IJRCS
International Journal of Robotics and Control Systems
Vol. 4, No. 3, 2024, pp. 1353-1381
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs
ASCEE

Arduino-Controlled Multi-Function Robot with Bluetooth and nRF24L01+ Communication

Faysal Ahmmed [a,1], Asef Rahman [a,2], Amirul Islam [b,3,*], Ajmy Alaly [a,4], Samanta Mehnaj [a,5], Prottoy Saha [a,6], Tamim Hossain [b,7]

[a] Dept. of Computer Science and Engineering, American International University-Bangladesh, Dhaka, Bangladesh
[b] Dept. of Electrical and Electronic Engineering, American International University-Bangladesh, Dhaka, Bangladesh
[1] 22-47069-1@student.aiub.edu; [2] 22-47106-1@student.aiub.edu; [3] amirul@aiub.edu; [4] 22-46733-1@student.aiub.edu; [5] 22-47081-1@student.aiub.edu; [6] 21-45640-3@student.aiub.edu; [7] tamim@aiub.edu
* Corresponding author

ARTICLE INFO

Article history
Received July 03, 2024
Revised August 06, 2024
Accepted August 13, 2024

Keywords
Robotics;
Robotic Arm;
Arduino;
nRF24L01+;
mpu6050;
L298 Motor Driver;
Flex Sensor;
Wireless Control;
Gesture Control;
Obstacle Avoidance;
Line Following;
Object Manipulation;
Autonomous Systems;
Sustainable Development Goals

ABSTRACT

This paper outlines the design and development of an advanced robotic system that integrates hardware implementation with theoretical simulation to address the need for versatile and user-friendly robotic solutions in various environments. Addressing the issue of limited adaptability in existing robotic systems, we propose a wireless, voice and gesture-controlled robot car with an integrated robotic arm capable of performing complex tasks such as line following, obstacle avoidance, object manipulation, and autonomous navigation over one-kilometer range. To improve operational efficiency and user involvement, this paper designs a multifunctional robotic platform that integrates user-friendly control interfaces with inexpensive, state-of-the-art sensor technologies. To achieve this, we integrate a variety of sensors, including ultrasonic sensors for precise distance measurement, infrared sensors for object detection and line following, an L298 motor driver for controlling geared motors, servo motors for controlling robotic arms, a flex sensor for claw control, and an mpu6050 accelerometer for gesture recognition. The system also uses a custom-made Bluetooth app for remote control, nRF24L01+ for long-range wireless control, and Arduino Mega and Nano for processing and control functions. The results demonstrate the robot functions well in dynamic conditions, and it can be used in hospitals to assist healthcare professionals, in restaurants for food delivery, and in industrial settings for object manipulation. The system's design proves robust in real-world scenarios, offering significant improvements in accessibility and operational efficiency. This study aligns with Sustainable Development Goals (SDGs) 3 (Good Health and Well-being), 9 (Industry, Innovation, and Infrastructure), and 17 (Partnerships for the Goals). The robotic arm's potential application in healthcare settings advances SDG 3, its contribution to industrial productivity advances SDG 9, and collaborations with tech companies to expand and improve the robot's capabilities promote SDG 17.

1. Introduction

Customized robotic systems are often designed for specific applications and environments, limiting their ability to adapt to a wide range of challenges. However, as automation and robotics technologies become more significant across industries, there is a need to design flexible robotic systems that can operate in a variety of environments. We aim to build a multipurpose robot car endowed with a robotic arm to meet diverse needs. The primary objective of this study is to design and develop a comprehensive system that embodies enhanced efficiency, safety, and accessibility of robotic systems. This system will be characterized by its multifaceted applications and seamless integration of cutting-edge sensors with user-friendly control interfaces. This study builds upon previous advancements in robotics to expand the potential applications and address the evolving requirements of contemporary environments.

1.1. Literature Review

Several studies have shown that wireless and Bluetooth control systems are very useful for robotic and automation applications. These technologies improve user interaction and operational flexibility by enabling smooth and flexible control over robotic vehicles and home automation systems. For example, Bluetooth-based home automation systems with Arduino enable authorized users to control home functionalities such as automatic door motor, water pumping motor, home illumination, and smoke detection remotely within their homes [2]. Similarly, studies have shown that nRF24L01 wireless transceiver modules combined with Arduino can be used to make radio controller units that can be used for different things and provide reliable communication and control over long distances [5], [6]. The TI-based CC1101, CC24XX, CC25XX, or similar wireless modules can be seamlessly integrated with the Arduino Mega MCU to design a sensor system. In contrast, the Nordic Semiconductor's nRF24L01 wireless module offers advantages of lower cost and reduced power consumption compared to the CC1101 and Wi-Fi modules, as well as the CC24XX and CC25XX series. In addition, this module can give up-to 2 Mbps of data transfer speed within a long-distance range [3]. RF modules enable real-time communication and command execution, which is essential for applications ranging from industrial automation to robotics education [4], [8]. A recent study shows, C# based application can be used to control robot car from laptop or pc effectively. But the transmitter needs to be connected with pc via a USB wire [4]. In our system we will make it more efficient by using the laptops or pc's inbuild Bluetooth functionalities and making it a transmitter which can communicate with a Bluetooth receiver installed in the robot car. Such systems show the revolutionary potential of wireless technologies in advancing automation solutions across various sectors by giving users greater flexibility in monitoring and controlling robotic operations.

Significant progress in human-machine interaction and control methodologies can be seen in research on gesture, voice, and smartphone control for robotic systems with the help of Arduino as MCU, nRF24L01 as long-range wireless controller, Bluetooth module as short-range voice and phone controller, and mpu6050 as gesture controller [9]-[20]. A methodology was proposed to convert hand gestures into robotic hand movements, consisting of two subparts: a transmitter and a receiving section. The system's wireless controller, a robotic glove, includes an Arduino Nano, Flex sensors, and an RF Transmitter module. The Arduino Nano processes the Flex sensor outputs, converting the degree of finger bending into data. This data is then transmitted by the RF Transmitter module to the RF Receiver module in the robotic hand. The module continuously receives feedback from the hand and transmits new processed signals [61]. Robots can be simply controlled by hand movements thanks to gesture recognition systems, such as those that utilize accelerometers and gyroscopes [11], [13]. Another simple way to give natural language commands to robots is through voice control. Through mobile applications, smartphone-controlled robotic platforms offer users flexibility and remote accessibility while allowing them to control robots remotely with their smartphones [1], [17]. These control techniques not only improve user experience but also increase robotic systems' operational capabilities in a variety of contexts, such as industrial automation, home automation, and surveillance.

Robotic arm control is a major development in assistive technology that provides more independence and functionality for people with disabilities who are unable to walk. Numerous methods, including hand and finger gestures [21], smartphone interfaces [22], and accelerometer-based feedback systems [23], have been explored in research for controlling robotic arms. A wireless-controlled robotic hand motion system with flex sensors was proposed, featuring a glove as the transmitter and a 3D-printed forearm as the receiver. The glove is equipped with five flex sensors to capture hand motions. However, in our proposed system, we use only one flex sensor to control the robot claw movement, significantly reducing costs while maintaining core functionalities [62]. mpu6050 can be a great sensor for gesture movement. It can detect shifts in axes and give feedback according to the shifting direction. Some researchers made a 5-degree-of-freedom robot arm with a 5-mpu6050 sensor [23]. Our system will utilize one mpu6050 sensor and control the robotic arm along with the robot's movement direction, which will improve performance and also significantly cut costs of the system. These wirelessly controllable robotic arms make use of Arduino-based systems to enable users to effectively interact with their surroundings.

The various automation applications of Arduino with Bluetooth module and IoT are highlighted in the papers [24]-[30], [52]-[57]. They emphasize how it can be used in precision manufacturing, such as color sorting machines, and how it can improve productivity through industrial automation and the IoT along with agricultural field. Arduino provides smartphone and Wi-Fi connectivity for smart features in home automation, enhancing energy efficiency and convenience. All of these studies show how adaptable and successful Arduino is in manufacturing, industry, agriculture, and home automation applications.

To enhance security and ensure human safety, robots are utilized to avoid adversaries in India's defense sector. In nations like India, robots are deployed to prevent conflicts. Automated robotic systems are employed in regions such as Mumbai and Kashmir to prevent altercations and safeguard human lives. These robots use ultrasonic sensors to detect nearby objects by emitting ultrasonic waves. When the waves encounter an object, they reflect back and are detected by a sensor, which sends a signal to the Arduino. The ultrasonic sensor then searches for a direction without obstacles to move freely [7]. Obstacle avoidance using Arduino and ultrasonic sensors demonstrates effective strategies for integrating automatic obstacle avoidance in robotic systems [31]-[35]. It ensures safe navigation in time-varying and dynamic environments by utilizing ultrasonic sensors to detect obstacles. Unmanned vehicles and autonomous floor-cleaning robots are two examples of applications that highlight how flexible these systems are in terms of improving operational autonomy and safety. These advancements underscore the critical role of robust obstacle avoidance in facilitating reliable and independent robotic operations across diverse applications.

Research on Arduino line following shows how to implement object-following functionality in a way that improves usability in dynamic environments. These studies develop line follower robots with precise navigation capabilities using various strategies [36]-[39]. In an era of advancing automation, the development of a color line-following robot is essential for reducing human effort. This versatile robot can be used in airports to transport baggage, in homes for automation, and in restaurants as a robotic waiter, such as the Robot Restaurant in Porur, Chennai. It is also valuable in industries for transferring large machinery and can be adapted for use in mass transit systems, enhancing efficiency and convenience [39]. Hospital environments are one of the applications, where robots can autonomously follow predetermined paths, ensuring efficient delivery of items or assistance [36]. These advancements highlight the versatility of line following robots in addressing navigation challenges across different sectors, enhancing their utility in dynamic and demanding environments such as restaurants, warehouses, and hospitals.

Engineering and robotic science benefit significantly from the use of Arduino-based technology [40]-[44]. Arduino platforms can be integrated to improve learning outcomes through hands-on experimentation and problem-solving skills. Because of Arduino's adaptability, teachers can design interesting projects that cover everything from simple robotic mechanisms to complex smart embedded systems, preparing students for a future filled with technological challenges [45]-[47]. In addition, the development of robot cars with motor drivers shows how automatic controls can be taught practically while giving students real-world experience with system integration and motor speed regulation [48]-[50]. Additionally, the ability to create software interfaces specifically tailored

Vol. 4, No. 3, 2024, pp. 1353-1381

to robotic functionalities through custom application development for robot control opens up new educational opportunities. Also, it enhances the understanding of IoT applications in software engineering education [51]. Through useful, project-based learning strategies, Arduino plays a critical role in advancing STEM education.

The literature review highlights the significant impact of robotic technologies in assisting humans with various tasks such as industrial automation, home automation, and assistive technologies for individuals with disabilities. These studies underscore the importance of control mechanisms like Bluetooth, wireless modules, and gesture-based systems in enhancing user interaction and operational flexibility. However, a major limitation across these reviewed papers is the lack of a comprehensive control system that integrates multiple control mechanisms into a single platform. Most systems focus on either gesture control, Bluetooth, voice, or smartphone control independently. Some systems also implement voice and text control with custom-made app [4], but they lack gesture control. Our research aims to address this gap by integrating long-range wireless gesture, voice, Bluetooth, phone, and laptop control by integrating both nRF24L01+ and HC-06 Bluetooth modules along with a custom-made C# controller app. The control software system will provide real-time feedback and seamless operation. Additionally, existing systems have a limited wireless range of 100-200 meters using the nRF24L01 module [6], [7]. We plan to extend this range significantly by using an antenna-based nRF24L01+ module, thereby enhancing the system's operational capabilities.

1.2. Research Objectives

In this paper, we integrate all control mechanisms—wireless gesture, Bluetooth, voice, text command, mobile phone, and laptop control—into a single robot, which has not been done previously. Furthermore, we aim to extend the wireless control range beyond what has been achieved in prior research. The key objective of this paper is summarized below:

1. Develop a Versatile Robot Platform:

- Create a multipurpose robot car equipped with a movable robotic arm and cost-effective sensors. This system aims to efficiently perform functions such as, line following, object following, and obstacle avoidance
- Emphasize low-cost solutions while maintaining high functionality.

2. Implement multiple control processes:

- Design control interfaces that include voice commands, wireless gesture control, and mobile applications.
- Ensure that the robot car can be easily operated using any of these methods.

3. Create a user-friendly control interface:

- Develop an optimal interface for controlling the robot's functionalities from a laptop or PC using C#.NET 8.0.

1.3. Paper Structure

In this paper, we provide a comprehensive analysis of the planning, creation, and application of a multifunctional robot car equipped with a robotic arm. First, an overview of the goals and development of our solution is presented. We then explore the design and development process, including the integration of advanced sensor technologies, control systems, and autonomous features. This paper outlines specific objectives, including the integration of a robotic arm and the design of user-friendly Bluetooth communication interfaces and gesture control systems. We explain the functionality and operating principles of each sensor component employed within the system. Furthermore, this paper elaborates on the methods utilized for autonomous operation and the validation procedures implemented to guarantee system reliability. Additionally, a comparative analysis between theoretical simulations and real-world deployments is presented. This is done by using virtual monitor of proteus to get sensors working feedback in simulation and Serial monitor of

Arduino IDE to get hardware sensors working feedback. Finally, we examine the contributions of this study to the field of robotics and evaluate the project's alignment with Sustainable Development Goals and societal needs. An overview of the research methodology used in this paper is presented in Fig. 1.

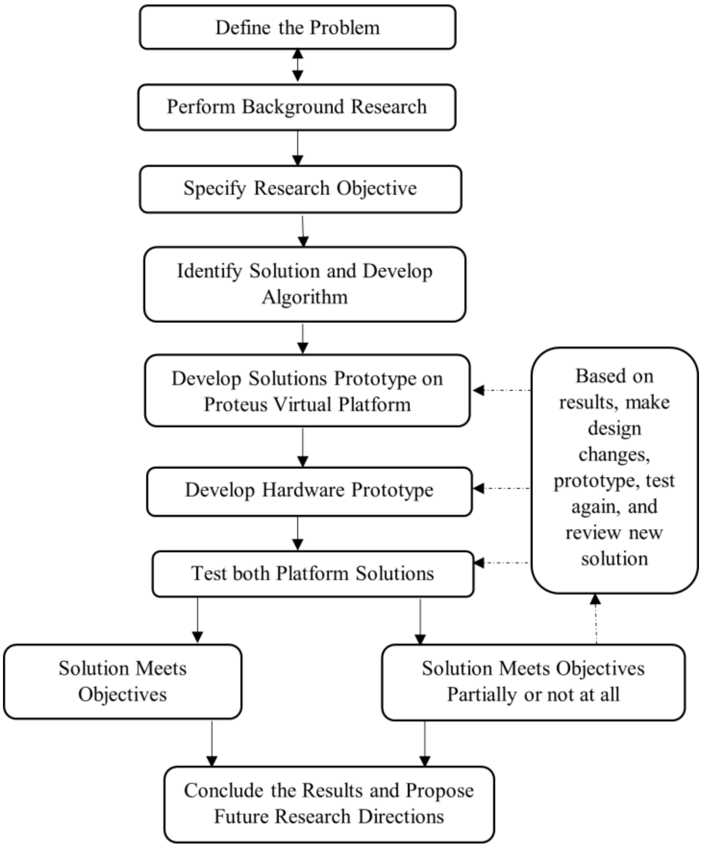


Fig. 1. Overview of the research methodology

2. Methodology and Modeling

This section outlines the systematic approach used to design and develop the multifunctional robotic system. Our methodology emphasizes modularity, user-friendliness, and the integration of advanced control technologies and algorithms to create a versatile robot capable of performing various tasks.

2.1. System Design

We began the design process by carefully examining the project specifications and defining key characteristics such as autonomy, wireless control, and object manipulation. Hardware components were selected based on affordability, performance, and compatibility, ensuring a modular system architecture that allows for seamless integration and future improvements. Our primary objective was to develop a user-friendly system that is easy to operate and maintain.

2.2. Working Principle

The proposed system operates by integrating various hardware components and technologies to enable the multifunctional robot car with a robotic arm. The Arduino Nano (Fig. 2) serves as the transmitter brain, using nRF24L01+ modules for wireless communication with the receiver unit. The receiver unit (Fig. 3) based on an Arduino Mega, processes incoming data and directs the robot's movements. The design process involves several steps:

- Integration of Components: Wiring the Arduino Mega and Nano boards with nRF24L01+ modules to enable wireless communication.

- Sensor Utilization: Using an ultrasonic sensor to measure object distances and an infrared sensor for object detection and following.
- Motor Control: Employing an L298 motor driver to precisely control the robot's wheels.
- Gesture Control: Utilizing an mpu6050 accelerometer for gesture control, allowing users to interact with the robot via hand gestures.
- Remote Control: Implementing a custom-made Bluetooth communication app and the HC-06 Bluetooth controller for remote operation.

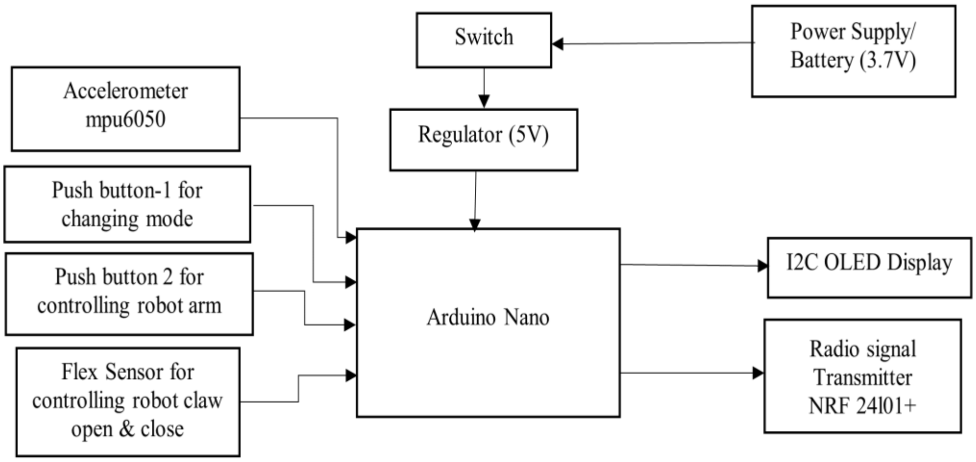


Fig. 2. Block diagram for the transmitter side

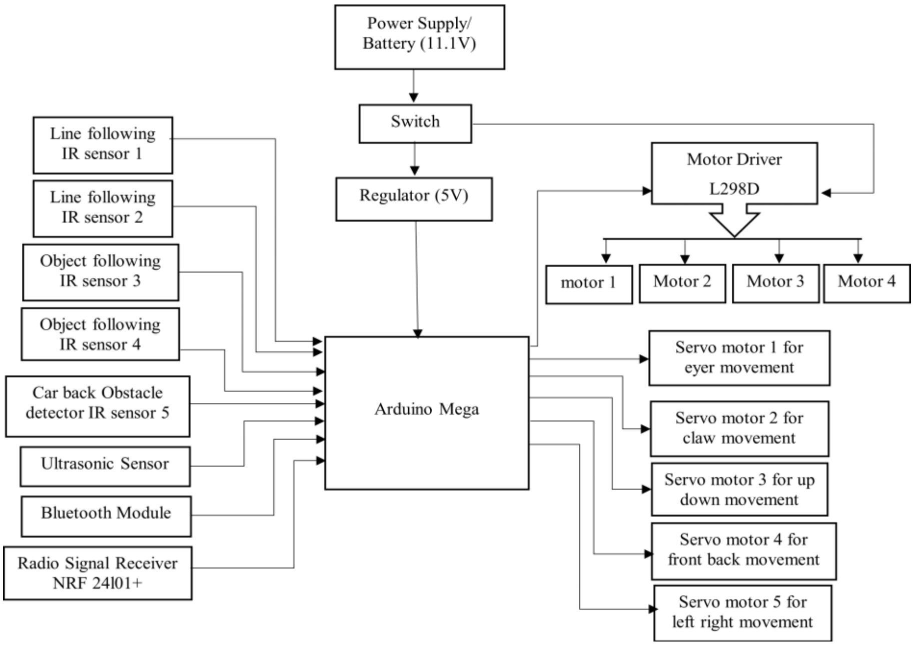


Fig. 3. Block diagram for receiver side

2.3. Description of Major Components

This section provides a comprehensive overview of the necessary components, including brief descriptions of their operating principles and the reasons for selecting them over other options.

A. Arduino Mega2560

The Arduino Mega serves as the brain of the robot car, employing AVR (Advanced Virtual RISC) technology with an 8-bit processor running at 16 MHz. It accepts an input voltage ranging from 7-12V. Communication interfaces on the Arduino Mega include four UARTs (universal asynchronous receiver-transmitter), I^2C (Inter-Integrated Circuit)/ TWI (Two-Wire Interface), SPI (Serial peripheral interface), and USB (Universal Serial Bus) [59]. The Arduino Mega is ideal for projects requiring multiple I/O operations, such as robotic control, home automation, and complex sensor networks. Its flexibility, ease of use, low-cost, and robust community support make it an excellent choice for our robotic system.

B. Arduino Nano

The Arduino Nano acts as the transmitter unit, utilizing the same AVR technology as the Mega but in a compact form factor. It has an ATmega328P microcontroller, with a clock speed of 16 MHz and an operating voltage of 5V. Key features include 32 KB of flash memory, 2 KB of SRAM, 1 KB of EEPROM, and I^2C, SPI, and UART communication interfaces, along with a mini-USB programming interface [72]. Its small size and sufficient computational power make it suitable for operating a mpu6050 accelerometer and nRF24L01+ module. The Arduino Nano is also cost-effective, making it ideal for wearable transmitter on the wrist due to its compactness and affordability.

C. nRF24L01+ Transceiver Module

The nRF24L01+ module is a 2.4 GHz RF transceiver designed for ultra-low power wireless applications. We used Power Amplifier (PA) and Low Noice Amplifier (LNA) enabled version for maximum range. It is configurable through a Serial Peripheral Interface (SPI). It offers a configurable air data rate up to 2 Mbps. It supports up to 125 channel frequencies and six pipelines per node, making it suitable for robust, low-cost wireless communication. [5] Compared to modules like LoRa, which offer longer range but lower data rates, the nRF24L01+ provides a balanced solution with high-speed data transfer and sufficient range at a lower cost, ideal for our robotic system where both range and speed are critical.

D. HC-06 Bluetooth Controller

The HC-06 Bluetooth Controller is a compact and cost-effective module designed for wireless communication between microcontrollers and Bluetooth-enabled devices. Operating within the 2.4GHz ISM band, it supports UART communication with a default baud rate of 9600, which is configurable and can operate at a voltage range of 3.36V to 6V [70]. Its ease of integration, combined with its low power consumption and broad compatibility with smartphones, tablets, and other Bluetooth devices, makes it an ideal choice for remote control applications, enhancing the flexibility and convenience of robot operation.

E. mpu-6050 Accelerometer

The mpu6050 accelerometer integrates a 3-axis gyroscope and a 3-axis accelerometer, enabling precise motion tracking and orientation sensing with high sensitivity and accuracy. The mpu6050 is equipped with a 16-bit analog-to-digital converter (ADC) to accurately track three-dimensional motions [67], [68]. Communication is facilitated via I^2C or optional SPI, with configurable I^2C addresses of 0x68 or 0x69. The mpu6050's compact 4mm x 4mm x 0.9mm package makes it ideal for use in our robot car transmitter system as a gesture recognizer, motion tracker, and wearable [71]. Its robust features and low power consumption make it a versatile and efficient choice for our gesture control application.

F. HC-SR04 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor utilizes ultrasonic waves to measure distances accurately from 2 cm to 400 cm, with an accuracy of ±3, making it ideal for obstacle detection and avoidance in robotics. Operating at 5V DC with a typical current consumption of 15 mA, it sends a 10 µs trigger pulse and emits an 8-cycle burst of 40 kHz ultrasonic waves [65]. The sensor then detects the reflected waves and outputs a pulse on the "Echo" pin, proportional to the distance of the object. To calculate the distance using the pulse width, the formula (1) is used:

The sensor's compact size (45 mm x 20 mm x 15 mm) and lightweight design make it easy to integrate into various projects. Its simple interface with the Arduino Mega microcontroller, high accuracy [65], and affordability makes the HC-SR04 a versatile and reliable choice for our robot car for detecting objects and measuring object distances.

$$Distance(cm) = \frac{Pulse\ Width\ (\mu S)}{58.2} \tag{1}$$

G. HW201 Infrared Sensor

The HW201 IR sensor detects and measures infrared radiation, comprising an emitting and receiving tube [64]. This IR sensor is ideal for our robot car application as it is compact, cheap, and easy to connect with an Arduino Mega microcontroller.

H. L298 Motor Driver

The L298 motor driver is a dual H-bridge motor control IC. An H-Bridge is an electronic circuit that allows voltage to be applied across a load in either direction, enabling complete control over a DC motor. This setup permits electronic management of the motor's forward, reverse, brake, and coast functions through a microcontroller, logic chip, or remote control [66]. It operates with a power supply voltage of up to 50V. The input and enable voltage range from -0.3V to 7V. The driver supports 2A for continuous DC operation. The L298 operates within a junction temperature range of -25°C to 130°C [73]. These specifications ensure efficient and reliable motor operation, making the L298 suitable for our robot car application and ensuring optimal performance even in challenging environmental conditions.

I. SG90 Servo Motor

The SG90 servo motor is a small and lightweight motor offering precise control over angular motion, making it ideal for tasks such as robotic arm movement and sensor orientation in robotic systems. It delivers a stall torque of 2.5 kg/cm [63]. It comes with a Futaba/JR 0.1 pitch connector on a 9-inch cable, making it compatible and easy to integrate into our robotic arm and also for controlling ultrasonic sensor movement for measuring object distance from various angles.

J. Flex Sensor

The flex sensor is a highly sensitive component used in robotic systems to detect bending or flexing movements, allowing for intuitive control mechanisms such as finger gestures. The output voltage (Vo) is determined by the formula Vo = VCC × (R1 / (R2 + R1)), where R1 represents the variable resistance of the flex sensor and R2 represents the constant resistance used in between ground and flex sensor pin [74]. Its flexible construction and high sensitivity make it ideal for controlling our robots' claw open and close movement.

K. SSD1306 White 0.91″ 128×32 OLED Display – I2C Interface

The SSD1306 White 0.91" 128×32 OLED Display features a high-resolution 128×32 pixel matrix and utilizes OLED technology for self-luminous operation without the need for a backlight, resulting in lower energy consumption. The OLED display module uses an I^2C connection interface. It offers a viewing angle exceeding 160^0 degrees and operates at a voltage range of 3.3 to 6V. Its dimensions are 12mm by 12mm by 38mm [69]. Low cost, low power consumption, and compact size are the reasons for using this as a mode indicator in our robot's transmitter.

2.4. Control Modes

The schematics (Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8) illustrate the system architecture and operational flow, helping to understand the interactions between different components and control mechanisms. The processes for each distinct mode of the receiver are illustrated separately in Fig. 6, Fig. 7, Fig. 8 for better clarity. A detailed explanation of these diagrams is provided below.

Transmitter Flowchart (Fig. 4):

- Initialize sensors.
- Gather data from sensors and selected modes.

- Transmit data to the robot car via the NRF transmitter.
- Update the display value for the selected mode.
- This cycle repeats continuously, enabling real-time control of the robot's movements and functions.

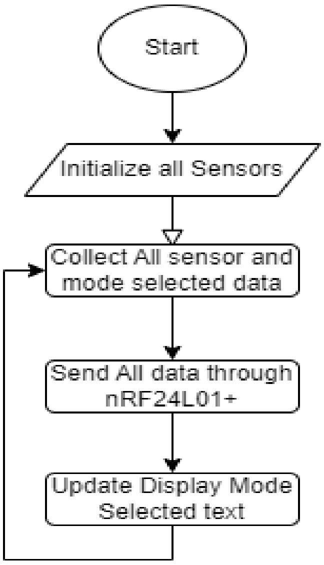


Fig. 4. Transmitter flowchart

Receiver Flowchart (Fig. 5):

- Initialize sensors.
- Determine control mode (Bluetooth or voice).
- If Bluetooth mode is selected:
  - Execute corresponding command if data is received.
  - Stop the vehicle if no data is received.
- If voice mode is selected:
  - Wait for incoming data.
  - Execute corresponding command if data is received.
  - Stop the vehicle if no data is received.
- If neither Bluetooth nor voice mode is selected, check for data from the NRF module.
  - If NRF data is received:
    - Execute the corresponding mode: (Mode 1: Claw control, Mode 2: Gesture control, Mode 3: Line following, Mode 4: Obstacle avoidance, Mode 5: Obstacle following.
  - If no NRF data is received, stop the car.

Claw Control (Fig. 6 (a)):

- Check if new data is coming from the receiver.
- If data is confirmed, update positions of servo motors based on the selected mode.
  - If mode equals 0:
    - Update the positions of the left, right, and open/close servo motors.

- If the mode is not 0:
    - Adjust the positions of the up-and-down servo motor.
- Stop all servo motors if no data is received to prevent unintended movement.

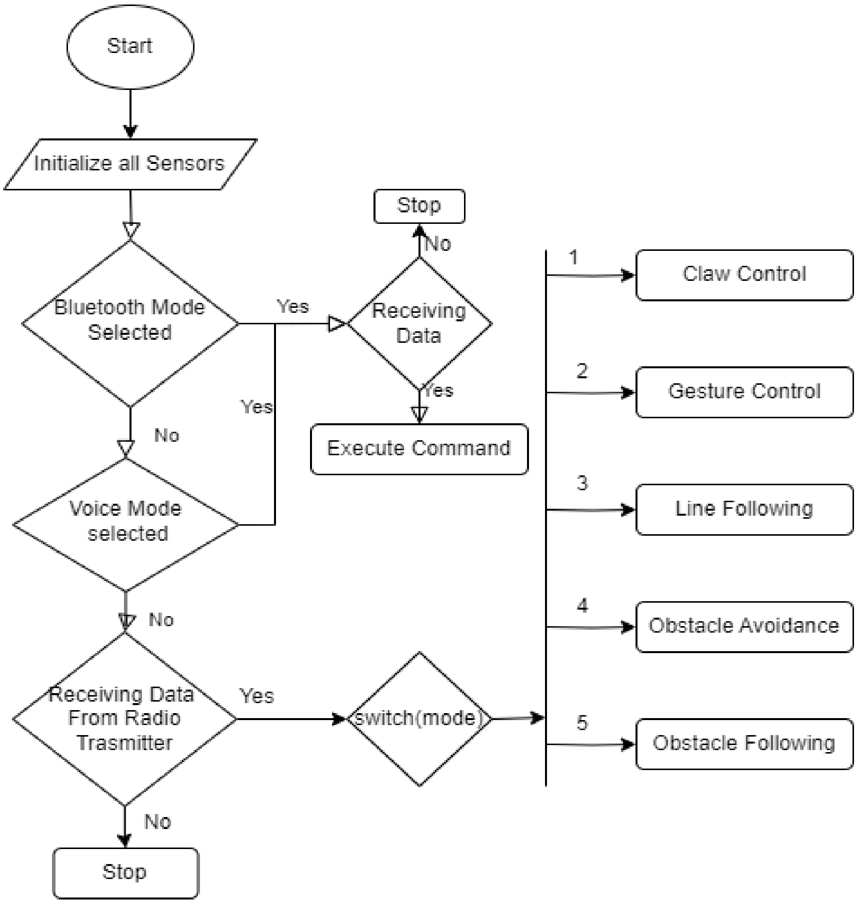


Fig. 5. Receiver flowchart

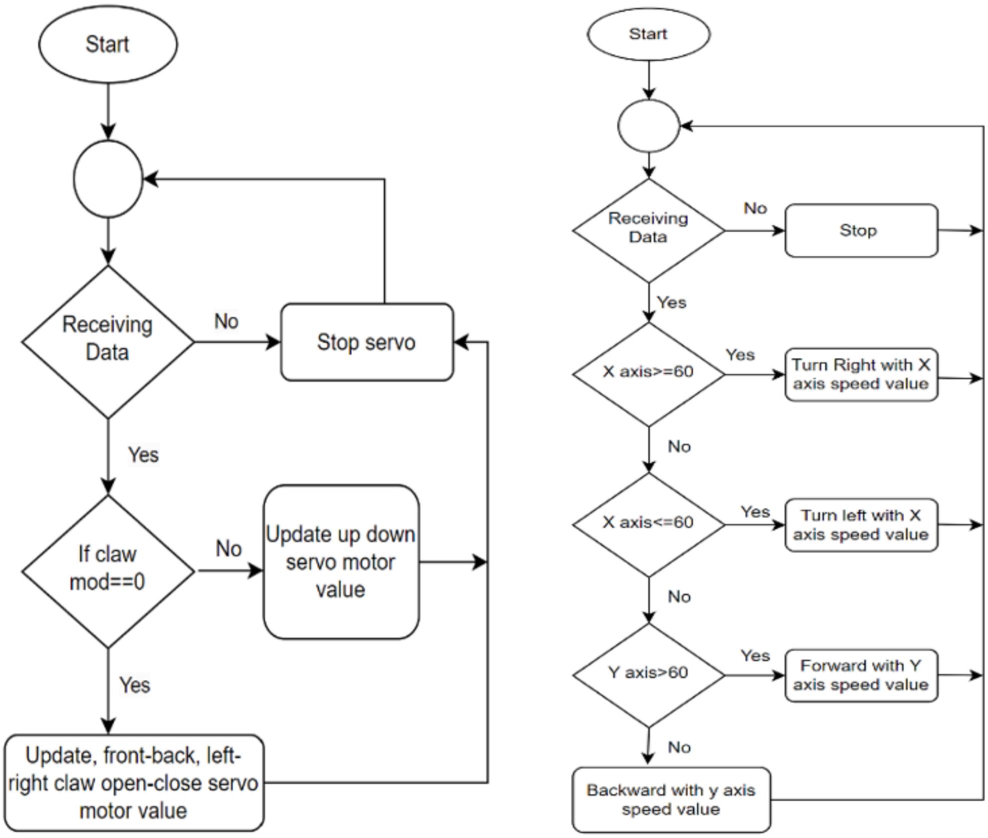
Gesture Control (Fig. 6(b)):

- Monitor incoming data.
- Trigger an emergency stop if no data is detected.
- Evaluate the x-axis value upon receiving data:
    - If x-axis value is ≥ 60, turn the motors right with the corresponding speed.
    - If x-axis value is ≤ -60, turn the motors left.
- Check the y-axis data:
    - If y-axis value is ≥ 60, move the motors forward.
    - If y-axis value is ≤ -60, move the motors backward.
- Ensure real-time responsiveness and precise control of movements through this iterative process.

Obstacle Avoidance (Fig. 7(a)):

- Measure the distance to obstacles using the ultrasonic sensor.
- If the distance is less than 20 units:
    - Stop the car.

- Reverse for 0.2 seconds.
- Turn to check both left and right.
- Compare distances and steer towards the side with more clearance.
  - If the distance is greater than 20 units, continue driving forward.
  - Continuously scan, detect, and navigate to avoid obstacles autonomously.



(a) (b)

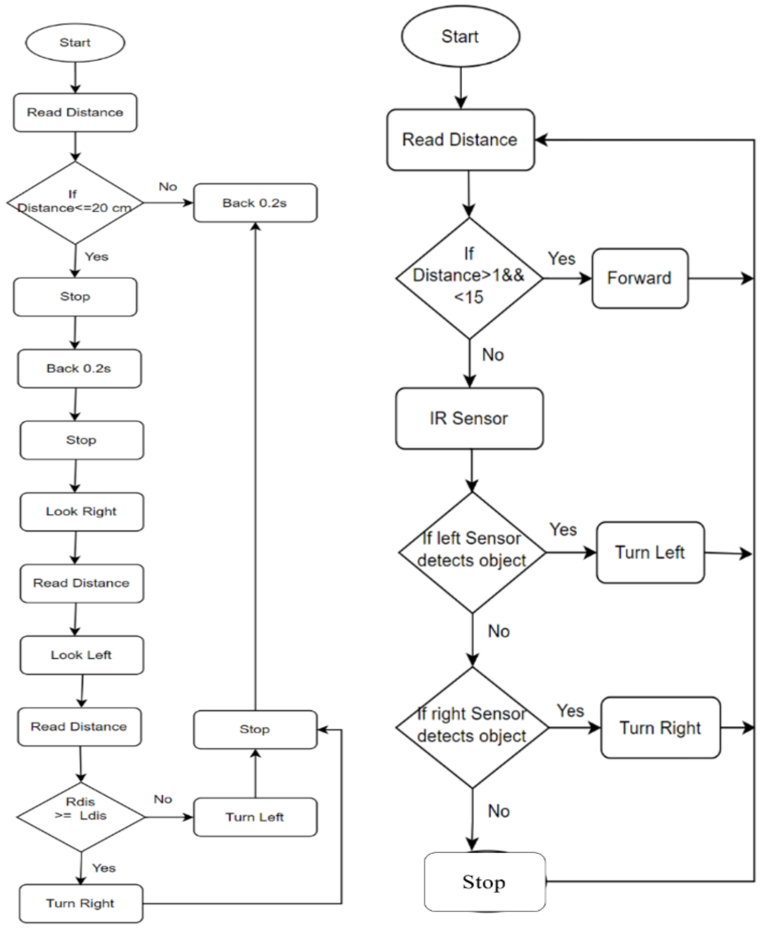
Fig. 6. Receiver (a) claw control and (b) gesture control mode flowchart

Obstacle Following (Fig. 7(b)):

- Measure the distance to the nearest object.
- If the distance is between one and fifteen units, indicating an obstacle is nearby, the car continues forward.
- If the distance is outside this range, the car analyzes data from the left and right infrared sensors.
- If the left sensor detects an obstacle, the car steers left to follow it.
- If the right sensor detects an obstacle, the car steers right.
- If neither sensor detects an obstacle, the car stops.
- This obstacle detection and navigation process is continuously repeated to ensure the vehicle can autonomously follow obstacles safely without collisions.

Line Following (Fig. 8):

- Assess input from IR sensors to follow a predefined path.
- Perform complex maneuvers based on sensor data for precise navigation.



(a) (b)

Fig. 7. (a) Obstacle avoidance and (b) obstacle following: mode flowchart

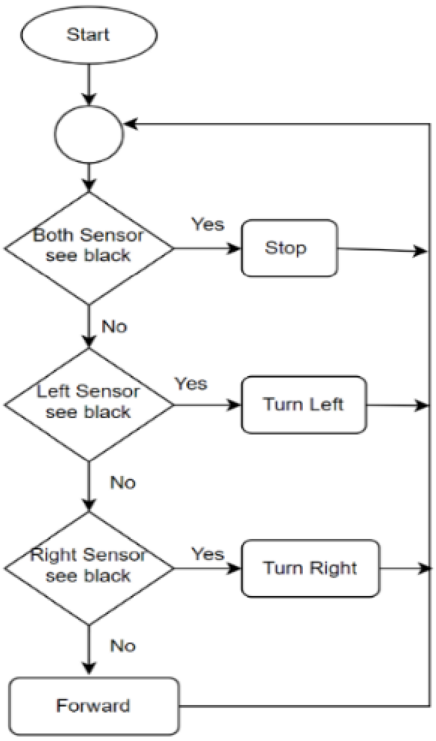


Fig. 8. Receiver line following mode flowchart

3. Experimental Setup

3.1. Circuit Setup

To grasp the robot's functionality, we begin by analyzing its core circuitry, depicted in the circuit diagrams. These diagrams reveal the intricate network of connections and components essential for the robot's operation. By understanding this foundational design, we can better appreciate the overall performance detailed in the paper.

We start by looking at the transmitter circuit diagram as shown in Fig. 9, which illustrates the complex web of connections and parts that are necessary to send data and commands to our versatile robot system. This schematic provides an extensive overview of the transmitter's architecture, enabling an in-depth understanding of its functioning and operation.

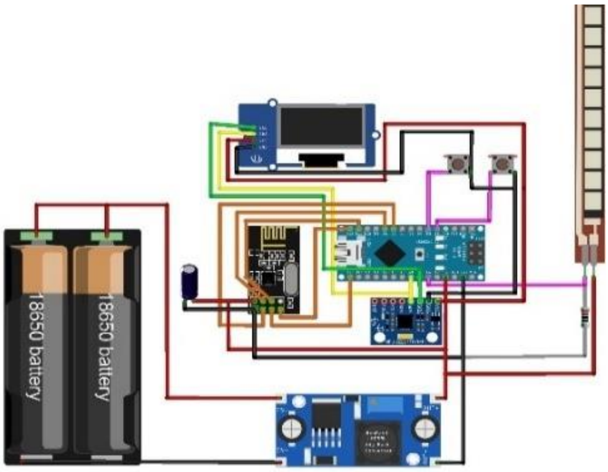


Fig. 9. Transmitter circuit diagram detailing the connections necessary for data transmission

We then present the receiver circuit diagram in Fig. 10, illustrating the relationships among components within our robotic system's receiving end. This provides detailed insights into the operation and interconnection of each component in the receiver unit.

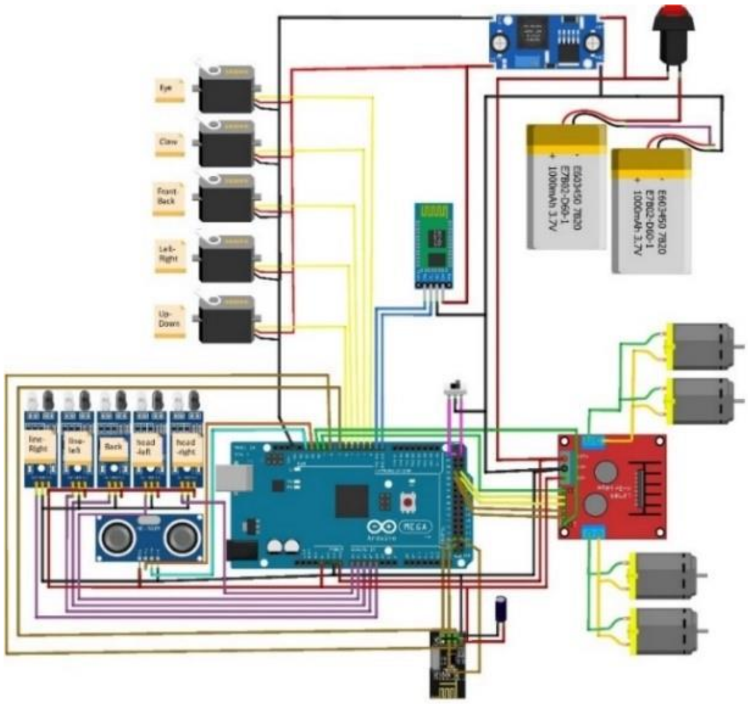


Fig. 10. Receiver circuit diagram detailing the connections necessary for performing proposed tasks

3.2. Connection Table

Tables 1 and Table 2 detail the Transmitter and Receiver connections, respectively. Table 1 summarizes the transmitter's pin connections, ensuring each component is correctly interfaced for optimal data transmission. Table 2 lists the pin connections for the receiver unit, ensuring that each component operates correctly to achieve optimal data-receiving performance, as well as the proper execution of the intended task functionalities.

Table 1. Transmitter pin connection table

Component Type	Arduino Nano Pin	Connected Component Pin	Component Name
Communication Module	7	CE	nRF24L01+
	8	CSN	
	13	SCK	
	11	MOSI	
	12	MISO	
Display/ Mode Indicator	A4	SDA	I2C OLED Display
	A5	SCL	
Sensor	A4	SDA	Mpu-6050
	A5	SCL	
	A6	Output and 47 kΩ resistors common point	Flex Sensor
Mode Change Buttons	2	Push Button 1	Claw selector
	3	Push Button 2	Mode selector

Table 2. Receiver pin connection tables

Component Type	Arduino Mega Pin	Connected Component Pin	Component Name
Communication module	7	CE	nRF24L01+
	8	CSN	
	52	SCK	
	51	MOSI	
	50	MISO	
	0 (RX)	TX	HC-06 Bluetooth Module
	1 (TX)	RX	
Actuator	2 (PWM)	Signal pin	Servo motor- (Ultrasonic as Front Eye)
	3 (PWM)	Signal pin	Servo motor- (Arm Claw)
	4 (PWM)	Signal pin	Servo motor- (Arm Left-Right)
	5 (PWM)	Signal pin	Servo motor- (Arm Up-Down)
	6 (PWM)	Signal pin	Servo motor- (Arm Up-Down)
	9 (PWM)	EN-A	L298 Motor Driver
	29	In1	
	31	In2	
	28	In3	
	30	In4	
	10 (PWM)	EN-B	
Sensor	12	Echo	HC-SR04 Ultrasonic Sensor
	11	Trigger	
	A0	Out	IR sensor- (Robot Head Right)
	A1	Out	IR sensor- (Robot Head Left)
	A2	Out	IR sensor- (Robot Car Back)
	A3	Out	IR sensor- (Line Left)
	A4	Out	IR sensor- (Line Right)
Mode Select Button	22, 23	Push Button pin	Bluetooth, Voice Mode Button

In connection setup Table 2 for the receiver part, we were careful to connect the motor driver enable pins to PWM connection pins of Arduino. As we used the Servo.h library to control the servo motors, which can use timers 1, 3, 4, and 5 according to the Arduino Servo Library documentation [58]. Also, according to the Arduino Mega 2560 Datasheet [59] and the ATmega640/V-1280/V-

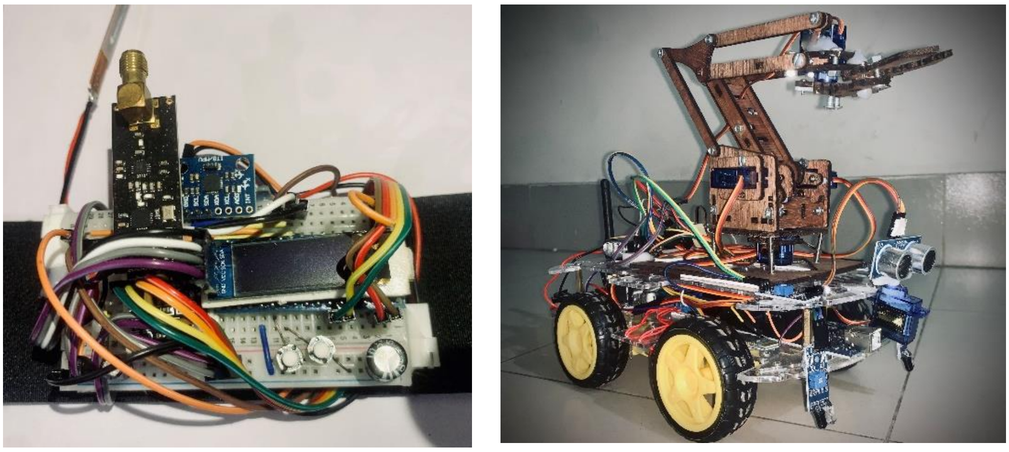
1281/V-2560/V-2561 Datasheet [60], specific pins are associated with specific timers on the Arduino boards. Those associated pins are:

- Timer 1 is associated with pins 11, 12, and 13.
- Timer 3 is associated with pins 2, 3, and 5.
- Timer 4 is associated with pins 6, 7, and 8.
- Timer 5 is associated with pins 44, 45, and 46.

The Arduino Servo Library documentation [58] says that if the Servo.h header uses any of those timers to control a servo motor, then the PWM functionalities with the analogWrite() function will not be available for those timer-associated pins. To avoid any kind of conflict, we totally avoided those PWM pins and used pins 9 and 10 for controlling motor speed on the L298 motor driver with the PWM signal shown in Table 2.

3.3. Hardware Setup

The hardware setup integrates various components, including the transmitter and receiver units, as shown in Fig. 11. The transmitter sends data to the receiver, which processes the information to control the robot's movements. The system is assembled according to the pin connection shown in Table 1 and Table 2.



(a) (b)

Fig. 11. (a) Transmitter unit setup; (b) Receiver unit setup

3.4. Bluetooth Controlled Application

The Bluetooth controller application, developed using C#.NET 8.0, facilitates seamless control of the robot. Users can easily manage the robot's movements and tasks via an intuitive interface, as depicted in Fig. 12 and Fig. 13. This application is compatible with both Windows and Mac OS X. It acts as a user interface for seamless control over the robotic arms and the robot car's movements. It connects to the robot via the Windows communication (COM) port, enabling remote control of various features. Users can control the robotic arm to perform precise tasks and easily maneuver the robot car through the user-friendly buttons and controls. This specially designed controller application enhances the versatility of the multipurpose robot system, providing an improved user experience.

The user must enter the COM port number that the system is using to communicate with the receiver's Bluetooth module during the connection setup phase as illustrated in Fig. 12. An unsuccessful connection will display an error message, while a successful connection will open the controller phase as shown in Fig. 13. By integrating a custom Bluetooth controller app, the multipurpose robot system gains a user-friendly interface. This interface allows for smooth control over the robot's movements and functionalities. The application enhances the robot's versatility by providing real-time control and feedback, significantly improving the user experience.

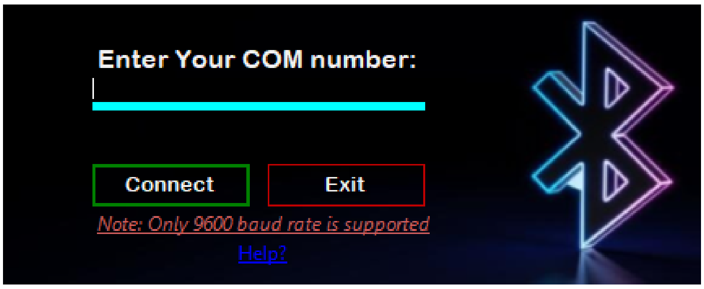


Fig. 12. Connection setup interface of Bluetooth Communicator

Fig. 13. Controller page of Bluetooth Communicator

4. Result and Discussion

4.1. Simulation Analysis

For simulation, we used Proteus IDE which is a virtual system modeling (VSM) and circuit simulation software. Proteus VSM supports advanced embedded simulation, providing system-level simulation based on schematic circuits. It includes a comprehensive database or library of components. Additionally, users can create and add new components to the library if they are not included in the default software library [70]. Simulations using the Proteus platform validated the robot's functionality, including data transmission, object manipulation, and obstacle detection. Fig. 14, Fig. 15, Fig. 16, Fig. 17, Fig. 18 illustrate the successful operation of each mode tested. Fig. 14 and Fig. 15 illustrate the circuit used for these tests. Simulations are conducted to evaluate the robot's ability to transfer data wirelessly, manipulate objects using servo motors, measure distance using ultrasonic sensors, detect objects using infrared sensors, view mode details using OLED displays, and move cars using DC motors.

[*Note:* nRF24L01+ and mpu-6050 accelerometer were unavailable in proteus for simulation. 433Mhz radio frequency (RF) module is used as a substitute for nRF24L01+ module and 4 push-buttons for controlling left-right-front-back movement are used as a substitute for mpu-6050 accelerometer.]

Before building the actual device, a simulation was run to check if all the components would work together smoothly. This includes circuitry, wireless communication, OLED display, gesture control mode, flex sensor, ultrasonic sensor, and infrared sensor. Fig. 16 demonstrates the successful transmission of data between the transmitter and receiver in the simulation. The OLED display correctly shows the indication of gesture control mode. Furthermore, the virtual terminal on the receiver confirmed the robot car moving forward, demonstrating the proper function of the L298D motor driver in gesture control mode.

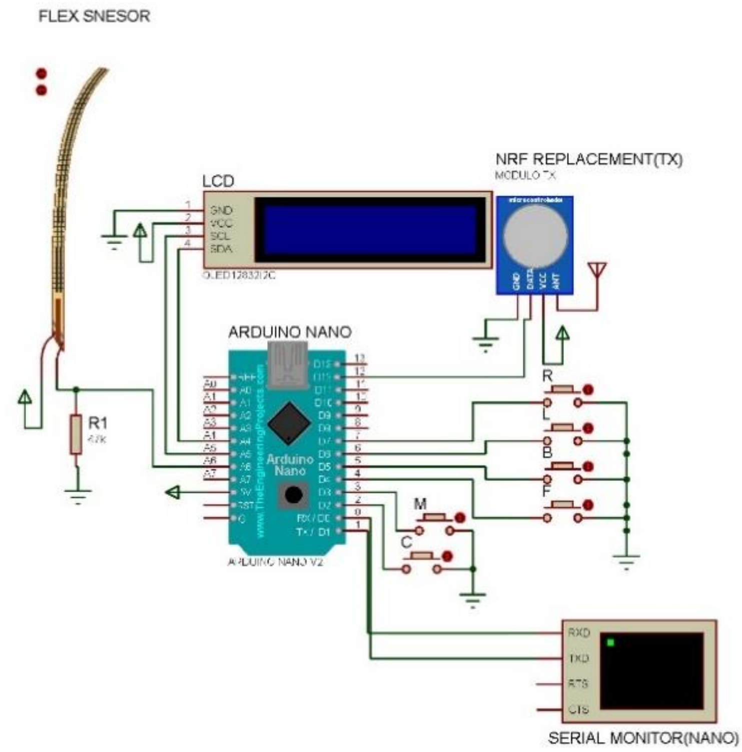


Fig. 14. Transmitter circuit simulation using Proteus

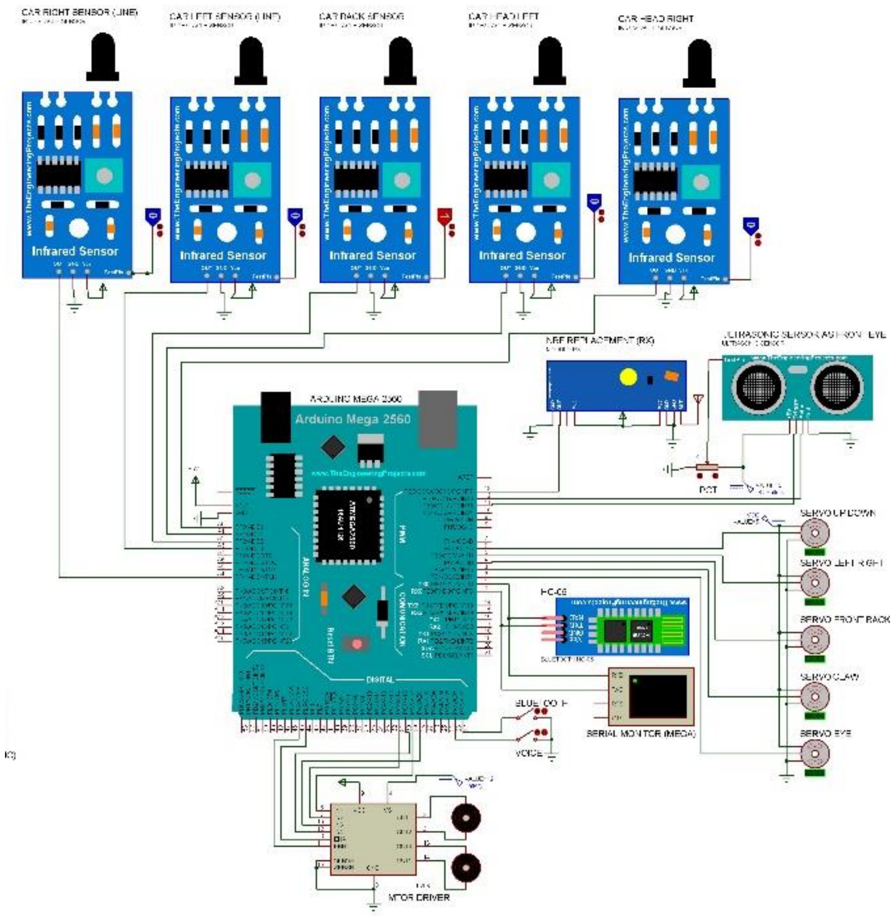


Fig. 15. Receiver circuit simulation using Proteus

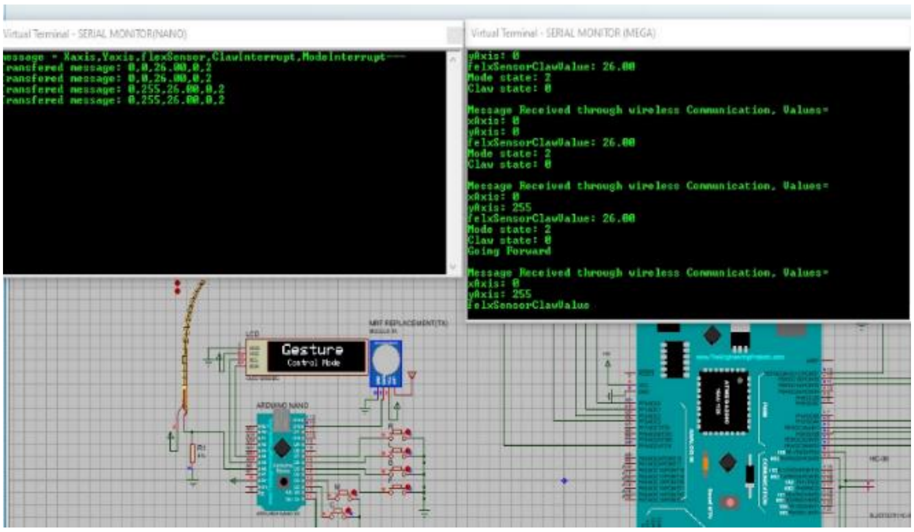


Fig. 16. Simulation for gesture control mode

Fig. 17 demonstrates the validation of a robotic arm control mode. It is evident from the figure that the receiver updates servo motor values based on data transmitted from a transmitter because "Xaxis" 255 value increased the robot claw "leftRightMovement" value to 95 from 90. Fig. 18 illustrates a successful test of the robot car's obstacle avoidance mode. The virtual terminal displays that the car is moving forward. This confirms that the infrared and ultrasonic sensors are providing accurate data, and the L298D motor driver is responding correctly.

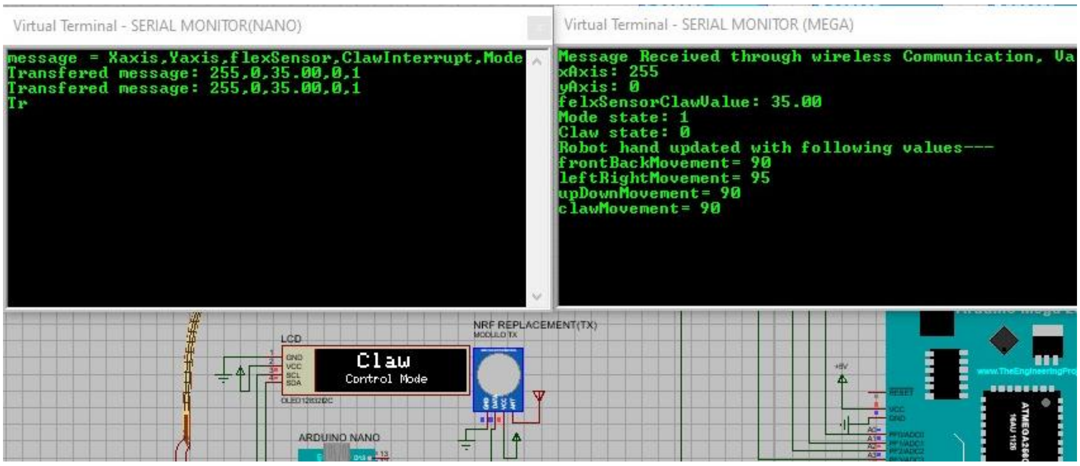


Fig. 17. Simulation for Robotic Arm controlling mode

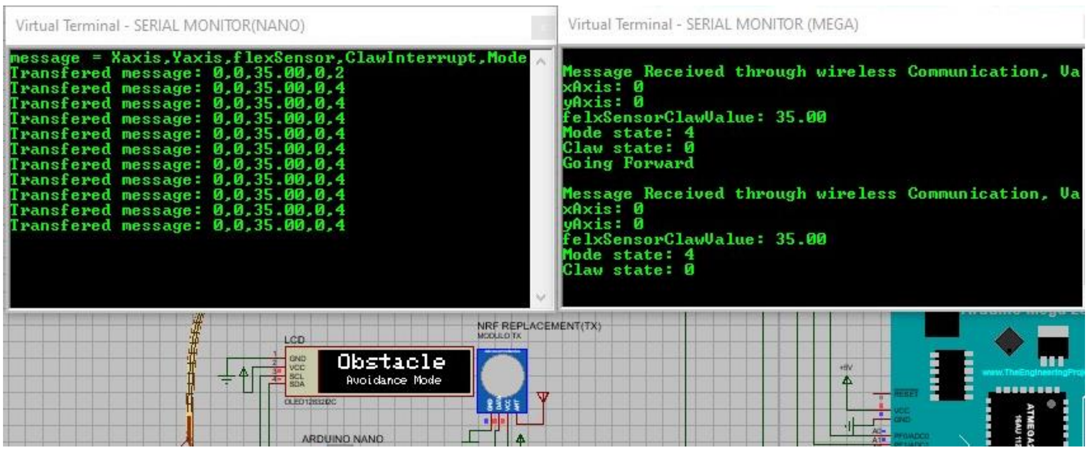
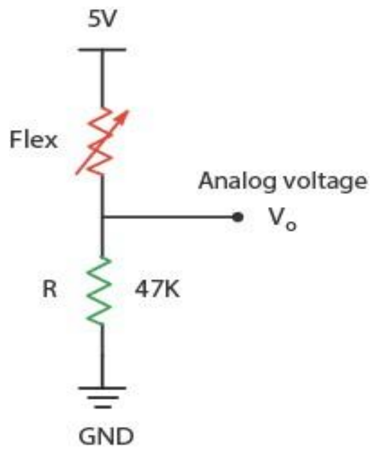


Fig. 18. Simulation for obstacle avoidance mode

The voltage divider circuit depicted in Fig. 19 is employed in the flex sensor simulation to measure the sensor's responses. It is established that the flex sensor's internal resistance varies with bending. We applied the voltage divider formulas provided below to determine the bending angle measurement of the flex sensor.



$$V_0 = V_{CC}\frac{R}{R + R_{Flex}}$$

Fig. 19. Voltage divider circuit for flex sensor

As the bend radius increases in this configuration, the output voltage decreases. For instance, with the sensor flat (0°) and using a 5V supply and a 47 kΩ pull-down resistor, the resistance is relatively low (approximately 25 kΩ). In these conditions, the output voltage is calculated using (2), as follows:

$$V_0 = 5V\frac{47\ k\Omega}{47\ k\Omega + 25\ k\Omega} = 3.26\ V \tag{2}$$

The resistance rises to about 100 kΩ when the bend reaches its maximum angle of 90º. Consequently, the output voltage as shown in (3) changes to:

$$V_0 = 5V\frac{47\ k\Omega}{47\ k\Omega + 100\ k\Omega} = 1.59\ V \tag{3}$$

This formula is included in the simulation code for measuring bending angles. Fig. 20 gives a 100% accurate bending angle value through simulation as calculated numerically.

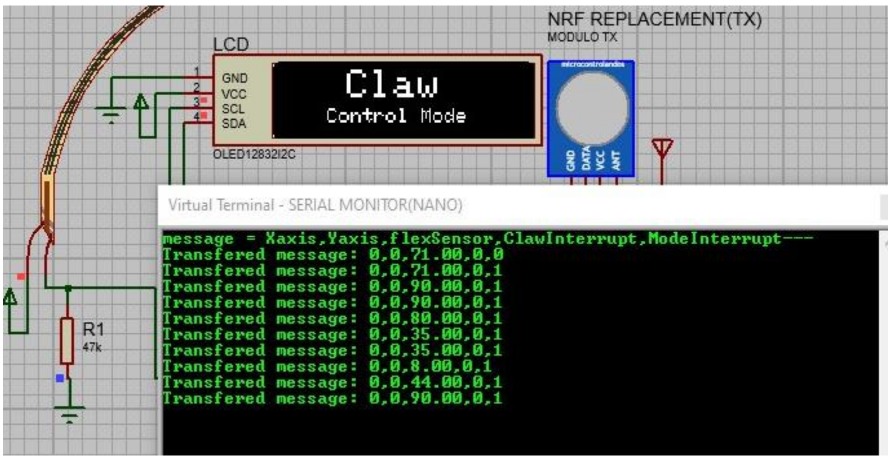
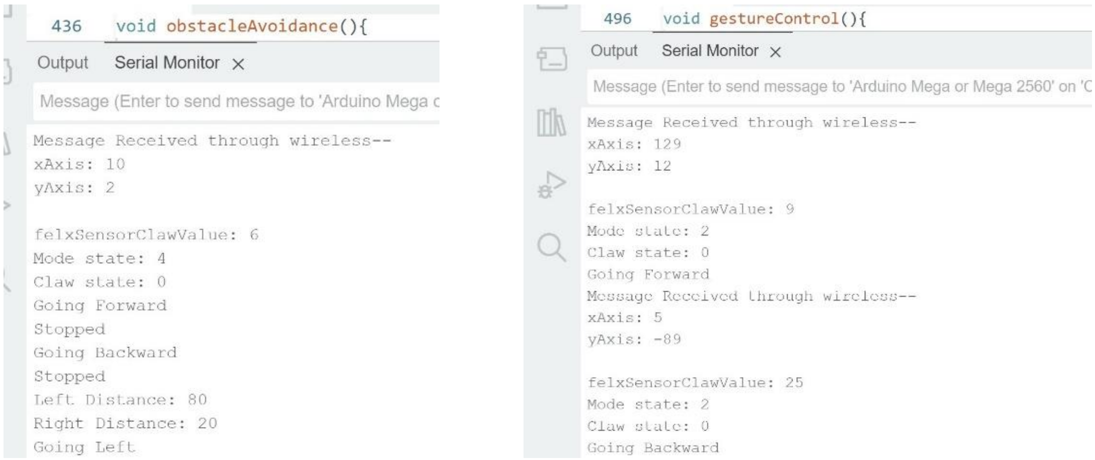


Fig. 20. Flex sensor bending in different angle simulation

4.2. Hardware Result

We tested the robot's capabilities in real-world situations to see how well it performed. This included how quickly it responded to gestures, how precisely it moved, how accurately it detected obstacles, and how reliably it communicated wirelessly. While the simulation predicted accurate performance, real-world tests revealed some inconsistencies due to factors like sensor response time and power supply limitations. Gesture recognition was evaluated using the Arduino IDE Serial monitor. The nRF24L01+ module worked consistently and achieved a peek data transmission range of 902 meter. The ultrasonic sensor precisely measured distances, while the MPU-6050 accelerometer's high sensitivity allowed it to track even small movements. Bluetooth functionality was verified by connecting to a smartphone and sending commands. Our custom software also passed all tests. Additionally, the Flex sensor consistently provided the expected values.

Fig. 21 and Fig. 22 demonstrate the verification process for different functionalities. In Fig. 21, we validate gesture control and obstacle avoidance modes through the Arduino IDE's serial monitor. This ensured the correct display of mode (mode state=2 for gesture control, mode state = 4 for obstacle avoidance) and axis values. Additionally, the left and right object distances in Fig. 21(a) confirmed the ultrasonic sensor's proper operation. Similarly, Fig. 22 showcases the verification of the robotic claw mode. The servo motor value's dynamic update based on the axis value guarantees the accuracy of the claw control mode.



(a) (b)

Fig. 21. Illustration of (a) Gesture control mode and (b) obstacle avoidance mode results

Fig. 22. Robotic claw value update result

The range of the nRF24L01+ module's data transmission and reception was tested on a long, straight road at 6 a.m. to ensure minimal obstacles for maximum performance. The testing location was in Dhaka, Bangladesh, specifically on Sayem Sobhan Anvir Road, near American International University-Bangladesh and North South University. As illustrated in Fig. 23, the maximum wireless connectivity range achieved under these ideal conditions was approximately 902 meters. The test setup involved positioning the transmitter and receiver modules at the two ends of the road, with minimal interference from surrounding structures. There could be a ±10m margin of error as we manually selected the location based on our position. The testing was conducted early in the morning to avoid any significant disruptions from pedestrian or vehicular traffic, providing an optimal environment to assess the module's range accurately.

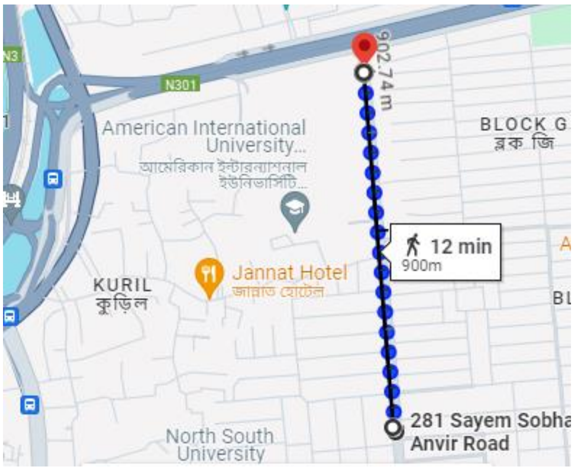


Fig. 23. Maximum wireless connectivity range of nRF24L01+ module

Finally, we have achieved all the objectives of this paper, demonstrating the robot system's effectiveness and efficiency.

4.3. Accuracy of the Hardware Result

Though our system functions as we intended, sometimes it fails to perform its assigned tasks, such as correctly measuring distances and correctly detecting black lines. Various internal and external factors can be reasoned for those unintended behaviors, such as ATmega2560 microchip clock speed, sensor response time, limited power supply, natural light influence and so on. These factors can be mitigated by using a higher clock speed MCU such as the STM32 or Raspberry Pi, a higher discharge rate battery, and a more sophisticated infrared sensor that is not influenced by natural light.

For line following mode, it can be seen from Table 3 that the robot car detects a black line with 75% accuracy and can navigate through the black line when we set the motor speed low, around 80 PWM value. Again, when the speed is a little bit increased with a 140 PWM value, then the line detection accuracy falls to 25%. Furthermore, when we increased the PWM value to its maximum value of 255 for fastest speed, the line detection accuracy became 8%.

Table 3. Line detection accuracy tables

Sample no	Line Detection		
	Motor Speed (80)	Motor Speed (140)	Motor Speed (255)
1	Detected	Not Detected	Not Detected
2	Detected	Detected	Not Detected
3	Not Detected	Not Detected	Not Detected
4	Detected	Not Detected	Detected
5	Not Detected	Detected	Not Detected
6	Detected	Not Detected	Not Detected
7	Not Detected	Not Detected	Not Detected
8	Detected	Not Detected	Not Detected
9	Detected	Not Detected	Not Detected
10	Detected	Not Detected	Not Detected
11	Detected	Detected	Not Detected
12	Detected	Not Detected	Not Detected

[*Note:* When the robot is navigating the predetermined path by detecting a black line with an IR sensor, one sample is taken for 5-second navigation durations for Table 3.

For object avoidance and detection mode, the ultrasonic sensor sometimes failed to detect objects. Especially when the robot is moving too fast with PWM value 250, its accuracy decreases relatively low, around 30%, as shown in Table 4. It can detect objects with 50% precision when the PWM value sets to 180. The highest precision of object detection, around 70%, is in the slow mode with PWM value 80.

Table 4. Object detection accuracy tables

Sample no	Object Detection		
	Motor Speed (80)	**Motor Speed (180)**	**Motor Speed (250)**
1	Detected	Not Detected	Not Detected
2	Detected	Detected	Not Detected
3	Detected	Not Detected	Not Detected
4	Detected	Not Detected	Detected
5	Not Detected	Detected	Not Detected
6	Detected	Detected	Detected
7	Detected	Not Detected	Not Detected
8	Detected	Detected	Detected
9	Not Detected	Not Detected	Not Detected
10	Not Detected	Detected	Not Detected

[*Note:* Object is placed in front of moving robot. One object equals to one sample for Table 4]

The accuracy of line detection and object avoidance varied with motor speed. Lower speeds improved detection accuracy, highlighting the importance of speed calibration for optimal performance.

4.4. Comparison Between Simulation and Hardware Implementation Result

We compared experimental results with numerical simulations to assess the accuracy of the simulation models and their ability to predict real-world behavior. The comparison between simulation and hardware results showed that while most sensors performed consistently, the flex sensor required calibration for accurate readings, emphasizing the need for real-world adjustments. This difference is likely caused by external factors, such as temperature fluctuations. Fig. 24 illustrates that the flex sensor can sometimes register values exceeding 90 degrees when bent.

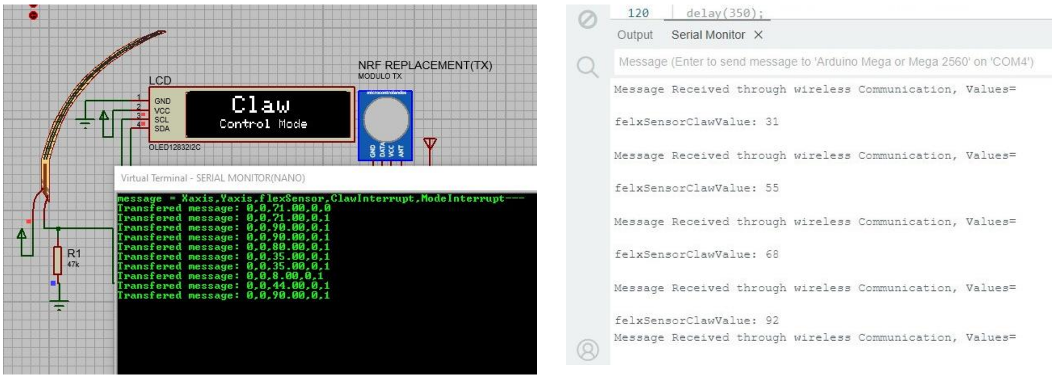


Fig. 24. Simulated and real-life value of flex sensor

Furthermore, precise object and line detection needed calibration of the infrared sensor intensity, which was not taken into consideration in the simulation. This emphasizes how crucial it is to take calibration procedures and real-world variables into account in order to get the best possible sensor performance in real-world applications. Calibration can be done by placing the robot in the black line and slowly adjusting the IR sensor potentiometer knob to the desired position where it can detect the black line perfectly. The comparisons also highlight the necessity of iterative testing and improvement to close the gap between the results of simulations and real-world experiences, guaranteeing the dependability and efficiency of the robot system in various settings.

4.5. Cost Analysis

We have estimated the total amount of funding required to complete the paper given in Table 5. This covered all the necessary purchases, including labor and overhead, as well as any additional materials and parts. Below is a summary of the expense categories.

- Material Costs: This is the money we spent on all the stuff we needed to build the robot car and robotic arm.
- Labor Costs: This is the money we paid for any work done by people to build or work on the paper. In our case, it is Zero as we build this paper on our own.
- Overhead Expenses: These are additional costs related to running the paper, like electricity, transport, and additional material (e.g., soldering lead, glue sticks) costs.

Table 5. Component cost analysis

Component Name	Unit Price (৳)	Quantity	Total Price (৳)
Arduino Mega	2050/-	×1	2050/-
Arduino Nano	500/-	×1	500/-
Flex Sensor	1200/-	×1	1200/-
Ultrasonic Sensor	100/-	×1	100/-
Infrared Sensor	95/-	×5	475/-
nRF24L01+ module	350/-	×2	700/-
nRF Base	200/-	×2	400/-
I^2C OLED Display	350/-	×1	350/-
L298D	200/-	×1	200/-
HC-06 Bluetooth Module	330/-	×1	330/-
Car Chassis	900/-	×1	900/-
Robotic Claw	950/-	×1	950/-
Mpu-6050	300/-	×1	300/-
Lipo 3S Battery (1100mah)	1450/-	×1	1450/-
3S Battery Charger	450/-	×1	450/-
18650 Li^+ battery	170/-	×1	170/-
3.7V charging module	75/-	×1	75/-
12V to 5 Volt Converter	200/-	×1	200/-
3.7V to 5V Converter	110/-	×1	110/-
Jumper Wire Set	150/-	×1	150/-
Resistor	5/-	×2	10/-
Capacitor	8/-	×5	40/-
Mini Breadboard	70/-	×1	70/-
Double Sided Tape	80/-	×1	80/-
Switches	20/-	×4	80/-
Servo Motor (SG90)	170/-	×5	850/-
		Total=	**12,190/-**

From Table 5, we can see that the components have a total cost of 12,190/-Tk. Our overall cost to complete this paper came to 12,890/-Tk approximately $117.45, includes materials, overheads, and additional components. This budget demonstrates the feasibility of developing a functional robot within a reasonable cost range.

4.6. Limitation in the Paper

Even though we were able to construct the multifunctional robot car with a robotic arm, there were still certain obstacles we had to overcome. As we did not integrate a camera module, real-time video footage was not available. Furthermore, the robotic arm can only move in specific directions due to its movement being limited to four degrees of freedom. Also, we are utilizing simple infrared sensors for line detection. As a result, the robot cannot move at its fastest in line following mode.

Furthermore, the accuracy of the robot's movements is sometimes compromised by its two infrared sensors for line following. In summary, the lack of a camera module limits real-time video feedback, while the robotic arm's restricted movement affects its versatility. Additionally, the reliance on simple infrared sensors for line detection reduces accuracy at higher speeds. Future iterations could integrate advanced sensors and a camera module for enhanced functionality and accuracy.

5. Conclusion and Future Endeavors

This multifunctional robot car with a robotic arm is a big step forward in making robots more efficient and adaptable for various tasks. The line-following feature can be used in hospitals to deliver goods to patients with contagious illnesses, minimizing direct human contact. At airports, it can automate luggage delivery, and in restaurants, it can serve food to customers by following a predetermined path. If no line is available, the robot can complete tasks using hand gesture control. In military applications, the object avoidance feature enables the robot to explore hazardous areas autonomously, avoiding obstacles. For assistive tasks, the object-following capability allows the robot to carry goods and follow a person, reducing the need for individuals to carry items themselves. The robotic hand developed in this research is highly beneficial in environments like painting shops and shot blasting chambers, where dusty and hazardous conditions pose risks to human workers. Operators can manage the robotic functions safely from outside these areas, observing through a protective glass barrier. Also in an industrial storage area, it can grab objects and use its line-following ability to deliver them to a predetermined place. The gesture control features could be used to deliver objects in a large storage area thanks to their high range of controllability. Disabled people who cannot walk properly can use this robot to manipulate any object without the need for other human assistance. Besides, its multi-control feature and user-friendly interface allow users to operate the robot car in their preferred manner, ensuring maximum comfort and convenience. Even though there are still challenges, this study shows how much innovation is possible in robotics.

The use of the Arduino Mega ensures readiness for future expansions without concerns about memory size or I/O ports. Future research can enhance the robot by incorporating real-time visual capabilities, increasing the arm's movement options, and improving sensors for the line-following mode. For example, the ESP32 cam module could be used to provide real-time visual feedback, and the QTRX-MD-16A reflectance sensor could enable precise line-following operations. Additionally, advanced control algorithms and artificial intelligence approaches, such as machine learning models for obstacle detection and avoidance, could significantly improve the robot's autonomous navigation and decision-making capabilities.

This paper aligns with Sustainable Development Goals (SDGs) 3, 9, and 17 by showcasing its multifaceted impact. SDG 3 (Good Health and Well-being) is supported through potential healthcare applications, enhancing patient care and improving quality of life. SDG 9 (Industry, Innovation, and Infrastructure) is advanced by the robotic arm's contribution to industrial productivity, promoting efficient and innovative manufacturing processes. SDG 17 (Partnerships for the Goals) is fostered through collaborations with technology companies, enhancing and distributing the robot’s capabilities. These efforts collectively demonstrate the project's commitment to leveraging technology for health, industrial advancement, and global partnerships.

However, due to budget constraints, we implemented this project on a small scale. We tested all functionalities, and they are working perfectly. In the future, we plan to scale up the project, ensuring it can perform all intended tasks with precision and perfection. Scaling up will bring some more challenges, such as power management, moving the scaled-up robot at the desired speed, and avoiding obstacles precisely. These challenges can be addressed in future research by increasing the battery size, integrating more powerful motors with motor drivers that can deliver sufficient power to move the robot at its desired speed, and increasing the number of ultrasonic and infrared sensors to avoid obstacles perfectly. However, the main control logic and working process of all the features will remain the same.

Overall, this work lays a solid foundation for future robotics research and development, poised to meet societal demands across various fields, and scaling up this project in the future will significantly advance robotics technology and practical applications.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] W. A. Salah, A. A. Sneineh, and A. A. Shabaneh, "A smartphone sensor-based development and implementation of a remotely controlled robot arm," *Journal of Robotics and Control*, vol. 5, no. 4, pp. 1180-1188, 2024, https://doi.org/10.18196/jrc.v5i4.21987.

[2] M. Kathe and S. Bakhtar, "Bluetooth based home automation using Arduino and android application," *International Journal of Automatic Control System*, vol. 7, no. 2, pp. 18-23, 2021, https://doi.org/10.18196/jrc.v5i4.21987.

[3] V. Y. Akgün and E. Maşazade, "Formation of a wireless sensor network using custom-designed sensors having low power and low cost components," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 5, pp. 2702-2717, 2020, https://doi.org/10.3906/elk-2001-107.

[4] S. M. A. Shah, "Multi-way controlled robot vehicle using Arduino and RF module," *Journal of Applied Engineering and Technology*, vol. 5, no 1, pp. 1-8, 2021, https://doi.org/10.55447/jaet.05.01.29.

[5] M. Mahbub, "Design and implementation of multipurpose radio controller unit using nRF24L01 wireless transceiver module and Arduino as MCU," *International Journal of Digital Information and Wireless Communications*, vol. 9, no. 2, pp. 21-37, 2019, https://www.rroij.com/open-access/design-and-implementation-of-multipurpose-radio-controller-unit-using-nrf24l01-wireless-transceiver-module-and-arduino-a.pdf.

[6] L. R. K. and V. Vijayaraghavan, "A self-powered, real-time, NRF24L01 IoT-based cloud-enabled service for smart agriculture decision-making system," *Wireless Personal Communications*, vol. 124, pp. 207-236, 2022, https://doi.org/10.1007/s11277-021-09462-4.

[7] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues and Y. Park, "Intrusion Detection Protocols in Wireless Sensor Networks Integrated to Internet of Things Deployment: Survey and Future Challenges," *IEEE Access*, vol. 8, pp. 3343-3363, 2020, https://doi.org/10.1109/ACCESS.2019.2962829.

[8] M. C. Baba, J. J. B. Grado, D. J. L. Solis, I. M. Roma, and J. T. Dellosa, "A multisensory Arduino-based fire detection and alarm system using GSM communications and RF module with an android application for fire monitoring," *International Journal of Innovative Science and Research Technology*, vol. 7, no. 3, pp.964-968, 2022, https://doi.org/10.5281/zenodo.6433836.

[9] Z. Zou, Q. Wu, Y. Zhang and K. Wen, "Design of Smart Car Control System for Gesture Recognition Based on Arduino," *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 695-699, 2021, https://doi.org/10.1109/ICCECE51280.2021.9342137.

[10] G. Young, H. Milne, D. Griffiths, E. Padfield, R. Blenkinsopp, and O. Georgiou, "Designing mid-air haptic gesture controlled user interfaces for cars," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. 81, pp. 1-23, 2020, https://doi.org/10.1145/3397869.

[11] C. K. Wen, L. C. Yong, and C. Nataraj, "Gesture controlled robotic arm for hazardous chemical control," *Journal of Applied Technology and Innovation*, vol. 6, no. 3, pp. 47-54, 2022, https://jati.sites.apiit.edu.my/files/2022/07/Volume6_Issue3_Paper8_2022.pdf.

[12] S. Shruti, S. K. Verma, S. Singh, and T. Gupta, "Arduino based hand gesture controlled robot," *International Research Journal of Engineering and Technology*, vol. 9, no. 5, pp. 826-831, 2022, https://www.irjet.net/archives/V9/i5/IRJET-V9I5236.pdf.

[13] L. Guo, Z. Lu and L. Yao, "Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 300-309, 2021, https://doi.org/10.1109/THMS.2021.3086003.

[14] V. Teeda, K. Sujatha, and R. Mutukuru, "Robot voice: A voice-controlled robot using Arduino," *arXiv* 2024, https://doi.org/10.48550/arXiv.2402.03803.

[15] S. Srivastava and R. Singh, "Voice controlled robot car using Arduino," *International Research Journal of Engineering and Technology*, vol. 7, no. 5, pp. 4033-4037, 2020, https://www.irjet.net/archives/V7/i5/IRJET-V7I5770.pdf.

[16] K. S. Kumar, P. S. Reddy, M. R. V. Revanth, and K. Samalla, "Voice-controlled robot vehicle using Arduino," *International Journal for Research in Applied Sciences and Engineering Technology*, vol. 10, no. 6, pp. 2786-2791, 2022, https://doi.org/10.22214/ijraset.2022.44458.

[17] B. Teterbay, A. Bhati, A. Srivastava, A. A. Deshpande, "Smartphone controlled multipurpose robot car," *International Journal of Engineering Research and Technology*, vol. 9, no. 05, pp. 485-488, 2020, https://doi.org/10.17577/IJERTV9IS050382.

[18] M. E. Sanyaolu, V. O. Amolegbe, and A. A. Willoughby, "Bluetooth and Arduino Uno-based voice-controlled home automation system," *International Journal of Research and Innovation in Applied Science*, vol. 7, no. 9, pp. 27-30, 2022, https://doi.org/10.51584/IJRIAS.2022.7903.

[19] A. Sood, R. Fofaliya, and R. B. Chandran, "Smart home automation using Arduino integrated with Bluetooth and GSM," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11S, pp. 1140-1143, 2019, https://doi.org/10.35940/ijitee.K1230.09811S19.

[20] S. Khawate, K. Prajapati, Y. Anand, and K. Chandodwala, "Voice controlled robotic car using Arduino," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 9, no. 3, pp. 119-129, 2022, https://doi.org//10.32628/IJSRSET122934.

[21] H. Zhou, C. Tawk and G. Alici, "A 3D Printed Soft Robotic Hand With Embedded Soft Sensors for Direct Transition Between Hand Gestures and Improved Grasping Quality and Diversity," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 550-558, 2022, https://doi.org/10.1109/TNSRE.2022.3156116.

[22] M. Sillang and S. Patricia, "The development of robot arm with smartphone control using Arduino," *International Journal of Advanced Research in Technology and Innovation*, vol. 3, no. 2, pp. 39-46, 2021, https://myjms.mohe.gov.my/index.php/ijarti/article/view/13439.

[23] K. Kunal, A. Z. Arfianto, J. E. Poetro, F. Waseel, and R. A. Atmoko, "Accelerometer implementation as feedback on 5 degree of freedom arm robot," *Journal of Robotics and Control*, vol. 1, no. 1, pp. 31-34, 2020, https://doi.org/10.18196/jrc.1107.

[24] P. S. Warankar, M. M. Dharmadhikari, "Research paper on Bluetooth based home automation using Arduino," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 12, no. 6, pp. 404-411, 2019, https://ijarcce.com/wp-content/uploads/2023/06/IJARCCE.2023.12670.pdf.

[25] S. Kannappan, K. Kanchana, B. Haritha, K. Bhavana, and G. Lokesh, "IoT based industrial automation," *South Asian Journal of Engineering and Technology*, vol. 12, no. 3, pp. 9-19, 2022, https://doi.org/10.26524/sajet.2022.12.34.

[26] M. Uzair, S. Y. Al-Kafrawi, K. M. Al-Janadi, and I. A. Al-Bulushi, "A low-cost IoT based buildings management system (BMS) using Arduino mega 2560 and Raspberry Pi 4 for smart monitoring and automation," *International Journal of Electrical and Computer Engineering Systems*, vol. 13, no. 3, pp. 219-236, 2022, https://doi.org/10.32985/ijeces.13.3.7.

[27] M. Alagarsamy, S. R. Devakadacham, H. Subramani, S. Viswanathan, J. Johnmathew, and K. Suriyan, "Automation irrigation system using Arduino for smart crop field productivity," *International Journal of Reconfigurable and Embedded Systems*, vol. 12, no. 1, pp. 70-77, 2023, http://doi.org/10.11591/ijres.v12.i1.pp70-77.

[28] C. Stolojescu-Crisan, C. Crisan, and B. Butunoi, "An IoT-based smart home automation system," *Sensors*, vol. 21, no. 11, p. 3784, 2021, https://doi.org/10.3390/s21113784.

[29] A. A, "Smart home automated control system using Android application based on Arduino," *International Journal for Multidisciplinary Research*, vol. 5, no. 6, pp. 1-7, 2023, https://doi.org/10.36948/ijfmr.2023.v05i06.9306.

[30] L. M. Satapathy, S. K. Bastia, and N. Mohanty, "Arduino based home automation using Internet of things (IoT)," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 17, pp. 769-778, 2018, https://ieeeprojectsmadurai.com/IEEE%202019%20IOT%20BASEPAPERS/36_IOT%20BASED%20HOME%20AUTOMATION.pdf.

[31] P. Q. Anh, T. duc Chung, T. Tuan and M. k. a. A. Khan, "Design and Development of an Obstacle Avoidance Mobile-controlled Robot," *2019 IEEE Student Conference on Research and Development (SCOReD)*, pp. 90-94, 2019, https://doi.org/10.1109/SCORED.2019.8896296

[32] Y. Irawan, M. Muhardi, R. Ordila, and R. Diandra, "Automatic floor cleaning robot using Arduino and ultrasonic sensor," *Journal of Robotics and Control*, vol. 2, no. 4, pp. 240-243, 2021, https://doi.org/10.18196/jrc.2485.

[33] P. Ramesh, S. Sudheera, and D. V. Reddy, "Distance measurement using ultrasonic sensor and Arduino," *Journal of Advanced Research in Technology and Management Sciences*, vol. 3, no. 2, pp. 1-5, 2021, http://jartms.org/admin/uploads/czhw3m.pdf.

[34] M. A. Baballe, M. I. Bello, S. H. Ayagi, and U. F. Musa, "Obstacle avoidance robot using an ultrasonic sensor with Arduino Uno," *Global Journal of Research in Engineering and Computer Sciences*, vol. 3, no. 5, pp. 14-25, 2023, https://doi.org/10.5281/zenodo.10015177.

[35] A. Goswami, B. Bhattacharjee, R. Debnath, and A. Sikder, "Analysis of obstacle detection with distance measuring using Arduino Uno and ultrasonic sensor," *International Research Journal of Engineering and Technology*, vol. 8, no. 11, pp. 1594-1599, 2021, https://www.irjet.net/archives/V8/i11/IRJET-V8I11259.pdf.

[36] J. Chaudhari, A. Desai and S. Gavarskar, "Line Following Robot Using Arduino for Hospitals," *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pp. 330-332, 2019, https://doi.org/10.1109/ICCT46177.2019.8969022.

[37] E. S. Pérez, and F. J. López, "An ultra-low cost line follower robot as educational tool for teaching programming and circuit's foundations," *Computer Applications in Engineering Education*, vol. 27, no. 2, pp. 288-302, 2019, https://doi.org/10.1002/cae.22074

[38] N. Gu, D. Wang, Z. Peng, J. Wang and Q. -L. Han, "Advances in Line-of-Sight Guidance for Path Following of Autonomous Marine Vehicles: An Overview," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 12-28, 2023, https://doi.org/10.1109/TSMC.2022.3162862.

[39] K. Lee, S. Jeon, H. Kim and D. Kum, "Optimal Path Tracking Control of Autonomous Vehicle: Adaptive Full-State Linear Quadratic Gaussian (LQG) Control," *IEEE Access*, vol. 7, pp. 109120-109133, 2019, https://doi.org/10.1109/ACCESS.2019.2933895.

[40] K. Karaahmetoğlu and Ö. Korkmaz, "The effect of project-based Arduino educational robot applications on students' computational thinking skills and their perception of basic STEM skill levels," *Participatory Educational Research*, vol. 6, no. 2, pp. 1-14, 2019, https://doi.org/10.17275/per.19.8.6.2.

[41] E. Lee, "A meta-analysis of the effects of Arduino-based education in Korean primary and secondary schools in engineering education," *European Journal of Educational Research*, vol. 9, no. 4, pp. 1503-1512, 2020, https://doi.org/10.12973/eu-jer.9.4.1503.

[42] M. N. H. Z. Alam, S. A. Samsudin, M. J. Kamaruddin, Z. Y. Zakaria, and R. Othman, "Introducing Arduino as an effective online distance learning tool in final year project for chemical engineering student," *ASEAN Journal of Engineering Education*, vol. 6, no. 1, pp. 69-82, 2022, https://doi.org/10.11113/ajee2022.6n1.88.

[43] F. M. Lopez-Rodriguez and F. Cuesta, “An android and Arduino based low-cost educational robot with applied intelligent control and machine learning,” *Applied Sciences*, vol. 11, no. 1, p. 48, 2020, https://doi.org/10.3390/app11010048.

[44] H. Gunes and S. Kucuk, “A systematic review of educational robotics studies for the period 2010–2021,” *Review of Education*, vol. 10, no. 3, p. e3381, 2022, https://doi.org/10.1002/rev3.3381.

[45] P. Peerzada, W. H. Larik, and A. A. Mahar, “DC motor speed control through Arduino and L298N motor driver using PID controller,” *International Journal of Electrical Engineering and Emerging Technology*, vol. 4, no. 2, pp. 21-24, 2021, https://ijeeet.com/index.php/ijeeet/article/download/94/76.

[46] S. C. Pande, S. S. Thakur, and C. S. Sharma, “IoT based 3 phase induction motor parameter monitoring and controlling,” *Journal of Emerging Technologies and Innovative Research*, vol. 8, no. 6, pp. 44-53, 2021, https://www.jetir.org/papers/JETIR2106560.pdf.

[47] A. S. Ahmed, H. A. Marzog, and L. A. Abdul-Rahaim, “Design and implement of robotic arm and control of moving via IoT with Arduino ESP32,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 3924-3933, 2021, http://doi.org/10.11591/ijece.v11i5.pp3924-3933.

[48] A. Ma'arif, and A. Çakan, “Simulation and Arduino hardware implementation of DC motor control using sliding mode controller,” *Journal of Robotics and Control*, vol. 2, no.6, pp. 582-587, 2021, https://doi.org/10.18196/jrc.26140.

[49] S. Saxena, “Android Guided Arduino Car,” *i-manager's Journal on Electronics Engineering*, vol. 14, no. 3, pp. 38-43, 2024, https://doi.org/10.26634/jele.14.3.20815.

[50] B. C. M. Henrique, L. C. M. Henrique, and H. M. Henrique, “Arduino based platform for process control learning,” *The Journal of Engineering and Exact Sciences*, vol. 6, no. 5, pp. 0585-0593, 2020, https://doi.org/10.18540/jcecvl6iss5pp0585-0593.

[51] N. Yusop, N. A. Moketar, and S. F. N. Sadikan, “Development of Arduino applications for IoT applications in software engineering education: A systematic literature review,” *Bulletin of Electrical Engineering and Informatics,* vol. 13, no. 3, pp. 1824-1831, 2024, https://doi.org/10.11591/eei.v13i3.4506.

[52] V. Pal, S. Kumar, V. Kumar, G. Jha, and P. Pant, “Voice controlled home automation system using Arduino,” *IITM Journal of Management and IT*, vol. 12, no. 1, pp. 102-108, 2021, https://www.indianjournals.com/ijor.aspx?target=ijor:iitmjmit&volume=12&issue=1&article=026.

[53] D. Chioran and H. Valean, “Arduino based smart home automation system,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, pp. 67-73, 2020, https://dx.doi.org/10.14569/IJACSA.2020.0110410.

[54] S. A. Ajagbe, O. A. Adeaga, O. O. Alabi, A. B. Ikotun, M. A. Akintunde, and M. Adigun, “Design and development of Arduino-based automation home system using the internet of things,” *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 33, no. 2, pp. 767-776, 2024, http://doi.org/10.11591/ijeecs.v33.i2.pp767-776.

[55] S. Kheman, P. S. Ramthirth, S. Chavan, S. D. Kaddaragi, and V. R. Kulkarni, “Design and development of an automatic color sorting machine on belt conveyor”, *International Journal of Research in Engineering Science and Management*, vol. 6, no. 6, pp. 15–17, 2023, https://journal.ijresm.com/index.php/ijresm/article/view/2722.

[56] M. Alagarsam, S. R. Devakadacham, H. Subramani, S. Viswanathan, J. Johnmathew, and K. Suriyan, “Automation irrigation system using Arduino for smart crop field productivity,” *International Journal of Reconfigurable and Embedded Systems,* vol. 12, no. 1, pp. 70-77, 2023, http://doi.org/10.11591/ijres.v12.i1.pp70-77.

[57] G. Sun and G. Bei, “Arduino-based intelligent handling robot design,” *Advances in Computer, Signals and Systems*, vol. 7, no. 1, pp. 67-74, 2023, https://dx.doi.org/10.23977/acss.2023.070109.

[58] Arduino, "Servo Library," *Arduino*, 2021, https://www.arduino.cc/reference/en/libraries/servo/.

[59] -Arduino, "Arduino Mega 2560 Datasheet," *Arduino*, 2024, https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf.

[60] Microchip Technology, "ATmega640/V-1280/V-1281/V-2560/V-2561 Datasheet," *Microchip Technology*, 2024, https://ww1.microchip.com/downloads/aemDocuments.

[61] B. Zain, S. Hassan, B. Mir, and R. H. Dar, "Robotic Hand Control using Hand Gesture Recognition for its Operational Behavior," *International Research Journal of Engineering and Technology*, vol. 6, no. 02, pp. 833-838, 2019, https://www.irjet.net/archives/V6/i2/IRJET-V6I2161.pdf.

[62] F. Salman, Y. Cui, Z. Imran, F. Liu, L. Wang, and W. Wu, "A Wireless-controlled 3D printed Robotic Hand Motion System with Flex Force Sensors," *Sensors and Actuators A: Physical*, vol. 309, p. 112004, 2020, https://doi.org/10.1016/j.sna.2020.112004.

[63] R. Sujatha, U. Hari, S. Suresh, D. Dastagiri, K. Sumanth, and B. N. Shareef, "System for Automatic Gate Control on Railways," *International Journal of Advances in Engineering and Management*, vol. 5, no. 4, pp. 501-505, 2023, https://ijaem.net/issue_dcp/System%20for%20Automatic%20Gate%20Control%20on%20Railways.pdf.

[64] T. Long, "Design of sweeping robot based on STM32 single chip microcomputer," *Journal of Physics: Conference Series*, vol. 2456, no. 1, p. 012045, 2023, https://doi.org/10.1088/1742-6596/2456/1/012045.

[65] M. K. Rihmi, G. Bintoro, M. A. Rahman, and G. Puspito, "Accuracy Analysis of Distance Measurement Using Sonar Ultrasonic Sensor HC-SR04 on Several Types of Materials," *Journal of Environmental Engineering and Sustainable Technology*, vol. 11, no. 1, pp. 10-13, 2024, http://dx.doi.org/10.21776/ub.jeest.2024.011.01.2.

[66] A. Sarada, B. Saritha, V. K. Likhitha, J. Himavarshini, and M. Akshethra, "Line Follower with Obstacle Avoiding Robot," *a et al., International Journal of Emerging Trends in Engineering Research*, vol. 11, no. 11, pp. 339-343, 2023, https://doi.org/10.30534/ijeter/2023/0311112023.

[67] Y. Zhang *et al.*, "Investigation of acoustic injection on the MPU6050 accelerometer," *Sensors*, vol. 19, no. 14, p. 3083, 2019, https://doi.org/10.3390/s19143083.

[68] N. Prasanth, K. Shrivastava, A. Sharma, A. Basu, R. A. Sinha, and S. P. Raja, "Gesture-based mouse control system based on MPU6050 and Kalman filter technique," *International Journal of Intelligent Systems Technologies and Applications*, vol. 21, no. 1, pp. 56-71, 2023, https://dx.doi.org/10.1504/IJISTA.2023.10055775.

[69] A. T. Abu-Jassar, H. Attar, V. Yevsieiev, A. Amer, N. Demska, A. K. Luhach, and V. Lyashenko, "Electronic user authentication key for access to HMI/SCADA via unsecured internet networks," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, pp. 1-13, 2022, https://doi.org/10.1155/2022/5866922.

[70] A. E. Amoran, A. S. Oluwole, E. O. Fagorola, and R. S. Diarah, "Home automated system using Bluetooth and an android application," *Scientific African*, vol. 11, p. e00711, 2021, https://doi.org/10.1016/j.sciaf.2021.e00711.

[71] InvenSense, "MPU-6000 and MPU-6050 Product Specification Revision 3.4," *InvenSense*, 2015, https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf.

[72] Arduino, "Arduino Nano Rev3.2," *Arduino*, 2024, https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf

[73] STMicroelectronics, "L298 Dual Full-Bridge Driver," *STMicroelectronics*, https://www.st.com/resource/en/datasheet/l298.pdf.

[74] SparkFun Electronics, "Flex Sensor 2.2 Datasheet," *SparkFun Electronics*, 2024, https://www.sparkfun.com/datasheets/Sensors/Flex/flex22.pdf.