Manual Document - Prony Brake Dynamometer

Table of Contents:

Bill of Materials

Table 1: General Tools

Mechanical	2mm Hex Wrench[1]	4mm Hex Wrench[1]	3D Printer[1]	Sandpaper[1,2]
Electrical	Soldering Iron[1]	Desoldering Gun[1]	Lead Free Solder[1]	

Table 2: Mechanical Components

Item	Amount	Source	Price/Unit
M3 Heat Insert[1]	1 1	McMaster Amazon	$22.63/100 $8.99/100
M2 Screws[1]	1	McMaster	$15.68/100
M3 Screws[1]	1 1	McMaster Amazon	$15.68/100 $8.48/50
M4 Screws[1]	1 1	McMaster Amazon	$12.02/100 $7.99/25
Filament (PLA)[1]	1 (124.3g)	Amazon Bambu	$14.99/1KG $19.99/1KG

Table 3: Electronics

Item	Amount	Source	Price/Unit
Photointerrupter	1	Amazon	$9.98/10
Load Cell & Driver	1	Amazon	$15.49/4
Current Sensor	1	Amazon	$7.99/4
Arduino Uno	1	Amazon	$22.08/1
Wiring[1]	1	Amazon	$6.98/1
Breadboard[1]	1	Amazon	$19.99/16
Lead Free Solder[1]	1	Amazon	$7.99/1

[1] *Can be sourced from OEDK consumables/tools*

[2] *Optional tools for ease of construction in the event of mistake or modification*

Instructions to Build

This section provides a comprehensive overview on how to assemble the “Prony Brake” torque meter. It is split into four parts to cover its development. These are an overview of the CAD files and their parameters, the 3D-printing of the parts, the assembly of the device, and the soldering and wiring of the electronic components.

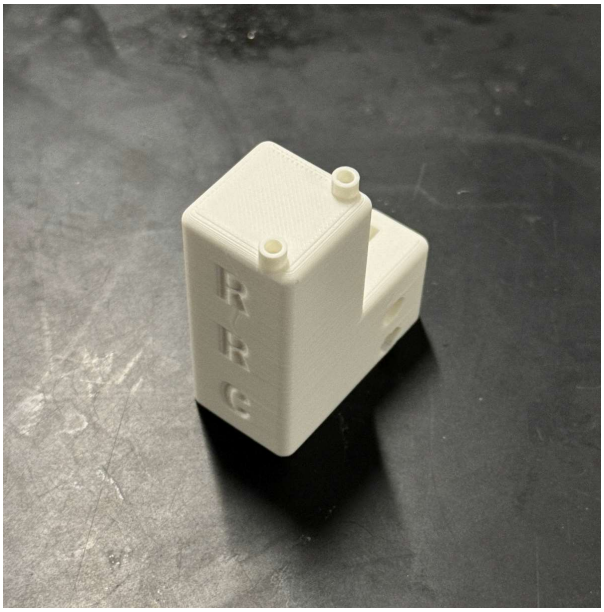
Part a). Adjusting CAD File Parameters

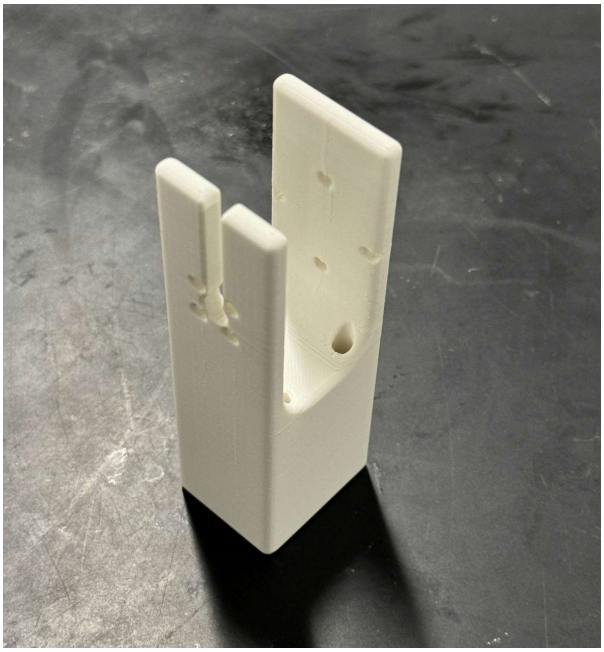
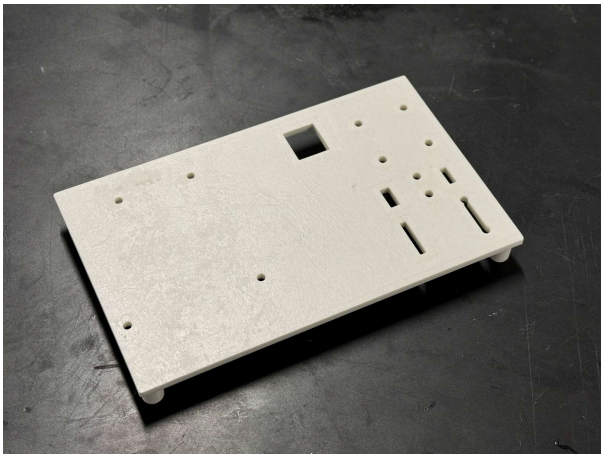
To provide an overview of the CAD components, a Pack&Go file of the project assembly is provided as a deliverable. With this folder, the user has access to modeled components and placeholder models for electronics to be able to view the complete torque meter assembly.

Part b). 3D-Printing

Table 4 shows the components that require 3D printing. The CAD files for these components should be saved as an STL and printed using filament of your choosing. For demonstration, these parts were printed using a Bambu-Lab X1-Carbon 3D Printer on White Basic PLA filament (See BOM for link to filament). The print used supports and had a duration of approximately 4 hours and 30 minutes. Please follow all proper 3D printing etiquette provided in the manual of your chosen 3D printer and by your respective makerspace.

Table 4: Parts to 3D Print

Part Image	Part Name
	Load Cell Mount

	Motor Mount of your choosing (shown is the Antigravity Motor Series Holder)
	Mounting Plate
	Photointerrupter Shaft

Once the parts are 3D printed, supports from the Load Cell Mount and Motor Mount need to be removed. Please use a hobby knife, fine tweezers, or other precision tools to chip and clip away support filament. Execute caution utilizing these tools and sand the parts as necessary for a refined look and feel.

Part c). Assembly

This section outlines the assembly of the device. Please take note of the Bill of Materials for all the mechanical and electrical components required for the assembly.

The first step focuses on the assembly of the load cell subsystem. To start, carefully align the screw holes of the load cell to the openings on the load cell mount. An ideal orientation for the load cell mount is for the wires to be facing away from the mount pillar as shown in Figure 1. Using M4 20mm Phillips head screws, screw the load cell into the mount.

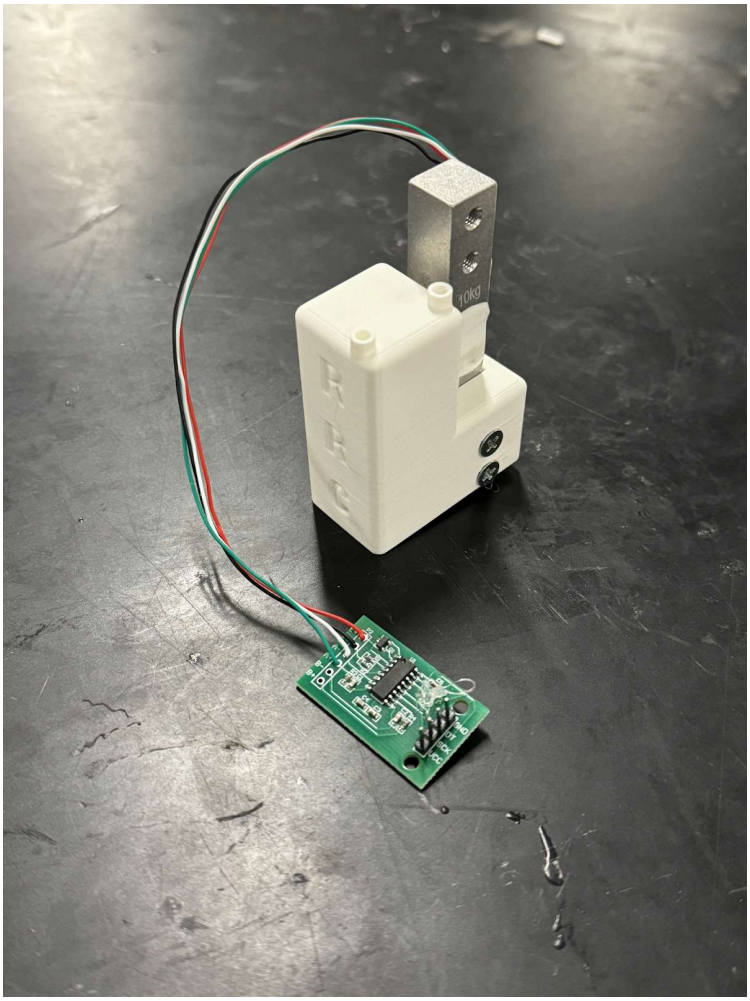


Figure 1: Load Cell Subsystem

Attach the subsystem to the main mounting plate of the device. To do so, apply some superglue to the base of the load cell mount. Carefully align the load cell with its respective opening on the mounting plate and press gently, but firmly to allow for contact. Let the superglue dry before proceeding. It is also recommended to screw in the accompanying HX711 ADC Module Weighing Sensor using two M2 8mm screws and respective M2 nuts.

The next step is to attach the photo interrupter to the load cell mount. Using superglue, carefully attach two M3 heat inserts onto the pegs of the load cell mount, and let the adhesive dry. After the heat inserts are secure, use 2 M3 10mm screws to screw in the photo interrupter. The assembled system is shown in Figure 2.

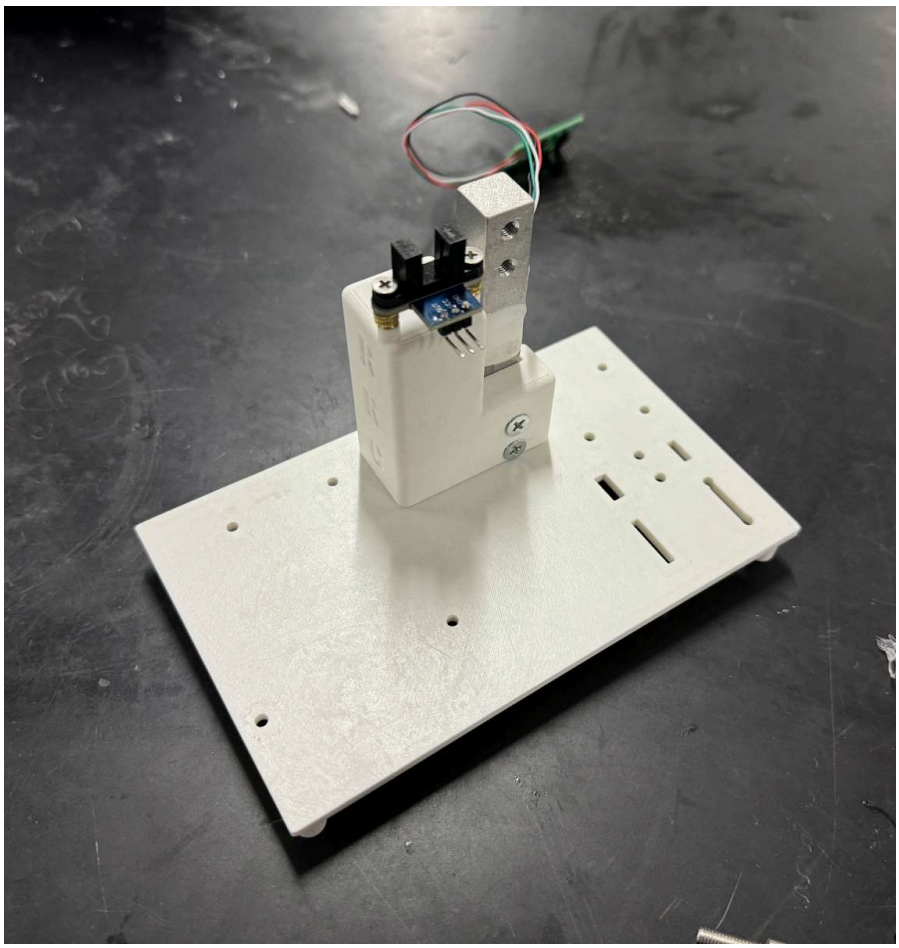


Figure 2: Load Cell Subsystem on Base

Next, heat inserts need to be applied to the mounting base of the device so that the motor mount can be attached. To do so, a soldering iron is required to heat up the inserts for attachment. For this manual, a Weller WES51 soldering station was used. To start, 4 M3 H4 inserts are needed for the motor mount, and pinpoint where the inserts will go (Figure 3). Next, set the soldering iron to be around 75 to 100 ℃ (167 to 212 °F) hotter than what the melting

point for the 3D printing material was. For example, since PLA was used at a melting point of 170 ℃ (338 °F) for printing, the soldering iron was set to 271 ℃ (520 °F) for the procedure (Figure 4). Once the soldering iron is fully hot, carefully align the heat inserts and press the soldering iron onto the top of the insert to heat it. As a safety precaution, do not touch the soldering iron nor the heat insert burns become a risk. When the heat begins to melt into the material, make sure to press gently, but firmly perpendicular to the base until the heat . If some material makes its way into the heat insert thread, remove with a thin object and/or melt away excess material. Finally, let all heat inserts dry before using them for assembly (Figure 5).

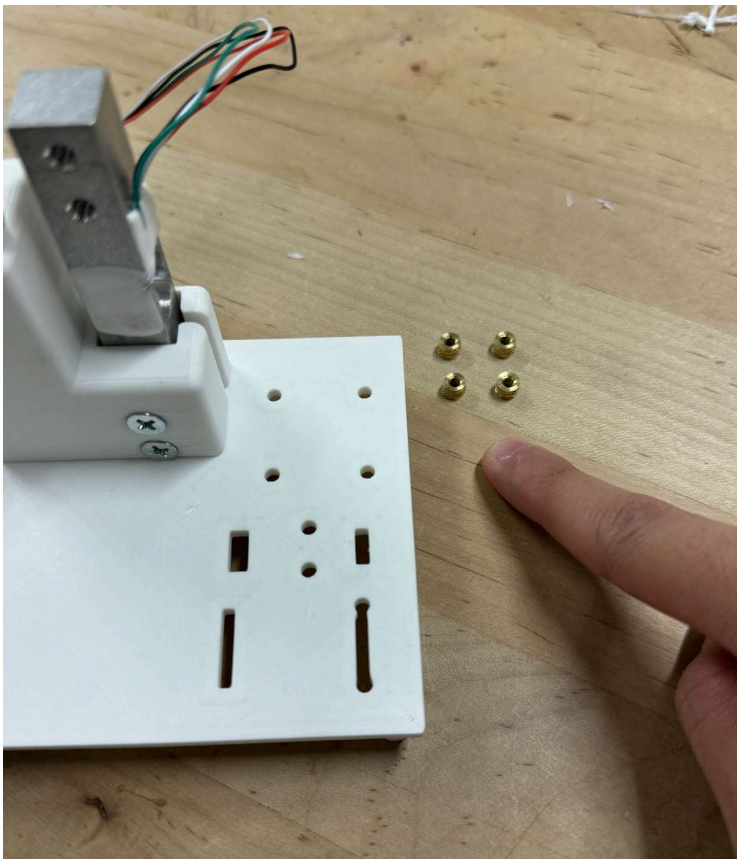


Figure 3: Heat Insert Locations

Figure 4: Setting the Heat on the Soldering Kit

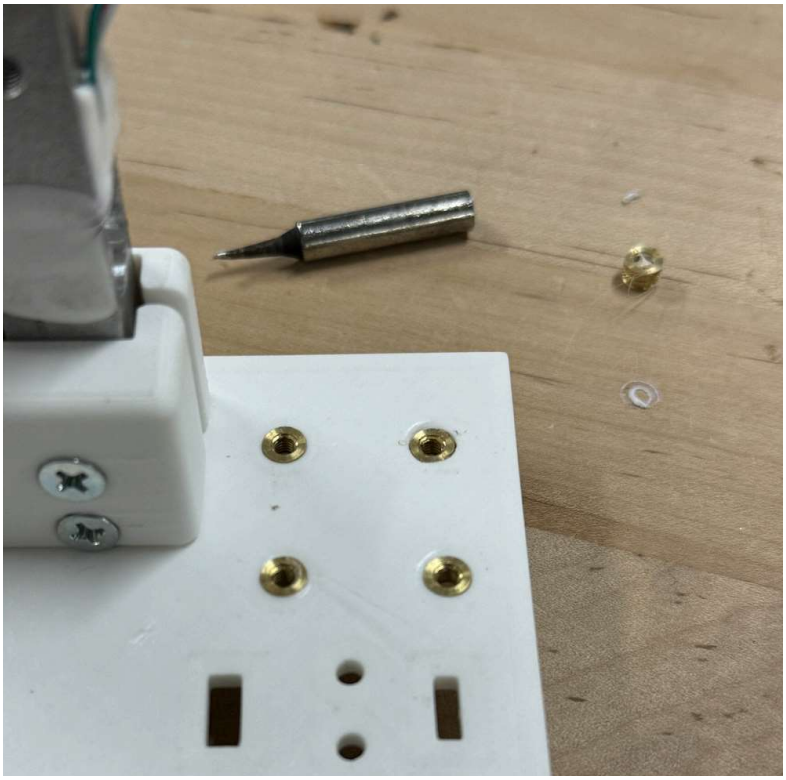


Figure 5: Inserted Heat Inserts

Once all heat inserts have dried, the motor mount may now be attached. Using four M3 16mm panhead screws, screw the chosen motor mount from the bottom of the base, making sure to align the front face of the motor mount with the photointerrupter. The assembled device up to this point is found in Figure 6.

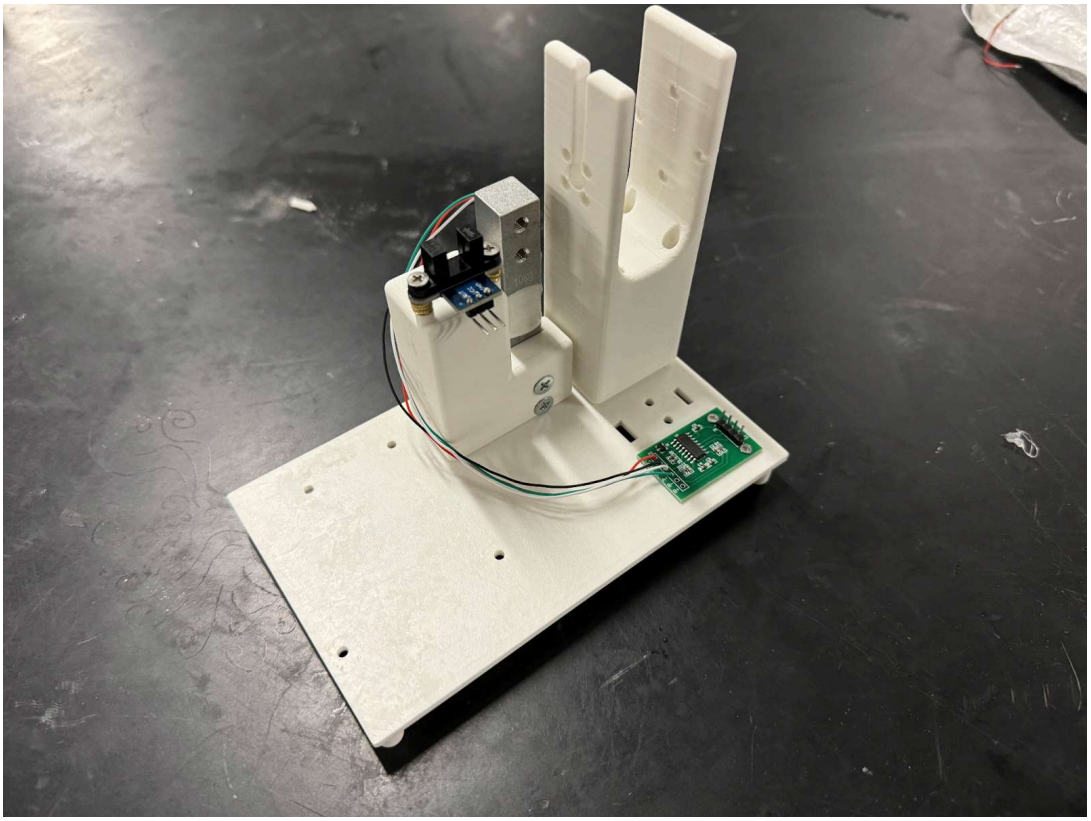
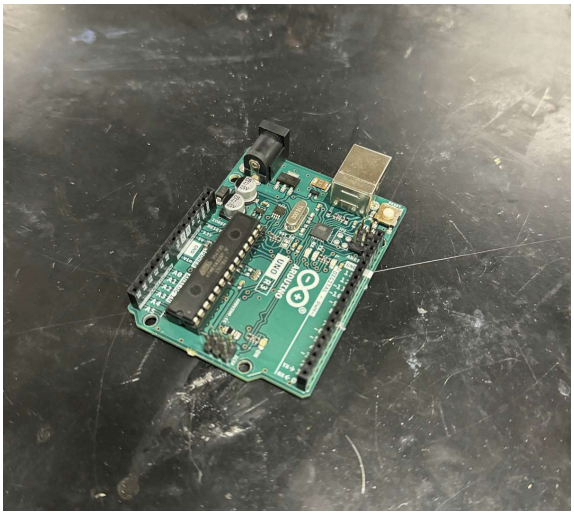
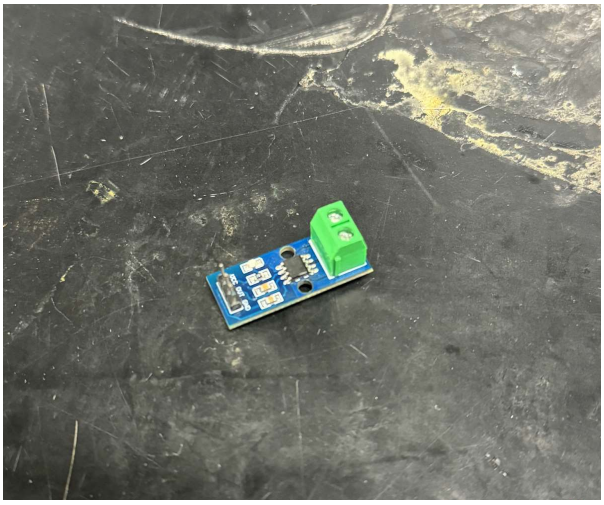


Figure 6: Load Cell Subsystem on Base

The next step is to attach the electronic components to the base. Please refer to Table 5 to identify the electronic components and their respective attachments.

Table 5: Electronic Components and Attachments

Electronic Component	Attachments
Arduino Uno R3	4 M3 12mm Panhead Screws 4 M3 Hex Nuts

	
ACS712 Current Sensor 	2 M2 12mm Screws 2 M2 Hex Nuts

Once all electronics are attached, attach the desired motor to test. As an example, an Antigravity MN5006 KV-450 motor was used to correspond to the mount. To attach the motor, align the screw openings on the faces of the motor to those of the mount, and use the respective screws as outlined by the motor manufacturer. Once the motor is in place, press-fit the photointerrupter shaft onto the motor shaft. The assembly should look like Figure 7.

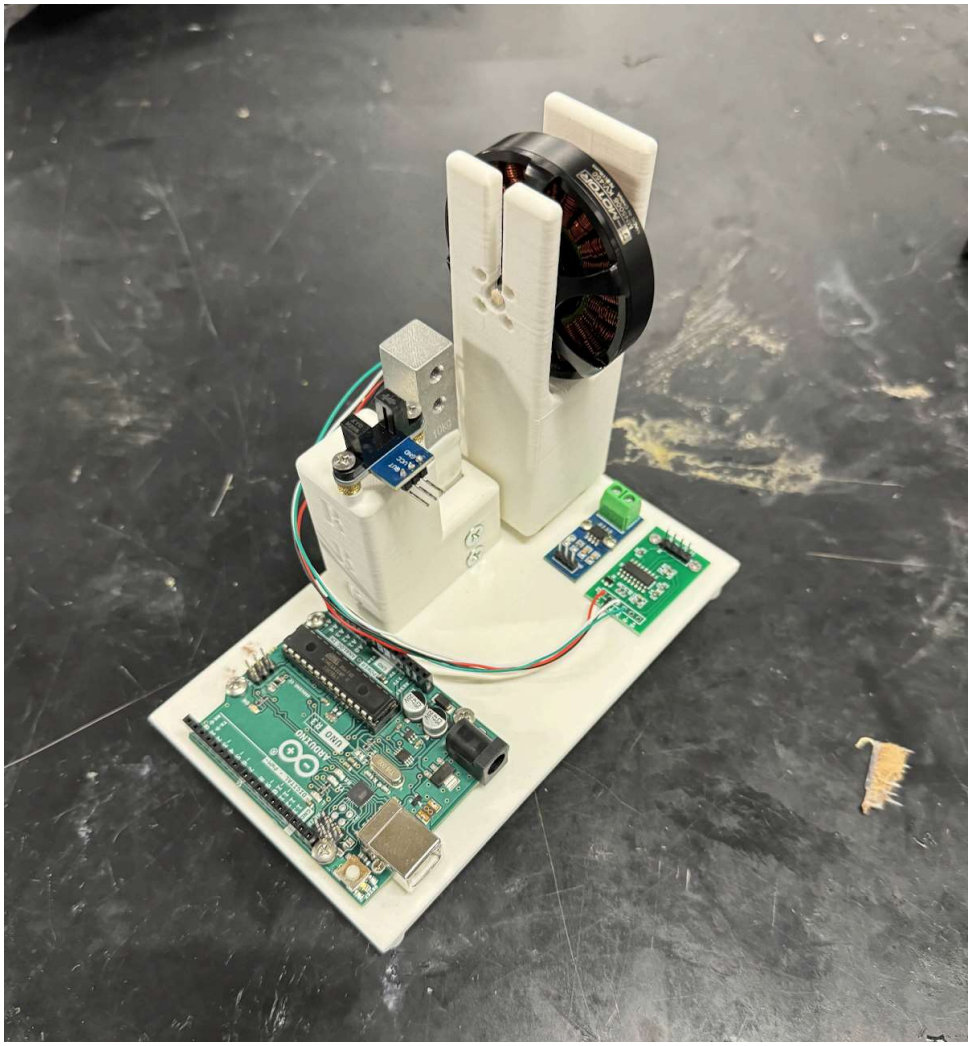


Figure 7: Electronic and Motor Attachment.

The final component to attach is the breadboard for electrical wiring. To do so, adhesive of preference or a pair of M2 8mm screws can be used for attachment to the last empty spot on the base as shown in Figure 8.

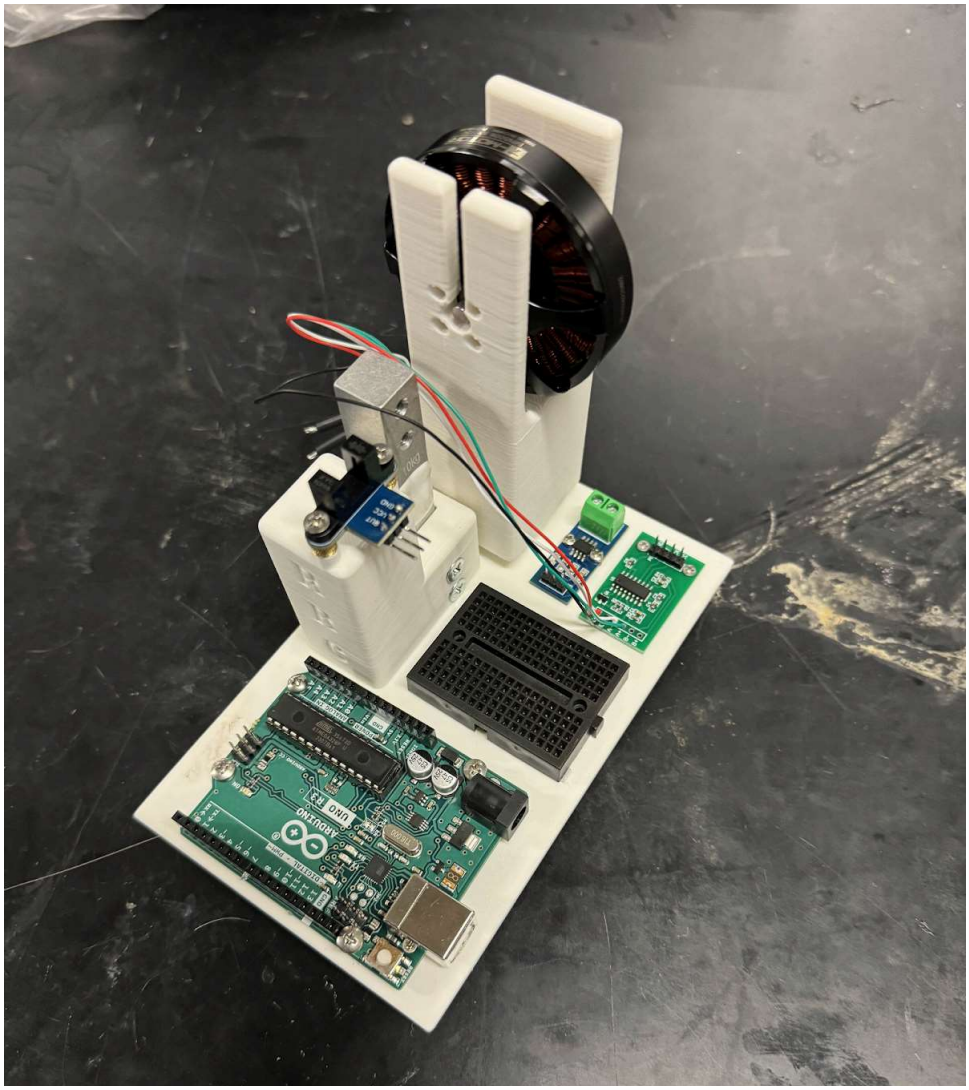


Figure 8: Breadboard Attachment.

Part d). Soldering and Wiring

This part provides an approach for assembling and wiring a dynamometer that tracks RPM, torque, and current using an Arduino Uno R3 or compatible microcontroller. The system is built around three primary components: the HX711 load cell amplifier for torque measurement, the ACS712 current sensor for current measurement, and an interrupt-based input to measure RPM.

The load cell usually comes in separate with the HX711 amplifier. It uses four wires that need to be connected to the HX711 amplifier. The correct pin mapping for the load cell to HX711 connection is shown in Table 6:

Table 6: Pin Mapping for Load Cell to HX711

Load cell Wiring	HX711 Pin Input
Red	E+
Black	E-
White	A-
Green	A+

Red (E+): Connects to the **E+** pin on the HX711 (Excitation Positive).
Black (E-): Connects to the **E-** pin on the HX711 (Excitation Negative).
White (A-): Connects to the **A-** pin on the HX711 (Signal Negative).
Green (A+): Connects to the **A+** pin on the HX711 (Signal Positive).

For the HX711 output pins to the Arduino UNO, the wiring is listed as follows in Figure 8 and Table 7:

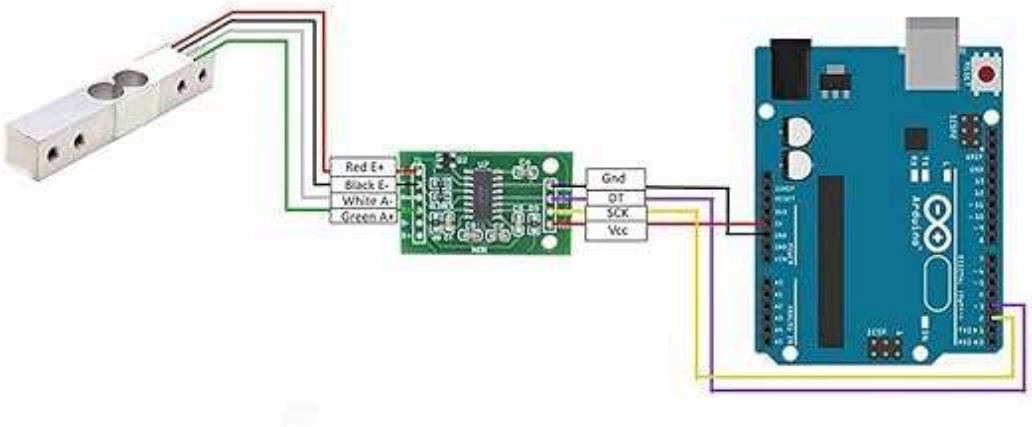


Figure 9: HX711 Output Pins to the Arduino UNO Connections

Table 7: HX711 to Arduino UNO Connections

HX711 Pin Output	Connection Description	Microcontroller Pin
VCC	Power to load cell	5V (Microcontroller)
GND	Ground of load cell	GND (Microcontroller)
DT	Data from load cell to microcontroller	Pin 4 (Digital Pin)

SCK	Clock for HX711	Pin 3 (Digital Pin)

Using the upper process as a reference, the ACS712 and the photo interrupter can also be wired with the following scheme:

Table 8: ACS712 and Photo Interrupter Wiring

ACS712 Pin	Connection Description	Microcontroller Pin
VCC	Power to Photo Interrupter	5V (Microcontroller)
GND	Ground of Photo Interrupter	GND (Microcontroller)
OUT	Data from ACS712 to microcontroller	A0 (Analog Pin)

Table 9: ACS712 and Photo Interrupter Wiring

Photo Interrupter Pin	Connection Description	Microcontroller Pin
VCC	Power to Photo Interrupter	5V (Microcontroller)
GND	Ground of Photo Interrupter	GND (Microcontroller)
OUT	Data from Photo Interrupter to microcontroller	Pin 2 (Digital interrupt Pin)

The final assembled and wired device is shown in Figure 9.

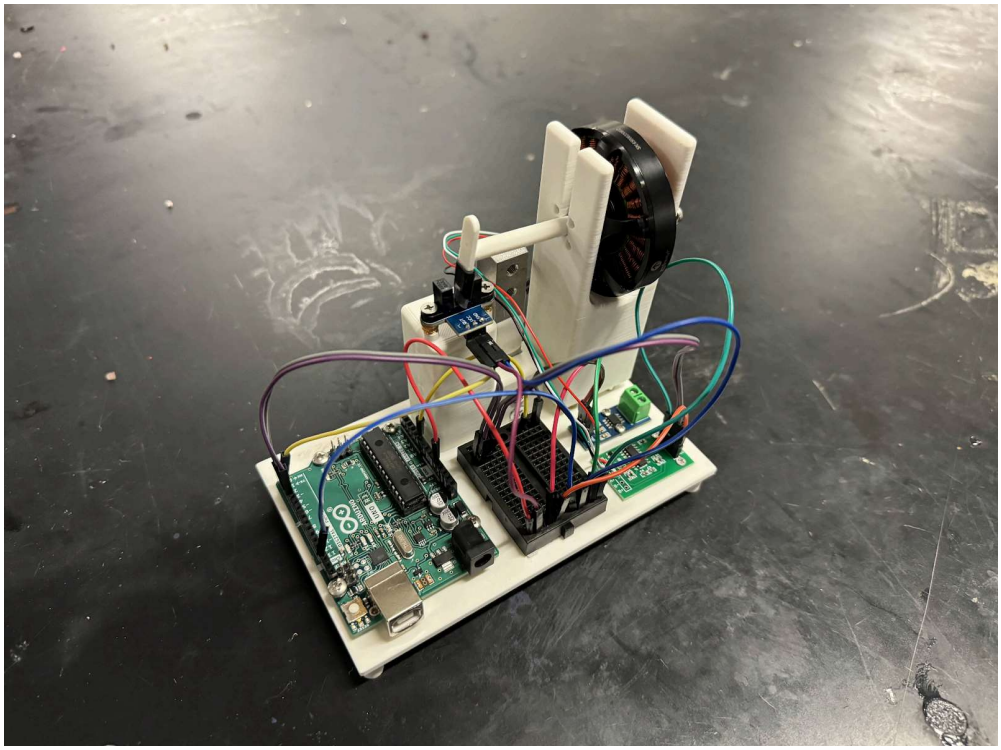


Figure 9: Assembled and Wired Device

Instructions to Use:

The dynamometer is designed for testing small, high-torque motors. Use the following instructions to operate the device safely and effectively.

Part a) Changing Out Motor Holders

To change motor holders, begin by turning off the power supply to ensure safety during the process. Use the appropriate hex wrench and/or screwdriver to loosen and remove the screws securing the current motor holder. Once detached, carefully lift the motor holder from its position and set aside. Select a compatible motor holder that matches the size of the motor you wish to test and align it with the mounting plate. Secure the new motor holder firmly using the provided screws, taking care not to over-tighten. Reattach any necessary wiring or components before proceeding.

Part b) Exchanging Load Cells

When exchanging load cells, start by powering off the system to reduce any potential risks/hazards. Disconnect the existing load cell by carefully unplugging its wires from the HX711 amplifier. Unscrew the load cell from its mounting point and remove it from the setup. Align the new load cell with the mounting location and fasten it securely in place. Re-connect the wires to the HX711 amplifier, ensuring they are attached to the correct pins as outlined in the wiring diagram provided in the manual.

Part c) Reading Data From Arduino

Steps to Upload Code and Read Data from Arduino

Github Resource Page: https://github.com/fadedfan/Prony-Brake-Dynamometer/tree/main

To upload the provided code to the Arduino UNO and set up the Serial Monitor to read data from the Arduino and tune the data, follow these steps in your IDE (VSCode is shown here):

1. Select the Correct Board and Port
2. Upload the Code to the Arduino
3. Open the Serial Monitor
4. Set the Serial Monitor Baud Rate (115200)
5. Monitor the Data

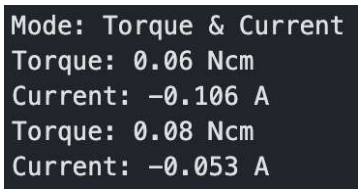
Once the Serial Monitor is open and the correct baud rate is set, you should start seeing data printed on the screen. This will include the RPM, torque, and current readings from the sensors connected to the Arduino, based on the mode you select (RPM or Torque/Current).

```
RPM, Torque, and Current Measurement
Waiting for signal...
Waiting for signal...
Waiting for signal...
```

You can input commands (e.g., **r** for RPM, **t** for Torque/Current) directly into the Serial Monitor's input box at the top and press **Enter** to switch between modes and see corresponding data.

If the input is **r (**RPM):

If the input is **t** (Torque/Current):



Instructions to Adjust Calibration Values for Load Cell and ACS712

To ensure accurate readings from your load cell (HX711) and current sensor (ACS712), it's essential to calibrate and adjust the MULTIPLIER value based on real reference measurements. Follow these steps to perform this adjustment:

Steps for adjusting the Load Cell Calibration (MULTIPLIER)

1. Prepare a Known accurate motor
2. Upload the Code
3. Open the Serial Monitor
4. Fix the known motor and the shaft on the Load Cell
5. Start the motor and observe the output

Copy the Data from the Serial Monitor over a period of time while the system is running. Alternatively, users can use a serial data logger (like the Arduino Serial Plotter or external logging software) to log this data to a file.

Depending on your system, users may expect a linear relationship between output torque and actual torque from the supplier. Users can fit a linear regression line to the data points to compare the actual behavior with the expected linear model.

Example data from a MN5006 Antigravity Type 4-6S UAV Motor KV450 is shown in Figure 10:

Propeller	Throttle	Voltage (V)	Current (A)	Power (W)	RPM	Torque (N*m)	Thrust (g)	Efficiency (g/W)
T-MOTOR P17*5.8" CF	40%	23.70	1.63	39	2481	0.10	501	13.02
	45%	23.67	2.21	52	2790	0.13	640	12.24
	50%	23.64	2.88	68	3103	0.15	784	11.5
	55%	23.61	3.71	88	3410	0.19	942	10.75
	60%	23.57	4.66	110	3689	0.22	1119	10.18
	65%	23.54	5.68	134	3947	0.25	1289	9.64
	70%	23.50	6.78	159	4203	0.29	1454	9.13
	75%	23.46	8.00	188	4442	0.32	1631	8.69
	80%	23.42	9.28	217	4664	0.35	1798	8.28
	90%	23.32	12.31	287	5105	0.43	2186	7.62
	100%	23.22	15.41	358	5491	0.50	2538	7.09

Figure 10: Sample Data (MN5006 Antigravity Type 4-6S UAV Motor KV450)

With the same current implemented on the motor, check the multiplier between the output torque from the dynamometer and the torque from a reliable source.

```
// Tuning multiplier for the loadcell
#define MULTIPLIER 2280.f
```

Troubleshooting (FMEA)

If users are experiencing issues where no data is being outputted or the Arduino is not taking in data, it's important to systematically check the potential causes. Below is an FMEA that helps identify common issues and guide troubleshooting for situations where the data isn't coming through properly.

Load cell not outputting Torque

Table 10: Load Cell Troubleshooting

Failure Mode	Possible Causes	Effect	Mitigation/Action
No output from HX711	Incorrect wiring between HX711 and Load Cell	No data from load cell	Double-check the connections to ensure the load cell's wires (Red, Black, White, Green) are correctly connected to the HX711 pins (E+, E-, A-, A+).
	HX711 not powered or misconnected	HX711 not operational	Ensure that the HX711 is connected to the power and ground rails (5V and GND) correctly. Confirm the microcontroller is powered and supplying the correct voltage.
	HX711 initialization failure in code	Arduino doesn't read the load cell	Check that the scale.begin() function in the code is called properly and verify the pin assignments for LOADCELL_DOUT_PIN and LOADCELL_SCK_PIN.
	Load cell damaged or defective	No readings	Test the load cell on a different system or try replacing the load cell to see if readings are outputted.
	Incorrect load cell calibration	Wrong data output	Adjust the MULTIPLIER value in the code according to the specific calibration for your load cell. Use a known weight to calibrate it.

Arduino Not Receiving Data (RPM or Current)

Table 11: Arduino Data Acquisition Troubleshooting

Failure Mode	Possible Causes	Effect	Mitigation/Action
No RPM signal detected	Incorrect wiring of the RPM signal pin	No RPM count or timing data	Verify the wiring of the interrupt signal (Pin 2 on the Arduino) from the RPM sensor. Ensure the sensor is outputting a pulsed lighting.
	Interrupt pin not configured correctly	Arduino does not react to RPM signal	Ensure the attachInterrupt() function is used correctly and the pin mode is set as INPUT_PULLUP for the interrupt pin. Check if the interrupt is enabled in code.
	Debouncing issue (frequent false interrupts)	Inaccurate RPM readings	Ensure the debounce logic in the interrupt service routine (ISR) is implemented correctly. Check the debounce delay (debounceDelay) for appropriate timing.
No current data from ACS712	Incorrect analog pin connection	No current readings	Check the connection of the ACS712 output pin to the correct analog pin (A0) on the Arduino. Verify the analogRead() function is correctly reading the data.
	Incorrect power supply to ACS712	ACS712 not operational	Verify that the ACS712 is correctly powered (5V) and ground is connected to the Arduino.
	Faulty ACS712 sensor	No current data	Replace the ACS712 sensor with a known working one to verify the sensor is functioning.
	Incorrect calibration of ACS712	Wrong current reading	Double-check the offset and sensitivity values in the code. Adjust ACS712_OFFSET and ACS712_SENSITIVITY according to your specific sensor and setup.

Arduino Code Issues

Table 12: Arduino Code Troubleshooting

Failure Mode	Possible Causes	Effect	Mitigation/Action
Arduino code not running correctly	Incorrect pin assignments in code	Data from sensors not read or output	Verify that the pin assignments in the code match the actual hardware wiring. Ensure all pins are declared correctly in the setup() function.
	Incorrect serial communication setup	Data not output to Serial Monitor	Check that Serial.begin(115200); is initialized in the setup() function. Confirm that the Serial Monitor baud rate is set to 115200.
	Missing or incorrect data processing functions	No output in serial monitor	Ensure functions like getIsrData() or readSensors() are being called within the main loop, and check that each sensor's data is being processed and printed.
	Unstable or blocking code (infinite loops, delays)	Arduino unable to process other tasks	Avoid using delay() in your main loop, use millis() for non-blocking timing. Review your code for infinite loops or blocking calls that might halt data collection.

Safety Guidelines:

BEFORE USE:

Please read this safety instruction in its entirety and check all mechanical and electrical components before utilizing the device. This document section will outline correct assembly procedures to ensure user and device safety during manufacturing and installation. This device was designed/fabricated with the intention of small, high torque motors. Please confirm motor outputs and weights before proceeding with using the device.

Part a) MECHANICAL:

1. Ensure that all components are securely fastened:
    a. Devices must be screwed into place securely (inspect for tightness)
    b. Screws should not be stripped; replace damaged screws as necessary
2. Verify that the motor is attached correctly to avoid misalignment or instability
3. Ensure the brake is properly secured to the shaft to prevent slippage or damage during operation

Part b) ELECTRICAL:

1. Double check all soldering connections for accuracy:
    a. Confirm that pins are soldered neatly with no loose or cold solder joints
2. Ensure all worse are plugged into the correct terminals as per the wiring diagram
3. Verify that exposed male wire connectors are not touching each other or any conductive surface to avoid short circuits

Part c) WHEN LOADING/CHANGING COMPONENTS:

1. Power down the system entirely before making any adjustments
2. Safely remove the shaft by loosening the corresponding screws or bolts
3. Carefully detach the motor, ensuring no undue stress on the wiring
4. Replace of modify mounts if necessary:
    a. Securely fasten the new mount and motor to ensure stability

b. Reattach the shaft and ensure all connections are tight before powering on the system

Part d) PROPER ENCLOSURE/ENVIRONMENT FOR ELECTRONICS:

1. Keep all liquids away from the device to avoid electrical hazards or damage to components
2. Avoid the presence of sharp objects near wires to prevent cuts or frayed connections
3. Ensure there is nothing nearby that could pull or snag worse, such as loose clothing or dangling objects

These guidelines aim to provide a safe and efficient operating environment for the Prony Brake Dynamometer, minimizing risks while ensuring the accuracy and reliability of test results.