© December 2025 | IJIRT | Volume 12 Issue 7 | ISSN: 2349-6002

# The "Radar System Using Arduino Uno"

Mr. SyedAbdusSabur[1], Ms.Sneha[2], Ms.Sukanya[3], Ms.Soumya[4], Prof.Nikita[5]
*[1,2,3,4]Student AIML*
*[5]Professor AIML*

***Abstract*—This project presents the design and implementation of a low-cost radar system using an Arduino Uno, HC-SR04 ultrasonic sensor, and SG90 servo motor. The system simulates the basic functioning of a traditional radar by using ultrasonic waves to detect the presence and distance of objects. As the servo motor rotates the sensor across a 0°–180° arc, distance measurements are captured in real time and visualized on a radar-like display using the Processing IDE. The project demonstrates key engineering concepts such as sensor interfacing, signal processing, motor control, and real-time visualization. The system achieves reliable object detection with reasonable accuracy, making it suitable for educational, prototyping, and hobby-level applications. Overall, this project highlights how simple hardware and open-source tools can be integrated to create an effective and interactive radar simulation.**

## I. INTRODUCTION

A radar (Radio Detection and Ranging) system is an electronic technology used to detect the presence, distance, and position of objects within a given range. Traditional radar systems work using radio waves, but for academic and prototype-level projects, ultrasonic sensors provide a simple and effective alternative. In this project, an Arduino Uno microcontroller is interfaced with an HC-SR04 ultrasonic sensor and an SG90 servo motor to create a functional radar model.The ultrasonic sensor emits sound pulses, detects their echo, and calculates distance based on the time taken for the sound to return. By mounting the sensor on a servo motor that rotates from 0° to 180°, the system scans the environment and collects real-time distance data. This data is sent to a computer and visualized using the Processing IDE, generating a radar-like display that shows the detected objects and their approximate positions.This project helps learners understand important engineering concepts such as sensor interfacing, servo motor control, serial communication, and real-time data visualization. It also provides a hands-on demonstration of how modern detection systems work in robotics, security, and automation.

## II. MATERIALS AND METHODS

Materials Required:

1. Arduino Uno – Serves as the main microcontroller that controls the sensor and servo motor.
2. HC-SR04 Ultrasonic Sensor – Used to transmit ultrasonic waves and measure the echo time to calculate distance.
3. SG90 Servo Motor – Rotates the ultrasonic sensor from 0° to 180° for scanning.
4. Breadboard – Used for making connections easily.
5. Jumper Wires – For connecting components like the servo, sensor, and Arduino.
6. USB Cable – For uploading code to Arduino and powering the system.
7. Buzzer/LED (optional) – Alerts when an object is detected at a close distance.
8. Software: Arduino IDE and Processing IDE.

Methods

1. System Setup:

The HC-SR04 ultrasonic sensor is mounted on the SG90 servo motor, enabling it to rotate and scan the area
from 0° to 180°. The Arduino Uno controls both the sensor and the servo motor.

2. Distance Measurement:

The ultrasonic sensor sends out a pulse, and the echo pin receives the reflected wave. The time taken for the echo to return is used to calculate distance using:
Distance = (Speed of Sound × Time) / 2

3. Servo Scanning:

The Arduino controls the servo motor using PWM signals. The servo rotates step-by-step across the scan

angle.
At each angle, the sensor measures distance and sends the data to the Arduino.

4. Data Processing:
The measured distance and angle information is sent to the computer through serial communication. Processing IDE reads this data and plots the detected objects on a radar interface.

5. Visualization:
A radar-like graphical interface is created in Processing. This interface displays objects as points or arcs
based on their distance and angle, allowing users to observe radar scanning behavior visually.

6. System Testing:
The setup is tested by placing objects at different angles and distances. The accuracy of the sensor, the rotation smoothness of the servo, and the visualization output are verified during testing.

## III. RESULTS AND DISCUSSION

Results:
The radar system built using Arduino Uno, HC-SR04 ultrasonic sensor, and SG90 servo motor successfully detected objects within the sensor's measurable range. The servo motor provided smooth rotation from 0° to 180°, allowing continuous scanning of the environment. The distance data captured by the ultrasonic sensor
was transmitted to the Processing IDE and accurately visualized on a radar-like display.

Key Observations:
- Objects were detected reliably within the range of 2 cm to 350 cm.
- The radar display updated in real time, showing the angle and distance of detected objects.
- The system maintained a detection accuracy with minimal errors under controlled conditions.
- The buzzer/LED alert system functioned correctly when obstacles were detected within a set threshold.

Discussion:
The project demonstrated that a simple ultrasonic-based radar system can effectively simulate the basic operation of traditional radar. The combination of Arduino, ultrasonic sensor, and servo motor provided reliable
object detection and smooth environmental scanning.

The Processing IDE visualization enhanced the understanding of radar operation by displaying live scanning
data in a graphical form. However, certain limitations were observed. Ultrasonic sensors tend to give inaccurate
readings for soft, curved, or absorbent surfaces because such materials do not reflect sound properly. Environmental noise and temperature also affect sensor accuracy.

Despite these limitations, the system performed well for educational and prototype applications. It also offers
a strong foundation for future upgrades such as integrating multiple sensors, increasing scanning speed, or
adding AI-based object classification.

## IV. HELPFUL HINTS:

1. Ensure Proper Power Supply:
Provide stable 5V power to the Arduino. Servo motors may draw extra current, so avoid connecting high- load components directly to the Arduino's 5V pin.

2. Secure Sensor Alignment:
Ensure the HC-SR04 sensor is mounted straight and parallel to the ground. Misalignment can cause inaccurate distance measurements.

3. Test the Ultrasonic Sensor Separately:
Before building the full radar system, test the sensor with a simple distance-measurement code to confirm proper working.

4. Avoid Environmental Noise:
Ultrasonic sensors perform best indoors. Loud noises, wind, or soft materials may cause inaccurate readings.

5. Smooth Servo Movement:
If the servo rotates too fast, increase delay between angle steps to ensure stable readings and smooth radar

visualization.

6. Use Hard Surfaces for Testing:
Hard objects like walls or metal surfaces reflect ultrasonic waves better than soft materials such as cloth or foam.

7. Double-Check Wiring:
Ensure correct connections for Trigger, Echo, VCC, GND, and servo signal wires. Loose connections are the most common cause of random sensor data.

8. Calibrate Angle Range:
Some servo motors may not rotate perfectly from 0° to 180°. Adjust angle limits slightly (e.g., 5° to 175°) to prevent jittering or motor strain.

9. Check Serial Communication:
Ensure the correct COM port and baud rate are selected in both Arduino IDE and Processing IDE for smooth data transfer.

10. Save Processing Sketch Frequently:
Processing may crash if serial port access fails. Save your radar visualization sketch often to avoid losing progress.

## V. CONCLUSION

The Arduino-based radar system successfully demonstrates the principles of object detection, distance measurement, and real-time visualization using simple and cost-effective components. By integrating the HC-SR04 ultrasonic sensor, SG90 servo motor, and Arduino Uno, the system effectively scans the environment and displays detected objects on a radar-like interface through the Processing IDE. The project provides a practical understanding of sensor interfacing, servo motor control, serial communication, and graphical data representation. Despite certain limitations, such as reduced accuracy with soft curved surfaces and slower scanning due to servo speed, the system performs reliably for educational prototype and hobby-level applications. Overall, this project serves as an excellent learning platform that bridges theoretical concepts with hands-on implementation. Future improvements—such as using multiple sensors faster scanning mechanisms, or AI-based object analysis—can further enhance the system's accuracy and functionality.

## REFERENCES

[1] Banzi, M., & Shiloh, M. (2014). Getting Started with Arduino. Maker Media, Inc.
[2] Monk, S. (2016). Programming Arduino: Getting Started with Sketches. McGraw-Hill Education.
[3] HC-SR04 Ultrasonic Sensor Datasheet. Elecfreaks, 2013.
[4] SG90 Micro Servo Motor Specifications. Tower Pro.
[5] Arduino Documentation. "Arduino Uno Board Overview." Retrieved from https://www.arduino.cc
[6] Processing Foundation. "Processing IDE Reference." Retrieved from https://processing.org
[7] Kumar, A., & Saini, R. (2020). "Object Detection Using Ultrasonic Sensors for Embedded Applications." International Journal of Engineering Research, 9(3), 112–118.
[8] Patel, V., & Shah, P. (2019). "Design and Implementation of Ultrasonic Radar System Using Arduino." International Journal of Innovative Science and Research Technology, 4(5), 560–564.
[9] Microcontroller Tutorials. "How to Interface HC-SR04 with Arduino." Retrieved from https://microcontrollerslab.com
[10] Singh, M., & Kaur, J. (2021). "Low-Cost Radar Simulation Using Ultrasonic Sensors. "International Journal of Advanced Research in Electronics and Communication Engineering, 10(4), 45–50.