



Namal University, Mianwali

Department of Electrical Engineering

EEN-221 (L) – Electric Machines (Lab)

Lab Project

Project Title DC Motor control using the bleutooth module

Student Name	Roll No.
Tahir Ali	NUM-BSEE-2023-28
Mohsin Raza	NUM-BSEE-2023-04

Date: 15/01/2025

Instructor: Dr. Farrukh Qureshi

Lab Engineer: Engr. Faizan Ahmad

Project Report: DC Motor Control Using Mobile Phone

Introduction

This project focuses on controlling two DC motors using a mobile phone via Bluetooth communication. It utilizes Arduino as the control unit, coupled with a Bluetooth module for wireless interaction. The system provides a user-friendly interface, where motor operations can be controlled remotely, making it ideal for applications in robotics and automation. The design is compact and efficient, offering an accessible way to manage motor-driven tasks wirelessly.

Literature Review

Bluetooth-controlled robotics, focuses on control algorithms, communication protocols, mobile application development, hardware configurations, and integration with other technologies. Many projects utilize direct control methods, where commands from a Bluetooth device, such as a smartphone, are used for real-time control. Alternatively, some systems incorporate feedback control using sensors like ultrasonic sensors to adjust movements based on environmental conditions, enhancing navigation and obstacle avoidance.

In terms of communication, most projects rely on serial communication with Bluetooth modules like the HC-05 for command transmission, while more advanced projects may develop custom protocols to optimize communication and error handling. Regarding mobile application development, many utilize existing apps (e.g., Bluetooth Terminal) for quick command input, whereas others create tailored mobile applications for enhanced user interfaces and specific functionalities.

Overall, the landscape of Bluetooth-controlled robotics encompasses a variety of methodologies, each with unique strengths. Future advancements may focus on improving communication protocols, user interfaces, and sensor integration to create more intelligent and responsive robotic systems.

Components Used

The following components were used in the project:

- **Arduino Uno:** The microcontroller that processes input and controls the motor driver.
- **Bluetooth Module (HC-05):** Facilitates communication between the mobile phone and the Arduino.
- **Push Buttons:** For manual motor control in addition to Bluetooth.
- **Battery (9V or Li-ion):** Powers the system.
- **Two DC Motors:** Provide motion for the project.
- **Motor Driver Module (L298N):** Controls the direction and speed of the motors.
- **Connecting Wires & Breadboard:** For establishing the circuit connections.
- **Mobile Phone:** The user interface to send commands via Bluetooth.

Circuit Diagram

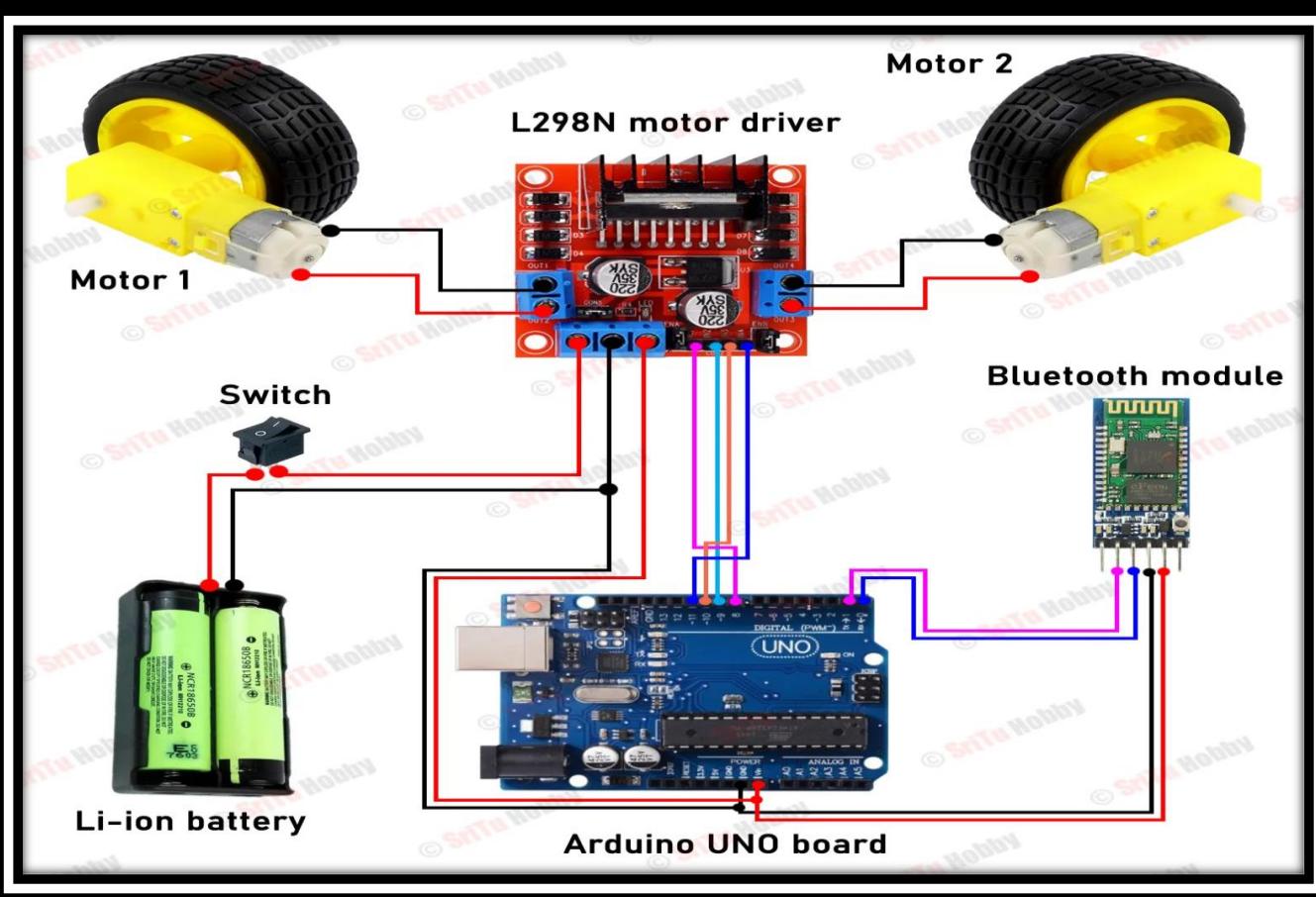


Figure 1

Pin Configuration

Component	Pin	Description
Arduino	8	IN1 (Motor A direction control)
Arduino	9	IN2 (Motor A direction control)
Arduino	10	IN3 (Motor B direction control)
Arduino	11	IN4 (Motor B direction control)
HC-05	VCC	Power supply (5V)
HC-05	GND	Ground
HC-05	TXD	Transmit data to Arduino
HC-05	RXD	Receive data from Arduino

Figure 2

Working Principle

The project operates as follows:

- **Bluetooth Communication:** The mobile phone sends commands to the Arduino via the Bluetooth module.
- **Command Processing:** Arduino interprets the commands and generates the corresponding motor control signals.
- **Motor Control:** Based on the received command, Arduino drives the motor driver, which powers the motors accordingly. For example:
 - Command 'F' causes the motors to rotate forward.
 - Command 'B' causes the motors to rotate backward.
 - Commands 'L' and 'R' control left and right turns, respectively.
 - Command 'S' stops the motors.
- **Manual Control:** Push buttons provide an alternate method of controlling the motors if Bluetooth communication is not available.

Methodology: Working of Each Module

1. Setup Block

- **Initialization:**
 - Initializes serial communication for Bluetooth at a specified baud rate.
- **Pin Configuration:**
 - Defines motor driver pins as outputs and sensor pins as inputs.
- **Bluetooth Module Setup:**
 - Prepares the Bluetooth module for communication with the mobile device.

2. Loop Block

- **Command Reception:**
 - Continuously checks for incoming commands from the Bluetooth device.
- **Command Processing:**
 - Interprets received commands to determine the appropriate action (e.g., forward, backward, left, right, stop).

3. Movement Functions

- **Moving Forward:**
 - Activates motors to drive the robot forward.
- **Moving Backward:**
 - Activates motors to reverse the robot.
- **Turning Left:**
 - Adjusts motor operations to turn the robot left.
- **Turning Right:**
 - Adjusts motor operations to turn the robot right.
- **Stopping:**
 - Stops all motor functions to halt the robot.

Arduino Code

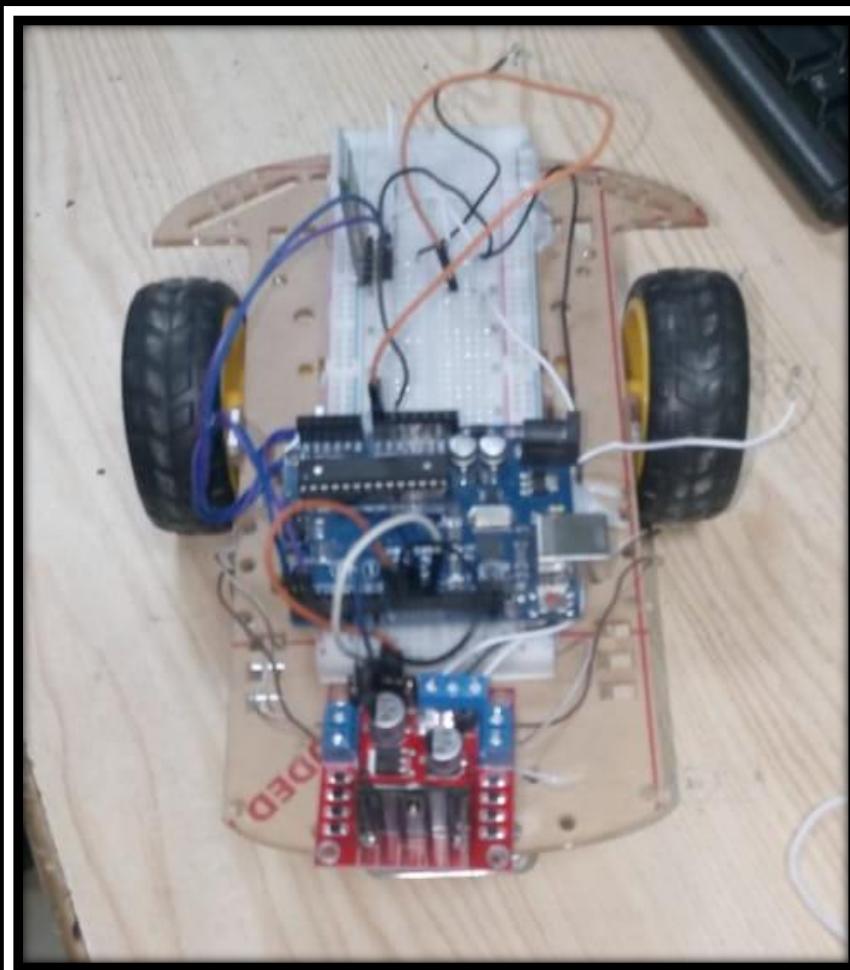
```
#define IN1 8
#define IN2 9
#define IN3 10
#define IN4 11
void setup() {
    Serial.begin(9600);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    Stop(); // Ensure motors are stopped initially
    Serial.println("Bluetooth Car Ready!");
}
void loop() {
    if (Serial.available() > 0) {
        char value = Serial.read();
        value = toupper(value);
        Serial.print("Received Command: ");
        Serial.println(value);
        switch (value) {
            case 'F':
                Forward();
                break;
            case 'B':
                Backward();
                break;
            case 'S':
                Stop();
                break;
            case 'L':
                Left();
                break;
            case 'R':
                Right();
                break;
            default:
                Serial.println("Invalid Command, Stopping!");
                Stop();
                break;
        }
    }
}

void Forward() {
    Serial.println("Moving Forward");
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void Backward() {
    Serial.println("Moving Backward");
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}
```

```
void Stop() {  
    Serial.println("Stopping");  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, LOW);  
}  
  
void Left() {  
    Serial.println("Turning Left");  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
}  
  
void Right() {  
    Serial.println("Turning Right");  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
}
```

Hard Ware



Results

The Bluetooth module successfully communicated with the Arduino, and the DC motors responded as expected to the commands from the mobile phone.

Manual control using push buttons also functioned properly, providing a backup for motor control.

The motor operation showed minimal delay, confirming that the Bluetooth communication was efficient and effective.

Challenges and Solutions

- **Bluetooth Pairing Issues:** There were difficulties initially in pairing the Bluetooth module with the mobile phone. This was resolved by ensuring proper configuration and using a dedicated Bluetooth app for control.
- **Power Supply Fluctuations:** The motors exhibited inconsistent performance due to insufficient power. The solution was to use a high-capacity battery for stable voltage supply.
- **Wiring Confusion:** The complex wiring setup led to mistakes in the circuit connections. To mitigate this, clear labeling of wires and the creation of a detailed circuit diagram helped resolve the issue.

Conclusion

This project successfully demonstrates the integration of Arduino, Bluetooth, and motor control to wirelessly operate DC motors. The system provides a reliable, user-friendly solution for remote motor control, suitable for various applications in robotics and automation. The optional manual control enhances flexibility, making the system more adaptable to different use cases.

References

- Arduino, "Arduino Documentation," *Arduino*, [Online]. Available: <https://www.arduino.cc>. [Accessed: Jan. 16, 2025].
- Arduino, "How to control a DC motor with Bluetooth," *YouTube*, 2017. [Online]. Available: <https://www.youtube.com/watch?v=Ey4xoG970Go>. [Accessed: Jan. 16, 2025].
- R. Nerd, "Arduino control DC motor via Bluetooth," *Random Nerd Tutorials*, [Online]. Available: <https://randomnerdtutorials.com/arduino-control-dc-motor-via-bluetooth/>. [Accessed: Jan. 16, 2025].
- Electrobes, "HC-05 Bluetooth Module," *Electrobes*, [Online]. Available: <https://electrobes.com/product/hc-05-bluetooth-module/>. [Accessed: Jan. 16, 2025].