Bluetooth-Controlled Obstacle Avoider/Line Follower

Abdul Moiz Ghumman
FCSE, GIK Institute
Topi, Khyber Pakhtunkhwa, Pakistan
u2021015@giki.edu.pk

Muhammad Sibtain Haider
FCSE, GIK Institute
Topi, Khyber Pakhtunkhwa, Pakistan
u2021459@giki.edu.pk

***Abstract*—This report details the collaborative development of a Bluetooth-controlled robot that has dual functionality, i.e., obstacle avoidance and line following. The project uses Arduino Uno as a micro-controller and combines hardware components, such as sensors and motors, with Bluetooth communication for remote control.**

I. Introduction

This project had a focus on making a robot car with dual functionality. The functionality then could be controlled by a Bluetooth device connected to the micro-controller. The versatile robot can navigate its environment by avoiding obstacles or following predefined paths. It is important to note that certain complexities, such as highly dynamic environments, may pose challenges beyond the scope of this work.

II. Literature Review

The development of this robot involved integrating various components and technologies. The following sources provided valuable insights and guidance:

1) Motor Driver Module (L298N): The motor control module, L298N, was integrated into the project for controlling the DC motors that are connected to the tyres. The information is based on the documentation from Components101. [1].
2) Ultrasonic Sensor (HC-SR04): The implementation of obstacle detection utilized the complete guide for the HC-SR04 ultrasonic sensor [2].
3) IR Sensor for Line Following: The line-following capability was achieved with the help of the tutorial on creating a line follower robot using Arduino and IR sensors [3].
4) Bluetooth Module (HC-05): Bluetooth communication was established using the HC-05 Bluetooth module with guidance from the comprehensive tutorial [4].

III. Methodology

A. Hardware Components

The robot's construction involved a selection of hardware components, these were provided by the lab instructor. The Arduino Uno micro-controller served as the central processing unit, orchestrating the robot's actions. The L298n motor driver was there to control the DC motors. Essential sensors, including the HC-SR04 for obstacle detection and IR sensor for line following.The incorporation of the HC-05 Bluetooth module enabled seamless remote control.

TABLE I
Hardware Components Used in the Project

Components	Description	Quantity
Arduino Uno	Micro-controller	1
Sensor Shield V5	Arduino Shield	1
L298n	Motor control module	1
HC-SR04	Obstacle detection	1
Servo Motor	Rotate HC-SR04	1
IR Sensor	Line-following Sensor	2
HC-05	Bluetooth module	1
DC 3-6V Dual Motors	Robot movement	2
2WD robot chassis kit	Robot body	1
3.6V Lithium-Ion Cells	Battery	4

B. Circuit Details

1) L298N Connection: This image shows the connections done between L298n driver and the motors and the Arduino. The Arduino is being connected using the sensor shield which has been placed on top of it.

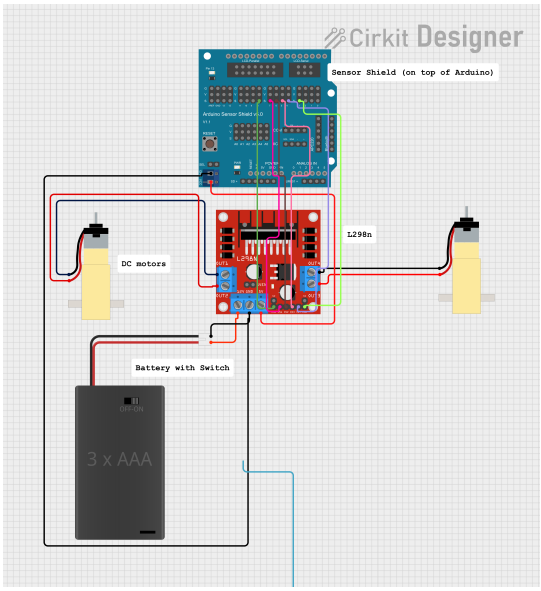

Fig. 1. L298N Motor Driver Connection.

2) Sensors and Modules: The following illustration shows how different sensors and modules are connected to the microcontroller.

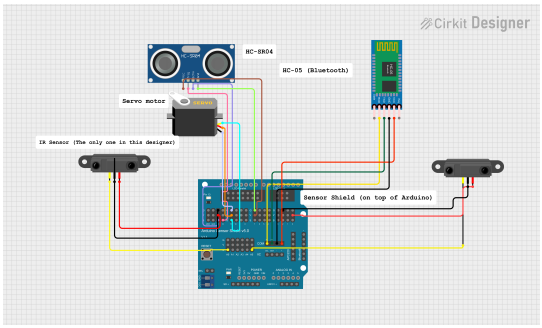

Fig. 2. Sensors and Module Connection.

3) Full Configuration: This illustartion shows the entire configuration connected together to bring the car into life.

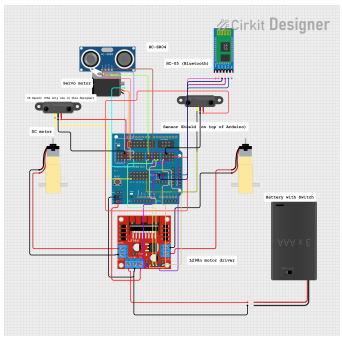

Fig. 3. Full Configuration of the Robot.

C. Robot Configuration

1) L298N Connection: The L298n is the motor driver that acts as an interface for the DC motors. The two wires from each motor are connected to the DC motor A output and DC motor B output. The motor is then commanded from here. There are 6 motor direction control inputs that are connected to Arduino Uno.

```
// In arduino code
enAPin = 6; //MotorA speed Control
in1Pin = 7; //motorA Direction 1
in2Pin = 4; //motorA Direction 2
in3Pin = 2; //motorB Direction 1
in4Pin = 3; //motorB Direction 2
int enBPin = 5; //MotorB speed control
```

To make the motors go forward or in reverse we give two inputs to the motors directional controls. Generally we use '1 0' to make the motors go forward and '0 1' to make them go backwards. We only had a 2WD chasis so we made the choice to implements sensors behind the wheels as illustrated by Figure 4. So the functions are written with this in mind. For example to make the car go forward our moveForward() function would have inputs '0 1' '0 1'.

```
analogWrite(enAPin, 150);
analogWrite(enBPin, 150);
digitalWrite(in1Pin, LOW);
digitalWrite(in2Pin, HIGH);
digitalWrite(in3Pin, LOW);
digitalWrite(in4Pin,HIGH);
```

TABLE II
ROBOT MOVEMENT COMMANDS AND PIN INPUTS

Command	Pin Inputs
Move Forward	in1: 0, in2: 1, in3: 0, in4: 1
Turn Right	in1: 0, in2: 1, in3: 1, in4: 0
Turn Left	in1: 1, in2: 0, in3: 0, in4: 1
Stop Motors	in1: 0, in2: 0, in3: 0, in4: 0
Move Reverse	in1: 1, in2: 0, in3: 1, in4: 0

2) HC-SR04 and Servo motor: The HC-SR04 serves as the obstacle detector of the robot. It has 4 pins, 2 are for VCC and ground while the remaining 2 are echo and trig pins. These are used for calculating distance between object It is attached to servo motor which is itself connected to Arduino. It has 1 signal pin and 1 VCC as well as a ground pin. We use the library Servo.h for controlling the servo motor.

```
// Define obstacle detection parameters
obstacleThreshold =20; //Distance in cm

// Rotate the servo to scan the area
for (int angle = 0; angle <= 180; angle += 45) {
    ultrasonicServo.write(angle);
    delay(500);
    int distance = getDistance();

    if (distance < obstacleThreshold) {
      // If detected, stop, reverse, and turn
      avoidObstacle();
      break; // Exit the loop
    }
  }
```

3) IR sensors: Two IR sensors have been used for this robot. They are used in line following where they help the robot keep on track. The IR sensors are connected using analog pins. We set a threshold value so to let the sensors know when to turn left or right.

```
 // Read sensor values
 leftSensorValue = analogRead(leftPin);
 rightSensorValue = analogRead(rightPin);

  if (leftSensorValue > threshold &&
  rightSensorValue > threshold) {
  // Both sensors on the line go forward
    moveForward();
  } else if (leftSensorValue > threshold) {
// Left sensor is on the line turn right
    turnRight();
```

```
  } else if (rightSensorValue > threshold) {
 //Right sensor is on the line  turn left
    turnLeft();
  } else {
 // Both sensors are off the line - stop
    stopMotors();
```

4) HC-05: Four pins of the Bluetooth module are utilised. Two of them being VCC and Ground while other two are TX (Transmitting Pin) and RX (Receiving Pin). These pins have specific location on Arduino Uno as well as the sensor shield. The TX of Arduino is connected with RX of HC-05 and the RX of Arduino is connected with TX of HC-05. The code uses serial input received by the Bluetooth module to decide which mode to operate in [5].

```
\\ keep getting input from android
  while (Serial.available() > 0) {
    val = Serial.read();
    Serial.println(val);
  }
```

TABLE III
ARDUINO BLUECONTROL INPUT MODES

Input Value	Mode
1	Obstacle Avoidance
2	Line Following
3	Turn Motors Off

IV. RESULTS

The robot successfully avoids obstacles when asked to do so using Bluetooth input. The robot must be manually placed on a line before enabling the mode. The IR sensors also needed configuration to detect line and not go off it. Figures **??** and 5 illustrate the robot in action.

V. DISCUSSION

During implementation we faced one major problem and that was related to the batteries. As the kit provided by the lab instructor only consisted of a 4 AA battery holder which was insufficient to power on the motors for too long. This is because of the low current provided by AA batteries, from an online forum I discovered that L298N, the motor driver we were given was outdated and required a lot of voltage to keep running. So we had to use Lithium Ion batteries to keep it running.

VI. CONCLUSION

This project was a success ,it allowed us to combine Bluetooth controls with obstacle avoidance and line following, showcasing the potential for versatile and remotely controlled robotic systems.

Fig. 4. Obstacle Avoiding Mode

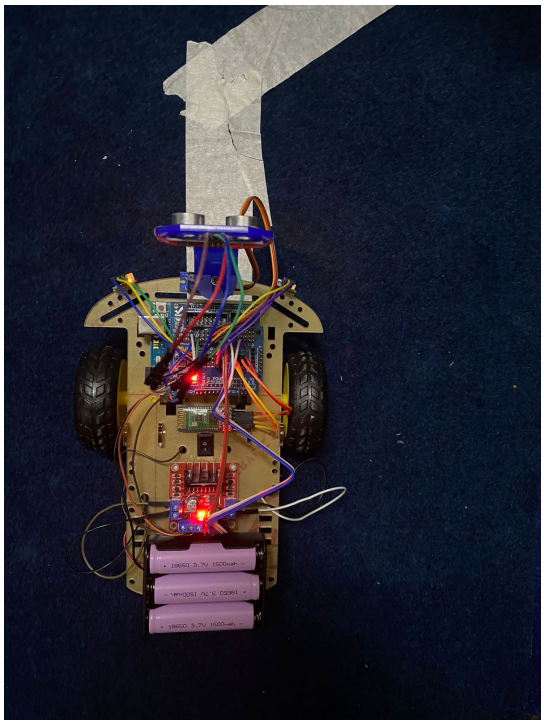

Fig. 5. Line follower Mode

ACKNOWLEDGMENTS

We would like to acknowledge GIK Institute and Faculty of Computer Science and Engineering for providing us with all the tools necessary to carry out this project.

REFERENCES

[1] L293N Motor Driver Module https://components101.com/modules/l293n-motor-driver-module.

[2] Complete Guide for Ultrasonic Sensor HC-SR04. https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/
[3] Line Follower Robot using Arduino. https://rootsaid.com/line-follower-robot-using-arduino/.
[4] Bluetooth Control using Arduino. https://www.makerguides.com/arduino-and-hc-05-bluetooth-module-complete-tutorial/.
[5] Arduino BlueControl. https://play.google.com/store/apps/details?id=com.broxcode.arduinobluetoothfreepli=1