

Double Transducer: Opacity to Rotation

Evelyn Bang, Aiden Magee

Table of Contents

[Final Images](#)

[Narrative Description](#)

[Progress Images](#)

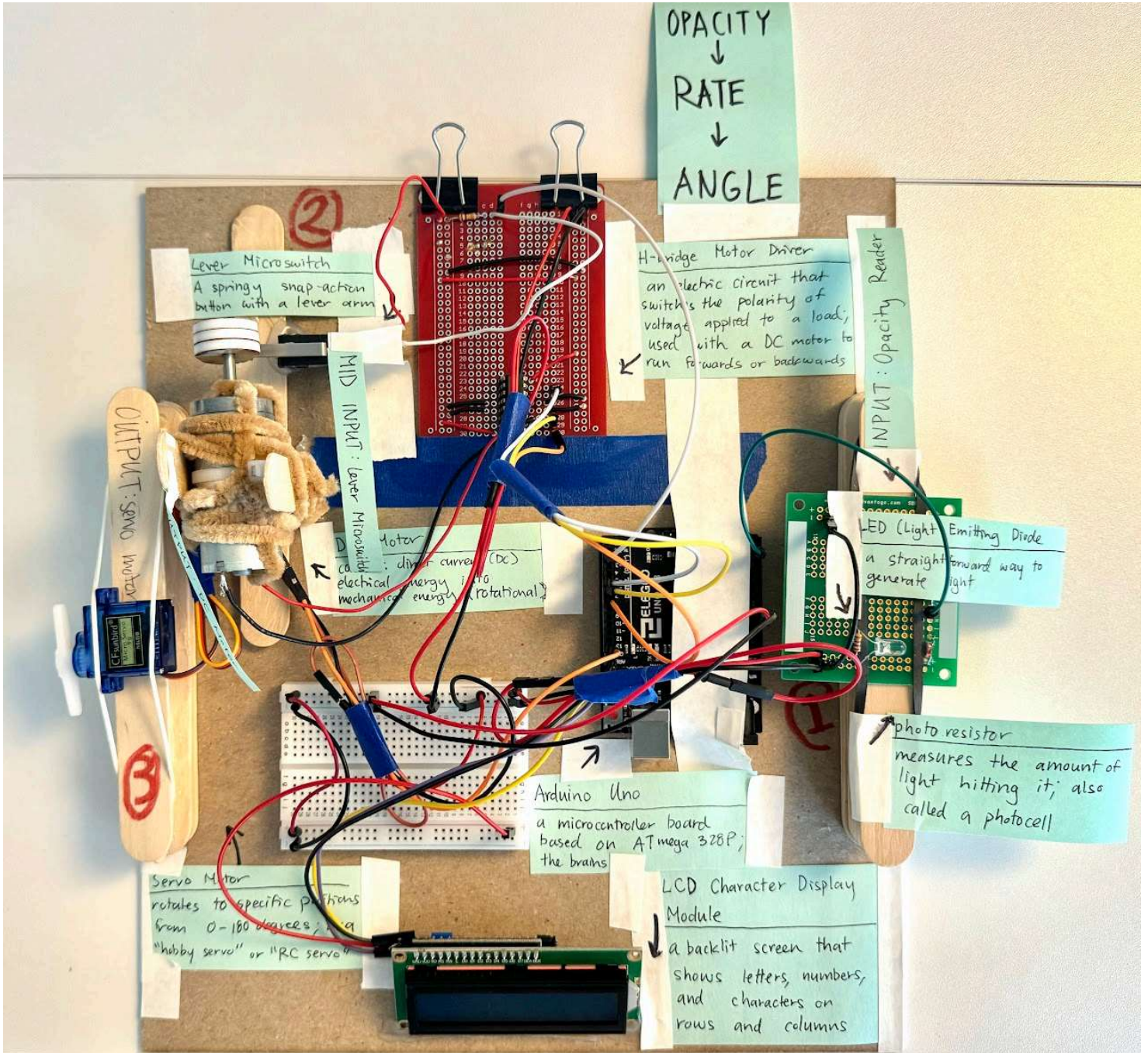
[Discussion](#)

[Functional Block Diagram and Electrical Schematic](#)

[Code](#)

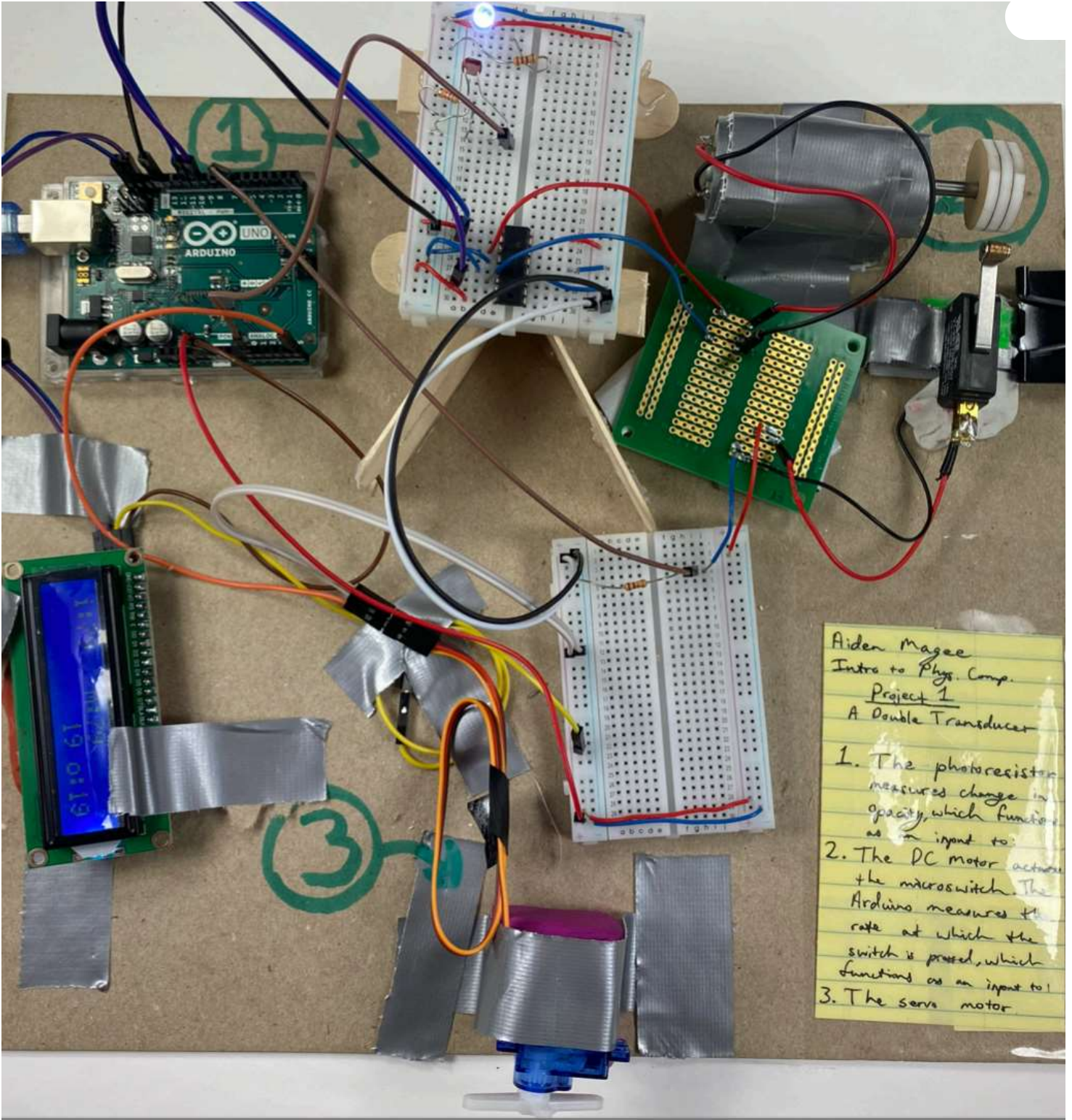


Final Images



Evelyn's Double Transducer

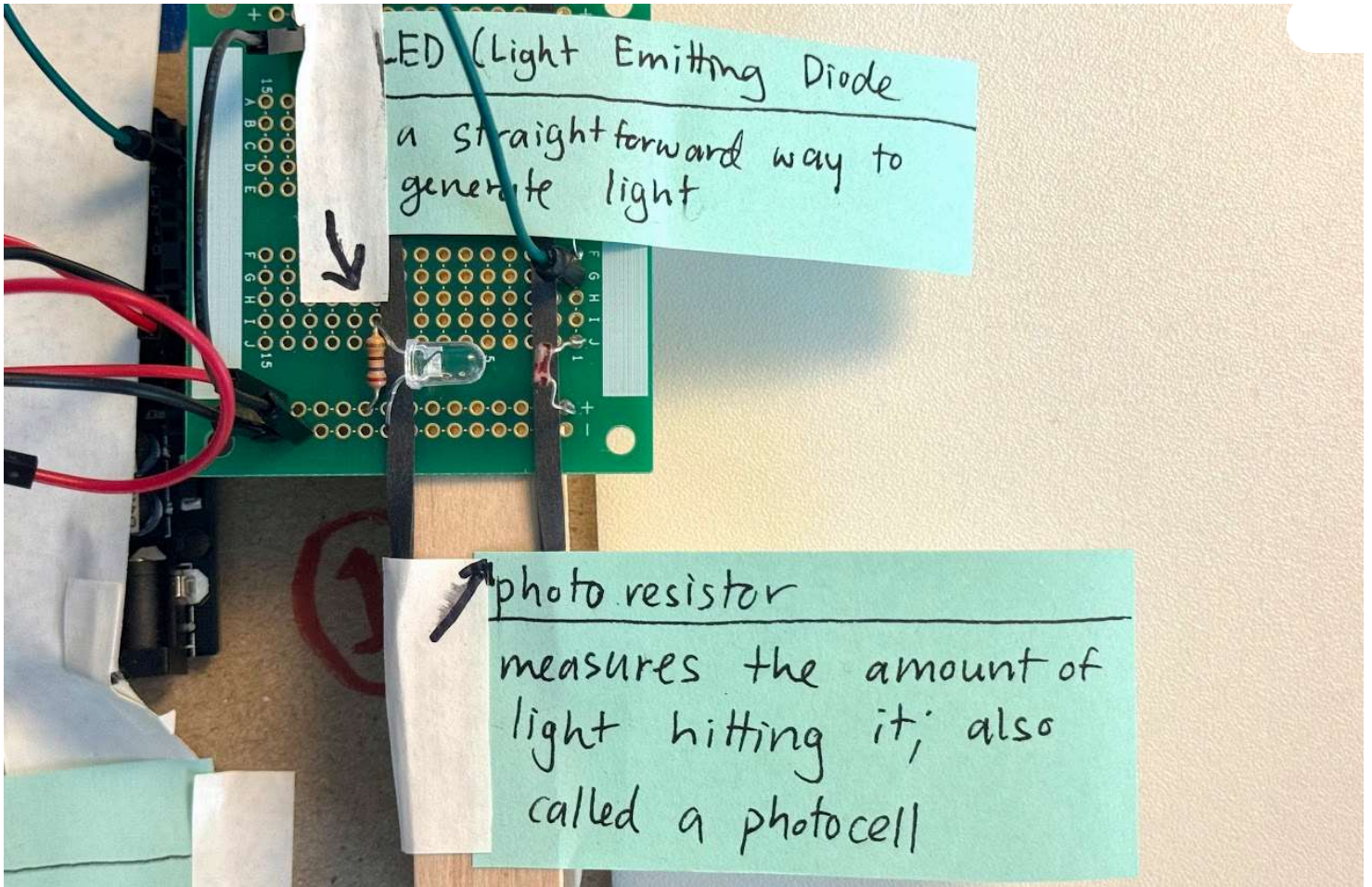




Aiden's Double Transducer

Detail Photos

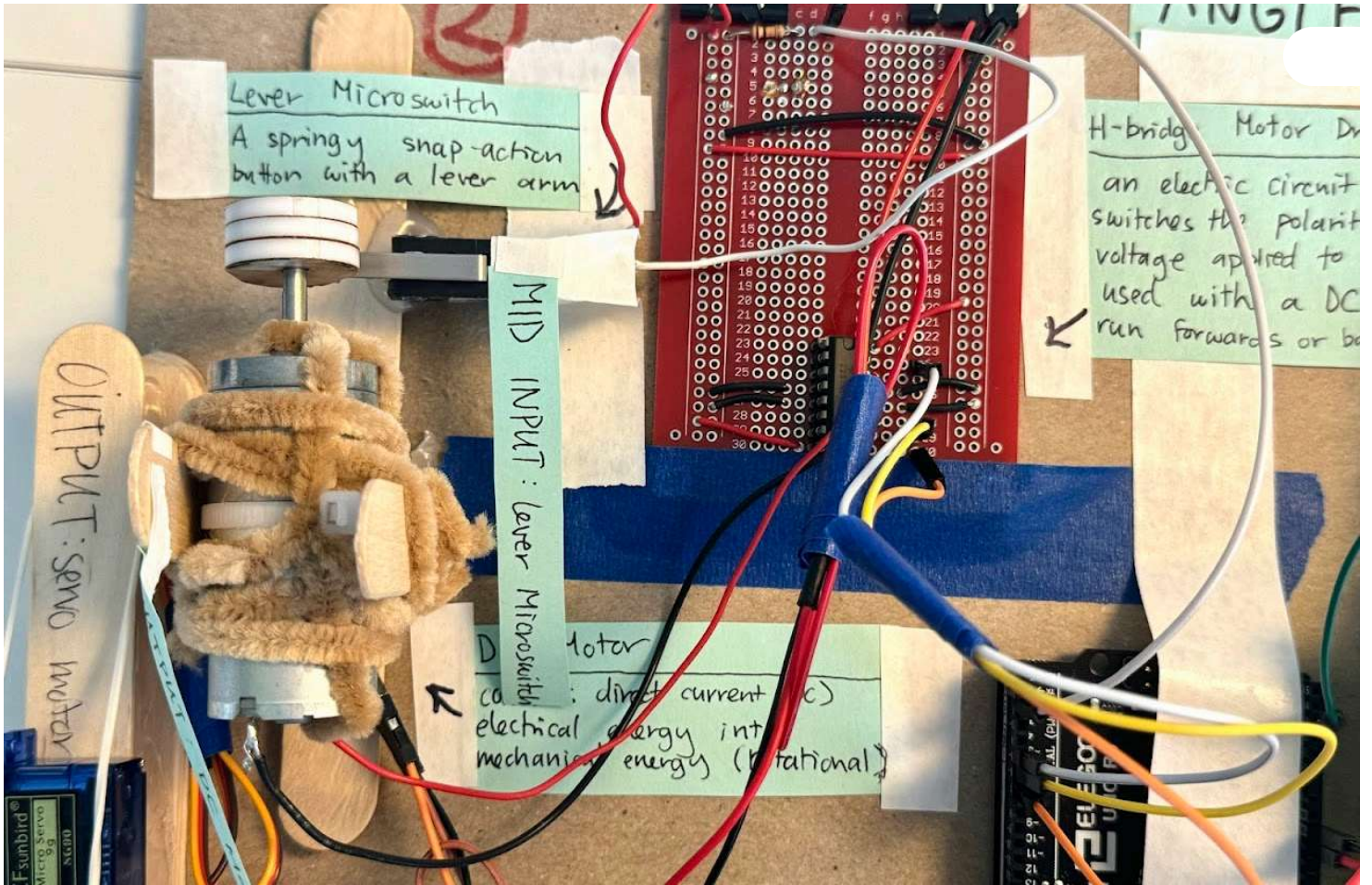




Input: Opacity

The photoresistor receives the opacity of a thin object by showing how dark the light has gotten compared to when there are no obstructions in between the LED and the photoresistor.

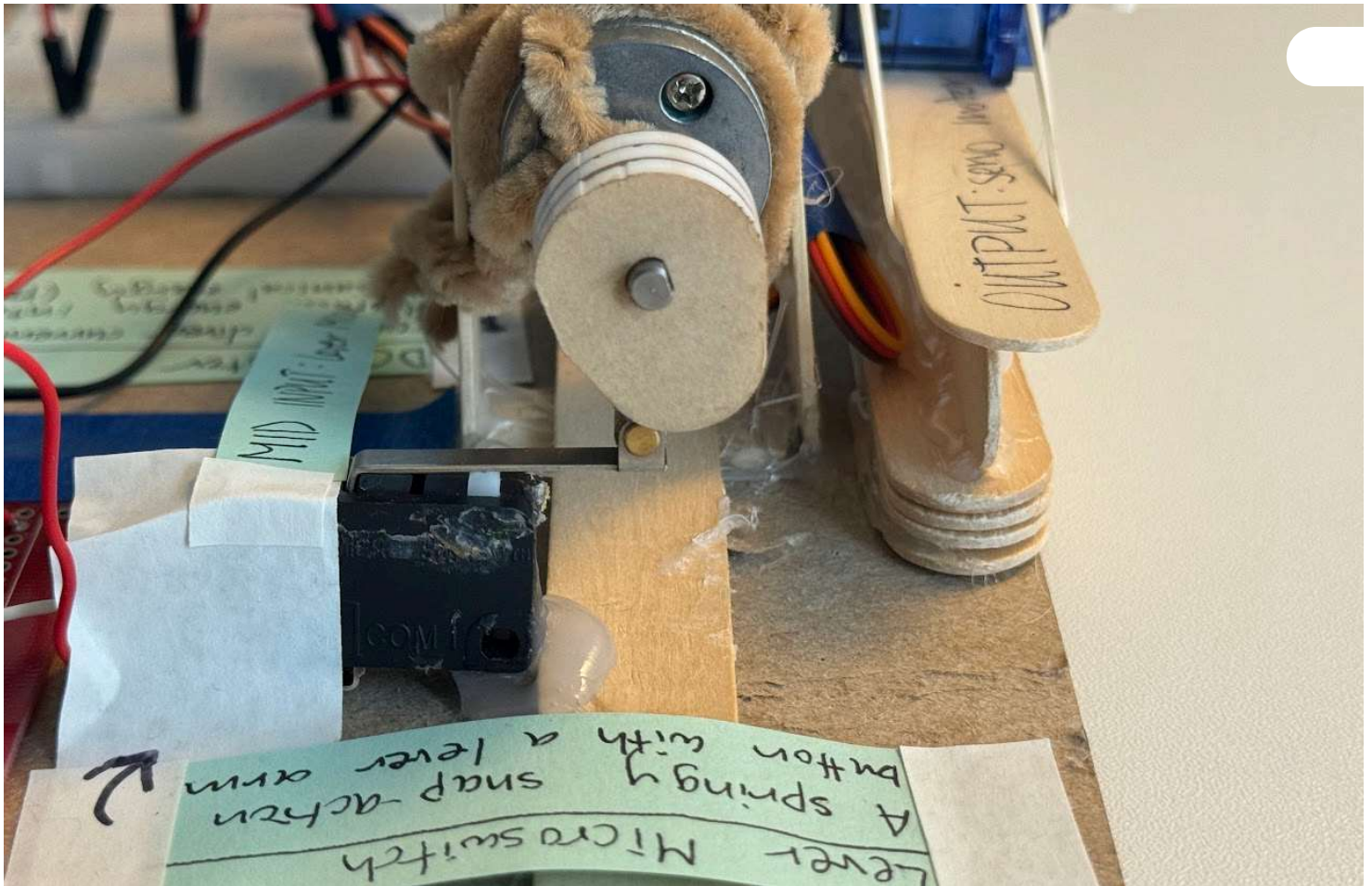




Middle Step: DC Motor Spin Rate

The DC motor spins faster as the light is not as obstructed. The higher the light level, the faster the DC motor spins and clicks the lever microswitch with a CAM, detecting the rate of rotation.

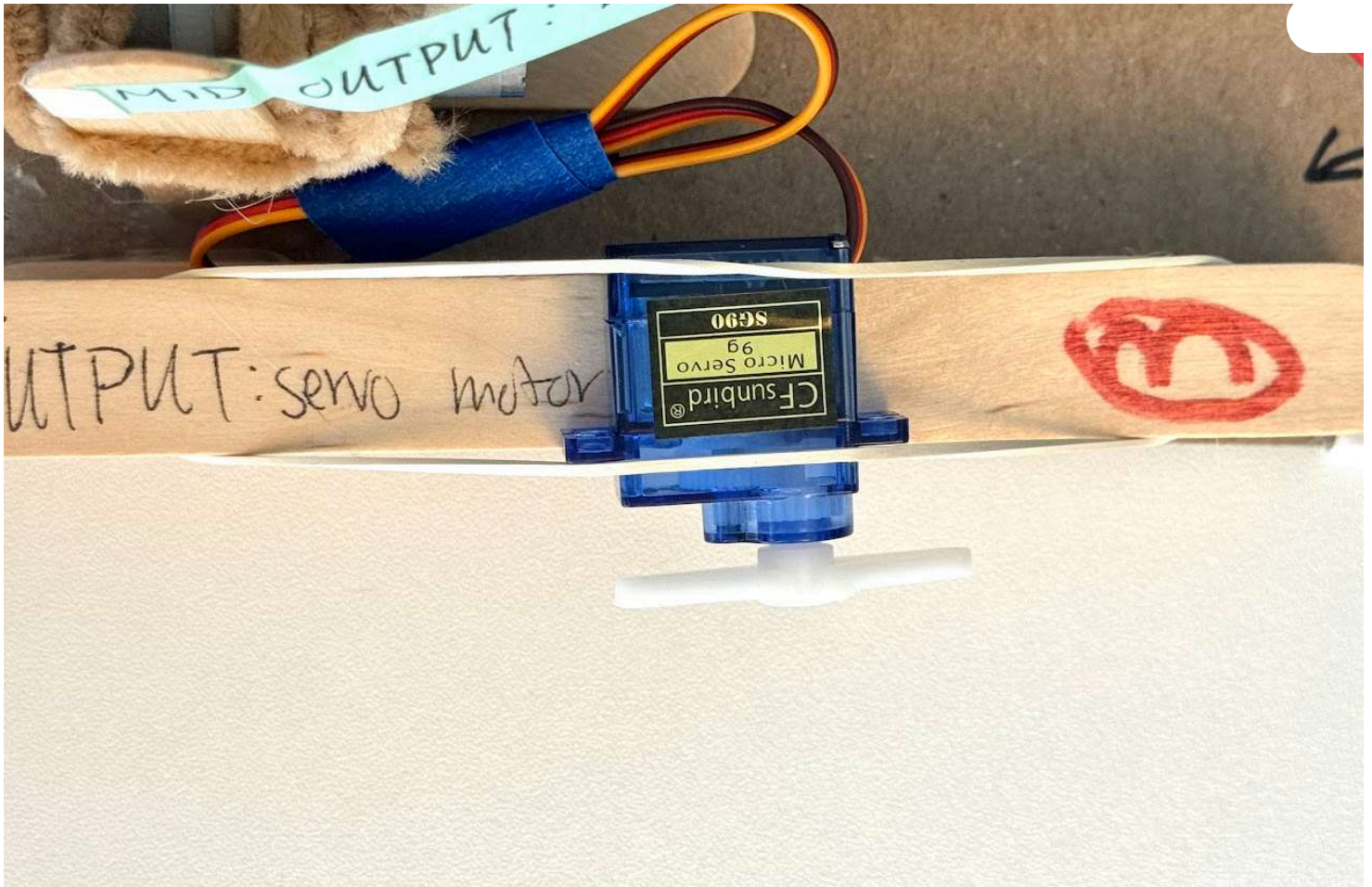




Middle Step: DC Motor Spin Rate

An alternative angle of the DC motor, CAM, and lever microswitch.






Output: Servo Motor

The servo motor receives the rate of rotation and then translates that into an angle between 0 and 170. The higher the rate of rotation, the higher the angle.

Final Project Movies



 Evelyn_Bang_Project_1.mp4

Evelyn's Video:

"Hello, my name is Evelyn Bang. This project shows the change from opacity to rotational rate to a positional rotation. The input is detected by a photoresistor that has an LED pointing at it, allowing the level of light reaching the photoresistor to be received. The DC Motor then responds to the light level, turning slower as there is less light. The DC Motor has a CAM attached to it, allowing the lever microswitch to be pressed, detecting the rate of rotation. Afterwards, the servo motor changes its angle, responding directly to the rate of the DC motor's rotation."



 Aiden_Magee_Proj1_movie.mp4

Aiden's Video:

When I pass an opaque material through my opacity-measuring mechanism, my DC motor slows down, causing my switch to actuate at a lower frequency. My LCD screen reflects this. Unfortunately, my Servo motor was not functioning properly. I believe this was due to a power issue, because when I tried to hard code an angle to set my Servo at, my Servo still moved sporadically.

Narrative Description

Evelyn:

A light shines into a little stub that tells how bright the light is. Then, the small motor spins slower when there is less light shining on the stub. That then makes the lever attached to the black box be pressed less, which then lowers the degrees of the white stick on the blue box. So, the brighter the light is, the slower the motor spins, so the lower the angle is; the darker the light is, the faster the motor spins, so the higher the angle is.

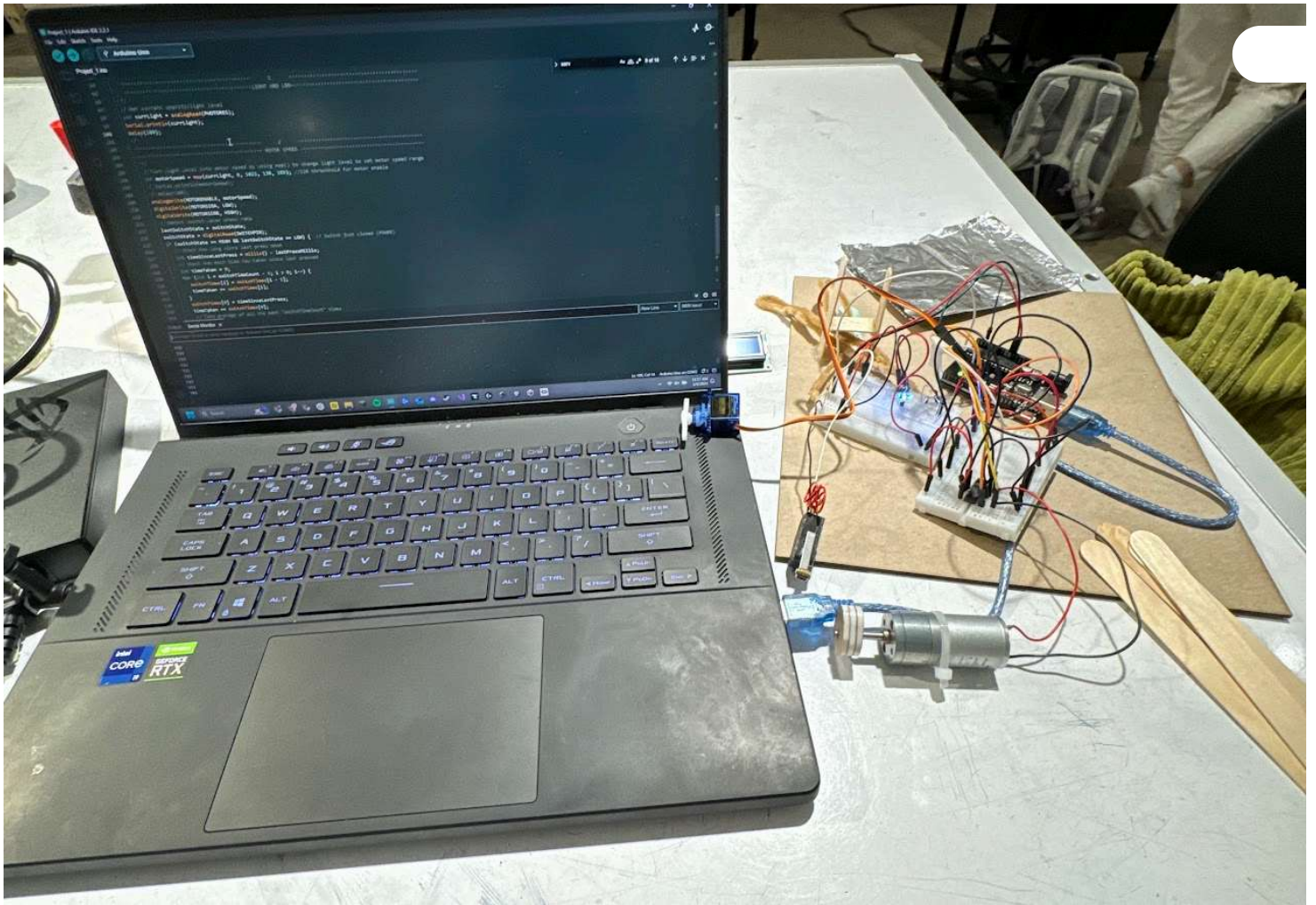


Aiden:

We use a small light and sensor (top middle) to see how much light passes through something. This helps us figure out how fast our DC motor (top right) should spin. The motor moves a small part that pushes a button. Our Arduino measures how often the button is pushed and uses that rate to tell the servo motor (bottom middle) how many degrees to rotate.

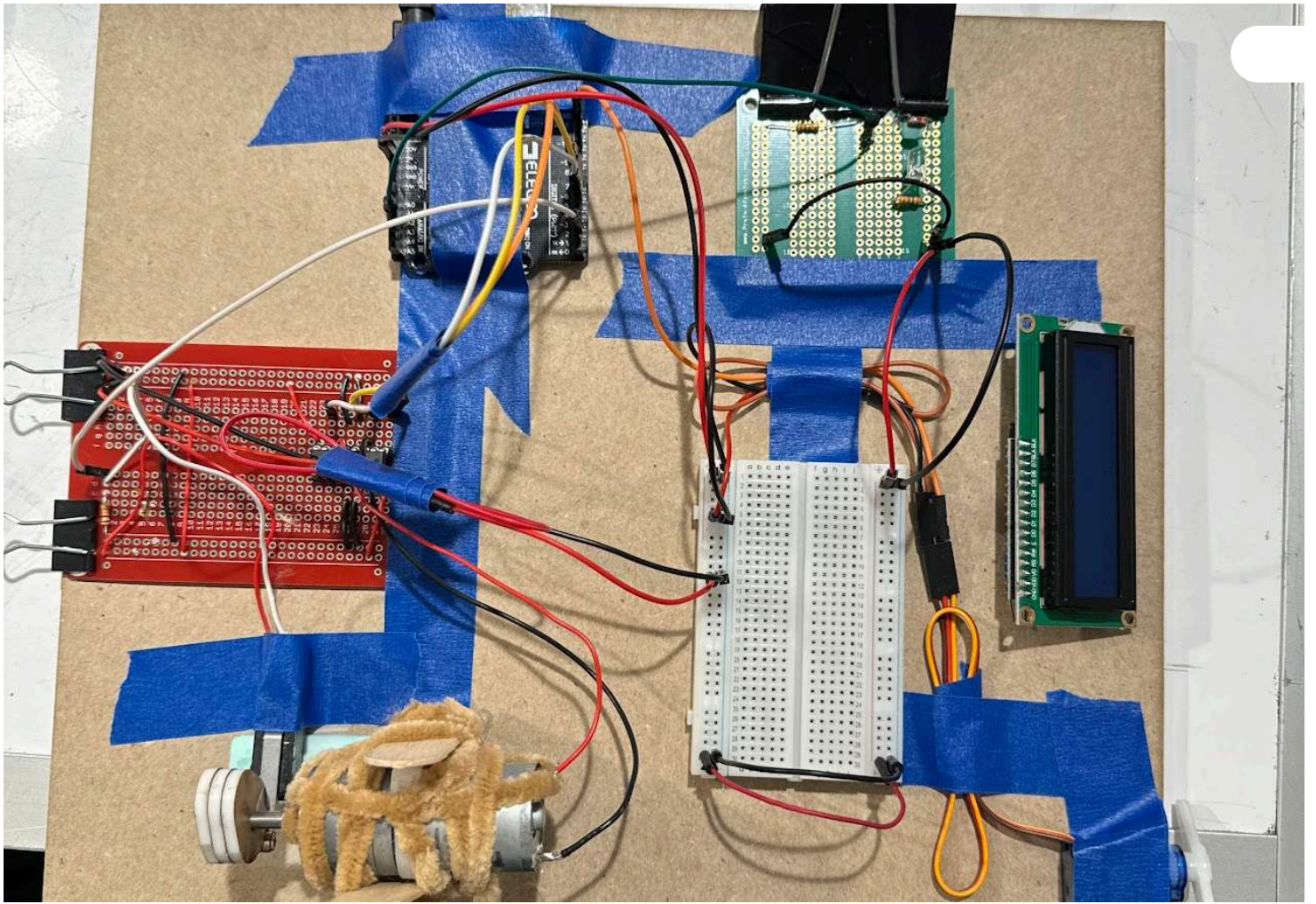
Progress Images





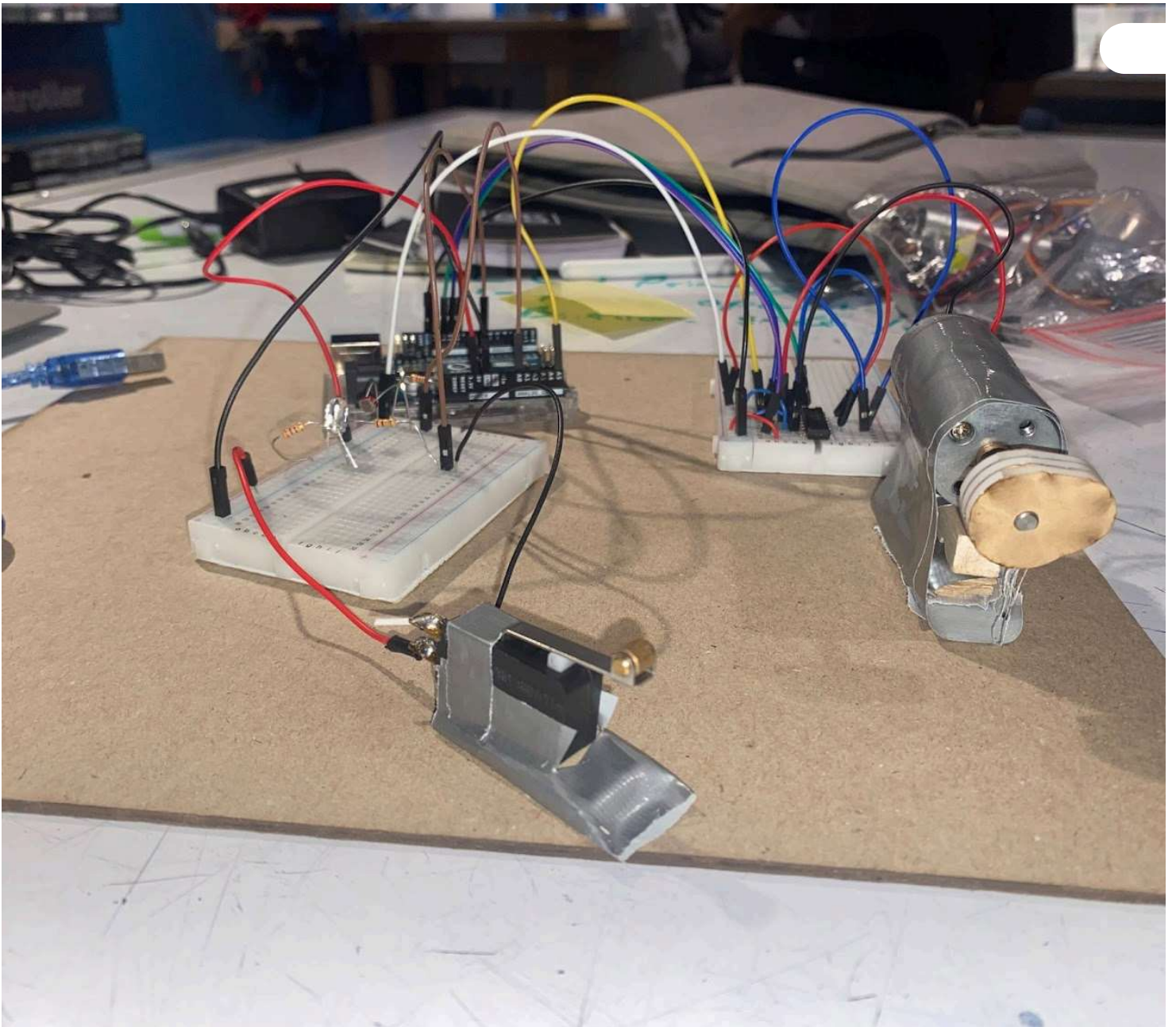
Evelyn - Troubleshooting the code. I needed to work on the code for multiple days in order to fine tune and get rid of kinks.





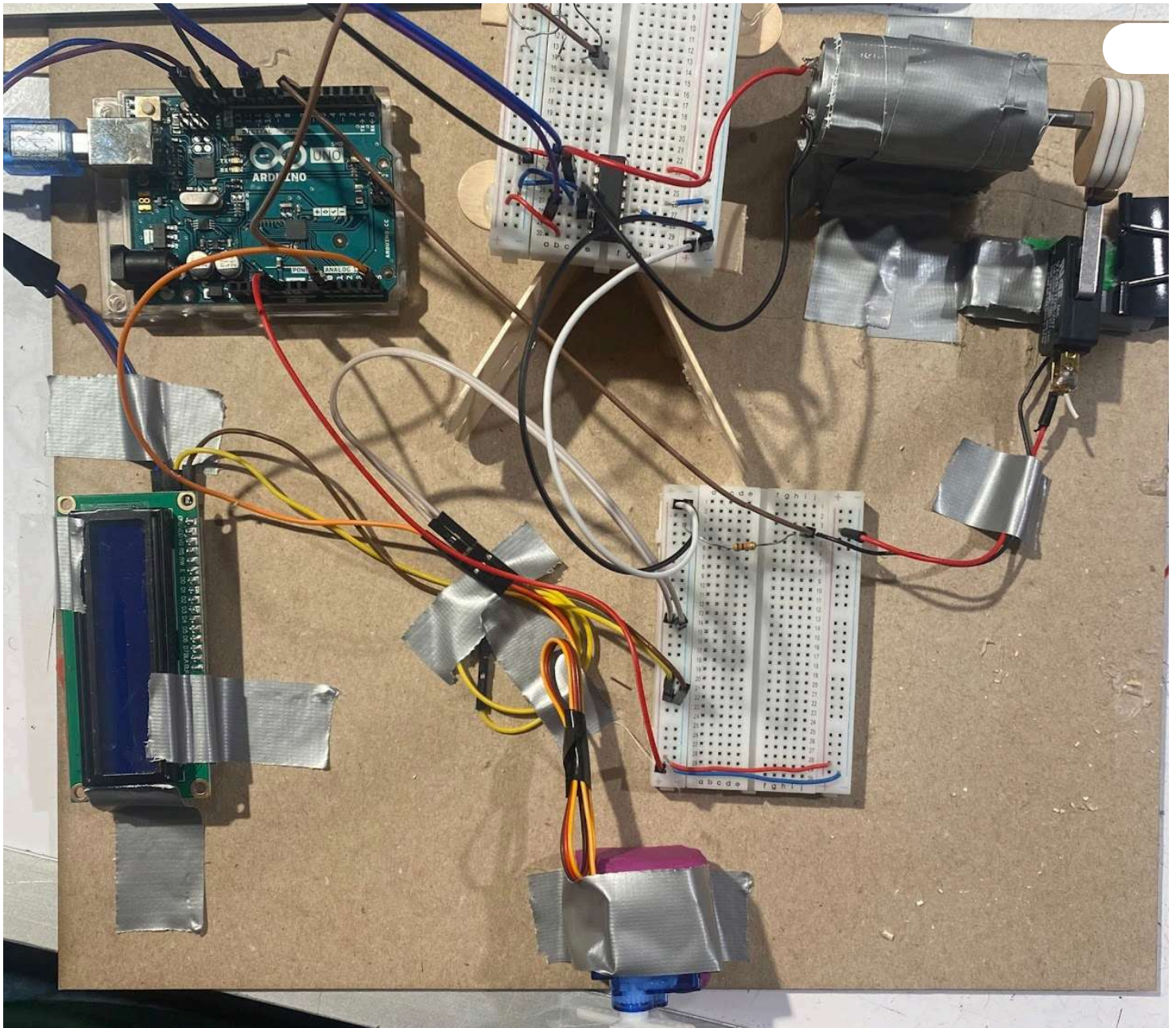
Evelyn - Placing the components onto a board layout before realizing there needed to be a format. I had to take off the tape in order to place the components in the right locations.





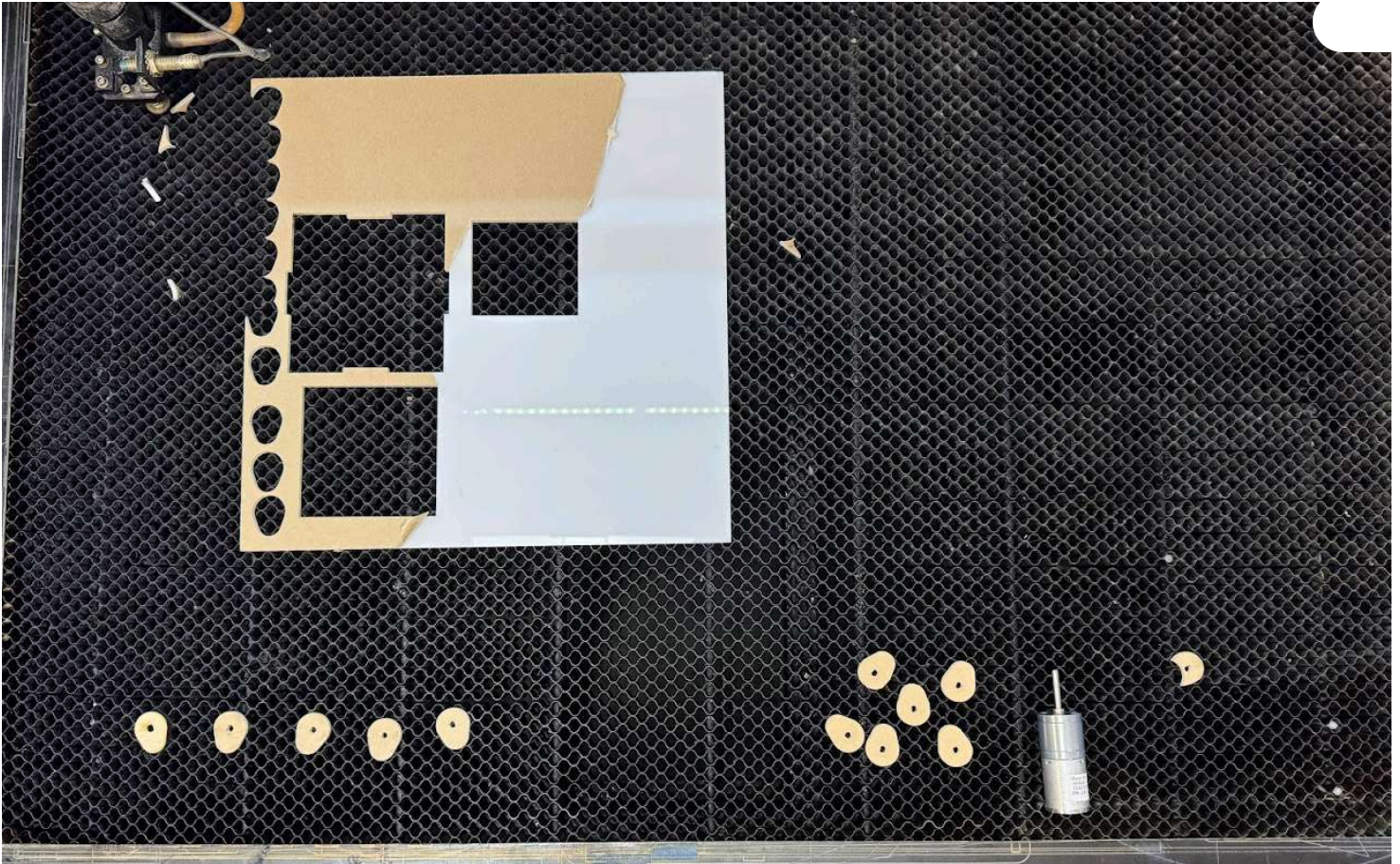
Aiden - I finally got my DC motor switch actuator functioning properly.





Aiden - After many trials of placing and replacing my components, I decided on my final layout.





Laser cutting the CAMs. The left bottom pieces did not fit the DC motor, so they needed to be reprinted in order to be able to click the lever microswitch well.

Discussion



Evelyn:

In retrospect, I found it to be quite different from my expectations, as someone new to Arduinos and circuitry. I thought that I would have to memorize the names and components more, but it wasn't like that in reality. I struggled when the components didn't function as expected, leading me to troubleshoot the wiring and the code. I learned the importance of organizing wires effectively, utilizing parallel placement on the breadboard for related components and specific colors to signify wire functions. Despite initial struggles, the task felt manageable and even enjoyable, since I like organizing and keeping things neat.

On the other hand, the code proved to be another hurdle, with comments and variables becoming jumbled. Despite this, debugging was both frustrating and satisfying, prompting me to refine and organize the code structure. What I found most rewarding was conceptualizing the middle steps in between the input and output, which allowed me to be creative with coming up with different ideas. This particular class reminds me of modded Minecraft, which has a lot of connections from physical computing. For example, Greg Tech is a famous mod that has resistors that you can use to craft certain blocks. These two experiences lining up enhanced my enjoyment of assembling the parts for this project.

Aiden:

Despite not being fully satisfied with my final project given that it was not fully functioning, I must say that the process was nevertheless rewarding. I am used to being confined in my coursework, but given the loose constraints of this project, I felt like I could freely pour my creativity into the square-foot cardboard canvas.

While I didn't find the circuitry to be too challenging, I did realize some poor practices that were made more and more clear as my project progressed. My wire management was poor. I failed to consistently color-coordinate my wires, which made my circuits harder to debug when I needed to. I also didn't cut my wires to size, leaving many of them too long and sticking out. I told myself that I would go back and fix this once I got everything working, but I never did. Next time I should properly manage my wires from the beginning to avoid this. Similarly, my poor aesthetics were clear in my final project. While I did appreciate the loose design constraints, I do recognize that aesthetics are important and it is something that I should work to refine more in the future.

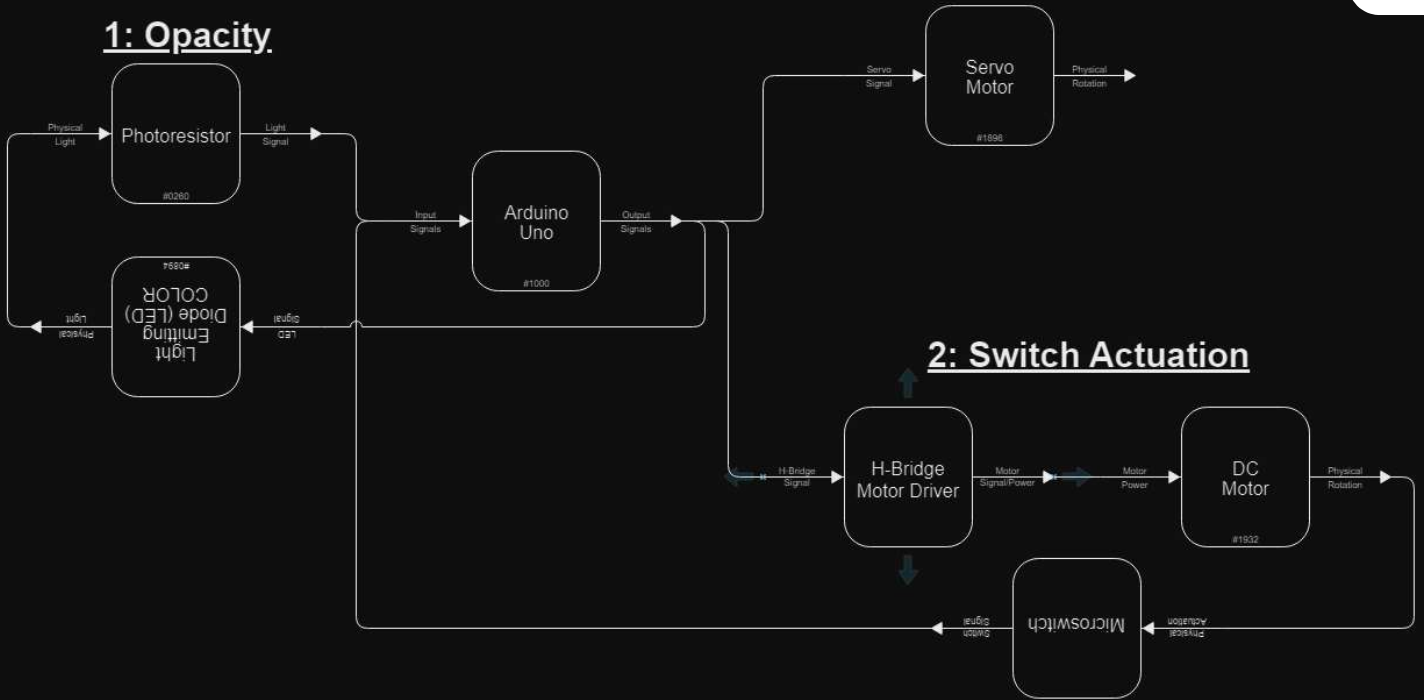
I am very grateful for this experience as it enabled me to feel comfortable as a Tinkerer. I have always wanted to get into hobby electronics or small projects like this one, but I was always held back by something. I think I needed to just try it. Now that I know that many of my imaginative projects are feasible, I can turn my ideas into reality.

Functional Block Diagram and Electrical Schematic



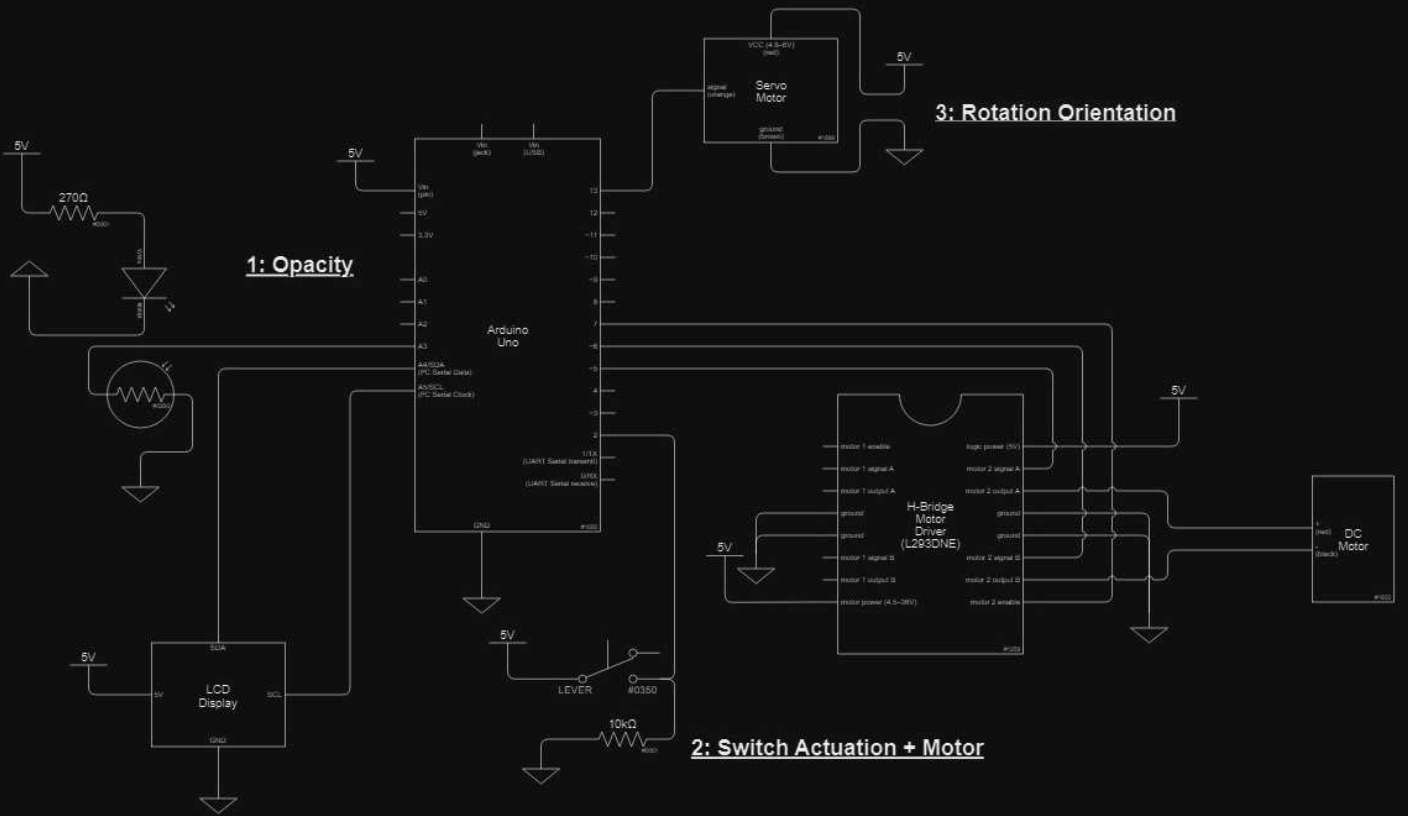
3: Rotation Orientation

1: Opacity



2: Switch Actuation

3: Rotation Orientation



1: Opacity

2: Switch Actuation + Motor

Code




```

/*
  Title: Double Transducer: Opacity to Rotation
  By: Evelyn Bang and Aiden Magee
  Description: The code takes in a value from the photoresistor that shows how much
  light is being shined into it. It then translates it into the DC motor speed. The lever
  microswitch detects the rotation by looking at the time difference in between each
  press, then adds it to an array of the last 5 recorded times it took for each press.
  These values are then averaged and then translated into the servo motor's angle which
  is in between 10 and 170 for the purpose of the servo motor's physical limitations.

```

Pin mapping:

pin	mode	description
2	INPUT	Lever Microswitch
5	OUTPUT	Motor signal A (LOW)
6	OUTPUT	Motor signal B (HIGH)
7	OUTPUT	Motor enable
13	OUTPUT	Servo motor
A3	INPUT	Photoresistor
5V	OUTPUT	LED (just on)

2	INPUT	Lever Microswitch
---	-------	-------------------

5	OUTPUT	Motor signal A (LOW)
---	--------	----------------------

6	OUTPUT	Motor signal B (HIGH)
---	--------	-----------------------

7	OUTPUT	Motor enable
---	--------	--------------

13	OUTPUT	Servo motor
----	--------	-------------

A3	INPUT	Photoresistor
----	-------	---------------

5V	OUTPUT	LED (just on)
----	--------	---------------

----- I²C LCD PINS -----

SDA		SDA (next to reset button to the right of 13)
-----	--	---

SCL		SCL (right next to SDA)
-----	--	-------------------------

NOTES:

- LED is constantly on
- Servo motor limit to 10-170 degrees due to hardware limitations
- RPM of DC motor range (870, 1430) ms between every rotation
- LCD on 1 frame for 200 ms (5 fps)

```

*/
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

```

```

// GLOBAL VARIABLES

```

```

// SERVO MOTOR

```

```

Servo serv;

```

```

const int SERVOPIN = 13;

```

```

// DC MOTOR

```

```

const int MOTORSIGA = 5;

```

```

const int MOTORSIGB = 6;

```

```

const int MOTORENABLE = 7;

```

```

// PHOTORESISTOR + LED

```

```

const int PHOTORES = A3;

```

```

// LEVER MICROSWITCH

```

```

const int SWITCHPIN = 2;

```

```

int switchState = LOW;

```

```

int TestSwitchState = LOW;

```



```

int lastSwitchState = LOW;
const int switchTimeCount = 5;
int switchTimes[switchTimeCount];
int lastPressMillis = 0;
int lowestTime = 870;
int highestTime = 1500;
int timeTaken;
// LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);
unsigned long frameTime = 100; // milliseconds in between each frame
unsigned long nextToRun = 0; // next time to turn the lcd

```

```

void setup() {
  // Attach pins
  serv.attach(SERVOPIN);
  pinMode(MOTORSIGA, OUTPUT);
  pinMode(MOTORSIGB, OUTPUT);
  pinMode(MOTORENABLE, OUTPUT);
  pinMode(PHOTOIRES, INPUT);
  pinMode(SWITCHPIN, INPUT);
  // Starts + prints on LCD
  lcd.init();
  lcd.backlight();
  // Timer for LCD
  nextToRun = millis() + frameTime;
  Serial.begin(9600);
}

```

```

void loop() {
  /*
  ----- 1 -----
  -----
  -----LIGHT AND LED-----
  -----
  */
  // Get current opacity/light level
  int currLight = analogRead(PHOTOIRES);
  /*
  ----- 2 -----
  -----
  ----- MOTOR SPEED -----
  -----
  */
  // Turn light level into motor speed
  int motorSpeed = int((currLight * 1023.0f) * 125.0f + 130.0f);
  digitalWrite(MOTORENABLE, HIGH);
  digitalWrite(MOTORSTGA, LOW);
}

```



```

digitalWrite(MOTORSION, LOW);
analogWrite(MOTORSIGB, motorSpeed);
// Detect switch lever press rate
lastSwitchState = switchState;
switchState = digitalRead(SWITCHPIN);
// Check how long since last press down
int timeSinceLastPress = millis() - lastPressMillis;
// Checks to see if switch was just pressed
if (switchState == HIGH && lastSwitchState == LOW && timeSinceLastPress >= 400) {
  // Check how much time has taken since last pressed
  timeTaken = 0;
  for (int i = switchTimeCount - 1; i > 0; i--) {
    switchTimes[i] = switchTimes[i - 1];
    timeTaken += switchTimes[i];
  }
  switchTimes[0] = timeSinceLastPress;
  timeTaken += switchTimes[0];
  // Take average of all the past "switchTimeCount" times
  timeTaken /= switchTimeCount;
  // Serial.println(timeTaken);
  lastPressMillis = millis(); // Update pressed time
}
/*
----- 3 -----
----- SERVO ANGLE -----
*/
// If fast => higher angle of servo (max set to 170) + if slow => lower angle of
servo (min set to 10)
float servPer = float(timeTaken - lowestTime) / float(highestTime - lowestTime);
int servMotVal = int((servPer * 160.0f) + 10.0f);
serv.write(servMotVal);
// Printing out to the LCD screen
if (millis() >= nextToRun) {
  // LCD setup
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("i:");
  lcd.setCursor(6, 0);
  lcd.print("m:");
  lcd.setCursor(12, 1);
  lcd.print("o:");
  // Input
  lcd.setCursor(2, 0);
  int inputMap = int((float(currLight) / 1023.0f) * 99.0f);
  lcd.print(inputMap);
  // Middle input
  lcd.setCursor(8, 0):

```



```
    lcd.setCursor(0, 0);  
    int midInput = int((float(motorSpeed - 130) / 125.0f) * 99.0f);  
    lcd.print(midInput);  
    // Middle output  
    lcd.setCursor(8, 1);  
    int midOutput = int(servPer * 99.0f);  
    lcd.print(midOutput);  
    // Output  
    lcd.setCursor(14, 1);  
    int output = int((float(serv.read() - 10) / 160.0f) * 99.0f);  
    lcd.print(output);  
    // Resetting timer for when to run LCD screen next  
    nextToRun = millis() + frameTime;  
}  
}
```







Intro to Physical Computing is sponsored by [IDeATe](#),
part of [Carnegie Mellon University](#) in Pittsburgh, Pennsylvania
[CMU legal notice](#)

