344	Heat Shrink Pack	1	Pack	$4.95
5	AdaFruit	181	Standard LCD 16x2 + extras - white on blue	2	Board	$19.90
6	AdaFruit	2046	Potentiometer Knob - Soft Touch T18 - Red	1	Knob	$0.50
7	AdaFruit	3395	Panel Mount Right Angle 10K Linear Potentiometer w/On-Off Switch - 10K Linear w/ Switch	1	Knob	$1.50
8	SparkFun	DEV-11061	Arduino Mega 2560 R3	1	Board	$48.95
9	Robotshop	RB-Vel-231	Obstacle Detection IR Sensor	2	# of Board	$10.48
10	Adafruit	4541	Strain Gauge Load Cell (5 kg)	2	# of Gauge	$7.90
11	Adafruit	1294	SD/MicroSD Memory Card (8 GB SDHC)	2	# of SD Ca	$19.90
12	Adafruit	1141	Adafruit Assembled Data Logging shield for Arduino	1	# of Board	$13.95
13	EMAX	101005002	EMAX Multicopter motor MT2213 (With Prop1045 Combo) 935KV	1	Set	$15.50

Contact Information

Daniel Noel
daniel.t.noel@maine.edu
danielthomasnoel@gmail.com

Dr. Michael Davis
michael.p.davis@maine.edu