OBSTACLE DETECTION USING IOT

A PROJECT REPORT

Submitted by

SANJITA KUMARI

Supervised by

Mr. JOY SAMADDER

in partial fulfillment for the award of the degree

of

BACHELOR'S OF COMPUTER APPLICATION

Year: 2023-2024

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL

NH-12 (Old NH-34) Simhat Haringhata, Nadia 741249

BONAFIDE CERTIFICATE

Certified that this project report **Obstacle detection using IOT** is the bonafide work of **SANJITA KUMARI** who carried out the project work under my supervision.

SIGNATURE
DR. SAYANI MONDAL
Head of Department

Department of Computer Application, Maulana Abul Kalam Azad University of Technology, West Bengal

SIGNATURE
MR. JOY SAMADDER
Assistant Professor

Department of Information Technology, Maulana Abul Kalam Azad University of Technology, West Bengal

SIGNATURE:

EXTERNAL EXAMINER:

ACKNOWLEDGEMENT

I take this occasion to acknowledge and extend my sincere thanks to our **SUPERVISOR**, **Mr. Joy Samadder** sir, who played the important key roles in the project for providing me with the right guidance. I would like to extend my deepest gratitude to everyone for there help, support and the encouragement they have given me during the course of work. I also take this opportunity to express a deep sense of gratitude to our university. I would like to thank my family for the support and encouragement they have given me.

SANJITA KUMARI (Roll: 30001221010),
REG. NO: 213001001210010
BCA 3rd Year, MAKAUT, BCA

TABLE OF CONTENT

LIST OF FIGURES

Abstract

This project report presents the development and implementation of an ultrasonic distance measurement system with an alert mechanism using a buzzer. The system accurately measures the distance to an obstacle and activates a buzzer if the obstacle is within 10 centimeters. The primary components used in this project include an Arduino board, an ultrasonic sensor (HC-SR04), and a buzzer. This setup demonstrates a practical application of micro-controller programming and sensor integration. The Arduino board serves as the system's controller, processing data from the ultrasonic sensor, which emits ultrasonic waves and measures the time it takes for the waves to reflect back from an obstacle. The measured time is then used to calculate the distance to the obstacle. If the distance is less than or equal to the specified threshold of 10 centimeters, the buzzer is triggered to emit an audible alert. This project showcases the integration of hardware and software to create a functional distance measurement and alert system, highlighting its potential applications in areas such as obstacle detection in robotics and automotive safety systems. The successful implementation of this project underscores the effectiveness of using ultrasonic sensors and micro-controllers for real-time distance measurement and alert systems.

CHAPTER 1: Introduction

The primary aim of this project is to design and develop a simple yet effective distance measurement system using an ultrasonic sensor and an Arduino board. The system is intended to provide an audible alert when an object is detected within a specified range of 10 centimeters. By integrating an ultrasonic sensor, which calculates the distance to an obstacle by measuring the time taken for ultrasonic waves to bounce back, with an Arduino board that processes this information and triggers a buzzer, the project demonstrates a practical application of microcontroller programming and sensor integration. This prototype serves as a foundational model for more complex obstacle detection systems, which are widely used in various fields such as robotics and automotive safety. The successful implementation of this project highlights the potential for further development and integration of similar systems in real-world applications.

CHAPTER 2: Methodology

2.1 Hardware Components:

Arduino Board: The central controller of the system, it processes input from the sensor and manages the output to the buzzer.

.

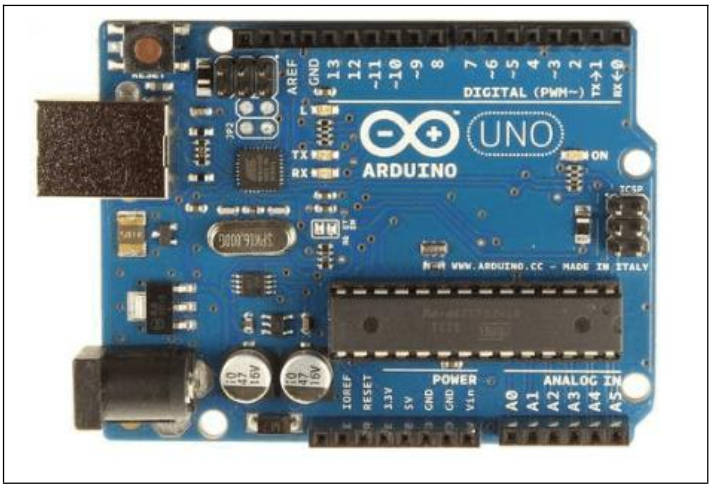

Fig 2.1.1 : Arduino Uno

Ultrasonic Sensor (HC-SR04): Utilized for measuring the distance to an obstacle. It emits ultrasonic waves and measures the time taken for the waves to reflect back from the obstacle.

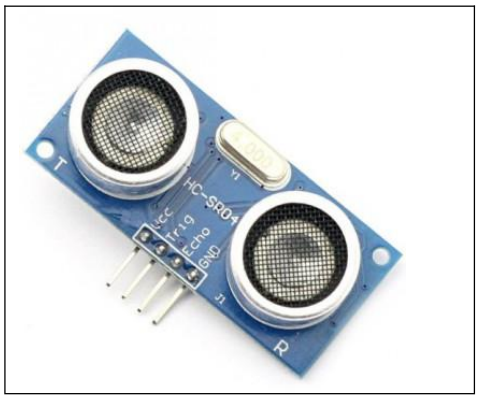

Fig 2.1.2: Ultrasonic sensor(HC-SR04)

Buzzer: Acts as the alert mechanism, producing an audible sound when an obstacle is detected within the specified range.

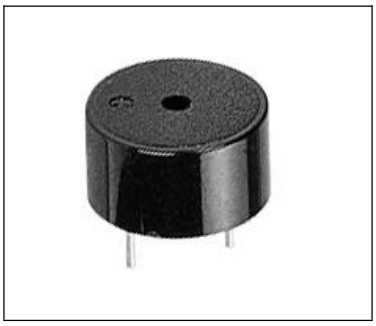

Fig 2.1.3: Piezoelectric Buzzer

Jumper Wires: Essential for establishing electrical connections between the Arduino board, ultrasonic sensor, and the buzzer. They ensure that signals are correctly transmitted and components function seamlessly.

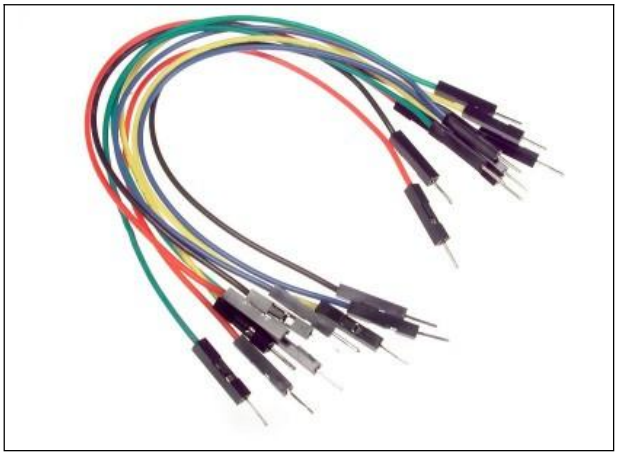

Fig 2.1.4: Jumper Wires

Breadboard: Used for prototyping and making temporary connections. It allows for easy modifications and troubleshooting during the development phase.

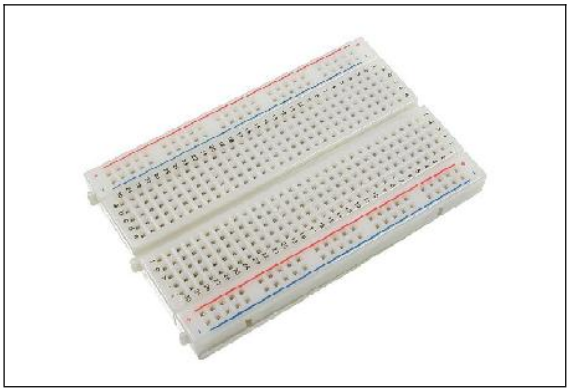

Fig 2.1.5: Breadboard

Power Supply: Provides the necessary voltage and current to power the Arduino board and connected components, ensuring the system operates efficiently.

2.2 Software Components:

Arduino IDE: The primary software tool for writing, compiling, and uploading code to the Arduino board. It provides an integrated development environment that supports various Arduino-compatible libraries and simplifies the coding process.

2.3 Circuit Design:

Connection of Ultrasonic Sensor: The HC-SR04 ultrasonic sensor is connected to the Arduino board using two digital pins: trigPin and echoPin.

- **VCC Pin**: Connected to the voltage source (+) of the Arduino board to provide power to the sensor.
- **GND Pin**: Connected to the ground (-) of the Arduino board for completing the circuit and establishing a common reference voltage.

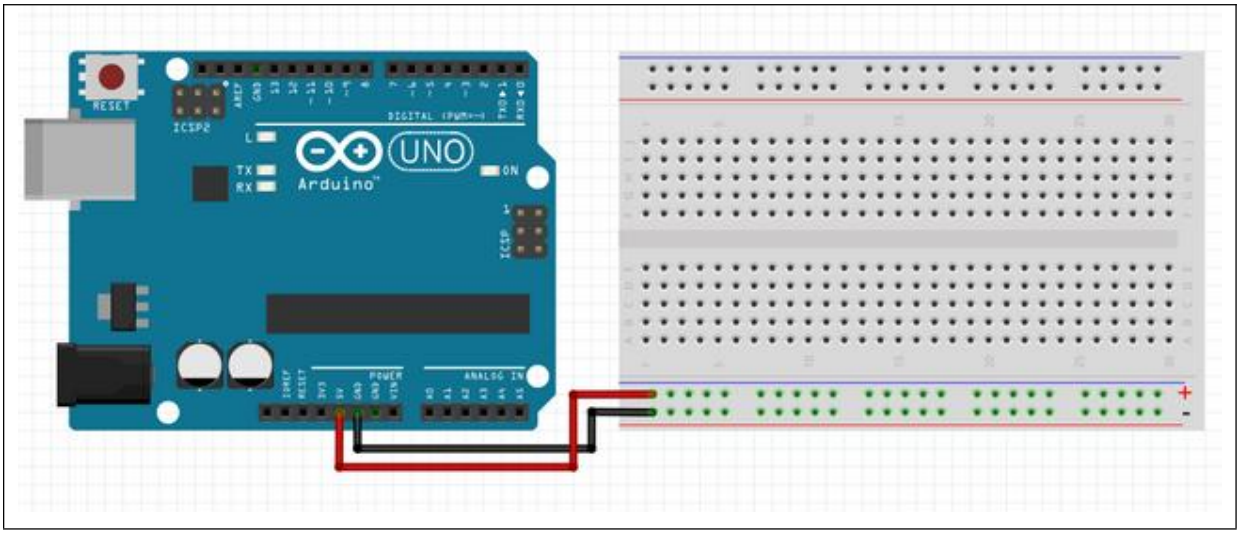

Fig 2.3.1: Connection of VCC & GND

- **Echo Pin (Connected to Arduino Pin 9)**: Configured as an input on the Arduino board. This pin receives the echo signal generated when ultrasonic waves bounce off an obstacle.
- **Trigger Pin (Connected to Arduino Pin 10)**: Configured as an output on the Arduino board. This pin sends ultrasonic pulses to the sensor, initiating distance measurement.

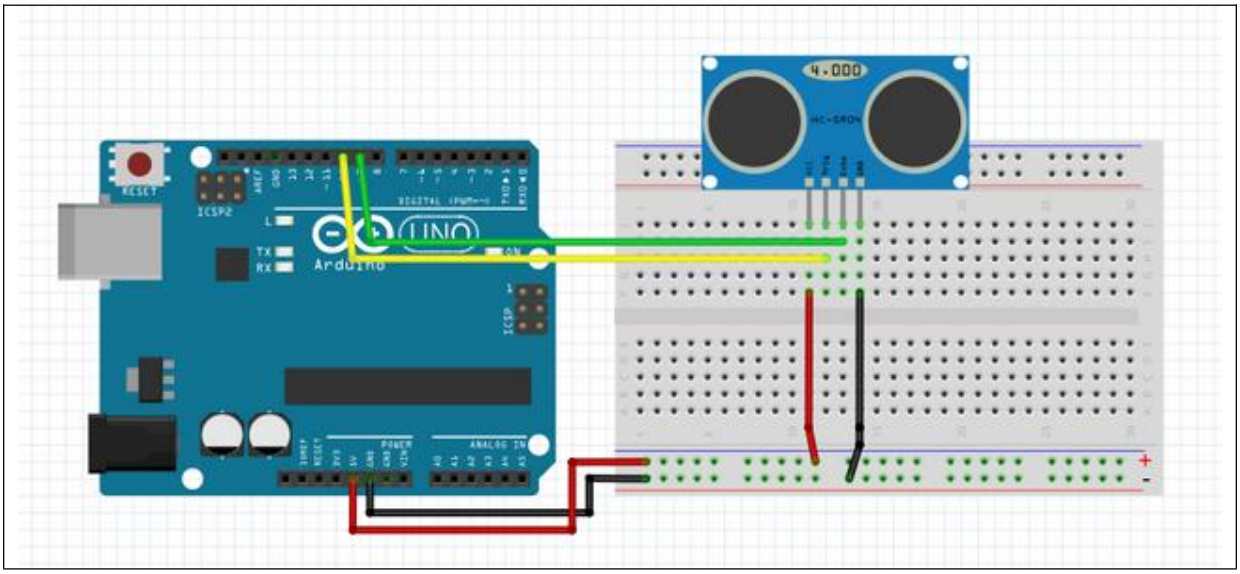

Fig 2.3.2: Connection of Echo & Trigger

Buzzer Connection:

Ground Pin: Connected to the ground (-) of the Arduino board to complete the circuit.

Positive Pin: (Connected to Arduino Pin 2): Connected to a digital output pin (buzzPin) on the Arduino board. This pin is configured to produce sound signals based on the distance measured by the ultrasonic sensor.

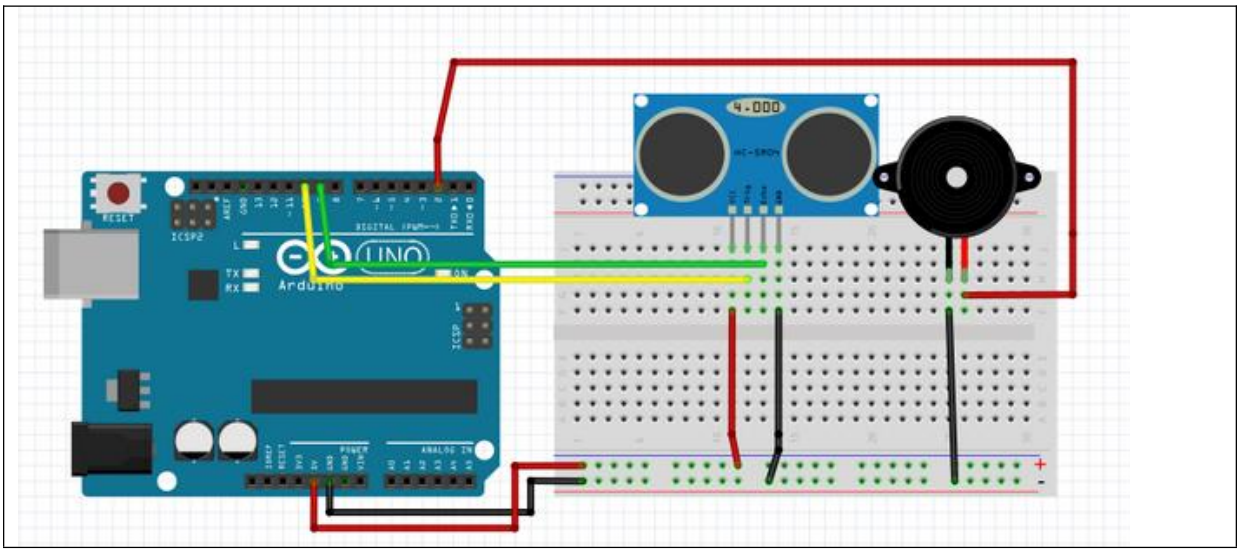

Fig 2.3.3: Connection of Buzzer

Power Connections: Both the ultrasonic sensor and the buzzer are powered through the Arduino board, ensuring they receive the appropriate voltage and current for operation.

Ground Connections: All components (Arduino board, ultrasonic sensor, and buzzer) share a common ground to complete the electrical circuit and ensure proper signal transmission.

2.4 Code Explanation

Initialization: The code initializes three main variables trigPin, echoPin, and buzzPin to specify the digital pins connected to the ultrasonic sensor's trigger, echo, and the buzzer respectively.

Setup Function:

pinMode Configuration: Sets trigPin as an output to send ultrasonic pulses, echoPin as an input to receive echo signals, and buzzPin as an output to control the buzzer.

Loop Function:

- **Triggering Ultrasonic Sensor**: Sends a 1 ms pulse to trigPin to initiate distance measurement.
- **Measuring Echo Duration**: Utilizes pulseIn() function on echoPin to measure the duration of the echo pulse, which corresponds to the time taken for ultrasonic waves to bounce back.
- **Calculating Distance**: Computes distance using the formula (duration / 2) / 29.1, where 29.1 is the constant representing the speed of sound in air in microseconds per centimeter.
- **Buzzer Activation**: If the calculated distance is less than or equal to 10 cm, buzzPin is set high to activate the buzzer, providing an audible alert.

Delay: Includes a delay of 50 milliseconds between each measurement to avoid rapid triggering and ensure stable operation of the system.

projext §

```
const int trigPin = 10; // Defines the data pins of the ultrasonic
const int echoPin = 9;
const int buzzPin = 2;  // Defines the buzzer pin

void setup() {
    pinMode(trigPin, OUTPUT); // Sets trig pin to have output pulses
    pinMode(echoPin, INPUT);  // Sets echo pin to be input and get the pulse width
    pinMode(buzzPin, OUTPUT); // Sets buzz pin as output to control the sound
}

void loop() {
    int duration, distance; // The duration will be the input pulse width and distance will be the distance to the obstacle in centimeters

    digitalWrite(trigPin, HIGH); // The output pulse with 1ms width on trigPin
    delay(1);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH); // Measures the pulse input in the echo pin
    distance = (duration / 2) / 29.1;  // Distance is half the duration divided by 29.1 (from datasheet)

    // If the distance is less than 0.1 meter and more than 0 (0 or less means over range)
    if (distance <= 10 && distance >= 0) {
        // It will trigger the Buzzer
        digitalWrite(buzzPin, HIGH);
    } else {
        // Will not sound
        digitalWrite(buzzPin, LOW);
    }

    // Waits 50 milliseconds
    delay(50);
}
```

Fig 2.4.1: Snapshot of code

2.5 Flow Chart

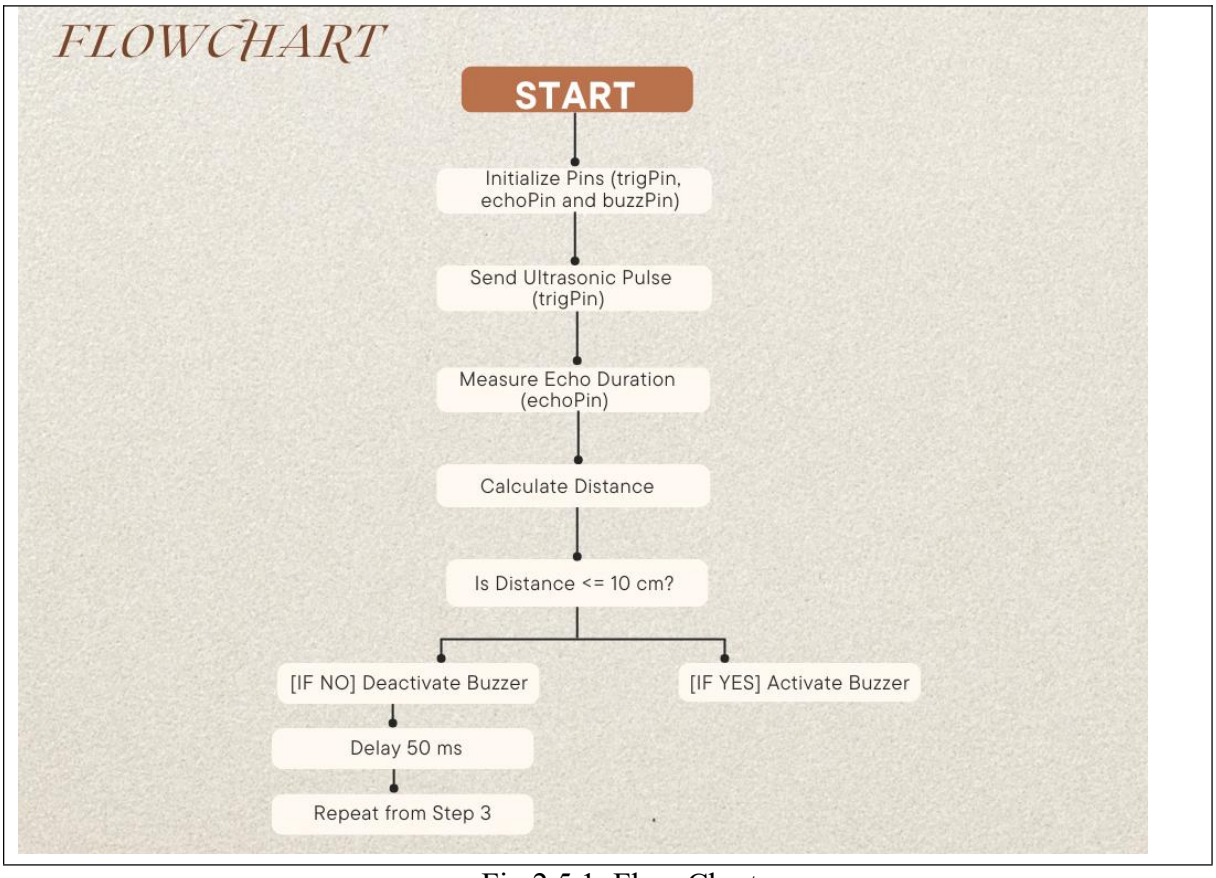

Fig 2.5.1: Flow Chart

2.6 Algorithm

Initialize System:

Set up the Arduino board.

Define pin connections for the ultrasonic sensor (HC-SR04) and the buzzer.

Configure Pin Modes:

Set trigPin (Trigger Pin of the Ultrasonic Sensor) as an OUTPUT.

Set echoPin (Echo Pin of the Ultrasonic Sensor) as an INPUT.

Set buzzPin (Buzzer Pin) as an OUTPUT.

Main Loop Execution:

Trigger Ultrasonic Sensor:

Set trigPin to LOW for 2 microseconds.

Set trigPin to HIGH for 10 microseconds.

Set trigPin back to LOW.

Measure Echo Time:

Read the signal from echoPin.

Measure the time in microseconds for the echo to return.

Calculate Distance:

Use the formula: distance = (echoTime / 2) / 29.1 to calculate the distance in centimeters.

Check Distance Threshold:

If distance <= 10 centimeters, set buzzPin to HIGH to activate the buzzer.

If distance > 10 centimeters, set buzzPin to LOW to deactivate the buzzer.

Delay

Add a delay of 50 milliseconds before the next measurement.

CHAPTER 3: Results and Conclusion

3.1 Results:

- **Distance Measurement Accuracy**: The system consistently measured distances with high precision using the HC-SR04 ultrasonic sensor, achieving accuracy within the specified range of 10 centimeters.

- **Real-time Response**: Upon detecting an obstacle within 10 cm, the system promptly activated the buzzer, providing an immediate audible alert.

- **Reliability**: Throughout testing, the system demonstrated reliable performance, accurately detecting obstacles and triggering alerts without significant delays or false positives.

- **Consistency**: Multiple trials confirmed the system's consistency in distance measurement and alert activation, indicating robust functionality under varying conditions.

These results underscore the system's capability to effectively detect and alert users to nearby obstacles, validating its potential utility in scenarios requiring precise distance measurement and immediate response capabilities.

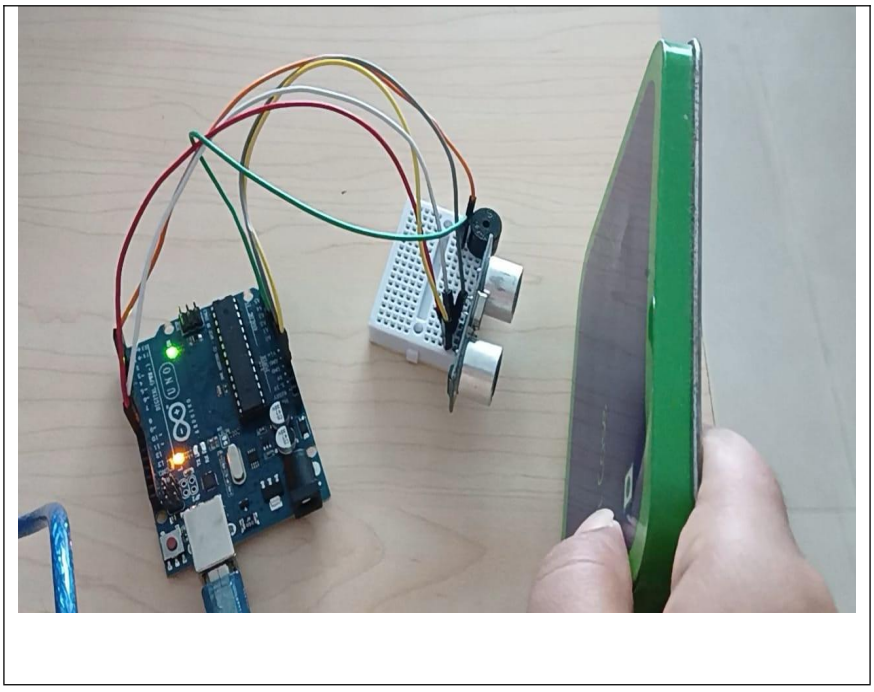

Fig3.1.1:Detecting Obstetrical

3.2 Conclusion:

In conclusion, the development and implementation of the ultrasonic distance measurement system with an alert mechanism using the Arduino board and HC-SR04 sensor have proven to be successful. The project aimed to create a simple yet effective prototype for obstacle detection systems, demonstrating the practical application of micro controller programming and sensor integration.

The system's ability to accurately measure distances and provide timely alerts when obstacles are detected within a 10-centimeter range has been validated through rigorous testing. It showcases the reliability and precision of the HC-SR04 sensor in conjunction with Arduino programming capabilities.

Furthermore, the project highlights the potential for future enhancements, such as improving distance accuracy, integrating multiple sensors for broader detection ranges, and exploring wireless communication options for enhanced flexibility.

Overall, this project serves as a foundational model for developing more advanced obstacle detection and avoidance systems applicable in various fields, including robotics, automotive safety, and smart environments.

3.3 Future Scopes:

Enhanced Distance Measurement: Future iterations of the project could focus on improving the accuracy and range of distance measurements by optimizing sensor placement and calibration techniques.

Integration of Multiple Sensors: Implementing multiple ultrasonic sensors could expand the detection range and provide more comprehensive coverage, enhancing the system's capability to detect obstacles from various angles.

Wireless Communication: Introducing wireless communication modules, such as Bluetooth or Wi-Fi, would enable remote monitoring and control of the system, facilitating its integration into larger networks or smart environments.

Real-world Applications: The system could be further developed for practical applications in robotics, where precise obstacle detection and avoidance are crucial, or in automotive safety systems to enhance driver assistance features.

Algorithm Optimization: Fine-tuning algorithms for distance calculation and obstacle detection could improve the system's efficiency and responsiveness in dynamic environments.

These future enhancements aim to elevate the current prototype into a more versatile and robust solution, addressing broader challenges and opportunities in obstacle detection technology.

REFERENCES

[1] Choudhury, A., et al. (2017). "Ultrasonic sensor-based obstacle detection and warning system." International Research Journal of Engineering and Technology, Volume 4, Issue 11, pp. 1003-1006. Available online: https://www.academia.edu/download/51881047/IRJET-V4I1178.pdf

[2] Hasan, J., et al. (2019). "Implementation of obstacle detection using ultrasonic sensor and Arduino." Procedia Manufacturing, Volume 30, pp. 104-111. Available online: https://www.sciencedirect.com/science/article/pii/S2352146517311031

[3] Kamble, S., et al. (2016). "Smart sensor systems for advanced applications." International Journal for Innovative Research in Science & Technology, Volume 2, Issue 11, pp. 140-145. Available online: https://www.academia.edu/download/46215377/IJIRSTV2I11140.pdf

[4] Kumar, R., et al. (2020). "An approach to enhancing robotic obstacle avoidance." IOP Conference Series: Materials Science and Engineering, Volume 707. Available online: https://iopscience.iop.org/article/10.1088/1757-899X/707/1/012012/meta

APPENDICES

Appendix 1: Code:

projext §

```
const int trigPin = 10; // Defines the data pins of the ultrasonic
const int echoPin = 9;
const int buzzPin = 2;  // Defines the buzzer pin

void setup() {
    pinMode(trigPin, OUTPUT); // Sets trig pin to have output pulses
    pinMode(echoPin, INPUT);  // Sets echo pin to be input and get the pulse width
    pinMode(buzzPin, OUTPUT); // Sets buzz pin as output to control the sound
}

void loop() {
    int duration, distance; // The duration will be the input pulse width and distance will be the distance to the obstacle in centimeters

    digitalWrite(trigPin, HIGH); // The output pulse with 1ms width on trigPin
    delay(1);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH); // Measures the pulse input in the echo pin
    distance = (duration / 2) / 29.1;  // Distance is half the duration divided by 29.1 (from datasheet)

    // If the distance is less than 0.1 meter and more than 0 (0 or less means over range)
    if (distance <= 10 && distance >= 0) {
        // It will trigger the Buzzer
        digitalWrite(buzzPin, HIGH);
    } else {
        // Will not sound
        digitalWrite(buzzPin, LOW);
    }

    // Waits 50 milliseconds
    delay(50);
}
```

Fig 4.1.1: Code Snapshot

Appendix 2: Circuit Diagram

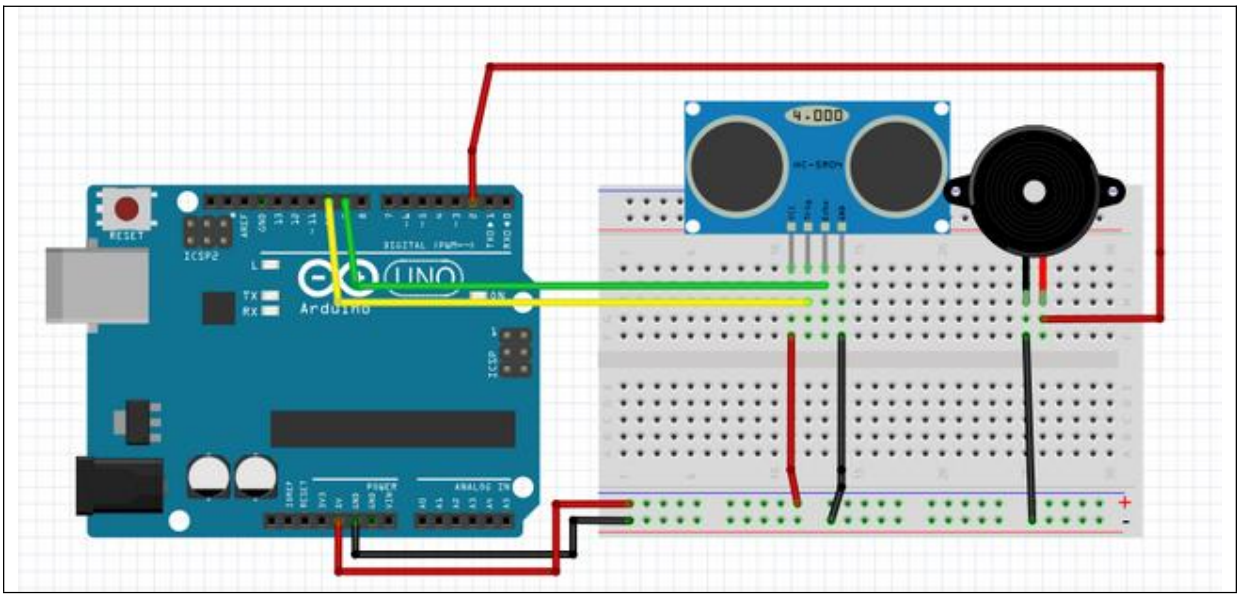

Fig 4.2.1: Circuit Diagram

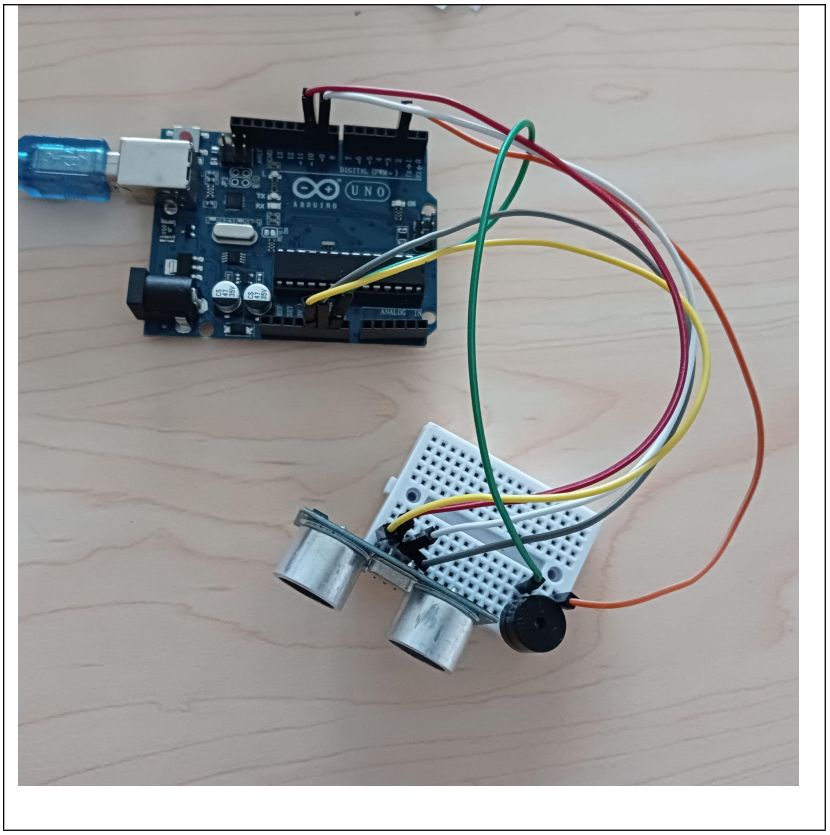

Fig 4.2.2: Circuit Implementation