Department of Electrical & Electronic Engineering

Rajshahi University of Engineering & Technology

LAB Report

Course Code : **EEE 3100**

Course Title : **Electronics Shop Practice**

Experiment No. : 04

Experiment Name : Control and Analyze the Operation of DC Motors Using Arduino and L293D Motor Driver IC.

Submitted By:

Name : Alsadi Md. Promi Milton

Roll : 2101098

Section : B

Submitted To:

Asif Zaman Rizve

Lecturer

Department of Electrical & Electronic Engineering

Rajshahi University of Engineering & Technology

Date of Experiment : 29 April, 2025

Date of Submission : 13 May, 2025

Experiment No.: 04

Experiment Name: Control and Analyze the Operation of DC Motors Using Arduino and L293D Motor Driver IC

Objective:

- To construct and analyze the operation of a basic DC motors control system with an Arduino Uno using the L293D motor driver IC.
- To learn how to control the direction and time of rotation of DC motors through Arduino programming.
- To gain hands on experience with circuit assembly and Arduino based motor driver control.

Theory:

DC motors are widely used in robotics and automation for their simplicity and precise control over speed and direction. The L293D motor driver IC, a dual H-Bridge motor driver, is ideal for controlling the direction and speed of DC motors, as it can handle higher currents than the Arduino pins directly. It allows for both forward and reverse motor rotation by controlling the direction of current flow. In this experiment, the Arduino Uno acts as the main control unit to drive two DC motors via the L293D. The motors are connected to the output pins of the L293D (Pins 3, 6 for Motor A and Pins 11, 14 for Motor B), while the enable pins (1 and 9) are linked to the PWM-capable digital pins (9 and 3) of the Arduino to control speed. Logic inputs (Pins 2, 7 for Motor A and Pins 10, 15 for Motor B) are connected to digital pins 8, 7, 5, and 4 for direction control. A 5V power supply powers the motors through the L293D, while the Arduino is powered via its USB connection, ensuring stable operation and effective control.

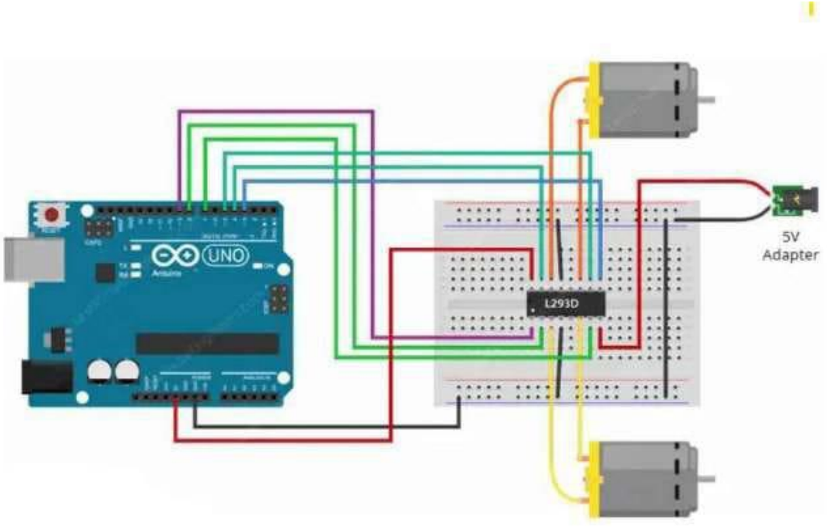

Fig.04.1: Control DC motors using Arduino uno and L293D motor driver IC.

Required Apparatus:

Table 4.1: Table For Required Apparatus

Sl. No.	Components	Specification	Quantity
1.	Arduino	UNO $_{\text{R3}}$	1
2.	L293D Motor Driver IC	Dual H-Bridge	1
3.	DC Gear Motor	6V, 200 RPM	2
4.	Wheel	-	3
5.	Chassis	-	1
6.	Breadboard	-	1
7.	Power Adapter	5V DC output	1
8.	Jumper wires	-	-
9.	Hot Gun	-	1

Circuit Diagram:

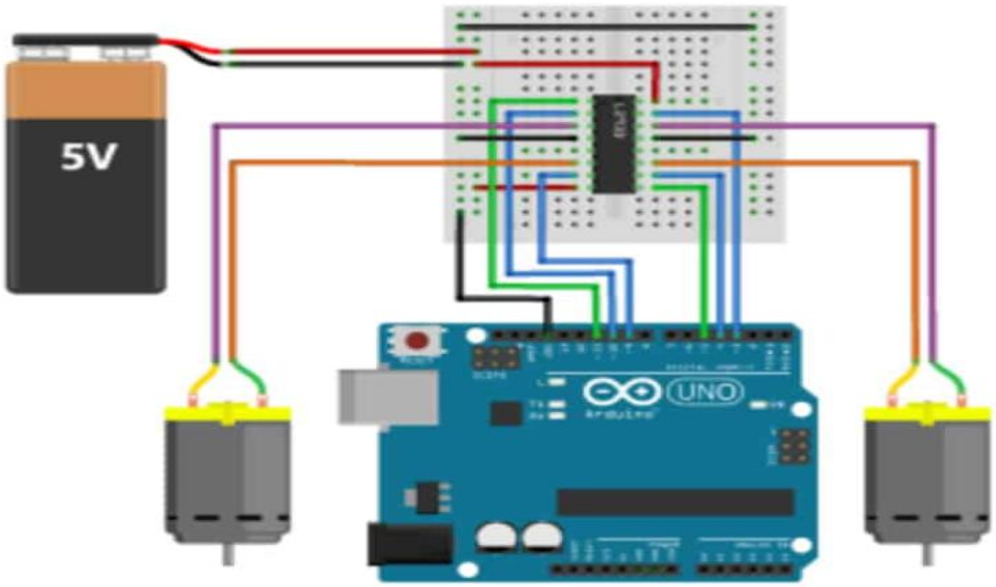

Fig.04.2: Circuit diagram for controlling DC motors using Arduino uno and L293D motor driver IC.

Experimental Setup:

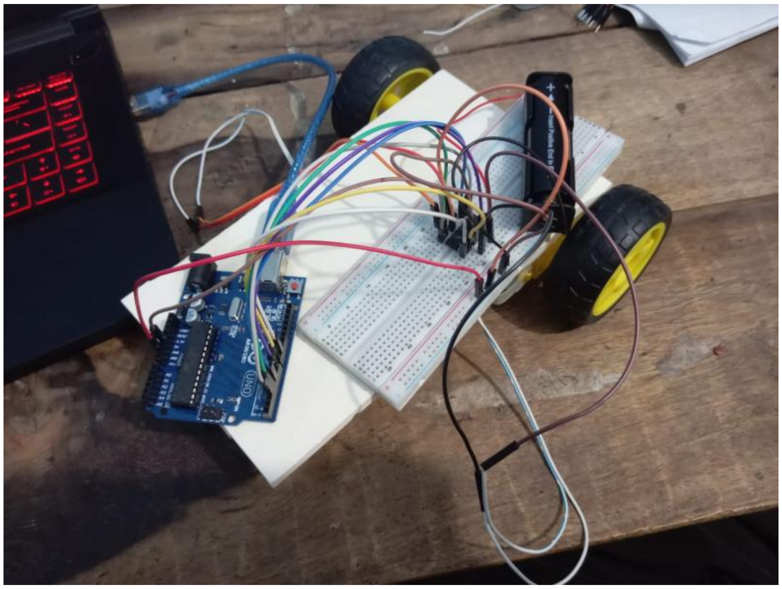

Fig.04.3: Experimental setup of the experiment

Code:

```
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;
// Motor B connections
int enB = 3;
int in3 = 5;
int in4 = 4;

void setup() {
	// Set all the motor control
pins to outputs
	pinMode(enA, OUTPUT);
	pinMode(enB, OUTPUT);
	pinMode(in1, OUTPUT);
	pinMode(in2, OUTPUT);
	pinMode(in3, OUTPUT);
	pinMode(in4, OUTPUT);

	// Turn off motors - Initial
state
	digitalWrite(in1, LOW);
	digitalWrite(in2, LOW);
	digitalWrite(in3, LOW);
	digitalWrite(in4, LOW);
}

void loop() {
	directionControl();
	delay(1000);
	speedControl();
```

```
      delay(1000);
}

// This function lets you control
spinning direction of motors
void directionControl() {
      // Set motors to maximum speed
      // For PWM maximum possible
values are 0 to 255
      analogWrite(enA, 255);
      analogWrite(enB, 255);

      // Turn on motor A & B
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      digitalWrite(in3, HIGH);
      digitalWrite(in4, LOW);
      delay(2000);

      // Now change motor directions
      digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);
      delay(2000);

      // Turn off motors
      digitalWrite(in1, LOW);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
}

// This function lets you control
speed of the motors
void speedControl() {
      // Turn on motors
      digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);

      // Accelerate from zero to
maximum speed
      for (int i = 0; i < 256; i++) {
            analogWrite(enA, i);
            analogWrite(enB, i);
            delay(20);
      }

      // Decelerate from maximum speed
to zero
      for (int i = 255; i >= 0; --i)
{
            analogWrite(enA, i);
            analogWrite(enB, i);
            delay(20);
      }

      // Now turn off motors
      digitalWrite(in1, LOW);
      digitalWrite(in2, LOW);
      digitalWrite(in3, LOW);
      digitalWrite(in4, LOW);
}
```

Experimental Procedure:

- Assembled the chassis, wheels, and motors.
- Connected the motors and L293D IC on a breadboard as per the circuit diagram.
- Connected L293D pins to Arduino Uno using jumper wires.
- Uploaded the Arduino code to the board.
- Powered the circuit using a 5V adapter.

Hands-on Learning Projects for Mastery of DC Motor Control with Arduino and L293D

Task No.: 01

Task Name: Designing Timed Sequential Control of Dual Motors in a Repetitive Cycle

Objective:

- To implement a time-based motor control system where Motor 1 runs for a specified duration, followed by Motor 2, and then both motors run together, forming a repeatable operational cycle.

Code:

```
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;
// Motor B connections
int enB = 3;
int in3 = 5;
int in4 = 4;

void setup() {
  // Set all the motor control pins to
outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}

void loop() {
  // Run Motor 1 for 5 seconds
  analogWrite(enA, 255);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  delay(5000);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);

  // Run Motor 2 for 5 seconds
  analogWrite(enB, 255);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  delay(5000);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);

  // Run both motors together for 2
seconds
  analogWrite(enA, 255);
  analogWrite(enB, 255);
  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(1000);  // Small pause before
the next cycle
}
```

Task No.: 02

Task Name: Implementing Conditional Activation in Timed Dual Motor Control

Objective:

- To design a control system where Motor 2 activates only upon receiving a user input or signal after Motor 1 stops, followed by simultaneous motor operation for a fixed duration, in a continuous cycle.

Code:

```
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;

// Motor B connections
int enB = 3;
int in3 = 5;
int in4 = 4;

void setup() {
  // Set all the motor control pins to
outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);

  // Initialize serial communication
  Serial.begin(9600);
   Serial.println("System  Ready.  Type
'start' to activate Motor 2.");
}

void loop() {
  // Run Motor 1 for 5 seconds
        Serial.println("Motor       1
Running...");
  analogWrite(enA, 255);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  delay(5000);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
   Serial.println("Motor  1  Stopped.
Type 'start' to activate Motor 2.");

  // Wait for the user to type 'start'
in the Serial Monitor
  while (Serial.available() == 0) {
    // Waiting for input
  }

  // Read the user input
         String        input        =
Serial.readStringUntil('\n');
  input.trim();  // Remove any leading
or trailing spaces

  if (input == "start") {
           Serial.println("Motor     2
Running...");

    // Run Motor 2 for 5 seconds
    analogWrite(enB, 255);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    delay(5000);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
           Serial.println("Motor     2
Stopped.");

    // Run both motors together for 2
seconds
       Serial.println("Running   Both
Motors...");
    analogWrite(enA, 255);
    analogWrite(enB, 255);
    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);
        Serial.println("Both   Motors
Stopped. Restarting Cycle...");
  } else {
      Serial.println("Invalid  input.
Please type 'start' to continue.");
  }

  delay(1000);  // Small pause before
the next cycle
}
```

Task No.: 03

Task Name: Developing an Interlock-Based Safety Control in Dual Motor Sequencing

Objective:

- To integrate a safety mechanism into the motor control sequence where the entire process halts upon detecting a danger signal, while preserving conditional and timed motor operation.

Code:

```
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;

// Motor B connections
int enB = 3;
int in3 = 5;
int in4 = 4;

// Timing variables
unsigned long previousMillis = 0;
unsigned long interval = 5000;  // 5
seconds
bool running = true;
int state = 0;

void setup() {
  // Set all the motor control pins to
outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Turn off motors - Initial state
  stopMotors();

  // Initialize serial communication
  Serial.begin(9600);
  Serial.println("System Ready. Type
'start' to activate Motor 2 or 'stop'
to end the cycle.");
}

void loop() {
  // Check for user input
  if (Serial.available() > 0) {
    String input =
Serial.readStringUntil('\n');
    input.trim();

    if (input == "stop") {
      Serial.println("Stopping all
motors. Exiting...");
      stopMotors();
      running = false;
      return;
    } else if (input == "start") {
      Serial.println("Motor 2
Running...");
      runMotorB();
      delay(5000);  // 5 seconds for
Motor 2
      stopMotors();

      Serial.println("Both Motors
Running...");
      runBothMotors();
      delay(2000);  // 2 seconds for
both motors
      stopMotors();
      Serial.println("Cycle Complete.
Restarting...");
      running = true;
    }
  }

  // Run Motor 1 for 5 seconds in a
non-blocking way
  if (running && state == 0) {
    Serial.println("Motor 1
Running...");
    runMotorA();
    previousMillis = millis();
    state = 1;
  }

  // Check if 5 seconds have passed
for Motor 1
  if (state == 1 && millis() -
previousMillis >= interval) {
    stopMotors();
    Serial.println("Motor 1 Stopped.
Type 'start' to activate Motor 2 or
'stop' to end the cycle.");
    state = 2;
  }
}

void runMotorA() {
  analogWrite(enA, 255);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
}

void runMotorB() {
  analogWrite(enB, 255);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void runBothMotors() {
  analogWrite(enA, 255);
  analogWrite(enB, 255);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}

void stopMotors() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  Serial.println("Motors Stopped.");
}
```

Results:

In this experiment, the operation of two DC gear motors was controlled using an Arduino Uno and an L293D motor driver IC. The motors operated as intended, with precise speed and direction control, responsive user inputs, and immediate stop functionality, demonstrating the effective use of PWM and H-Bridge for real-time motor control and verifying the effectiveness of the code in achieving directional and speed control of DC motors.

Discussion:

The experiment successfully demonstrated how DC motors can be controlled using an Arduino Uno and an L293D motor driver IC. At first, the motors weren't working. Then we had to change L293D IC and even Arduino. Then we had to do some modifications to successfully perform this experiment. Maybe there were some issue with improper wirings, loose connections, faulty IC etc.

Justification of CO2 and PO (j): Explain the design and reasoning behind the timed, conditional, and safety-interlocked motor operations, and clearly communicate your setup and results.

In designing the control system, external 5V power supply is connected on 8 no pin ; 5V output of Arduino on 16 no pin; 4,13 and Arduino GND on supply negative or ground; input and enable pins (1,2,7,10,15,9) of IC with Arduino digital output pins (9,8,7,5,4 and 3); motor-1 between output pins 3-6 and motor 2 between 14-11 and finally Arduini uno with laptop through USB.

Conclusion:

The experiment effectively demonstrated a structured motor control system using Arduino and L293D, achieving both speed modulation and direction control. Different operational modes were tested, proving the adaptability of the setup for various applications. Careful circuit design, programming logic, and safety precautions ensured smooth functionality and prevented damage to components.

Reference:

1. https://lastminuteengineers.com/l293d-dc-motor-arduino-tutorial/