Path Finding Radar Robot Car : Design and implementation

1st R. Sania
Department of Electronics and Communication Engineering
Jalpaiguri Government Engineering College
Jalpaiguri, India
rifasania458@gmail.com

2nd N. Rahaman
Department of Electronics and Communication Engineering
Jalpaiguri Government Engineering College
Jalpaiguri, India
norul.rahaman232002@gmail.com

***Abstract*—In this project we will be making an automated smart robot car which works on the principle of smart navigation and automated pathfinding. One of the main functions required for this automated mobile robot to work is obstacle avoidance. This project's goal is to focus on developing and implementing a smart robot car using an Arduino Uno based microcontroller, paired with ultrasonic sensors that enables real-time obstacle detection and path navigation. The robot is also capable of scanning its environment, detecting the obstacles within a certain given distance, and choosing the most convenient path available to reach its destination. Unlike basic obstacle-avoiding robots that simply turn away upon detection, this system attempts to evaluate multiple directions and prioritize forward motion using a decision-making algorithm. A robot is a machine that carries out tasks on its own or under supervision from an external force. So, here we are trying to implant a brain in our robot and for that purpose we will use Arduino software to write the program. This project proves that when low-cost components are integrated together, it can perform meaningful tasks related to autonomous navigation. While there are some limitations due to hardware constraints, the robot provides a useful foundation for further research into intelligent robotic systems and real-time decision systems in embedded platforms.**

***Index Terms*—Autonomous Robot Car, Obstacle Avoidance, Arduino UNO, Ultrasonic Sensor, Smart Navigation, Pathfinding Algorithm, Environment Scanning, Sensor-Based Robot.**

I. Introduction

Radar systems use the sending and receiving of radio waves to recognize and keep track of different objects. They give pertinent details on the position, range, velocity, and even the trajectory of different objects. [1]–[3] The applications of radar technology are vast and include, but are not limited to, military defense structures, weather forecasting, aviation, and maritime navigation. There are different categories of radar systems created with specific functions in mind, and each with marked features designed for particular actions. Some are made to collect data in real time with a high degree of accuracy, while others are better for long-range surveillance. In the past few years, a lot of better radar-like systems have been made. These systems can now do better motion analysis and object detection. Such technologies are commonly found in robots and autonomous vehicles. They inspired us to work on our project. We created a simpler, smaller version using commonly available components. An Arduino microcontroller which controls a servo motor and an ultrasonic sensor forms the basic structure of our system. The ultrasonic sensor measures distance by emitting and receiving sound pulses. It plays an essential role in the primary detection of obstacles. The servo motor moves the ultrasonic sensor from 0° to 180° and connects to the Arduino's digital pins. This setup allows the robot to scan the area in front of it. As the servo rotates, the sensor measures distance at various angles. The Arduino processes this information to determine which direction offers the most space. Consequently, the robot selects a safe path and avoids accidents. The robot can perform tasks when the servo and ultrasonic sensor work together. We plan to use a servo motor with an ultrasonic sensor mounted on it, controlled by an Arduino microcontroller. The ultrasonic sensor is the main tool for The Processing development environment shows the angles and distances on a computer screen in real time detecting things. It sends and receives sound pulses to find out how far away objects are. This software serves as a graphical user interface. It helps users understand how the robot senses its environment and decides where to go. The entire system demonstrates basic radar simulation. It also highlights important robotics concepts like integrating sensors, controlling motors, and choosing a path independently. This visual interface helps users see how the robot views its surroundings. As the servo motor turns the ultrasonic sensor, the distance readings at each angle appear on the screen right away, similar to a radar sweep. This lets users engage with the system and simplifies the steps for diagnosing problems and checking the sensor's accuracy. By turning raw distance measurements into visual plots, users can find blind spots, locate obstacles, and assess scanning efficiency. These insights help refine the robot's pathfinding logic. This method shows important radar-like features and promotes hands-on learning in embedded systems and robotics. The initiative links theoretical concepts with hands-on application by offering an engaging and informative experience. Through the integration of electronic components and logic-driven programming, it illustrates how basic hardware can display intelligent behavior when directed by clever algorithms. It also paves the way for advancements ahead, including SLAM (Simultaneous Localization and Mapping), AI-driven navigation, or the integration

Identify applicable funding agency here. If none, delete this.

of various sensors in robotic system.

II. State Of Art

Radar has a long, long history — it was actually first invented in 1886. [4]–[6] It was the then-physicist Heinrich Hertz who demonstrated that electromagnetic waves would penetrate certain materials, and reflect off metal. This key realization had been supplied by the theorizing of James Clerk Maxwell. Maxwell's equations of electromagnetic fields primed the pump for Hertz. Hertz demonstrated that such waves would reflect from metals, paving the way for later radar.

As we get into the twentieth century, they're more. German engineer Christian Hülsmeyer patented the telemobiloscope in 1904. Until then, he constructed to locate vessels in mist by reflected radio. So it was the first to do this in practice for object detection, but it didn't catch.

Through their 20s and 30s, scientists and engineers literally all over the world pushed out that experimental edge, expanding the catalog of applications for radio-based sensing systems. Italian physicist Guglielmo Marconi had proposed tuning for shortwaves all the way back in 1922. Everyone else made these inventions the sample models.

Radar was spotlighted with the advent of WWII. War ignited a radar research and deployment sprint. Mining urgency mining research and manufacturing. These endeavors solidified radar's position as a military and civilian staple.

The original was years ago. Radar's idea is from the late 19th century. By 1886 physicist Heinrich Hertz had demonstrated that radio waves could be transmitted and reflected from solid objects. His experiments validated James Clerk Maxwell's theoretical observations about the behavior of electromagnetic fields. Hertz's discovery that radio waves could be reflected off of metal gave radar its theoretical foundation. By the turn of the century, radar was starting to seem possible. One of the earliest was Christian Hülsmeyer, a German engineer who in 1904 patented what he termed the Telemobiloscope. This breakthrough looked for radar's pings of ships, sidestepping fog crashes. That was the first known experiment in using radio echoes to remotely sense objects — the foundation of contemporary radar. Development proceeded in the 20s and 30s, as engineers and scientists across the globe dicked around with techniques to use radio waves to track down stuff. In 1922, Italian physicist Guglielmo Marconi proposed employing shortwave radio for this, observing that these shorter wavelengths offered greater resolution. A few countries test drove these ideas, skeining up level by level from idea to working prototype. would be the basis for the radars we depend on today. By the time WWII came around, the potential of radar as a weapon was well appreciated. Allied powers like Britain, Germany and the US invested enormous resources to radar advancement, enabling it to thrive in distance, precision and transportability. Radar spotted foe planes, guided ships and alerted attacks. The war effort did more than a technological jolt. It trained thousands of engineers and technicians who would develop peacetime applications for radar in the years to come.

Post-war radar found a multitude of civilian applications. It was adopted by aviation for air traffic control, in weather for storm tracking and at sea for navigation and collision avoidance. Radar's trajectory from crude science to ubiquitous detection and communication tool underscores the magic mix of scientific curiosity, engineering pragmatism, and societal necessity. Today radars haven't stopped innovating — with DSP, automation, and integration into autonomous vehicles and robots, such as our project's version of affordable, intelligent radar-esque sensing.

III. Block Diagram

A. System Architecture and Component Interfacing

The robotic system is designed using the Arduino UNO as the central microcontroller, interfaced with multiple hardware modules including a motor driver, sensors, and a Wi-Fi-enabled ESP32 module. The system supports both autonomous obstacle detection and manual control through a web-based interface.

1) Arduino UNO as Central Controller: The Arduino UNO serves as the main controller, interfacing with the following components:

- **L293D Motor Driver Shield:** Mounted on the Arduino UNO, it is responsible for driving two DC motors that control the robot's wheels for movement in various directions.
- **Ultrasonic Sensor (HC-SR04):** Connected with Trig to Digital Pin 9 and Echo to Digital Pin 10, this sensor provides non-contact distance measurement. It helps in detecting obstacles in the robot's path.
- **Servo Motor:** Connected to Digital Pin 6 (PWM), it is used to rotate the ultrasonic sensor, allowing environmental scanning in multiple directions.
- **ESP32-WROOM-32E:** Serially connected to the Arduino UNO using TX and RX pins. It enables wireless communication for command reception and data transmission between the robot and a remote interface.

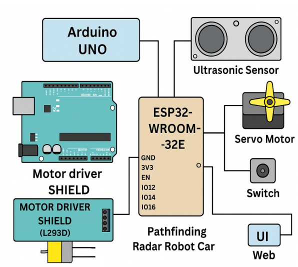

Fig. 1. Block Diagram

2) ESP32-WROOM-32E for Web UI and Communication: The ESP32 module handles network operations and hosts a responsive Web UI. It operates in two possible modes:

- **Access Point (AP) Mode:** The ESP32 creates its own Wi-Fi network, allowing users to connect directly for local control.
- **Client Mode (STA):** The ESP32 connects to an existing Wi-Fi network (router), allowing the robot to be controlled from any device within the same network.

Key responsibilities of the ESP32 include:

- Hosting an interactive web interface for manual control.
- Sending directional commands (e.g., `F` for forward, `L` for left) to the Arduino via Serial.
- Receiving and displaying sensor data (e.g., distance from obstacles).

3) Power Management: The system is powered through a unified control switch that simultaneously powers both the ESP32 and the motor driver shield. The power supply consists of:

- A battery pack (e.g., 9V or 12V) for Arduino and motor shield.
- On-board voltage regulation to safely power the 3.3V ESP32 module.

4) Operational Workflow: The high-level operational sequence of the robot is as follows:

1) Upon activation, both Arduino UNO and ESP32 are powered on.
2) ESP32 initializes in AP or client mode and hosts the Web UI.
3) Users connect via smartphone or PC to the ESP32's IP address and access the control interface.
4) Commands issued from the UI are sent to Arduino via serial communication.
5) Arduino processes the commands and drives the motors or updates sensor data accordingly.
6) Feedback from sensors is transmitted back to ESP32 and reflected on the Web UI.

This architecture offers a modular, flexible, and scalable design for implementing semi-autonomous and manually controlled robotic systems with real-time feedback capabilities.

IV. Working Principle

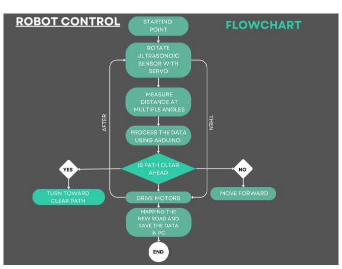

Fig. 2. Flow Chart

The Path Finding Radar Robot Car works by continuously scanning its surroundings using an ultrasonic sensor which is adjusted on top of the servo motor, simulating a radar system. When the car starts, the servo motor rotates the sensor to measure distances at multiple angles, creating a panoramic view of the environment. [16], [17] This data is processed by the Arduino kit, which checks whether the path ahead is clear or not. If it detects an obstacle, then it takes a step back, then determines which direction has more open space and turns to to that direction. If the path is clear, it moves forward and if not then it repeats the same actions again and again until it has found a suitable path for reaching it's destination.

V. Hardware Components

A. Arduino UNO

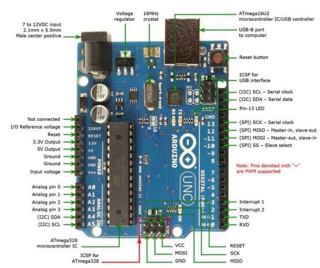

Fig. 3. Arduino UNO

Arduino is an open-source platform [18] for electronic prototyping designed for users with limited experience. electronics and programming experience. We utilized the Arduino UNO, the most famous Arduino replica. The Arduino Uno is a microcontroller board primarily built on the ATmega328 and features its own integrated development environment (IDE) for writing, compiling, and uploading programming codes to the microcontroller. Various sensors will supply environmental data as input to the microcontroller, which will send output to connected peripherals, such as actuators. It has 28 pins — 14 digital in/out pins (6 PWM), six analog — used for connecting to electronic components like sensors and motors, 3 GND, 3.3 V, 5 V, VIN, RESET, and AREF. Arduino's microcontroller has 32 KB of memory, 2 KB of SRAM, and 1 KB of EEPROM. Arduino mainly provides support for a C compiler, macro-assemblers, and evaluation kits. It features 16 MHz ceramic resonators, a USB port for computer connection, a jack for external power supply, an ICSP (in-circuit serial programmer) header, and a reset button to restore factory settings. It operates with an input of 7 to 12 V (maximum limit of 20 V)

B. Espressif ESP32-WROOM-32E

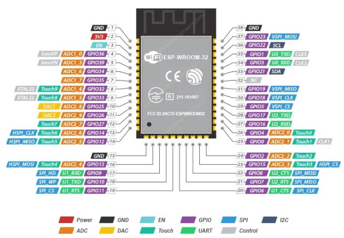

Fig. 4. Espressif ESP32-WROOM-32E

The ESP32-WROOM-32E is a powerful Wi-Fi and Bluetooth-enabled microcontroller module from Espressif, designed for IoT and embedded applications [19], [20]. It features a dual-core 32-bit processor, integrated antenna, and low-power operation ideal for robotics. In this project, it enables real-time wireless control, sensor data collection, and web dashboard communication for a path-finding robot.

C. Motor Driver Shield(L293D)

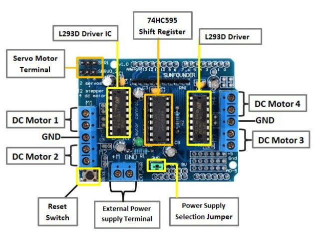

Fig. 5. Motor Driver Shield(L293D)

The L293D consists of quadruple high-current half-H drivers, used to manage the direction and speed of as many as 4 DC motors with individual 8-bit speed selection, concurrently supporting up to 0.6 A. It is designed to supply bidirectional drive currents of up to 600 mA at voltage levels ranging from 4.5 to 36 V. There are eight output pins available for connecting 4 DC or 2 Stepper motors, 1 reset switch, 6 pins for connecting two servo motors, 1 +M pin for external motor power, 5 analog inputs, and 13 digital inputs for interfacing with Arduino along with ground pins.

D. Ultrasonic Sensor

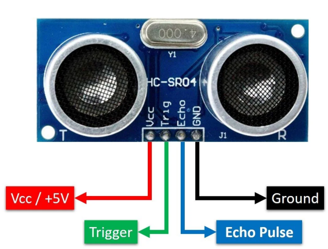

Fig. 6. Ultrasonic Sensor

This HC-SR04 ultrasonic sensor [21], [22] (similar to the one shown in figure 2) utilizes SONAR to measure the distance to an object, much like a bat does. Providing excellent non-contact distance detection, the VL6180X delivers accurate, dependable range data in an easy-to-connect design, measuring from 2 to 400 cm or 1 to 13 feet. It isn't affected by sunlight or dark fabric, though from an acoustic perspective, soft materials like those can be challenging. This contains an ultrasonic transmitter and a receiver module. The ultrasonic sensor operates similarly, but it uses sound reflection to determine the time difference between the emitted wave and the received wave. It typically emitted a pulse at the transmission site and monitored for the echoes of the reflected pulses. The duration measured is subsequently applied with the typical speed of sound in air (340 ms-1) to calculate the distance from the sensor to the obstacle. The Ultrasonic sensor, utilized by numerous researchers, detects the movements of nearby objects.

E. DC Gear Box Motor

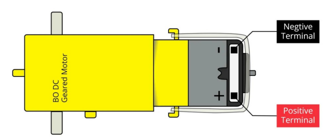

Fig. 7. DC Gear Box Motor

A TT motor [23] is a kind of DC motor that is equipped with a gearbox. The gearbox diminishes the motor's speed while amplifying its torque. A TT motor is frequently utilized in applications like powering wheels, propellers, fans, and more. A TT motor consists of a positive wire and a negative wire. Typically, the positive wire is red while the negative wire is black. The product utilizes a TT DC gearbox motor featuring a gear ratio of 1:48, and it includes two 200mm wires with 0.1" male connectors compatible with a breadboard. Ideal for connecting to a breadboard or terminal block. These motors can be operated within a range of 3 to 6VDC, although they will run somewhat faster when supplied with higher voltages. A TT motor takes electrical energy and transforms it into movement. Applying a voltage to the motor's wires generates a magnetic field and spins the motor. The motor spins at a speed and direction proportional to the voltage and polarity of whatever power source you hook to it. The more voltage, the quicker the motor turns. If you reverse the polarity it's going to go the other way.

F. Servo Motor

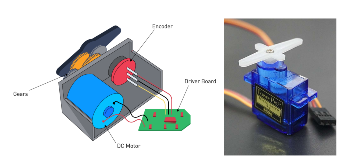

Fig. 8. Servo Motor

An ordinary SG90 servo motor [24] weighs just 9g, which simplifies operations and enables accurate rotation using Pulse Width Modulation (PWM). A micro servo motor features stator and rotor windings on its fixed and moving sections, respectively. Besides consuming less power, this motor is highly efficient and easy to service. Driver amplifiers power servo motors, receiving velocity commands from controllers. Feedback mechanisms like resolvers and encoders deliver data regarding the velocity and location of servo motors. The resolver or encoder can be integrated into the motor, or it might be located externally. Closed-loop systems enable servo motors to operate according to a specific motion profile that is programmed into the controller. The SG90 Servo motor comes with three pins: Vcc, GND and PWM. Servo motors are prevalent in robotics for precisely this reason — they can provide the precise positioning required for certain functions. The SG90 is made up of four components: electronic assembly, protective casing, drive gears, and output spline. Additionally, check out our blog on the Working Principle of AC Motor, which details the operation of AC motors, includes diagrams, and describes various types of AC motors and their functionalities.

G. Jumper Wires

Fig. 9. Jumper Wires

A jumper wire (commonly referred to as a jumper, jumper wire, or DuPont wire) is a type of electrical wire, or a collection of them in a cable, with a connector or terminal at both ends (or occasionally without them – just ”canned”), which is typically employed to link the elements of a breadboard or another prototype or experimental circuit, internally or alongside other devices or parts, without welding.

H. Switch

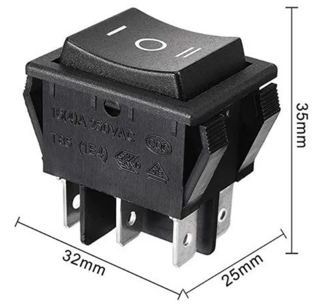

Fig. 10. Switch

A switch is utilized to activate the vehicle’s power. It manages the battery link of the vehicle.

VI. Software

A. UI Web

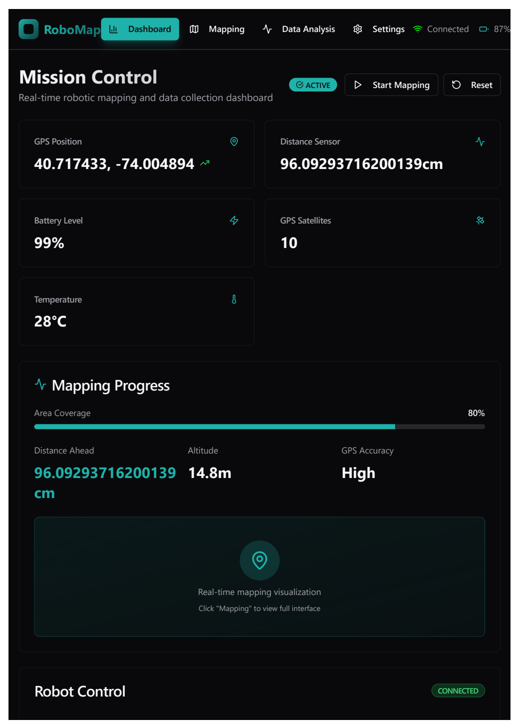

Fig. 11. UI Web

This project details the development of a real-time robotic mapping and data collection system leveraging an ESP32-WROOM-32 microcontroller. [25] The robot actively transmits vital information to a live web dashboard for real-time monitoring and visualization, including GPS coordinates, distance measurements, and other sensor readings. Although the system’s autonomous and manual control modes provide operational flexibility, features like real-time path visualization, battery status indicators, and CSV data export capabilities significantly enhance usability and facilitate thorough post-mission analysis.

VII. Equation

A. Ultrasonic Sensor Distance Calculation

The ultrasonic sensor calculates the distance using the time delay between transmission and reception of sound waves. [26], [27]

$$\text{Distance (cm)} = \frac{\text{Time} \times 0.0343}{2}$$

Where:

- **Time** is the time taken for the echo to return (in microseconds, μs).
- 0.0343 cm/μs is the speed of sound in air.

B. Motor Speed using PWM Control

DC motor speed is controlled by the average voltage delivered via PWM.

$$V_{\text{avg}} = V_{\text{max}} \times \frac{\text{Duty Cycle}}{100} \tag{1}$$

C. Robot Linear Velocity

The linear velocity of the robot is calculated based on wheel RPM and radius.

$$v = r \times \omega = r \times \frac{2\pi \times \text{RPM}}{60} \tag{2}$$

Where:

- v is linear velocity (m/s)
- r is wheel radius (in meters)
- ω is angular velocity in radians per second
- V_{avg} is the average voltage applied
- Duty Cycle is in percentage

D. Servo Motor Angle Update

The servo angle is updated in defined steps during scanning.

$$\theta_{\text{new}} = \theta_{\text{prev}} + \Delta\theta \tag{3}$$

Where:

- $\Delta\theta$ is the step angle (e.g., 15°)

E. Path Decision Logic (Algorithmic)

The following logic controls the robot's path based on sensor input distances. It selects movement direction by comparing the center, left, and right ultrasonic sensor values.

Algorithm 1 Path Decision Logic Based on Ultrasonic Sensor Readings

Distance readings: D_C (Center), D_L (Left), D_R (Right) Robot Movement Direction FMainDecidePath FnFunction: D_C, D_L, D_R, Threshold $D_C >$ Threshold Move Forward $D_L > D_R$ Turn Left Turn Right

F. Wi-Fi Communication Equations

In our path-finding radar robot car, we use Wi-Fi (via ESP32/ESP8266) for real-time communication and control. The following theoretical equations are relevant to evaluate and describe the wireless data transmission system.

1) Shannon–Hartley Theorem (Maximum Data Rate): The maximum achievable data rate C of a communication channel with bandwidth B and signal-to-noise ratio SNR is given by:

$$C = B \cdot \log_2(1 + \text{SNR}) \tag{4}$$

Where:

- C: Channel capacity (bits per second)
- B: Channel bandwidth (Hz)
- SNR: Signal-to-noise ratio (unitless)

2) Friis Transmission Equation (Signal Strength): To estimate how the signal strength decays with distance, the Friis free-space equation is used:

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} \tag{5}$$

Where:

- P_r: Received signal power (W)
- P_t: Transmit power (W)
- G_t, G_r: Antenna gains of transmitter and receiver (unitless)
- λ: Wavelength ($\lambda = \frac{c}{f}$)
- d: Distance between transmitter and receiver (m)
- L: System loss factor (unitless)

3) Transmission Latency: The time taken to transmit a data packet is:

$$\text{Latency}_{\text{tx}} = \frac{\text{Packet Size (bits)}}{\text{Transmission Rate (bps)}} \tag{6}$$

4) Throughput: The effective data rate (throughput) achieved during transmission is:

$$\text{Throughput} = \frac{\text{Total Data Received}}{\text{Total Time Taken}} \tag{7}$$

These equations help analyze the performance and constraints of Wi-Fi communication in our robotic car system.

VIII. Result

Here we introduce the concept and prototype of a low-cost autonomous robotic vehicle that can intelligently explore via radar-like sensor scanning. Radar mimicry with an arduino UNO-based setup and servo-mounted ultrasonic sensor. The really difficult task, of course, is to create a clever robot that can wander and avoid hazards in real-time, taking in sensor information and determining which direction to pivot.

Bringing SRS into radar 101, the article follows it back to early theorists Maxwell and Hertz, to its military and civilian uses. Ideal situation for a uber-tiny, student-affordable, radar navigation system. The impetus is new robotics traction and surging demand for real-time autonomous path planning in cluttered environments.

The robot's architecture is highlighted in a block diagram, emphasizing the connection between the central pieces — the L293D motor driver, an ultrasonic sensor, a servo motor and the ESP32 module wireless communication. An Arduino UNO acting as the hub of sensor input and motor movement. Robot overlays servo-driven scanning ultrasonic distance for panoramic sensing and intelligent navigation

Toss in a slick, custom, web ui displaying the sensor data in real time, gps coordinates, battery level, distance, mapping, etc. This adds interactivity and attention.

What the report really underscored was how educational the project was, offering first-hand experience with embedded systems, control logic and sensor integration. It describes all of the hardware and software with real-world advice on component selection and interfacing.

Unless it's equations, the paper is a glorious mashup of theory and practice that culminates in a potent scalable infrastructure for autonomous navigation. This gets us ready for the modeling and performance analysis in the next section.

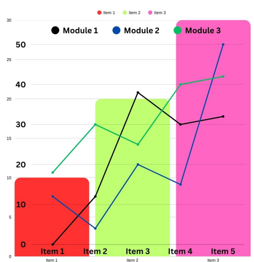

Fig. 12. Bar Graph

The graphical comparison displays the performance of three functional modules in the robotic car system:

- Module 1 (Path Finding): Represented in black, this module does progressively better going from Item 1 to Item 5. The early ones (1,2) do worse, maybe implying optimization, learning or simpler experiments. With System rising from 3 – 4, and peaking at 5, the strong momentum suggests it might customize its algorithms. This upgrade is paramount as it dictates the car's ability to sense and circumvent obstacles with its onboard sensors (i.e. LiDAR, ultrasonic, cameras). The persistent growth indicates an intense, growing barrier sensor and avoidance system — essential for self-driving cars.
- Module 2 (Web UI and Data Collection) – indicated by the blue line, this module behaves inconsistently throughout the Item. Possible metrics for this module could be latency of data to the UI, bandwidth required to collect it, web front end responsiveness, or data collected/processed per unit time. The swings are a reflection of changes in data volume and interface activity. This might be because in some phases, the car was generating more sensor data (i.e. sharp/complex maneuvers or richer environments) or the web UI was being more heavily stressed resulting in these performance swings. Even if it's ping-ponging, the module is pretty active overall, so it appears to be taking in information and user requests.
- Module 3 (Car Road Movement) – Demonstrated by the green line, this module typically retains a higher baseline across all phases, achieving its peak at Item 5. These measurements would presumably relate to accuracy of steering, velocity, balance, or path adherence. Its consistent dominance through Items 1-5 shows that the module's sweet spot is in making car movement and navigation smooth, reliable and precise. Such solid Tier 1-level performance is crucial, the physical reality for the car's driverless aspirations, demonstrating its robust hardware and control systems.

This drill-down of the three modules, detailed in the chart, is a tribute to the robotic car unit's successful execution. The car has proven capable of facilitating dependable tangible momentum (Module 3), its bedrock. Meanwhile it smartly collects and preprocesses required sensor data (Module 1) for smart decisions, while managing evolving interface and data requirements (Module 2). With the breakout of Module 3, supported by the fresh and volatile inputs of modules 1 and 2, the robotic car architecture offers more than a component hash. It is not rather, an actual, embodied, agential subject. This amalgamated action is essential for the bot to navigate intricate environments safely and prosperously.

IX. Conclusion

To us, the necessity of the Internet [28] and Internet-based technologies is extremely significant at present. The Internet of Things, or IOT, plays a crucial role in both computers and our everyday life. The model outlined above explains how the Arduino program functions. car engine unit and, as a result, we effectively spin the wheels and provide direction to the vehicle. It provides us the chance to engage with various platforms, aiding us in creating. different engaging modules to focus on. We additionally evaluated the applications utilized to operate the vehicle. Owing to the innovative idea of a Wireless Controlled Car utilizing Bluetooth and Wifi, we were capable of generating several potential scenarios that could occur. Therefore, in this project, we created an Android application using the software Android Studio to manage the RC module. We have coded the Arduino and created the RC car as illustrated in the diagram above. The vehicle will obtain instructions through Bluetooth and act accordingly. The Arduino is a publicly available device that has served as the brain for many projects. The Arduino contains all the necessary components for the user, including its integrated converter, input/output pins, etc. By integrating Arduino with the Bluetooth Shield, we can manage various other elements, such as home lighting, air conditioning, and much more via our mobile devices. The Arduino can be a huge help to the Smart Home development. Ok For example. Through this Project, we learned a great deal about the Arduino and its capabilities. enabled us to simplify the transformation of digital signals into physical actions. Another benefit One advantage of Arduino is that after a program is uploaded, there's no need to think about it again. being eliminated as long as it is not RESET. Arduino surpasses all other microcontroller due to its effectiveness and ease of use. This initiative involves a simple prototype of a Bluetooth-controlled vehicle mentioned earlier. The A prototype vehicle is able to identify user commands and can accurately turn left, right, or stop. Different sensors can be utilized to enhance it further, such as ultrasonic or infrared and featuring different coding levels. Our Bluetooth remote car operates within a distance of 10-20 meters using mobile Bluetooth. regulating system. The distance primarily relies on the transmission strength of the receiver.

X. Future Scope

We turn our newest iteration of the radar-guided path-finder bot into a testbed for autonomous navigation in close quarters. It's a long way from making it more flexible, smart and usable in real life. Perhaps the most obvious potential upgrade is adding additional sensors. If you supplement with infrared sensors or cameras or lidar, the robot would have a more accurate and multi-dimensional understanding of its environment both for obstacle avoidance and for mapping.

Another major thrust is machine learning. [29]–[31] With the appropriate training data, the robot could be trained to recognize common objects or terrain types and act more contextually. This would be most helpful in semi-structured or dynamic environments, where encoded rules and simple distance metrics are ill-suited to the safety issue. In this system, real-time learning and adaptive responses would replace crude condition based logic.

By adding GPS and wireless modules, the robot could potentially function over longer outdoor distances and transmit data back to a server or phone. and would prove useful in agriculture, farming or remote sensing. Interfacing with IoT platforms may allow for centralized control, remote diagnostics and even coordination among autonomous units.

We can bring the current Processing visualization to a complete UI with logging, analytics and path simulation. Paired with SLAM, the robot could create live maps of novel environments and update them on the fly for safer, more efficient navigation.

As robotics advance, power efficiency and power management will become more crucial. Smart batteries, solar charging modules, or energy-optimized driving algorithms — anything that could increase the robot's range and trustworthiness. With a few minor design innovations and modular hardware, this robot may even operate in crushing or uneven terrain where current designs would struggle.

For the long term, this could be a project to help develop smart swarm robotics — a flock of robots sharing information and swarm intelligence. Systems like this show huge promise for disaster recovery, military scouting, industrial automation, and space exploration. Well, the radar robot car, primitive as it is right now, is the future in a brave new world of wiser-than-thou widgets.

XI. Acknowledgment

We would like to express our special thanks of gratitude to our Supervisor **Dr. Jayendra Kumar** as well as Director **Dr. Goutam Sutradhar** who gave us the golden opportunity to do this wonderful innovative project on the topic (Path Finding Radar Robot Car : Design and implementation), which also helped us in doing a lot of Research and we came to know about so many new ideas and different views to solve things. Secondly we would also like to thank our parents and friends who helped us a lot in finishing this project within the limited time.

References

[1] Shreyes Mehta, Shashank Tiwari, *Radar System Using Arduino and Ultrasonic Sensor*, International Journal of Novel Research and Development (IJNRD), 3(4), pp. 14–20, 2018.

[2] Prachi Gupta, Aman Singh, Aviral Kumar, V. V. Mehtre, *Radar System Using Arduino and Ultrasonic Sensor*, Iconic Research and Engineering Journals, 6(1), pp. 74–80, 2022.

[3] Prerna Kharate, Tanya Hamav, Santoshi Pote, *Radar System Using Arduino and Ultrasonic Sensor*, Journal of Emerging Technologies and Innovative Research (JETIR), 10(8), pp. 1–7, 2023.

[4] T. P. Rajan, K. K. Jithin, K. S. Hareesh, C. A. Habeeburahman, A. Jithin, *Range Detection Based on Ultrasonic Principle*, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), 3(2), pp. 7638–7643, 2014.

[5] O. V. Amondi, *Collision Avoidance System*, University of Nairobi Dissertation, 2009.

[6] U. Papa, G. D. Core, *Design of Sonar Sensor Model for Safe Landing of a UAV*, Proceedings of IEEE Metrology for Aerospace, pp. 346–350, 2015.

[7] R. Praveen Kumar, M. Divya, *Ultrasonic Radar System for Object Detection*, Futuristic Trends in IoT, IIP Series, Vol. 3(1), pp. 58–62, 2024.

[8] Ahmad Suaif, Ferizandi Q. Gani, *Implementation of Ultrasonic Sensor as a Radar for Obstacle Detection*, Andalas Journal of Electrical and Electronic Engineering Technology, 1(1), pp. 12–15, 2021.

[9] P. Divya, N. Bhavana, M. George, *Arduino Based Obstacle Detecting System*, International Conference of Advance Research & Innovation (ICARI), pp. 1–4, 2020.

[10] Anonymous, *Exploring the Potential of Arduino and Ultrasonic Sensor Integration*, International Research Journal on Advanced Engineering and Humanities (IRJAEH), 2023.

[11] Allan Zapata, *Speed Radar Using HC-SR04 and Arduino: Calibration and Data Analysis*, arXiv preprint, 2025.

[12] D. B. Kadam, Y. B. Patil, K. V. Chougale, S. S. Perdeshi, *Arduino Based Moving Radar System*, International Journal of Innovative Science and Smart Engineering Technologies (IJISSET), 3(4), pp. 23–27, 2017.

[13] Anonymous, *Radar System Using Arduino and Ultrasonic Sensor*, International Journal of Creative Research Thoughts (IJCRT), 12(4), pp. 1–8, 2024.

[14] P. S. Abhay, S. K. Akhilesh, P. Amrit, Kriti, *A Review on Ultrasonic Radar Sensor for Security System*, Journal of Emerging Technologies and Innovative Research (JETIR), pp. 137–140, 2016.

[15] Ahammad, G. R. Rohman Shafi, *Enhancing Surveillance Systems with Cost-Effective Ultrasonic Radar for Object Detection and Tracking*, Conference Proceedings, 2024.

[16] Anonymous, *Arduino RADAR using Ultrasonic Sensor for Detection and Range*, International Journal of Engineering and Scientific Research (IJESR), 15(2), pp. 45–53, 2023.

[17] G. Bhor, P. Bhandari, R. Ghodekar, S. Deshmukh, *Mini Radar*, International Journal of Technical Research and Applications, pp. 68–71, 2016.

[18] M. Banzi, M. Shiloh, *Getting Started with Arduino*, 3rd ed., Maker Media Inc., Sebastopol, CA, pp. 1–124, 2014.

[19] P. P. Arun, M. A. Sudhakar, P. Megha Sunil, S. S. Balaji, *Ultrasonic Distance Meter*, SVERIAN Scientific, pp. 1–4, 2015.

[20] V. Singh, P. Agarwal, *Integration of Ultrasonic Radar with Wi-Fi Based Data Logging on ESP32*, International Journal of Internet of Things, 5(4), pp. 200–207, 2022.

[21] C. Y. Zhao, P. Y. Li, C. M. Zhang, *Numerical Analysis of Effective Refractive Index Ultrasonic Sensor*, arXiv preprint, 2022.

[22] Y. Chen, H. Li, B. Zhao, *Smart Obstacle Detection in Robotics using Multiple Ultrasonic Sensors*, Robotics and AI Review, 7(1), pp. 34–41, 2023.

[23] A. Johnson, R. Kumar, *Design and Analysis of DC Gear Motors for Mobile Robotics*, International Journal of Mechanical and Robotics Engineering, 6(2), pp. 45–52, 2019.

[24] Mohammed H. AlSharif, Mohamed Saad, T. Y. Al-Naffouri, *Hand Gesture Recognition Using Ultrasonic Waves*, arXiv preprint, 2017. H

[25] A. J. Devlin, M. R. Hasan, *ESP32 Microcontroller Applications in IoT and Robotics*, Journal of Embedded Systems and Applications, 9(1), pp. 34–42, 2021.

[26] Shamsul A., Tajrian M., *Design of an Ultrasonic Distance Meter*, International Journal of Scientific and Engineering Research (IJSER), pp. 1–10, March 2013.

[27] Zhiqing Wei, Fengkai Zhang, Shuo Chang, Yangyang Liu, Huici Wu, Zhiyong Feng, *MmWave Radar and Vision Fusion for Object Detection in Autonomous Driving: A Review*, arXiv preprint, 2021.

[28] Arduino Project Hub, *Ultrasonic Radar with Arduino*, 2021.

[29] J. Liu, K. Zhang, *SLAM and Sensor Fusion Techniques in Autonomous Robot Navigation*, International Journal of Robotics Research, 17(3), pp. 112–120, 2022.

[30] J. Li, W. Chen, L. Zhou, *Arduino-Based Obstacle Avoidance Robot with Ultrasonic Sensor*, International Journal of Advanced Robotic Systems, 16(1), pp. 1–8, 2019.

[31] C. Patel, D. Shah, *Low-Cost Ultrasonic Radar System for Indoor Mapping*, Journal of Sensor Technology, 10(2), pp. 64–72, 2020.