Activity No. 11	
Motor Control Driver	
Course Code:CPE006	**Program: CPE**
Course Title: Microprocessor Systems	**Date Performed: 11/6/2024**
Section: CPE41S1	**Date Submitted: 11/13/2024**
Name/s: Cuevas, Christian Jay L. Cruz, Daniel Y. Dela Cruz, Alexander Jake Lucero, Carl Walter	**Instructor:** Engr. NJ Lusung

1. Objective:

The activity aims to demonstrate the students' capability to code a program, build a circuit, and test the functionality of a digital system coupled to a motor control driver.

2. Intended Learning Outcomes (ILOs):

After completion of this activity the students should be able to:

1. Develop a circuit that will apply DC motors.
2. Conduct experimentation procedures to test the functions.

3. Discussion

A motor controller is a device or group of devices that serves to govern in some predetermined manner the performance of an electric motor. A motor controller might include a manual or automatic means for starting and stopping the motor, selecting forward or reverse rotation, selecting and regulating the speed, regulating or limiting the torque, and protecting against overloads and faults.

Motor controllers can be manually, remotely or automatically operated. They may include only the means for starting and stopping the motor or they may include other functions.An electric motor controller can be classified by the type of motor it is to drive such as permanent magnet, servo, series, separately excited, and alternating current.A motor controller is connected to a power source such as a battery pack or power supply, and control circuitry in the form of analog or digital input signals.

A small motor can be started by simply plugging it into an electrical receptacle or by using a switch or circuit breaker. A larger motor requires a specialized switching unit called a motor starter or motor contactor. When energized, a direct on line (DOL) starter immediately connects the motor terminals directly to the power supply.

A reduced-voltage, star-delta or soft starter connects the motor to the power supply through a voltage reduction device and increases the applied voltage gradually or in steps. In smaller sizes a motor starter is a manually operated switch; larger motors, or

those requiring remote or automatic control, use magnetic contactors. Very large motors running on medium voltage power supplies (thousands of volts) may use power circuit breakers as switching elements.

4. Resources:

The activity will require the following software, tools and equipment:

4.1 Desktop Computer

4.2 Dev C/C++/Processing

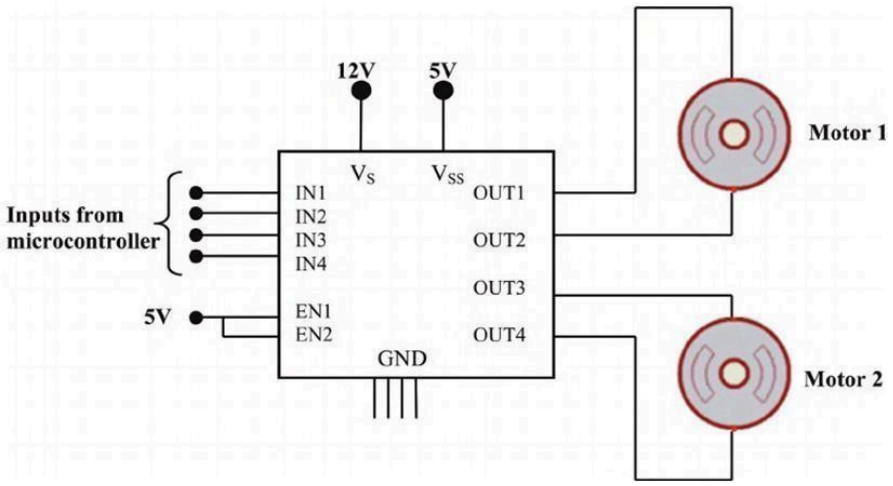
4.3 Sketch/Flowcode

4.4 Multisim or _____

4.5 Other tools: _____

5. Procedures:

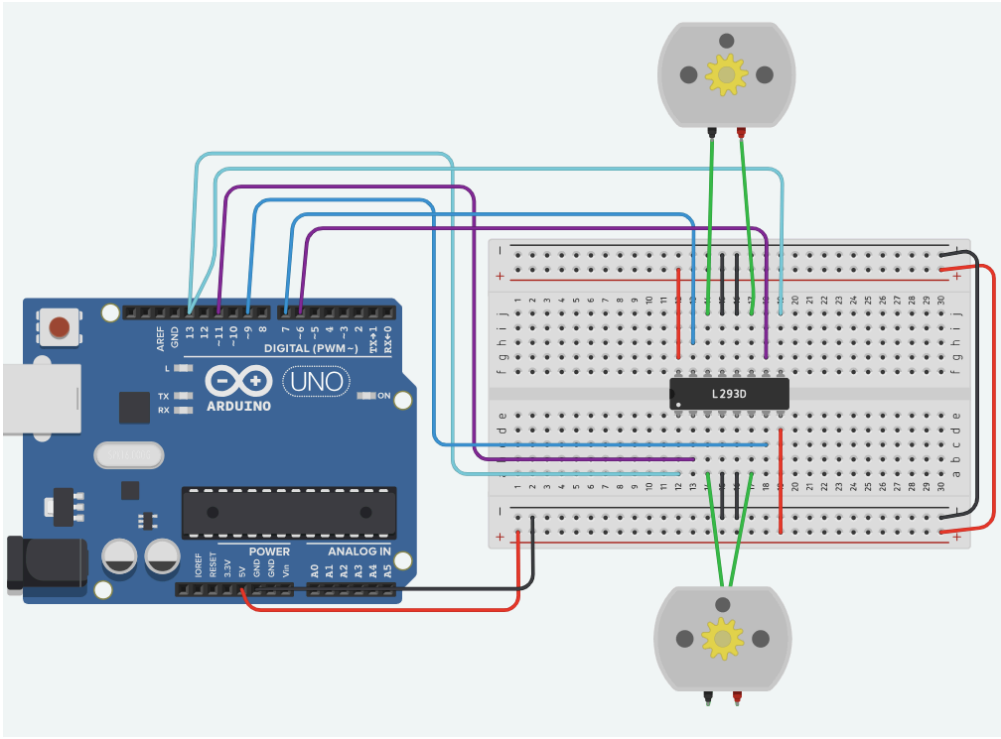
1. Configure the motor driver and the motors based on the following figure:

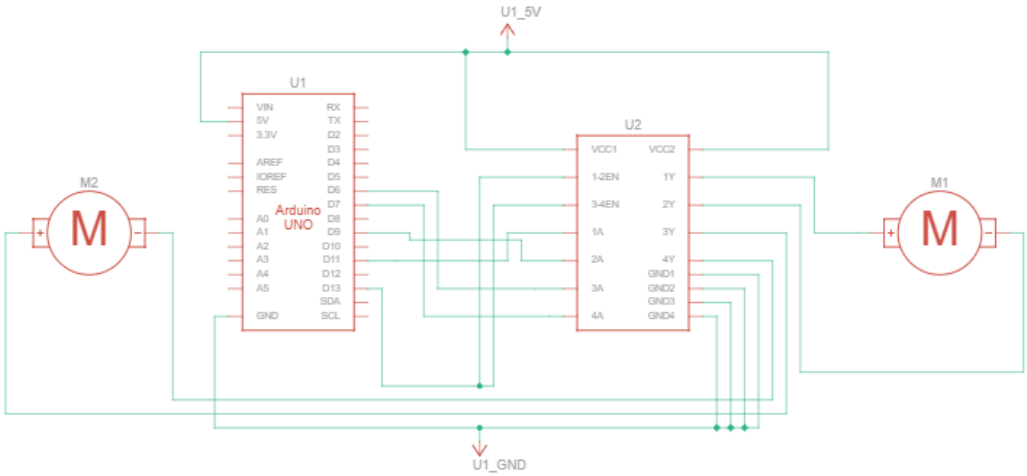


2. Connect the input pins and the enable pins to the microcontroller. Draw the complete schematic diagram on the space provided at the following section.
3. Write the source code to test the rotation of the motors in a single direction. Write the source code in the space at the following section.
4. Modify the wiring and code to change the function of the circuit. The speed of the motors should now be controlled by a potentiometer. Write the diagram and code on the space provided.

6. Results

Schematic Diagram (Rotation Test):



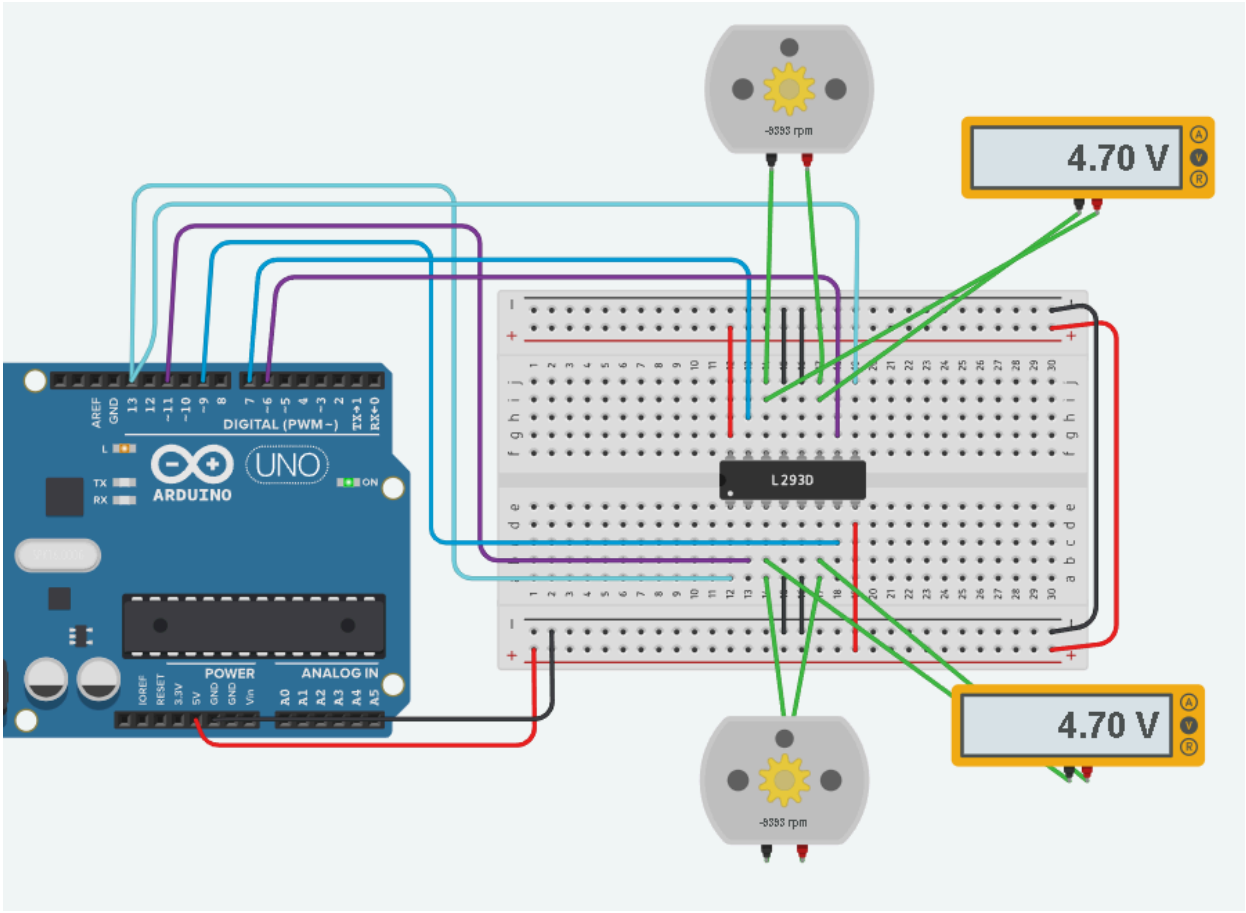


Source Code(Rotation Test):

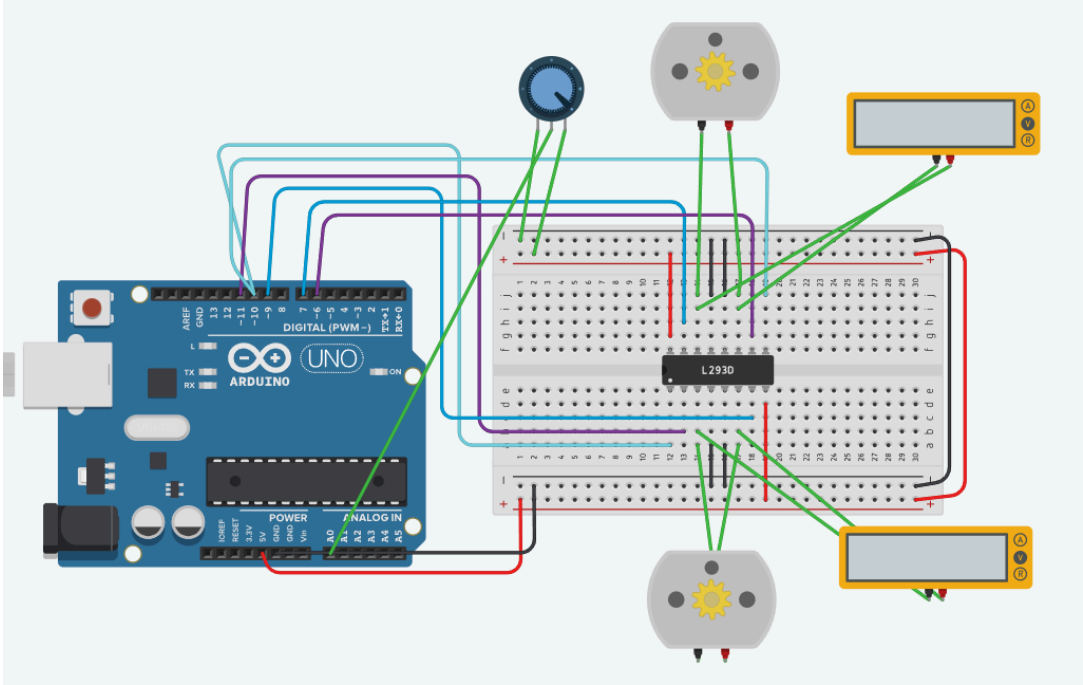
```
const int acti=13;
const int in1=11;
const int in2=9;
const int in3=6;
const int in4=7;

void setup()
{
  pinMode(acti, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  digitalWrite(acti, HIGH);
}

void loop ()
{
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  delay(2000);

  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  delay(2000);

  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(2000);

  digitalWrite(in1, HIGH);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, HIGH);
  delay(2000);
```

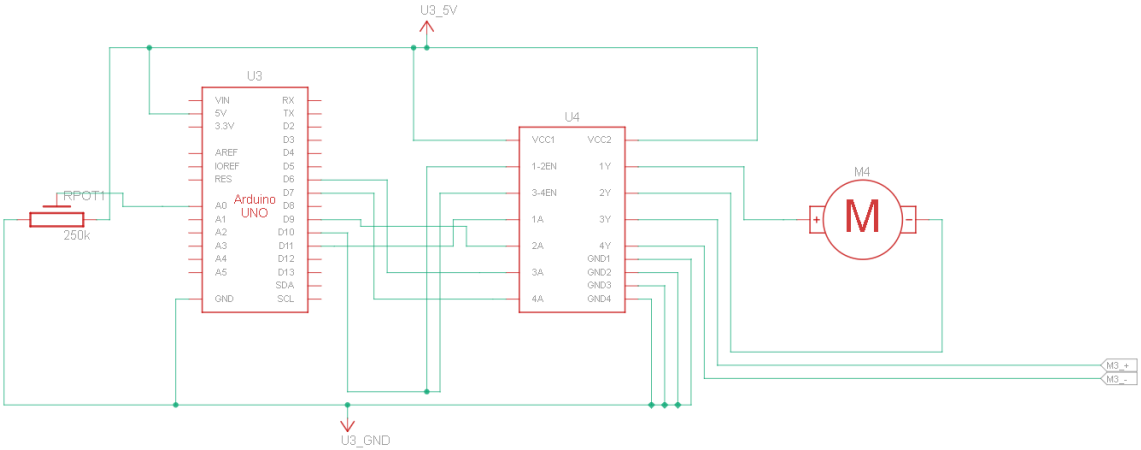
}

Output (Rotation Test):



Schematic Diagram (Potentiometer Control):





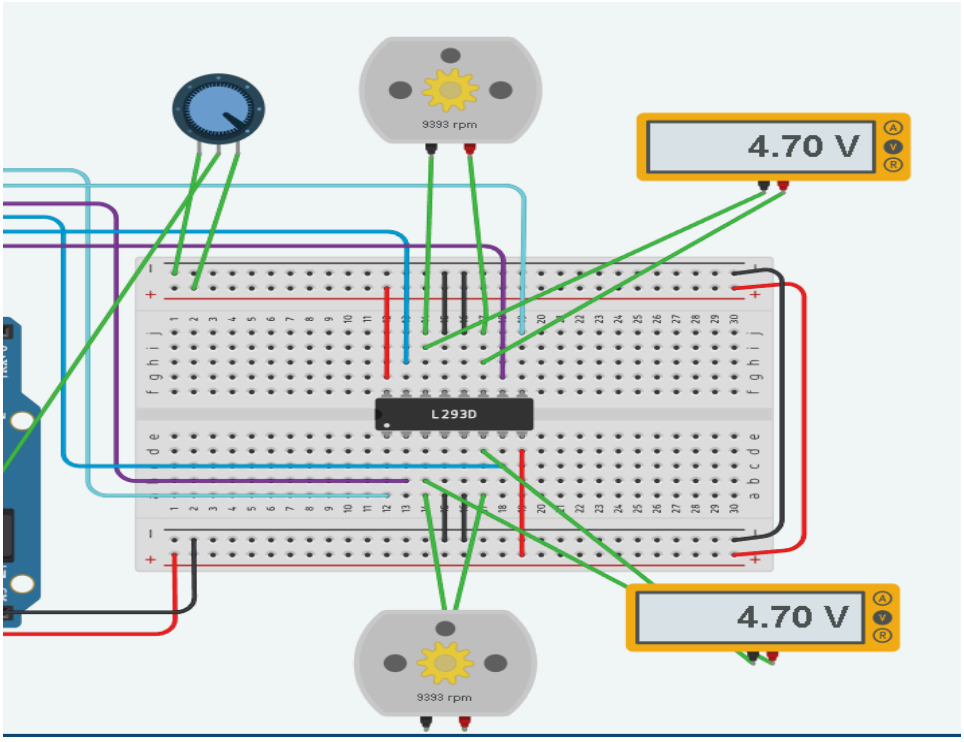
Source Code (Potentiometer Control):

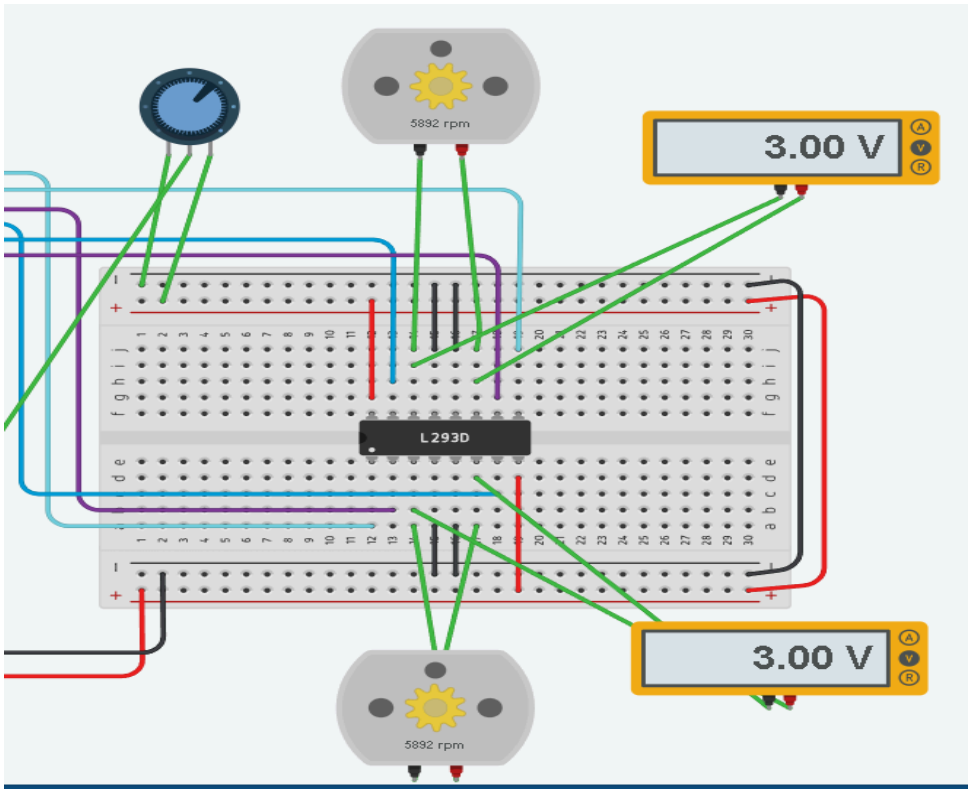
```
// Pin Definitions
const int acti = 10;   // Enable pin for motor driver (PWM control, changed to pin 9)
const int in1 = 11;   // Input 1 for Motor A
const int in2 = 8;    // Input 2 for Motor A
const int in3 = 6;    // Input 1 for Motor B
const int in4 = 7;    // Input 2 for Motor B

const int potPin = A0; // Pin connected to the potentiometer

void setup() {
  // Initialize all pins as outputs
```

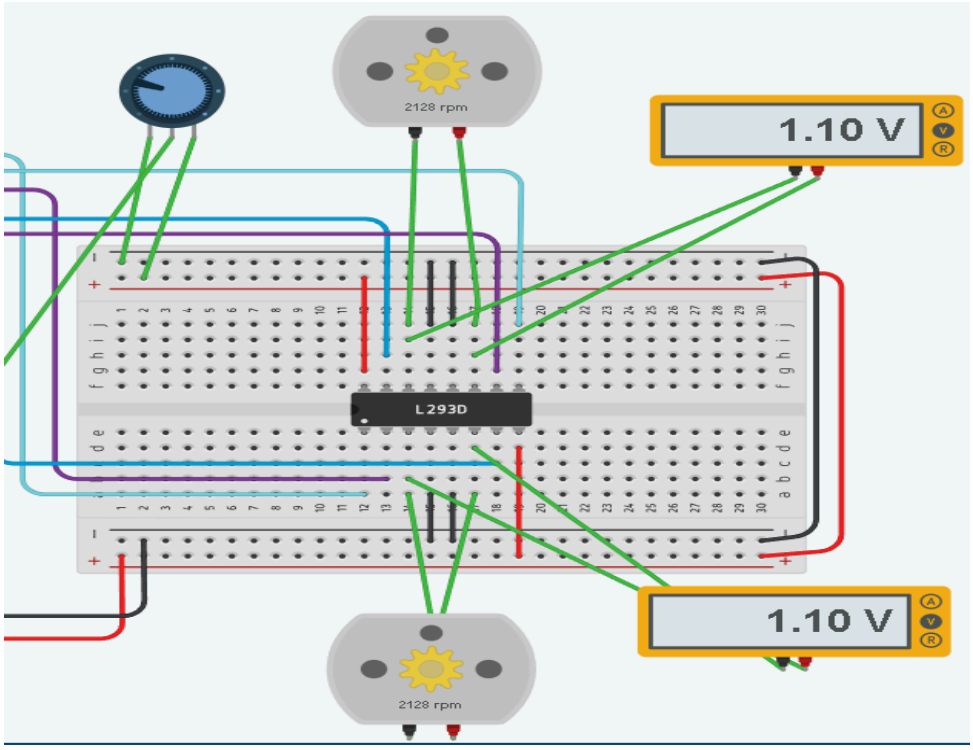
```
  pinMode(acti, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  Serial.begin(9600); // Initialize Serial Monitor for debugging
}

// Function to read the potentiometer and set motor speed
int getMotorSpeed() {
  int potValue = analogRead(potPin);       // Read the potentiometer value (0-1023)
  int motorSpeed = map(potValue, 0, 1023, 0, 255); // Map it to PWM range (0-255)
  return motorSpeed;
}

// Function to move motors forward at a specified speed
void moveForward(int speed) {
  analogWrite(acti, speed); // Set motor speed using PWM
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

// Function to move motors backward at a specified speed
void moveBackward(int speed) {
  analogWrite(acti, speed); // Set motor speed using PWM
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}

// Function to stop motors
void stopMotors() {
  analogWrite(acti, 0); // Set speed to 0
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

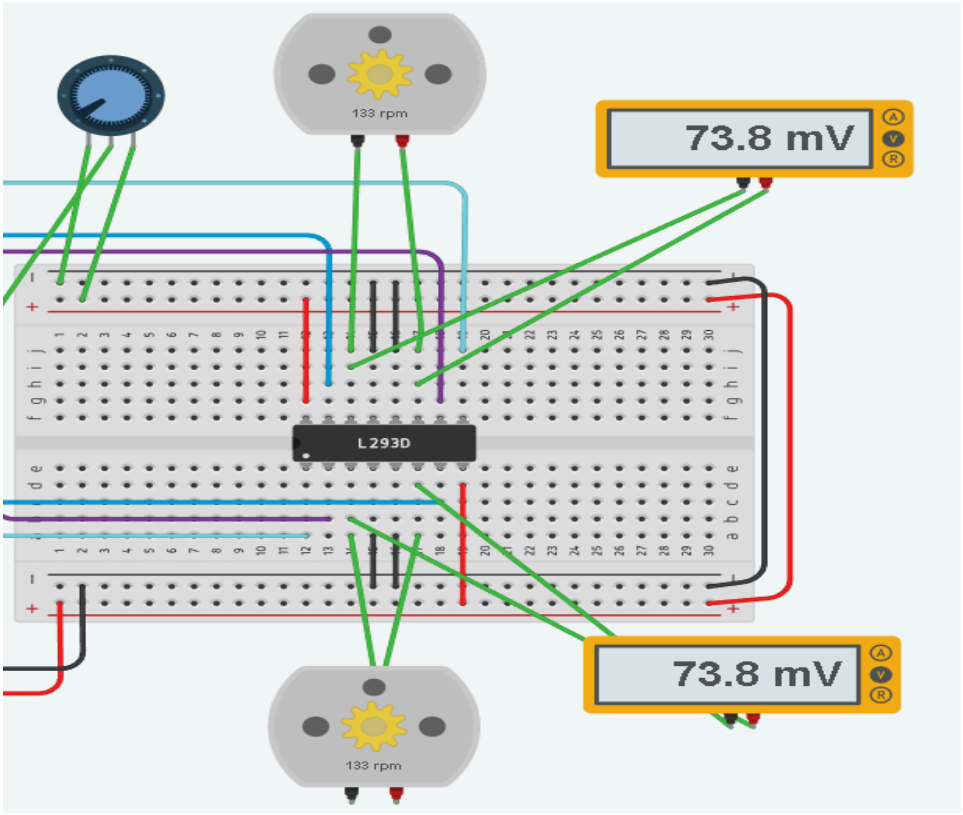
```
void loop() {
  int speed = getMotorSpeed(); // Get the current motor speed from potentiometer

  Serial.print("Potentiometer Value: ");
  Serial.print(speed);
  Serial.print(" | Motor Speed: ");
  Serial.println(speed);

  // Control the motors based on the potentiometer value
  if (speed > 0) {
    // Move forward if speed is above 0
    moveForward(speed);
  } else {
    // Stop the motors if the speed is 0
    stopMotors();
  }

  delay(100); // Small delay for stability
}
```

Output:




5892 rpm
3.00 V
L293D
3.00 V
5892 rpm

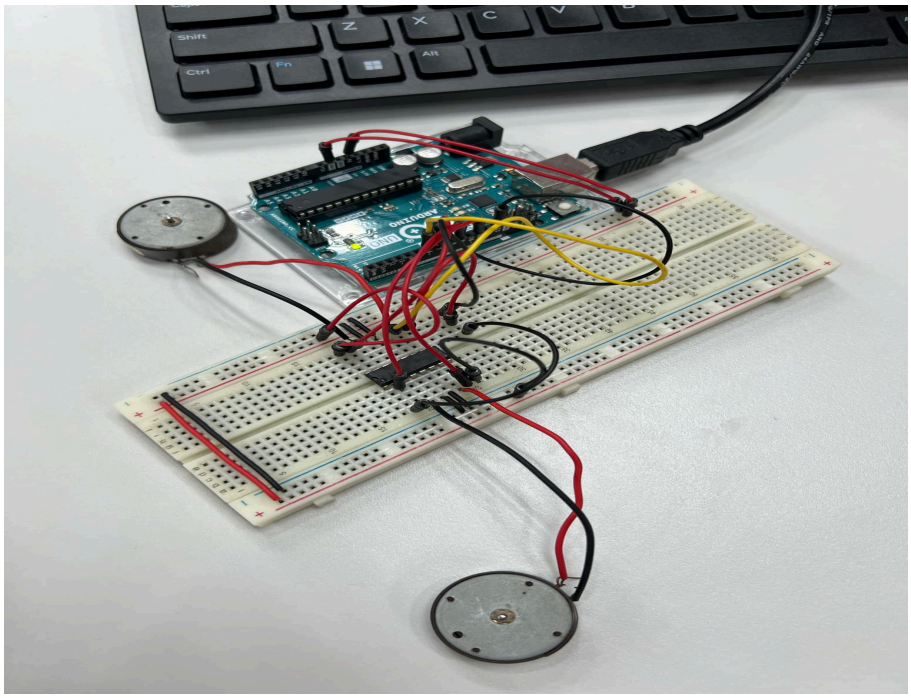

2128 rpm
1.10 V
L293D
1.10 V
2128 rpm



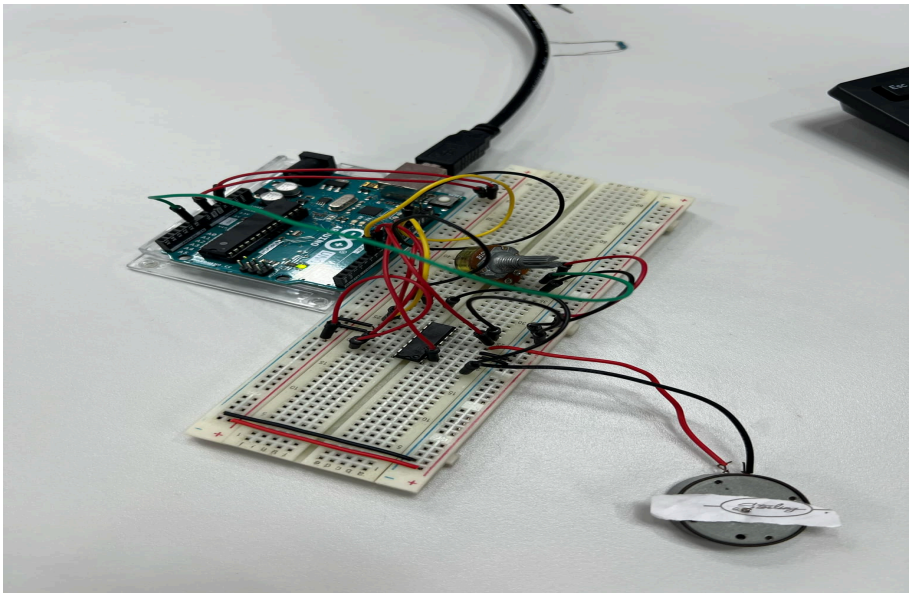
TinkerCAD Link:
[Circuit design Group 3 - Tinkercad - Tinkercad]

Actual Circuit for Procedure:

Speed Test



Potentiometer



Video Link:

https://drive.google.com/file/d/1IxaPQkYX66SukgWgSZFEKxhE4ucWF1JC/view?usp=sharing

7. Observations

In this activity, we found that maintaining secure and stable connections was essential, as any loose or poorly connected wires led to unpredictable motor performance. We also noted that the motors responded effectively to changes in speed; however, there was a noticeable delay when reducing the speed, which we suspect may be attributed to the inherent characteristics of the motor, such as inertia or friction.

The team made several modifications to the code, which resulted in a more precise and responsive motor control system. Additionally, it became evident that operating two motors simultaneously required a significant amount of power, underscoring the importance of closely monitoring the power supply during testing to prevent issues such as voltage drops or overheating.

These observations highlight key areas for potential optimization and will inform our efforts to enhance both the circuit's reliability and overall stability in future iterations. This initial testing phase provided critical insights that will guide the improvement of the system's design and functionality.

8. Conclusions

In conclusion, this activity provided us with the opportunity to design, construct, and evaluate a digital motor control system using a microcontroller and an L293D motor driver, both through Tinkercad simulation and physical prototyping. We began by implementing the initial motor control circuit diagram as outlined in the procedure, followed by developing a source code to test motor rotations. The code was subsequently modified to include speed control functionality.

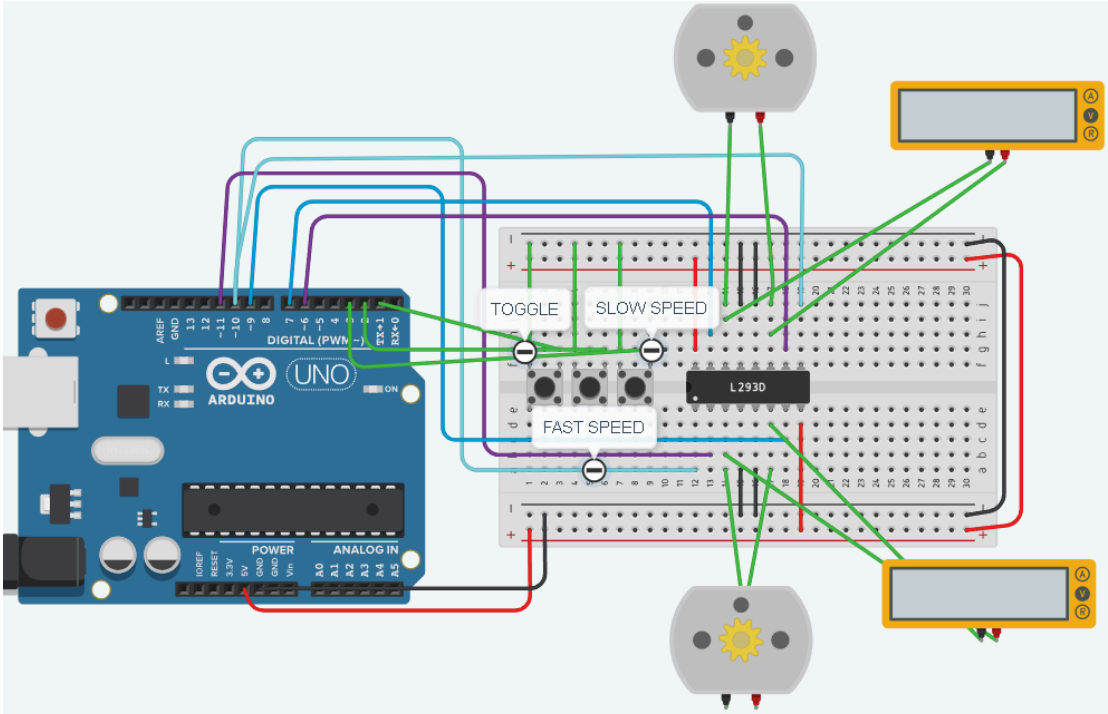
We successfully integrated additional components, such as a potentiometer and switches, allowing us to dynamically adjust the motor speed between low and high settings and to selectively activate specific motors. Through this process, we gained practical experience in assembling motor control circuits and refining control algorithms, which enhanced our understanding of motor drivers and their applications.

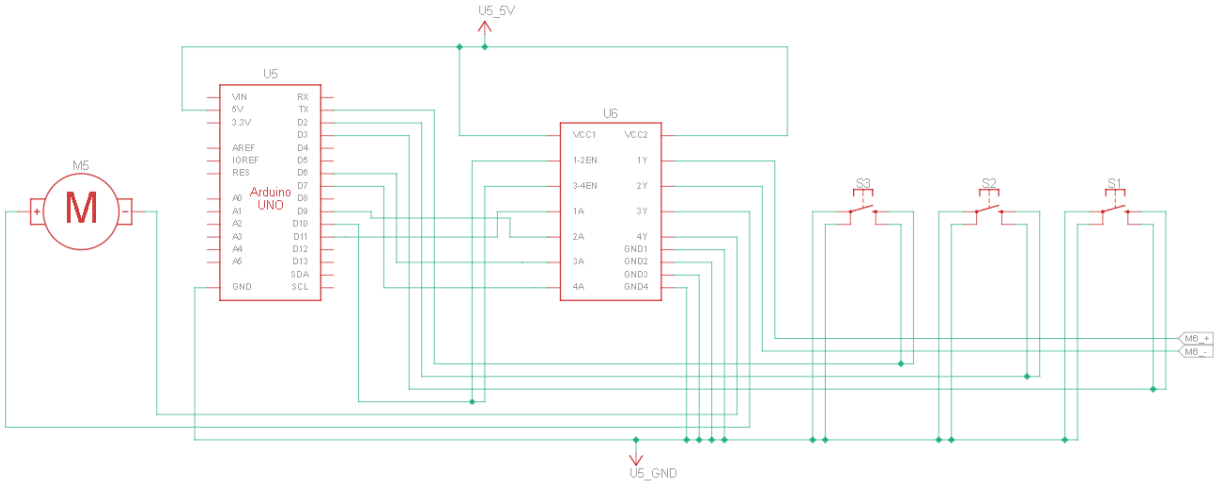
Overall, this hands-on activity equipped us with valuable skills in digital motor control, circuit design, and troubleshooting, reinforcing core concepts in microcontroller programming and hardware interfacing.

9. Supplementary Activity

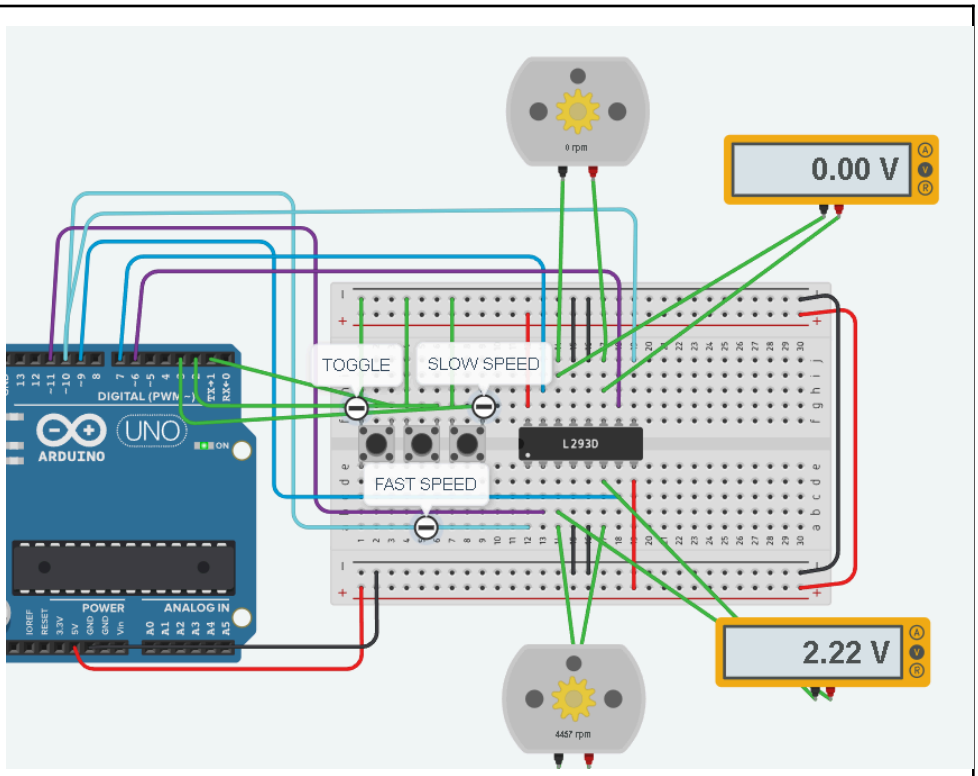
Modify the source code such that the two motors connected to the motor driver have controllable speed. Use a switch to toggle between HIGH SPEED and LOW SPEED rating for the motors. Use a third switch toggle to choose which motors are active.

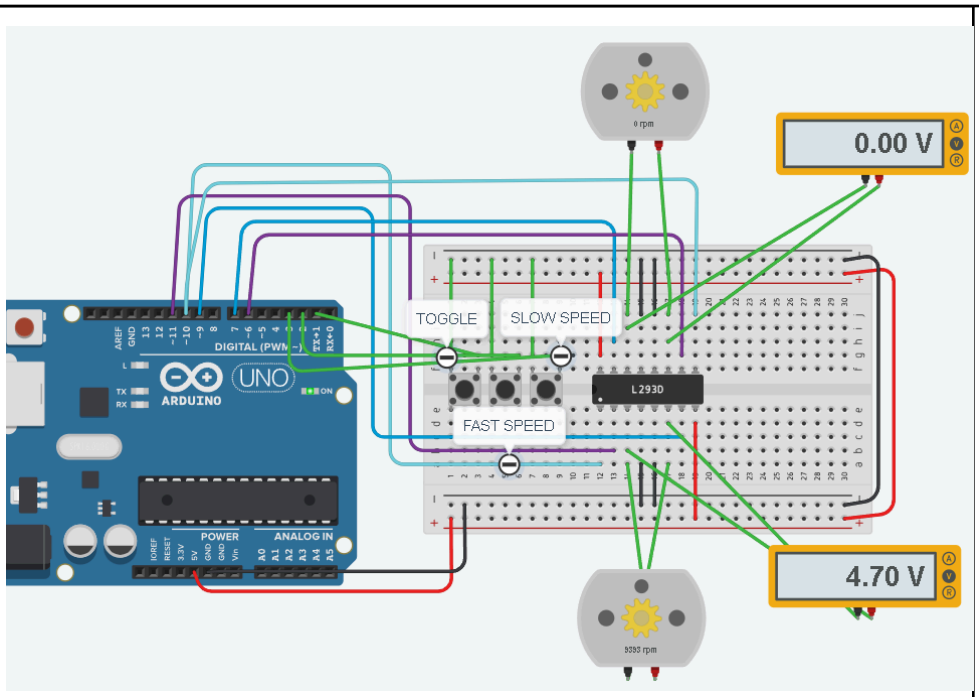
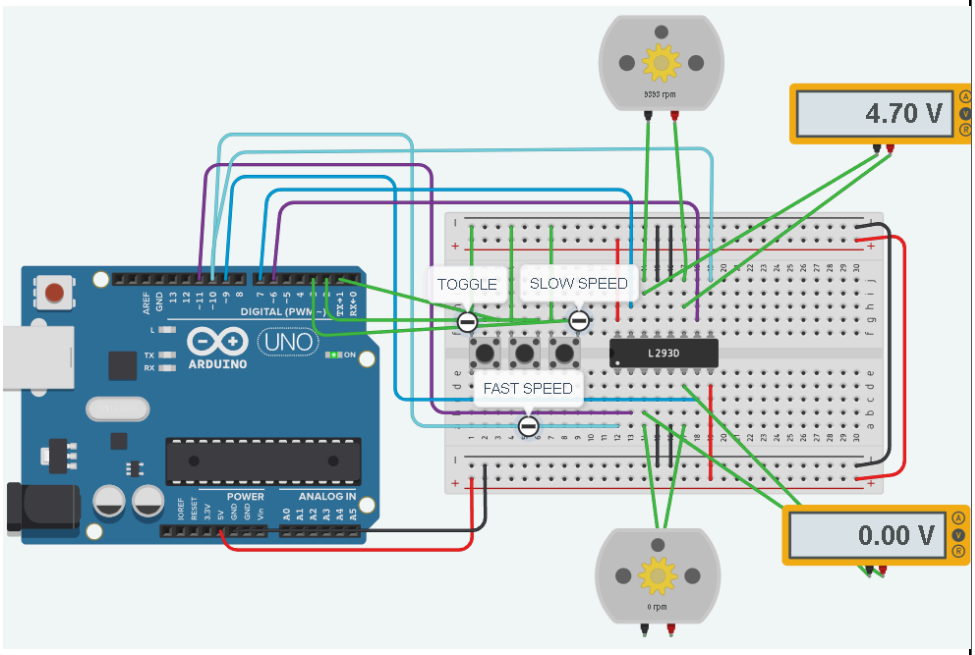
Schematic Diagram:

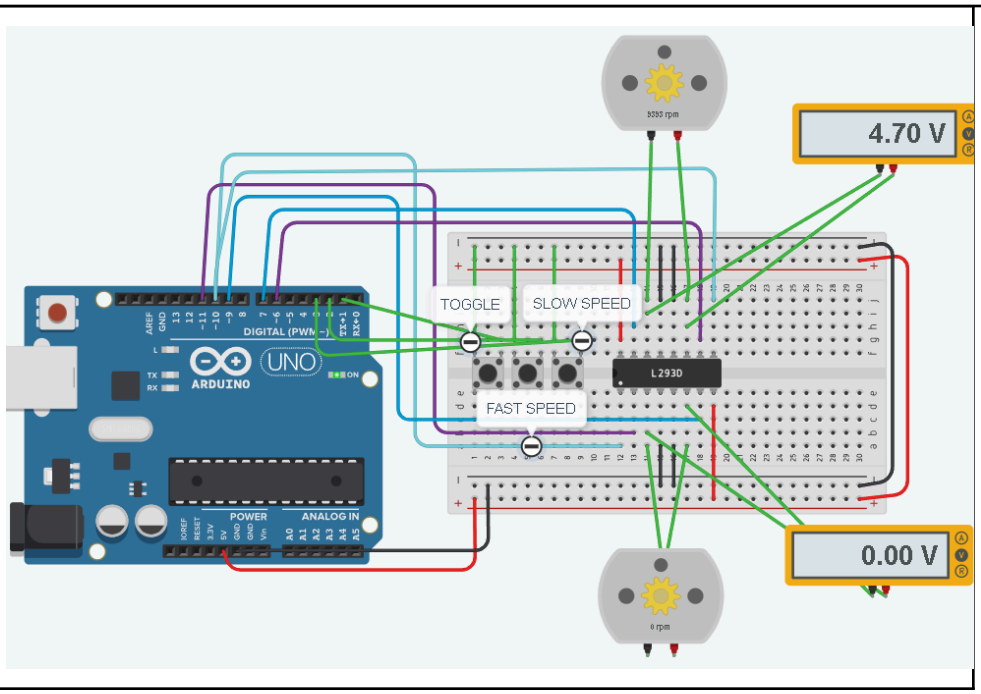




Output:

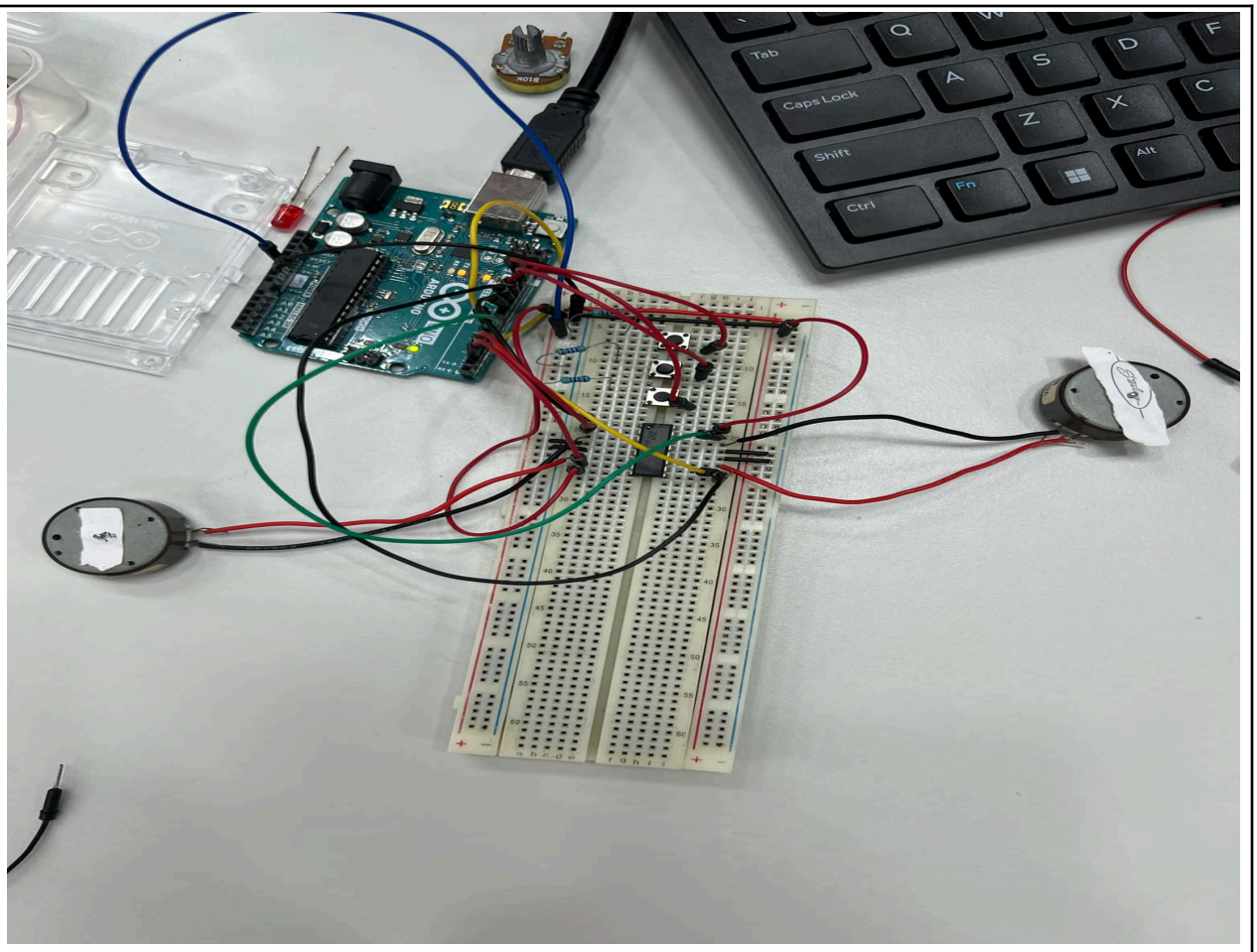
Low Speed	

Fast Speed	
Left Motor	

Right Motor	

TinkerCAD: [Circuit design Group 3 - Tinkercad - Tinkercad]

Actual Circuit:



Video Link:
https://drive.google.com/file/d/1hFtOzBhvBw47iAVtJtGFQBBauUbcKnRE/view?usp=drive_link

10. Assessment (Rubric for Laboratory Performance):