

Smart Radar Security System

Ms. Divya Singh, Verma Anandkumar, Sandeep Kumar Yadav, Rajat Kuma

Assistant Professor, Computer Science Engineering (Internet of Things)

UG Students, Computer Science Engineering (Internet of Things)

Raj Kumar Goel Institute of Technology, Ghazaibad

varmaanandkumarramnath@gmail.com, yadav9717sandeep@gmail.com, rathorerk31@gmail.com,

Abstract: *Traditional radio-frequency (RF) radar systems tend to be expensive and complicated, making them less suitable for simple indoor security or educational robotics projects. To tackle this issue, this project introduces a low-cost spatial mapping system using an ESP32 microcontroller combined with an ultrasonic sensor. Unlike standard ultrasonic sensors that only measure distance in one fixed direction, this system mounts the sensor on a small servo motor that sweeps across 180 degrees, allowing it to scan a wider area. The ESP32 calculates the distance to obstacles by measuring the time it takes for sound waves to bounce back, then uses trigonometry to convert these distances into two-dimensional coordinates. This information is sent via Wi-Fi to a local dashboard, where the environment is mapped in real time. During indoor tests ranging from 10 cm to 4 meters, the system was able to map static objects with impressive accuracy—over 98% within the 0.5 to 3-meter range. Although the sensor's effective range is limited to about 4 meters due to how ultrasonic waves spread, the results clearly show that combining simple ultrasonic hardware with an ESP32 offers a practical, accurate, and affordable alternative to traditional RF radar systems for short-range obstacle detection and smart home applications.*

Keywords: Arduino UNO, Ultrasonic Sensor, Servo Motor, Embedded System, Object Detection, Radar Technology

I. INTRODUCTION

Short-range obstacle detection is a key need for indoor robotics and smart home security systems. Traditional radio-frequency (RF) radar systems do a great job at long-range tracking, but they tend to be expensive and require complicated signal processing. This makes them less suitable for affordable, embedded applications. Ultrasonic sensors like the HC-SR04 are a cheaper option for measuring distance, but they have a big limitation—they can only detect objects directly in front of them because their line of sight is fixed and one-dimensional.

This project solves that problem by creating a low-cost, active sweeping radar system. By attaching an HC-SR04 ultrasonic sensor to a 180-degree SG90 servo motor, the system can scan across a wide area instead of just one direct it detect the frequency at 180 degree in front of the sensor and there is an alarm sensor or buzzer to make an alert for object detected near its sensor and futher implementation is that it has telegram notification feature which alert the owner when he is outside the home . .

II. LITERATURE REVIEW

Considerable research has been conducted on the application of ultrasonic sensors for localized spatial mapping, primarily utilizing standard Arduino microcontrollers. Patel et al. (2023) developed a foundational low-cost radar prototype using an Arduino Uno and an HC-SR04 sensor. While their system successfully demonstrated high accuracy for distances under 4 meters, its physical architecture restricted scanning to a fixed plane and lacked remote monitoring capabilities. To address visualization, Rahman et al. (2022) and Mehta and Reddy (2021) expanded on this architecture by interfacing the microcontroller with the Processing IDE via serial communication. This allowed for real-time polar



coordinate mapping on a computer screen. However, these tethered systems required a constant physical connection to a PC, limiting their viability for standalone security applications.

Recent advancements have focused on integrating wireless and Internet of Things (IoT) capabilities to remove these physical limitations. Sharma et al. (2024) implemented a smart radar system combining ultrasonic sensors with external IoT modules to transmit data to a cloud server. While this enabled remote monitoring, the addition of separate network modules increased both the system's cost and coding complexity. Similarly, Thomas et al. (2023) demonstrated an energy-efficient radar approach utilizing the onboard Wi-Fi of an ESP32 microcontroller.

While these prior works successfully prove the concept of ultrasonic radar, the majority focus strictly on basic environmental mapping or robotic obstacle avoidance. There remains a distinct gap for a dedicated, low-cost Smart Radar Security System that leverages the integrated dual-core processing and Wi-Fi of the ESP32 to provide both accurate spatial mapping and dedicated security alerts without the need for heavy, tethered external hardware..

Table 1: Ultrasonic and Arduino Based Radar System

Paper / Project	System Architecture	Cost	Ease of Implementation	Performance	User Experience	Limitations
Patel et al. (2023)	Arduino Uno + HC-SR04	Low	Easy	High accuracy (under 4m)	Clear radar visualization	Fixed scan, no wireless capability
S. Sharma et al. (2024)	Arduino + External IoT	Moderate	Moderate	High	Cloud data transmission	Complex coding and network dependency
M. Rahman et al. (2022)	ESP32 + HC-SR04	Low	Easy	Medium	Energy-efficient Wi-Fi	Strictly mapping, no security focus
R. Gupta et al. (2021)	ESP32 + Servo + HC-SR04	Moderate	Moderate	Short range	180° Sweep + Local Wi-Fi UI	Range physically limited to 4m

III. PROPOSED SYSTEM ARCHITECTURE

The core objective of this project was to keep the hardware simple, low-cost, and easy to troubleshoot during physical testing. Therefore, the system architecture relies on three primary hardware components: the ESP32 development board, an HC-SR04 ultrasonic sensor, and an SG90 micro servo motor.

A. Hardware Selection

- **ESP32 Microcontroller:** We selected the ESP32 instead of a standard Arduino Uno primarily because of its built-in Wi-Fi capabilities. This allowed the system to transmit radar data to a local screen without requiring heavy external network modules, which keeps the circuit board clean and reduces overall power draw. Its dual-core processor also easily handles the motor movement and Wi-Fi transmission at the same time without lagging.
- **HC-SR04 Ultrasonic Sensor:** This serves as the main distance detector of the radar. It operates by sending out a 40kHz sound wave and measuring the time it takes for the echo to return. Its four pins (VCC, GND, Trigger, and Echo) are wired directly into the digital pins of the ESP32.
- **SG90 Servo Motor:** Because the ultrasonic sensor can only read distances in a fixed, straight line, we mounted it on top of the SG90 servo motor. The servo acts as a mechanical neck, sweeping the sensor back and forth across a 180-degree area to create a wide scanning zone.

B. Data Flow and Hardware Integration

- The architecture operates in a continuous, synchronized loop. First, the ESP32 commands the servo motor to move by a single degree. At that exact angle, the HC-SR04 is triggered to fire a sound wave and record the distance of any object in front of it. The ESP32 collects this distance measurement, pairs it with the current angle of the servo motor,



and packages the data. Using its onboard Wi-Fi, the microcontroller then broadcasts this data pair (Distance and Angle) over the local network to a connected dashboard, which plots the physical obstacle on a 2D screen.

C. Software Logic and Dashboard

- To turn the raw distance measurements into a usable radar map, the ESP32 code applies basic trigonometry. Using the measured distance (D) and the servo's current angle (θ), it calculates the exact 2D position of the object with $X = D \times \cos(\theta)$ and $Y = D \times \sin(\theta)$.

- These coordinates are then sent over Wi-Fi to a local dashboard. The interface visually mimics the servo's sweeping motion and plots a red marker at those exact (X, Y) coordinates whenever an obstacle is detected. This provides a clean, real-time visual of any movement in the room right on a remote PROPOSED METHODOLOGY

The goal of this project is to design and implement a functional ultrasonic Arduino-based radar system capable of detecting and displaying the distance and position of surrounding objects. The methodology involves several systematic steps, including hardware design, sensor integration, signal processing, data acquisition, visualization, and testing. The system uses the HC-SR04 ultrasonic sensor to emit ultrasonic waves and measure the time interval between transmission and reception of the echo to calculate the distance to an object.

A. Initialization and Scanning Cycle

When the system is powered on, the ESP32 first establishes a connection to the local Wi-Fi network and calibrates the SG90 servo motor by resetting it to the 0-degree starting position. Once initialized, the main program loop begins.

The servo motor is programmed to rotate incrementally from 0° to 180° and then reverse back from 180° to 0° . At every single degree of movement, the ESP32 sends a 10-microsecond HIGH pulse to the HC-SR04 Trigger pin. This causes the sensor to emit a 40kHz ultrasonic burst. The microcontroller then measures the time it takes for the Echo pin to go HIGH, which represents the time-of-flight of the sound wave. The raw distance is calculated using the standard formula $D = \frac{V \times T}{2}$, where V is the speed of sound in air (approximately $0.034 \text{ cm}/\mu\text{s}$) and T is the measured echo time.

B. Signal Filtering and Error Handling

During early testing, we noticed that the HC-SR04 would occasionally output random "garbage" values (e.g., reading 1500 cm when no object was present). This happens when the ultrasonic sound wave bounces off angled walls or gets lost in the room. To stabilize the radar dashboard, a simple software filter was implemented inside the ESP32 code. Since the reliable physical limit of the sensor is about 400 cm, any distance measurement calculated above 400 cm is automatically discarded by the microcontroller and is not sent over Wi-Fi. This simple logic gate keeps the radar screen clean and prevents false security alerts.

C. Experimental Testing Procedure

To evaluate the real-world accuracy of the system, a controlled physical test was conducted indoors. A standard measuring tape was laid out on the floor directly in front of the radar. We placed static obstacles (such as cardboard boxes) at fixed physical distances—starting from 10 cm out to 400 cm. For each physical placement, we recorded the exact distance displayed on the radar dashboard. By comparing the known physical distance against the system's calculated distance, we were able to determine the overall measurement accuracy and identify the operational limits of the hardware.

II. RESULT AND DISCUSSION

A. Experimental Results

To verify the system's reliability, we conducted multiple indoor tests. A flat, solid object (a cardboard box) was placed at various known physical distances from the sensor array using a standard measuring tape. The ESP32's calculated



distances were recorded directly from the Wi-Fi dashboard and compared against the physical measurements to calculate the error margin.

Table II: Distance Measurement Accuracy

Actual Distance (cm)	Measured Distance (cm)	Error (cm)	Accuracy (%)
10	10.2	+ 0.2	98.00%
50	50.5	+ 0.5	99.00%
100	99.1	- 0.9	99.10%
200	202.4	+ 2.4	98.80%
300	295.5	- 4.5	98.50%
400	388.0	- 12.0	97.00%

B. Data Analysis and System Performance

As shown in Table II, the radar system performs exceptionally well within the core operational zone of 0.5 to 3 meters, consistently maintaining an accuracy rate of over 98%. Within this range, the red marker on the dashboard accurately reflected the physical placement of the object in the room.

However, we observed that the accuracy drops off as the target approaches the 4-meter mark. At 400 cm, the sensor recorded a measurement of 388 cm (a 12 cm error). This physical limitation occurs because the 40kHz ultrasonic wave spreads out as it travels. By the time the echo bounces back from 4 meters away, the sound wave is weak, causing the sensor to occasionally miscalculate the time-of-flight.

Additionally, during prolonged testing, the SG90 servo motor introduced a very slight mechanical jitter at the extreme scanning angles (near 0° and 180°), causing minor graphical stuttering on the dashboard. Despite these minor hardware limits, the results prove the system is highly viable. For a low-cost indoor security system, detecting a potential intruder at 3 meters with a margin of error of less than 5 cm is more

III. CONCLUSION

This paper presented the design, implementation, and testing of a low-cost, IoT-integrated Smart Radar Security System. The primary objective of this project was to overcome the high costs and heavy processing requirements associated with traditional radio-frequency (RF) radar systems, while simultaneously solving the fixed-direction physical limitation of standard ultrasonic sensors. By utilizing an ESP32 dual-core microcontroller paired with an HC-SR04 sensor and an SG90 servo motor, the project successfully delivered an accessible, highly functional spatial mapping alternative.

The mechanical integration of the servo motor allowed the ultrasonic sensor to perform a continuous 180-degree sweep, effectively converting a simple 1D distance-measuring tool into a dynamic 2D area scanner. Throughout the operation, the ESP32 proved highly capable of handling the simultaneous tasks of mechanical motor actuation, time-of-flight sensor polling, and trigonometric data conversion without system lag. Furthermore, leveraging the microcontroller's onboard Wi-Fi enabled the real-time transmission of these coordinates to a remote dashboard, successfully fulfilling the core requirement for a wireless, localized security monitor.

Physical experimental testing validated the system's reliability for indoor environments. The hardware consistently mapped static obstacles with a measurement accuracy exceeding 98% within a practical 0.5 to 3-meter radius. While the testing phase honestly identified a maximum effective range of approximately 400 cm—primarily due to natural ultrasonic wave dispersion and acoustic absorption by soft materials—the system met all of its intended design goals. Ultimately, this ESP32-based prototype demonstrates that simple, budget-friendly ultrasonic hardware can be practically engineered to provide highly accurate short-range obstacle detection, offering a reliable and scalable foundation for educational robotics and smart home security applications.



REFERENCES

- [1] A. Patel, et al., "Low-cost radar prototype using Arduino Uno and ultrasonic sensors for distance detection," International Journal of Advanced Research in Science, vol. 5, 2023.
- [2] M. Rahman, et al., "Portable radar model displaying results on Processing IDE," Journal of Embedded Systems and Robotics, 2022.
- [3] S. Sharma, et al., "Smart radar system combining ultrasonic sensors with IoT integration," IoT Security and Automation Journal, 2024.
- [4] L. Thomas, et al., "Energy-efficient radar system utilizing an ESP32 microcontroller and Wi-Fi connectivity," Wireless Technology Review, 2023.
- [5] P. Mehta and S. Reddy, "Radar visualization interface using Processing IDE for polar coordinate mapping," Engineering Education Reports, 2021.
- [6] Espressif Systems, "ESP32 Series Datasheet," 2023. [Online].
- [7] "HC-SR04 Ultrasonic Sensor Datasheet," Component Specifications.
- [8] "SG90 Micro Servo Motor Specifications and Timing."

