**A yellow and black logo

Description automatically generatedA logo of a university of sciences and technology

Description automatically generatedDEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERING**

**COLLEGE OF E&ME, NUST, RAWALPINDI**

**PHY-102 APPLIED PHYSICS**

**DE-47-CE-B**

**PROJECT REPORT**

**Project:**

To make a cardboard/metal/wood car which should be able to move on its own either through an electric motor or some mechanical mechanism.

**SUBMITTED TO:** Lec. Sabeen Malik

**SUBMITTED BY:**

|  |  |
| --- | --- |
| **STUDENT NAME** | **CMS ID** |
| NS MUHAMMAD HASHIR EHTISHAM | 543606 |
| NS TAIMOOR ATIF | 548296 |
| NS LAMEEA MUBASHIR KHAN | 570918 |
| PC TOOBA FATIMA | 577687 |

**Project Overview:**

This project involves the development of a Bluetooth-controlled robotic car using an Arduino Uno as the main microcontroller. The system integrates an L298N motor driver, an HC-05 Bluetooth module, and a pair of DC motors to enable wireless, real-time movement control.

The Arduino Uno acts as the central processing unit, receiving control commands from a smartphone or Bluetooth-enabled device via the HC-05 module. These commands are then interpreted and translated into motor actions. The L298N motor driver serves as the interface between the Arduino and the motors, providing the required current and directional control for the car's movement. By adjusting the motor speed and direction, the car can move forward, backward, left, or right based on user input.

Overall, this project demonstrates a simple yet effective embedded system application that combines wireless communication, motor control, and microcontroller programming to create a fully functional Bluetooth-operated robotic car.

**Parts and Components:**

1) Arduino Uno (or any compatible board)



2) Bluetooth module (HC-05/HC-06)

A close-up of a blue circuit board

AI-generated content may be incorrect.

3) Motor driver (L298N)

A close-up of a circuit board

AI-generated content may be incorrect.

4) TT Gear motors and wheels

A group of yellow and black wheels

AI-generated content may be incorrect.

5) Acrylic Car Floor

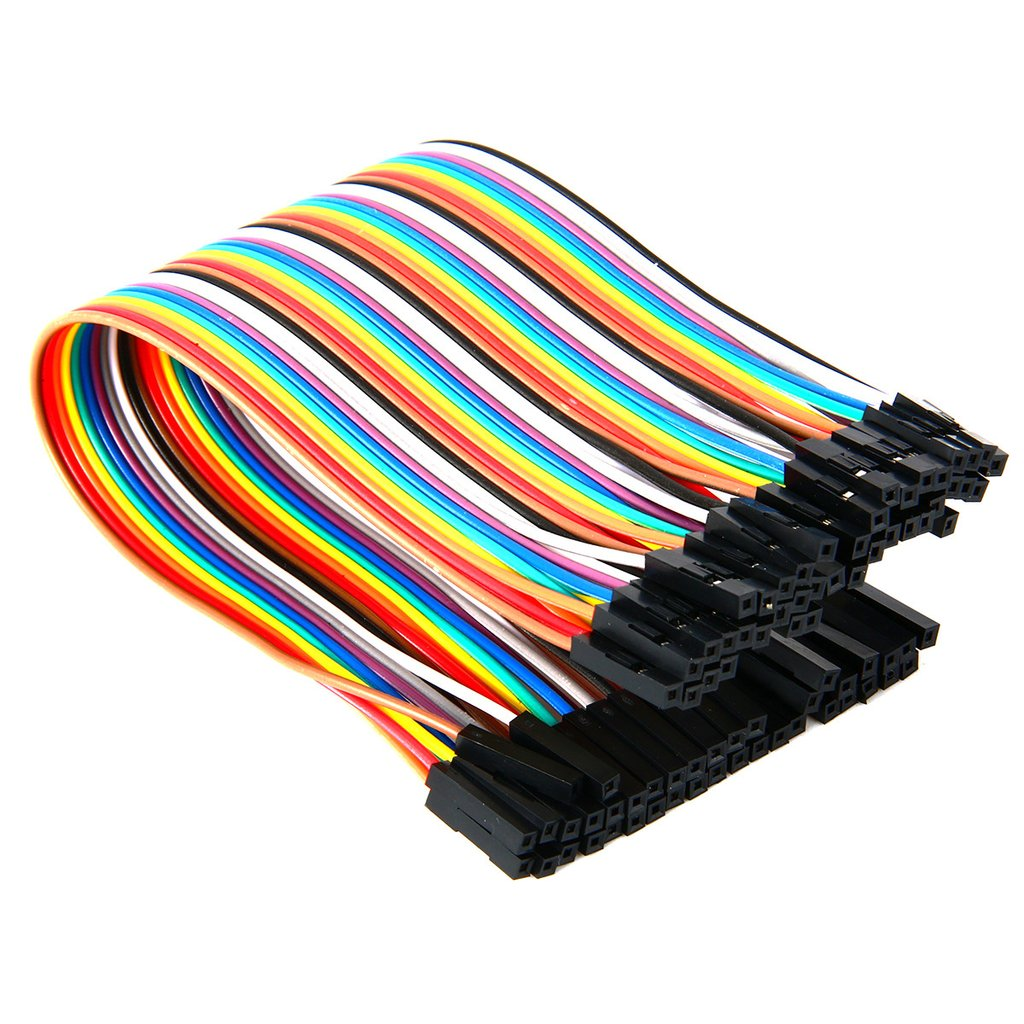
A close-up of a cardboard piece

AI-generated content may be incorrect.

6) Li - Iron batteries



7) Jumper wires



8) Nylon Hex Standoffs Male- Female and Bolts



9) 1k ohm Resistors

A close-up of a resistor

AI-generated content may be incorrect.

10) Smartphone with a Bluetooth control app

**Theoretical Background and Pre-requisite Knowledge:**

Below is a clear, complete theoretical background and prerequisite knowledge you need before building an Arduino-based Bluetooth-controlled car using an HC-05 Bluetooth module and an L298N motor driver.

1. Basic Electronics Knowledge

Before building the car, you should understand:

● Voltage, Current, and Power

• Difference between 5V logic (Arduino) and motor supply voltage (often 6–12V).

• How current draw from motors affects battery choice.

● Circuit Connections

• How to connect components using jumper wires.

• Identifying GND common connection (all components must share ground).

● Reading Basic Schematics

• Understanding labels like VCC, GND, IN1–IN4, ENA, ENB.

2. Arduino Fundamentals

You need basic microcontroller skills:

● Arduino Uno (or similar) pin modes

• pinMode(), digitalWrite(), analogWrite() (PWM).

● Serial Communication

• Understanding the Arduino Serial object.

• How Bluetooth uses serial communication as well.

● Writing Simple Control Logic

Example concepts:

• How to read incoming characters from Bluetooth.

• How to map characters to motor actions (F,B,L,R,S).

3. DC Motors & Motor Drivers (L298N)

● How a DC Motor Works

• Voltage controls speed.

• Reversing polarity changes direction.

● Motor Drivers

The L298N is an H-bridge motor driver; you should know:

How an H-bridge works

• It lets the Arduino control motor direction and speed.

• Arduino cannot drive motors directly because motors draw high current.

L298N Pins

• IN1, IN2 control Motor A direction

• IN3, IN4 control Motor B direction

• ENA/ENB pins (enable pins) for speed control using PWM

• 12V motor supply vs. 5V logic supply

Common ground

• The battery ground, Arduino ground, and L298N ground must be connected together.

4. Bluetooth Communication (HC-05)

● UART Serial Basics

Bluetooth HC-05 communicates via TX/RX pins:

• HC-05 TX → Arduino RX

• HC-05 RX → Arduino TX (with level shifting if needed)

● Baud Rate Concept

• Default HC-05 baud is 9600.

● Receiving Data

You must understand how to read Bluetooth commands:

char command = Serial.read();

● AT Commands (Optional, good to know)

• Renaming the module

• Changing PIN code

• Adjusting role (master/slave)

5. Mechanical Basics

● Chassis Assembly

• Fixing motors, battery, wheels.

● Gear Ratio / Torque (Basic Idea)

• DC motors must match the wheel size and vehicle weight.

6. Power Management

Proper power handling prevents overheating or burning components.

● Understanding separate power supplies

• Motors use battery (6–12V)

• Arduino uses 5V USB or regulator

● Why motors and Arduino must share ground

To allow signals between them to reference the same voltage.

● Battery Types

• Li-ion, Li-Po, AA battery pack.

7. Smartphone App / Control Interface

● Bluetooth Control Apps

• Knowledge of using an Android Bluetooth controller app.

• Sending characters like ‘F’, ‘B’, ‘L’, ‘R’, ‘S’.

or

• Ability to create a simple app using MIT App Inventor (optional).

Summary: What Knowledge You Need

Here is the minimum checklist:

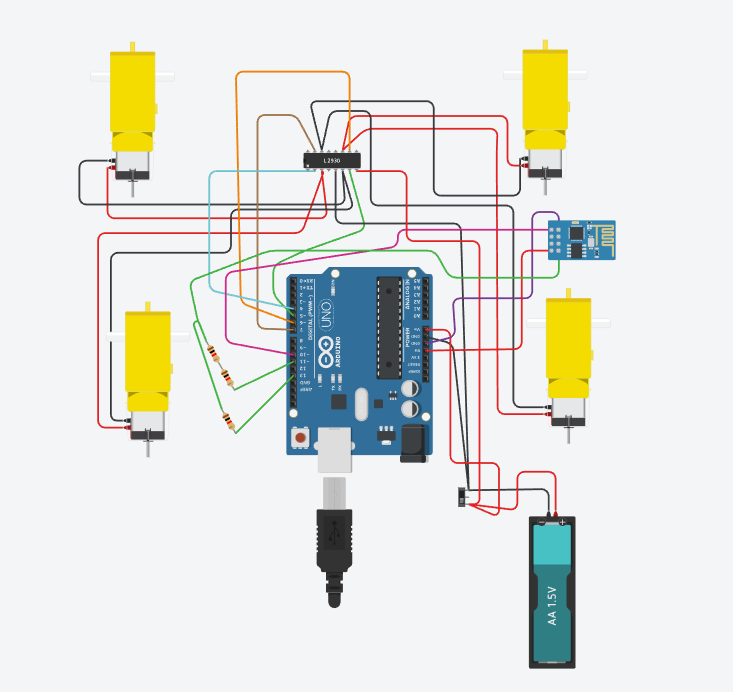
|  |  |
| --- | --- |
| Area | Required Knowledge |
| Electronics | Voltage, ground, polarity, current, basic circuits |
| Arduino C/C++ | pinMode, digitalWrite, analogWrite, Serial.read |
| Motor Drivers | H-bridge concept, L298N logic pins and power pins |
| Bluetooth | TX/RX, baud rate, pairing, serial communication |
| Power | Separate motor/logic power, common ground |
| Mechanical | Basics of assembling wheels, motors, chassis |

**Dimensions:**

The design of our car and its construction required us to keep in mind the physical dimensions. For the car we built, we used a 10x4 chassis. For the upper body, we had to make it sturdy and also keep the overall weight small enough so as not to affect the speed of the car. We therefore made use of cardboard with dimensions mentioned as under:

* Height: 8 inches
* Length: 13 inches
* Width (with tyres): 6.5 inches
* Width (without tyres): 4.5 inches

**Circuit Diagram:**

****

**Working Procedure and Circuit Connections:**

**1.Motor Connections:**  
There are four motors in total, 2 of them being on the right side and 2 on the left. Each motor possesses 2 terminals that being positive (red) and negative (black).

* **RIGHT SIDE MOTORS:**
* **Positives** of the top right and bottom right motors are connected to the OUT4 of L298N.
* **Negatives** of the above-mentioned motors are connected to OUT3.
* **LEFT SIDE MOTORS:**
* **Positives** of the top left and bottom left motors are connected to OUT1.
* **Negatives** of the above-mentioned motors are connected to OUT2.

The L298N motor driver ensures that all four motors run at the same speed and direction, by distributing equal power to each. It receives signals from the Arduino, and ensures that each wheel is turned at a constant speed and is in sync. Without the driver there would be instability in movement.

**2.Power Supply Setup:**

The L298N motor driver has two main power terminals:

* **THE +12V PORT**
* **GND (ground)**

The 12V power input comes from two lithium batteries that are connected through an ON-OFF switch. The switch controls the power supply.  
  
From the Switch:  
1. A **yellow wire** goes to the 12V terminal of the motor driver.

2. A **red wire** goes to the Arduino’s power input ‘Vin’ .

This way, both the motor driver and Arduino are powered simultaneously through a single switch. Turning the switch **ON** activates the entire circuit.

**3.Bluetooth Module Connection:**

The Bluetooth module allows wireless control of the car VIA a mobile phone. The connections are as follows:

* **5V** of the Bluetooth module is connected to the 5V port on the Arduino
* **GND** (ground) of the module is connected to the **GND** (ground) of the Arduino
* **TX** (Transmitter) of the module is connected to the 10th port

**Digital** PWM of the Arduino.

* **RX** (Receiver) of the module is connected to a voltage divider circuit.

**4.Voltage Divider Circuit for RX Pin:**

The **RX pin** of the Bluetooth module can only handle 3V, while the Arduino outputs 5V signals.

So to protect the module, a voltage divider is used consisting of two branches.

* The first branch has 2k ohm (two resistors each of 1k ohm are connected in series) is connected to the 5V port on the Arduino
* The second branch has 1k ohm and is connected to the GND (ground) of the Arduino.

This Divider reduces the 5V signal to 3V, making it safe for the Bluetooth module to function as it should.

**SUMMARY:**

1. The **switch** powers both the Arduino and motor driver from the Lithium batteries.

2. The **Arduino** sends signals (received via Bluetooth) to the **L298N driver**.

3. The **motor driver** distributes power evenly to all four motors, ensuring smooth and stable movement of the car.

4. The **Bluetooth module** allows remote control through wireless commands.

**Code:**

#include <SoftwareSerial.h>

SoftwareSerial BT(10, 11); // RX, TX

// Motor control pins

#define IN1 4

#define IN2 5

#define IN3 6

#define IN4 7

void setup() {

// Motor pins as output

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

// LED for status (optional)

pinMode(LED\_BUILTIN, OUTPUT);

Serial.begin(9600);

BT.begin(9600); // HC-05 default speed

Serial.println("Bluetooth Motor Control Ready");

BT.println("Motor Control Ready");

}

void loop() {

if (BT.available()) {

char cmd = BT.read();

Serial.print("Received: ");

Serial.println(cmd);

switch (cmd) {

case 'F': // Forward

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

BT.println("Forward");

break;

case 'B': // Backward

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

BT.println("Backward");

break;

case 'L': // Left

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

BT.println("Left");

break;

case 'R': // Right

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

BT.println("Right");

break;

case 'S': // Stop

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

BT.println("Stop");

break;

case '1': // Optional — LED ON

digitalWrite(LED\_BUILTIN, HIGH);

BT.println("LED ON");

break;

case '0': // Optional — LED OFF

digitalWrite(LED\_BUILTIN, LOW);

BT.println("LED OFF");

break;

}

}

}

**Speed Readings & Recording:**

The car was tested on a straight distance of 100 inches. This distance is equal to 2.54 meters, which is 0.00254 kilometers. The car took 6.22 seconds to travel this distance. Using the basic formula of speed, the distance was divided by the time, and after converting the value into km/h, the speed of the car came out to be 1.47 km/h.

Using the speed formula:

Speed = Distance\Time

Speed = 0.00254 km\6.22 s

Converted to km/h, the speed of the car comes out to be:

Speed ≈ 1.47 km/h

The car moves with an average speed of 1.47 km/h over the test distance.

The motion is steady, controlled, and consistent with the power supplied by the motors and the driver module. It shows that the circuit and motor driver are delivering stable performance.

**Troubleshooting:**

During the development and testing of the Bluetooth-controlled car, several issues were encountered. The causes, technical reasoning, and corrective actions are explained below.

**1. Power Supply Issue (9 V Battery vs. 2 × 3.7 V Li-ion Batteries)**

Initially, a 9 V battery was used to power the system, but the motor speed was very slow. This occurred because a standard 9 V battery has low current capacity and high internal resistance, making it unsuitable for high-current loads like DC motors. Even when the Arduino and motor driver were powered separately using the 9 V battery, significant voltage drop occurred under load, resulting in poor motor performance.

To resolve this, we replaced the 9 V battery with two 3.7 V lithium-ion batteries in series (≈7.4 V). Lithium-ion batteries can deliver higher current with lower internal resistance, leading to less power loss and more efficient energy transfer to the motors and motor driver. This provided stable voltage under load, improved torque, and smoother operation, making the system more efficient overall.

**2. Bluetooth Module Damage (HC-05 RX Pin Voltage Mismatch)**

When the Bluetooth module HC-05 was connected directly to the Arduino, the module stopped functioning. The issue was caused by a voltage level mismatch. The HC-05 RX pin operates at 3.3 V logic level, whereas the Arduino TX pin outputs 5 V. Directly applying 5 V to the RX pin can permanently damage the module.

To fix this, a potential (voltage) divider was added to the RX line of the HC-05. A 1 kΩ and 2 kΩ (1 kΩ + 1 kΩ in series) resistor network was used:

The HC-05 Bluetooth module is connected to the Arduino Uno using a voltage-divider approach on the RX line to ensure safe 5V → 3.3V level shifting. The wiring configuration is as follows:

**1. TX Connection (HC-05 → Arduino Uno)**

* The **TX pin of the HC-05** is connected **directly** to the **Arduino Uno digital pin 10 (PWM)**.
* No voltage divider is needed here because the HC-05’s TX output (3.3V) is sufficient for the Arduino’s digital input.

**2. RX Connection (Arduino Uno → HC-05)**

The **RX pin of the HC-05** is connected through a **voltage divider** made from two resistors totaling **2 kΩ** (1 kΩ + 1 kΩ).

* The **Arduino digital pin 11 (PWM)** outputs the signal.
* This signal passes through the **2 kΩ voltage divider**, and the resulting 3.3V logic level is sent to the **HC-05 RX pin**.

**3. Voltage Divider Ground Path**

* The lower side of the voltage divider includes a **1 kΩ resistor** branch that connects to **Arduino digital pin 13 (PWM)**, which is configured as ground or low output for the circuit.
* This ensures the voltage divider has a stable reference to ground.

This divider drops the voltage from 5 V to approximately 3.3 V, which is safe for the HC-05. Using the potential divider protects the Bluetooth module and ensures reliable communication.

**3. Motors Rotating in Opposite Directions (Polarity Issue)**

At one stage, the wheels were rotating in opposite directions (one clockwise and the other anticlockwise), even though the same control signals were applied. This happened because the polarity of the motor connections to the L298N motor driver was reversed for one motor.

Polarity is important because DC motors rotate in a direction determined by the orientation of the positive and negative terminals. If the polarity is reversed, the motor rotates in the opposite direction.

Two possible ways to fix this issue are:

* Swap the positive and negative wires of the motor at the L298N output terminals.
* Change the motor control logic in software by reversing the control signals.

The preferred solution is swapping the motor wires, as it keeps the software logic simple and avoids unnecessary changes in code.

**4. Continuous Power Drain (No ON/OFF Switch)**

Initially, the circuit had no power switch, so the system turned ON immediately when the batteries were connected. This caused unnecessary battery drain even when the car was not in use.

To solve this, a power switch was added between the battery and the main power line. This allows manual control over the system’s power, prevents unwanted battery discharge, and improves battery life and safety.

These troubleshooting steps significantly improved system performance, reliability, and efficiency, leading to proper operation of the Bluetooth-controlled car.

**Conclusion and Comments:**

This project successfully demonstrates how Bluetooth communication, motor control, and microcontroller programming can be combined to create a functional robotic car. The integration of the Arduino Uno, HC-05 module, and L298N driver ensures smooth wireless operation and efficient motor handling. The system is simple to build, cost-effective, and easily expandable for future upgrades. Overall, it provides a solid foundation for understanding embedded systems, automation, and remote-controlled robotics.