

**MINI PROJECT**

**LINE FOLLOWING ROBOT**

**DEC50122**

**EMBEDDED ROBOTIC**

**KELAS: DET 5C**

**NAMA PENYELIA:**

**TUAN ISHAM SAHAR BIN ISMAIL**

|  |  |  |
| --- | --- | --- |
| **NO.** | **NAMA** | **NO. MATRIK** |
| **1** | **MUHAMMAD FAISAL FAKHRUDDIN BIN AZIZAN** | **03DET22F1003** |
| **2** | **CHE SYAHMI ZUHAIRI BIN CHE ZAHARI** | **03DET22F1056** |
| **3** | **CHONG KHENG CHEN** | **03DET22F1043** |

**Introduction**

In recent years, mobile robots have become an essential part of various applications ranging from industrial automation to service robots and academic research. One of the most basic yet important tasks for mobile robots is line-following, where a robot is designed to detect and follow a line or path using sensors and motors. This task demonstrates key principles of robotics, including sensor integration, motor control, and autonomous navigation.

This project focuses on designing and implementing a line-following robot that can navigate based on input from infrared (IR) sensors. The robot is controlled by an Arduino Nano and utilizes three infrared sensors to detect the line, along with DC motors for movement. A motor driver circuit is used to interface the Arduino with the motors, enabling it to steer the robot in various directions such as forward, left, right, reverse, or halt based on the sensor readings.

The robot's movement is determined by the status of the IR sensors in relation to the path it follows. Depending on the combination of sensor readings, which indicate whether the robot is aligned with the line or deviating from it, the Arduino adjusts the motor speeds and direction to correct the robot’s path. The simplicity of this system makes it ideal for demonstrating fundamental robotics concepts while maintaining a functional design for real-world applications.

**Scope**

Hardware Development:

The robot will be built using a mobile robot base equipped with two DC motors and wheels for movement, controlled by an Arduino Nano and a motor driver. Three infrared sensors will be used to detect the line on the surface, providing input for navigation.

Sensor Integration:

The robot will utilize three infrared sensors positioned in a specific configuration to sense the path. The sensors will be programmed to detect the presence or absence of the line and provide real-time data to the Arduino for processing.

Control System:

The Arduino microcontroller will be programmed to process the sensor input and control the motors accordingly. The robot's movement will follow a predefined set of rules based on a truth table that corresponds to sensor readings. These rules will determine whether the robot moves forward, turns left or right, reverses, or halts.

Motor Control:

The motor driver will interface the Arduino with the DC motors, allowing it to adjust motor speed and direction in response to sensor data. This ensures smooth and responsive movement as the robot navigates along the line.

Simulation and Testing:

The robot’s functionality will be simulated using Proteus software to ensure proper sensor-motor interaction and navigation before physical implementation. Real-time testing on hardware will validate the robot’s ability to follow a line in various conditions, such as curved or straight paths.

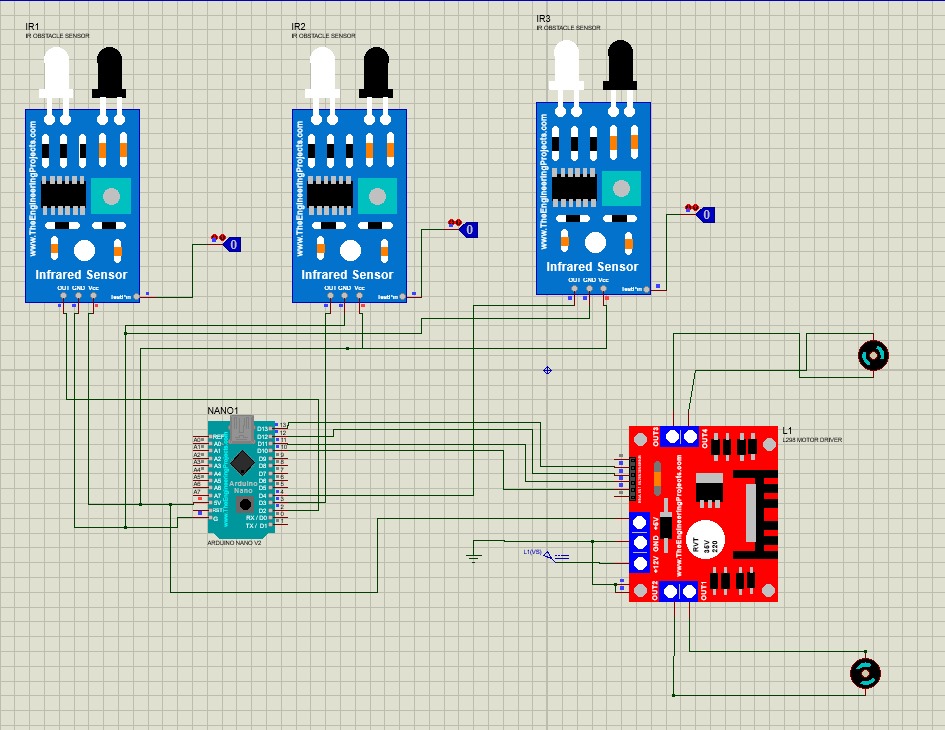
Software Development:

The control logic will be implemented using Arduino’s programming environment, ensuring that the robot can autonomously navigate along a line using the specified sensor-motor interaction rules. Debugging and code optimization will be performed to enhance performance.

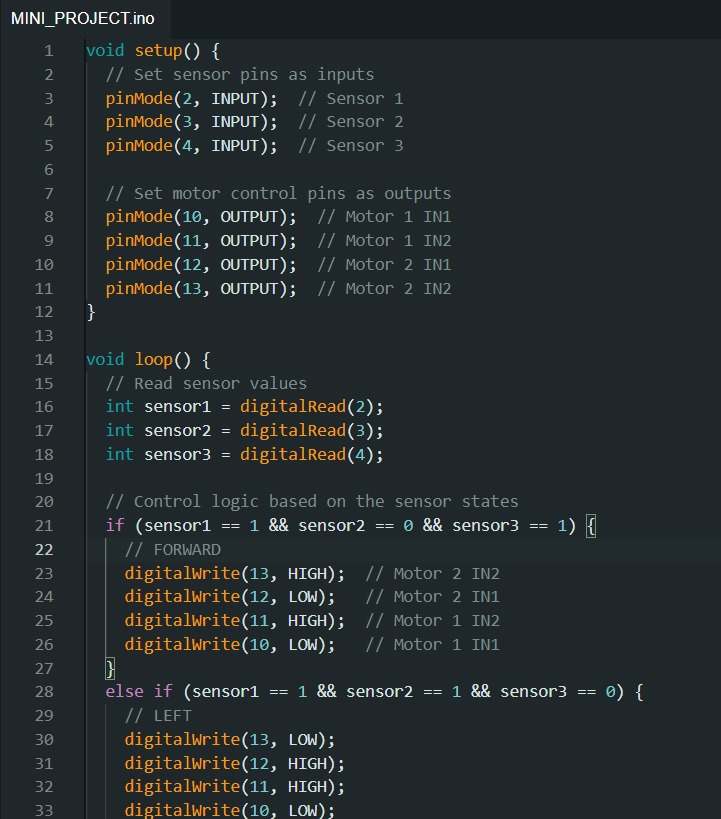
**Problem Statement**

* Autonomous mobile robots are important in automating tasks in industries, such as material transport.
* Designing a line-following robot involves challenges like sensor integration, motor control, and accurate path detection.
* The project aims to develop a cost-effective, simple line-following robot using: Three infrared sensors, Arduino Nano, DC motors with a motor driver.
* Mobile robot base
* The robot will follow a predefined truth table to navigate in different directions (forward, left, right, reverse, and halt).
* Simulation using Proteus software will validate the robot’s functionality before real-world testing.

**Circuit**



**Arduino coding (ARDUINO IDE)**

****

**A screenshot of a computer program

Description automatically generated**

**Result**

A computer screen shot of a computer program

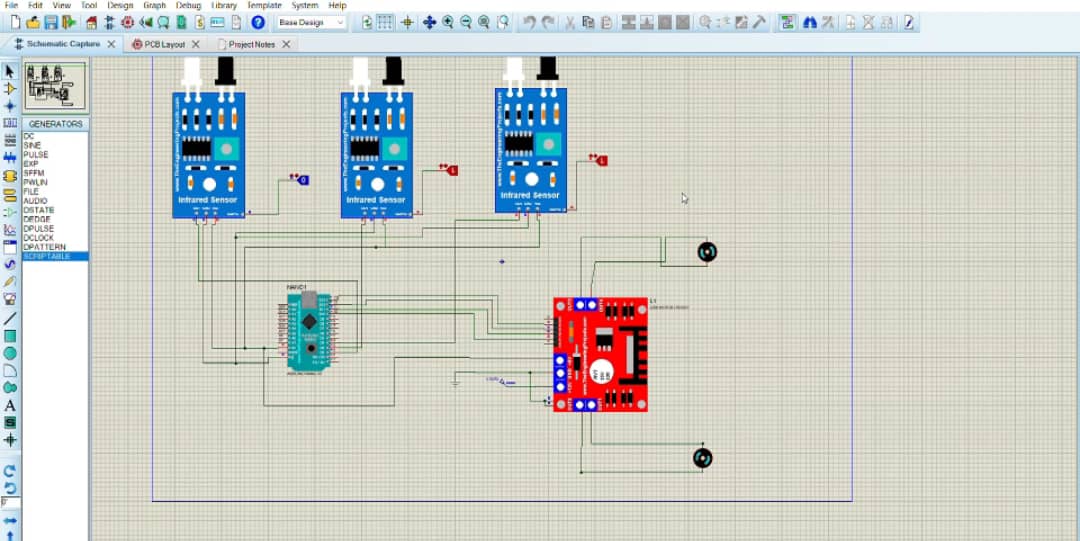
Description automatically generated

**FORWARD**

A computer screen shot of a computer program

Description automatically generated

**LEFT**



**RIGHT**

A computer screen shot of a computer program

Description automatically generated

**REVERSE**

A computer screen shot of a computer

Description automatically generated

**HALT**

**Discussion and analysis**

Based on this mini project, we discuss and analysis to developing a line following robot involves several challenges, including sensor integration, motor control, path detection, and the limitations of simulation accuracy. Accurate line detection can be difficult on different surfaces, which can be addressed by calibrating the sensors to varying conditions and considering additional sensors for improved reliability. Ensuring motors run at consistent speeds is crucial to avoid drifting from the line; this can be achieved using a PID control system for stabilization. The robot must also effectively navigate curves and intersections, which can be managed by creating a detailed truth table for sensor readings that guides movement decisions. While simulation results from software like Proteus are helpful, real-world testing is essential for refining parameters based on actual performance. Balancing quality components with budget constraints is another consideration; researching affordable parts and utilizing open-source resources can help achieve this. The performance analysis should include simulation testing to validate control logic, followed by real-world testing to evaluate adaptability on various line shapes. Iterative improvements based on test results will refine both hardware and software, while future enhancements could involve adding features like obstacle detection or advanced decision-making algorithms. Overall, this project effectively demonstrates key robotics concepts and lays a strong foundation for further developments in mobile robotics.

**Recommendation**

* **Enhance Sensor Calibration:** Implement adjustable thresholds for the infrared sensors to improve line detection accuracy on various surfaces and lighting conditions. This will ensure more reliable performance in different environments.
* **Utilize PID Control:** Consider integrating PID (Proportional-Integral-Derivative) control for motor speed regulation. This approach can help maintain a smoother path following by dynamically adjusting motor speeds based on sensor input.
* **Implement Redundancy:** Add additional infrared sensors to provide redundancy and improve reliability. This will help the robot continue functioning even if one sensor fails or provides inaccurate readings.
* **Test Different Path Configurations:** Conduct extensive testing on various line shapes, including curves and intersections. This will help refine the control logic and truth table used for navigation, ensuring the robot can handle real-world scenarios effectively.
* **Explore Advanced Algorithms:** Investigate the potential for using more sophisticated algorithms, such as fuzzy logic or machine learning, to improve decision-making in complex environments. This can enhance the robot's adaptability and performance.
* **Optimize Power Management:** Ensure efficient power management for the motors and sensors. Consider using battery monitoring to optimize battery life and performance during operation.
* **Incorporate Obstacle Detection:** To expand the robot's capabilities, consider integrating ultrasonic or infrared distance sensors for obstacle detection. This will allow the robot to navigate around obstacles rather than just following a line.
* **Enhance Communication:** If applicable, implement wireless communication features (like Bluetooth or Wi-Fi) for remote monitoring or control. This can make it easier to observe the robot's performance and make adjustments on the fly.
* **Document the Process:** Keep detailed records of the design process, testing results, and code versions. This documentation will be valuable for troubleshooting and future enhancements.
* **Engage in Community Feedback:** Share your project with robotics communities or forums to gather feedback and suggestions. Engaging with others can provide new ideas and perspectives that can improve your design.

**Refferences**

 **Books:**

* **"Robotics, Vision and Control: Fundamental Algorithms in MATLAB" by Peter Corke**  
  This book provides a comprehensive introduction to robotics, covering key concepts, algorithms, and practical applications, including sensor integration and motor control.
* **"Arduino Robotics" by John-David Warren, Josh Adams, and Alicia G. C. Wright**  
  This book focuses on building robots with Arduino, covering components, programming, and practical projects like line-following robots.
* **"Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms" by George A. Bekey**  
  This book gives insights into the design and functionality of autonomous robots, emphasizing the integration of sensors and control systems.

**Articles and Journals:**

* **"A Survey of Line Following Robots"**  
  This article reviews various approaches and technologies used in line-following robots, discussing different sensors and control methods.
* **"Design and Implementation of a Line Following Robot" by Shashank Y. et al.**  
  A detailed paper discussing the design process, challenges, and solutions for creating a line-following robot using Arduino and IR sensors.

**Online Resources:**

* **Arduino Project Hub: Line Following Robot**  
  Arduino Project Hub contains a variety of projects and tutorials, including those focused online-following robots.
* **Spark Fun Electronics: Line Following Robot Tutorial**  
  Spark Fun provides an informative tutorial on building a line-following robot, including circuit diagrams and programming tips.
* **YouTube Tutorials**  
  Look for video tutorials on platforms like YouTube that demonstrate building and programming line-following robots using Arduino. These can provide visual guidance and troubleshooting tips.

**Datasheets and Technical Documentation:**

* **Arduino Nano Documentation**  
  The official Arduino documentation provides technical specifications, pin configurations, and programming guidance for the Arduino Nano.
* **Motor Driver Documentation (e.g., L298)**  
  Datasheets for motor drivers like the L298 provide essential information on wiring, usage, and electrical characteristics necessary for motor control.

**Open-Source Libraries:**

* **Arduino Library for PID Control**  
  Consider exploring libraries such as the Arduino PID Library, which can help implement PID control for motor speed regulation in your project.