**Laboratory Report**

**On**

# Smart Parking Assistant

***( Open Ended - 2)***

**Submitted by**

**Adyasha Mohanty (2330007)**

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

**B. Tech Programme in Electronics and Telecommunication Engineering**

**School of Electronics Engineering**
**Kalinga Institute of Industrial Technology, Deemed to be University**
**Bhubaneswar, India**

**19th March 2025**

## OBJECTIVE/AIM OF THE EXPERIMENT

The objective of this design experiment is to create a smart parking assistant using an Arduino Uno and an ultrasonic sensor. The aim is to accurately measure the distance to obstacles behind a vehicle and provide real-time feedback through a buzzer to assist the driver during parking. This project involves designing the hardware connections and implementing the necessary code to read sensor data and trigger the buzzer based on proximity.

## THEORY

The experiment utilizes an ultrasonic sensor (HC-SR04) to measure distance. The sensor emits a short burst of ultrasonic sound waves. These waves bounce off any object in their path. The sensor then measures the time it takes for the waves to return. This time is used to calculate the distance to the object. The formula to calculate the distance is:

$$\text{Distance} = \frac{(Time \times Speed\ of\ Sound)}{2}$$

Where:

- Time is the duration the sound wave takes to travel to the object and back.
- Speed of Sound in air is approximately 340 m/s or 0.034 cm/μs.

The Arduino processes this distance information and triggers a buzzer when the object is within a specified range (e.g., 20 cm) to alert the driver.

## REQUIREMENTS

Hardware

- Arduino Uno
- Ultrasonic Sensor (HC-SR04)
- Buzzer
- Breadboard
- Jumper Wires
- LED (optional)

Software

- Arduino IDE

## CIRCUIT DIAGRAM

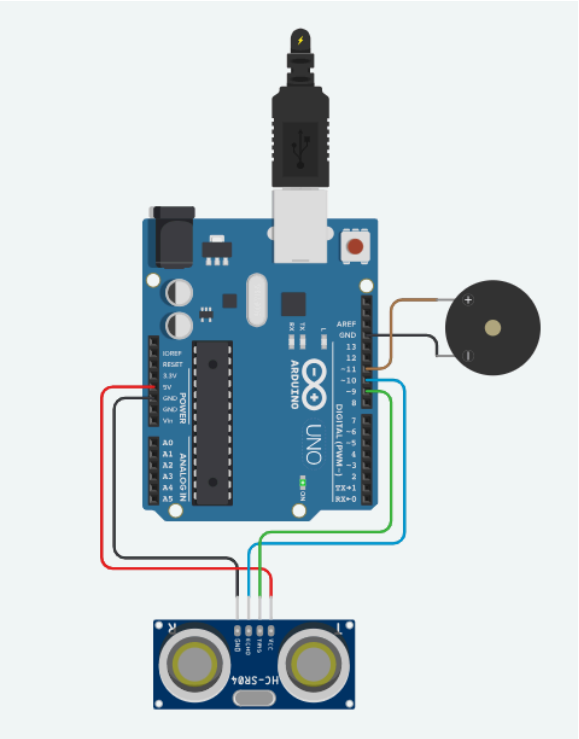


Fig. 1. Circuit diagram using Tinkercad

The circuit diagram illustrates the connections between the Arduino Uno, the HC-SR04 ultrasonic sensor, and the buzzer. The Arduino Uno serves as the microcontroller that processes the sensor data and controls the buzzer.
The HC-SR04 ultrasonic sensor has four pins: VCC, GND, TRIG, and ECHO.

- The VCC pin is connected to the 5V pin on the Arduino to provide power to the sensor.
- The GND pin is connected to the GND pin on the Arduino, providing a common ground.
- The TRIG pin, which triggers the ultrasonic pulse emission, is connected to digital pin 9 on the Arduino.
- The ECHO pin, which receives the reflected pulse, is connected to digital pin 10 on the Arduino.

The buzzer has two pins: positive (anode) and negative (cathode).

- The positive pin is connected to digital pin 11 on the Arduino.
- The negative pin is connected to the GND pin on the Arduino, completing the circuit.

When the Arduino detects an object within a specified range (e.g., 20 cm), it sends a signal to digital pin 11, activating the buzzer to alert the driver.

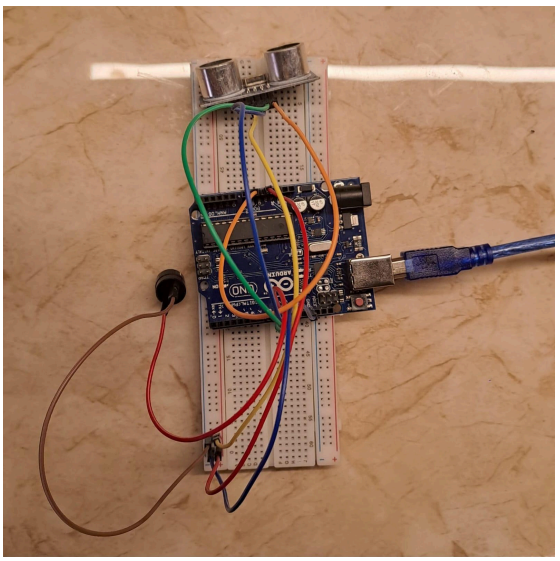

Fig. 2. Circuit connection

## CODE

```
// Define pins
const int trigPin = 9; // Trigger pin for ultrasonic sensor
const int echoPin = 10; // Echo pin for ultrasonic sensor
const int buzzer = 11;   // Buzzer pin

// Variables for distance calculation
long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT); // Set trigPin as output
  pinMode(echoPin, INPUT);  // Set echoPin as input
  pinMode(buzzer, OUTPUT);   // Set buzzer as output

  Serial.begin(9600); // Start serial communication at 9600 baud rate
}

void loop() {
  // Clear the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Set the trigPin HIGH for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  // Read echoPin and calculate distance
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2; // Convert to cm

  // Activate buzzer based on distance
  if (distance < 20) {
    digitalWrite(buzzer, HIGH); // Turn on buzzer if too close
    Serial.println("Warning: Object too close!");
  } else {
    digitalWrite(buzzer, LOW); // Turn off buzzer if safe
    Serial.println("Safe distance.");
  }

  // Print distance to Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(500); // Wait before next measurement
}
```

## DESIGN CALCULATIONS

The speed of sound in air is approximately 340 m/s, which is 0.034 cm/µs.
The Arduino code calculates the distance using the formula:
distance = duration * 0.034 / 2;
If the distance threshold to activate the buzzer is set to 20 cm, the condition in the code is:

```
if (distance < 20) {
digitalWrite(buzzer, HIGH); // Turn on buzzer
}
```

## OBSERVATIONS / RESULTS

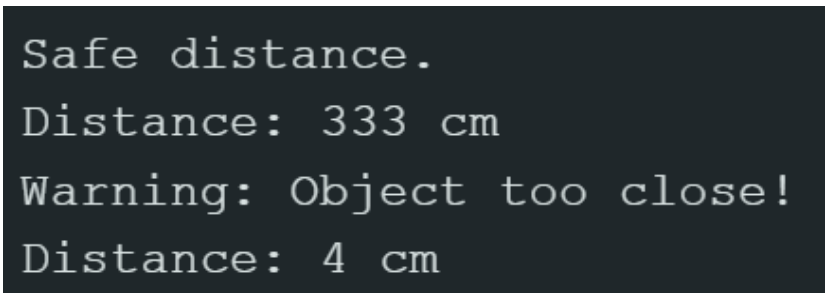


Fig. 3. Result in serial monitor of Arduino IDE

## DISCUSSION OF RESULTS

The experimental results show that the buzzer activates when the measured distance is less than 20 cm. There may be slight variations in the measured distance due to the accuracy limitations of the ultrasonic sensor and environmental factors such as temperature and humidity. Overall, the system performs as expected, providing an audible alert when an object is close to the sensor.

## CONCLUSION:

A functional smart parking assistant was successfully developed using an Arduino Uno and an ultrasonic sensor. The system accurately measures distances and provides an audible warning via a buzzer when an object is within a specified range, which can assist drivers in parking safely.

## REFERENCES:


1. Bhatti, D., Singh, D., & Bhatti, T. S. (2016). Smart Car Parking System Using Internet of Things. In *2016 International Conference on Internet of Things and Applications (IOTA)* (pp. 266-270). IEEE. doi: 10.1109/IOTA.2016.7562729
2. Moussa, A., Hussein, M., & El-Bendary, N. (2017). Intelligent car parking management system based on wireless sensor networks and fuzzy logic. *International Journal of Advanced Computer Science and Applications, 8*(11), 257-264. doi: 10.14569/IJACSA.2017.081135
3. Boylestad, R. L., & Nashelsky, L. *Electronic Devices and Circuit Theory*. Pearson Education, 2009. (Basic electronics)