

Project Report: Arduino-Based Ultrasonic Object Detection System

1. Abstract & Introduction

This project details the design and implementation of a low-cost, real-time object detection system using an **Arduino Uno** microcontroller. The system utilizes an **HC-SR04 ultrasonic sensor** mounted on a **micro servo motor** to scan a 180-degree area. Distance and angular data are collected by the Arduino and transmitted via serial communication to a host computer, where the **Processing IDE** software visualizes the data on a graphical user interface (GUI) resembling a traditional radar display. The system's primary objective is to demonstrate object detection and ranging in a defined, short-range area.

2. Hardware Requirements

The main components required for the project are:

- **Microcontroller:** Arduino Uno board (ATmega328P).
 - **Sensor:** HC-SR04 Ultrasonic Sensor (for distance measurement).
 - **Actuator:** Micro Servo Motor (e.g., SG90, for rotating the sensor).
 - **Interfacing:** Jumper wires and a breadboard.
 - **Power:** USB cable for power and programming.
-

3. Working Principle

3.1. Distance Measurement (Ultrasonic Sensor)

The HC-SR04 sensor works on the principle of echo-location.

1. The Arduino sends a short **Trigger** pulse to the sensor.
2. The sensor emits an 8-cycle burst of ultrasonic sound (40 kHz).
3. The sound wave travels until it hits an object and reflects back.
4. The sensor's **Echo** pin goes HIGH upon transmission and goes LOW upon reception of the reflected wave.
5. The Arduino measures the **pulse duration (\$T\$)** (time the Echo pin was HIGH).
6. The distance (**\$D\$**) is calculated using the speed of sound (**\$C \approx 343 \text{ m/s}\$** at **\$20^\circ\text{C}\$**):

$$D = \frac{T \times C}{2}$$

The division by 2 is because the measured time (**\$T\$**) is for the round trip (out and back).

3.2. Scanning Mechanism (Servo Motor)

- The ultrasonic sensor is physically mounted onto the shaft of the servo motor.
 - The Arduino is programmed to sweep the servo motor from an initial angle (e.g., 0°) to a final angle (e.g., 180°) and back.
 - At small, defined angular increments (e.g., every 1° or 2°), the Arduino stops the motor momentarily to take a distance measurement from the ultrasonic sensor.
 - The paired data (**Angle, Distance**) is then sent to the computer's **Serial Port**.
-

4. Software Implementation

The system requires two main software parts:

4.1. Arduino IDE (Microcontroller Code)

- The code manages the servo motor's sweeping motion and precisely controls the timing for the ultrasonic sensor's *Trig* and *Echo* pins to calculate the distance.
- It packages the data as a string (e.g., "Angle,Distance.") and sends it to the computer via the **Serial Monitor** at a specific baud rate (e.g., 9600).

4.2. Processing IDE (Graphical User Interface - GUI)

- The Processing software reads the serial data sent by the Arduino.
 - It parses the received string to extract the **Angle** and **Distance** values.
 - It then draws a **visual radar sweep** on the computer screen.
 - Objects are plotted on the GUI as points or arcs at their corresponding angle and scaled distance from the center, creating the characteristic "Smart Radar" visualization.
-

5. Advantages and Applications

5.1. Advantages

- **Low Cost:** Uses readily available and inexpensive components.
- **Simplicity:** Easy to assemble and program, making it an excellent beginner project.
- **Real-time Output:** Provides instantaneous visual feedback on object location.

5.2. Applications

- Small-scale obstacle avoidance systems (e.g., in mini-robotics).
 - Indoor mapping and object positioning demonstrations.
 - Educational tool for demonstrating ranging technology and microcontroller interfacing.
-

6. Conclusion

The Arduino Uno based ultrasonic object detection system successfully functions as a "Smart Radar" prototype. It integrates a microcontroller, sensing hardware, and a graphical software interface to provide a reliable, low-power solution for determining the range and angle of objects in its vicinity.