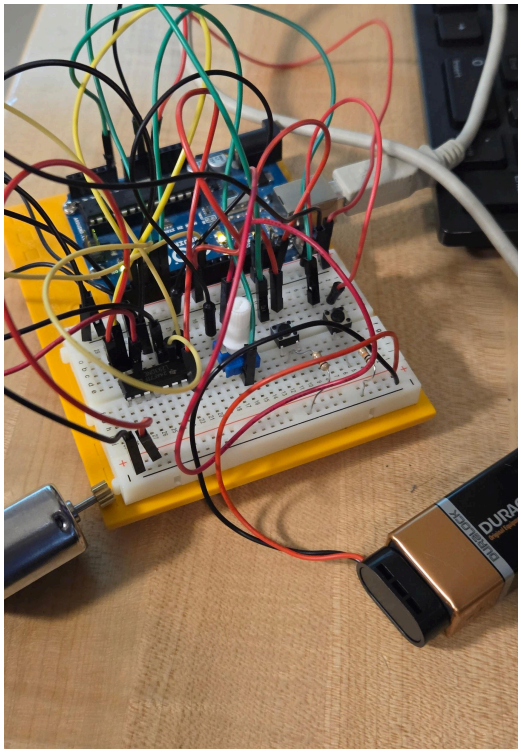
Project #10: The Zoetrope

This project aims to create the illusion of motion through a spinning cylinder that tricks your eyes into seeing an image move. Clicking one button will turn the motor on/off, while the other button toggles the direction the motor will spin. The potentiometer will control the speed of the motor.

Required Items Motor, 9V Battery, H-Bridge, Potentiometer, 2 Push Buttons, 2 Resistors (10 Kohm)

Breadboard Diagram



As seen in the diagram, the 9V battery is activating the right rail, with 2 push buttons being grounded to the 9v Battery. The grounds of both rails are connected, and the left rail is connected directly to the 5V from the Arduino. The potentiometer is connected to the left rail, and then read through the ANALOG IN pins located on the Arduino. The motor is connected to the H-Bridge, which receives power from the 9V battery, and is then connected to the digital pins on the Arduino.

Arduino IDE Code

```
//H-Bridge Control Pins
const int HBridge1Pin = 2;
const int HBridge2Pin = 3;
const int HBridgeTogglePin = 9;

//Pins for the two buttons
const int directionButtonPin = 4;
const int powerTogglePin = 5;
const int potPin = A0;                    // potentiometer pin to control
speed

// booleans that keep track of direction and on/off state
bool toggleButtonValue = false;
bool previousToggleButtonValue = false;
bool directionButtonValue = false;
bool previousDirectionButtonValue = false;

//booleans that actually control the motor
bool motorEnabled = false;
int motorSpeed = 0;
bool motorDirection = true;

void setup() {
//set the two buttons to inputs, and the H Bridge as an output
  pinMode(directionButtonPin, INPUT);
  pinMode(powerTogglePin, INPUT);
  pinMode(HBridge1Pin, OUTPUT);
  pinMode(HBridge2Pin, OUTPUT);
  pinMode(HBridgeTogglePin, OUTPUT);

  // have the motor be off by default
  digitalWrite(HBridgeTogglePin, LOW);
  Serial.begin(9600); //for debugging in case we need to log to serial
window
}

void loop() {
   // read the value of the on/off button
```

```
  toggleButtonValue = digitalRead(powerTogglePin) == 0 ? false : true;
  delay(1) //give the code time to read the value
  // read the value of the direction button
  directionButtonValue = digitalRead(directionButtonPin) == false ? false :
true;

//since the motor is controlled with PWM, we need to divide the value by 4
to be compatible with the range of accepted values
  motorSpeed = analogRead(potPin) / 4;

//if we see a change in the button's value, then make the boolean that
controls the motor change
  (toggleButtonValue != previousToggleButtonValue) && (toggleButtonValue ==
HIGH) ? motorEnabled = !motorEnabled : false;

//if we see a change in the direction button's value, then make the boolean
that controls the motor direction change
(directionButtonValue != previousDirectionButtonValue) &&
(directionButtonValue == HIGH) ?   motorDirection = !motorDirection :
false;

 //make the H Bridge choose a direction based on the value of
motorDirection boolean
  if (motorDirection == true) {
    digitalWrite(HBridge1Pin, HIGH);
    digitalWrite(HBridge2Pin, LOW);
  } else {
    digitalWrite(HBridge1Pin, LOW);
    digitalWrite(HBridge2Pin, HIGH);
  }

  // if the motor is supposed to be on, then set the speed of the motor to
the potentiometer value, or turn it off by setting it to 0
  motorEnabled == true  ? analogWrite(HBridgeTogglePin, motorSpeed) :
analogWrite(HBridgeTogglePin, 0);
  //track the previous state so this code can keep working
  previousDirectionButtonValue = directionButtonValue;
  previousToggleButtonValue = toggleButtonValue;
}
```

Final Working Proof

Video Link