Title of Experiment:

Stepper Motor Control via Microcontroller (Using L293D Driver)

Aim:

To control the rotation, direction, and speed of a stepper motor using Arduino and an L293D motor driver.

Apparatus:

- Arduino Uno board
- L293D Motor Driver IC or module
- Bipolar or Unipolar Stepper Motor (e.g., 28BYJ-48 or NEMA17)
- Power supply (5V–12V depending on motor)
- Jumper wires
- Breadboard (optional)
- Arduino IDE

Precautions:

1. Use an external power supply to drive the motor, not the Arduino 5V pin.
2. Double-check motor and L293D wiring before powering on.
3. Connect ground of motor power and Arduino together (common ground).
4. Do not change connections while the circuit is powered.
5. Make sure L293D does not overheat; use a heatsink if needed.

Theory:

A **stepper motor** is an electromechanical device that moves in precise steps when its coils are energized in a specific sequence. It is widely used in robotics, CNC machines, and automation systems.

The **L293D** is a dual H-bridge motor driver IC that can drive two motors or one stepper motor. It allows current to flow in both directions through each motor coil, enabling control of direction and stepping sequence.

The **Arduino** sends control signals to the L293D, which in turn energizes the motor coils to move the stepper motor either **clockwise** or **counter-clockwise**. By adjusting the **delay between steps**, we can control the **speed** of rotation.

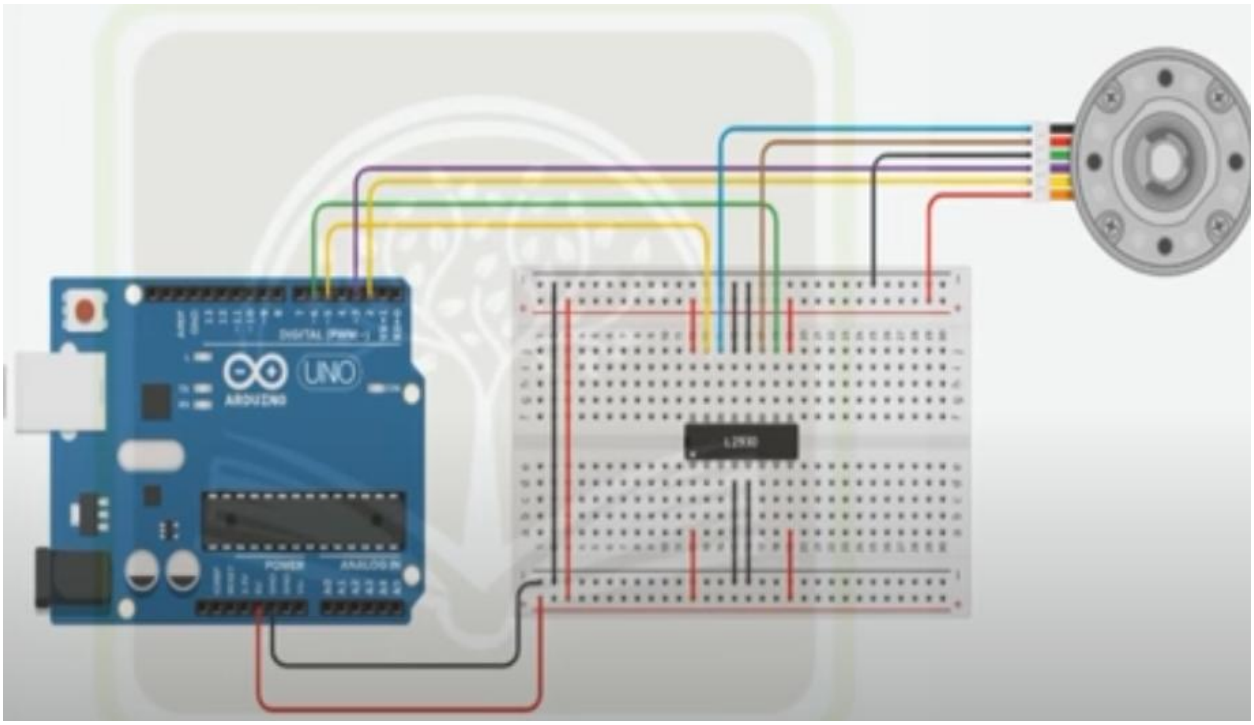
Diagram (Textual Description):

Example Wiring for 4-wire Stepper Motor with L293D:

- Connect motor coils to L293D OUT1, OUT2, OUT3, and OUT4.
- Connect IN1, IN2, IN3, IN4 of L293D to Arduino pins 8, 9, 10, 11.
- VCC1 (logic) of L293D → Arduino 5V
- VCC2 (motor supply) of L293D → External 9V/12V supply
- GND of L293D → Arduino GND and motor power GND
- EN1 and EN2 → Connect to 5V or Arduino HIGH

Procedure:

1. Connect the stepper motor to L293D driver outputs (OUT1–OUT4).
2. Connect L293D input pins (IN1–IN4) to Arduino digital pins (e.g., 8–11).
3. Connect L293D to appropriate power sources.
4. Write and upload code to Arduino using Arduino IDE.
5. Observe the motor rotating in clockwise and counter-clockwise directions.
6. Adjust delay values in code to see changes in motor speed.



Arduino Programming (with Comments):

```
// Include Stepper library
#include <Stepper.h>

// Define number of steps per revolution (depends on motor; commonly 200)
const int stepsPerRevolution = 200;

// Create stepper motor object with 4 control pins connected to L293D
// (IN1 = 8, IN2 = 9, IN3 = 10, IN4 = 11)
Stepper myStepper(stepsPerRevolution, 8, 10, 9, 11);

void setup() {
  // Set motor speed in RPM
```

```
  myStepper.setSpeed(60); // You can vary this to control speed (e.g., 30, 100)

  // Initialize serial monitor
  Serial.begin(9600);
  Serial.println("Stepper Motor Control Start");
}

void loop() {
  // Rotate clockwise 1 full revolution
  Serial.println("Clockwise rotation");
  myStepper.step(stepsPerRevolution); // 200 steps for 1 revolution
  delay(1000);

  // Rotate counterclockwise 1 full revolution
  Serial.println("Counterclockwise rotation");
  myStepper.step(-stepsPerRevolution); // negative for opposite direction
  delay(1000);

  // Change speed
  myStepper.setSpeed(120); // Increase speed to 120 RPM
  Serial.println("Faster clockwise rotation");
  myStepper.step(stepsPerRevolution);
  delay(1000);

  // Slow down speed
  myStepper.setSpeed(30); // Decrease speed to 30 RPM
```

```
  Serial.println("Slower counterclockwise rotation");

  myStepper.step(-stepsPerRevolution);

  delay(1000);

}
```

Conclusion:

The experiment successfully demonstrated how to control a stepper motor using Arduino and the L293D motor driver. The motor was rotated in both clockwise and counter-clockwise directions using step sequences. Motor speed was controlled by changing the delay between steps, illustrating basic motion control.

---

4 Short Questions and Answers:

Q1: What is the purpose of the L293D driver in this experiment?
Answer: The L293D driver allows Arduino to control higher current required by the stepper motor and enables bidirectional movement.

Q2: How is the motor direction controlled in the code?
Answer: By changing the order of the coil energizing sequence (normal vs. reverse loop), the motor rotates clockwise or counter-clockwise.

Q3: How can the motor speed be changed?
Answer: By changing the `delay()` value between steps — smaller delay means faster rotation.

Q4: Why is an external power supply recommended for the motor?
Answer: Stepper motors draw more current than the Arduino can provide; an external supply ensures proper and safe operation.

Source: https://youtu.be/O8jEk7KuNhs?si=1JvwNGVIsRglR2Y2