Fan Speed Control Using Light

1. Aim of the project

The Aim of this project is to design a system to control the direction and speed of a **DC motor** using an **L293D Motor Driver IC** interface with **Arduino Uno**. Controlling **DC motor** based on light using the Tinkercad simulation platform for circuit design, testing and troubleshooting.

---

2. Purpose of the Project

The purpose of this project is to develop a light-controlled DC motor system that allows for both speed and direction control using the L293D Motor Driver IC interfaced with an Arduino Uno. This project focuses on utilizing light intensity, sensed through an LDR, to adjust the motor's behaviour, thereby demonstrating how environmental inputs can control electromechanical systems. The project also emphasizes virtual circuit design and testing using the Tinkercad simulation platform, which helps in understanding circuit functionality, debugging, and safe prototyping. This hands-on project aims to build core embedded system skills such as motor control, sensor interfacing, signal processing, and embedded C programming.

---

3. Application of the Project

- Automatic Fan Speed Control:
  Adjusts the speed of cooling fans based on ambient light or temperature conditions, improving energy efficiency in homes or greenhouses.
- Solar Panel Tracking Systems:
  Helps rotate solar panels toward the direction of maximum sunlight by sensing light intensity, thereby increasing energy collection.
- Smart Window Curtains/Blinds:
  Controls the opening or closing of curtains or blinds automatically based on sunlight, enhancing comfort and reducing heat.

- Light-Based Security Systems:
  Activates or deactivates security motors or barriers based on changes in light, useful in automatic gates or restricted zones.

---

4. Components Required

S. No.	Quantity	Components
1.	1	Arduino Uno
2.	1	L293D Motor Driver IC
3.	1	Light Dependent Resistor
4.	1	DC Motor
5.	1	Breadboard
6.	4	Jumper Wires
7.	1	Power Supply (5V)

---

5. What is PWM? How is it used in Arduino?

PWM (Pulse Width Modulation) is a technique used to control the amount of power delivered to electrical devices by switching the power ON and OFF at a high frequency.

In Arduino, the Analog Write(pin, value) function generates PWM signals.

The value ranges from 0 to 255, 0 means 0% duty cycle (always OFF)

255 means 100% duty cycle (always ON)

This allows speed control of the motor by varying the average voltage supplied to the motor.

---

6. H-Bridge Concept and L293D IC

An H-Bridge allows current to flow in one direction at the same time. It has four switches, at the same time two switches will be closed and rest two will be opened.

- Rotate the motor clockwise
- Rotate the motor counterclockwise
- **Stop the motor**

L293D Motor Driver IC :

The **L293D Motor Driver IC** is based on **H-Bridge** concept. It has dual **H-Bridge Motor Driver IC**. Which we can use to control two DC Motors independently.

Pin Diagram & Functions:

Pin No.	Name	Function
1	Enable 1 (EN1)	Enables Motor 1 (connected to PWM pin)
2	Input 1 (IN1)	Controls direction (from Arduino)
3	Output 1 (Out1)	Connected to Motor Terminal A
4-5	GND	Connected to Motor Terminal B
6	Output 2(Out2)	Connected to Motor Terminal B
7	Input 2(INT2)	Controls Direction (From Arduino)
8	Vcc2	Motor Power Supply (typically 6-12V)
9	Enable 2 (EN2)	Enables Motor 2 (Unused here)
10-15	IN3-OUT4	For second motor (Not used here)
16	Vcc1	Logic Voltage (5v from Arduino)

What Is Mapping in Arduino?

Mapping is the process of converting a value from one numerical range to another. In embedded systems, especially when working with Arduino, mapping is commonly used to adjust sensor inputs to a range suitable for actuators. For example, an LDR (Light Dependent Resistor) provides an analog value between 0 and 1023, depending on the light intensity.

However, to control a DC motor’s speed using PWM (Pulse Width Modulation), we typically need values between 0 and 255. The map()

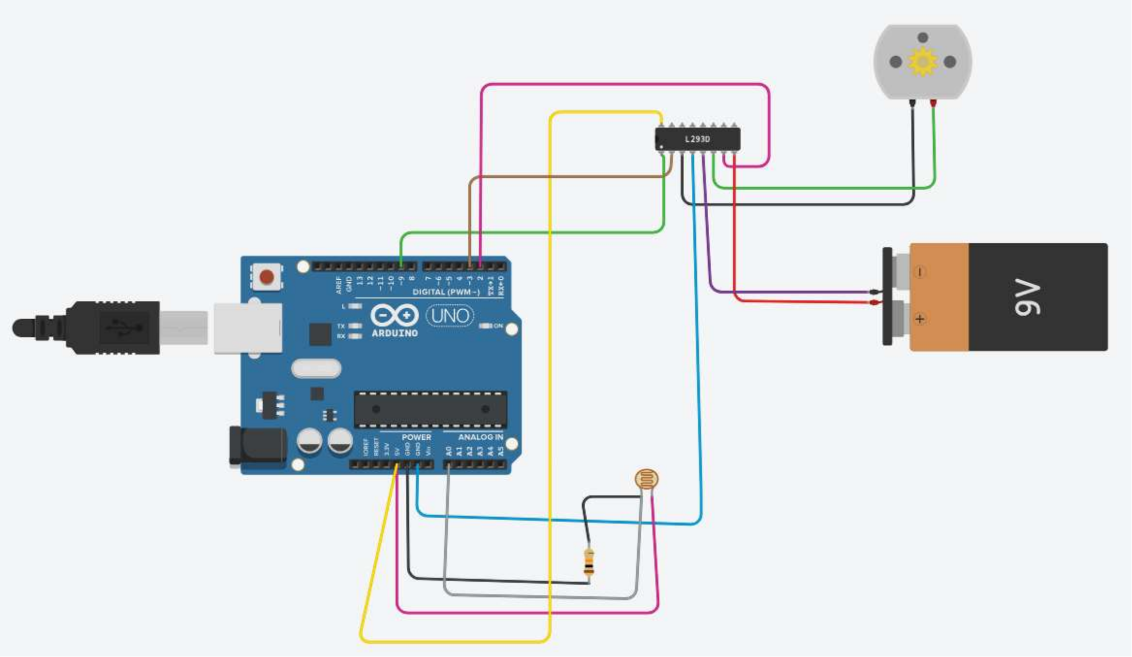
function in Arduino helps in this conversion by scaling the input range to the desired output range.

This ensures that the motor speed changes smoothly and proportionally to the amount of light detected by the sensor. Mapping makes it easier to handle real-world inputs and apply them effectively to outputs, improving the overall response and performance of embedded systems.

7. Circuit Diagram:

The Circuit consists of:

- Arduino Uno connected to EN1, EN2, IN1, GND pin of L293D.
- OUT1 and OUT2 connected to the motor terminals.
- Power1 connected to Arduino 5v and Power2 connected external supply.
- LDR is Analog sensor which is connected to Analog pin of Arduino
- LDR has two terminals.

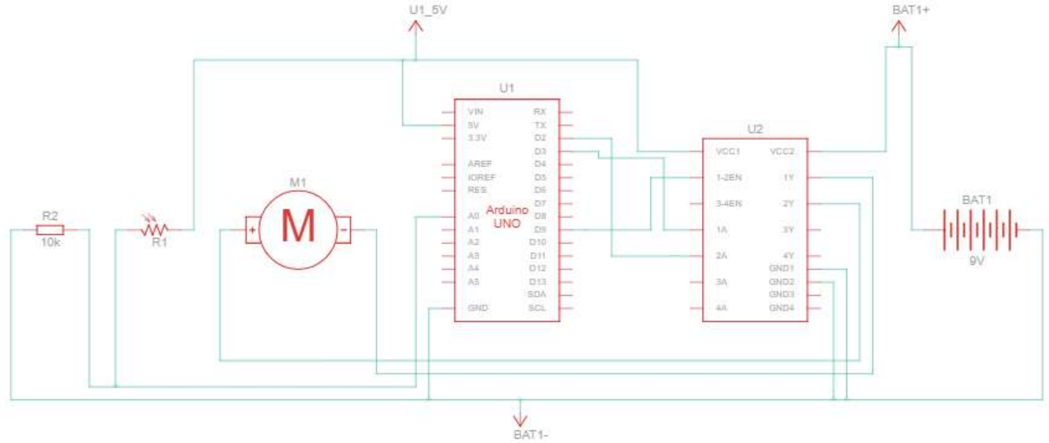


8. Layout (Tinkercad Simulation):

In the Tinkercad Simulation:

- Use breadboard to mount the L293D IC.

- Connected IN1, IN2, to digital pins (D2, D3) of Arduino Uno.
- Connected EN1 to PWM pin (D9) of Arduino.
- Connected Motor to OUT1 and OUT2.
- Use breadboard to mount the LDR.
- Connected Terminal1 of LDR to Analog pin (A0) of Arduino.
- Connected Terminal2 of LDR to 5v of Arduino.
- Upload code via Arduino editor in Tinkercad.



9. **Code With Explanation**

   Let's break down and justify each part of your Arduino program step by step, based on what it's trying to achieve — likely to control a DC motor using PWM via the L293D motor driver and based on LDR value.

Code Analysis

```
int pin1 = 2, pin2 = 3, pin3 = 9;
int analog = A0;
void setup ()
{
 pinMode(pin1, OUTPUT);
 pinMode(pin2, OUTPUT);
 pinMode(pin3, OUTPUT);
 Serial.begin(9600);
}

void loop ()
{
 int value = analogRead(analog);
 int speed = map(value,0,1023,0,255);
 Serial.print(" | Speed: ");
 Serial.println(speed);

 digitalWrite(pin1, HIGH);
 digitalWrite(pin2, LOW);
 analogWrite(pin3, speed);
 delay (2000);
}
```

Justification

- This Arduino code is written to control the speed and direction of a DC motor based on the input from an Analog sensor, such as an LDR (Light Dependent Resistor).

- The Analog value is read from pin A0 and then mapped from a 10-bit range (0–1023) to an 8-bit PWM range (0–255), which is required for motor speed control using the analogWrite() function.
- The direction of the motor is set using two digital pins (pin1 and pin2), where one pin is set HIGH and the other LOW to rotate the motor in a specific direction.
- The pin3 is connected to the EN (Enable) pin of the motor driver, where PWM signals are used to vary motor speed. The Serial.begin(9600) and Serial.print() functions are used for debugging and monitoring the speed values on the Serial Monitor.
- A delay of 2 seconds is added to observe the effect clearly. This code demonstrates real-time control of a motor based on analog input and effectively integrates concepts of PWM, sensor interfacing, and motor driver logic in embedded systems.

10. Conclusion

This project demonstrates the fundamental concepts of motor control systems using the L293D motor driver and a Light Dependent Resistor (LDR). It showcases how to control the speed of a DC motor based on varying light intensity and how to safely change the motor's direction using logic signals.

The L293D driver ensures the motor operates efficiently without damaging the microcontroller. Additionally, the project is designed and tested virtually using Tinkercad, providing a safe and damage-free environment for learning and experimentation. Overall, this project serves as a strong foundation for understanding embedded applications in motor control using sensors like LDR.