PATUAKHALI SCINECE AND TECHNOLOGY UNIVERSITY

SUBJECT CODE: EEE-212

ELECTRICAL TECHNOLOGY SESSIONAL

PROJECT REPORT

Autonomous Line-Following Robot Car Using IR Sensors and Arduino

SUBMITTED TO:

Md. Naimur Rahman

Professor Department of Electrical and Electronics Engineering

Faculty of Computer Science and Engineering

SUBMITTED BY:

Jihadul Islam

ID: 2102071

Nazmus Sakib

ID: 2102044

Faculty of Computer Science and Engineering

Date of Submission: 29-10-2024

Objective:

The primary objective of this project is to design and construct an autonomous line-following car that can detect and follow a defined path. Utilizing infrared (IR) sensors and a motor driver, the car will move along a line, adjusting its direction as needed to stay on course. This project demonstrates the integration of sensor data and motor control in robotics, achieving a system that can perform basic navigation autonomously.

Necessary Components:

- **Arduino Board** (e.g., Arduino Uno): The microcontroller responsible for processing sensor data and controlling the motors.

- **IR Sensors** (2 units): These detect the presence of a line and provide input to the Arduino for navigation.
- **L298N Motor Driver**: Interfaces between the Arduino and the motors, allowing controlled forward and turning movements.
- **DC Motors** (2 units): Drive the wheels of the car and are controlled via the motor driver.
- **Battery Pack**: Provides the power supply for the motors and the Arduino.
- **Chassis**: Holds all components securely, forming the car structure.
- **Connecting Wires**: Facilitate connections between components.components.

Experimental Procedure:

1 • **Component Connections**:

- Start by connecting the **IR sensors** to the Arduino's digital input pins (e.g., pin 2 for the left sensor and pin 3 for the right sensor).
- Connect the **L298N motor driver** to the Arduino. Attach pins IN1 and IN2 on the motor driver to digital pins 4 and 5 on the Arduino to control the left motor, and pins IN3 and IN4 to pins 6 and 7 for the right motor.
- Connect the ENA and ENB pins of the motor driver to PWM pins 9 and 10 on the Arduino to control motor speed.
- Attach the **DC motors** to the OUT1, OUT2, OUT3, and OUT4 terminals on the motor driver.

  - Use a **battery pack** to power the motor driver, while powering the Arduino separately through USB or a dedicated power source.

- **Upload Code to Arduino**:

  - Open the Arduino IDE on your computer and connect the Arduino board.
  - Write or paste the Arduino code designed for line following, which will read IR sensor values and control the motors accordingly.
  - Upload the code to the Arduino and observe the output in the Serial Monitor if needed, to confirm that the IR sensors are correctly detecting the line.

- **Testing and Calibration**:

  - Place the robot car on a flat surface with a clearly defined line, preferably black tape on a light-colored background.
  - Observe the robot's movement as it follows the line. If the car does not respond correctly to the line, check sensor readings in the Serial Monitor and adjust the sensor height or alignment as needed.

- **Adjusting Motor Speed**:

  - Use PWM values in the Arduino code to set the motor speed, ensuring that the car moves at a steady rate without overshooting turns.
  - Test various speeds by adjusting the analogWrite values on pins 9 and 10 in the code, based on track conditions and surface friction.

- **Final Testing**:

  - Run the car along the entire length of the track to ensure it consistently follows the line.
  - Note any areas where the car might lose the line or struggle to make turns, and make fine adjustments to the motor speed or sensor positions to enhance performance.

Experimental Circuit Setup:

• **IR Sensors** are connected to digital input pins on the Arduino (e.g., pins 2 and 3), which detect the line and send signals to the Arduino.

• **L298N Motor Driver** connects the Arduino to the motors, controlling their direction and speed.

  - **IN1** and **IN2** on the motor driver connect to Arduino pins (e.g., 4 and 5) to control the left motor.
  - **IN3** and **IN4** connect to Arduino pins (e.g., 6 and 7) to control the right motor.

- **ENA** and **ENB** on the motor driver (connected to pins 9 and 10) allow PWM control for motor speed.

- **Power Supply**:
    - The motor driver receives power from the battery pack to drive the motors, while the Arduino is also powered separately through the USB or battery.

- **DC Motors** connect to the **OUT1, OUT2, OUT3, and OUT4** terminals of the motor driver to enable movement of the car.

Arduino Code:

```
int leftSensor = 2;      // Pin for left IR sensor

int rightSensor = 3;     // Pin for right IR sensor

int motorA1 = 4;         // L298N IN1 for left motor

int motorA2 = 5;         // L298N IN2 for left motor

int motorB1 = 6;         // L298N IN3 for right motor

int motorB2 = 7;         // L298N IN4 for right motor

int enableA = 9;         // L298N ENA for left motor speed

int enableB = 10;        // L298N ENB for right motor speed

void setup() {

  // Set sensor pins as input

  pinMode(leftSensor, INPUT);

  pinMode(rightSensor, INPUT);

// Set motor control pins as output

  pinMode(motorA1, OUTPUT);

  pinMode(motorA2, OUTPUT);

  pinMode(motorB1, OUTPUT);

  pinMode(motorB2, OUTPUT);

  pinMode(enableA, OUTPUT);

  pinMode(enableB, OUTPUT);
```

```
 // Set initial motor speeds (0-255)
 analogWrite(enableA, 200); // Set left motor speed
 analogWrite(enableB, 200); // Set right motor speed
 Serial.begin(9600); // Initialize Serial Monitor
}
void loop() {
 // Read the states of the sensors
 int leftState = digitalRead(leftSensor);
 int rightState = digitalRead(rightSensor);
 // Print sensor states to the Serial Monitor
 Serial.print("Left: ");
 Serial.print(leftState);
 Serial.print(" Right: ");
 Serial.println(rightState);
 if (leftState == LOW && rightState == LOW) {
   // Both sensors detect line (move forward)
   digitalWrite(motorA1, HIGH);  // Forward for left motor
   digitalWrite(motorA2, LOW);
   digitalWrite(motorB1, HIGH);  // Forward for right motor
   digitalWrite(motorB2, LOW);
 }
 else if (leftState == LOW && rightState == HIGH) {
   // Left sensor detects line, turn right
   digitalWrite(motorA1, LOW);   // Stop left motor
   digitalWrite(motorA2, LOW);
   digitalWrite(motorB1, HIGH);  // Forward for right motor
   digitalWrite(motorB2, LOW);
 }
 else if (leftState == HIGH && rightState == LOW) {
   // Right sensor detects line, turn left
   digitalWrite(motorA1, HIGH);  // Forward for left motor
   digitalWrite(motorA2, LOW);
   digitalWrite(motorB1, LOW);   // Stop right motor
   digitalWrite(motorB2, LOW);
 }
 else {
   // Both sensors HIGH (no line detected), stop and correct direction
   digitalWrite(motorA1, LOW);
   digitalWrite(motorA2, LOW);
   digitalWrite(motorB1, LOW);
   digitalWrite(motorB2, LOW);
   // Add a small delay to allow time to correct
   delay(100); // Adjust delay as necessary
 }
}
```

Observation:

During testing, the car consistently moved forward when both sensors detected the line. When only one sensor detected the line, the car correctly adjusted its direction to align with the path. Adjustments in motor speed and sensor positions were critical in achieving reliable line-following behavior.in motor speed and sensor positions were critical in achieving reliable line-following behavior..

Discussion:

This project illustrates how integrating simple sensors and motor control logic allows basic autonomous movement. The IR sensors effectively detect the line, with adjustments in code logic and sensor alignment greatly impacting performance. Challenges encountered included sensitivity to lighting conditions and maintaining steady line detection, highlighting the importance of a well-calibrated setup.

Conclusion:

The line-following car successfully met the objective, demonstrating autonomous navigation along a defined path. The project underscores the potential of combining sensor data with motor control in robotics applications. Future enhancements could include speed modulation and obstacle avoidance, expanding the car’s capabilities in more complex environments.