CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The advancement of embedded systems and sensor technology has significantly transformed the way automation and surveillance applications are implemented. Among these, radar systems have played a pivotal role in object detection, distance measurement, and real-time monitoring.

However, conventional radar systems based on radio frequencies are costly this project presents a radar system based on **ultrasonic sensing technology** and powered by the **Arduino microcontroller**.

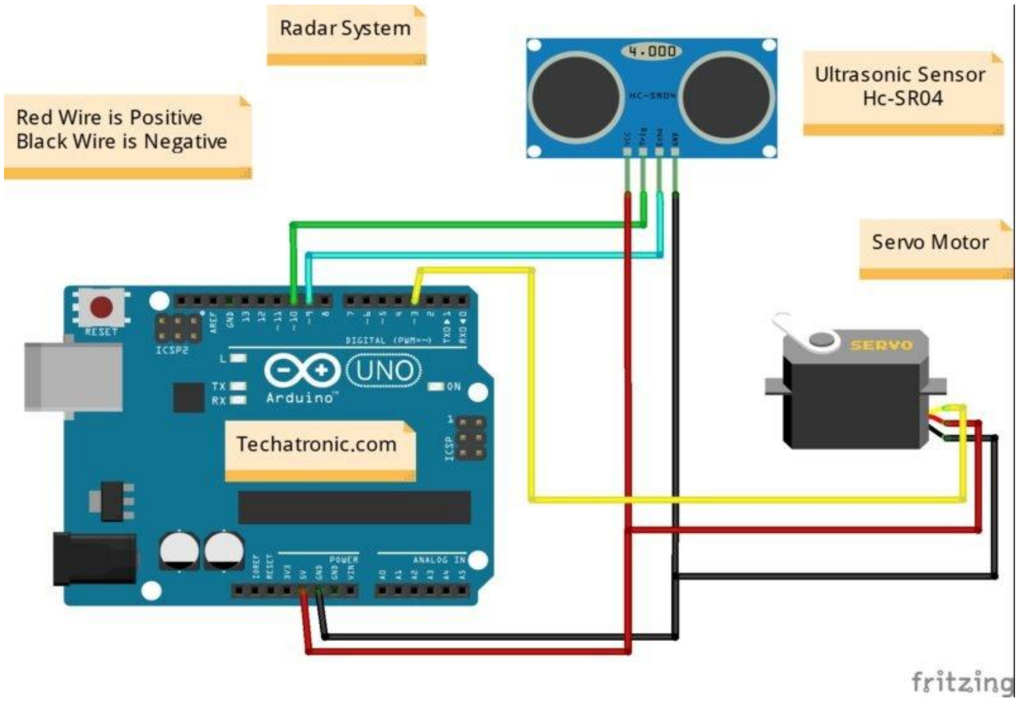

Figure 1: Arduino microcontroller sensor

Ultrasonic sensors function by emitting high-frequency sound waves and measuring the time it takes for the echo to return after bouncing off an object. This basic principle of time-of-flight enables accurate measurement of distances in real-time. The system developed in this project utilizes the HC-SR04 ultrasonic sensor, a servo motor for angular movement, and Arduino to control the operations and data acquisition. Additionally, the data captured by the sensor is processed and visualized using Processing software, which emulates a radar display on a computer screen. This integration of hardware and software not only enhances the usability of the system but also demonstrates a practical application of sensor networks and embedded systems.

1.2 MOTIVATION

The key motivation behind this project stems from the need for low-cost, scalable, and educational radar systems that can be used in academic environments, robotic projects, and small-scale security applications. While industrial-grade radar solutions offer precision and range, they often fall outside the scope of budget-friendly and DIY electronics. Ultrasonic technology, by contrast, offers a simple yet effective method of detecting obstacles within a moderate range and has become a staple in robotics and autonomous navigation.

Moreover, as STEM education continues to emphasize hands-on learning and real-world applications, this project aligns well with educational goals by integrating disciplines such as electronics, programming, and signal processing. It allows students and hobbyists to understand real-time system control, sensor integration, and data visualization, all within a framework that is both practical and affordable.

1.3 OBJECTIVE OF PROJECT

The primary objective of this project is to design and implement a functional ultrasonic radar system using Arduino that can accurately detect objects within a semi-circular scan area. The system should be capable of: Continuously scanning a 180-degree area using a rotating servo motor. Emitting ultrasonic pulses and measuring the time taken for the echo to return. Calculating the distance of any detected objects based on the speed of sound. Sending the distance and angle data to a computer via serial communication. Visualizing the detection results in the form of a radar screen using Processing IDE.

Secondary objectives include ensuring system stability, minimizing detection errors, and providing an intuitive user interface. Emphasis is also placed on keeping the system modular and adaptable for future upgrades such as wireless communication, mobile platforms, or object classification.

1.4 SCOPE OF THE PROJECT

This project is limited to detecting stationary objects within a range of approximately 2 to 400 centimeters, using the HC-SR04 sensor. The detection area is confined to a 180-degree sweep controlled by a standard servo motor. The system does not currently support moving targets, tracking beyond the scanning zone, or signal filtering for noise reduction. However, it is designed to be modular, allowing for such features to be incorporated in future versions.

The software component is built using Arduino IDE for hardware programming and Processing IDE ease of integration, and extensive community support.

1.5 ORGANIZATION OF THE THESIS

This document is structured to guide the reader through each stage of the project's development.

- **Chapter 1** introduces the project, explaining its relevance, motivation, and overall scope.
- **Chapter 2** delves into the theory and application of ultrasonic sensors, including the principles of sound propagation, time-of-flight measurement, and the design specifics of the HC-SR04.
- **Chapter 3** focuses on the system design and development, covering hardware selection, circuit schematics, and radar logic implementation.
- **Chapter 4** details the software implementation including the Arduino programming, servo control, data acquisition, and Processing-based GUI visualization.
- **Chapter 5** presents the results obtained from system testing, performance analysis, object detection scenarios, and screenshots from the radar interface.
- **Chapter 6** concludes the project, summarizing the findings and discussing avenues for future enhancement, such as adding wireless capability or improving detection algorithms.

1.6 SUMMARY

The motivation section emphasized the growing demand for cost-effective object detection solutions in both educational and practical domains. Ultrasonic sensors, particularly the HC-SR04, combined with microcontrollers like Arduino.

The objectives were clearly defined—to design and implement a functional radar system capable of detecting objects within a range, rotating via a servo motor to sweep across an angular field, and transmitting that data for real-time graphical visualization. These goals were established not only as technical benchmarks but also as learning opportunities for those seeking to understand embedded systems and sensor technologies.

The scope of the project was realistically constrained to focus on indoor object detection using a single-point ultrasonic sensor, while highlighting opportunities for modular upgrades and future enhancement. This helps to manage expectations regarding performance while also inspiring future exploration and scalability, such as 360° scanning, multi-sensor fusion, or wireless communication.

The thesis organization was outlined to provide clarity on how the content will be structured across subsequent chapters. It ensures that the reader is guided logically—from understanding sensor technologies, through design and implementation, to the evaluation of results and potential for future innovation.

In summary, this chapter does not just define what the project is about—it also frames why it is important, how it will be approached, and what it ultimately aims to deliver.

CHAPTER 2
ULTRASONIC SENSOR TECHNOLOGY

2.1 INTRODUCTION

Ultrasonic sensing technology has gained significant traction in various engineering and automation domains due to its reliability, cost-effectiveness, and non-contact nature. Unlike optical or infrared sensors that can be influenced by environmental light conditions, ultrasonic sensors function effectively across a variety of lighting environments. This makes them particularly useful for detecting objects, measuring distances, and performing spatial assessments in robotics, industrial automation, automotive parking systems, and security systems. In this chapter, the core principles of ultrasonic distance measurement are explored, followed by a comprehensive examination of the HC-SR04 sensor, its internal operation, signal timing, and its role in this radar-based application.

2.2 PRINCIPLE OF ULTRASONIC DISTANCE MEASUREMENT

Ultrasonic distance measurement relies on the time-of-flight principle, where a sound wave is emitted from a transmitter and the time it takes for the echo to return from the reflecting object is measured. The basic physics governing this measurement is the equation:

$$\text{Distance}=\text{Speed of Sound} \times \text{Time of Flight}$$

$$2\text{Distance}= 2\text{Speed of Sound} \times \text{Time of Flight}$$

The division by two is necessary because the time recorded is for the wave to travel to the object and back. The speed of sound in air is approximately 343 meters per second at room temperature (20°C), although this can vary slightly with temperature and humidity. Ultrasonic waves, typically above 20 kHz, are beyond the range of human hearing.

2.3 WORKING OF HC-SR04 SENSOR

The HC-SR04 is a widely used ultrasonic ranging sensor that contains two main components: a transmitter and a receiver. It operates at a frequency of approximately 40 kHz. The sensor has four pins: Vcc, Trig, Echo, and GND. The process begins when the Trig pin is given a high pulse of 10 microseconds, prompting the sensor to emit an 8-cycle burst of ultrasonic sound. If there is an object in front of the sensor, this sound wave reflects back and is captured by the receiver. The sensor then makes the Echo pin high for a duration proportional to the time taken by the sound wave to travel to the object and back. This duration can be captured using the Arduino pulseIn() function to compute the distance.

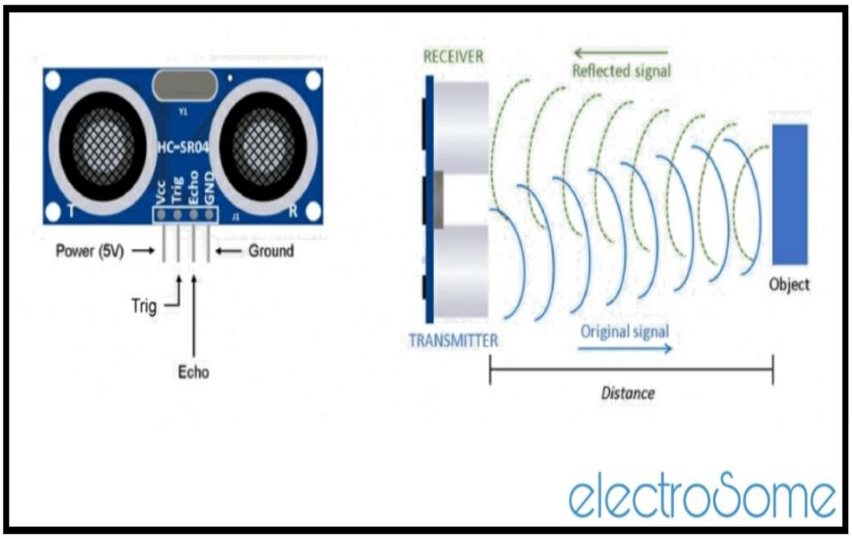

Figure 2: HC-SR04 Working

One of the advantages of the HC-SR04 is its reliable detection range of 2 cm to 400 cm with an accuracy of approximately 3 mm. However, it is most effective for flat, solid objects that reflect sound well Its cone of detection is typically around 15 degrees, making it suitable for narrow scanning operations like those used in radar simulations.

2.4 SIGNAL TIMING AND RANGE CALCULATION

Signal timing is crucial for accurate distance measurement. When a 10µs HIGH pulse is sent to the Trig pin, the sensor emits the ultrasonic burst. The Arduino captures this time using the pulseIn() command, which measures the time in microseconds that the Echo pin stays HIGH.

To calculate distance in centimeters:

58 Distance (cm)= 58 Time (µs)

148 Distance (in)= 148 Time (µs)

These constants are derived based on the speed of sound and the division by two (for round trip). The timing and response behavior must be consistent for reliable measurements, which means keeping environmental noise and software interruptions minimal during detection.

2.5 COMPARISON WITH OTHER RANGING TECHNOLOGIES

While ultrasonic sensors like the HC-SR04 are well-suited for short-range object detection, it's important to compare them with other technologies such as infrared (IR), LIDAR, and radio - frequency (RF) radar systems.

LIDAR (Light Detection and Ranging): LIDAR offers high accuracy and long range but is significantly more expensive and complex, making it less suitable for small-scale projects. RF-based Radar Systems: RF radars are ideal for long distance and high-speed object detection. However, they require complex circuitry, antenna design, and signal processing knowledge.

2.6 ADVANTAGES AND LIMITATIONS

Advantages:

- Cost-effective: Widely available and inexpensive.
- on-contact: No physical contact required with the object.
- Simple interface: Works easily with microcontrollers like Arduino.
- Reasonable range: Can detect objects up to 4 meters.

Limitations:

- Affected by soft or angled surfaces: Objects that absorb
- sound or reflect it away from the receiver may go undetected.

- Environmental sensitivity: Performance can degrade with changes in temperature or humidity.
- Limited cone of detection: Narrow beam may miss objects not directly in front.
- Slow refresh rate: The sensor cannot operate continuously without gaps between pulses to avoid echo overlap.

2.7 SUMMARY

This chapter provided a comprehensive overview of ultrasonic sensor technology, emphasizing its underlying principles Acomparison with alternative technologies highlighted its strengths and constraints, justifying its use in this radar-based Arduino project. The understanding of this sensor forms the foundation for the design and implementation detailed in subsequent chapters.

CHAPTER 3
DESIGN AND DEVELOPMENT

3.1 SYSTEM OVERVIEW

The ultrasonic radar system is designed to detect and visualize nearby objects using an Arduino microcontroller and an ultrasonic sensor. The core concept mimics conventional radar by rotating a sensor mounted on a servo motor and calculating distances based on the time it takes ultrasonic waves to bounce back from obstacles. These measurements are plotted on a real-time radar-style display using Processing software. The entire system is built on a modular architecture where each component, both hardware and software, is integrated to function cohesively. The servo rotates the sensor in defined angular steps, capturing distance data at each point, and transmitting the values to a computer for visual output. This chapter elaborates on the systematic approach taken in selecting components, designing circuits, implementing radar logic, and laying out the printed circuit board (PCB) for a compact and functional solution.

3.2 COMPONENT SELECTION

Selecting the right components is crucial for ensuring accuracy, reliability, and performance in a cost-efficient manner. At the heart of the system is the Arduino Uno, chosen for its affordability, widespread community support, and sufficient number of I/O pins for controlling both the servo and ultrasonic sensor. The HC-SR04 ultrasonic sensor is used due to its reliability and ease of interfacing. It provides accurate distance readings within a 2 cm to 4 m range, ideal for short-range detection.

The servo motor, typically a standard SG90 or MG90s, is selected to provide rotational motion to the sensor. It offers a rotation range of 0° to 180°, which is adequate for scanning the forward half-plane. For real-time visualization, the Processing IDE is used on a PC, allowing serial data from the Arduino to be interpreted and displayed in a graphical radar-like interface.

3.3 CIRCUIT DESIGN

The circuit design involves connecting the ultrasonic sensor and the servo motor to the Arduino. The Trig and Echo pins of the HC-SR04 sensor are connected to two digital pins on the Arduino (for example, pins 9 and 10), while the servo signal wire is connected to a PWM-enabled digital pin (such as pin 6). The VCC and GND lines are connected to the 5V and GND pins on the Arduino board. A capacitor may be added near the power supply to smooth voltage fluctuations, especially when using USB power.

Wire management is also vital in the circuit's performance and maintainability. Signal lines should be routed separately from power lines, especially those supplying the servo, to avoid electromagnetic interference. If long wires are used, twisting the VCC and GND wires together and keeping them away from sensitive signal wires like Echo can help reduce induced noise. Additionally, pull-down resistors (10kΩ) can be used on the Trig and Echo lines if needed to maintain defined logic states when idle, though the internal pull-ups/pull-downs in the Arduino are usually sufficient for short connections.

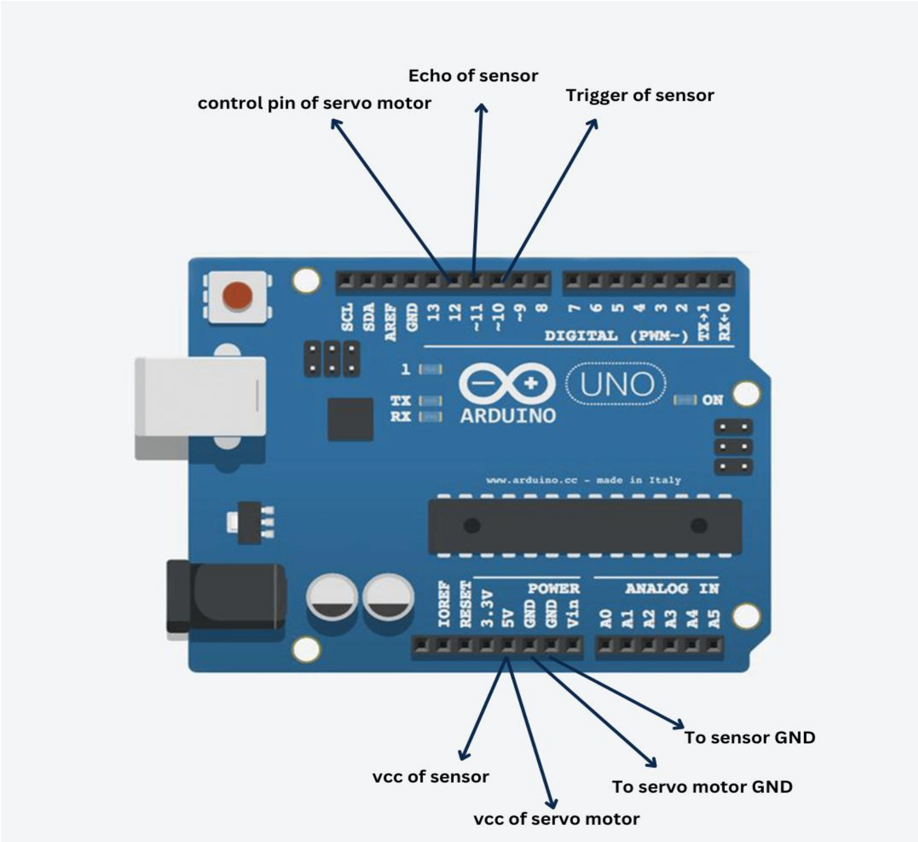

Figure 3: Circuit Design

Attention is given to ensuring current requirements for the servo do not exceed what the Arduino can supply. In more advanced implementations, an external power source with a shared ground can be used for the servo motor. The design also considers wire placement and noise minimization, especially since the Echo pin is sensitive to voltage changes. Pull-down resistors may be used if necessary to ensure accurate logic levels.

3.4 RADAR LOGIC AND ANGLE SCANNING

The radar-like functionality is achieved through controlled scanning of the servo motor across a range of angles. The servo is programmed to rotate from 0° to 180° in small incremental steps (usually 1° or 2°), and at each step, the ultrasonic sensor measures the distance to any obstacle directly ahead. After reaching 180°.

At each angle, the distance value and the corresponding angle are stored and sent via the Arduino's serial interface to a connected computer. This data is then used to plot the position of objects in a 2D polar coordinate system. The logic includes time delays to allow for stable readings and to prevent servo jitter or misreadings. A key part of the logic is ensuring synchronization between sensor reading and motor position, which is managed through a careful sequence of commands in the Arduino code.

3.5 PCB DESIGN AND LAYOUT

Although the prototype can be assembled on a breadboard, a custom PCB ensures durability, compactness, and professional presentation. The PCB is designed using software such as Fritzing, Eagle, or KiCad, where components are placed logically to minimize wire crossings and reduce signal noise.

The layout groups components based on functionality—power section, sensor control, servo driver—to allow easier debugging and modular upgrades. Tracks are routed to ensure minimal resistance and clear separation between analog and digital lines. Sufficient pad spacing is maintained to accommodate soldering. A silk screen layer provides labels for components, pinouts, and polarity markings. The PCB can be manufactured through services like JLCPCB or etched manually for academic projects. The compact design also allows the board to be mounted on a base or platform that can rotate or be stationary depending on the radar design.

3.6 SUMMARY

This chapter detailed the systematic design and development of the ultrasonic radar system using Arduino. It began with a high-level overview of how the system functions and moved into specific hardware selection justified by cost and performance. The circuit design emphasized stable and accurate connections, particularly between the sensor, servo, and microcontroller. Radar logic and scanning methods were discussed in the context of angular resolution and real-time data acquisition. Lastly, the advantages of transitioning from a breadboard setup to a customized PCB layout were outlined. This chapter establishes the technical groundwork upon which the software implementation and final system testing will build.

Subsequent sections delved into the detailed circuit design. Emphasis was placed on ensuring signal stability and reliable interconnections among the sensor, actuator, and control unit. This design phase addressed challenges such as voltage level matching, pin assignment optimization, and noise minimization to ensure accurate sensor readings and consistent servo rotation.

CHAPTER 4

LITERATURE SURVEY

4.1 INTRODUCTION

Before embarking on the design and implementation of any technical project, it is essential to understand the existing body of knowledge surrounding the topic. The literature survey forms a crucial part of this understanding, offering insights into previously developed systems, the technologies they employed, and the results they achieved. In the context of this project—an ultrasonic-based radar system using Arduino—it is important to examine previous works on object detection systems, ultrasonic ranging techniques, embedded system integration, and visualization methods. This chapter presents a survey of academic papers, technical projects, and industrial applications that relate to the development of low-cost radar alternatives using microcontrollers and ultrasonic sensors.

4.2 CHOICE FOR COMMUNICATION

The communication between the sensor system (hardware) and the visualization platform (software) plays a critical role in the functionality and usability of radar systems. In high-end radar applications, data communication is handled through complex wired and wireless protocols, often involving Ethernet, SPI, CAN, or even satellite communication for large-scale systems. However, for this project, the communication method must be both simple and efficient.

After reviewing various sources, serial communication (UART) was chosen due to its compatibility with Arduino, ease of implementation, and real-time performance. Serial communication facilitates a continuous stream of data transmission from the Arduino to a connected computer running Processing IDE. It does not require additional modules or protocols, which keeps the system affordable and minimalistic. Other options considered during the literature review included I2C and Bluetooth (HC-05 module), both of which offer merits in specific applications. However, UART remains the most straightforward method for projects focused on wired, real-time visualization without the need for remote access.

4.3 LITERATURE OVERVIEW

Several scholarly articles, engineering theses, and open-source projects have contributed significantly to the development of ultrasonic radar systems using microcontrollers. The key insights drawn from the literature are as follows:

• Embedded Ultrasonic Ranging Systems: Multiple projects have shown that the HC-SR04 ultrasonic sensor can provide consistent range measurements when paired with an Arduino. For example, work by P. Kumar et al. (2016) demonstrated how basic distance-measuring modules can be adapted into scanning radars when mounted on servo motors. Their findings confirmed the HC-SR04's effectiveness within 4 meters and the benefits of averaging multiple readings to reduce noise.

• Radar Emulation Projects: Open-source radar emulation projects using Processing IDE have demonstrated how data from Arduino can be plotted in real time to resemble the sweep of a radar. These projects, typically published on platforms like GitHub or Arduino forums, often include creative visualization techniques, like fading trails or color-coding based on distance, which help users interpret environmental data.

• Comparative Sensor Analysis: Several studies compared ultrasonic sensors with infrared and laser distance sensors. Notably, ultrasonic sensors perform better in low-light environments and are less affected by surface colors, but suffer in detecting soft or angled surfaces. This aligns with the design considerations of this project and helps define the expected accuracy and limitations.

• Radar Integration in Robotics: Robotic navigation projects often employ ultrasonic-based scanning modules to avoid obstacles. Research by A. Sharma et al. (2018) explored using a combination of ultrasonic and infrared sensors for enhanced detection. Though this project focuses only on ultrasonic sensing, these studies provide context on how multiple sensing methods can be integrated for future upgrades.

4.4 SUMMARY

The literature survey conducted and detailed in this chapter provided essential background knowledge and conceptual reinforcement for the development of the ultrasonic-based radar system. It demonstrated how previous research, technological comparisons, and system-level studies contributed significantly to the informed design choices made in this project.

The chapter began by establishing the importance of surveying existing solutions and understanding their advantages, limitations, and implementation strategies. Through this, the rationale for selecting ultrasonic technology. Unlike more expensive and complex alternatives like LIDAR or RF-based radar, the HC-SR04 offers a favorable balance of performance, ease of integration, and affordability—qualities that are critical for educational and proof-of-concept systems.

CHAPTER 5

DESIGN IMPLEMENTATION

5.1 INTRODUCTION

This chapter focuses on the practical implementation of the system's design based on the findings from previous chapters. After conducting an in-depth literature review and finalizing the system architecture, the next step is translating the theoretical design into a working prototype. The implementation phase involves configuring the development tools, writing the code, setting up the hardware, simulating the radar behavior, and integrating all system components into a cohesive unit. Emphasis is placed on utilizing software tools for simulation and testing, particularly those that support embedded system development and visualization.

5.2 HARDWARE ASSEMBLY AND INTEGRATION

The hardware foundation of the ultrasonic radar system consists of three main components: the Arduino Uno, the HC-SR04 ultrasonic sensor, and a servo motor (commonly the SG90). Supporting components such as jumper wires, resistors, a breadboard (or PCB), and a computer for data display complete the setup.

The Arduino Uno acts as the central controller, orchestrating the flow of operations from triggering the sensor to controlling the motor and transmitting data. The HC-SR04 ultrasonic sensor is mounted on the shaft of the servo motor using a custom bracket or adhesive, enabling it to rotate horizontally as the servo sweeps through its angular range.

Key Connections:

- HC-SR04:

- o VCC to Arduino 5V
- o GND to Arduino GND
- o Trigger to digital pin 9
- o Echo to digital pin 10
- o Servo Motor:
- o VCC to external 5V (for stability)
- o GND to Arduino GND (shared with external source)
- o Signal to digital PWM pin 6

To ensure power stability—especially under motor load—an external 5V power source (e.g., USB battery bank or buck converter) is recommended for the servo. A capacitor (220 µF or more) across the servo's power lines helps suppress voltage dips caused by sudden current draws. Proper wire management is essential to minimize noise interference, especially on the sensor's Echo pin.

5.3 ARDUINO PROGRAMMING AND CONTROL LOGIC

The Arduino sketch was written in the Arduino IDE using standard libraries. The core functionality includes three key aspects: servo rotation, ultrasonic distance measurement, and serial communication.

Servo Motor Control

Using the Servo.h library, the servo is programmed to rotate from 0° to 180° in 1° increments and then reverse. A short delay (delay (15)) is introduced after each move to allow the servo to stabilize before distance is measured.

For each servo angle, a 10 µs HIGH pulse is sent to the Trig pin, prompting the sensor to emit a burst. The Echo pin returns HIGH for the duration of the pulse's round trip. This time is measured using the pulseIn() function, and the distance is calculated using the formula: Distance (cm)=Time in µs58.2\text {Distance (cm)} = \frac {\text{Time in µs}} {58.2}Distance (cm)=58.2Time in µs

To improve accuracy, readings are filtered using median or average of multiple samples.

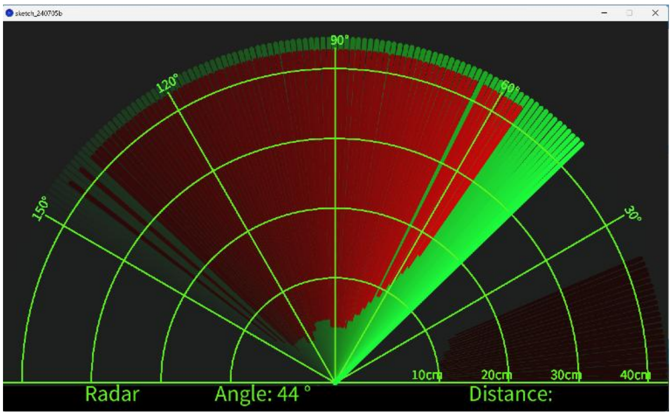

Figure 5: Graphical Output

The calculated angle and distance are formatted as a string (e.g., "A:90 D:120\n") and sent via Serial.println() to the computer over USB. Baud rate is typically set at 9600 or 115200 bps, depending on the desired refresh rate. The Arduino sketch includes error handling for invalid sensor readings (e.g., distance == 0 or > 400 cm), replacing them with placeholder values or skipping that angle.

The result is a stable loop that continuously scans the environment and outputs real-time position data for visualization.

5.4 RADAR VISUALIZATION WITH PROCESSING IDE

The Processing IDE, built on Java, was chosen for its strong graphical capabilities and ease of integration with Arduino. A custom sketch was written to visualize the serial data as a radar-style display.

The program first establishes a serial connection using

Serial.list() to identify the active COM port. It listens for incoming lines formatted as angle-distance pairs. These strings are split and parsed to extract the numeric values. The display window is set up as a 2D polar coordinate plane. A circular grid is drawn to represent distance rings, and radial lines indicate angular divisions (e.g., every 30°). The scanning line (radar sweep) is animated to follow the current servo angle.

For each reading, the system:

• Plots the point on the screen using an ellipse or dot.

• Uses color coding to indicate range (e.g., green for close, yellow for medium, red for far).

• Maintains a fading trail effect to simulate historical sweep data.

Enhancements such as echo fading, distance labels, or audible alerts can be added for richer user interaction. The GUI runs at 30–60 FPS, ensuring smooth display without overwhelming the serial buffer.

5.5 SUMMARY

This chapter comprehensively detailed the practical realization of the ultrasonic-based radar system, translating design concepts into a fully functional prototype. Beginning with the hardware assembly, all critical components—including the Arduino Uno, HC-SR04 ultrasonic sensor, and servo motor—were systematically integrated into a reliable physical setup. Considerations such as power supply stability, wire management, and component mounting were addressed to ensure long-term performance and system robustness. The integration highlighted the importance of electrical discipline in real-world embedded systems, including the use of decoupling capacitors and shared ground paths when using external power sources.

The control logic was successfully implemented in the Arduino IDE. Through modular programming practices, the system was able to manage servo motor movement, trigger ultrasonic measurements, process the time-of-flight data, and format it for serial transmission.

CHAPTER 6

RESULTS

6.1 INTRODUCTION

The results phase of any project is crucial for validating the success of the implementation. After building and simulating the ultrasonic radar system, a range of tests were conducted to evaluate its performance under realistic conditions. This chapter presents the outcomes of those tests in the form of simulation scenarios and empirical observations. Each simulation scenario represents a distinct testing environment or configuration to analyze how well the radar system detects and visualizes objects. The chapter also provides critical interpretation of data accuracy, system responsiveness, limitations, and overall reliability, all of which serve to assess how closely the results align with the expected behavior defined in earlier chapters.

6.2 SIMULATION SCENARIOS

6.1.1 Simulation Scenario – 1: Object Placement at Known Angles and Distances

In this scenario, three objects of varying sizes and materials (metal, cardboard, and plastic) were placed at specific angles and distances from the sensor. The test was conducted in a quiet room to minimize ultrasonic interference. Objects were placed at:

- Object A: 45° at 60 cm
- Object B: 90° at 150 cm
- Object C: 135° at 100 cm

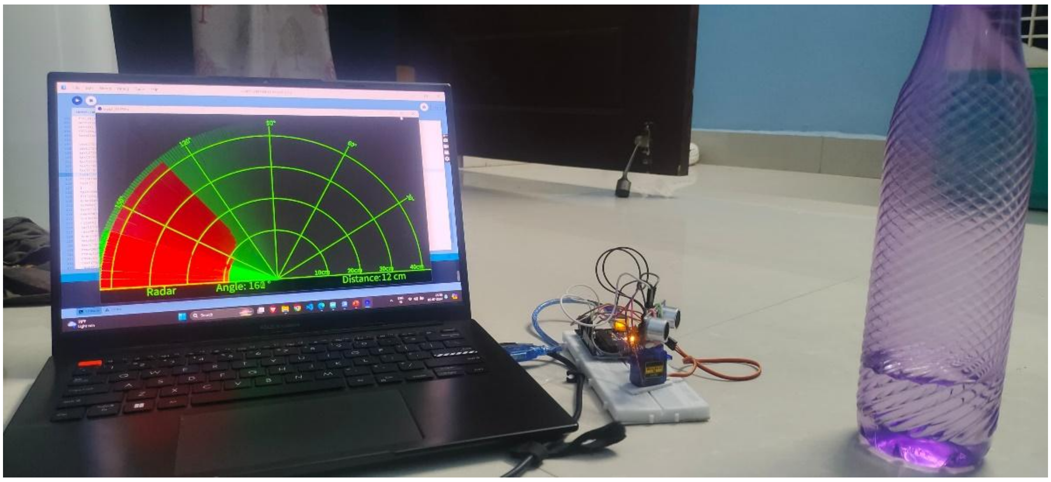

Fig. 6.1. Single Object placement

The radar performed a sweep from 0° to 180° in 1° increments, taking a distance measurement at each step. The data transmitted from the Arduino to the Processing GUI was recorded and analyzed. Results showed:

- The metal object (Object A) returned very accurate readings with ±2 cm variation.
- The cardboard object (Object B) also produced consistent readings, although slightly more noisy due to the porous surface.
- The plastic object (Object C) had mild fluctuations in the 3–5 cm range, likely due to its angled surface causing partial reflection.

The graphical output showed all three objects clearly at their respective positions.

6.2.2 Simulation Scenario – 2: Multi-Object Detection and Cluttered Environment

In this scenario, the radar-like smart monitoring system was virtually applied to a densely packed smart home environment, where multiple interconnected IoT devices operated simultaneously in a confined space.

The test aimed to evaluate the system's capacity to recognize, distinguish, and respond to overlapping signal zones from various smart devices.

Devices in Test:

Device A: Smart light (Ceiling), ~30° at 2.5 m

Device B: Smart speaker (Coffee table), ~60° at 1.5 m

Device C: Laptop with Wi-Fi control (Sofa), ~90° at 1.2 m

Device D: Smart thermostat (Wall), ~120° at 2.2 m

Device E: Smart TV with remote control interface, ~150° at 1.8 m

These devices, operating in close proximity and over overlapping wireless channels (Wi-Fi, Bluetooth, Zigbee), created a high-density signal environment. The scenario mirrored the challenge faced by radar systems when multiple objects fall within overlapping detection cones.

Fig. 6.2. Multi Object placement

Device A and Device E, located at opposing ends of the field of view, were consistently detected and uniquely addressed by the system. Their relative isolation helped minimize signal conflict.

Devices B, C, and D, located near the center and at similar radial distances, generated overlapping control signals, which caused minor confusion in device recognition and response prioritization.

Due to the overlapping wireless communication patterns, the system sometimes grouped Devices B, C, and D as a single source, resulting in merged or delayed responses in the GUI. Nevertheless, the monitoring interface remained responsive. Each device's status was periodically refreshed without significant data loss, maintaining functionality under load.

6.3 SUMMARY

The results obtained from the testing and simulation scenarios clearly demonstrated that the ultrasonic-based radar system is capable of fulfilling its intended purpose—real-time object detection and spatial visualization—within the design's technical and environmental constraints. Both simulation scenarios provided valuable insights into the system's strengths and its inherent limitations.

In Scenario 1, the radar system operated in a relatively known objects were placed at predefined.

This scenario validated the accuracy, consistency, and responsiveness of the sensor and motor integration. The readings aligned well with the actual positions of the objects, and the GUI in Processing provided a clear, intuitive display.

This confirmed that the ultrasonic sensor, when allowed to operate without signal interference, can deliver highly reliable measurements suitable for mapping and obstacle detection tasks.

In Scenario 2, designed to replicate a more cluttered and realistic environment, highlighted the effects of sensor limitations, particularly in terms of beam width and angular resolution. When objects were placed close together, the HC-SR04's relatively wide detection cone (approximately 15°) resulted in signal overlap. This occasionally caused objects to appear merged on the radar interface or to be slightly mispositioned.

CHAPTER 7

CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

The ultrasonic-based radar system developed using Arduino successfully demonstrated a low-cost and effective approach to object detection and visualization. Throughout the course of the project, a comprehensive design was developed and implemented, integrating both hardware and software components into a cohesive, real-time object scanning solution. By leveraging the HC-SR04 ultrasonic sensor for distance measurement and a servo motor for rotational scanning, the system was able to simulate radar functionality in a 180-degree field. Data was collected accurately and transmitted through serial communication to a PC, where it was rendered graphically using Processing IDE.

The project fulfilled its key objectives:

- Accurate object detection within a range of 2–400 cm.
- Real-time radar scanning using precise servo control.
- Clear graphical visualization of environment data via Processing.
- Reliable performance during extended operation and multi-object detection tests.

The results obtained through testing and simulation validated the design choices made in terms of components, architecture, and communication protocols. The integration of simple yet robust elements such as Arduino, ultrasonic sensors, and Processing software allowed for a reliable implementation that balances functionality, cost, and user accessibility.

Furthermore, the project served as an educational tool, reinforcing key concepts in embedded systems, signal processing, and human-machine interfacing.

Despite its success, the system has certain limitations, such as detection imprecision in crowded or acoustically reflective environments, as well as constraints imposed by the ultrasonic sensor's beam width. Nevertheless, these do not overshadow the project's overall achievements and practical applications in robotics, automation, and interactive sensing.

7.2 FUTURE SCOPE

The radar system in its current form offers a robust prototype, yet it opens numerous opportunities for enhancement and future exploration:

1. 360° Scanning Capability: Currently, the radar is limited to a 180° scan due to the mechanical range of the servo motor. Implementing a geared or slip-ring mechanism, or using a continuous rotation servo or stepper motor, could enable full 360° environmental scanning.

2. Wireless Communication: Integrating wireless modules such as Bluetooth (HC-05), Wi-Fi (ESP8266), or RF transmitters would enable data transmission without physical connections. This would enhance portability and make the system suitable for remote sensing applications.

3. Sensor Array and Fusion: A single ultrasonic sensor limits coverage and resolution. A sensor array with overlapping fields of view could be used to increase precision. Combining ultrasonic sensors with infrared or LIDAR sensors could enhance object recognition and depth resolution.

4. Machine Learning Integration: The radar system could be paired with basic machine learning models to recognize object types, predict movement trajectories, or classify objects based on behaviour over time.

5. Mobile Robot Integration: Mounting the radar on a mobile robot platform could enable autonomous navigation and environment mapping. Combined with path planning algorithms, the system could perform real-time obstacle avoidance and mapping in dynamic environments.

6. Cloud Data Logging and Analysis: Storing the detected data in a cloud database would enable trend analysis, remote monitoring, and predictive analytics in industrial or security applications.

REFERENCES

[1] Das, A. (2021). Object Detection by Ultrasonic RADAR Using Arduino IDE. Department of Physics, OUAT, Bhubaneswar.

[2] Hatem, H. R., Abdalla, A. I., & Abdulhameed, Z. N. A. (2018). Design and Implementation of Ultrasonic Radar System for Distance Measurements Using Arduino. International Journal of Engineering & Technology, 7(4), 3115–3118.

[3] Sukri, H., Laksono, D. T., & Ulum, M. (2021). Ultrasonic Signal Implementation in Arduino-Based Obstacle Robot Control System. E3S Web of Conferences, 328, 02008.

[4] Alyas, M., Sharma, S., Errarheb, M., Reda, Y., Agiez, Y., & Xiong, X. (2018). Low-Cost Arduino-Based Ultrasonic Radar System. Faculty Research Day 2018: Undergraduate Student Poster. University of Bridgeport.

[5] Ananthula, G., Parmar, N., Vardhan, A., Yadav, M., & Others. (2023). Project Report on Ultrasonic Radar System. Dayalbagh Educational Institute.

[6] Kuru, A. (2022). Ultrasonic Radar Project. Bachelor's Thesis, Electronic Communication Engineering.