**Artificial Intelligence for Robotics**

**(CSL 349)**

**Arduino Bluetooth Controlled Car Project**

**Project Report**

A logo of a university

Description automatically generated

Faculty name: **Dr. Shraddha Arora** Student name:

**Piyush Gambhir (21CSU349),**

**Himansh (21CSU324),**

**Ankit (21CSU277),**

**Riten (21CSU318)**

Semester: **VI**

Group: **A3 (AIML B)**

**Project GitHub Repository:** [**ncu-lab-manual-and-end-semester-projects/NCU-CSL349 - AIR - End Semester Project at main · piyush-gambhir/ncu-lab-manual-and-end-semester-projects (github.com)**](https://github.com/piyush-gambhir/ncu-lab-manual-and-end-semester-projects/tree/main/NCU-CSL349%20-%20AIR%20-%20End%20Semester%20Project)

**Department of Computer Science and Engineering**

**NorthCap University, Gurugram- 122001, India**

**Session 2022-23**

**Introduction:**

The Arduino Bluetooth Controlled Car project is designed to develop a wireless controlled car using Arduino as the main processor interfaced with a Bluetooth module. The aim of this project is to understand and implement the principles of microcontroller programming, wireless communication, and robotics fundamentals.

**Objectives:**

* To build a Bluetooth-controlled robotic car using Arduino.
* To demonstrate the capability of wireless control through a Bluetooth interface.
* To apply knowledge of electronics and programming to create a functional robotic vehicle.

**Materials and Methods:**

1. **Components Used:**

* Arduino Uno: Serves as the central controller for all vehicle functions.
* HC-05 Bluetooth Module: Enables remote control via Bluetooth.
* HC-SR04 Ultrasonic Sensor: Provides obstacle detection capabilities.
* L293D Motor Driver Module: Manages motor operations.
* Servo Motor SG90: Allows for rotational movement of the ultrasonic sensor to scan a wider area.
* Motors and Wheels: Provide the locomotion necessary for the vehicle.
* Battery Pack: Powers the vehicle, ensuring all components function seamlessly.

1. **Assembly:**
   * Assembled the car chassis with motors and wheels.
   * Connected the motors to the L298N motor driver.
   * Interfaced the motor driver with Arduino Uno.
   * Integrated the HC-05 Bluetooth module with Arduino for wireless control.
2. **Programming:**
   * Developed the code in Arduino IDE to control the car via Bluetooth.
   * Implemented functions for forward, reverse, left, and right movements based on Bluetooth commands.
   * Uploaded the final code to the Arduino board.
3. **Testing:**
   * Tested the car for basic movements to ensure proper wiring and code functionality.
   * Conducted field tests to evaluate the performance, range, and stability of Bluetooth communication.

**Detailed Component Description**

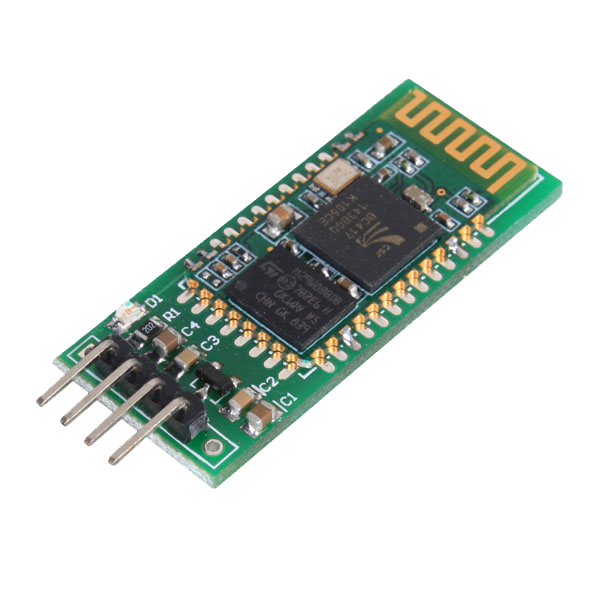
**Arduino Uno**

* **Function:** Acts as the central controller for the vehicle.
* **Description:** The Arduino Uno is a microcontroller board based on the ATmega328P chip. It features 14 digital input/output pins, 6 of which can be used as PWM outputs, and 6 analog inputs. It also includes a 16 MHz quartz crystal, USB connection, power jack, ICSP header, and a reset button. It is used to process inputs from the sensors and control outputs to various components like motors and servos, executing the logic defined in the uploaded program.



**HC-05 Bluetooth Module**

* **Function:** Enables wireless control over Bluetooth.
* **Description:** The HC-05 module is a versatile, easy-to-use Bluetooth transceiver that operates in master or slave mode. It allows the Arduino car to connect with any Bluetooth-enabled device, such as smartphones, enabling users to send control commands to the car. It supports serial communication, which makes it ideal for embedded applications requiring remote control.



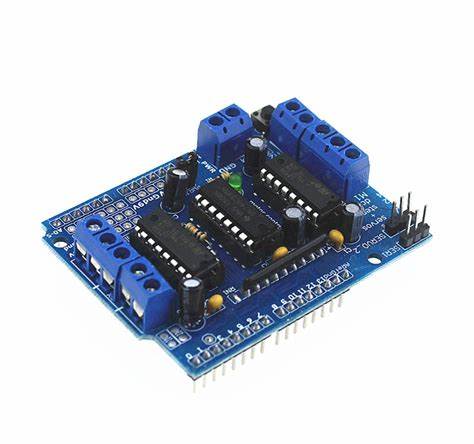
**HC-SR04 Ultrasonic Sensor**

* **Function:** Detects obstacles in the car's path.
* **Description:** This sensor measures distance by emitting ultrasonic waves and receiving the echoes back. The HC-SR04 calculates the distance to an object by timing the interval between sending the signal and receiving the echo. This feature is crucial for the car’s obstacle avoidance capability, allowing it to sense and navigate around obstacles.



**L293D Motor Driver Module**

* **Function:** Manages the operation of the motors.
* **Description:** The L293D is a dual H-bridge motor driver IC that lets the Arduino control two DC motors simultaneously in both clockwise and counterclockwise directions. It can handle up to 600 mA per channel with a voltage range between 4.5V and 36V, making it suitable for driving small to medium robotics motors.



**Servo Motor SG90**

* **Function:** Rotates the ultrasonic sensor for broader detection.
* **Description:** The SG90 is a small and lightweight servo motor capable of precise position control. It is commonly used in robotics for controlling steering mechanisms and sensor orientation. In this project, it adjusts the angle of the ultrasonic sensor, allowing the car to scan a wider area for obstacles, thus enhancing its navigation capabilities.



**Motors and Wheels**

* **Function:** Provide the necessary movement to the car.
* **Description:** Typically, geared DC motors are used in such projects for their simplicity and effectiveness in providing adequate torque and speed. The wheels are mounted directly on the motor shafts, transferring the motor's rotational motion into linear motion to propel the car.

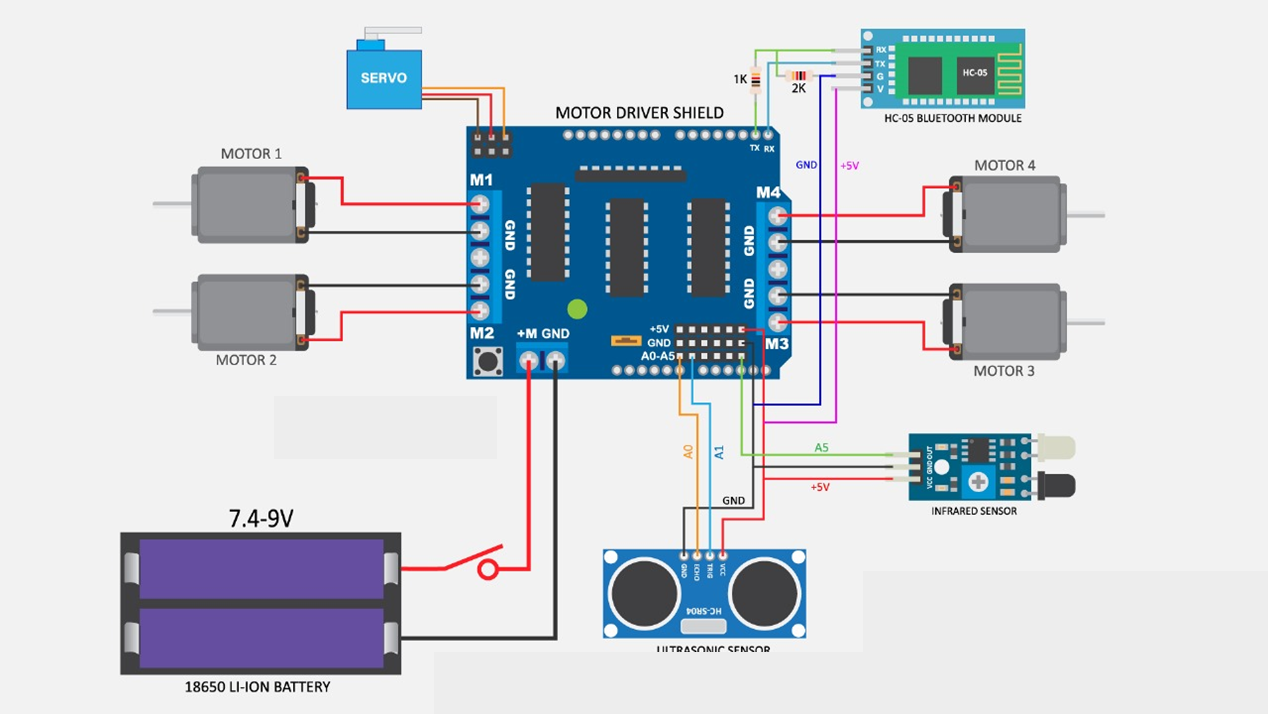


**Battery Pack**

* **Function:** Supplies power to the entire setup.
* **Description:** This consists of either standard AA batteries or a rechargeable LiPo battery pack. It needs to supply adequate voltage and current to power the Arduino, motors, sensors, and other components. The choice of battery impacts the car's operational time, weight, and overall performance.



**Circuit Diagram**

****

**A circuit board with wires and wires

Description automatically generated**

**Code**

#include <Servo.h>

#include <AFMotor.h>

#include <SoftwareSerial.h> // Required for voice control

#include <Arduino.h>

#include <Wire.h>

#define Echo A0

#define Trig A1

#define motor 10

#define Speed 170

#define spoint 103

char value;

int distance;

int Left;

int Right;

int L = 0;

int R = 0;

bool bypassObstacleAvoidance = false; // Flag to bypass obstacle avoidance

Servo servo;

AF\_DCMotor M1(1);

AF\_DCMotor M2(2);

AF\_DCMotor M3(3);

AF\_DCMotor M4(4);

SoftwareSerial voiceSerial(2, 3); // RX, TX for voice control

void setup() {

  Serial.begin(9600);

  voiceSerial.begin(9600); // Initialize voice serial communication

  pinMode(Trig, OUTPUT);

  pinMode(Echo, INPUT);

  servo.attach(motor);

  M1.setSpeed(Speed);

  M2.setSpeed(Speed);

  M3.setSpeed(Speed);

  M4.setSpeed(Speed);

}

void loop() {

  Obstacle();

  Bluetoothcontrol();

  Voicecontrol();

}

void Obstacle() {

  if (!bypassObstacleAvoidance) {

    distance = ultrasonic();

    if (distance <= 12) { // Use ultrasonic sensor for obstacles within 12 cm

      Stop();

      backward();

      delay(500);

      Stop();

      L = leftsee();

      servo.write(spoint);

      R = rightsee();

      servo.write(spoint);

      if (L < R) {

        right();

        delay(500);

        Stop();

      } else if (L > R) {

        left();

        delay(500);

        Stop();

      }

    } else {

      forward(); // If no obstacle is detected, move forward

    }

  } else {

    // Obstacle avoidance bypassed

  }

}

int ultrasonic() {

  digitalWrite(Trig, LOW);

  delayMicroseconds(3);

  digitalWrite(Trig, HIGH);

  delayMicroseconds(5);

  digitalWrite(Trig, LOW);

  long t = pulseIn(Echo, HIGH);

  long cm = t / 29 / 2; //time convert distance

  return cm;

}

void forward() {

  M1.run(FORWARD);

  M2.run(FORWARD);

  M3.run(FORWARD);

  M4.run(FORWARD);

}

void backward() {

  M1.run(BACKWARD);

  M2.run(BACKWARD);

  M3.run(BACKWARD);

  M4.run(BACKWARD);

}

void right() {

  M1.run(BACKWARD);

  M2.run(BACKWARD);

  M3.run(FORWARD);

  M4.run(FORWARD);

}

void left() {

  M1.run(FORWARD);

  M2.run(FORWARD);

  M3.run(BACKWARD);

  M4.run(BACKWARD);

}

void Stop() {

  M1.run(RELEASE);

  M2.run(RELEASE);

  M3.run(RELEASE);

  M4.run(RELEASE);

}

int rightsee() {

  servo.write(60);

  delay(1000);

  Left = ultrasonic();

  return Left;

}

int leftsee() {

  servo.write(180);

  delay(1000);

  Right = ultrasonic();

  return Right;

}

void Bluetoothcontrol() {

  if (Serial.available() > 0) {

    value = Serial.read();

    Serial.println(value);

    if (value == 'U') {

      forward();

      bypassObstacleAvoidance = true; // Bypass obstacle avoidance when any command is received

    } else if (value == 'D') {

      backward();

      bypassObstacleAvoidance = true;

    } else if (value == 'L') {

      left();

      bypassObstacleAvoidance = true;

    } else if (value == 'R') {

      right();

      bypassObstacleAvoidance = true;

    } else if (value == 'S') {

      Stop();

      bypassObstacleAvoidance = true;

    } else {

      bypassObstacleAvoidance = false; // Resume obstacle avoidance if no command is received

    }

  }

}

void Voicecontrol() {

  if (voiceSerial.available() > 0) {

    value = voiceSerial.read();

    Serial.println(value);

    if (value == 'F') {

      forward();

      bypassObstacleAvoidance = true; // Bypass obstacle avoidance when any voice command is received

    } else if (value == 'B') {

      backward();

      bypassObstacleAvoidance = true;

    } else if (value == 'L') {

      left();

      bypassObstacleAvoidance = true;

    } else if (value == 'R') {

      right();

      bypassObstacleAvoidance = true;

    } else if (value == 'S') {

      Stop();

      bypassObstacleAvoidance = true;

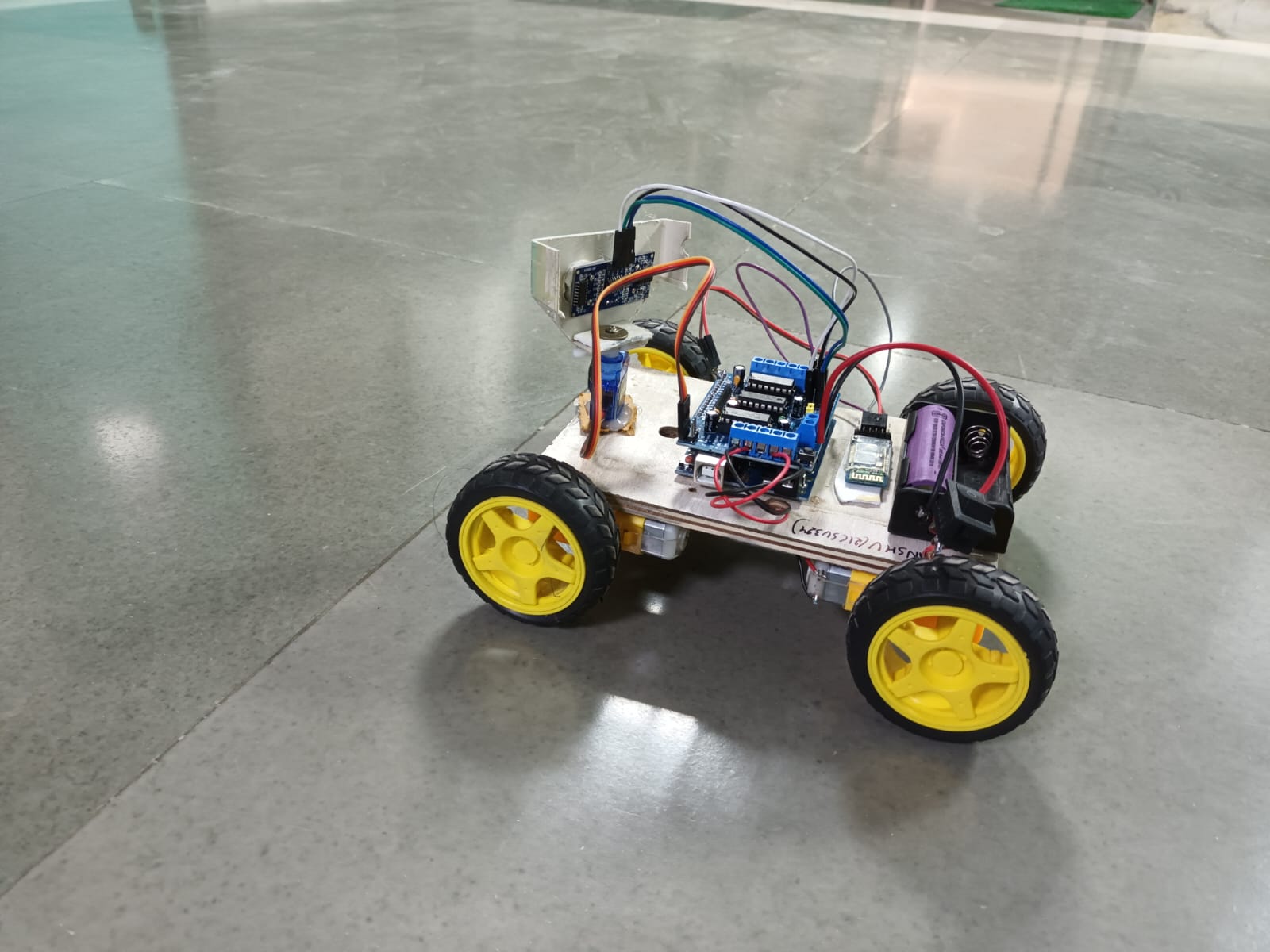
    } else {

      bypassObstacleAvoidance = false; // Resume obstacle avoidance if no voice command is received

    }

  }

}

**Project Image**

**Results:**

The project successfully resulted in a fully functional Bluetooth-controlled car. The vehicle responded accurately to the commands sent from a Bluetooth-enabled smartphone using a dedicated app. Range tests showed effective control at distances up to 10 meters.

**Challenges:**

* Ensuring stable and consistent communication between the Bluetooth module and the smartphone.
* Power management for sustained operation during extended testing periods.
* Interference issues in areas with multiple wireless signals.

**Conclusion:**

The Arduino Bluetooth controlled car project was successful in demonstrating the integration of various electronic components and software to achieve a wirelessly controlled robotic vehicle. This project serves as a practical example of applying theoretical knowledge in electronics, programming, and wireless communication in a real-world application.

**Future Work:**

* Implementing additional features such as obstacle avoidance using ultrasonic sensors.
* Enhancing the control app with more functionalities like speed control and automated movements.
* Exploring alternative power sources to increase operational time and efficiency.

**References:**

* Arduino.cc
* HC-05 Bluetooth module datasheet
* L298N Motor Driver datasheet

**Appendices:**

* Arduino Code
* Schematics of the wiring
* List of suppliers for components