World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Design of Arduino-controlled ultrasonic radar for dynamic distance measurement and hazard detection

Gbadouo Stephane Kevin Gnako *

Mechanical Engineering, Texas Tech University, Texas United States.

World Journal of Advanced Research and Reviews, 2026, 30(02), 1874-1885

Publication history: Received on 15 April 2026; revised on 21 May 2026; accepted on 23 May 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.30.2.1465

Abstract

As more industries look for intelligent sensing and real-time obstacle detection, there is a growing need for affordable radar-based monitoring. In this study, we designed and built an Arduino-controlled ultrasonic radar system that measures distance and detects hazards. The system uses an HC-SR04 ultrasonic sensor, an Arduino UNO microcontroller, an SG90 servo motor, a buzzer module, and the Processing IDE to detect objects and scan the area in real time.

The ultrasonic sensor works by sending out sound waves and measuring how long it takes for the echoes to return from nearby objects. The Arduino UNO moves the servo motor between 0° and 180°, allowing the system to scan the environment like a radar. It sends real-time distance data to a graphical interface for visualization.

The developed design system measures the distance to obstacles at different angles and distances. The radar detected objects in the hazard zone and gave both visual and audio alerts when something got too close. The results showed that the system measured distances accurately, detected obstacles effectively, and displayed radar data clearly. Overall, the radar system provided reliable real-time hazard monitoring and quick obstacle detection.

This Arduino-controlled ultrasonic radar system is a simple, affordable, and efficient solution for intelligent sensing. Real-time visualization tools improve monitoring and can be used in robotics, autonomous navigation, industrial safety, smart surveillance, and IoT sensing.

Keywords: Arduino; Ultrasonic; Radar System; Detection

1. Introduction

People now spend more time driving each day than in the past. With more cars on busy roads and city streets, it can be hard for drivers to quickly spot both moving and stationary obstacles, such as trolleys, buggies, pedestrians, bicycles, and cyclists[1]. There is need for a reliable obstacle detection system that operate in real time to detect potential collision risks and alert drivers so they can react in time[2].

Control of vehicles has shifted from manual operation to software and computer decision-making, allowing cars to drive themselves[3]. As roads become more complex, vehicles need additional detection and analysis tools to drive safely and smoothly on their own[4]. Even the most advanced cars today, which often have more sensors and software than a fighter jet, still cannot guarantee complete safety. This makes it essential to have accurate obstacle-detection systems that can quickly spot both still and moving objects that could cause accidents.

---

* Corresponding author: Gbadouo Stephane Kevin Gnako

Active methods use sensors with a laser transmitter and receiver, such as RADAR, LIDAR, or ultrasound[5]. The system or the driver can control the frequency and direction of these sensors. On the other hand, passive methods use devices like cameras to measure the scene. In passive systems, detection relies solely on what the camera captures, and there is no control over which object or area is scanned[6].

Good obstacle-detection systems need to process large amounts of information quickly and in real time [7]. In an active system, the vehicle can make smart decisions about when and how to capture images and gather data. The main jobs of these systems are to detect roads and obstacles. This usually involves collecting sensor data, assessing the level of risk, and providing feedback to the driver if action is needed.

Today, automation and smart sensing are important across fields such as robotics, defense, transportation, and security[8]. These systems rely on object detection and distance measurement to understand their surroundings and respond smartly. The Ultrasonic Arduino-Based Radar System is designed to detect and display the positions and distances of objects, providing real-time visual feedback similar to traditional radar systems[9].

Traditional radar systems use radio waves and are often expensive and complicated. This project offers a simpler, low-cost solution by using ultrasonic technology with the Arduino platform. The system uses the HC-SR04 ultrasonic sensor, which sends out high-frequency sound waves. When these waves hit an object, they bounce back, and the time it takes for the echo to return helps measure the distance to the object. The Arduino UNO gathers this data and moves a servo motor to rotate the sensor over a set angle, copying the sweeping motion of a radar[10].

The system sends this spatial data to a visualization tool, such as the Processing IDE or the Blynk app, which displays detected objects on a radar-style display that updates in real time. This gives users a clear, ongoing view of the area around them, showing distances and the locations of objects. Building this system involves assembling the hardware, applying signal-processing methods, writing code for control and communication, and generating real-time visual feedback. Programming is done with the Arduino IDE, and sensor data is sent to the visualization platform using serial communication.

2. Literature Review

2.1. Principles of ultrasonic sensors and radar systems

Ultrasonic sensors measure distance and detect objects without touching them. They work by sending out ultrasonic pulses and listening for the echo that bounces back. The range in air depends on the sensor and the object, but it usually covers a few centimeters to several meters. The sensor emits these pulses, and when an object is in its path, the pulses are reflected back to the sensor.

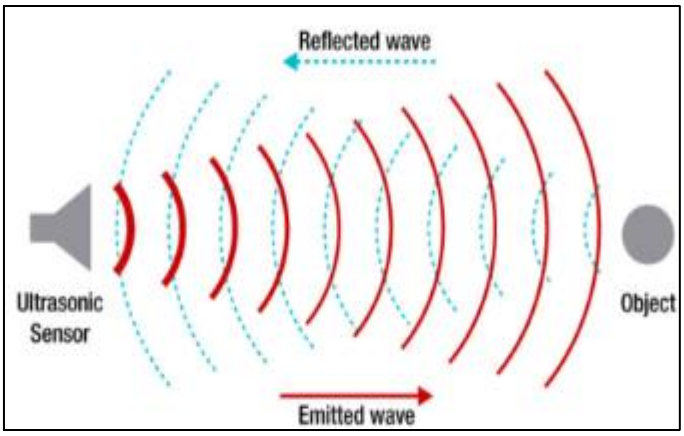


Figure 1 Ultrasonic Time-of-Flight Measurement

An ultrasonic sensor uses a piezoelectric transducer to change electrical signals into mechanical vibrations and vice versa. In a monostatic setup, the sensor acts as both a speaker and a microphone at one frequency.

The sensor measures the time between sending a sound pulse and receiving its echo. Since the speed of sound is known, this round-trip time helps calculate the distance to an object. Equation 1 shows how to calculate this distance.

$$d_{oneway} = \frac{t_{RoundTrip} \times V_{sound}}{2} \tag{1}$$

This ultrasonic sensing method measures how long it takes sound to travel from one point to another. The speed of sound in air changes with temperature. For example, in dry air at 20°C (68°F), sound travels at 343 meters per second, so it covers a kilometer in 2.91 seconds.

2.2. Applications of ultrasonic radar in robotics, autonomous vehicles, and industrial automation

Radar technologies for UGVs are used across many areas, including self-driving vehicles, environmental exploration, and disaster response[11]. UGVs serve different purposes, including military surveillance, transportation, and indoor operations, each with its own requirements[12]. This paper primarily examines radar technologies for navigation, obstacle detection, and environmental mapping in civilian and unstructured settings. The discussion focuses on how radar helps UGVs sense and navigate in general civilian and industrial situations.

Radar systems in UGVs use advanced methods to address challenges such as environmental clutter and limited range in harsh conditions. Some important innovations are listed below.

- FMCW (Frequency-Modulated Continuous-Wave) radar is popular because it can detect objects with high resolution, around 2 cm. It lets UGVs measure both velocity and range simultaneously, up to 250 meters. This makes it well-suited for real-time collision avoidance in self-driving vehicles and off-road navigation.
- SAR(Synthetic Aperture Radar) creates detailed environmental maps with a resolution of below 10 cm. It works especially well in agriculture and forestry, where accurate terrain mapping is important. SAR also performs reliably in low-visibility conditions, which helps in disaster response situations.
- Radar systems are often combined with LiDAR and cameras to overcome the limits of each sensor. Sensor fusion algorithms help detect small objects nearby and keep accurate detection at long distances, over 100 meters. This makes radar especially useful for autonomous navigation.
- Different radar wavelengths are used for different needs. For example, mmWave at 77 GHz is good for short-range, high-resolution detection up to 50 meters, while Ku-band (12–18 GHz) works for long-range sensing up to 20 kilometers. Shorter wavelengths are best for detecting obstacles in cities, and longer wavelengths work well in cluttered rural areas.

Here are some real-world examples that show how radar can be useful in different situations:

- In self-driving vehicles, radar systems play a key role in spotting other cars, pedestrians, and obstacles in cities. Because they work well in bad weather and low light, radars help prevent collisions and support safe autonomous driving.
- During disaster response, radar systems help unmanned ground vehicles find victims through smoke, debris, or snow[13]. Unlike optical and LiDAR sensors, radar can see through these obstacles, making it especially useful in search-and-rescue missions.
- In agriculture and forestry, unmanned ground vehicles equipped with radar can navigate rough terrain and avoid obstacles such as fallen trees, rocks, and equipment[11]. In forests, radar also helps map dense vegetation, improving efficiency and safety.
- For indoor navigation, warehouse, or industrial unmanned ground vehicles use radar to sense their surroundings when GPS (Global Positioning System) is unavailable. Radar can spot shelves, equipment, and moving people, and its short-range, high-resolution features work well in these environments[12].

Radar stands out among sensing technologies and is essential for UGV operations due to several key advantages.

- Radar sensors work reliably in fog, rain, and snow, unlike optical and LiDAR sensors. This means UGVs(Unmanned Ground Vehicles) can keep operating even in tough weather.
- Radar is very good at spotting objects from far away, which is important for fast-moving autonomous vehicles and for long-range surveillance.
- Advanced algorithms help radar filter out unwanted signals from the environment, enabling it to work well even in busy or changing environments[14].

The literature suggested several potential innovations in the application of radar technologies to UGVs such as point-cloud improvements, object detection, navigation, mapping, slam, material classification, and obstacle detection. Point-cloud data derived from radar systems play a critical role in enhancing spatial accuracy for UGVs. Accuracy requirements vary, depending on the application.

- Object detection: Precision levels of 1–2 cm is typically required for detecting small objects, such as debris or tools, ensuring reliable obstacle identification and safe navigation in cluttered environments.
- Mapping and SLAM (Simultaneous Localization and Mapping): Autonomous navigation requires accurate spatial data to maintain localization and path planning, particularly in GPS-denied environments. For instance, centimeter-level accuracy is essential for high-resolution mapping and SLAM (Simultaneous Localization and Mapping) performance.

2.3. Hardware Components

2.3.1. MB-102 Solderless Bread Board

The breadboard features two split power buses, each with 10 columns and 63 rows, giving a total of 830 tie points. All pins are spaced at the standard 0.1 inch, as shown in figure 2. The board has a self-adhesive backing and interlocking parts, making it easy to connect multiple boards. The term "breadboard" now commonly refers to both plugboards and terminal array boards. Breadboards are useful for building temporary prototypes and testing circuit designs. It can be used to prototype many types of electronic systems, from small analog and digital circuits to full central processing units (CPUs).

Table 1 MB-102 Solderless Bread Board Component

No	Contents	Characteristics
1.	Terminal Strips	Tip-point 630
2.	Distribution Strips	Tie-point 200
3.	Solderless breadboard	MB-102
4.	Wire size	20-29 AWG wires
5.	Size	16.5 x 5.5 x 0.85 cm

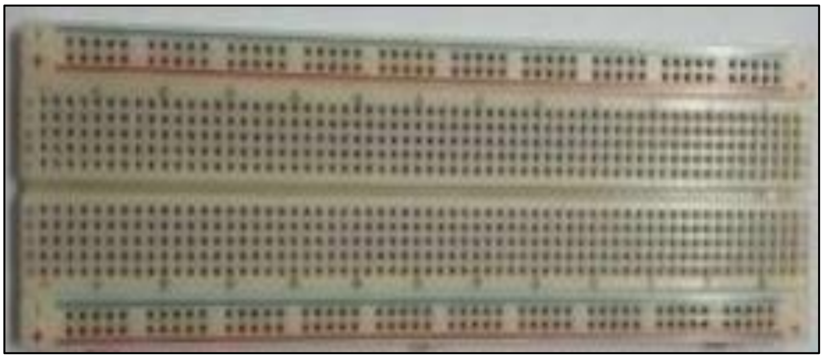


Figure 2 Breadboard, 830 Point Solderless PCB Breadboard MB102

2.3.2. The Jumper

A jumper is an electrical wire, or sometimes a group of wires in a cable, with a pin at each end. It is usually used to connect parts on a breadboard or other equipment without soldering. To use a jumper wire, you insert its end connectors into the slots on a breadboard, a circuit board header, or a test component.

Figure 3 Jumper wires with different standards

2.3.3. Tower PRO TM Microservo SG90

A servomotor is a rotary actuator that allows for precise control of velocity, acceleration, and angular position. It consists of a suitable motor coupled to a sensor for position feedback. It also contains a relatively sophisticated controller for a dedicated module designed specifically for use with servomotors. Servomotors use servomechanism to achieve closed loop control with a generic open loop motor and are used in applications such as robotics, CNC machinery or automated manufacturing as shown in Figure 4. SG90 digital servo used to achieve this study, which is the new version of SG90 analog servo and its properties as in the following table 2.

Figure 4 TowerPro SG90 Servo

Table 2 TowerPro SG90 Servo Characteristics

Contents	**Characteristics**
Modulation:	Analog
Torque:	4.8V: 25.00 oz-in (1.80 kg-cm)
Speed:	4.8V: 0.12 sec/60°
Weight:	9 g
Dimensions:	Length:0.91 in (23.0 mm) Width:0.48 in (12.2 mm) Height:1.14 in (29.0 mm)
Motor Type:	3-pole
Gear Type:	Plastic

Rotation/Support:	Bushing
Pulse Width:	500-2400 µs
Connector Type:	JR

2.3.4. The Transceivers

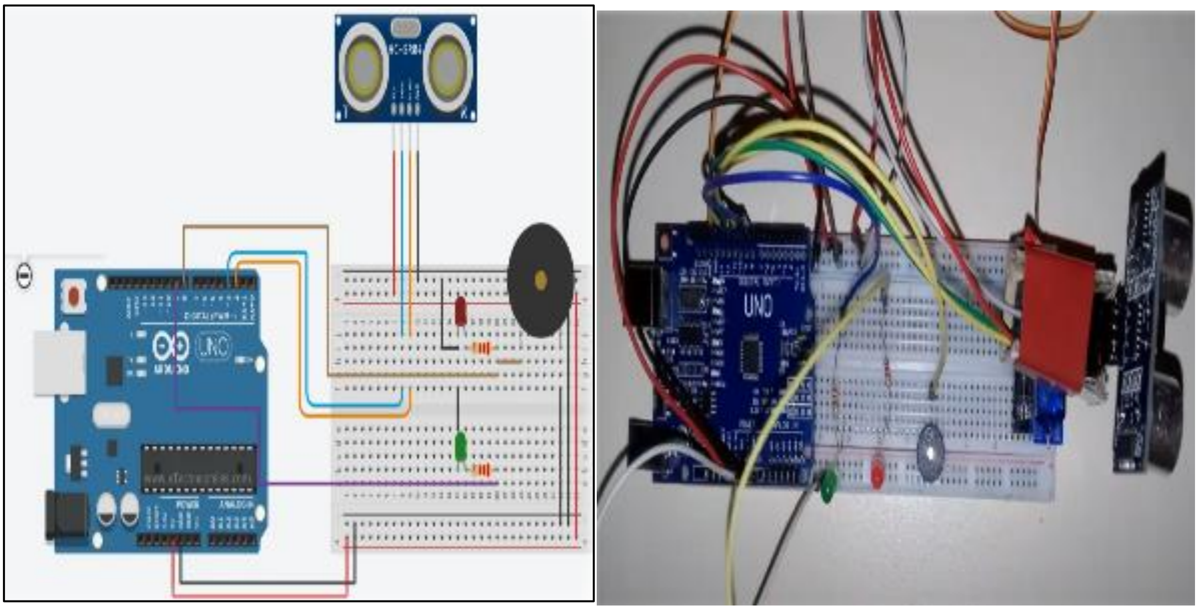
The transceivers work on a principle similar to radar or sonar which evaluate attributes of the objects (targets) by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in the air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module.

Figure 5 Transceivers

2.3.5. Arduino UNO

Arduino is an open-source project that offers microcontroller kits for building digital devices and interactive objects that can sense and control the physical world. These systems have digital and analog input/output pins, which let you connect to expansion boards called shields and other circuits. The Arduino project also includes an integrated development environment (IDE) based on the Processing language, and it supports C and C++. The Arduino UNO board is the most popular choice for developers working with Arduino. It is especially useful for testing both software and hardware during development.

Figure 6 Arduino UNO



(a) Schematic Assembly (b) Physical Assembly

Figure 7 Complete Assembly Arduino-Controlled Ultrasonic Radar

2.4. Calculations

30^0 Coordinate

$$x = (width - width \times 0.4994) + \frac{width}{2} cos30° \qquad (1)$$

$$y = (width - width \times 0.4994) + \frac{width}{2} sin30° \qquad (2)$$

60^0 Coordinate

$$x = (width - width \times 0.503) + \frac{width}{2} cos60° \qquad (3)$$

$$y = (width - width \times 0.0888) + \frac{width}{2} sin60° \qquad (4)$$

90^0 Coordinate

$$x = (width - width \times 0.507) + \frac{width}{2} cos90° \qquad (5)$$

$$y = (width - width \times 0.0833) + \frac{width}{2} sin90° \qquad (6)$$

3. Results

For this test, an obstacle proximity was measured via the sensor at different distances and compared these readings to a tape measure. The obstacle closer to the sensor in steps to find the shortest distance it could detect. The system was designed to make a beep sound feedback when something unknown comes within 5 cm range of the sensor and to turn the screen completely red. However, the screen started turning red when the obstacle was within 10 cm. When the obstacle moves out of range, the radar signal on the screen turns green. The performance of an ultrasonic sensor radar can be improved by increasing the frequency of the ultrasonic waves, using more advanced processing algorithms, adding more sensors, and implementing advanced calibration methods. Depending on how it will be used, some adjustments may however, be needed to ensure the sensor works well in real-world situations. The experimental results illustrate the developed Arduino-controlled ultrasonic radar system to accurately detect obstacles at different angular positions and distances. As the servo motor rotates from 30° to 150°, the ultrasonic sensor continuously measures the distance of nearby objects using echo pulse reflection. The results indicate that shorter distances correspond to objects located within the hazard detection zone, while larger distances represent safe regions outside the detection range as shown in Figure 8. The increasing bar heights in the graphical result confirm that the radar system successfully performs dynamic distance measurement and real-time hazard detection with reliable angular scanning capability.

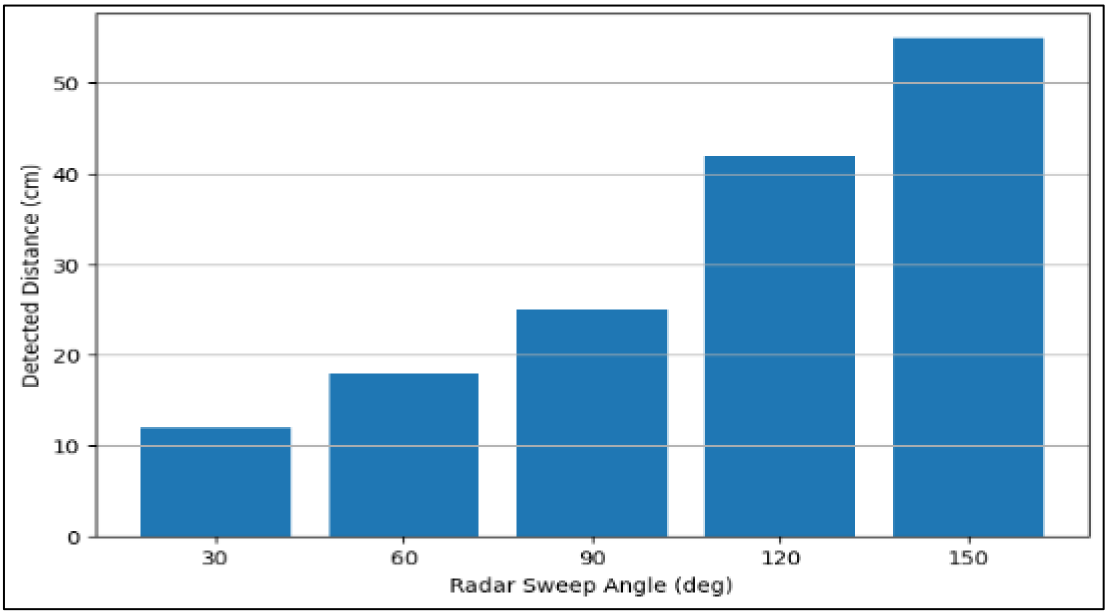


Figure 8 Ultrasonic Radar Hazard Detection Outcome

4. Conclusion

The developed radar-based ultrasonic system provides a simple and effective approach for real-time obstacle detection. Its integration with visualization tools enhances situational awareness and demonstrates strong potential for IoT-enabled sensing applications. The system architecture consists of an ultrasonic sensing unit, a microcontroller-based processing unit, and a visualization interface. The ultrasonic sensor continuously measures distance by emitting and receiving acoustic signals, while the servo motor provides angular scanning across a defined range.

The system also successfully performs dynamic distance measurement and hazard detection using ultrasonic pulse reflection principles. The implementation of Arduino, servo motor scanning, and Processing IDE visualization enables real-time radar monitoring with reliable obstacle localization. The developed system demonstrates strong potential for applications in robotics, autonomous navigation, industrial monitoring, and intelligent safety systems.

The microcontroller serves as the central processing unit, coordinating sensor measurements, controlling servo movement, and transmitting real-time data via serial communication. A buzzer module is integrated for proximity alerts when objects fall within a predefined safety threshold. The acquired data is transmitted to a Processing-based graphical interface, where it is visualized as a radar display, enabling real-time monitoring of object position and distance. Field testing and data analysis can be used to pinpoint performance problems and offer suggestions for future development.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] D. Burden, “Streets and Sidewalks, People and Cars THE CITIZENS’ GUIDE TO TRAFFIC CALMING Acknowledgements.” [Online]. Available: www.lgc.org

[2] W. Song, Y. Yang, M. Fu, F. Qiu, and M. Wang, “Real-Time Obstacles Detection and Status Classification for Collision Warning in a Vehicle Active Safety System,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 758–773, Mar. 2018, doi: 10.1109/TITS.2017.2700628.

[3] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice, “Autonomous vehicle control systems — a review of decision making,” Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 225, no. 2, pp. 155–195, Mar. 2011, doi: 10.1177/2041304110394727.

[4] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice, “Autonomous vehicle control systems — a review of decision making,” Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 225, no. 2, pp. 155–195, Mar. 2011, doi: 10.1177/2041304110394727.

[5] S.-L. Chen, “Review of Laser-Generated Ultrasound Transmitters and Their Applications to All-Optical Ultrasound Transducers and Imaging,” Applied Sciences, vol. 7, no. 1, p. 25, Dec. 2016, doi: 10.3390/app7010025.

[6] C.-Y. Chen, R.-C. Hwang, and Y.-J. Chen, “A passive auto-focus camera control system,” Appl. Soft Comput., vol. 10, no. 1, pp. 296–303, Jan. 2010, doi: 10.1016/j.asoc.2009.07.007.

[7] X. Yu and M. Marinov, “A Study on Recent Developments and Issues with Obstacle Detection Systems for Automated Vehicles,” Sustainability, vol. 12, no. 8, p. 3281, Apr. 2020, doi: 10.3390/su12083281.

[8] K. Bhatt, C. Agrawal, and A. M. Bisen, “A Review on Emerging Applications of IoT and Sensor Technology for Industry 4.0,” Wirel. Pers. Commun., vol. 134, no. 4, pp. 2371–2389, Feb. 2024, doi: 10.1007/s11277-024-11054-x.

[9] D. Burden, “Streets and Sidewalks, People and Cars THE CITIZENS’ GUIDE TO TRAFFIC CALMING Acknowledgements.” [Online]. Available: www.lgc.org

[10] Neetu Settia and Krishan Mohan Mandal, “Exploring the Potential of Arduino and Ultrasonic Sensor Integration in Radar Systems,” International Research Journal on Advanced Engineering Hub (IRJAEH), vol. 2, no. 05, pp. 1530–1536, May 2024, doi: 10.47392/irjaeh.2024.0210.

[11] C. Ersü, E. Petlenkov, and K. Janson, “A Systematic Review of Cutting-Edge Radar Technologies: Applications for Unmanned Ground Vehicles (UGVs),” Sensors, vol. 24, no. 23, p. 7807, Dec. 2024, doi: 10.3390/s24237807.

[12] I. Munasinghe, A. Perera, and R. C. Deo, “A Comprehensive Review of UAV-UGV Collaboration: Advancements and Challenges,” Journal of Sensor and Actuator Networks, vol. 13, no. 6, p. 81, Nov. 2024, doi: 10.3390/jsan13060081.

[13] S. Ahmed, A. Joret, N. Katiran, and M. S. Sulong, “Evolution of Information, Communication and Computing Systems (EICCS) https Human Detection Techniques for Search and Rescue of Trapped Victims Under Debris: A Review.”

[14] J. Le Kernec et al., “Radar Signal Processing for Sensing in Assisted Living: The Challenges Associated With Real-Time Implementation of Emerging Algorithms,” IEEE Signal Process. Mag., vol. 36, no. 4, pp. 29–41, Jul. 2019, doi: 10.1109/MSP.2019.2903715.

Appendices

```
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
#include <Servo.h>

Servo radarServo;

const int trigPin = 10;
const int echoPin = 11;

long duration;
float distance;

void setup() {
  Serial.begin(9600);

  pinMode(trigPin, OUTPUT);
```

```
 pinMode(echoPin, INPUT);

 radarServo.attach(12);
}

void loop() {

 for(int angle = 0; angle <= 180; angle++) {

   radarServo.write(angle);
   delay(30);

   digitalWrite(trigPin, LOW);
   delayMicroseconds(2);

   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);

   digitalWrite(trigPin, LOW);

   duration = pulseIn(echoPin, HIGH);

   distance = duration * 0.034 / 2;

   Serial.print(angle);
   Serial.print(",");
   Serial.print(distance);
   Serial.print(".");
 }

 for(int angle = 180; angle >= 0; angle--) {

   radarServo.write(angle);
   delay(30);

   digitalWrite(trigPin, LOW);
   delayMicroseconds(2);

   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);

   digitalWrite(trigPin, LOW);

   duration = pulseIn(echoPin, HIGH);

   distance = duration * 0.034 / 2;

   Serial.print(angle);
   Serial.print(",");
   Serial.print(distance);
   Serial.print(".");
 }
}
```

Processing IDE Radar Visualization Code

```
import processing.serial.*;
Serial myPort;
String angle = "";
String distance = "";
```

```
String data = "";
String noObject;

float pixsDistance;
int iAngle, iDistance;

void setup() {

  size(1200, 700);
  smooth();

  myPort = new Serial(this, "COM3", 9600);
  myPort.bufferUntil('.');
}

void draw() {

  fill(0,4);
  rect(0,0,width,height);

  fill(0,255,0);

  drawRadar();
  drawLine();
  drawObject();
  drawText();
}

void serialEvent(Serial myPort) {
  data = myPort.readStringUntil('.');
  data = data.substring(0,data.length()-1);
  int index1 = data.indexOf(",");
  angle = data.substring(0, index1);
  distance = data.substring(index1+1);
  iAngle = int(angle);
  iDistance = int(distance);
}

void drawRadar() {

  pushMatrix();

  translate(width/2,height-height*0.074);

  noFill();
  strokeWeight(2);
  stroke(0,255,0);

  arc(0,0,width,width,PI,TWO_PI);

  popMatrix();
}

void drawLine() {

  pushMatrix();

  strokeWeight(4);
  stroke(0,255,0);
```

```
 translate(width/2,height-height*0.074);

 line(0,0,(height-height*0.12)*cos(radians(iAngle)),
 -(height-height*0.12)*sin(radians(iAngle)));

 popMatrix();
}

void drawObject() {

 pushMatrix();

 translate(width/2,height-height*0.074);

 strokeWeight(9);
 stroke(255,0,0);

 pixsDistance = iDistance*6;

 if(iDistance < 40) {

  line(pixsDistance*cos(radians(iAngle)),
  -pixsDistance*sin(radians(iAngle)),
  (width-width*0.505)*cos(radians(iAngle)),
  -(width-width*0.505)*sin(radians(iAngle)));
 }

 popMatrix();
}

void drawText() {

 fill(0,255,0);

 textSize(25);

 text("Arduino Ultrasonic Radar System",
 width-width*0.875,height-height*0.0277);

 text("Angle: " + iAngle + "°",
 width-width*0.48,height-height*0.0277);

 text("Distance: " + iDistance + " cm",
 width-width*0.26,height-height*0.0277);
}
```