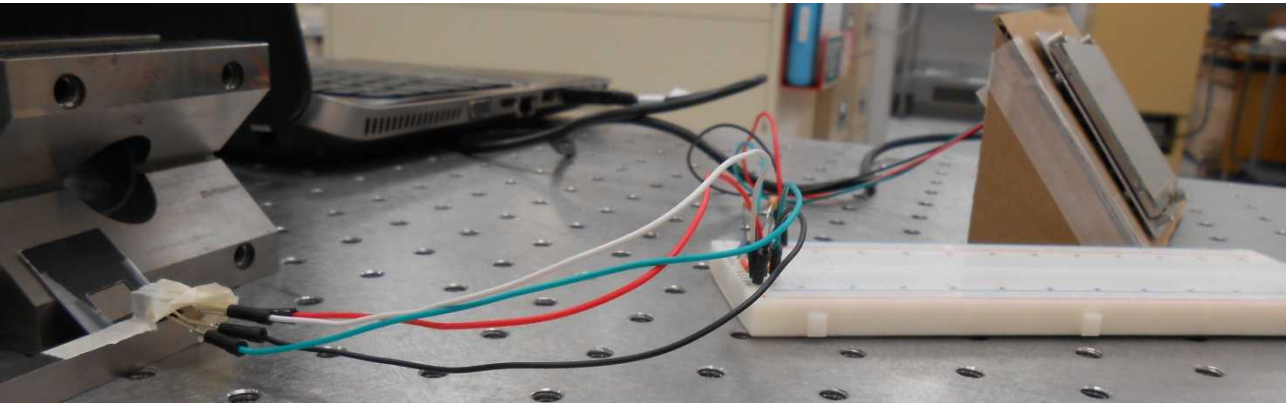


Exceptional service in the national interest

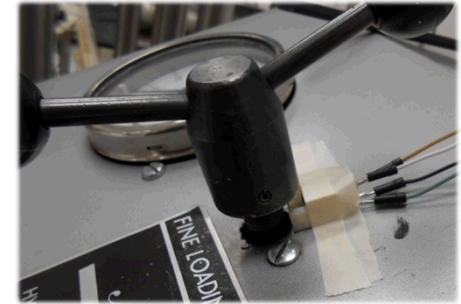
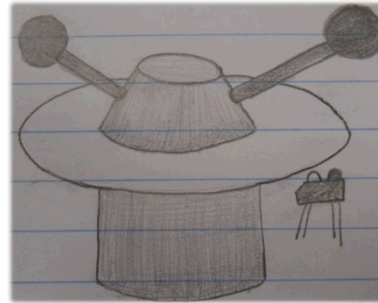


100K Force Calibrator Fine Loading Gauge

MIDN 2/C Kelli A. McQuiston

Outline

- What Metrology is
- Problem and Project Description
- Design Process
- Recommendations for the Future
- What I learned



What is Metrology?

Metrology is the Science of Measurement



I worked in the Length, Mass, Force Project at the Primary Standards Lab at Sandia's Albuquerque Location



About the Primary Standards Lab

- The mission of the Primary Standards Lab is,
“assuring the integrity of measurements throughout the entire National Security Enterprise by certifying standards and measurement equipment, developing measurement techniques and standards, and advancing the science of metrology”
 - Calibrates standards for all of NSE as well as outside customers
- Why are they important?
 - Highest level metrology organization for the Standards and Calibration Program in the NNSA Nuclear Security Enterprise (NSE) with many measurement accuracies that approach those of the National Institute of Standards and Technology.
 - **The qualification and reliability of every product in the Nuclear Weapon stockpile is based upon measurements traceable back to the PSL**

About the Length, Mass, Force Lab



- All primary mass and force standards are directly traceable to the SI (in our case, through NIST)
- The 1,000-100,000 lbf Force Calibrators best uncertainties is within 200 ppm
- **Utilizing the Fine Loading Screw on the 100,000 lbf Force calibrator the operator can get within one lbf of their desired load**
- Customers prefer lbf units, but traceability is through newtons

Project Description

- I worked to improve the ease of use of the 100,000 lbf Force Calibrator
- Before, the operator was never sure how much Fine Loading they had left which would lead to frustration when using the calibrator and fell *just* short or *just* over the desired load

So what I created is:

A sensing system that displays the percentage of space left to apply force



Design Process: Defining the Problem

- **What is the project supposed to do?**
 - Read out and tell the user how much fine loading they have left
- **How will you know if it is doing what you want it to?**
 - Manual testing
 - Testing the project read-out against actual value of the system
- **What could keep me from making it do this?**
 - Not getting the correct calibration values.
 - Movement of the system, changing the sensitivity of the IR sensor

Design Process: Specify Requirements

- Use the ezLCD-303 as the microcontroller and display of readings
- Use TCRT5000 IR sensors as distance sensors
- Must be able to readout amount of turns user has left in each direction
 - If simply reading rotations, must have a set start point
- Simple to use.
 - ☀ **The point of creating this system is to make force calibrations easier for the operator.**

Design Process: Brainstorm Solutions

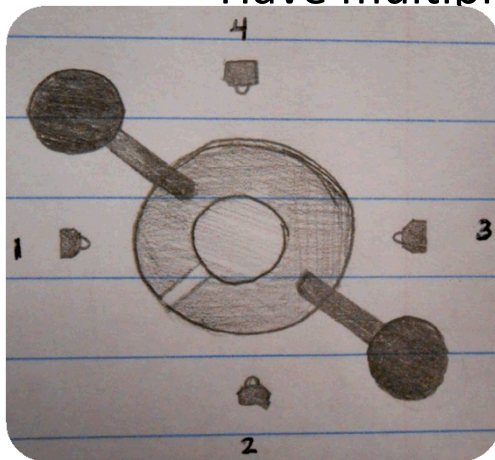
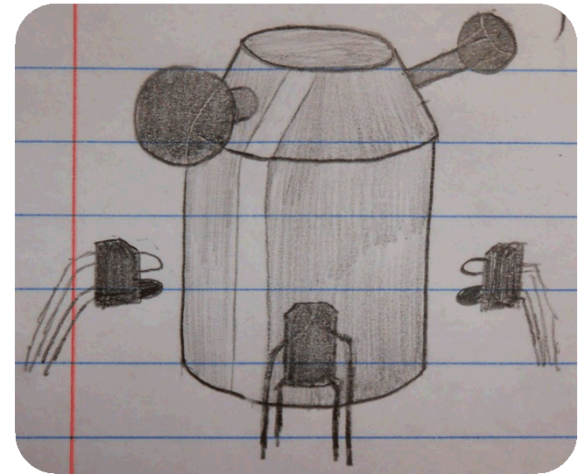
■ Rotation Reader

■ For initialization point

- Physical arm that shows where the middle is
- Requiring the user to set the fine loading to zero at beginning

■ For Counting Rotations

- Have a singular, double and triple stripe all 120 degrees apart
- Different colors could alter reflection intensity and therefore output
- Have multiple sensors and one line.

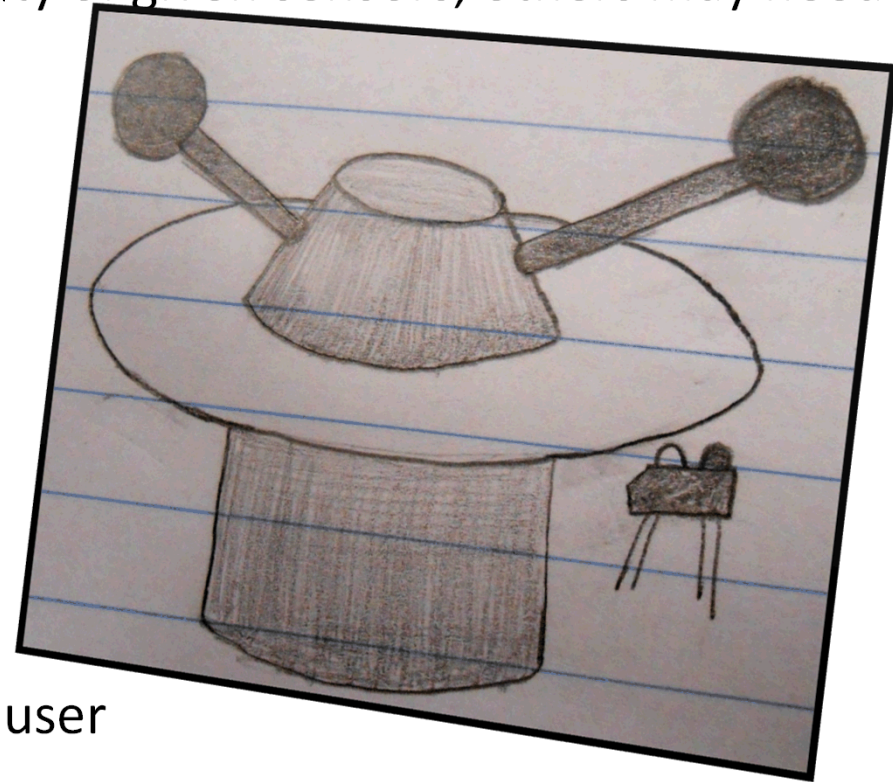


- Can figure out direction of rotation based off of which sensors read the line in which order
- Problem: Range of screw compared to the size of it

Design Process: Brainstorm Solutions

■ Single Height Read

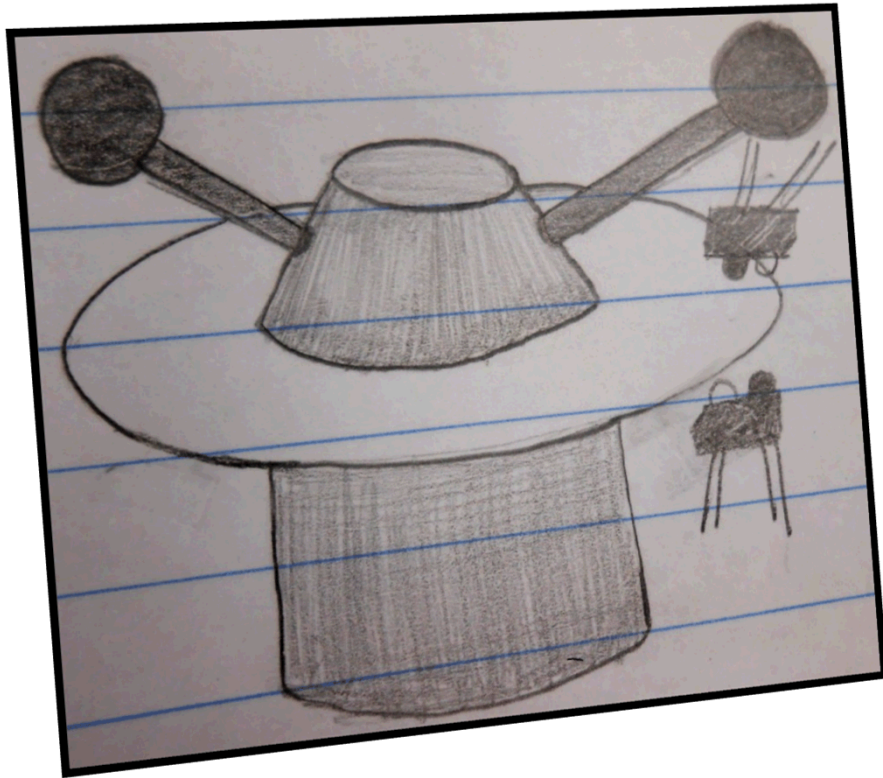
- Have 1-4 sensors at the base of the screw that 'look' up and face a white material that would be surrounding the screw.
- Depending on the range and sensitivity of given sensors, others may need to be purchased.
- Value readout would produce a percentage of how much room user has to rotate the fine loading up, or down.
- If more than one sensor is used, you can average out the readouts to get an even more accurate value for the user



Design Process: Brainstorm Solutions

■ Double Height Read

- Have a white material encircling the screw, that would have a sensor above facing down towards the material, and a sensor below facing up, towards the material



- Possible solution if range of sensors is very small and poor
- Based on value readouts from both sensors, or based off of a calculated/updated variable of the height of the screw, you could based which sensor you took the read out value from.

Design Process: Evaluate Solutions

Design Ideas	Rotation Reader	Single Height Read	Double Height Read
Simple/User Friendly	Medium	High	Low
# of Sensors Needed	2-4	1-4	2-4
Initialization Point?	Yes	No	No
Additional Cost	None	Possibly	None
Interference to User	Medium-Low	Low	Medium
Accuracy	Medium	Medium-High	Medium-Low

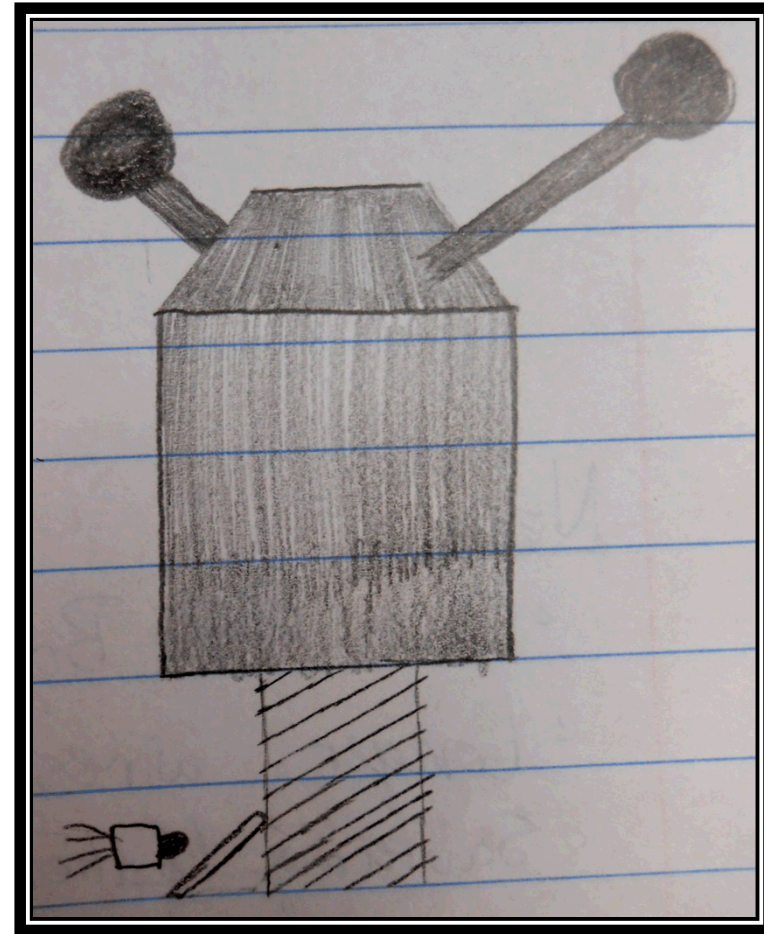
Chose

Design Ideas	Rotation Reader	Single Height Read	Double Height Read
Simple/User Friendly 25%	4	5	3
# of Sensors Needed 5%	1	3	4
Initialization Point? 15%	1	5	3
Additional Cost 10%	5	1	5
Interference to User 25%	3	4	1
Accuracy 20%	5	2	1
	if I can get it right	Questionable	
Total Desirability	69%	73%	47%

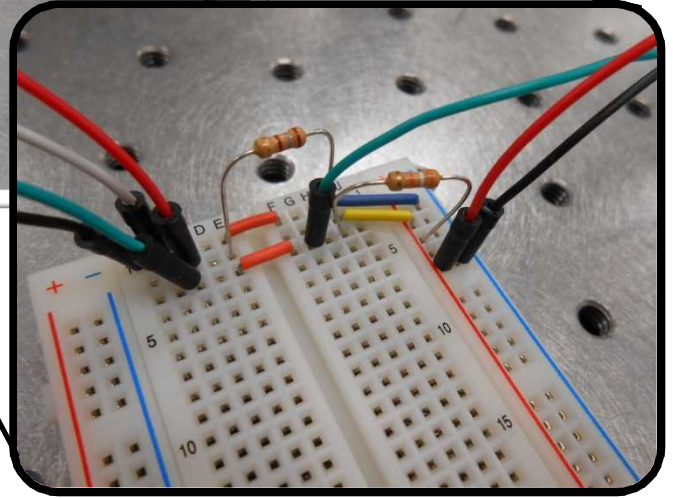
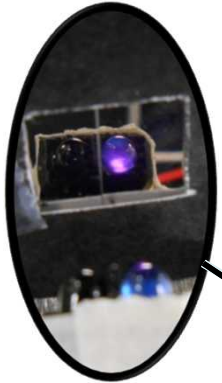
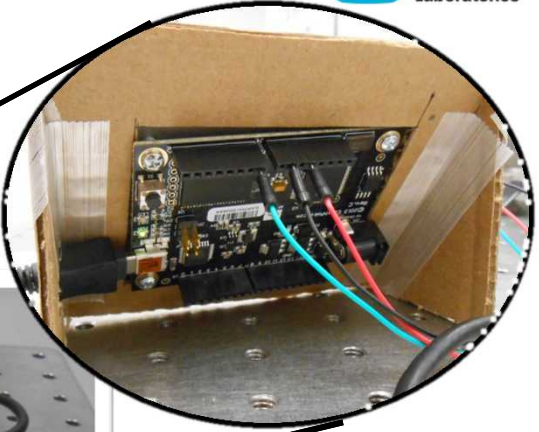
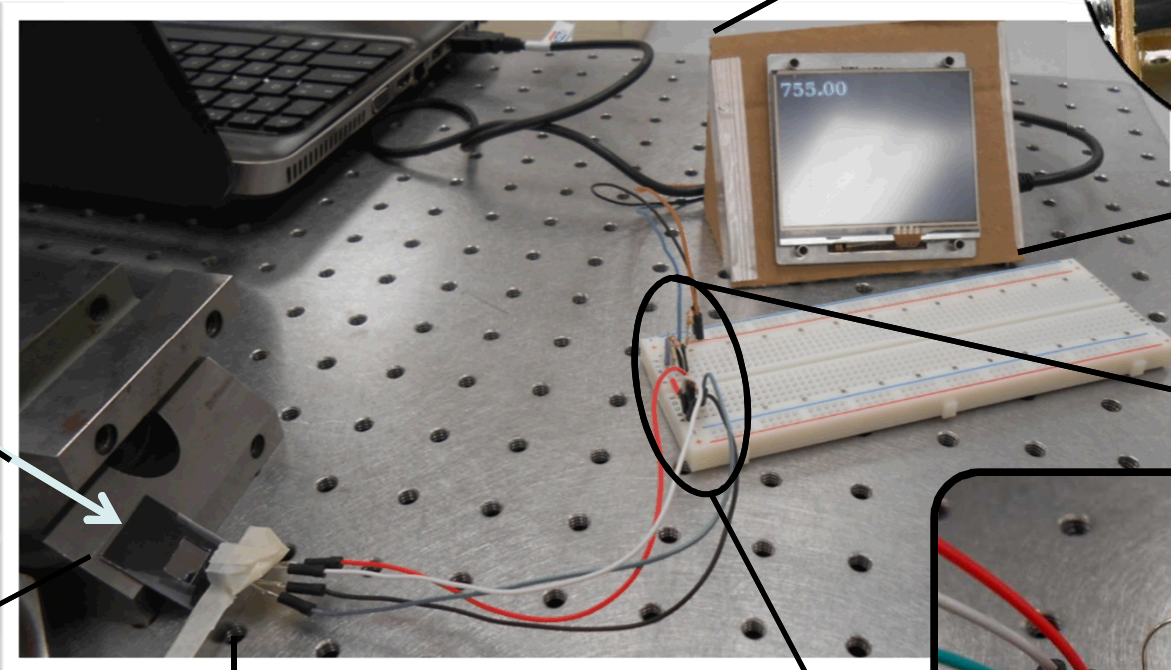
Out of 5, 5 being high, 1 being low

Design Process: Choose

- With the help of my Project Manager, Hy Tran, we further developed the design of the Single Height Reader in order to lessen the interference of the system to the user
- To do so, we incorporated a mirror
- After testing final product; with the mirror, the sensor can detect the overhang of the screw by itself, with no extra reflective disc or even white paint on the bottom

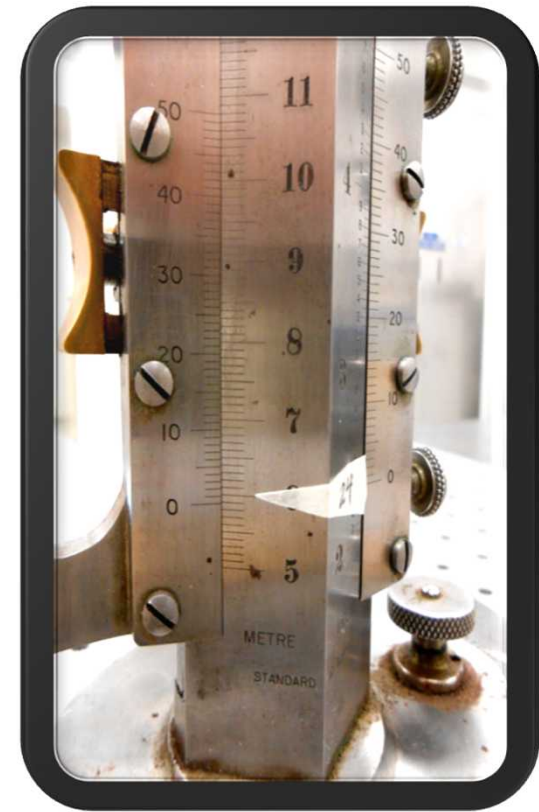
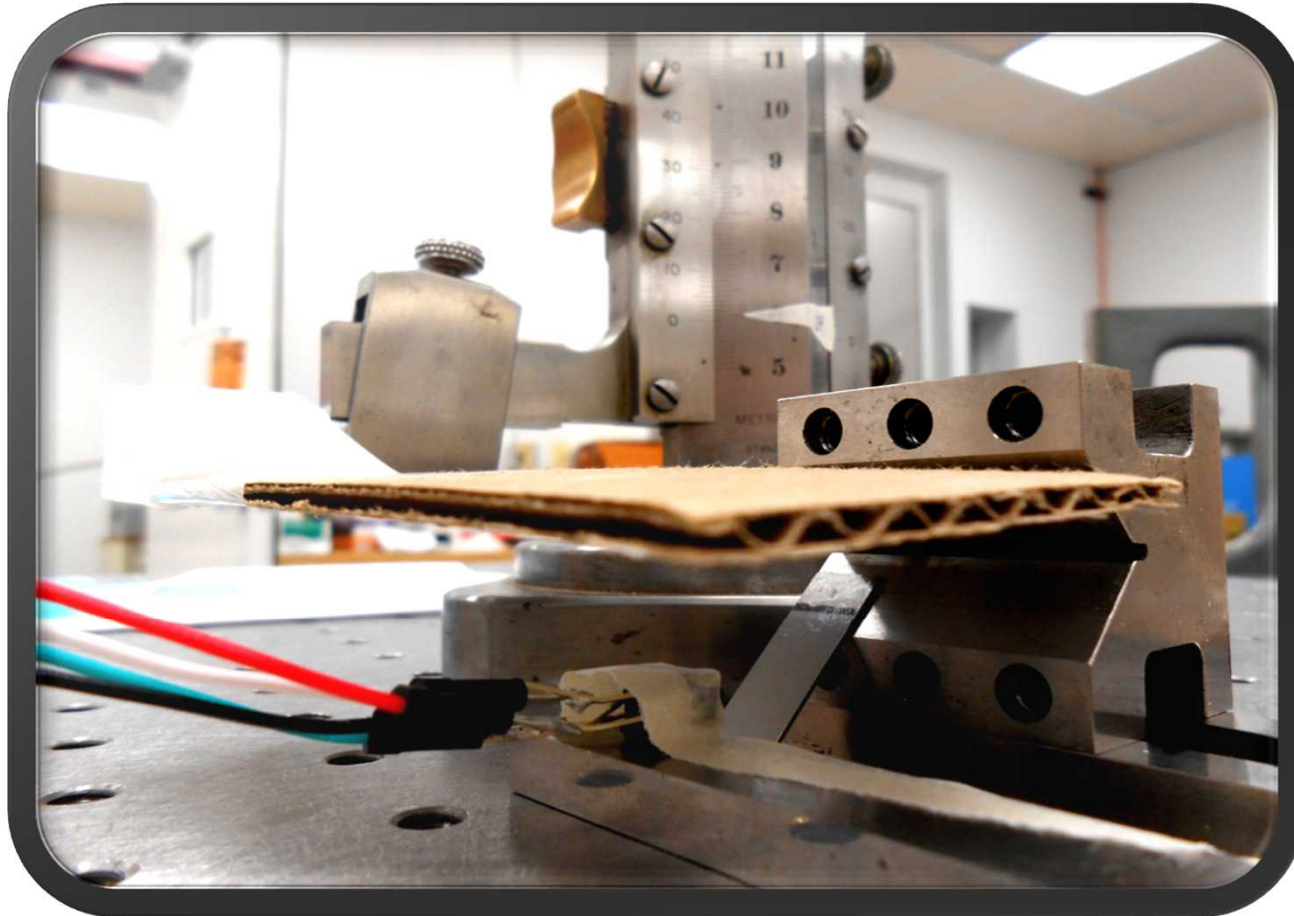


Design Process: Prototype



Prototype Height Calibration

I was able to get calibration values from 15-44mm above the sensor to get a sense of the analog output pattern and practice coding the analog output to the actual height relationship



Coding: Raw Value to Calibrate

```
#include <ezLCDLib.h>
#include <SoftwareSerial.h>

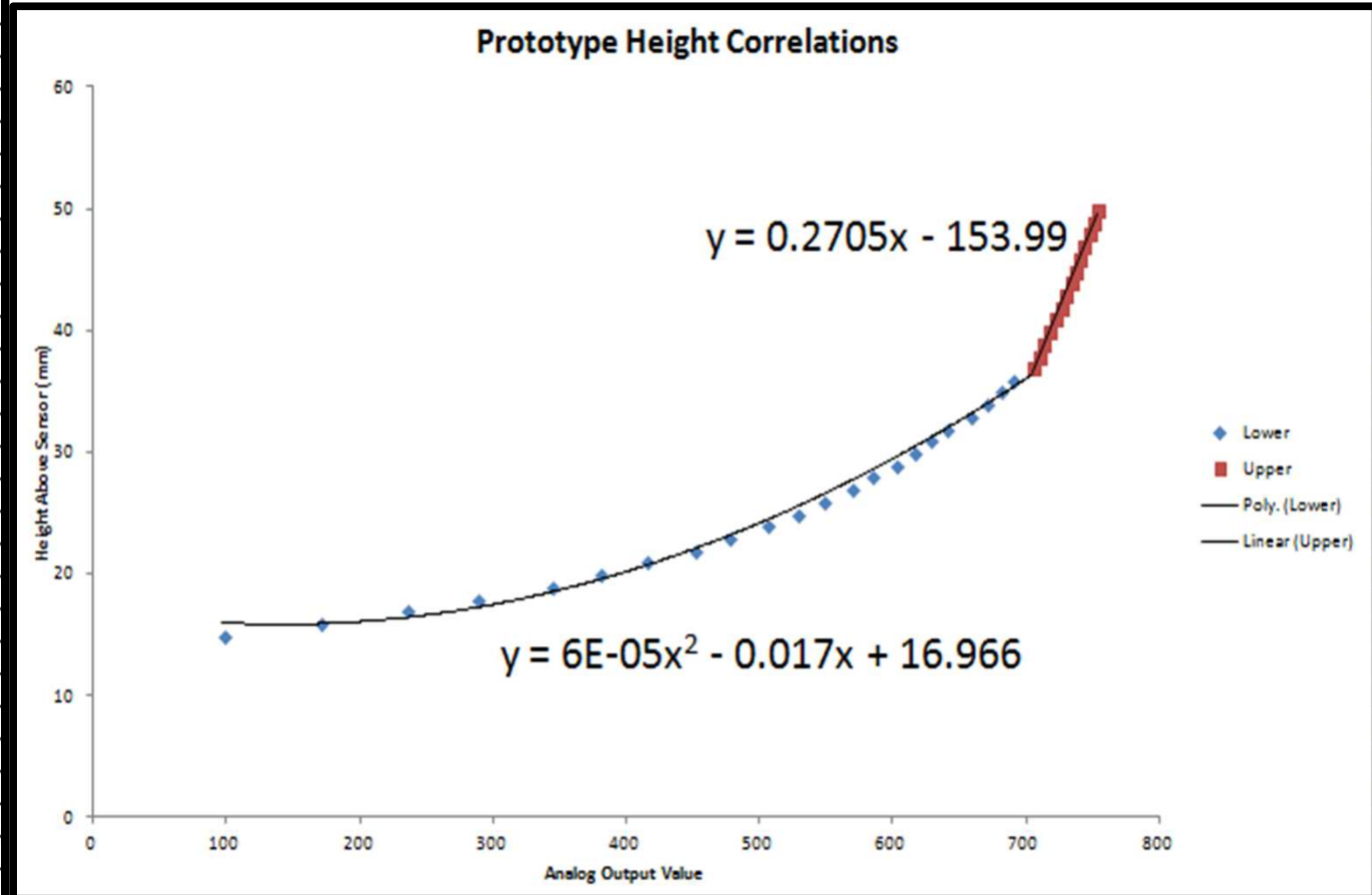
ezLCD3 lcd;      // create lcd object
float ir = A0;   //Reads analog input from pin A0
int rval = 0;

void setup() {
  lcd.begin(EZM_BAUD_RATE);
  lcd.font("serif72");      //Use Large font
  lcd.cls(BLACK, WHITE);
}

void loop() {
  lcd.cls();
  rval = analogRead(ir);    //Store Analog value
  lcd.print(rval);          //Print out stored value
  delay(750);                //Print value every .75 seconds
}
```

Graphing Calibration

Height (mm)	Analog ReadOut
8	33
15	97
16	169
17	235
18	287
19	343
20	380
21	414
22	450
23	477
24	505
25	528
26	548
27	568
28	583
29	601
30	615
31	628
32	639
33	658
34	669
35	680
36	689
37	705
38	710
39	713
40	717
41	721
42	726
43	729
44	734



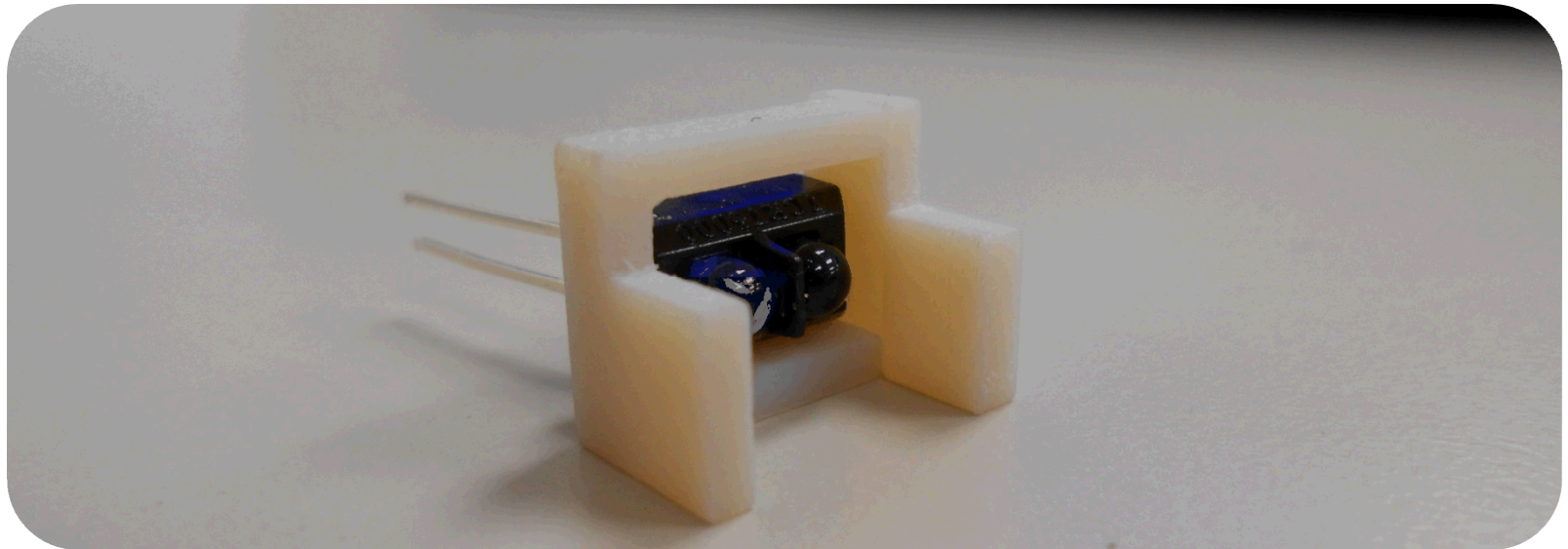
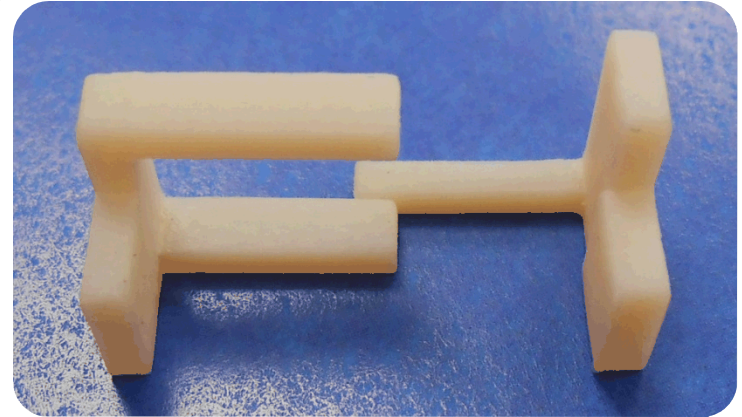
Coding: Final Prototype Code

```
rval = analogRead(ir);
lower = (.00006 * pow(rval, 2.0)) - (.017 * rval) + 16.966; //lower poly fit equation
upper = (.2705 * rval) - 153.99; //upper linear fit equation
if (rval <= 700){
    percent = (lower / 24) * 100; //if the raw value is in the lower range, calculate with lower equation
    eheight = lower;
}
else if (rval > 700){
    percent = (upper / 24) * 100; //if the raw value is in the upper range, calculate with upper equation
    eheight = upper;
}

lcd.println("    You have");
lcd.print("    ");
lcd.print(percent); //estimated amount of space left to go down
lcd.println("% of");
lcd.println("    Fine Loading");
lcd.println("    left");
lcd.print("    ");
lcd.println(eheight); //For testings sake, print estimated height from sensor as a check
if( percent > 97 ){
    lcd.println("CLOSE TO TOP!"); //If above a certain percentage, warn user they are close to top limit
}
else if(percent <= 7 ) {
    lcd.println("CLOSE TO BOTTOM!"); //If below a certain percentage, warn user they are close to lower limit
}
delay(750);
```

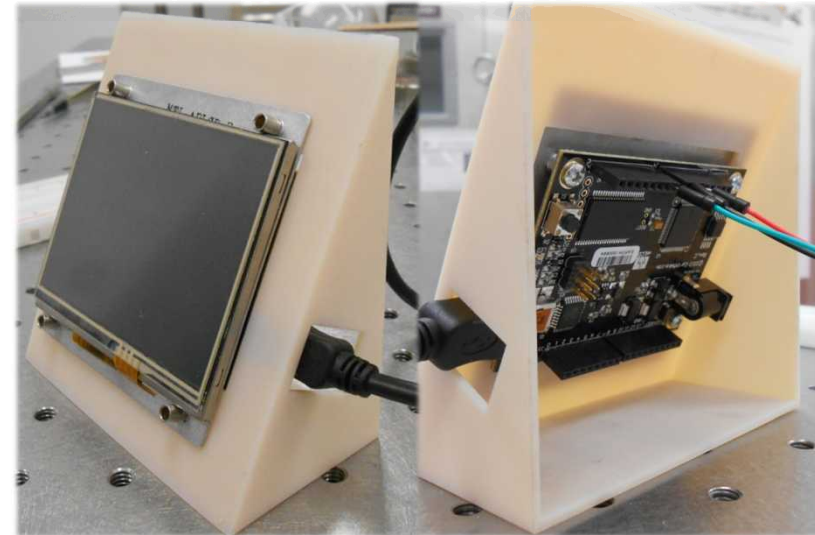
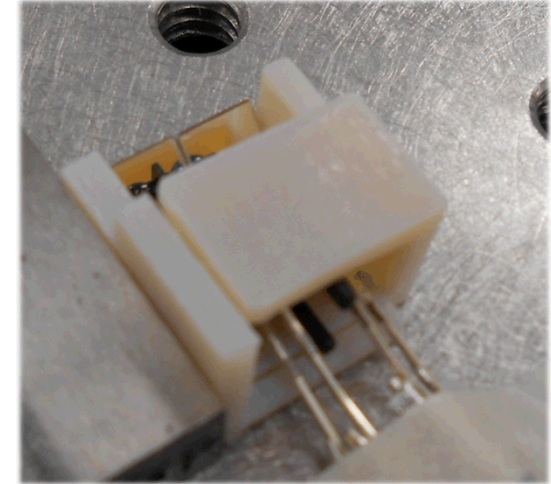
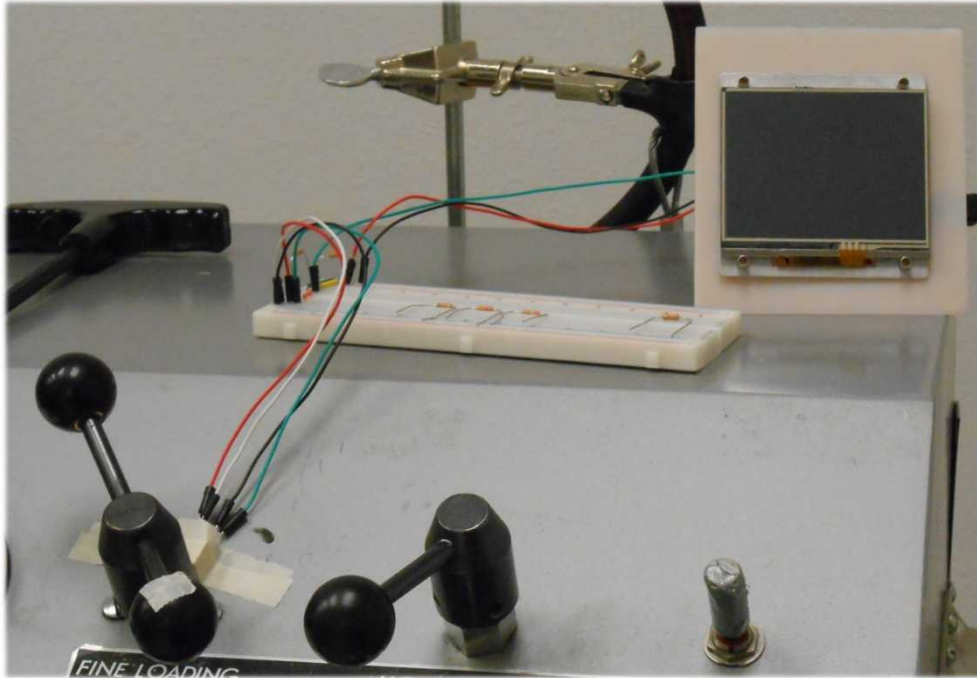
Design Process: Prototype

- The most important part of the system is ensuring that the IR Sensor and mirror stay a set distance from each other
- I worked with Celia Flicker to develop something that would be able to hold both the sensor and mirror



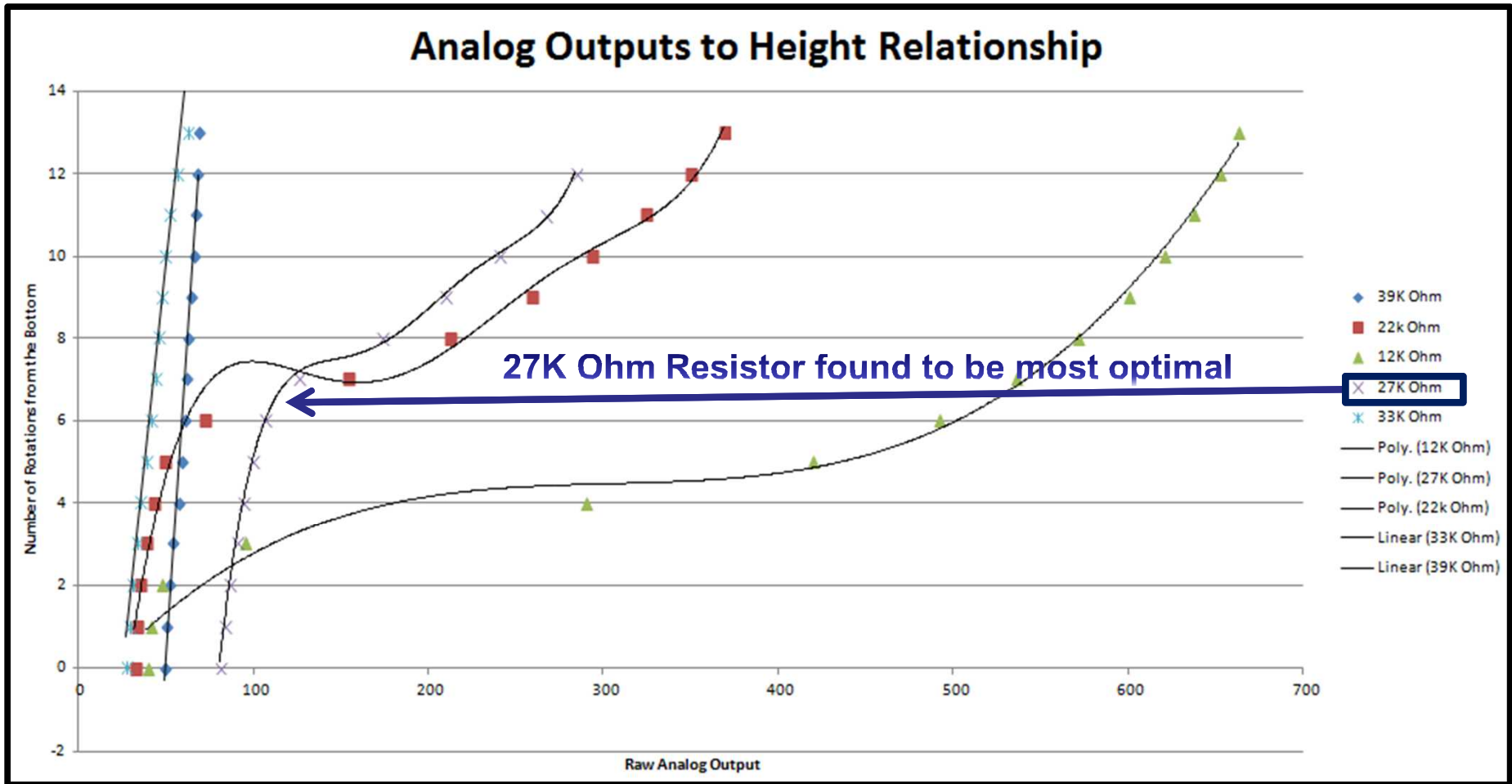
Design Process: Refine

- For the second and final 3D Printed design we extended the back so that the IR sensor would be held more securely
- This also made the overall design sturdier

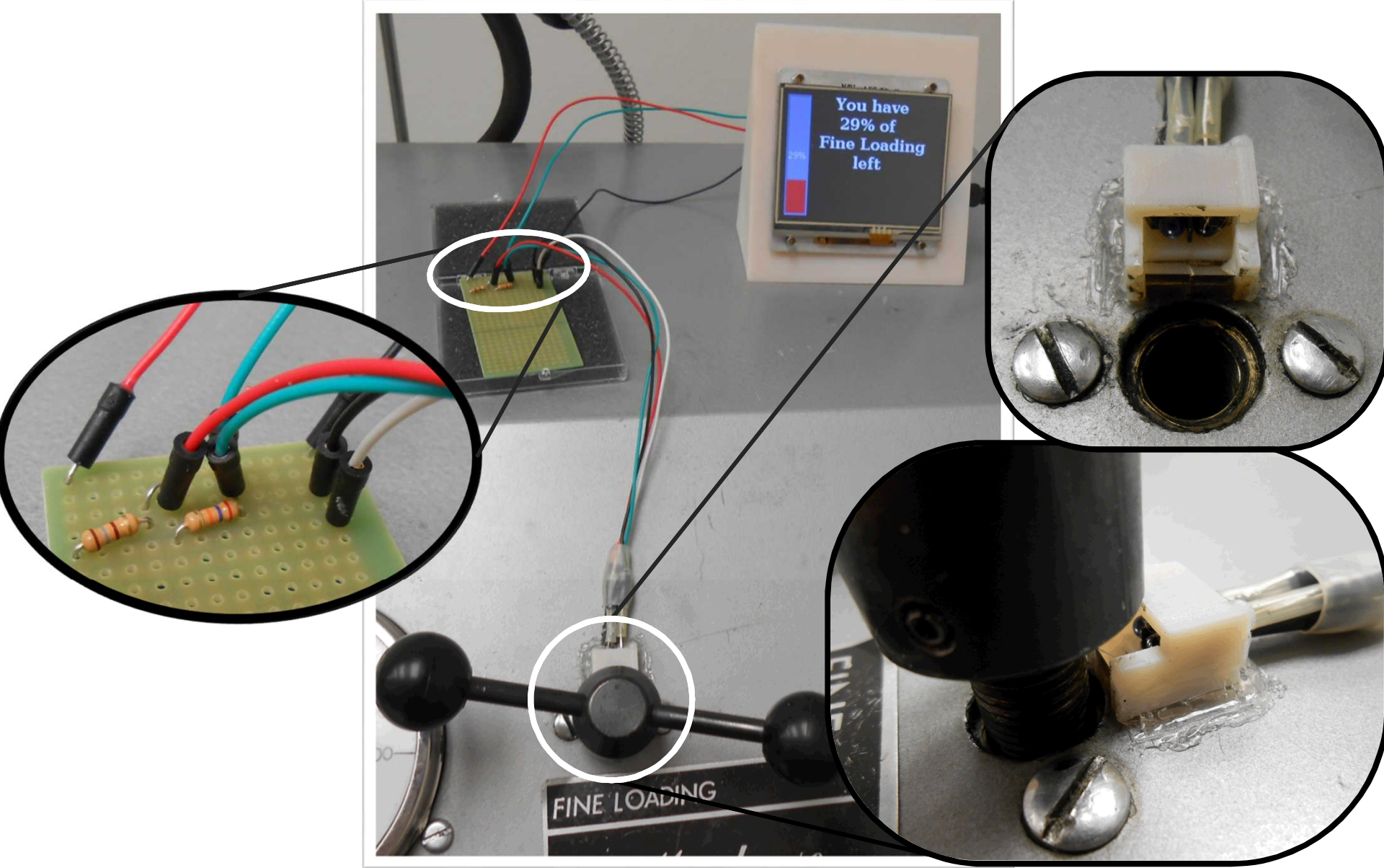


Design Process: Refine

Needed to find the most optimal resistor for the range of the system

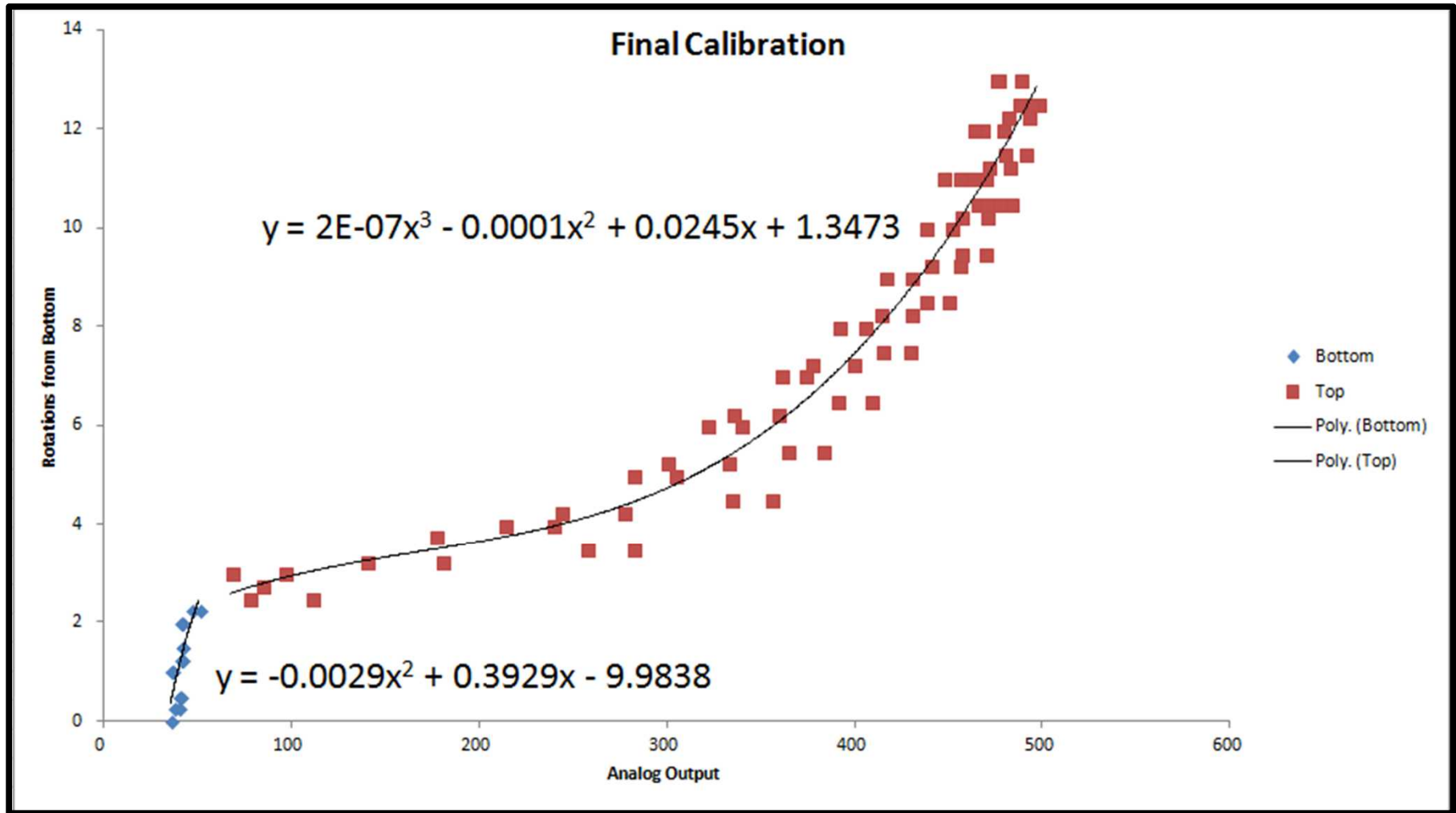


Design Process: Implement



Final Calibration

For the final calibration of the system I did 3 trials in order to capture the slight range of values at each location.



Final Code

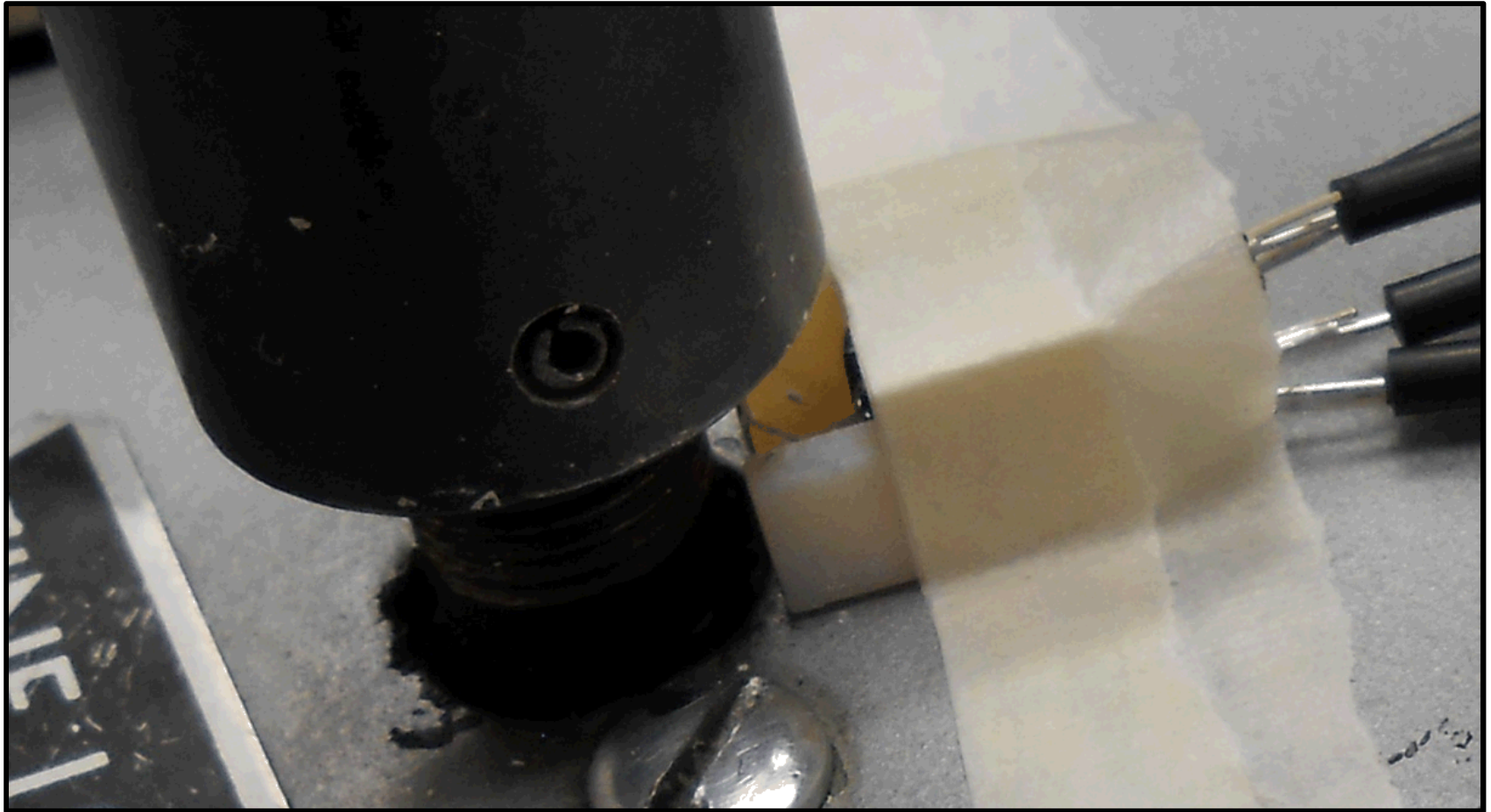
```
lcd.progressBar(1, 5, 5, 45, 230, 3, 10, 100, 1, 1);           //Progress Bar on left of screen
rval = analogRead(ir);                                         //Store Analog Value in rval variable
top = (.0000002 * pow(rval, 3.0)) - (.0001 * pow(rval, 2.0)) + (.0245 * rval) + 1.3473; //Upper Equation
bottom = (-.0029 * pow(rval, 2.0)) + (.3929 * rval) - 9.9838; //Lower Equation
if (rval <= 50){                                               //If in lower range, find percentage using lower equation
    percent = (bottom / 13) * 100; //Out of 13 because it's approximately 13 rotations to get to top range
}
else if (rval > 50){                                           //If in upper range, find percentage using upper equation
    percent = (top / 13) * 100;
}
lcd.wvalue(1, percent); //Show percentage of force left on Progress bar
```

```
lcd.println("        You have");
lcd.print("        ");
lcd.print(percent); //Prints the amount of room left to add force
lcd.println("% of");
lcd.println("        Fine Loading");
lcd.println("        left");
lcd.print("");
if( rval >= 450 ){
    lcd.println("CLOSE TO TOP!");
} //Because of variation of values, proximity messages are based on raw values
else if(rval <= 40 ) {
    lcd.println("CLOSE TO ");
    lcd.print("BOTTOM!");
}
delay(750); //Repeat loop every .75 seconds
```

Design Process: Test



Recommendations for the Future



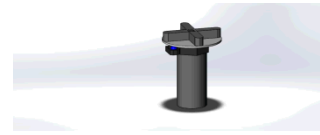
Finding or requesting a 5-7mm longer screw for the Vernier Pump from Morehouse so the operator no longer has to sacrifice that last 5mm of Fine Loading Force.

Recommendations for the Future

- Because the ezLCD has touch screen capabilities, alter the program to where the operator could choose whether to run the Raw Value program for calibration or program for performing tests
 - I will be able to do before I leave.
- Finding the amount of Force the Fine Loading Screw can add at each significant Load
- A code can then be written that would prompt the user to enter their desired load and then, if it had the proper inputs, could calculate and tell the operator when they are able to reach their desired load using the Fine Loading Screw.

Recommendations for the Future

- Pictured to the right is the Fine Loading Screw of both the 10,000 lbf and 30,000 lbf Force Calibrators
- Add a reflective disk to the bottom of the screw
- Attach an IR Sensor to the side, facing upwards
- Calibrate the system
- Slightly alter the already written code
- By simply changing the program and which sensor is plugged into the Microcontroller you could use the ezLCD display for all three Force Calibrators



What I Learned

- Had to teach myself how to use the ezLCD Arduino unit
 - Updating Newest Driver on it
 - Code, New Library of functions
- Optical Sensor
 - Learning circuitry and Logic
 - Range of Provided Sensors
- Working with Celia Flicker to create parts to hold the sensor all together in one piece
- Learned through experience the real deal, actual, design and development process.

Acknowledgements

I would like to extend a big thank you to all those who helped me complete my project:

- Hy Tran
- Celia Flicker
- Ruth Smelser
- Rick Mertes
- Sam Ramsdale
- Jeremy Gray
- Brian Perdue

