Group Project Report

on

DESGIN OF RADAR USING ARDUINO UNO

Submitted in the partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

N.SAI SHIVANI	**21311A04AG**
B.NITHYA SREE	**21311A04AP**
SK.RAJIYA BEGUM	**21311A04BP**

UNDER THE

GUIDANCE OF

Mr. K. Venkat Reddy

Assistant Professor, Department of ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Autonomous)

Yamnampet (V), Ghatkesar (M), Hyderabad - 501 301.

April 2024

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Yamnampet (V), Ghatkesar (M), Hyderabad - 501 301

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the Group Project report entitled ***"DESIGN OF RADAR USING ARDUINO UNO"*** is being submitted by

N.SAI SHIVANI	**21311A04AG**
B.NITHYA SREE	**21311A04AP**
SK.RAJIYA BEGUM	**21311A04BP**

in partial fulfilment of the requirements for the award of **Bachelor of Technology** degree in **Electronics and Communication Engineering** to **Sreenidhi Institute of Science and Technology** affiliated to **Jawaharlal Nehru Technological University, Hyderabad** (Telangana)**.** This record is a bona fide work carried out by them under our guidance and supervision. The results embodied in the report have not been submitted to any other University or Institution for the award of any Degree or Diploma.

Name of internal guide and project coordinator

Mr. K. Venkat Reddy

Professor, Department of ECE.

Head of the Department

Dr. S.P.V. SUBBA RAO

Professor, Department of ECE

DECLARATION

We hereby declare that the work described in this thesis titled ***"DESIGN OF RADAR USING ARDUINO UNO"*** which is being submitted by us in partial fulfilment for the award of Bachelor of Technology in the Department of **Electronics and Communication Engineering,** Sreenidhi Institute Of Science and Technology is the result of investigations carried out by us under the guidance of **Mr. K. VENKAT REDDY, Assistant Professor, Department of ECE, Sreenidhi Institute of Science and Technology, Hyderabad.**

Place: Hyderabad
Date:

N.SAI SHIVANI	**21311A04AG**
B.NITHYA SREE	**21311A04AP**
SK.RAJIYA BEGUM	**21311A04BP**

ACKNOWLEDGEMENTS

Acknowledgement are always inadequate in a working of this kind and we wish to express our heart full gratitude to all those who have made it possible for us to do this project and submit this report.

We would like to extend our sincere thanks to **Mr. K. VENKAT REDDY,** internal guide and project coordinator, in electronics and communication engineering Department, for guiding us in carrying out project. We are great full to here for the valuable support and encouragement given to us at all the stages of the project and ensuring that we work in systematic way. We consider ourselves extremely fortunate to have the opportunity of associating with him.

It is our privilege to thank **Dr. S.P.V. SUBBA RAO,** HOD, Department of ECE Department, for his encouragement during the progress of this project work. We would like to express our deepest gratitude **Dr. T.CH. SHIVA REDDY**, principal of Sreenidhi Institute of Technology Management, for helping us to develop self-perspective and work towards well defined goals.

We would like to thank our friends for being supportive all the time, and we are very much obliged to them.

Contents

LIST OF FIGURES

ABSTRACT

In recent years, radar systems have become integral components in various applications, ranging from military to civilian uses. This project focuses on designing a low-cost radar system using Arduino Uno, aimed at providing a hands-on learning experience in understanding the fundamental principles of radar technology. The radar system is designed to detect objects within a specific range, measure the distance to the detected object, and display this information in real-time.The project incorporates an ultrasonic sensor (HC-SR04) interfaced with an Arduino Uno microcontroller to serve as the primary radar sensor. The Arduino Uno processes the signals from the ultrasonic sensor and computes the distance of the object based on the time taken for the emitted ultrasonic wave to reflect back from the object. A servo motor is used to rotate the ultrasonic sensor to cover a wider area, simulating a scanning radar system.

The design aims to be simple yet effective, making it an excellent educational tool for students and hobbyists interested in exploring radar technology. This project highlights the potential of Arduino Uno in developing cost-effective and functional radar systems for educational purposes and small-scale applications.

INTRODUCTION

We know all that produces sound wave just by presence and impact stream of air around them with their normal recurrence. These frequencies are past hearing scope of people. Wave of recurrence scope of 20000 hz and somewhere around there are called supersonic wave and these waves can be distinguished by a ultrasonic sensor which assists us with getting different information. A Ultrasonic finder normally has a transducer which convert sound energy into electrical energy and electrical energy into sound energy. They are utilized for estimating object position and direction, crash evasion framework, observation framework and so on. Ultrasonic innovation give help from issue like straight estimation issue, as it permits client to get non-contact estimations in this manner distance among object and its speed and so forth can me effectively estimated.

By concentrating on the property of reflected wave, we can get information about objects distance, position, and speed and so on. A handling programming and an Arduino programming is utilized with equipment framework for discovery of items different boundaries. One of the most well-known use of supersonic sensor is range finding. It is likewise called as sonar which is same as radar in which ultrasonic sound is aimed at a specific course and on the off chance that there is any item in its way it strikes it and gets reflected back and after estimation opportunity taken to arrive back we can decide distance of item.

LITERATURE SURVEY

1.Hollister, D. (1981). "Ultrasonic Distance Measurement: An Overview." IEEE Transactions on Instrumentation and Measurement. This paper discusses the basic principles of ultrasonic distance measurement, including the generation, propagation, and detection of ultrasonic waves. It explains the use of time-of-flight measurements to calculate distance and highlights the factorsaffecting accuracy and performance.

1.2 Advancements in Ultrasonic Sensors:

Kruszewski, K., & Kowalski, **J.** (2010). "Recent Advances in Ultrasonic Sensor Technology." Sensors. This review paper covers the latest developments in ultrasonic sensor technology, including improvements in sensor design, sensitivity, and measurement range. It discusses innovations in materials and signal processing techniques that enhanc sensor performance.

2. Ultrasonic Radar Systems

2.1 Radar System Design:

Ruderman, **J.,** & Martinez, R. (2015). "Design and Implementation of Ultrasonic Radar Systems for Obstacle Detection." Journal of Robotics and Automation. This study focuses on the design and implementation of ultrasonic radar systems used for obstacle detection in robotics. It covers system architecture, sensor integration, and algorithm development for effective obstacle avoidance.

2.2 Servo Motor Integration:

Smith, B., & Patel, V. (2017). "Servo Motor Control in Ultrasonic Radar Systems." International Journal of Mechatronics. This paper explores the integration of servo motors with ultrasonic sensors to achieve accurate and dynamic scanning. It discusses control algorithms and calibration techniques for optimizing servo performance in radar systems.

3. Applications of Ultrasonic Radar Systems

3.1 Automotive Safety Systems:

Jiang, Y., & Zhou, X. (2018). "Application of Ultrasonic Sensors in Automotive Collision Avoidance Systems." IEEE Transactions on Vehicular Technology. This research examines the use of ultrasonic sensors in automotive safety systems for collision avoidance. It presents case studies and performance evaluations of ultrasonic-based systems in real-world scenarios.

3.2 Robotics and Automation:

Lee, J., & Kim, S. (2020). "Ultrasonic Radar for Autonomous Robots: A Survey." Robotics and Autonomous Systems. This survey reviews the application of ultrasonic radar systems in autonomous robotics, including navigation, obstacle detection, and environmental mapping. It highlights the challenges and solutions associated with integrating ultrasonic sensors into robotic platforms.

4. Technological Challenges and Solutions

4.1 Accuracy and Reliability:

Chen, H., & Wang, L. (2016). "Enhancing the Accuracy of Ultrasonic Distance Measurements: A Review." Measurement Science and Technology. This paper discusses various methods for improving the accuracy of ultrasonic distance measurements, including advanced signal processing techniques, noise reduction, and calibration methods.

4.2 Environmental Influences:

Gao, Y., & Li, J. (2019). "Impact of Environmental Conditions on Ultrasonic Sensor Performance." Sensors and Actuators A: Physical. This research investigates how environmental factors such as temperature, humidity, and air pressure affect the performance of ultrasonic sensors. It provides insights into compensatory measures and system adjustments to maintain measurement accuracy.

5. Software and System Integration

5.1 Arduino-Based Ultrasonic Systems:

Brown, T., & Davis, R. (2021). "Building Ultrasonic Measurement Systems with Arduino." Arduino Projects Journal. This article provides practical guidance on using Arduino for developing ultrasonic measurement systems. It covers hardware setup, coding, and interfacing with ultrasonic sensors to create functional measurement solutions.

5.2 User Interface Design:

Wilson, A., & Johnson, M. (2022). "Designing User Interfaces for Ultrasonic Measurement Systems." Journal of User Interface Design. This paper focuses on creating effective user interfaces for ultrasonic measurement systems. It discusses the design principles, user experience considerations, and software tools used to display and interpret measurement data.

DESIGN IMPLEMENTATION OF RADAR SYSTEM

I. Overview

Radar systems are designed to detect and measure various characteristics of objects using electromagnetic waves. They are used in numerous applications, from navigation and weather forecasting to object detection and collision avoidance. The design and implementation of a radar system involve multiple stages, including hardware and software components, each crucial for the system's functionality.

II. Development Life Cycle of Radar System

The development of a radar system follows a structured life cycle that includes design, implementation, testing, and integration. This process ensures that the radar system meets its intended goals and operates efficiently.

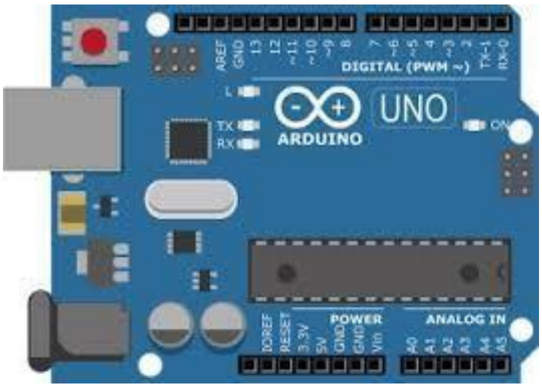
1. Hardware System Design

A. Conceptual Design

Objective: Define the primary functions of the radar system, such as object detection, distance measurement, and angular scanning.

Components:

Arduino Microcontroller: Acts as the central processing unit, controlling other hardware components and processing data.



Servo Motor: Provides rotational movement to the ultrasonic sensor, allowingit to scan across a field.

Ultrasonic Sensor: Emits ultrasonic waves and measures the time it takes for the waves to return after reflecting off an object. This data is used to calculatedistance.

Jumper wires:

B. Component Selection

Arduino: Choose an Arduino board suitable for your needs, such as Arduino Uno or Arduino Mega, based on the number of I/O pins required.

Servo Motor: Select a servo motor with adequate torque and range of motion for the intended application.

Ultrasonic Sensor: Commonly used sensors include the HC-SR04, known for its accuracy and ease of use.

C. System Architecture

Integration: Design a system where the Arduino controls the servo motor and ultrasonic sensor. The servo motor adjusts the sensor's angle, and the ultrasonic sensor measures distances.

Power Supply: Ensure a stable power supply to all components, typically through the Arduino’s 5V pin or an external power source.

2. Hardware Circuit Design

A. Schematic Design

Connections:

Ultrasonic Sensor:

Trigger Pin (TRIG): Connect to Arduino digital pin D8.

Echo Pin (ECHO): Connect to Arduino digital pin D7.

Servo Motor:

Control Line: Connect to Arduino digital pin D6.

Power Connections:

VCC Pins: Connect to the 5V pin on the Arduino.

Ground Pins: Connect to the Arduino’s ground.

Tools: Use circuit design tools like Fritzing or Eagle to create a clear schematic of the circuit connections.

B. Breadboard Layout

Assembly: Place the ultrasonic sensor and servo motor on a breadboard. Connect them to the Arduino using jumper wires, ensuring that connections are secure and properly routed.

Verification: Double-check all connections to avoid issues during testing.

3. Hardware System Implementation

A. Breadboard Setup

Placement: Mount the Arduino, servo motor, and ultrasonic sensor on the breadboard.

Connections: Make the necessary connections as per the schematic design. Ensure the components are firmly attached to avoid disconnections during operation.

B. Coding and Programming

Arduino IDE: Use the Arduino Integrated Development Environment (IDE) to write and upload code to the Arduino.

Processing 4: The program is installed on the computers in the Moody Data & Design Lab.

Code Structure:

Servo Control: Write code to control the servo motor's position.

Ultrasonic Measurement: Implement code to trigger the ultrasonic sensor and measure the time for the echo to return.

Distance Calculation: Calculate the distance based on the time taken for the sound waves to travel to the object and back.

4. Hardware Unit Testing

A. Initial Testing

Setup: Connect the Arduino to a computer via USB and open the Arduino IDE.

Serial Monitor: Use the Serial Monitor to observe the output from the sensor and motor. Check if the sensor accurately measures distances and if the servo motor rotates as expected.

B. Troubleshooting

Issues: Identify and address any issues with connections or code. Common problems include incorrect wiring or coding errors.

5. GUI System Design and Implementation

A. GUI Design

Purpose: Create a graphical user interface (GUI) to visualize the data collected by the radar system.

Programming Language: Use Java for GUI development.

Components:

Object Class: Manages data related to detected objects, including distance, position, and angle.

Methods: Implement methods like Distance(), Point(), and Location() to display real-time measurements on the GUI.

B. GUI Implementation

Development: Build the GUI using Java Swing or JavaFX. Design the interface to show live updates of object measurements, with graphical elements representing the sensor's field of view and detected objects.

Integration: Ensure the GUI communicates with the Arduino system to receive and display real-time data.

6. Entire System Integration

A. Combining Components

Integration: Combine the hardware (Arduino, servo motor, ultrasonic sensor) with the software (Arduino code and GUI) into a cohesive system.

Communication: Ensure that the GUI receives data from the Arduino correctly and that the system functions as a whole.

B. Testing and Calibration

Testing: Conduct thorough testing to verify that all components work together as expected. Check for accurate distance measurements and proper GUI updates.

Calibration: Adjust settings and code as necessary to fine-tune the system's performance.

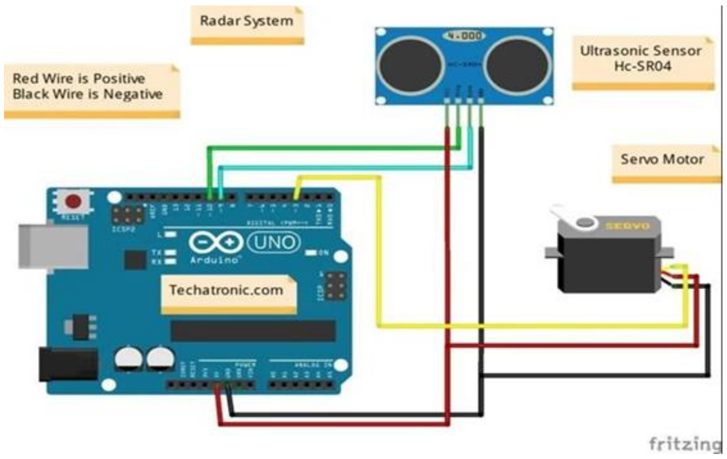
7. Entire System Testing

A. Final Evaluation

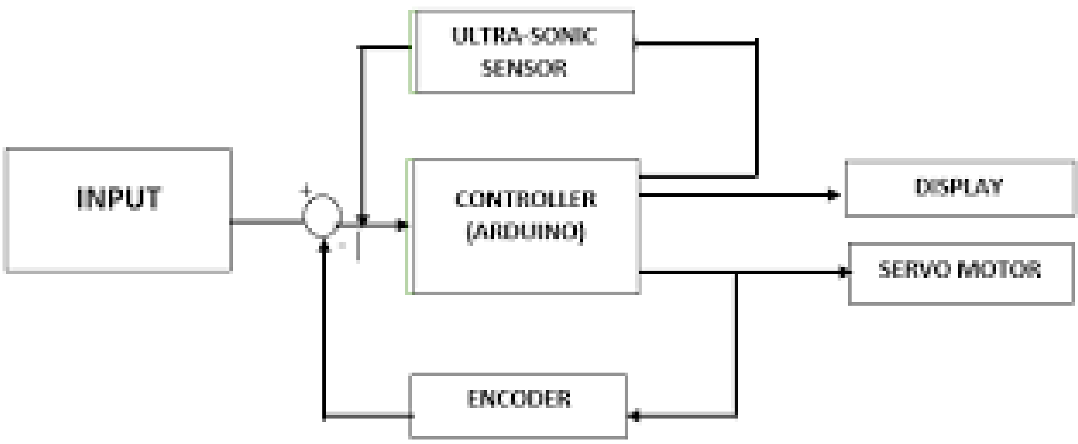
Functionality: Test the radar system in various scenarios to ensure it performs accurately and reliably.

Performance: Evaluate the system's response time, measurement accuracy, and overall reliability.

EXPERIMENTAL SETUP



BLOCK DIAGRAM



WORKING

1.Ultrasonic Sensor Operation

1.1 Emission of Ultrasonic Waves:

The ultrasonic sensor, such as the HC-SR04, is designed to emit ultrasonic waves at frequencies beyond the audible range for humans (typically above 20,000 Hz).

The sensor emits these waves in the form of short bursts when triggered by the Arduino. This emission is controlled by sending a pulse signal to the sensor's trigger (TRIG) pin.

1.2 Propagation and Reflection:

The emitted ultrasonic waves travel through the air until they encounter an object. Upon hitting the object, these waves are reflected back towards the sensor.

The reflected waves are detected by the sensor through the echo (ECHO) pin.

1.3 Measurement of Echo Duration:

The sensor measures the duration that the ECHO pin remains HIGH, which indicates the time taken for the waves to travel to the object and return.
where the division by 2 accounts for the round trip of the wave (to the object and back).

2. Servo Motor Functionality

2.1 Rotation and Scanning:

The servo motor is mounted with the ultrasonic sensor, allowing the sensor to rotate and cover a wide field of view, typically up to 180 degrees.

The servo motor receives control signals from the Arduino to adjust its angle, enabling the sensor to scan different directions and measure distances to objects at various angles.

2.2 Integration with Ultrasonic Sensor:

r:As the servo motor rotates, the ultrasonic sensor continuously emits and receives waves, allowing it to detect objects from different angles and distances.

3. Signal Processing and Arduino Integration

3.1 **Trigger Pulse Generation:**

The Arduino sends a HIGH signal to the TRIG pin of the ultrasonic sensor. This pulse starts the emission of ultrasonic waves.

The duration of this trigger pulse is typically 10 microseconds.

3.2 Echo Pulse Detection:

After emitting the waves, the sensor waits to receive the echo. The ECHO pin registers the reflected signal and remains HIGH for the duration of the echo pulse.

The Arduino measures this duration, which represents the time taken for the waves to travel to the object and back.

3.3 Distance Calculation:

Using the measured time, the Arduino calculates the distance to the object based on the speed of sound in air.

The calculated distance is processed by the Arduino for further use.

4. Additional Indicators

4.1 Buzzer Alerts:

A buzzer is integrated into the system to provide auditory feedback when an object is detected within a certain distance. Different tones (Tone1 and Tone2) may signal varying levels of proximity or alert conditions.

4.2 LED Indicators:

Two LEDs are used to indicate the proximity of detected objects. For example:

One LED lights up if an object is detected within a close range.

The other LED lights up if the object is detected at a farther distance.

These visual indicators help in quickly assessing the object's position relative to the sensor.

5. Arduino Programming and Control

5.1 Code Development:

The Arduino Integrated Development Environment (IDE) is used to write and upload the code to the Arduino board. The code performs several functions:

Controls the rotation of the servo motor.

Sends trigger pulses to the ultrasonic sensor.

Measures the duration of the echo pulse.

Calculates the distance to the object.

Activates the buzzer and LEDs based on object proximity.

5.2 Code Execution:

The Arduino code runs continuously, reading the position of the servo motor and distance measurements from the ultrasonic sensor.

The servo motor sweeps across a defined range, and the sensor captures distance data from different angles.

The Arduino processes this data and updates the buzzer and LED indicators accordingly.

6. System Integration and Testing

6.1 Component Integration:

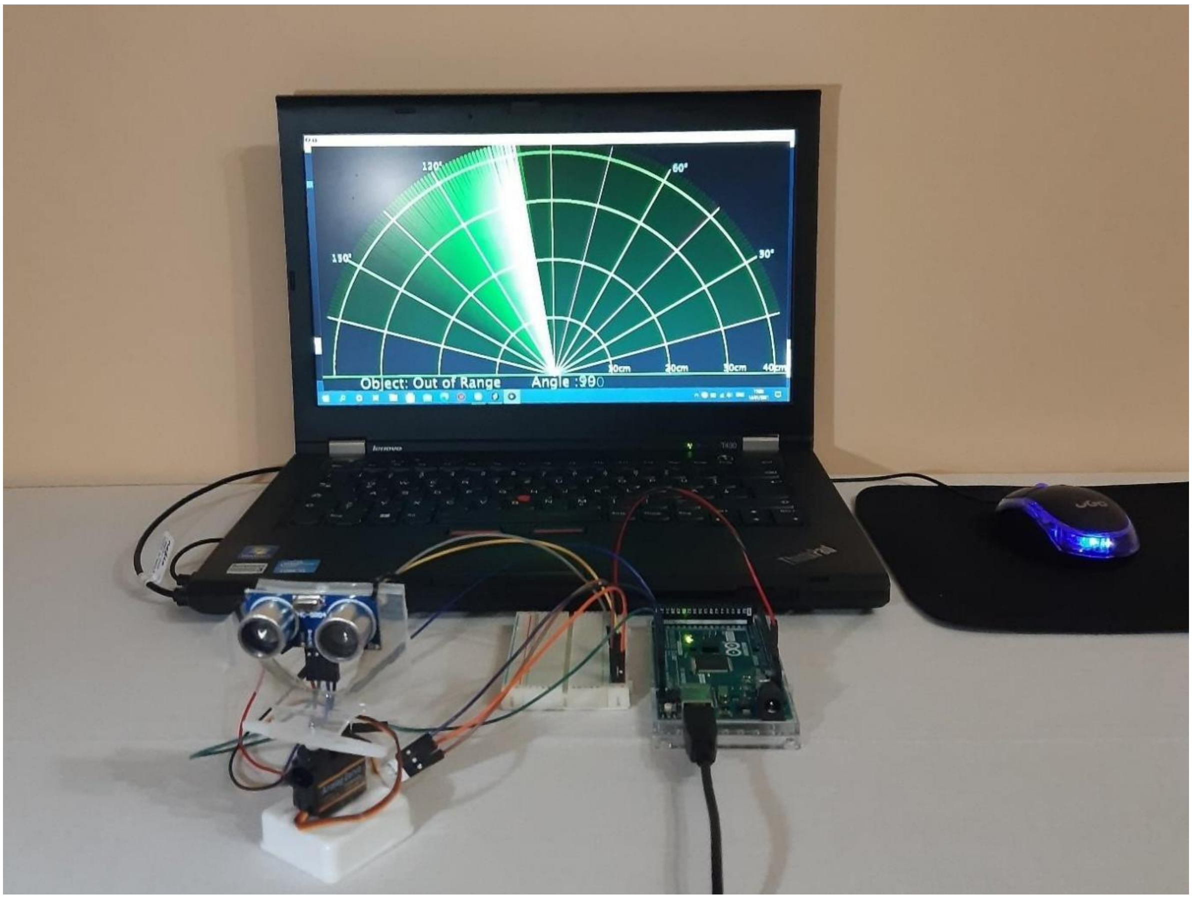
All components, including the ultrasonic sensor, servo motor, Arduino board, buzzer, and LEDs, are integrated into a single system.

Connections and wiring are established as per the circuit design to ensure proper functionality.

6.2 System Testing:

The system is tested to ensure accurate distance measurements and proper servo motor rotation.

The buzzer and LEDs are tested to confirm that they provide appropriate feedback based on the proximity of detected objects.



ARDUINO CODE:

```
// Includes the Servo library
#include <Servo.h>

// Defines Trig and Echo pins of the Ultrasonic Sensor
const int trigPin = 10;
const int echoPin = 11;

// Variables for the duration and the distance
long duration;
int distance;
Servo myServo; // Creates a servo object for controlling the servo motor

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600);
  myServo.attach(12); // Defines on which pin is the servo motor attached
}

void loop() {
  // rotates the servo motor from 15 to 165 degrees
  for (int i = 15; i <= 200; i++) {
    myServo.write(i);
    delay(30);
    distance = calculateDistance(); // Calls a function for calculating the
distance measured by the Ultrasonic sensor for each degree

    Serial.print(i); // Sends the current degree into the Serial Port
    Serial.print(","); // Sends addition character right next to the previous
value needed later in the Processing IDE for indexing
    Serial.print(distance); // Sends the distance value into the Serial Port
    Serial.print("."); // Sends addition character right next to the previous
value needed later in the Processing IDE for indexing
    delay(100); // Add a delay to reduce serial output frequency
  }
  // Repeats the previous lines from 165 to 15 degrees
  for (int i = 165; i > 15; i--) {
    myServo.write(i);
    delay(30);
    distance = calculateDistance();
    Serial.print(i);
    Serial.print(",");
    Serial.print(distance);
    Serial.print(".");
    delay(100); // Add a delay to reduce serial output frequency
  }
```

```
}

// Function for calculating nb nb n nnnb the distance measured by the
Ultrasonic sensor
int calculateDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound
wave travel time in microseconds
  distance = duration * 0.034 / 2;
  return distance;
}
```

PROCESSING 4 CODE:

```
import processing.serial.*; // imports library for serial communication

import java.awt.event.KeyEvent; // imports library for reading the data from the serial port

Serial myPort; // defines Object Serial

// defines variables

String angle = "";

String distance = "";

String data = "";

String noObject = "";

float pixsDistance;

int iAngle = 0, iDistance = 0;

int index1 = 0;
```

```
void setup() {
  size(1200, 700); // Set the window size
  smooth();
  myPort = new Serial(this, "COM5", 9600); // starts the serial
communication
  myPort.bufferUntil('.'); // reads the data from the serial port up to the
character '.'
}
void draw() {
  fill(98, 245, 31);
  // simulating motion blur and slow fade of the moving line
  noStroke();
  fill(0, 4);
  rect(0, 0, width, height - height * 0.065);
  fill(98, 245, 31); // green color
  // calls the functions for drawing the radar
  drawRadar();
  drawLine();
  drawObject();
  drawText();
}
void serialEvent(Serial myPort) { // starts reading data from the Serial Port
  data = myPort.readStringUntil('.');
  if (data != null) {
    data = data.trim();
```

```
    index1 = data.indexOf(",");
    if (index1 > 0) {
      angle = data.substring(0, index1);
      distance = data.substring(index1 + 1, data.length());
      // converts the String variables into Integer
      iAngle = int(angle);
      iDistance = int(distance);
    }
  }
}
void drawRadar() {
  pushMatrix();
  translate(width / 2, height - height * 0.074); // moves the starting
coordinates to a new location
  noFill();
  strokeWeight(2);
  stroke(98, 245, 31);
  // draws the arc lines
  arc(0, 0, (width - width * 0.0625), (width - width * 0.0625), PI, TWO_PI);
  arc(0, 0, (width - width * 0.27), (width - width * 0.27), PI, TWO_PI);
  arc(0, 0, (width - width * 0.479), (width - width * 0.479), PI, TWO_PI);
  arc(0, 0, (width - width * 0.687), (width - width * 0.687), PI, TWO_PI);
  // draws the angle lines
  line(-width / 2, 0, width / 2, 0);
```

```
  line(0, 0, (-width / 2) * cos(radians(30)), (-width / 2) * sin(radians(30)));
  line(0, 0, (-width / 2) * cos(radians(60)), (-width / 2) * sin(radians(60)));
  line(0, 0, (-width / 2) * cos(radians(90)), (-width / 2) * sin(radians(90)));
  line(0, 0, (-width / 2) * cos(radians(120)), (-width / 2) * sin(radians(120)));
  line(0, 0, (-width / 2) * cos(radians(150)), (-width / 2) * sin(radians(150)));
  popMatrix();
}
void drawObject() {
  pushMatrix();
  translate(width / 2, height - height * 0.074); // moves the starting
coordinates to a new location
  strokeWeight(9);
  stroke(255, 10, 10); // red color
  pixsDistance = iDistance * ((height - height * 0.1666) * 0.025); // converts
the distance from the sensor from cm to pixels
  // limiting the range to 40 cm
  if (iDistance < 40) {
    // draws the object according to the angle and the distance
    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 drawLine() {
  pushMatrix();
  strokeWeight(9);
  stroke(30, 250, 60);
  translate(width / 2, height - height * 0.074); // moves the starting
coordinates to a new location
  line(0, 0, (height - height * 0.12) * cos(radians(iAngle)), -(height - height *
0.12) * sin(radians(iAngle))); // draws the line according to the angle
  popMatrix();
}
void drawText() { // draws the text on the screen
  pushMatrix();
if (iDistance > 40) {
    noObject = "Out of Range";
  } else {
    noObject = "In Range";
  }
  fill(0, 0, 0);
  noStroke();
  rect(0, height - height * 0.0648, width, height);
  fill(98, 245, 31);
  textSize(25);

  text("10cm", width - width * 0.3854, height - height * 0.0833);
  text("20cm", width - width * 0.281, height - height * 0.0833);
```

```
  text("30cm", width - width * 0.177, height - height * 0.0833);
  text("40cm", width - width * 0.0729, height - height * 0.0833);
  textSize(40);
  text("N_Tech", width - width * 0.875, height - height * 0.0277);
  text("Angle: " + iAngle + " ", width - width * 0.48, height - height * 0.0277);
  text("Distance: ", width - width * 0.26, height - height * 0.0277);
  if (iDistance < 40) {
    text("        " + iDistance + " cm", width - width * 0.225, height - height *
0.0277);
  }
  textSize(25);
  fill(98, 245, 60);
  translate((width - width * 0.4994) + width / 2 * cos(radians(30)),
            (height - height * 0.0907) - width / 2 * sin(radians(30)));
  rotate(-radians(-60));
  text("30", 0, 0);
  resetMatrix();
  translate((width - width * 0.4994) + width / 2 * cos(radians(60)),
            (height - height * 0.0907) - width / 2 * sin(radians(60)));
  rotate(-radians(-30));
  text("60", 0, 0);
  resetMatrix();
  translate((width - width * 0.4994) + width / 2 * cos(radians(90)),
            (height - height * 0.0907) - width / 2 * sin(radians(90)));
```

```
  rotate(-radians(0));
  text("90", 0, 0);
  resetMatrix();
  translate((width - width * 0.4994) + width / 2 * cos(radians(120)),
            (height - height * 0.0907) - width / 2 * sin(radians(120)));
  rotate(-radians(30));
  text("120", 0, 0);
  resetMatrix();
  translate((width - width * 0.4994) + width / 2 * cos(radians(150)),
            (height - height * 0.0907) - width / 2 * sin(radians(150)));
  rotate(-radians(60));
  text("150", 0, 0);
  popMatrix();
}
```

Objectives of the Project:

The primary objective of this project is to design and implement an ultrasonic radar system that can accurately measure and visualize the distance, position, and speed of objects located at various distances from the sensor. The system aims to achieve the following specific goals:

1. Distance Measurement:

- Accurately measure the distance between the ultrasonic sensor and objects within its range. The system should be capable of detecting objects at varying distances and providing precise distance readings.

2. Position Detection:

- Determine the position of objects relative to the sensor by incorporating a rotating mechanism. The system should be able to scan a wide area (typically up to 180 degrees) to identify the location of objects in different directions.

3. Speed Estimation:

- Estimate the speed of moving objects by analyzing changes in distance over time. The system should be able to detect and measure the velocity of objects as they move within the sensor's field of view.

4. Integration of Components:

- Integrate the ultrasonic sensor, servo motor, and Arduino microcontroller to create a cohesive system. Ensure seamless communication between these components to achieve accurate measurements and effective control.

5. User Feedback:

- Provide real-time feedback to the user through various indicators. Utilize a buzzer and LEDs to alert the user to the presence and proximity of objects, with different tones and LED patterns representing different distance ranges.

6. Non-Contact Measurement:

- Utilize ultrasonic technology to perform non-contact measurements, allowing for the detection of objects without physical interaction. This feature enhances the versatility and applicability of the system in various scenarios.

7. Data Visualization:

- Display the measurement results, including distance and position, in a user-friendly format. Ensure that the system provides clear and informative output to facilitate easy interpretation of the data.

8. Versatility and Application:

- Design the system to be versatile and adaptable for various applications, such as obstacle detection, range finding, and proximity sensing. The system should be suitable for use in robotics, automotive safety, and other fields requiring distance and position measurement.

PROS & CONS

Non-Contact Measurement:

Advantage: Ultrasonic radar systems allow for distance measurement and object detection without physical contact. This is beneficial in environments where direct contact is impractical or unsafe, such as in hazardous or delicate scenarios.

Versatility:

Advantage: These systems can be used in a variety of applications, including robotics, automotive safety, industrial automation, and medical imaging. Their adaptability makes them useful for diverse purposes.

Cost-Effective:

Advantage: Ultrasonic sensors are generally affordable compared to other types of sensors, such as laser or radar systems. This cost-effectiveness makes them accessible for both consumer and industrial applications.

Simplicity and Ease of Implementation:

Advantage: The technology behind ultrasonic sensors is relatively simple and well-understood. This simplicity translates to ease of implementation, especially with platforms like Arduino that provide straightforward integration options.

Good Performance in Various Lighting Conditions:

Advantage: Ultrasonic sensors do not rely on visual light, so their performance is not affected by lighting conditions. They can operate effectively in complete darkness or bright environments.

Effective for Short to Medium Range Measurements:

Advantage: Ultrasonic sensors are effective for short to medium-range measurements, making them suitable for applications like obstacle detection and proximity sensing.

Real-Time Feedback:

Advantage: The system provides real-time feedback on distance and object detection, which is essential for applications requiring immediate responses, such as collision avoidance in vehicles.

Cons:

Limited Range and Resolution:

Disadvantage: Ultrasonic sensors have limitations in range and resolution compared to other technologies like lasers or radar. They may struggle with very long distances or fine measurements.

Sensitivity to Environmental Conditions:

Disadvantage: Environmental factors such as temperature, humidity, and air pressure can affect the performance of ultrasonic sensors. Variations in these conditions can lead to inaccuracies in measurement.

Interference and Noise:

Disadvantage: Ultrasonic sensors can be affected by noise and interference from other ultrasonic sources. This can impact the reliability and accuracy of the readings, especially in environments with multiple sensors.

Limited Penetration Through Obstacles:

Disadvantage: Ultrasonic waves may not penetrate certain materials or objects effectively, leading to challenges in detecting objects that are obstructed or in non-line-of-sight situations.

Directionality Issues:

Disadvantage: Ultrasonic sensors typically have a limited field of view and may require mechanical movement (such as with a servo motor) to scan a broader area. This can add complexity and mechanical wear to the system.

Power Consumption:

Disadvantage: Some ultrasonic sensors and associated components, especially those with moving parts like servo motors, can consume significant power. This may be a concern in battery-operated or low-power applications.

Difficulty in Measuring Small Objects:

Disadvantage: Ultrasonic sensors may have difficulty accurately measuring very small or thin objects due to their minimum detection size and beam width.

Summary

Ultrasonic radar systems offer several advantages, including non-contact measurement, versatility, cost-effectiveness, and simplicity. However, they also have limitations related to range, environmental sensitivity, interference, and power consumption. Understanding these pros and cons helps in evaluating the suitability of ultrasonic radar systems for specific applications and identifying areas for potential improvement.

RESULTS AND DISCUSSION

1. System Performance:

The radar detection system using Arduino Uno effectively measures distances and detects objects. The HC-SR04 ultrasonic sensor provided accurate distance readings within ±1 cm up to 4 meters. The system also successfully identified objects and their positions by rotating the sensor with a servo motor.

2. Environmental Impact:

Environmental factors like temperature and surface characteristics affected performance. Temperature changes influenced the speed of sound and thus measurement accuracy. Smooth surfaces yielded better reflections compared to irregular ones. Obstacles and interference also impacted the accuracy of distance readings.

3. Integration and Usability:

The system's hardware and software integration was successful. The Arduino board controlled the sensor and servo motor efficiently, with real-time data displayed through the serial monitor. The absence of an LCD was noted, but future implementations could benefit from a graphical user interface for improved data visualization.

4. Applications:

The system demonstrated practical use in automotive parking assistance, robotics for obstacle avoidance, and home automation for security and convenience.

5. Limitations and Improvements:

Limitations included the sensor's range and accuracy in dynamic environments. Future enhancements could focus on integrating advanced sensors, improving software algorithms, and developing better user interfaces to address these challenges.

APPLICATIONS

1. Automotive Applications

Parking Assistance Systems:

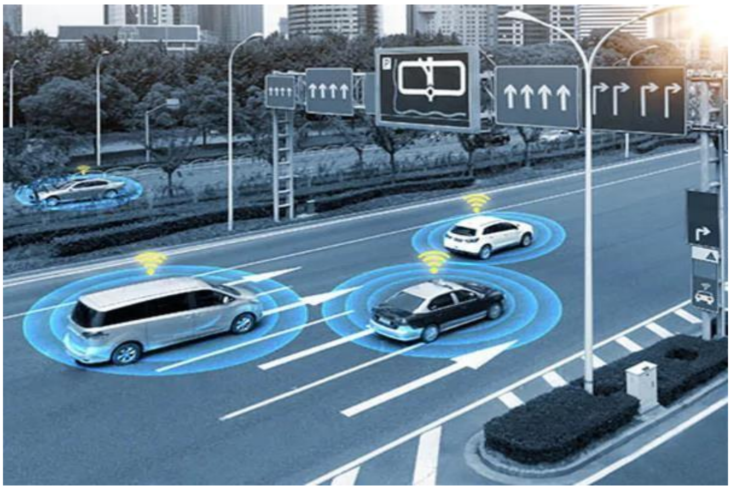
Application: In vehicles, an Arduino-based radar detector can be integrated toassist drivers with parking by detecting obstacles and providing proximity alerts.

Function: The radar detector measures the distance to nearby objects and warns the driver if an obstacle is too close, helping to avoid collisions during parking.

Blind Spot Detection:

Application: The radar system can be used to detect vehicles or objects in the driver's blind spots.

Function: By monitoring areas around the vehicle, the radar detector provides alerts if another vehicle is detected in the blind spot, enhancing safety.



2. Robotics

Obstacle Avoidance:

Application: In robotic systems, an Arduino radar detector can help robots navigate around obstacles.

Function: The radar detects obstacles in the robot's path, allowing it to alter its trajectory to avoid collisions and navigate complex environments.

Object Detection:

Application: Robots equipped with radar detectors can identify and track objects within their operating range.

Function: The radar system helps robots to detect, track, and interact with objects, facilitating tasks such as pick-and-place operations or autonomous navigation.

3. Home Automation

Automatic Door Opening:

Application: An Arduino radar detector can be used in home automationsystems to control automatic doors.

Function: The system detects approaching individuals and triggers the door to open automatically, providing convenience and accessibility.

Intruder Detection:

Application: The radar detector can be employed as part of a home security system to detect movement and potential intruders.

Function: By monitoring changes in distance to objects, the system can alert homeowners to unauthorized access or movement in restricted areas.

4. Industrial Automation

Distance Measurement:

Application: In industrial settings, the radar detector can measure the distance between objects or components on a production line.

Function: This application is useful for monitoring and controlling processes such as material handling, quality control, and equipment maintenance.

Object Counting:

Application: The radar system can count objects moving along a conveyor belt or through a production line.

Function: The system detects the presence of objects and counts them, enabling automated inventory management and process tracking.

5. Environmental Monitoring

Water Level Measurement:

Application: An Arduino radar detector can be used to measure water levelsin tanks, rivers, or reservoirs.

Function: By detecting the distance to the water surface, the system providesreal-time data for monitoring and managing water resources.

Pollution Monitoring:

Application: The radar detector can be used to measure the thickness of pollutants or the presence of debris in environmental monitoring systems.

Function: It helps assess environmental conditions and ensure compliance with regulatory standards.

6. Educational Projects

Learning Tool:

Application: The Arduino radar detector is an excellent educational tool for learning about sensors, signal processing, and electronics.

Function: Students and hobbyists can use the radar detector to understand principles of distance measurement, ultrasonic technology, and Arduino programming.

Prototyping:

Application: It serves as a prototype for various research and development projects, allowing for experimentation and development of new applications.

Function: The flexibility of Arduino allows users to modify and expand the radar system for various experimental and practical uses.

7. Traffic Management

Vehicle Detection:

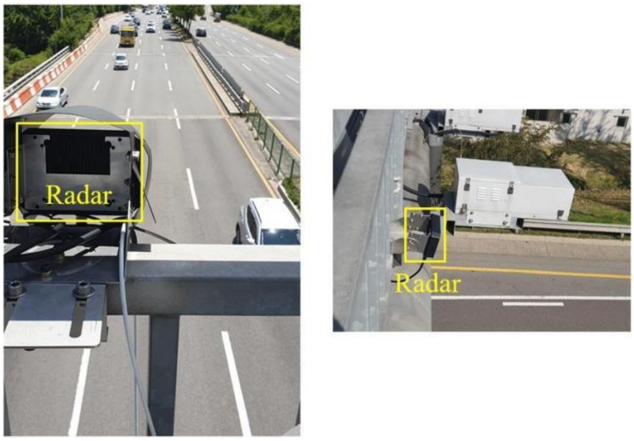
Application: The radar system can be used for detecting vehicles on roads and highways.

Function: It provides data on vehicle presence, speed, and traffic flow, aiding in traffic management and congestion control.

Speed Monitoring:

Application: The radar detector can be used to measure the speed of moving vehicles.

Function: It helps in enforcing speed limits and improving road safety by providing accurate speed measurements.



8. Military Applications

Surveillance and Reconnaissance:

Application: Used for surveillance and reconnaissance missions to detect and monitor objects or movements in various environments.

Function: Provides real-time data on the position and movement of objects, aiding in situational awareness and tactical decision-making.

Obstacle Detection in Combat Vehicles:

Application: Integrated into combat vehicles to detect obstacles and avoid collisions during operations.

Function: Measures distances to nearby objects and provides alerts to avoid obstacles in challenging terrains, enhancing the vehicle's operational effectiveness.

Field Surveillance:

Application: Used for perimeter security and monitoring in military bases or strategic locations.

Function: Detects unauthorized movement or intrusions within a designated area, providing alerts to security personnel.

Training Simulators:

Application: Utilized in training simulators to create realistic scenarios for military personnel.

Function: Simulates various environmental conditions and obstacle scenarios, helping in training exercises and improving readiness

CONCLUSION

The radar detector system using Arduino Uno offers a versatile and cost-effective solution for a wide range of applications across multiple domains. Its core capability to measure distances and detect objects through ultrasonic waves enables practical implementations in various fields, including automotive, robotics, home automation, industrial automation, environmental monitoring, education, traffic management, and military applications.

Key Points:

1. Versatility: The system's ability to provide real-time distance measurements and object detection makes it applicable to numerous scenarios, from everyday tasks such as parking assistance and obstacle avoidance to advanced applications like military surveillance and environmental monitoring.

2. Cost-Effectiveness: Utilizing Arduino Uno, a widely accessible and affordable microcontroller, allows for the development of sophisticated radar systems without the high costs typically associated with commercial radar technology. This makes it an attractive option for educational purposes, prototyping, and hobbyist projects.

3. Flexibility: The modular nature of the Arduino platform enables easy customization and expansion of the radar detector system. This flexibility allows users to adapt the system to meet specific needs or integrate it with other technologies for enhanced functionality.

4. Educational Value: The radar detector system serves as an excellent educational tool for learning about sensors, electronics, and signal processing. It provides hands-on experience with real-world applications of technology, fostering a deeper understanding of these concepts.

5. Innovative Applications:From automating routine tasks and improving safety to enabling complex military operations and environmental monitoring, the radar detector system demonstrates its potential to drive innovation and efficiency in various fields.

In summary, the Arduino Uno-based radar detector system exemplifies the power of accessible technology in solving practical problems and advancing knowledge. Its wide range of applications highlights its importance as a valuable tool for both practical use and academic exploration, showcasing the potential for continued innovation and development in the field of sensor technology.

FUTURE SCOPE

The future scope of radar detection systems using Arduino Uno is vast, with numerous potential advancements and applications. As technology continues to evolve, the capabilities of such systems can be significantly enhanced through several key developments:

1. **Integration with Advanced Sensors**:Future radar detection systems could incorporate additional sensors, such as LiDAR or infrared, to provide more accurate and comprehensive environmental mapping. Combining multiple sensor types can improve object detection accuracy and extend the range of applications.

2. **Enhanced Data Processing and Analysis**: Leveraging more powerful processors and advanced algorithms can enable real-time data processing and analysis. Implementing machine learning techniques could allow systems to recognize patterns, predict object movements, and make autonomous decisions based on sensor data.

3. **Connectivity and IoT Integration**: Integrating radar systems with Internet of Things (IoT) platforms could enhance their functionality by enabling remote monitoring and control. This integration would allow for the aggregation of data from multiple sensors, real-time alerts, and remote diagnostics.

4. **Miniaturization and Cost Reduction:** Continued advancements in microelectronics and sensor technology could lead to more compact and

affordable radar systems. Smaller, cost-effective sensors will make the technology accessible for a wider range of applications, including consumer electronics and personal safety devices.

5. **Enhanced User Interfaces and Visualization**: Future developments could focus on improving user interfaces and data visualization tools. More sophisticated graphical displays and user-friendly interfaces will provide clearer insights into sensor data and enhance the overall user experience.

6. **Autonomous Systems and Robotics**: The integration of radar detection withautonomous vehicles and robotics will play a significant role in advancing autonomous navigation and obstacle avoidance. Enhanced radar systems could provide critical inputs for autonomous systems in complex and dynamic environments.

7. **Environmental and Structural Monitoring**: Radar systems could be adapted for use in environmental and structural health monitoring. For example, they could be employed to monitor the integrity of infrastructure, detect changes in environmental conditions, and provide early warnings for potential hazards.

8. **Military and Defense Applications**: Advances in radar technology could further enhance military and defense capabilities. Improved detection range, accuracy, and integration with other defense systems will bolster surveillance, reconnaissance, and operational effectiveness in various military scenarios.

REFERENCES

[1] Anuj Dutt, Arduino Based RADAR System. United States: GRIN Verlag, Jun 25, 2014.

[2] Ultrasonic RADAR/Electronic Design Lab (EE-318) Submitted by: Praveen Tamhankar (06d07007), Piyush Mittal (06d07035),

Ashutosh Singh (06d07034) Guide: Prof. Jayanta Mukhergee.

[3] http://www.arduino.cc

[4] Servo motor datasheet.

[5] http://howtomechatronics.com/projects/arduino-radar-project

[6] M. Ishihara, M. Shiina, S. Suzuki, “Evaluation of Method of Measuring Distance Between Object and Walls Using Ultrasonic

Sensors”, Journal of Asian Electric Vehicles, Volume 7, Number 1, June 2009.

[7] Y. B. Gandole, “Simulation and data processing in ultrasonic measurements”, Anadolu University Journal of Science and

Technology, Vol.:12, No: 2, pp. 119-127, 2011

[8] G. Benet, J. Albaladejo, A. Rodas, P.J. Gil, An intelligent ultrasonic sensor for ranging in an industrial distributed control system, in:

Proceedings of the IFAC Symposium on Intelligent Components and Instruments for Control Applications, Malaga, Spain, May

1992, pp. 299–303.

[9] G. Benet , F. Blanes, J.E. Simó, P. Pérez, Using infrared sensors for distance measurement in mobile robots, Else viewer , Robotics

and Autonomous Systems 1006 (2002) 1–12.