VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering and Information Systems

Winter Semester 2024-25

COURSE CODE: SWE1901.

COURSE TITLE: TARP

FACULTY: VIJAY ANAND.R

SLOT: TAA1.

Title of the Project:

Arduino controlled Bluetooth, Voice and obstacle avoidance Car

Tharun S

21MIS0408

Abstract

The Arduino Bluetooth-controlled car is an innovative project that demonstrates the integration of embedded systems and wireless communication technologies to develop a cost-effective, remote-controlled vehicle. This project utilizes an Arduino microcontroller, an HC-05 Bluetooth module, an L298N motor driver, and a smartphone for control. The car's movement—forward, backward, left, right, and stop—is directed through simple commands sent via a Bluetooth connection. This project highlights the principles of automation, wireless communication, and robotics, making it an ideal solution for applications in surveillance, smart transportation systems, and educational robotics. The design emphasizes simplicity, affordability, and scalability, allowing future enhancements like obstacle detection and autonomous navigation.

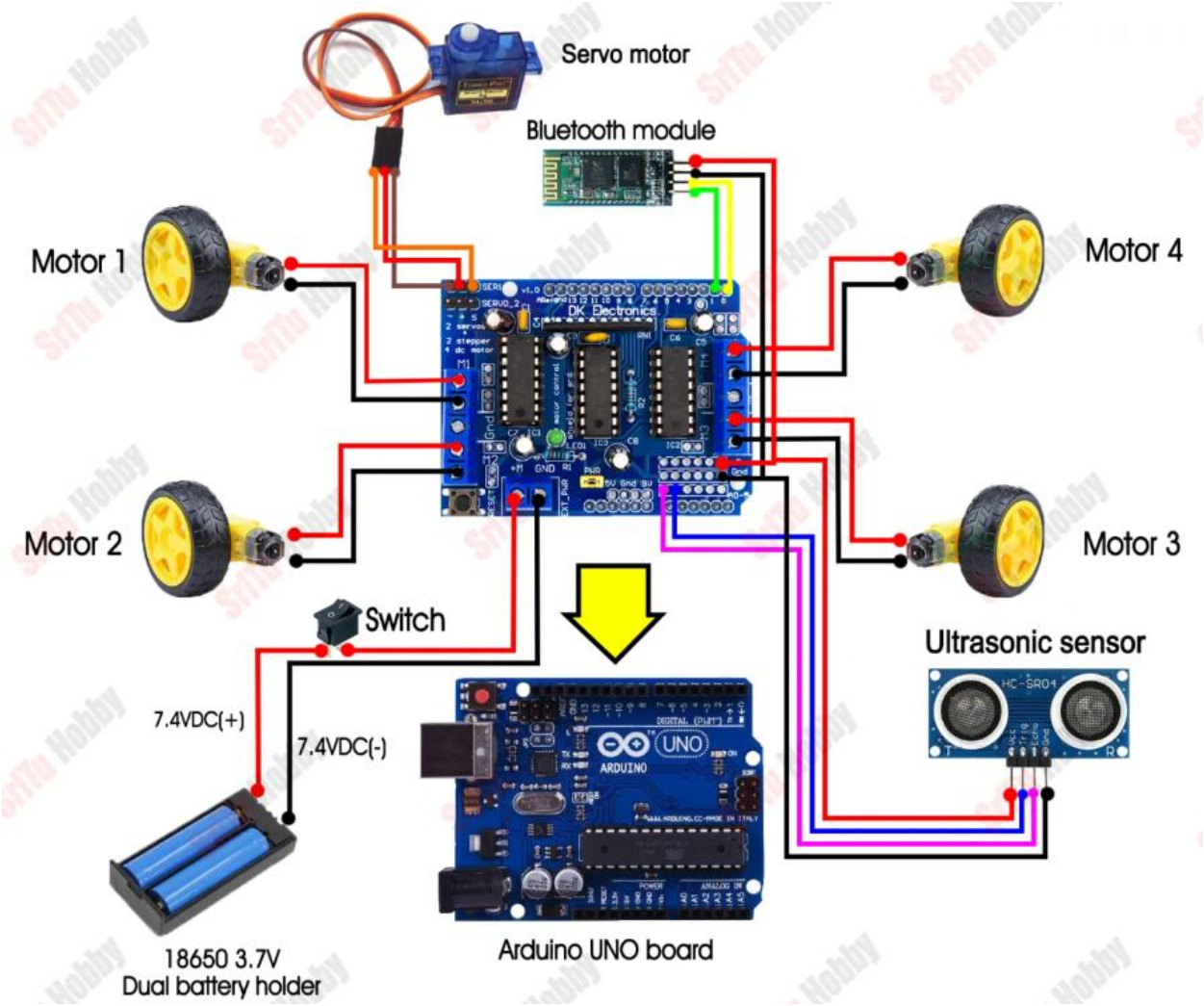
Problem Description

Traditional remote-controlled vehicles often rely on radio frequency (RF) modules with limited range, poor signal stability, and lack of flexibility. Moreover, RF controllers require dedicated hardware, which adds to the cost and complexity. In contrast, modern devices like smartphones are equipped with Bluetooth capabilities that can provide a more flexible and user-friendly control interface. The challenge is to develop a robust, low-cost, Bluetooth-controlled car that can be operated easily using a smartphone. This system should ensure reliable communication, smooth motor control, and scalability for additional features like automation or obstacle avoidance.

Language Used

The given code is written in C++, specifically designed for Arduino programming. It utilizes the AF Motor library to control DC motors and the Servo library to control a servo motor. The program is intended for an autonomous or Bluetooth-controlled robotic vehicle with obstacle detection and voice control functionality.

Circuit Diagram



Methodology

1. Hardware Components

- Arduino Board (Compatible with AFMotor Library)
- Ultrasonic Sensor (HC-SR04 for distance measurement)
- Servo Motor (For scanning left and right)
- L298N motor driver
- 4 DC Motors (Connected to an Adafruit Motor Shield for movement)

2. Setup Process (setup() function)

- Initializes the serial communication (Serial.begin(9600)) for Bluetooth or voice commands.
- Defines the trigger (Trig) and echo (Echo) pins for the ultrasonic sensor.
- Attaches the servo motor to pin 10.
- Sets the speed of the 4 DC motors.

3. Control Mechanisms

The robot can be controlled in three ways:

- Bluetooth Control (Bluetoothcontrol())
- Obstacle Avoidance (Obstacle())
- Voice Control (voicecontrol())

3.1. Bluetooth Control (Bluetoothcontrol())

Reads the incoming Bluetooth command from a mobile app.

Based on the received character, it calls the corresponding movement function:

'F' → Move Forward

'B' → Move Backward

'L' → Turn Left

'R' → Turn Right

'S' → Stop the Robot

3.2. Obstacle Avoidance (Obstacle())

Uses an ultrasonic sensor to detect objects ahead.

If an obstacle is within 12 cm:

The robot stops and moves backward.

It then checks left (leftsee()) and right (rightsee()) distances.

Moves towards the direction with more space.

If no obstacle is detected, the robot moves forward.

3.3. Voice Control (voicecontrol())

Listens for voice commands:

'^' → Move Forward

'-' → Move Backward

'<' → Turn Left (Only if the left side is clear)

'>' → Turn Right (Only if the right side is clear)

'*' → Stop the Robot

4. Movement Functions

These functions control the 4 DC motors for movement:

- forward(): Moves all motors forward.
- backward(): Moves all motors backward.
- left(): Turns left by reversing the right motors.
- right(): Turns right by reversing the left motors.
- Stop(): Stops all motors.

5. Distance Measurement (ultrasonic())

Uses the HC-SR04 Ultrasonic Sensor:

Sends a trigger pulse.

Measures the time taken for the echo to return.

Converts time into distance in centimeters.

6. Servo Motor Control

- leftsee(): Rotates the servo left (180°) and measures distance.
- rightsee(): Rotates the servo right (20°) and measures distance.
- The robot decides its direction based on these readings.

Summary of Methodology

Initialization of components (Serial, Motors, Servo, Ultrasonic).

Listening for commands (Bluetooth, Voice, or Autonomous).

Decision Making:

If Bluetooth/Voice command received, follow movement instructions.

If obstacle detected, measure left and right distances and choose the best path.

Execution of movement functions using DC motors.

Continuous Monitoring for new inputs.

Pseudo Code:

Main System Overview

START

Initialize:

-

and servo motor

- Ultrasonic sensor pins

LOOP:

// Choose ONE mode at a time by calling its function:

- Obstacle Avoidance Mode
- Bluetooth Control Mode
- Voice Control Mode

END LOOP

Obstacle Avoidance Mode

FUNCTION ObstacleAvoidance:

Measure distance using ultrasonic sensor

IF distance is too close:

Stop

Move backward briefly

Measure distances to left and right by rotating servo

IF right has more space:

Turn right

ELSE:

Turn left

ELSE:

Move forward

Bluetooth Control Mode

FUNCTION BluetoothControl:

IF a Bluetooth command is received:

Read command character

IF command == 'F': Move Forward

IF command == 'B': Move Backward

IF command == 'L': Turn Left

IF command == 'R': Turn Right

IF command == 'S': Stop

Voice Control Mode

FUNCTION VoiceControl:

IF a voice command is received:

Read command character

IF command == '^': Move Forward

IF command == '-': Move Backward

IF command == '<':

Check if left side is free

IF free: Turn Left

ELSE: Stop

IF command == '>':

Check if right side is free

IF free: Turn Right

ELSE: Stop

IF command == '*': Stop

Supporting Functions

FUNCTION ultrasonic():

Trigger the sensor

Measure echo duration

Convert duration to distance

RETURN distance

FUNCTION forward(): All motors move forward

FUNCTION backward(): All motors move backward

FUNCTION left(): Left motors forward, Right motors backward

FUNCTION right(): Left motors backward, Right motors forward

FUNCTION stop(): All motors stop

FUNCTION leftsee():

Rotate servo to left

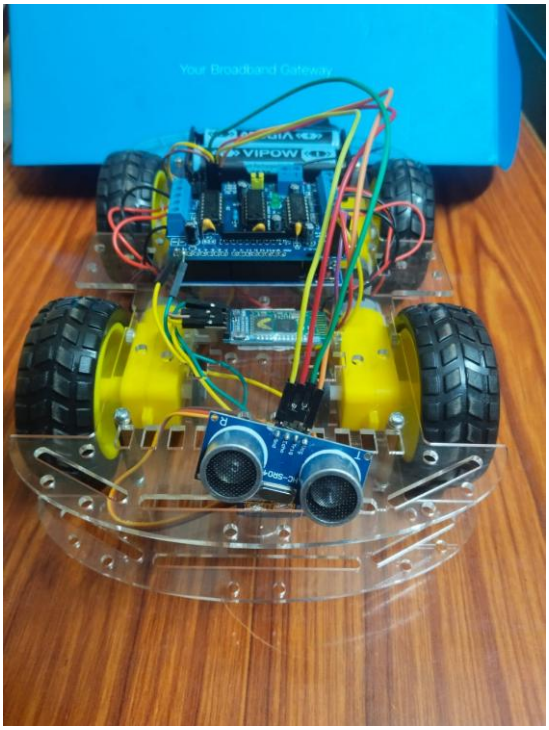
RETURN distance

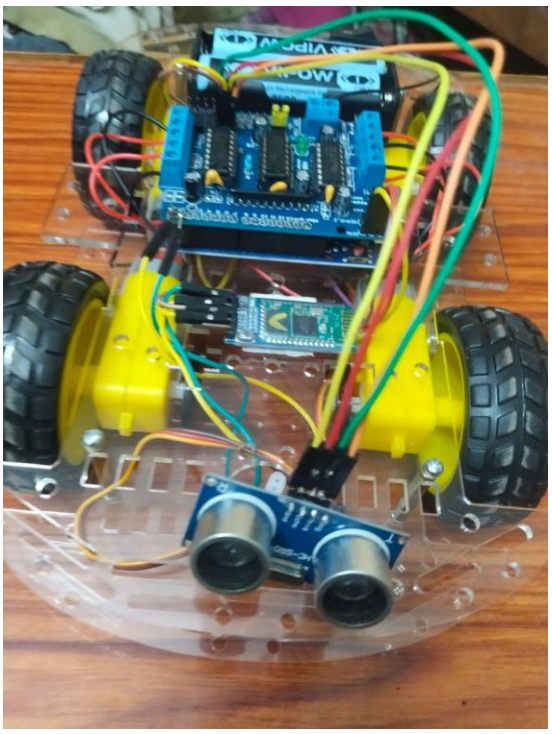
FUNCTION rightsee():

Rotate servo to right

RETURN distance

Model





Video Link

https://drive.google.com/drive/folders/1nvTIyDmjMur5Hy0EiYVkcgbFFBoRiKRR

Conclusion

The Arduino Bluetooth-controlled car effectively showcases the synergy between embedded systems and wireless communication to create a functional, low-cost, and user-friendly robotic solution. By replacing traditional RF-based control with a Bluetooth-enabled smartphone interface, the project enhances both usability and range while reducing hardware complexity. The use of easily accessible components like the Arduino, HC-05 Bluetooth module, and L298N motor driver makes this system not only affordable but also highly scalable. This project lays the groundwork for more advanced applications such as autonomous navigation and obstacle avoidance, making it a valuable prototype for educational purposes, smart mobility solutions, and surveillance systems. Overall, it exemplifies how modern technologies can be harnessed to develop efficient and innovative solutions in the field of robotics.