

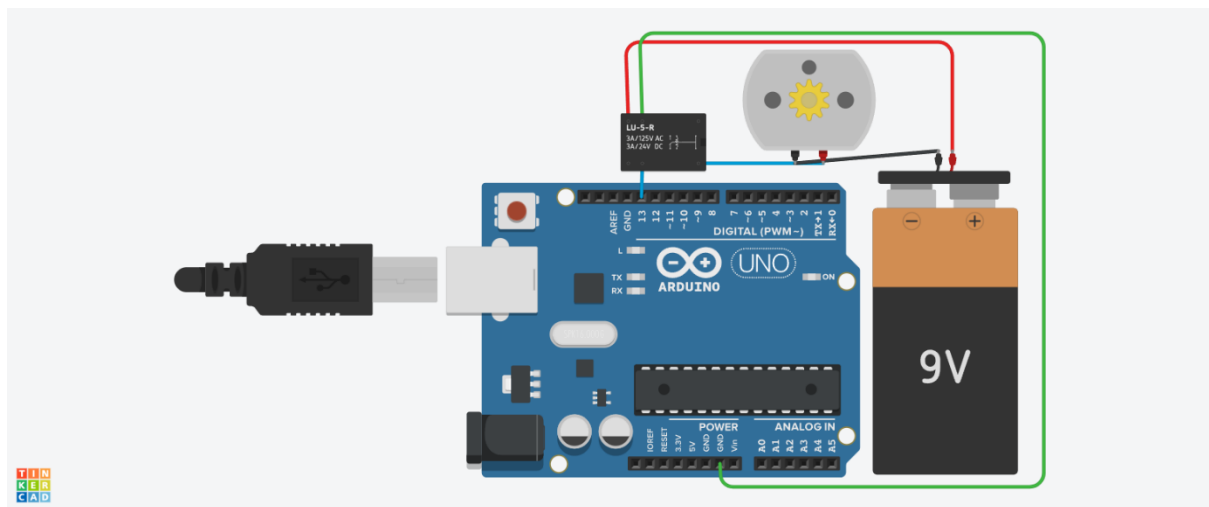
# Assignment 4

Name: Rinshi Kumari

Roll No.: 210108040

1. Use a relay with arduino to control DC Motor on/off.

<https://www.tinkercad.com/things/eg8yeadLqWo-relay-with-dc-motor/editel?sharecode=pnPVUgPWtmumLfog0eC1Ov9AApXu5KuJvs3Lv5W8KY>

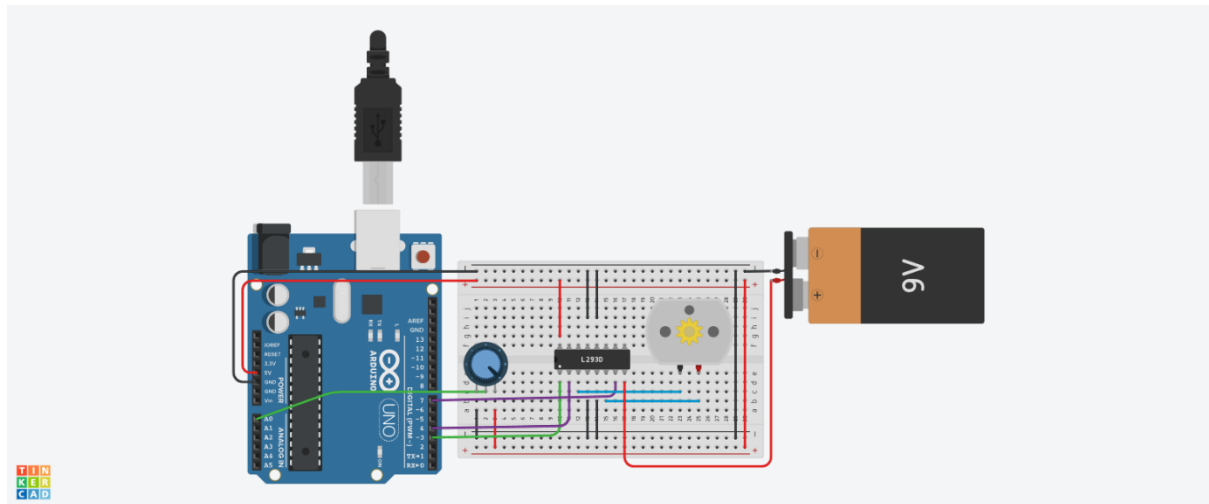


Code:

```
int relay = 13;
int buttonState = 0;
void setup() {
  pinMode(relay, OUTPUT);
}
void loop() {
  digitalWrite(relay, HIGH);
  delay(5000);
  digitalWrite(relay, LOW);
  delay(5000);
}
```

2. Use L293D Motor Driver to operate DC Motor at a variable speed controlled by a potentiometer.

[https://www.tinkercad.com/things/8HZMi6HfDXY-/editel?sharecode=oIGOVYpSoaFDCjEvYLWHApY-I7\\_yhTq4WdhOA8NF\\_8](https://www.tinkercad.com/things/8HZMi6HfDXY-/editel?sharecode=oIGOVYpSoaFDCjEvYLWHApY-I7_yhTq4WdhOA8NF_8)



Code:

```
int enable= 3;

int A = 4;

int B = 7;

int pos, veloc, oldpos=-1, Speed;

void setup() {

  pinMode(A, OUTPUT);

  pinMode(B, OUTPUT);

}

void loop(){

  pos = analogRead(A0);

  Speed= analogRead(A0);

  if (pos != oldpos){

    veloc = map(pos, 0, 1023, 0, 255);

    Speed = map(pos, 0, 1023, 0, 255);

    analogWrite(enable, veloc);

    oldpos = pos;
```

```

}

digitalWrite(A, LOW);

digitalWrite(B, HIGH);

delay(200);

}

```

3. Use L293D, DC Motor to simulate a 4-wheel Drive with 3 Digital and 1 analogue input. Inputs are to be used as follows:

Digital Input 000 -> Stop

001 -> Forward

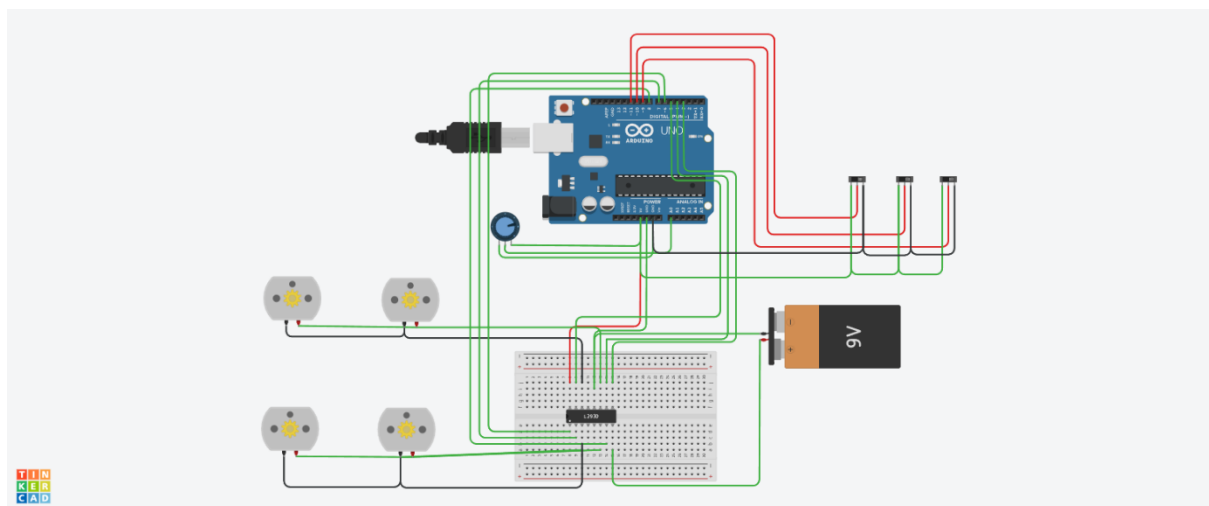
010 -> Backward

011 -> Right

100 -> Left

Analog input to Control Speed

<https://www.tinkercad.com/things/9UABxvYSBQ5-ee-312-lab-4-q3/editel?sharecode=kKy9tUylk0B38BB8aOVQMAAtsgNvTKogoa5m2c-lEvsA>



**Code:**

```

void setup() {
  pinMode(A0, INPUT);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
}

```

```
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, INPUT);
pinMode(10, INPUT);
pinMode(11, INPUT);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
digitalWrite(7, LOW);
digitalWrite(8, LOW);
}
void loop() {
  int x1 = digitalRead(9);
  int x2 = digitalRead(10);
  int x3 = digitalRead(11);
  int x = analogRead(A0);
  x = map(x, 0, 1023, 0, 255);
  analogWrite(3, x);
  analogWrite(6, x);
  if (x1 == 0 && x2 == 0 && x3 == 0) {
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
  }
  else if (x1 == 1 && x2 == 0 && x3 == 0) {
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
```

```

    digitalWrite(8, HIGH);
    digitalWrite(7, LOW);
}
else if (x1 == 0 && x2 == 1 && x3 == 0) {
    digitalWrite(5, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
}
else if (x1 == 0 && x2 == 0 && x3 == 1) {
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(7, LOW);
}
else if (x1 == 1 && x2 == 1 && x3 == 0) {
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
}
}
}

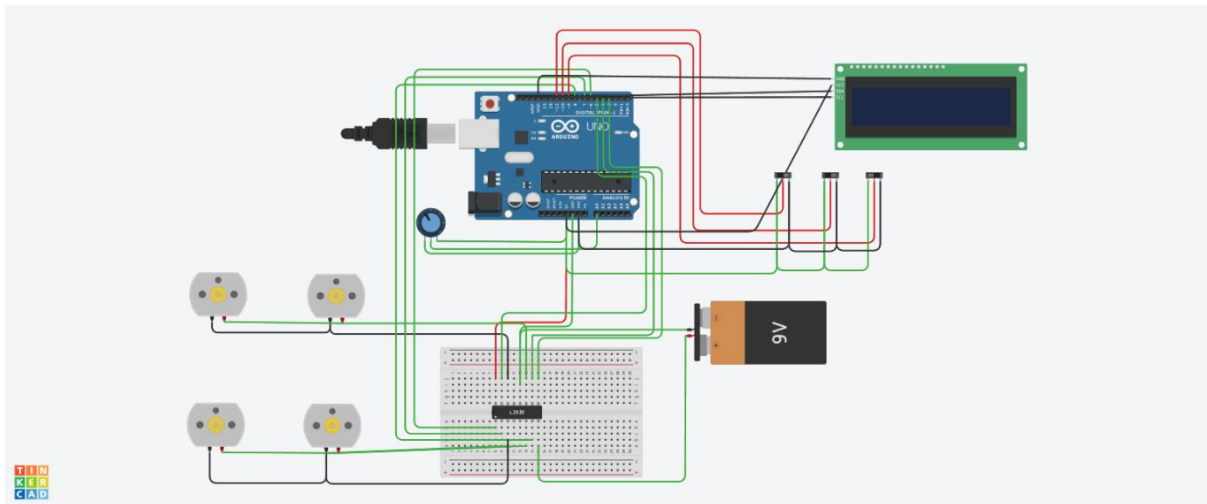
```

### Approach:

This Arduino code orchestrates a dynamic light show using inputs and outputs. It responds to three switches (x1, x2, x3) and an analog sensor (A0). The sensor's data influences the brightness of two LEDs (pins 3 and 6). Depending on switch combinations, LEDs on pins 4, 5, 7, and 8 illuminate in various patterns. The switches determine the display: all off, diagonal flashes, alternate flashes, or synchronized flashes. The analog sensor controls the LEDs' brightness. This compact code elegantly transforms inputs into an engaging visual symphony, with LEDs dancing to the rhythm of switches and sensor values.

#### 4. Integrate LCD Display with Q3 to Display movement and Speed of Bot.

<https://www.tinkercad.com/things/OrJFeyiq1Bw-ee-312-lab-4-q4/editel?sharecode=FeyFpNzyf63ZHq1W05qJPt3DdJfV1ZZjsxy8vNyN4vY>



#### Code:

```
#include<Adafruit_LiquidCrystal.h>

Adafruit_LiquidCrystal lcd_1(0);

#include <Wire.h>

void setup() {
  lcd_1.begin(16, 2);
  pinMode(A0, INPUT);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, INPUT);
  pinMode(10, INPUT);
  pinMode(11, INPUT);
}
```

```
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
digitalWrite(7, LOW);  
digitalWrite(8, LOW);  
lcd_1.setCursor(0, 0);  
lcd_1.print("Initializing...");  
Wire.begin();  
}
```

```
void loop() {  
    int x1 = digitalRead(9);  
    int x2 = digitalRead(10);  
    int x3 = digitalRead(11);  
    int x = analogRead(A0);  
    x = map(x, 0, 1023, 0, 255);  
    analogWrite(3, x);  
    analogWrite(6, x);  
  
    lcd_1.clear();  
  
    if (x1 == 0 && x2 == 0 && x3 == 0) {  
        digitalWrite(4, LOW);  
        digitalWrite(5, LOW);  
        digitalWrite(7, LOW);  
        digitalWrite(8, LOW);  
  
        lcd_1.setCursor(0, 0);  
        lcd_1.print("STOP");  
        lcd_1.setCursor(0, 1);  
    }
```

```
    lcd_1.print("Speed: 0");  
}  
else if (x1 == 1 && x2 == 0 && x3 == 0) {  
    digitalWrite(4, HIGH);  
    digitalWrite(5, LOW);  
    digitalWrite(8, HIGH);  
    digitalWrite(7, LOW);  
  
    lcd_1.setCursor(0, 0);  
    lcd_1.print("FRONT");  
    lcd_1.setCursor(0, 1);  
    lcd_1.print("Speed: " + String(x * 64));  
}  
else if (x1 == 0 && x2 == 1 && x3 == 0) {  
    digitalWrite(5, HIGH);  
    digitalWrite(4, LOW);  
    digitalWrite(7, HIGH);  
    digitalWrite(8, LOW);  
  
    lcd_1.setCursor(0, 0);  
    lcd_1.print("BACK");  
    lcd_1.setCursor(0, 1);  
    lcd_1.print("Speed: " + String(x*64));  
}  
else if (x1 == 0 && x2 == 0 && x3 == 1) {  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(8, HIGH);  
    digitalWrite(7, LOW);
```



```

    lcd_1.setCursor(0, 0);
    lcd_1.print("LEFT");
    lcd_1.setCursor(0, 1);
    lcd_1.print("Speed: " + String(x*64));
}
else if (x1 == 1 && x2 == 1 && x3 == 0) {
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);

    lcd_1.setCursor(0, 0);
    lcd_1.print("RIGHT");
    lcd_1.setCursor(0, 1);
    lcd_1.print("Speed: " + String(x*64));
}

delay(100);
}

```

### Approach:

This Arduino sketch choreographs a dynamic display using inputs, outputs, and a character LCD. It responds to three switches (x1, x2, x3) and an analog sensor (A0). LEDs on pins 3 and 6 illuminate with brightness determined by the sensor. Depending on switch combos, LEDs on pins 4, 5, 7, and 8 shine in diverse patterns. A character LCD narrates this performance, displaying movement directions and associated speeds. Whether it's a halt or forward, backward, left, or right motion, this code intertwines hardware and LCD storytelling, crafting an interactive narrative of luminous exploration in a concise yet captivating script.