

**Dr. D. Y. Patil Pratishthan’s College of Engineering Salokhenagar,**

**Kolhapur**

**2023-2024**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**(DATA SCIENCE)**



**A PROJECT REPORT ON**

# “OBJCT DETECTION AND ALARMING SYSTEM”

**Submitted By:**

Mr. Ashutosh Shailendra Jarag (24)

Mr. Aditya Kishor Khadake (30)

Mr. Aniket Santosh Mali (34)

Mr. Vinay Vijay Naik (43)

**Under the Guidance of**

**Mr. Rajesh D. Gade**

(Assistance Professor)

## SHIVAJI UNIVERSITY KOLHAPUR



**Dr. D. Y. Patil Pratishthan’s College of Engineering Salokenagar, Kolhapur**

**2023-2024**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**(DATA SCIENCE)**



# CERTIFICATE

Certified that the Project topic entitled “OBJECT DETECTION AND ALARMING SYSTEM” a bonafide work carried out by,

Mr. Ashutosh Shailendra Jarag

Mr. Aditya Kishor Khadake

Mr. Aniket Santosh Mali

Mr. Vinay Vijay Naik

in partial fulfilment for the award of Degree of Bachelor of Engineering in 6th semester of the **SHIVAJI UNIVERSITY, KOLHAPUR** during the year **2023-2024**. It is certified that all correction/suggestion indicated for Internal Assessment have been incorporated in the report deposited in the Department Library. The Project report has been approved as it satisfies the

Academic requirement in respect of Project work prescribed for **BACHELOR OF ENGINEERING DEGREE**.

Mr. Rajesh D. Gade Dr. Shrikant D. Bhopale Dr. S. D. MANE

(Project Guide) (Head of Department) (Principal)

**CONTENTS**

1. **INTRODUCTION**

1.1 Introduction……………………………………………………….

1. **LITURATURE SURVVEY**

2.1 Literature Survey………………………………………………….

1. **PROPOSED SYSTEM**

3.1 Proposed System………………………………………………….

## 4. SYSTEM ANALYSIS

4.1 Problem Statement………………………………………………..

4.2 System Requirements……………………………………………..

4.3 Software Requirements……………………………………………

4.4 Hardware Requirements…………………………………………..

## 5. SYSTEM DESIGN

5.1 System Implementation………………………………………….

5.2 Snapshots………………………………………………………...

**6. CONCLUSION**

## 7. REFERENCES

**ABSTRACT**

This project explores the integration of the HC-SR04 ultrasonic sensor with a Raspberry Pi to develop a collision detection system. The primary goal is to measure distances between objects and the sensor, employing Python programming for data processing and alarm triggering functionalities. The system's architecture involves real-time monitoring of distance readings from the sensor, comparing them against a predefined threshold, and activating a sound alarm when an object approaches dangerously close. By leveraging the versatility of the Raspberry Pi and the simplicity of the HC-SR04 sensor, this project aims to demonstrate an accessible and effective solution for collision detection in IoT environments.

Through this project, we investigate the practical implementation of a collision detection mechanism using readily available hardware components and open-source software tools. The Python programming language facilitates seamless communication between the sensor and the Raspberry Pi, enabling efficient data processing and decision-making. Additionally, the project emphasizes the importance of affordable and easily accessible technologies in developing practical IoT solutions for real-world applications. By providing a reliable means of detecting potential collisions, this system contributes to enhancing safety measures in various contexts, ranging from home automation to industrial settings.

### 1.1 Introduction

In this project, we focus on developing a collision detection system using the HC-SR04 ultrasonic sensor and a Raspberry Pi microcomputer. The HC-SR04 sensor is renowned for its accuracy in measuring distances using ultrasonic waves, while the Raspberry Pi serves as a versatile platform capable of executing complex tasks and interfacing with various hardware components.

Our collision detection system aims to provide real-time monitoring of distances between objects and the sensor, with the objective of triggering an alarm when an object approaches dangerously close. The system's functionality relies on the interplay between the HC-SR04 sensor, which continuously measures distances, and the Raspberry Pi, which processes this data and activates the alarm when necessary.

Key components of the project include the HC-SR04 ultrasonic sensor, Raspberry Pi microcomputer, and the Python programming language, which is utilized for coding the system's functionalities. By leveraging these components, we aim to demonstrate the feasibility of using affordable hardware and open-source software for developing practical IoT solutions, particularly in the realm of collision detection.

### 2.1 Literature Survey

Prior research in collision detection systems has explored various methodologies and technologies to address safety concerns in dynamic environments. Studies have highlighted the effectiveness of ultrasonic sensors, such as the HC-SR04, in providing accurate distance measurements for collision avoidance applications. Additionally, research has demonstrated the versatility of Raspberry Pi as a platform for interfacing with sensors and executing collision detection algorithms.

Several projects have implemented similar collision detection systems using ultrasonic sensors and microcomputers. These projects typically employ Python as the programming language for interfacing with the hardware components and processing sensor data. Furthermore, literature in the field emphasizes the importance of real-time monitoring and timely response mechanisms to mitigate collision risks effectively. Overall, the existing body of research provides valuable insights into the design considerations, implementation challenges, and performance evaluations of collision detection systems using ultrasonic sensors and microcomputers. By building upon this prior work, our project aims to contribute to the advancement of collision avoidance technology, particularly in the context of IoT applications.

### 3.1 Proposed Work

Our project aims to develop a collision detection system utilizing the HC-SR04 ultrasonic sensor interfaced with a Raspberry Pi microcomputer. The system will be designed to monitor distances between objects and the sensor in real-time, with the objective of triggering an alarm when an object approaches dangerously close.

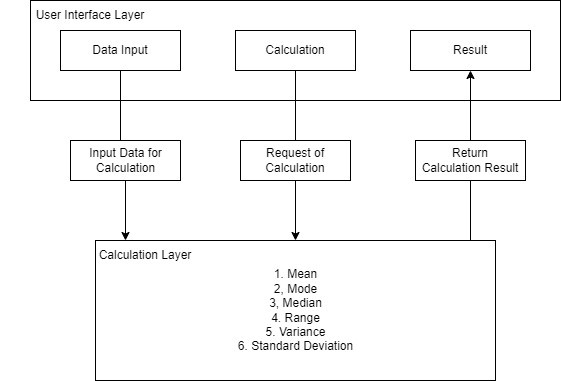
Hardware Setup: We will assemble the HC-SR04 ultrasonic sensor with the Raspberry Pi microcomputer according to the manufacturer's specifications. This setup will enable communication between the sensor and the Raspberry Pi, facilitating distance measurement capabilities.

Software Development: Using the Python programming language, we will develop the software components necessary for the collision detection system. This includes coding algorithms to interface with the HC-SR04 sensor, process distance measurements, and activate an alarm when predefined thresholds are exceeded.

Testing and Validation: We will conduct rigorous testing to ensure the functionality and reliability of the collision detection system. This will involve testing the system's response to various object distances and velocities to validate its effectiveness in detecting potential collisions.

By following these steps, we aim to develop a robust and efficient collision detection system that demonstrates the capabilities of IoT technologies in addressing safety challenges in real-world scenarios.

### 3.2 Architecture



**3.3 Modules:**

**Hardware Setup:**

This module involves setting up the Raspberry Pi and connecting the HC-SR04 ultrasonic sensor to it. It includes physical connections, ensuring power supply, and establishing communication protocols between the sensor and the Raspberry Pi.

**Distance Measurement:**

This module focuses on programming the Raspberry Pi to read distance measurements from the HC-SR04 sensor accurately. It includes configuring GPIO pins, sending trigger signals, receiving echo signals, and calculating distances based on the time delay.

**Collision Detection Algorithm:**

Here, you'll design the algorithm that determines whether an object is too close to trigger a collision warning. This may involve setting a threshold distance and comparing real-time distance measurements to this threshold.

**Sound Triggering:**

This module deals with programming the Raspberry Pi to produce sound alerts when a potential collision is detected. It involves controlling a buzzer or a speaker connected to the Raspberry Pi to emit a warning sound.

### 3.4 Algorithm

Step 1: Begin

Step 2: Import the required libraries

Step 3: Set GPIO mode to BCM

Step 4: Set the GPIO pins for LED, Trigger, Echo, and Buzzer

Step 5: Set the LED, Trigger, and Buzzer pins as output

Step 6: Set the echo pin as input

Step 7: Call the measure\_distance() function and print it

Step 8: Set threshold value

Step 9: if threshold\_value < distance

Turn on the LED and Buzzer

else

Turn off the LED and Buzzer

Step 10: Wait for 0.0001 seconds

Step 11: if user terminates program

Print "Program terminated by User"

else

Goto Step 7

Step 12: End

### 4.1 Problem Statement

Our project aims to address this by developing a cost-effective collision detection system using the HC-SR04 ultrasonic sensor and Raspberry Pi. This system will monitor object distances in real-time and trigger alarms to prevent potential collisions, offering a reliable solution for enhancing safety in various applications, ultimately striving for safer and more efficient.

### 4.2 System Requirements

**4.2.1 Software Requirements:**

1. The Raspbian Operating System
2. Python Interpreter
3. An Integrated Development Environment (IDE) like Thonny.

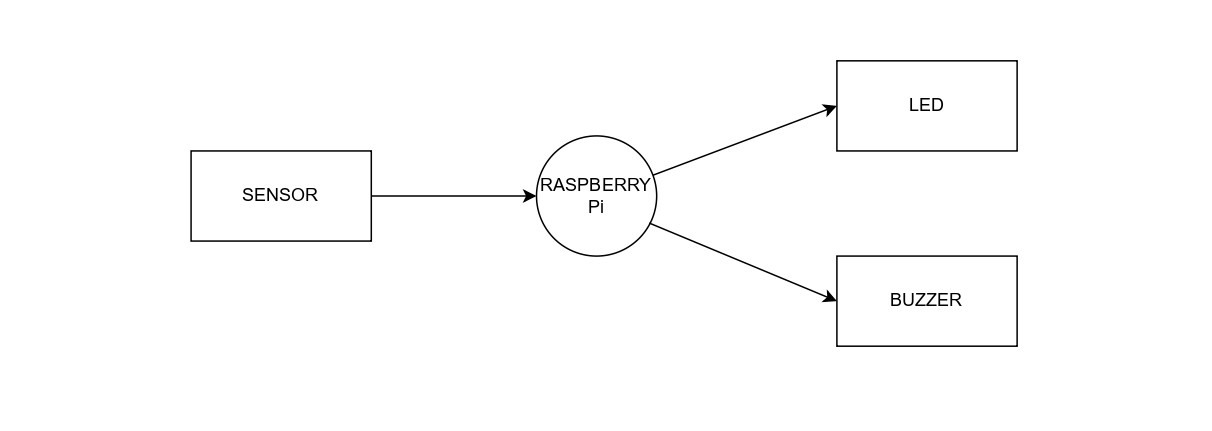
**4.2.2 Hardware Requirements:**

1. Raspberry Pi 4B
2. HC-SR04 Ultrasonic Sensor
3. Electronic Buzzer
4. Breadboard
5. Jumper Cables.

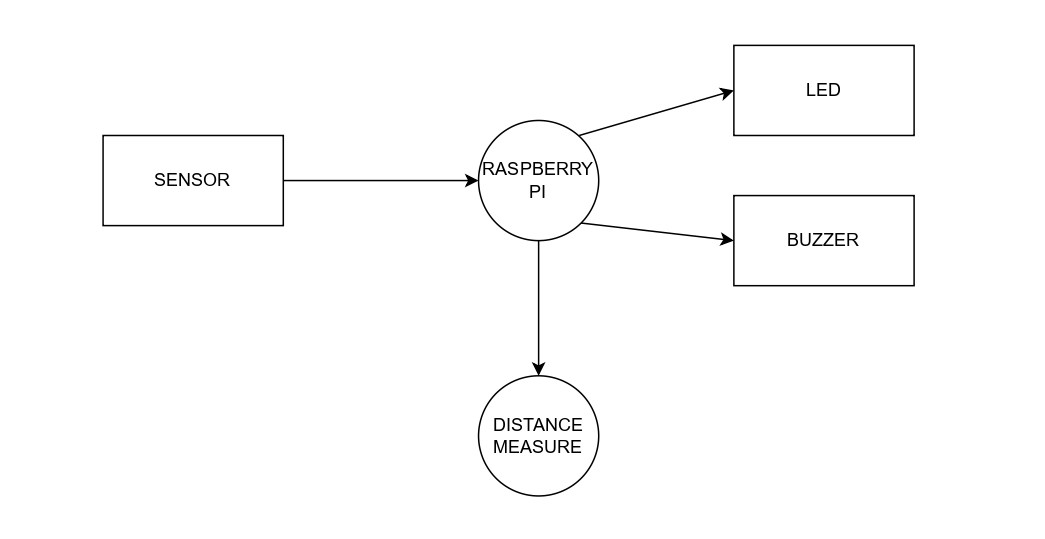
### 5.1 System Implementation

#### 5.1.1 DFD

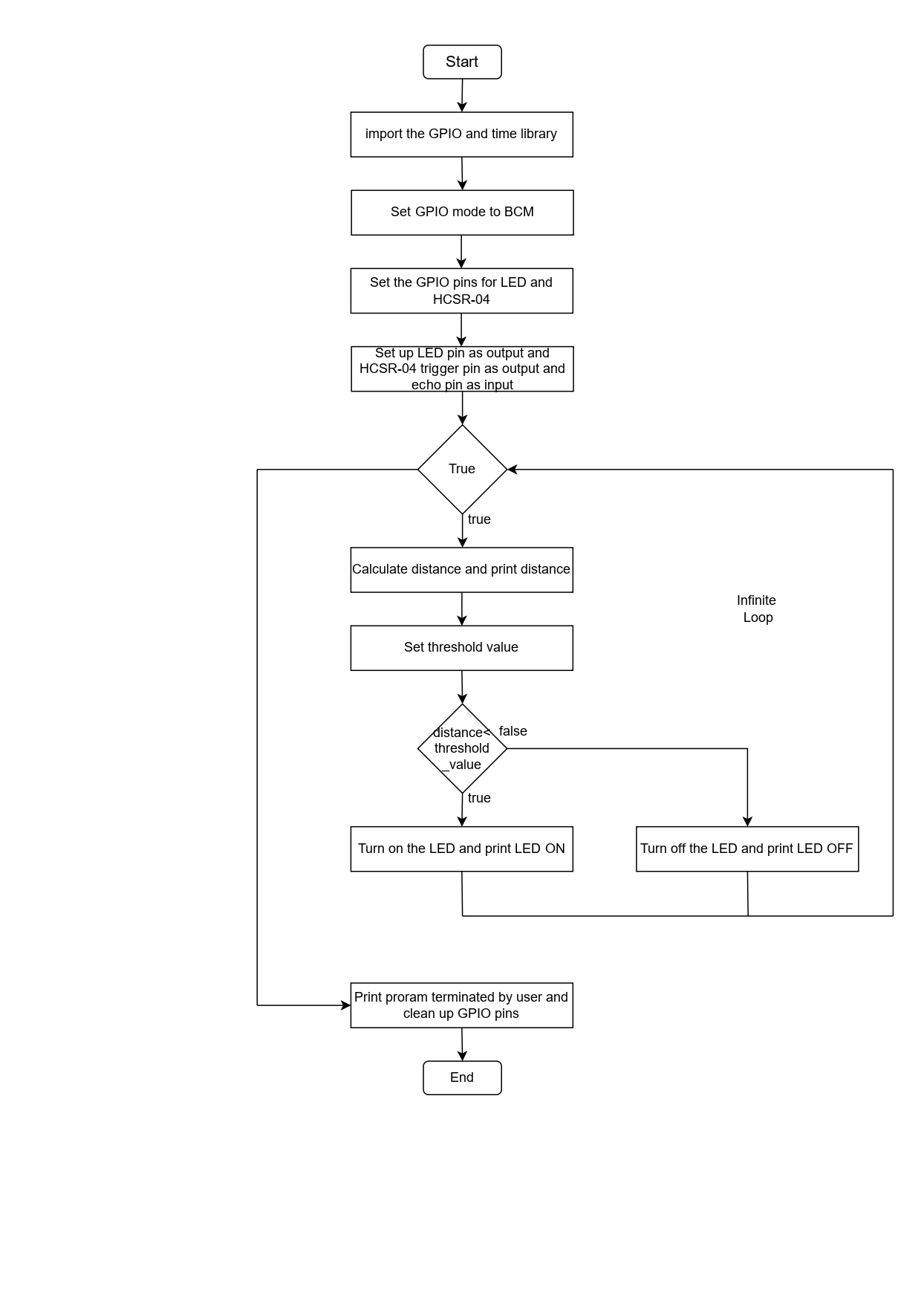
**Level 0:**



**Level 1:**



#### 5.1.2 Flowchart

****

### 6.1 Conclusion

In conclusion, our project has successfully developed a collision detection system utilizing the HC-SR04 ultrasonic sensor interfaced with a Raspberry Pi microcomputer. Through meticulous hardware setup, software development, and testing procedures, we have created a robust and efficient system capable of monitoring distances in real-time and triggering alarms when objects approach dangerously close.

This project underscores the potential of IoT technologies in addressing safety challenges in diverse environments. By leveraging affordable hardware components and open-source software, we have demonstrated the feasibility of implementing collision detection systems with practical applications. The success of our project highlights the importance of innovation and collaboration in advancing IoT solutions for real-world scenarios.

Moving forward, further enhancements and optimizations could be explored to improve the system's performance and expand its capabilities. Additionally, future research may focus on integrating additional sensors or implementing machine learning algorithms to enhance collision detection accuracy and adaptability. Overall, our project contributes to the ongoing efforts in leveraging IoT technologies for enhancing safety measures and mitigating risks in dynamic environments.

**7.1 References**

## BOOKS

* Raspberry Pi Cookbook: Software and Hardware Problems and Solutions" by Simon Monk
* Building Internet of Things with the Raspberry Pi" by Marco Schwartz and Olivier Englert
* Python Crash Course: A Hands-On, Project-Based Introduction to Programming" by Eric Matthes

## LINKS

* <https://projects.raspberrypi.org/en/projects/physical-computing/1>
* <https://www.w3schools.com/python/>
* <https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/2>