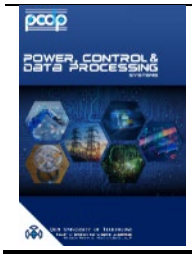
DOI: 10.30511/pcdp.2026.2091135.1082 - ISSN: 3060-8376 - Vol 4(1), 2027



POWER, CONTROL AND DATA PROCESSING SYSTEMS

Available Online at: https://pcdp.qut.ac.ir/

Design and Implementation of a Safe Line-Following Robot with Automatic Climate and Lighting Control Systems

ARTICLE INFO

Article Type
Original Research

Authors
Ali Razzaghi[1*]
Azin Rahimi Shateranloo[2]
Negar Bakhtiari[3]
Behzad Ebrahimi[4]

[1] Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran, a.razzagh@modares.ac.ir
[2] L'École d'Ingénieurs Polytech Sorbonne, Spécialité Électronique et Informatique Embarquée (EIE), Sorbonne Université, Paris, France, Azin.Rahimi_Shateranloo@etu.sorbonne-universite.fr
[3] Department of Electrical and Computer Engineering, SR.C., Islamic Azad University, Tehran, Iran, negar.bakhtiari79@gmail.com
[4] Department of Electrical and Computer Engineering, SR.C., Islamic Azad University, Tehran, Iran, behzad.ebrahimi@iau.ac.ir

**** Correspondence***
Address: Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran,

Phone: -
Fax: -
a.razzagh@modares.ac.ir

Article History
Received: June 09, 2026
Accepted: August 04, 2026
ePublished: March 01, 2027

ABSTRACT

This paper presents the design, implementation, and experimental evaluation of an autonomous line-following robot developed for efficient material transportation in dynamic industrial environments. Unlike conventional line-following robots that focus only on path tracking, the proposed system integrates automatic lighting control, onboard thermal management, and access security as core operational functions. These additional features enable the robot to maintain reliable performance under variable lighting and temperature conditions while preventing unauthorized use in shared workspaces. The navigation system is based on five TCRT5000 infrared sensors, which provide accurate line detection and stable maneuvering along complex trajectories, including sharp 90-degree turns. To enhance environmental adaptability, the robot automatically activates its lighting system in low-light conditions and employs a temperature-regulated cooling mechanism to protect onboard electronics and sensitive payloads. An access control layer is also incorporated to improve operational safety and reduce tampering risks. Experimental results demonstrate that the proposed robot achieves reliable path-following performance and robust subsystem response under different operating conditions. The combination of precise navigation, environmental adaptation, and security functions makes the system a practical and scalable solution for industrial automation and internal logistics in modern factory settings.

Keywords: Automatic Climate Control, Automatic Lighting Control, AVR ATmega32 Microcontroller, Line-Following Robot, Security System.

1 Introduction

Line-following robots are systems designed to detect and follow designated paths, typically characterized by contrasting colors. Their primary function is to operate independently within environments, making them ideal for applications in industrial automation, education, and competitive scenarios. Initially introduced in the early days of robotics, line-following robots have long served as an essential project for learning fundamental concepts in robotics, such as sensor integration and control systems. Over time, these robots have evolved into sophisticated systems, with applications spanning factory automation to intelligent transportation solutions.

In [1] an automated solution for hospital room service was proposed using line-following robots. These robots effectively transported medications, meals, and other supplies to isolated patients, thereby reducing the risk of infection for healthcare workers. Connected via IoT over a Wi-Fi network, the robots navigated hospital corridors using sensors and predefined pathways, showcasing improvements in efficiency, collision avoidance, and reduced delays. Future upgrades could include advanced sensors, more powerful processors, and improved inter-robot communication.

A study in [2], compared two control strategies: a standard Proportional-Derivative (PD) and a dynamic PD for line-following robots. The dynamic PD controller adapted its parameters in real-time, whereas the standard PD controller used fixed parameters. The performance evaluation included measuring completion time and stability on a test track.

In [3], the enhancement of a competitive line-following robot was explored through improved control strategies and the use of digital filters. The research emphasized mathematical modeling, hardware design, and control analysis, focusing on dual control systems and digital signal processing. The robot featured optimized hardware components and advanced software, utilizing PID control and digital filters for noise reduction and improved stability.

An analysis in [4] focused on an Automated Guided Vehicle (AGV) employing infrared (IR) sensors for line-tracing in industrial storage facilities. This AGV aimed to improve material handling efficiency by autonomously navigating along a line of reflective tape in warehouse environments. The integration of components like microcontrollers, IR sensors, and proximity sensors enabled efficient navigation and obstacle evasion, contributing to streamlined warehouse operations, reduce labor costs, and enhanced productivity.

In [5], a line-following robot designed for complex navigation was discussed, featuring an L298 motor driver, an Arduino UNO, and twelve infrared sensors. Nine sensors facilitated spiral path navigation, while three supported dashed-line tracking. The design minimized tracking inaccuracies by employing advanced sensor arrangements and programming strategies.

A low-cost Arduino-based robot designed for automated garden maintenance was presented in [6] facilitating the distribution of water, fertilizers, and pesticides. This robot followed a black line on a white surface within the garden and stopped at designated points to release liquids from its onboard reservoir. Constructed from basic materials like wood and plastic, it operated on a 12-volt battery. While it effectively detected lines and performed watering tasks, its performance was susceptible to variations in lighting and sensor limitations.

In [7] a line-following robot is described that utilizes the Xilinx ZYNQ-7000 SoC for both hardware and software functions. The innovation maintains low energy consumption while achieving high accuracy in line tracking through advanced power management techniques. The methodology includes defining the robot's parameters, designing its hardware, developing the software, and optimizing power usage.

A project discussed in [8] introduced the development and deployment of an infrared line-following robot capable of moving independently along pre-arranged tracks. To control its movements, the robot utilized an Arduino Uno, while the ESP8266 microcontroller enabled web-based interaction. The project involved creating a robust framework for payload support and designing a strategy that allowed the robot to navigate tight corners smoothly. By incorporating IoT capabilities, users could remotely operate the robots and safeguard the items being transported. The study also demonstrated how effectively the robot traveled along a test path with varying backgrounds [8].

A study in [9] examined the design and development of a line-following robot, focusing on its electronic, mechanical, and control systems. The methodology involved designing and simulating the robot's PCB using specialized tools, assessing mechanical loadings on the body frame, as well as implementing a PID controller for accurate positioning. This research highlighted the importance of an interdisciplinary approach to optimize the robot's productivity and versatility in manufacturing applications.

A study in [10], discussed the design and hardware application of a line-following robot using an Arduino microcontroller. It covered the structure of the robot, including the use of IR sensors for line detection, a driver circuit for motor control, and the overall movement system. The practical implementation involved interfacing the microcontroller with sensors and motors, programming the system, and ensuring precise movement along a predefined path. The line-following robot was highlighted for its potential applications in both industrial and domestic settings, offering a cost-effective solution for tasks such as transportation, cleaning, and material transfer.

Recent advancements in 2025 and 2026 have further expanded the capabilities and applications of Automated Guided Vehicles (AGVs) and line-following robots. For instance, in agricultural settings, microcontroller-based line follower robots integrating infrared arrays and color sensors have been developed for automated post-harvest transportation [11]. In intralogistics, modern frameworks have integrated AGVs with RFID and obstacle-avoidance control to significantly enhance

warehouse throughput and productivity [12]. Security and controlled environments have also seen new applications, such as the design of specialized AGVs for transporting sensitive exam papers and documents in exam control rooms [13]. Furthermore, contemporary research has focused on advanced hardware architectures, demonstrating the use of ESP32 microcontrollers combined with Time-of-Flight (ToF) sensors for robust line following and obstacle avoidance [14], as well as high-speed line-following platforms designed to enhance autonomous robotics [15].

While existing research has extensively explored advancements in line-following robots, focusing primarily on sensor accuracy, navigation algorithms, and basic mobility, there remains a significant gap in integrating robust environmental adaptation mechanisms for reliable operation in real-world industrial and healthcare settings. Many previous works have treated factors like lighting and temperature as external variables that might affect performance, rather than as integral aspects to be managed by the robot itself. This paper addresses this limitation by presenting the development of a line-following robot that integrates automatic lighting control and thermal regulation as core functionalities. This unique combination ensures consistent operational reliability in diverse factory conditions, such as low-light environments and areas with fluctuating temperatures, thus preserving the integrity of transported goods and internal robotic systems (e.g., electronic boards, lighting modules). Furthermore, when deployed in healthcare environments, the thermal regulation system offers precise temperature control, crucial for transporting sensitive pharmaceuticals and food supplies.

Distinct from previous efforts, our system also incorporates an advanced access security layer to restrict operation to authorized personnel, mitigating tampering risks in shared workspaces. Crucially, the robot's navigation relies solely on its onboard sensors and pre-defined paths, eliminating the need for external infrastructure like GPS or constant internet connectivity. This not only reduces deployment costs and complexity but also ensures uninterrupted operation even during GPS signal loss or network disruptions, making it ideal for environments with unreliable connectivity.

The collective integration of these features—robust navigation, proactive environmental adaptation, enhanced security, and independent operation—distinguishes our work from prior efforts. This paper presents a prototype that demonstrates the potential for a versatile and scalable solution for automating internal logistics, material handling, and delivery services in modern industrial, warehousing, and healthcare facilities, adaptable to various dimensions and operational scales.

Figure 1 illustrates the operational concept of the proposed robotic system within a healthcare environment. As shown, the line-following robot is capable of adapting to varying ambient lighting conditions by automatically activating its onboard lighting system in dark corridors and deactivating it in sufficiently illuminated areas. In addition, the robot is equipped with a dedicated storage compartment featuring automatic cooling control, which enables the safe transportation of temperature-sensitive items such as medications. This conceptual representation highlights the integrated functionality of autonomous navigation, adaptive lighting, and thermal management in a hospital setting.



Figure 1 Functional overview of the robot: adaptive lighting in varying environments and the automatic cooling compartment for medical transport

2 Structure of the System

The proposed robot is designed as an embedded autonomous navigation platform intended for operation in indoor environments such as hospitals, laboratories, and industrial corridors. In addition to basic line-following capability, the robot integrates three auxiliary functions to improve operational reliability and safety: an automatic lighting system, a self-ventilation system, and an access security mechanism. These functions are coordinated through a deterministic microcontroller-based architecture to ensure real-time response under varying environmental conditions.

The overall system is implemented using two AVR microcontrollers, namely the ATmega32 and ATmega8, programmed in C using the CodeVisionAVR development environment. The ATmega32 serves as the main control unit responsible for navigation, environmental monitoring, and actuator coordination, while the ATmega8 is assigned to auxiliary control and security-related tasks. This dual-controller arrangement reduces computational load on the main processor, simplifies task distribution, and improves the modularity of the embedded system. From a control perspective, the robot does not rely on model-based continuous control such as PD or PID algorithms. Instead, it employs a set of deterministic rule-based algorithms composed of threshold comparison, discrete state transitions, and sequential decision logic. This approach is well suited to the limited computational resources of 8-bit AVR microcontrollers and provides stable real-time performance for the intended application.

2.1 First Subsystem

The first subsystem is the line-following navigation unit, which uses an array of TCRT5000 infrared sensors. This

subsystem operates according to a discrete sensor-based decision algorithm in which the reflectance values of the floor are converted into binary navigation states. Based on the detected line position, the controller generates corresponding motor commands for forward motion, left correction, right correction, sharp turning, or stopping. In algorithmic terms, this subsystem can be described as a finite-state decision process in which each sensor pattern is mapped to a predefined movement command. Therefore, the navigation logic is implemented as a rule-based line tracking algorithm using discrete state mapping.

2.2 second subsystem

The second subsystem is the automatic lighting unit, which is activated when the robot enters a low-illumination environment. A photo resistive sensor is used to detect ambient light intensity, and its analog output is digitized through the internal analog-to-digital converter of the microcontroller. The measured value is then compared with a predefined threshold to determine whether auxiliary lighting should be turned on or off. Thus, the lighting control is based on a threshold-driven switching algorithm. If desired, hysteresis can also be considered in the control logic to prevent repeated switching near the threshold boundary.

2.3 third subsystem

The third subsystem is the self-ventilation or thermal protection unit, which is designed to protect the internal electronic components and, where applicable, temperature-sensitive payloads. A temperature sensor continuously monitors the thermal condition of the robot. When the measured temperature exceeds a specified upper limit, the cooling fan is activated automatically; when the temperature returns to an acceptable level, the fan is deactivated. This mechanism follows an on-off thermal control strategy, also referred to as bang-bang control or threshold-based thermal regulation. Such a method is particularly effective in embedded applications where simplicity, reliability, and fast response are more important than continuous proportional control.

2.4 fourth subsystem

The fourth subsystem is the security mechanism, which restricts robot activation to authorized users. This unit is implemented through a keypad-based password entry system. The entered digits are compared with a predefined four-digit code stored in the microcontroller memory. Access is granted only if the entered sequence exactly matches the stored pattern; otherwise, the system remains locked. From an algorithmic viewpoint, this subsystem employs a sequential verification algorithm based on pattern matching and logical comparison. This provides a simple but effective access control layer for operation in shared or public environments.

The complete robot operation can therefore be interpreted as an integrated supervisory sequence. First, the system verifies user authorization through password validation. After successful activation, the line-following module continuously determines the robot trajectory using sensor-based state decisions. Simultaneously, the lighting subsystem monitors ambient illumination and activates the onboard lamp, when necessary, while the thermal protection subsystem monitors temperature and controls ventilation accordingly. Figure 2 presents the overall operational flow of the robot and the interaction among these embedded control modules.

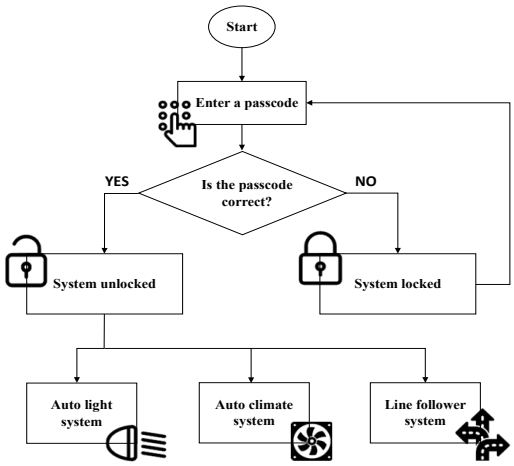


Figure 2 Flowchart illustrating the robot operation process

3 Electronic Design

The system incorporates two microcontrollers: the ATmega-32 and ATmega-8. The ATmega-32 microcontroller is responsible for managing the robot's additional functionalities, while the ATmega-8 handles the primary line-following task. Upon powering the robot, the system is designed to activate a security feature, prompting the user to enter a password. If the correct password is entered, the ATmega-32 sends a command to activate the robot, and the ATmega-8 is engaged to perform its line-following function. The following section outlines the robot's capabilities in each part of the electronic circuit design.

3.1 Line Follower

The robot is designed to recognize path lines and autonomously navigate along path lines using TCRT5000 sensors, which consist of infrared transmitters and receivers. The sensors are strategically positioned at the front and underneath the robot. The line should appear black against a white background, allowing the robot to identify and track it accordingly. Due to the winding nature of the path, relying on a single sensor is insufficient. The number of sensors needed depends on the path's complexity. To enable the robot to navigate 90-degree turns, two sensors are placed on the left and right, with three positioned in the center. When the path

curvature is less than 90 degrees, the robot uses the three central sensors for guidance; however, for 90-degree turns, the side sensors are employed.

The robot's sensors and motors must be connected to the microcontroller for effective navigation. The distinction between the two lies in that sensors act as input devices, whereas motors serve as outputs to the microcontroller. The infrared transmitting sensor directs its signal forward, and when this signal encounters a white object or surface, a significant portion of the infrared is reflected in various directions. A substantial amount of this reflection reaches the receiving sensor located near the transmitter, indicating the presence of a white object. In contrast, if a black object is in front of the sensors, it absorbs most of the infrared radiation, reflecting very little. Consequently, the receiver detects the absence of light, identifying the black object.

3.1.1 Sensor Signal Processing and Mathematical Modeling

The autonomous guidance of the robot relies on detecting the optical contrast between the high-absorption black navigation line and the high-reflectance white background using a five-sensor array of TCRT5000 infrared reflective sensors.

To convert the continuous analog reflectance levels into discrete digital signals suitable for the 8-bit ATmega32 microcontroller, a signal conditioning circuit using an operational amplifier (OPAMP) as a voltage comparator is employed. The electrical behavior of each infrared receiver channel can be modeled using a voltage divider circuit. The sensor output voltage $V_{sensor,i}$ (for $i \in \{1,2,3,4,5\}$ is expressed as equation 1:

$$V_{sensor,i} = V_{cc} \cdot \left(\frac{R_{series}}{R_{series} + R_{IR,i(\rho i)}} \right) \quad (1)$$

- V_{cc} is the system supply voltage (5V).
- R_{series} is the series current-limiting resistor connected to the phototransistor emitter (10kΩ)
- $R_{IR,i(\rho i)}$ is the dynamic internal resistance of the i-th infrared phototransistor, which is a function of the surface reflectivity coefficient ρi

The surface reflectivity coefficient ρi takes values in the range [0,1] where $\rho i \approx 0$ represents a black line (low reflection/high resistance) and $\rho i \approx 1$ represents a white background (high reflection/low resistance). Consequently, the dynamic resistance behaves as equation 2:

$$R_{IR,i(\rho i)} = \begin{Bmatrix} R_{high \approx \infty} & if\ \rho i \to 0\ (Black\ Line) \\ R_{low \approx 0} & if\ \rho i \to 1\ (White\ Background) \end{Bmatrix} \quad (2)$$

The analog voltage $V_{sensor,i}$ is processed by the OPAMP, which functions as a comparator generating a binary output $D_{in,i}$ based on a step function equation 3:

$$D_{in,i} = H(V_{sensor,i} - V_{ref} = \begin{Bmatrix} 1 & if\ V_{sensor,i} \geq V_{ref} \\ 0 & if\ V_{sensor,i} < V_{ref} \end{Bmatrix} \quad (3)$$

Where $H(x)$ is the Heaviside step function, and V_{ref} is the reference voltage threshold. In this design, V_{ref} is set to 1V using a 10kΩ multi-turn potentiometer. When a sensor passes over the black line ($\rho i \to 0$) the phototransistor cuts off, causing $R_{IR,i}$ to spike, which drops $V_{sensor,i}$ below V_{ref} yielding $D_{in,i} = 0$. Conversely, over the white background ($\rho i \to 1$) the sensor outputs $D_{in,i} = 1$.

3.1.2 Navigation Control Algorithm: Discrete State Logic

To regulate the robot's heading along the path, a Deterministic Finite State Machine (FSM) is implemented. Let the state of the sensor array be represented by a 5-dimensional binary vector equation 4:

$$S = [D_{in,1}, D_{in,2}, D_{in,3}, D_{in,4}, D_{in,5}]^T \in \{0,1\}^5 \quad (4)$$

The control system calculates the lateral tracking error $e(k)$ at any discrete time step k. The target position is alignment with the central sensor ($D_{in,3}$). The lateral error $e(k)$ is defined by assigning spatial weights w_i to each senso equation 5:

$$e(k) = \frac{\sum_{i=1}^{5} (1 - D_{in,i}).w_i}{\sum_{i=1}^{5} (1 - D_{in,i})} \quad (5)$$

where the spatial weights corresponding to the sensors from left to right are defined as equation 6:

$$W = [w1, w2, w3, w4, w5] = [-2, -1, 0, 1, 2] \quad (6)$$

Instead of using a computationally demanding continuous proportional-derivative (PD) controller, the control action is computed via a discrete state mapping function $f(s)$ that outputs the velocity commands for the left and right motors, denoted as $M = [vL, vR]^T$ in equation 7

$$M(k) = f(S(k)) \quad (7)$$

The mapping function $f(s)$ resolves the steering commands based on three operating regions:

1. **Central Guidance** ($e(k) \approx 0$): If the line is detected by the central sensors ($D_{in,2}, D_{in,3}, D_{in,4} = 0$) the robot maintains forward progress with balanced nominal velocity $v0$ (equation 8):
$$vL = v0 \ , vR = v0 \quad (8)$$
2. **Cornering Logic** ($|e(k)| \geq 1$): If a lateral deviation or a 90° turn is detected via peripheral activations $D_{in,1} = 0$ or $D_{in,5} = 0$ differential steering is executed to apply a restorative yaw moment (equation 9):
$$\begin{Bmatrix} vL = -v_{turn,} & vR = v_{turn,} & if\ e(k) < 0\ (Turn\ Left) \\ vL = v_{turn,} & vR = -v_{turn,} & if\ e(k) > 0\ (Turn\ Right) \end{Bmatrix} \quad (9)$$
Where $v_{turn,}$ is the preset turning velocity designed to handle sharp curves.
3. **Safety Stop Condition:** If all sensors detect the black path line simultaneously, indicating a designated stop station equation 10:
$$\sum_{i=1}^{5} (1 - D_{in,i}) = 5 \rightarrow vL = 0,\ vR = 0 \quad (10)$$

This discrete rule-based mapping is summarized in Table 1.

Table 1 Robot movements based on sensor states

Sensor Number	Binary Value and Robot Movement				
Sensor 1	1	0	0	0	0
Sensor 2	0	1	0	0	0
Sensor 3	0	0	1	0	0
Sensor 4	0	0	0	1	0
Sensor 5	0	0	0	0	1
Move	Left=90°	Left<90°	straight	Right<90°	Right=90°

By utilizing this deterministic control logic, the 8-bit AVR microcontroller minimizes execution latency, avoiding complex floating-point calculations while preserving stability during high-speed transitions.

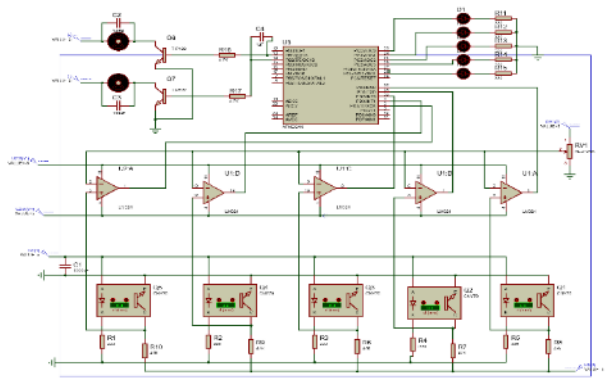


Figure 3: Schematic diagram of the line-tracking sensor array and control interface.

The sensor interface module consists of five TCRT5000 or CNY70 infrared reflective sensors arranged in a linear array to facilitate path detection. The output of each sensor is conditioned using an LM324 operational amplifier (OPAMP) configured as a voltage comparator. This configuration provides a clean transition between the logic levels 0 (low reflectivity, black line) and 1 (high reflectivity, white surface), which are directly interfaced with the I/O ports of the ATmega8 microcontroller.

To provide real-time visual feedback of the sensor states, five LEDs (D1–D5) are integrated into the signal path, where each LED illuminates upon the detection of the black line by its corresponding sensor (Q1 -Q5) The motor drive stage utilizes two TIP122 Darlington transistor pairs, allowing the microcontroller to control high-current DC motors via Pulse Width Modulation (PWM) or discrete high/low switching signals. The inclusion of a 10 kΩ variable resistor (RV1) allows for manual calibration of the reference voltage (Vref), enabling the robot to maintain navigation stability across varying surface conditions and ambient light levels.

3.2 Auto Light

One of the key challenges in developing this robot lies in its ability to navigate effectively under low-light conditions. The automatic lighting system ensures that the robot's headlights automatically turn on when it enters dark or poorly lit areas. Conversely, the lights switch off independently in daylight or well-lit environments. This system employs a photosensitive sensor that assesses the intensity of ambient light, determining whether the surroundings are bright or dim.

The sensor then transmits this information to the electronic board, which activates or deactivates the robot's lights via a relay.

For the robot's headlights, 12-volt LEDs were used to provide optimal illumination. Since the microcontroller's output voltage is limited to a maximum of 5 volts, it cannot directly activate the lights, necessitating a separate 12-volt power supply for this purpose. The circuit designed for this includes a 12-volt relay; a BC337 transistor, a 1N4007 diode, and a resistor connected to the base of the transistor. When selecting the resistor, it is essential to ensure that the current drawn by the microcontroller does not exceed 20 milliamps.

3.2.1 Photo-Electric Sensing and Signal Conversion

The adaptive lighting module utilizes a Light Dependent Resistor (LDR) to detect ambient illuminance. The LDR is integrated into a voltage divider circuit to convert the variable resistance R_{LDR}(which is inversely proportional to the light intensity E_e) into a measurable voltage V_{ADC} The output voltage is defined as equation 11:

$$V_{ADC} = V_{cc}.\left(\frac{R_{LDR}(E_e)}{R_{fixed} + R_{LDR}(E_e)}\right) \quad (11)$$

Where $V_{cc} = 5V$ and $R_{fixed} = 1k\Omega$ The analog voltage is then processed by the internal 10-bit Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) of the ATmega microcontroller. The resulting digital value, X_{ADC} is calculated as equation12:

$$X_{ADC} = \left(\frac{V_{ADC}}{V_{ref}}.(2^n - 1)\right) \quad (12)$$

Given a 10-bit resolution $n = 10$ and $V_{ref} = 5V$ the quantization levels range from 0 to 1023.

3.2.2 Hysteresis Control Logic

To prevent rapid switching (flickering) of the headlights during marginal lighting transitions or due to sensor noise, a Hysteresis-based Thresholding logic is implemented. Let $S_{light}(k)$ be the state of the relay at time step k. The control law is governed by equation 13:

$$S_{light}(k) = \begin{cases} 1(ON) & if\ X_{ADC} > \theta_{high} \\ 0(Off) & if\ X_{ADC} < \theta_{low} \\ S_{light}(k-1) & if\ \theta_{low} \le X_{ADC} \le \theta_{high} \end{cases} \quad (13)$$

In this implementation, the upper and lower thresholds are empirically tuned to $\theta_{high} = 880$ and $\theta_{low} = 850$ respectively. This creates a dead-band of 30 units, ensuring system stability.

3.2.3 Actuation and Power Interface

The microcontroller's I/O pin drives a BC337 NPN bipolar junction transistor (BJT) operating in the saturation region to

act as an electronic switch for the 12V relay. The base current I_B is limited by a 4.7kΩ resistor (R_B) to ensure it remains within the safe operating limits of the AVR microcontroller ($I_{pin} < 20mA$) equation 14:

$$I_B = \frac{V_{OH} - V_{BE}}{R_B} = \frac{5 - 0.7}{4700} = 0.91mA \quad (14)$$

Figure 4 illustrates the schematic diagram of the automatic lighting subsystem implemented in the proposed robot. The circuit is built around an ATmega32 microcontroller and uses an LDR-based voltage divider as the ambient-light sensing element. The analog signal generated by the sensor is read through the microcontroller's ADC input and processed according to the programmed threshold values. When the detected light intensity falls below the specified limit, the microcontroller activates a BC337 transistor, which in turn energizes a 12-V relay to switch the headlights on. A 1N4007 diode is connected across the relay coil to suppress reverse voltage spikes and protect the switching transistor. In addition, the LED indicators provide real-time visual feedback of the sensor and system status during operation.

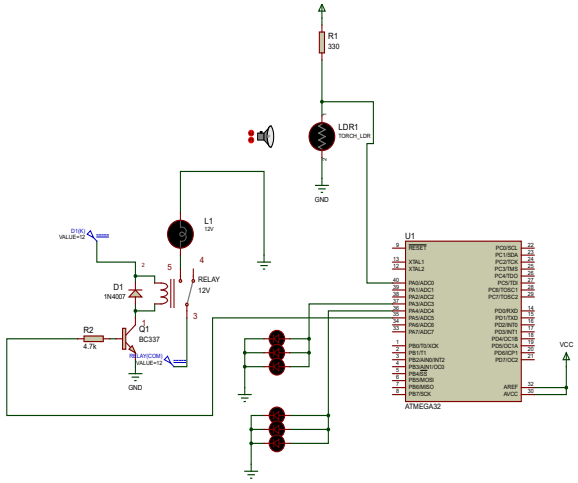


Figure 4 Schematic of the automatic lighting subsystem implemented in the robot.

Figure 5 presents a flowchart illustrating the operation of the automatic lighting system.

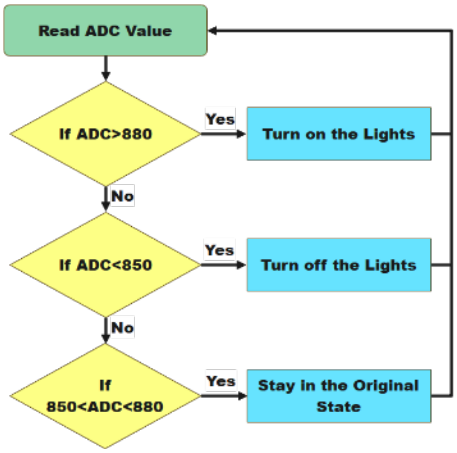


Figure 5 Flowchart of the auto-light system operation

3.3 Auto Climate

Achieving optimal performance over extended periods is another critical factor in the design of a navigation robot. The presence of regulators, DC motors, and other components that generate heat over time can shift the operating point and reduce the robot's efficiency. To address this issue, a system has been developed that continuously monitors ambient temperature using the LM35 sensor. Based on the sensor reading, the system automatically activates the cooling fans, as programmed, while displaying the measured temperature and fan status on an LCD screen. The activation circuit for the ventilation system is similar to the auto-light system, utilizing a BC337 transistor, a 1N4007 diode, and a single-pole 12-volt relay. According to previous calculations, a 4.7 kilohm resistor is suitable for the transistor base, and the circuit is connected to the microcontroller pin.

Heat generated by the voltage regulators, motor-driving components, DC motors, and other electronic devices may accumulate during extended operation and affect the repeatability and reliability of the robot. Therefore, an automatic thermal-management subsystem was implemented to monitor the temperature near the electronic control board and activate the cooling fans when necessary. An LM35 temperature sensor was used because its output voltage is approximately linearly proportional to temperature in degrees Celsius. The sensor output can be expressed as equation 15:

$$V_{LM35}(k) = \alpha T(k) + V0 \quad (15)$$

where $V_{LM35}(k)$ is the sensor output voltage at sampling instant k, $T(k)$ is the measured temperature in degrees Celsius, $\alpha = 10mV/°C$ is the nominal sensitivity of the LM35, and $V0$ is its nominal zero-temperature offset, which is approximately zero for the operating range considered in this study.

The analog output of the LM35 is sampled by the n-bit analog-to-digital converter (ADC) of the microcontroller. If $N_T(k)$ denotes the resulting ADC code, the measured voltage is estimated as equation 16:

$$V_{LM35}(k) = \frac{N_T(k)}{2^n - 1} V_{ref} \quad (16)$$

Where V_{ref} is the ADC reference voltage. By combining the preceding equations, the temperature calculated by the microcontroller is equation 17:

$$T(k) = \frac{V_{ref}}{\alpha(2^n - 1)} N_T(k) \quad (17)$$

Thus, the ADC codes corresponding to the selected switching temperatures are approximately equations 18,19:

$$N_{ON} = round\left(\frac{\alpha T_{ON}}{V_{ref}}(2^n - 1)\right) \quad (18)$$

$$N_{OFF} = round\left(\frac{\alpha T_{OFF}}{V_{ref}}(2^n - 1)\right) \quad (19)$$

The cooling system is controlled using two temperature thresholds rather than a single switching point. Let $S_f(k) \in \{0,1\}$ denote the fan state, where $S_f = 1$ represents an

energized cooling fan and $S_f = 0$ represents a deactivated fan. The implemented hysteresis control law is equation 20:

$$S_f(k) = \begin{cases} 1 & T(k) \geq T_{ON} \\ 0 & T(k) \leq T_{OFF} \\ S_f(K-1) & T_{OFF} < T(k) < T_{ON} \end{cases} \quad (20)$$

The values of 33 and 21 were adopted as prototype-level control settings rather than universal thermal limits. The upper threshold represents an early cooling point for the robot's internal environment: it activates ventilation before substantial heat accumulation occurs around the regulators, motor-driving circuitry, and control board. It should not be interpreted as the maximum permissible junction temperature of an individual electronic component, since component junction temperatures may be higher than the temperature measured by the LM35 and depend on power dissipation, thermal resistance, airflow, and sensor placement. Figure 6 shows the circuit schematic of this system.

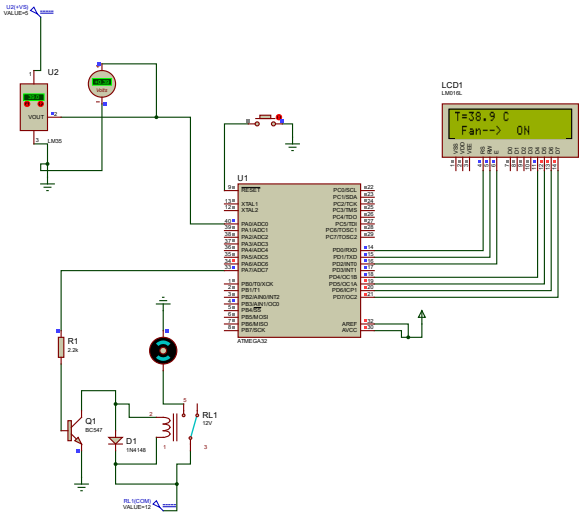


Figure 6 Automatic cooling system circuit diagram

Figure 7 shows the flowchart of the auto-climate system's performance across different temperature ranges.

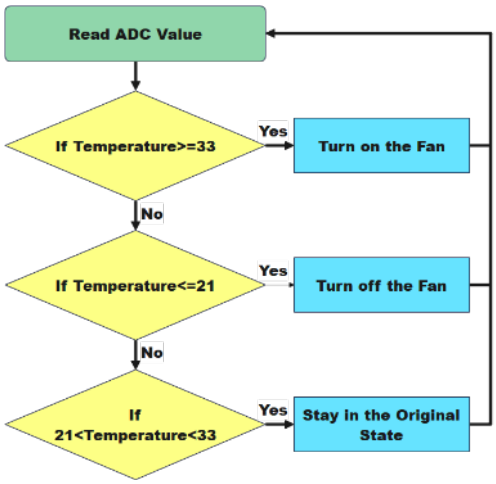


Figure 7 Flowchart of the auto-climate system's performance

3.4 Security System

This section focuses on enhancing the robot's security. The system requires the user to enter a password after powering up the circuit. As long as the user has not entered the password, the robot remains in its initial state and does not perform any actions. The password is a four digits code consisting of numbers 0 to 9, and the desired number is displayed on the LCD when the user presses a key.

The user is prompted to input a four-digit code. If the entered digits match the predefined code, the system deems the input correct and notifies the user via a buzzer and flashing lights, subsequently activating the robot. If the entered digits do not match the predefined code (indicating an incorrect password), the system will prompt the user to re-enter the password. If the wrong code is entered again, the alarm system is triggered, accompanied by flashing lights and an audible alarm. Afterward, the system locks for 30 seconds, after which the user can attempt to enter the password again.

Since the functionalities of the robot, particularly the security system, are programmed on the ATmega-32 microcontroller, and the ATmega-8 microcontroller manages the line-following system, it is crucial to establish communication between these two microcontrollers. This connection ensures that the robot does not move unless the correct password is entered. To achieve this, a base for the ATmega-32 microcontroller is utilized as the output, and a two-stage circuit is employed. The first stage includes an AND gate and a switch, while the second stage incorporates a relay, transistor, diode, and a snubber circuit, connecting to the Vcc pin of the ATmega-8 microcontroller for power and activation. Once the correct password is entered, the ATmega-32 microcontroller is fully activated, while the ATmega-8 microcontroller remains in standby mode, awaiting user input. A 7408 IC, which contains four two-input AND gates, is used for this purpose.

Figure 8 shows the circuit schematic of the security system. In this figure, the password system is activated using the ATmega-32 microcontroller. By entering the correct programmed password through the keyboard, the robot is activated. LED D1 indicates the activation of the second microcontroller on which the robot's operation is programmed.

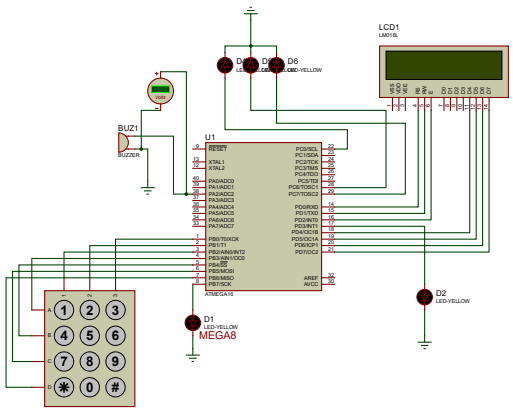


Figure 8 Circuit schematic of the robot security system

4 Mechanical Design

This robot is built with a two-layer design and is equipped with three wheels: a caster wheel at the front and two rear wheels with gearboxes. The wheels and sensor board are located beneath the first layer. For navigation, the robot uses five TCRT5000 sensors strategically placed at the front. These sensors are evenly spaced, with three located in the center and two at the far right and left edges. This configuration enables the three central sensors to guide the robot along straight paths and during turns of less than 90 degrees, while the two outer sensors handle navigation paths through 90-degree turns.

The first layer houses the main board, ventilation system, and light board. As the robot is designed for automated transport of tools within a factory setting, it may occasionally operate in brightly lit areas. To protect the essential electronic components from intense sunlight, they are shielded between the two layers. This section also contains the ventilation system, which automatically cools the board when the robot operates in warmer conditions. The upper layer serves as the control interface, featuring a numeric keypad for code entry, a system initiation button, a character LCD for displaying information, and a photoelectric sensor for detecting ambient light. These components provide the user with full control and feedback during operation. For the front lights, 12-volt LEDs are used to provide enhanced brightness, while 5-volt red LEDs are employed for the rear lights. Additionally, indicator lights are integrated at both the front and rear, signaling the accuracy of the password entry.

The mechanical structure of the robot employs a modular multi-tier design to optimize space and functional isolation. As illustrated in Figure 9, the hardware is organized into three distinct operational levels: the undercarriage, the primary functional layer (first floor), and the user-interface layer (second floor).

- **Undercarriage and Navigation Base:** The locomotion system follows a differential drive configuration, comprising two rear wheels driven by high-torque gearboxes and a front omnidirectional caster wheel (idle wheel) to ensure a zero-turn radius capability. The navigation array, consisting of five TCRT5000 infrared sensors, is mounted on a dedicated sensor board at the leading edge. The strategic spacing of these sensors—three central units for high-resolution path tracking and two peripheral units for detecting 90° junctions—ensures robust navigation in complex industrial layouts.
- **Primary Functional Layer (First Floor):** This internal compartment houses the “Main Board” and the active thermal management system. Placing the core electronics and the “Ventilation Fan” between the layers provides a dual advantage: it protects sensitive components from environmental hazards (such as dust and direct sunlight in factory settings) and facilitates a localized cooling tunnel. The “Headlight” and “Backlight” LED arrays are also integrated at this level to provide path illumination and status signaling.
- **User-Interface and Control Layer (Second Floor):** The uppermost tier serves as the Human-Machine Interface (HMI). It features a numeric keypad for secure access, a “Start Switch” for system ignition, and a character LCD for real-time telemetry (displaying temperature and system status). The “LDR Sensor” is positioned on this top surface to ensure unobstructed measurement of ambient light intensity.

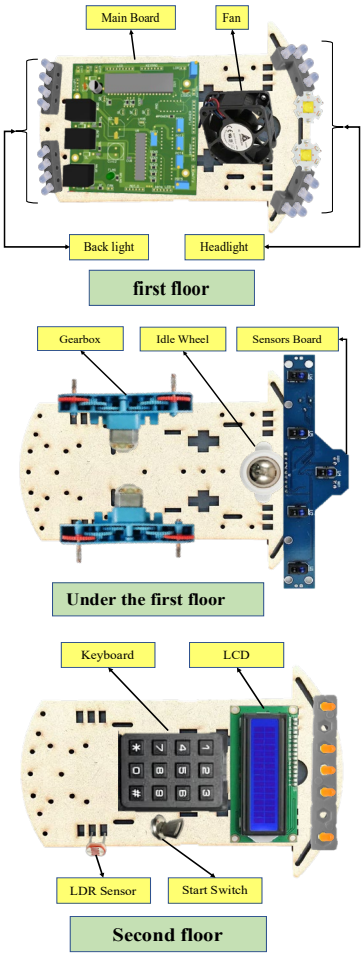


Figure 9 Arrangement of the robot’s components across each layer

5 Test and Results

5.1 Line Following Results

To validate the navigation capability of the proposed robot, the line-following system was experimentally evaluated using the five-sensor array mounted at the front side of the platform.

The sensors were indexed from left to right as (S1 to S5) Based on the detected line position, the controller generated differential PWM commands for the left and right motors, where the PWM duty cycle ranged from 0 to 255.

The experimental results confirmed that the robot was able to distinguish between straight motion, moderate turns, sharp turns, and 90° corners using predefined sensor activation

patterns. When only the center sensor (S3) detected the line, the robot moved straight ahead at maximum speed. In contrast, activation of the outermost sensors (S1 or S5) indicated a 90°turn, and the controller reduced one motor speed to zero while driving the opposite motor at a moderate PWM value to produce a pivoting motion. Intermediate sensor combinations were used to implement gradual steering corrections for turns smaller than 45° approximately 45° and greater than 45°.

This rule-based strategy provided a simple and reliable mapping between sensor states and motor commands. In particular, it enabled smooth path correction on curved segments while maintaining accurate cornering performance at sharp junctions. In addition, if all sensors were inactive or all sensors were simultaneously active, the controller interpreted this condition as either loss of the path or arrival at the terminal mark, and both motors were stopped for safe operation.

Table 2 summarizes the experimental decision rules used by the navigation controller. The selected PWM values were chosen to provide a gradual steering response according to the detected curvature of the path.

Table 2 Sensor-based navigation logic and suggested PWM commands

Case	Active Sensor Pattern	Path Interpretation	Left Motor PWM	Right Motor PWM	Robot Action
1	S3	Straight path	255	255	Move forward at maximum speed
2	S1	90°left turn	0	127	Perform a sharp left pivot
3	S5	90° right turn	127	0	Perform a sharp right pivot
4	S2+S3	Left curve < 45°	191	255	Gentle left correction
5	S3+S4	Right curve < 45°	255	191	Gentle right correction
6	S1+S2	Left curve, > 45°	64	191	Sharp left correction
7	S4+S5	Right curve > 45°	191	64	Sharp right correction
8	S2	Approximately 45°left turn	127	255	Moderate left turn
9	S4	Approximately 45° right turn	255	127	Moderate right turn
10	All sensors OFF or all sensors ON	White background or end of path	0	0	Stop

For straight motion, both motors operated at the maximum PWM value of 255. For slight turns < 45° a small speed difference between the two motors was sufficient to maintain smooth tracking. For approximately > 45° the inner wheel speed was further reduced to improve turning accuracy. Finally, for 90°corners, one wheel was stopped completely while the other wheel continued rotating at a moderate speed, enabling accurate pivoting around the corner.

5.2 Auto-Light System Result

The automatic lighting subsystem was experimentally evaluated under different ambient illumination levels in order to quantify the response of the LDR-based control unit. The purpose of this test was to verify that the robot can reliably detect low-light conditions and activate its lighting module without oscillation or false switching near the decision threshold.

The auto-light controller was implemented using an analog light-dependent resistor (LDR) connected to the microcontroller ADC. To improve switching stability, two distinct thresholds were adopted in the control logic. The headlights were activated when the sensed illumination dropped below the lower threshold, and they were deactivated only when the measured illumination exceeded the upper threshold. This hysteresis-based strategy prevented repeated ON/OFF transitions in marginal lighting conditions.

Experimental measurements were recorded under five representative lighting conditions, ranging from bright indoor illumination to dim industrial environments. For each condition, the LDR output voltage, ADC reading, lamp state, and system response time were measured. The results confirmed that the robot responded consistently to ambient light variations and maintained stable operation around the switching region.

As presented in Table 3, the lighting system exhibited a clear transition between illuminated and dim conditions. In the implemented logic, the lamp activation threshold was set at an ADC value below approximately 850, whereas lamp deactivation occurred only after the ADC value rose above approximately 880. This hysteresis band improved control robustness by eliminating chattering near the decision boundary. The measured response time remained below 50 ms in all tested cases, demonstrating that the lighting system reacts rapidly to environmental changes. Furthermore, the observed separation between the ON and OFF regions confirms that the proposed design is suitable for practical deployment in industrial corridors and low-visibility areas.

Table 3 Quantitative evaluation of the automatic lighting system

Test Condition	Ambient Light	LDR Output Voltage (V)	ADC Reading	Headlight State	Response Time (ms)
Bright environment	720	4.46	912	OFF	42
Normal indoor light	510	4.28	876	OFF	44
Threshold transition zone	470	4.18	855	OFF	45
Dim environment	430	4.10	838	ON	47
Very low-light environment	180	3.62	741	ON	49

5.3 Automatic cooling subsystem Result

The automatic cooling subsystem was quantitatively evaluated to verify its thermal response under elevated ambient

temperature conditions. The main objective of this test was to assess whether the robot could autonomously activate the cooling fan at high temperature, maintain safe thermal operation during prolonged use, and deactivate the fan only after the temperature returned to the predefined safe range.

The cooling controller was designed using a threshold-based strategy with thermal hysteresis. According to the implemented logic, the fan was switched ON when the measured temperature exceeded 33˚ and remained active until the temperature dropped to 21˚ This hysteresis band was introduced to avoid rapid repeated switching of the fan near the threshold temperature and to ensure stable thermal regulation.

To evaluate the performance of the proposed cooling mechanism, the robot was tested under several thermal conditions. For each test, the ambient or internal temperature was recorded together with the fan state, fan activation delay, and the time required for the system to return to the safe temperature range. The experimental results confirmed that the thermal management unit responded reliably and maintained stable operation without undesired oscillation.

The results presented in Table 4 confirm the proper operation of the proposed automatic cooling subsystem under different thermal conditions. As the measured temperature increased, the output voltage of the LM35 sensor rose proportionally according to its linear transfer characteristic, reaching approximately 0.33 V at 33°C, which corresponds to the fan activation threshold. Once activated, the cooling fan remained in operation until the temperature decreased to 21°C, where the sensor output dropped to approximately 0.21 V and the fan was turned off. This behavior verifies the correct implementation of the hysteresis-based thermal control logic.

Table 4 Quantitative Evaluation of the Automatic Cooling System

Test Condition	Measured Temperature (°C)	LM35 Output Voltage (V)	Fan State	Activation/Deactivation Response Time (s)	Cooling Status
Normal thermal condition	20	0.20	off	0	Safe, no cooling required
Lower safe boundary	21	0.21	off	1.0	Fan deactivated
Moderate temperature	27	0.27	off	0.0	Safe operating region
Upper threshold	33	0.33	ON	1.2	Fan activated
High-temperature condition	36	0.36	ON	1.1	Active cooling

5.4 Hardware Implementation and PCB Realization

Figure 10 presents the complete hardware schematic of the proposed robot, designed and integrated in Altium Designer. The circuit combines the main control unit, sensing interfaces, actuator drivers, user-interface modules, and regulated power distribution within a single embedded architecture. The primary microcontroller manages the keypad, LCD, and external control signals, while the secondary microcontroller processes the analog sensor inputs and handles auxiliary functions such as indication and local control.

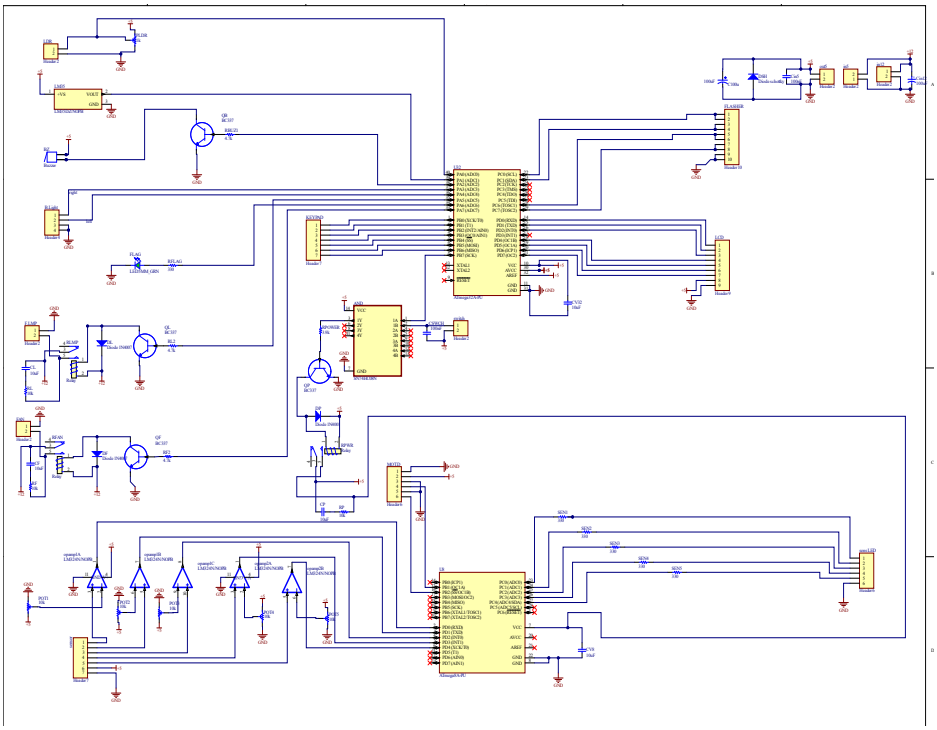


Figure 10 General schematic of the robot circuit in Altium Designer.

The sensing section includes the LDR-based light detector and the LM35 temperature sensor, both of which are connected through signal-conditioning stages to ensure stable and accurate readings. The actuator section incorporates motor-driving circuitry and transistor-based relay drivers for controlling the robot motion, fan, and other external loads. Flyback diodes are used across the relay coils to suppress inductive voltage spikes and improve circuit reliability. In addition, the schematic includes programming headers, status indicators, and power regulation networks to support reliable operation of the complete prototype.

Figure 11 shows the fabricated two-layer PCB of the proposed robot controller. The second layer is used as a continuous ground plane to provide a low-impedance return path and to reduce the probability of ground-loop formation across different sections of the circuit. This grounding strategy improves noise immunity and helps maintain stable operation of the sensing, control, and actuation blocks, particularly in a mixed-signal embedded system.

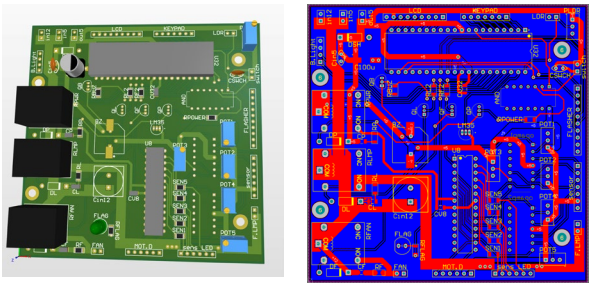


Figure 11 Two-layer PCB of the proposed robot controller with a continuous ground plane, five multi-turn potentiometers for TCRT5000 sensor calibration, and external front-mounted sensor interfacing.

The board also includes five multi-turn potentiometers, labeled POT1 to POT5, which are used to adjust the sensitivity and threshold levels of the TCRT5000 line sensors. These reflective sensors are mounted externally at the front of the robot rather than on the PCB itself, allowing them to be positioned more appropriately for line detection. The dedicated sensor header on the board provides the electrical interface for these external sensors and supports individual calibration of each sensing channel.

In addition to the sensor adjustment network, the PCB integrates the LCD and keypad interfaces, motor and relay driver sections, power distribution paths, and auxiliary modules such as the temperature-sensing and lighting-control circuits. Overall, the layout reflects a practical two-layer embedded design in which the ground plane, sensor calibration elements, and modular connectors are arranged to support reliable autonomous operation.

Figure 12 shows the fabricated prototype robot, which was designed in a two-level configuration to improve accessibility, modularity, and functional separation among subsystems. The upper deck carries the keypad and LCD module so that the user can easily interact with the robot and monitor its operating status without direct access to the internal electronics. A light-sensing element is also mounted on the top section to measure ambient illumination more effectively and with less obstruction from other components.

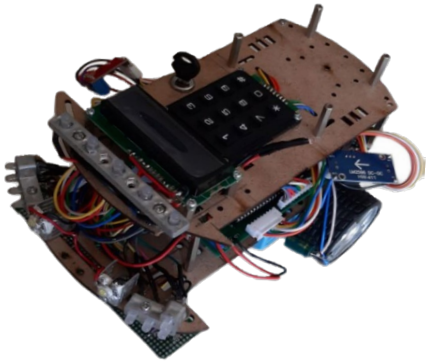


Figure 12 Overview of the line follower robot

The middle layer contains the main electronic board, wiring, power-conditioning circuitry, and the fan assembly. This arrangement allows the control electronics and auxiliary modules to be integrated in a compact structure while maintaining mechanical stability. In addition, the TCRT5000 reflective sensors are installed underneath the front section of the robot, close to the floor surface, so that the system can detect the line path with higher accuracy during motion.

Overall, the two-level mechanical design separates the user interface, sensing, and control subsystems into distinct physical layers, which improves assembly convenience, maintenance, and the reliability of line-following operation.

Figure 13 shows the experimental line-following course used to evaluate the robot's navigation capability under more difficult path conditions. The course is designed on a white background with a continuous black track, which improves contrast for the reflective sensors and allows reliable detection of the path. The route includes several challenging geometric features, such as a junction, a vertical branch, a U-turn, sharp directional changes, a cross-line, and a curved loop, all of which are useful for testing the robot's tracking accuracy and cornering performance.

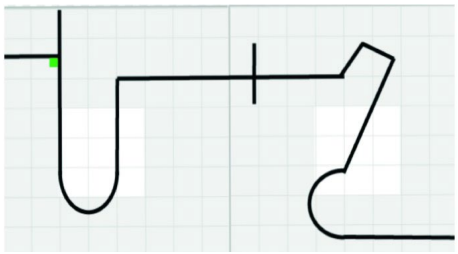


Figure 13 An example of the path defined for the robot

A continuous black line is also included as a stop boundary to define the end of the test path and to assess the robot's stopping response after completing the route. This test pattern provides a practical benchmark for verifying the robot's ability to follow the line accurately and maintain stable motion in the presence of complex track geometry.

6 Conclusion

The development of a line-following robot for transporting supplies and tools within factory environments has been successfully demonstrated. The robot was designed to operate reliably under various conditions, including low-light settings and areas with fluctuating temperatures, made possible by its integrated auto-light and automatic ventilation systems. These features ensured that environmental factors did not hinder the robot's operation. A security system was also incorporated, restricting access to authorized personnel and preventing unauthorized use or interference, particularly in public or shared spaces. The robot's design, which included five strategically placed TCRT5000 sensors at the front and beneath the chassis, allowed for precise line detection and navigation of complex paths, including sharp 90-degree turns, with remarkable accuracy and stability. The consistent performance of the sensors established the robot as an effective solution for automating the transport of supplies, enhancing efficiency, and maintaining stability within factory settings.

For future work, acknowledging the potential deployment of numerous robots and vehicles within manufacturing environments, the implementation of an adaptive cruise control system is advisable to mitigate the risk of collisions during operation. Such a system would allow the robot to maintain an appropriate speed and safe distance from nearby vehicles, thereby avoiding potential collisions. Furthermore, it is imperative to conceptualize a framework that facilitates the remote control of the robot. Another feature worth exploring is the integration of a smart voice assistant, which would allow users to issue vocal commands that the robot could execute, enhancing usability and functionality.

Disclosure of Potential Conflicts of Interest

The Authors declare that there is no conflict of interest

Reference

[1] C. T. A. Wise, S. Fouziya Sulthana, and A. Sundaramoorthy, "Line follower robot based room service automation in hospital environment," in *AIP Conference Proceedings*, AIP Publishing, 2022.

[2] L. M. T. Dalmazo *et al.*, "Analysis of Dynamic PD Control Performance Compared to Standard PD in Track Follower Vehicle," in *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, IEEE, 2023, pp. 591–596.

[3] J. S. J. C. C. Amorim, J. H. O. Fernandes, A. L. C. Canella, T. M. B. Santos, J. Lima, and M. F. Pinto, "Control Tunning Approach and Digital Filter Application for Competitive Line Follower Robot," in *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, 2023, pp. 597–602. doi: 10.1109/LARS/SBR/WRE59448.2023.10332987.

[4] M. A., P. K., A. k. S, and N. N., "Design and Development of Automated Guided Vehicle with Line Follower Concept using IR," in *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2023, pp. 1–11. doi: 10.1109/ICECCT56650.2023.10179618.

[5] S. Shirmohammadi and F. Baghbani, *Design and Implementation of a Line Follower Robot*. 2024. doi: 10.1109/QICAR61538.2024.10496637.

[6] M. A. Rahman, P. Chakroborty, and M. F. Ali, "Design and Implementation of a Low-Cost Arduino Based Line Follower Gardener Robot," in *2022 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET)*, 2022, pp. 1–4. doi: 10.1109/ICRPSET57982.2022.10188540.

[7] R. R. Chekuri, V. Shivaram, K. Shravan, T. V Sai, M. C. Chinnaiah, and D. H. Krishna, "Design and Implementation of Line Follower Robot on SoC with SDK," in *2024 3rd International Conference for Innovation in Technology (INOCON)*, 2024, pp. 1–5. doi: 10.1109/INOCON60754.2024.10511509.

[8] S. Parihar and S. Dutta, "Enhancing Line Following Robot Navigating with Smart Assistant Capabilities," in *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2023, pp. 1–7. doi: 10.1109/ICSES60034.2023.10465582.

[9] W. Pavón, J. Campaña, P. Santamaria, C. Iza, and R. Zapata, "Intelligent Control System for Line Follower Robot: From Theory to Implementation on the Mechatronics Field," in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2023, pp. 1143–1148. doi: 10.1109/DASC/PiCom/CBDCom/Cy59711.2023.10361359.

[10] S. Tayal, H. P. G. Rao, S. Bhardwaj, and H. Aggarwal, "Line Follower Robot: Design and Hardware Application," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 10–13. doi: 10.1109/ICRITO48877.2020.9197968.

[11] M. Yuliza, F. Wirda, and P. Ardila, "Microcontroller-Based Line Follower Robot for Automated Agricultural Goods Transportation System," *Acitya Wisesa J. Multidiscip. Res.*, pp. 86–109, 2025.

[12] H. E. Hafdaoui, *Energy Futures: ICEF 2026*. in Materials Research Proceedings Series. Materials Research Forum LLC, 2026. [Online]. Available: https://books.google.com/books?id=a-7UEQAAQBAJ

[13] S. G. Bawane, B. Dawade, T. Kale, and D. Baghele, "Design of Automated Guided Vehicle for Exam Control Room," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 14, no. 4, pp. 4092–4097, 2026, doi: 10.22214/ijraset.2026.79934.

[14] L. R. V, V. M., S. Bhat, S. K. Panda, and M. G. Asuti, "Line Following and Obstacle Avoidance Robot Using ESP-32," in *2025 International Conference on Emerging Technologies in Electronics and Green Energy (ICETEG)*, 2025, pp. 1–6. doi: 10.1109/ICETEG66194.2025.11473188.

[15] H. A. Neamah and L. Xiaodong, "Development of an Advanced High-Speed Line-Following Robot for Enhancing Autonomous Robotics," in *2025 IEEE 12th International Conference on E-Learning in Industrial Electronics (ICELIE)*, 2025, pp. 1–6. doi: 10.1109/ICELIE64733.2025.11244757.