A Mini Project Report
On

Arduino Based Radar System for Object detection and Distance Monitoring

Submitted by:

I Sem BE, “ F ” Section
A.J. Institute of Engineering & Technology, Mangaluru
May, 2025

MINI PROJECT REPORT

1	**Name of college and place**	:	**A.J Institute of Engineering &Technology, Mangaluru**
2	**Mini Project Title**	**:**	**Arduino Based Radar System for Object Detection and Distance Monitoring**
3	**Section**	**:**	
4	**Name of Team Members**		**1.Sahan**
5	**Contact Details**	1	**email: vpoojarysahan@gmail.com**
		2	
		3	
		4	
6	**Date of commencement of the mini project**	:	**26 / 05 / 2025**

Name and Signature of project coordinators

INDEX

Arduino Based Radar System for Object Detection and Distance Monitoring

S.No	Title	Page No
1.	Introduction	1
2.	The Fundamental Principles of Radar: Transmitting and Receiving Radio Waves	2
3.	Radar Systems: Key Hardware Components	3-4
4.	Radar Systems: Software Used	5-6
5.	Circuit Diagram	7
6.	Code to Program Arduino Uno (Arduino IDE)	8
7.	Python Code to Display Output Screen(IDLE)	9-10
8.	Output Screen	11
9.	Radar Systems: Applications	12
10.	References	13

INTRODUCTION

Heinrich Hertz's groundbreaking experiments in the late 19th century established the existence of radio waves, laying the groundwork for radar technology. His research demonstrated that radio waves could be reflected, paving the way for the development of radar systems.

During World War II, radar technology rapidly advanced as scientists and engineers developed sophisticated systems to detect enemy aircraft and ships. This period saw the introduction of pulse radar and Doppler radar, revolutionizing military operations.

Radar is an object detection system which uses radio waves to determine the range, altitude, direction, or speed of an objects. It can be used to detect aircraft, ships, spacecraft, guided missiles, motor vehicles, weather formations, and terrain.

The modern uses of radar are highly diverse, including air traffic control, radar astronomy, air-defence systems, anti-missile systems ;marine radar start locate landmarks and other ships; aircraft anti-collision systems; ocean surveillance systems, outer space surveillance and rendezvous systems; meteorological precipitation monitoring; altimetry and flight control systems; guided missile target locating systems; and ground-penetrating radar for geological observations.

High tech radar systems are associated with digital signal processing and are capable of extracting useful information from very high noise levels.

The Fundamental Principles of Radar: Transmitting and Receiving Radio Waves

The fundamental principle behind working of any radar system is -

Transmitting Radio Waves:

Radar system consists of a transmitter, the main role of a transmitter is to emit pulses or radio waves continuously, these emitted waves travel outward at the speed of light. These pulses are carefully timed and modulated to carry specific information.

Reflecting Signals:

Whenever radio waves encounter objects or obstacles in their path, the waves get reflected back and the receiver receives the reflected signal. The time taken by the waves to return back after reflecting provides information about the distance to the object.

Processing Received Signals:

The received radar signals are decoded, analyzed and processed to extract all the data about the target’s location, speed, distance and other characteristic information. Generally, a microcontroller or a microprocessor is used to process or extract the received data signals. These information’s are then displayed to the operator through a screen.

Radar Systems: Key Hardware Components

- **Arduino Uno**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

- **Ultrasonic Sensor**

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.

Distance calculation ---Distance $L = 1/2 \times T \times C$

Where L is the distance, T is the time between the emission and reception, and C is the sonic speed.

(The value is multiplied by 1/2 because T is the time for go-and-return distance.) Features

Since ultrasonic waves can reflect off a glass or liquid surface and return to the sensor head, even transparent targets can be detected.

- **Servo Motor**

A servomotor is a rotary or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration in a mechanical system. It constitutes part of a servomechanism, and consists of a suitable motor coupled to a sensor for position feedback and a controller. Servo motors are like powerful little engines with a high torque-to-inertia ratio, meaning they can generate a lot of twisting force relative to their size and weight. This makes them perfect for tasks that need quick acceleration, deceleration, and precise movements. They come in various sizes and power levels to suit different jobs and can be powered by either AC or DC voltage. What's good is that we can control their speed and force using special signals called pulse width modulation (PWM). Some servo motors have extra features like built-in controllers and communication interfaces, which makes them adaptable and easy to integrate into complex systems.

Radar Systems: Software Used

- **Arduino IDE**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - connects to the Arduino boards to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension (.ino). The Arduino environment can be extended through the use of libraries. Just like most programming platforms, libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from **Sketch > Import Library**.

Libraries Used-

- **< Servo.h >**

This library allows an Arduino board to control RC (hobby) servo motors. Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. Continuous rotation servos allow the rotation of the shaft to be set to various speeds.

- **Python (IDLE)**

Python is a programming language that is interpreted, object-oriented, and considered to be high-level. Python is one of the easiest yet most useful programming languages and is widely used in the software industry. People use Python for Competitive Programming, Web Development, and creating software. Due to its easiest syntax, it is recommended for beginners who are new to the software engineering field. Its demand is growing at a very rapid pace due to its vast use cases in Modern Technological fields like Data Science, Machine learning, and Automation Tasks.

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default. IDLE can be used to execute a single statement just like Python Shell and also to create, modify, and execute Python scripts. IDLE provides a fully-featured text editor to create Python script that includes features like syntax highlighting, autocompletion, and smart indent. It also has a debugger with stepping and breakpoints features.

Libraries Used-

- **Matplotlib.pyplot**

Matplotlib.pyplot is a state-based interface to the Matplotlib library, which is a comprehensive library for creating static, animated, and interactive visualizations in Python. Pyplot provides a MATLAB-like interface, making it easy to create plots by calling functions that modify a figure. Each function in pyplot makes some change to a figure, such as creating a figure, creating a plotting area in a figure, plotting some lines in a plotting area,etc

- **NumPy**

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transformation. NumPy stands for Numerical Python. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behaviour is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

- **Time**

The time module in Python provides various functions to handle time-related tasks such as reading the current time, formatting time, and pausing execution. This module is always available and offers a range of functionalities for different platforms

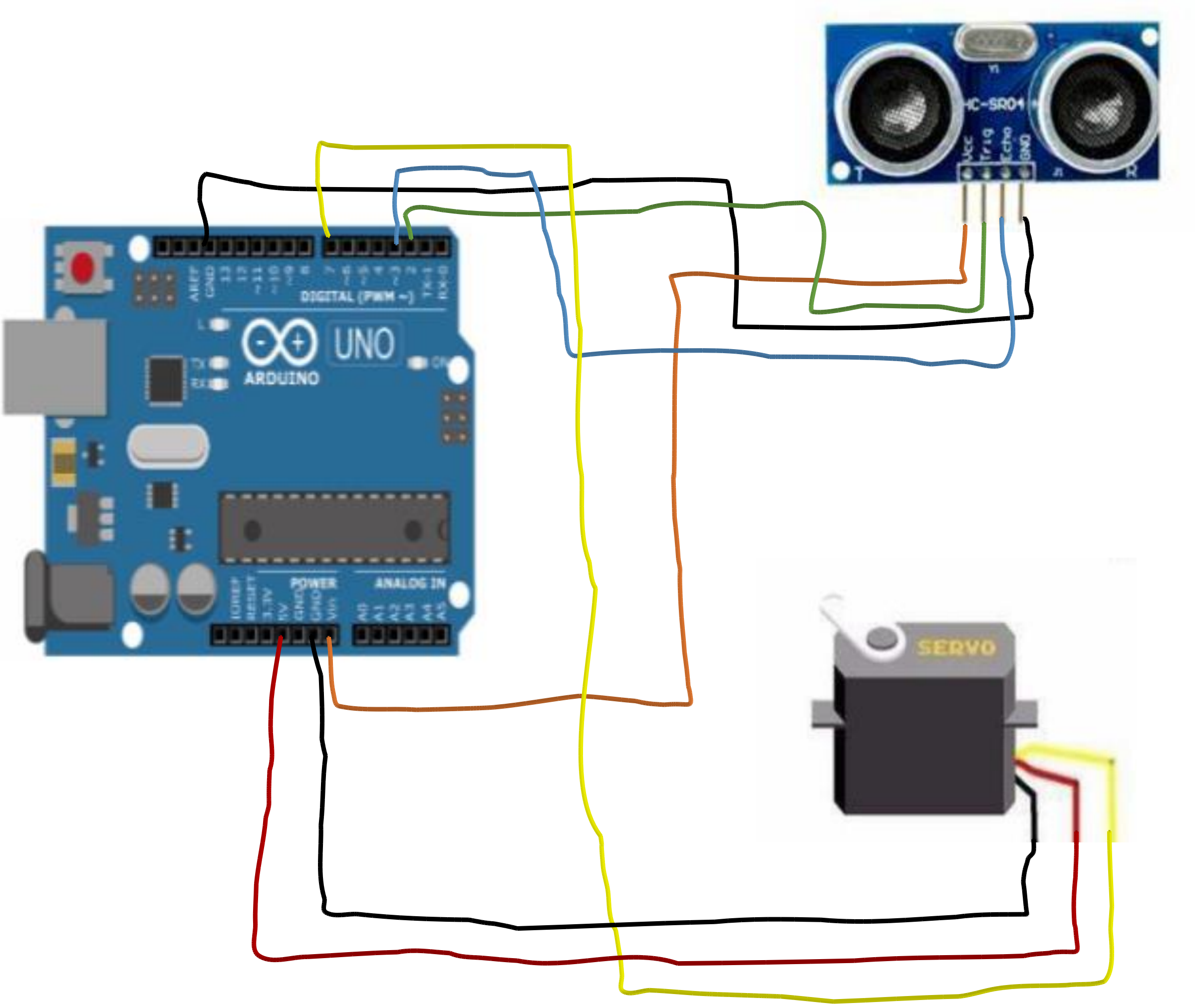
- **pySerial**

This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “serial” automatically selects the appropriate backend.

- **Keyboard**

Python provides a library named keyboard which is used to get full control of the keyboard. It's a small Python library which can hook global events, register hotkeys, simulate key presses and much more. It helps to enter keys, record the keyboard activities and block the keys until a specified key is entered and simulate the keys

Circuit Diagram



Code to Program Arduino Uno (Arduino IDE)

```
#include<Servo.h>
Servo micro_servo;
int pos=0;
const int servo_pin=7;
const int trig_pin=2;
const int echo_pin=3;
float duration , distance;
void setup() {
  pinMode(servo_pin,OUTPUT);
  pinMode(trig_pin,OUTPUT);
  pinMode(echo_pin,INPUT);
  Serial.begin(9600);
  micro_servo.attach(servo_pin);
}

void loop() {
  for(pos = 0; pos<180; pos++)
  {
    micro_servo.write(pos);
    delay(50);
    read_ultrasonic_distance(pos);
  }
  for(pos = 180; pos>=0; pos--)
  {
    micro_servo.write(pos);
    delay(50);
    read_ultrasonic_distance(pos);
  }
  delay(200);
}

void read_ultrasonic_distance(int angle)
{
  digitalWrite(trig_pin, LOW);
  delayMicroseconds(2);
  digitalWrite(trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_pin, LOW);
  duration=pulseIn(echo_pin,HIGH);
  distance=(duration*0.033)/2;
  Serial.print(angle);
  Serial.print(",");
  Serial.println(distance);

}
```

Python Code to Display Output Screen(IDLE)

```
from matplotlib import pyplot as plt
import matplotlib
matplotlib.use('TkAgg')
import numpy as np
import time
import serial
import keyboard

ser=serial.Serial('COM3',baudrate=9600,bytesize=8,parity='N',stopbits=1,timeout=2)
fig=plt.figure(facecolor='k')
fig.canvas.toolbar.pack_forget()
fig.canvas.manager.set_window_title('Ultrasonic Radar')
mgn=plt.get_current_fig_manager()
mgn.window.state('zoomed')
ax=fig.add_subplot(1,1,1,polar=True,facecolor='#006b78')
ax.tick_params(axis='both',colors='w')

r_max=200
ax.set_ylim([0.0,r_max])
ax.set_xlim([0.0,np.pi])
ax.set_position([-0.05,-0.05,1.1,1.05])
ax.set_rticks(np.linspace(0.0,180,10))

angles=np.arange(0,181,1)
theta=angles*(np.pi/180)
pols,=ax.plot([],linestyle='',marker='o',markerfacecolor='r',markeredgecolor='w',markeredgewidth=1.0,markersize=3.0,alpha=0.5)
line1,=ax.plot([],color='w',linewidth=0.01)
#fig.canvas.draw()
dists=np.ones((len(angles),))
axbackground=fig.canvas.copy_from_bbox(ax.bbox)
```

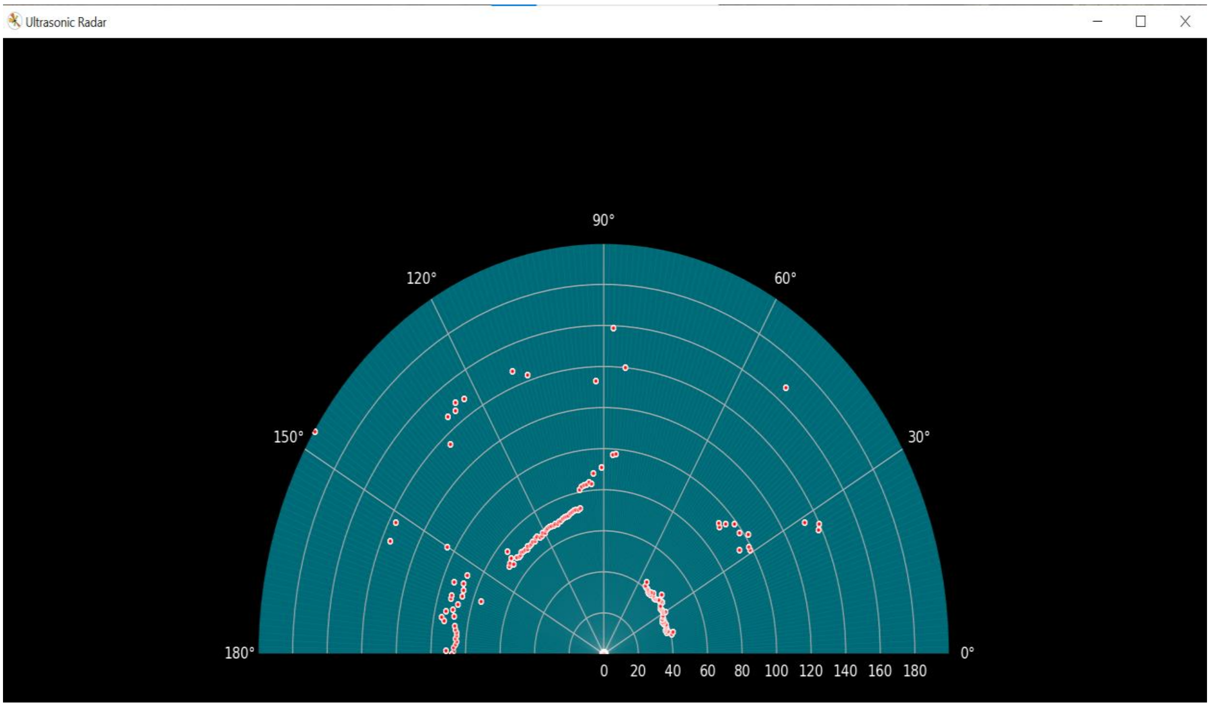
```
while True:
    try:
        data=ser.readline()
        decoded=data.decode()
        data=(decoded.replace('\r',''))
        vals=[float(ii)for ii in data.split(',')]

        if len(vals)<2:
            continue
        angle,distance=vals
        #print(angle,distance)
        dists[int(angle)]=distance
        pols.set_data(theta,dists)
        fig.canvas.restore_region(axbackground)
        ax.draw_artist(pols)
        line1.set_data(np.repeat((angle*(np.pi/180)),2),np.linspace(0.0,r_max,2))
        ax.draw_artist(line1)
        fig.canvas.blit(ax.bbox)
        fig.canvas.flush_events()

        if keyboard.is_pressed('q'):
            plt.close('all')
            print("User need to quit the application")
            break

    except KeyboardInterrupt:
        print('Keyboard Interrupt')
        break
exit()
```

Output Screen



Radar Systems: Applications

Radar technology has a wide range of applications across various fields. Here are some key uses:

1. Air Traffic Control

- Helps monitor and manage aircraft movement.
- Ensures safe distances between planes.
- Assists in landing and take-off, especially in poor visibility.

2. Weather Forecasting

- Detects precipitation like rain, snow, and hail.
- Tracks storm movements and intensity.
- Provides real-time data for meteorologists to issue warnings.

3. Military Surveillance

- Detects and tracks aircraft, ships, and missiles.
- Provides early warning of potential threats.
- Used for defence systems and battlefield monitoring.

4. Space Exploration

- Tracks satellites and spacecraft.
- Helps in landing and docking operations.
- Used for studying planetary surfaces and asteroids.

5. Automotive Safety

- Used in adaptive cruise control and collision avoidance systems.
- Helps detect obstacles and assist in autonomous driving.
- Enhances vehicle navigation in low visibility conditions.

6. Maritime Navigation

- Guides ships safely through fog and bad weather.
- Detects nearby vessels and obstacles.
- Used for coastal surveillance and port management.

7. Law Enforcement

- Speed detection for traffic monitoring.
- Helps in search and rescue operations.
- Used in border security and surveillance.

8. Remote Sensing & Geology

- Detects underground structures and archaeological sites.
- Used for earthquake monitoring and environmental studies.
- Helps in mapping terrain and tracking land changes.

Radar's ability to function in various weather conditions makes it an essential tool in many industries.

References

- https://www.academia.edu
- https://projecthub.arduino.cc
- https://www.researchgate.net
- https://www.github.com
- http://dspace.daffodilvarsity.edu.
- https://www.instructables.com
- https://srituhobby.com