Visvesvaraya National Institute of Technology, Nagpur

Presented by:

Miss. Laxmi Shankar Kusagale

Roll Number**: 63**

Miss. Vidhisha Chandankar

Roll Number**: 53**

Miss. Arangi Venkata Lakshmi

Roll Number**: 75**

Miss. Mayuri Garge

Roll Number**: 62**

(2nd Year, 3rd semester)

Department:
Electronic And Communication Engineering

Course Code: ECP204 **Semester:** 2024-2025 (Odd Semester)

Faculty In-Charge: Prof.K. SURENDER

Date of Submission: November 19th, 2024

Content

Ultrasonic Sensor Proximity Circuit

Abstract

This project focuses on designing a proximity detection circuit using an ultrasonic sensor. The circuit utilizes basic electronic components, including a 555timer IC, a comparator and ESP32 to generate and process signals. The primary purpose is to detect objects within a specific range by sending and receiving ultrasonic pulses, with the detected proximity indicated by an LED or buzzer and determining the distance at which object is placed. This simple, cost-effective approach to object detection is suitable for basic automation and alert systems.

Introduction

This project focuses on designing an ultrasonic proximity detection circuit, allowing for a simpler, cost-effective, and analog-based design. Ultrasonic sensors are commonly used in automation, robotics, and security systems due to their ability to measure distances accurately and perform non-contact object detection. The 555 timer IC operates as a pulse generator, sending periodic trigger signals to the ultrasonic sensor, which then emits sound waves at a frequency above human hearing. When these sound waves encounter an object within a specific range, they reflect back to the sensor, creating an echo pulse. By measuring the time delay between the transmitted pulse and received echo, the circuit can determine if an object is within the set proximity range. This distance-based output activates an LED or buzzer, serving as an indicator. The simplicity of this setup makes it suitable for applications like automated warning systems, obstacle detection in basic robotics, and entry-level object detection tasks

Aim:

To build a ultrasonic sensor based proximity sensor.

Components Required:

1. NE555 IC
2. HC-SR04 Ultrasonic Sensor
3. Bread board
4. 1K resistors(5)
5. 56K resistor(2)
6. 22uF capacitor (1)

7. 1uF capacitor (1)
8. 10k variable resistor (2)
9. BC557 transistor (3)
10. LEDs blue (4)
11. 3.7V battery

Specifications of Components:

1. NE555 IC:

The NE555 is a popular timer IC widely used in electronics for generating accurate time delays and oscillations. It operates in three main modes:

1. **Monostable Mode:** Produces a single timed pulse, useful for timers.

2. **Astable Mode:** Continuously outputs a square wave, ideal for blinking lights and generating clock pulses.

3. **Bistable Mode:** Works like a flip-flop, toggling between two states with each input pulse.

- **NE555 Pin Configuration:**

1. Pin 1 (GND): Ground; connects to the negative terminal of the power supply.

2. Pin 2 (Trigger): When voltage drops below 1/3 of the supply, it starts the timing cycle in monostable mode.

3. Pin 3 (Output): Outputs either a high or low signal based on the IC's configuration.

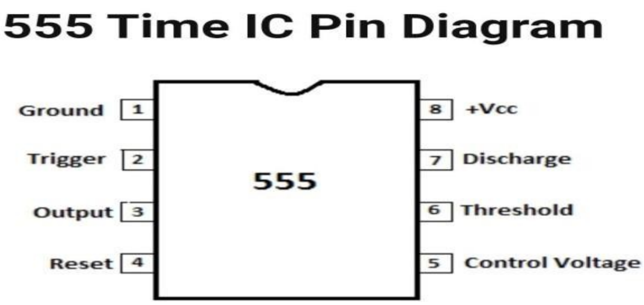
4. Pin 4 (Reset): Active-low reset; grounding this pin stops the timing function and resets the output.

5. Pin 5 (Control Voltage): Adjusts the threshold and trigger levels; typically connected to ground via a capacitor if not used.

6. Pin 6 (Threshold): Ends the timing cycle when voltage here rises above 2/3 of the supply.

7. Pin 7 (Discharge): Discharges the timing capacitor in astable and monostable modes.

8. Pin 8 (Vcc): Positive power supply; typically between 4.5V and 15V. The NE555 IC is commonly used in timing circuits, LED flashers, pulse generators, tone generators, and delay circuits due to its simplicity, versatility, and low cost.



2. HC-SR04 Ultrasonic sensor:

The HC-SR04 is an ultrasonic distance sensor commonly used for measuring distances in the range of 2 cm to 400 cm with an accuracy of about 3 mm. It operates by emitting ultrasonic sound waves and measuring the time it takes for the echo to return**.**

Key Features of the HC-SR04 Sensor:

1. Operating Voltage: 5V
2. Operating Current: < 15 mA
3. Range: 2 cm to 400 cm
4. Accuracy: ±3 mm

Pin Configuration:

1. VCC: Connects to the 5V power supply.
2. Trig (Trigger): Input pin; triggers the sensor to send out a sound wave when a 10µs pulse is applied.
3. Echo: Output pins; outputs a high pulse that reflects the time taken for the echo to return.
4. GND: Ground connection

3. Resistors (1K ohm, 56K ohm):

A 1kΩ resistor (1,000 ohms) and a 56kΩ resistor (56,000 ohms) are common in electronics, used to control current, divide voltage, and set bias levels.

1.1kΩ Resistor: Ideal for current limiting in LEDs, voltage division, and pull-up/pull-down roles in digital circuits. **2. 56kΩ Resistor:** Higher resistance, used for biasing in amplifiers, high-impedance voltage dividers, and signal pull-up/pull-down to minimize current flow in sensitive circuits.

4. Capacitors (1 microfarad, 22 microfarad):

a) 1 µF Capacitor:

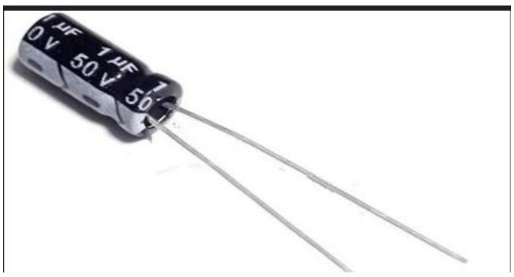
Common Uses:

a. **Decoupling/Bypass:** Filters high-frequency noise in power supply lines.
b. **Signal Coupling:** Blocks DC while allowing AC signals in audio circuits.
c. **Filtering:** Used in RC filters to smooth power supplies and manage unwanted frequencies.
d. **Timing Circuits:** Works in RC configurations for timing applications.

b) 22 µF Capacitor:

Common Uses:

a. **Power Supply Smoothing:** Reduces voltage ripple in DC supplies for stable output.
b. **Timing Circuits:** Paired with resistors in RC circuits for timing and pulse generation.
c. **Audio Filtering:** Blocks DC signals while allowing AC in audio applications



5. 10kΩ variable resistor:

A 10kΩ variable resistor, commonly known as a potentiometer or trimmer, is used in electronic circuits to adjust resistance and control current flow, voltage levels, and signal strength.

10kΩ Variable Resistor Pin Configuration:

a. Pin 1 (Terminal A): One end of the resistive track (connect to positive voltage or ground).

b. Pin 2 (Wiper - Terminal B): Adjustable middle terminal (output for variable voltage).
c. Pin 3 (Terminal C): Other end of the resistive track (connect to opposite terminal of Pin 1).

6. BC557 Transistor (pnp):

The BC557 is a PNP bipolar junction transistor (BJT) commonly used in electronic circuits for switching and amplification applications.
Type: PNP
Maximum Collector Current (I_C): 100 mA
Maximum Collector-Emitter Voltage (V_CE): -45 V
Maximum Power Dissipation (P_D): 500 mW
Current Gain (h_FE): Typically ranges from 100 to 400, depending on the operating conditions.

7. LEDs:

A LED (Light Emitting Diode) is a semiconductor device that emits light when an electric current flows through it. LEDs are widely used in various applications due to their efficiency, longevity, and compact size. Here blue LED is used for indication.

8. Battery: A 3.7 volt battery is used for power supply to the set up.

ESP32 micro-controller and Arduino IDE

The ESP32 is a powerful micro-controller with Wi-Fi and Bluetooth capabilities. In this project, its role is:

- **Control**: It acts as the central controller for managing the ultrasonic sensor's operation and processing the sensor data.
- **Signal Processing**: The ESP32 reads the signal from the ultrasonic sensor's echo pin, calculates the time delay, and uses that information to calculate the distance.
- **Data Calculation**: Using the time delay and the known speed of sound (typically 343 meters per second), the ESP32 calculates the distance. The formula used is:

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

The division by 2 accounts for the round trip of the sound wave (from the sensor to the object and back).

- **Output/Display**: The ESP32 can display the calculated distance on a display (e.g., LCD, OLED) or send it wirelessly to another device, such as a smartphone or a computer, via Wi-Fi or Bluetooth.

How It Works in the Arduino IDE:

1. **Setup the Project in Arduino IDE**:

   - **Install the ESP32 Board in Arduino IDE**: Before coding, make sure you have the **ESP32 board** installed in your Arduino IDE. You can do this by going to **Tools > Board > Board Manager** and searching for **ESP32**.

2. **Wiring**:

   - **Ultrasonic Sensor to ESP32**:

     1. **VCC** of the ultrasonic sensor goes to **3.3V** (on ESP32).
     2. **GND** of the ultrasonic sensor goes to **GND** (on ESP32).
     3. **Trigger Pin** of the ultrasonic sensor to a GPIO pin, e.g., **GPIO 23**.
     4. **Echo Pin** of the ultrasonic sensor to another GPIO pin, e.g., **GPIO 22**

Arduino Code:

```
#include <Wire.h>

#define echoPin 2             // CHANGE PIN NUMBER HERE IF YOU WANT TO USE A DIFFERENT PIN

#define trigPin 4             // CHANGE PIN NUMBER HERE IF YOU WANT TO USE A DIFFERENT PIN

long duration, distance;

void setup(){

  Serial.begin (9600);

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

}

void loop(){

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);


  duration = pulseIn(echoPin, HIGH);

  distance = duration / 58.2;

  String disp = String(distance);


  Serial.print("Distance: ");

  Serial.print(disp);

  Serial.println(" cm");

  delay(1000);

}
```

Working

In this project, the **ESP32** is used as the central controller to calculate the distance measured by the **ultrasonic sensor** (e.g., HC-SR04) with the help of a **555 timer IC**. The **555 timer** is configured in **astable mode**, where it generates a precise timing pulse to trigger the ultrasonic sensor. When the **ESP32** sends a signal to the **trigger** pin of the ultrasonic sensor, it emits an ultrasonic pulse. The sensor's **echo** pin receives the reflected pulse, and the ESP32 measures the time it takes for the pulse to travel to the object and back. The **555 timer** helps ensure that the timing for the pulse generation is accurate. The ESP32 then calculates the distance using the time-of-flight formula, which relates the duration of the pulse to the distance. Finally, the calculated distance is displayed on a **serial monitor** or transmitted over Wi-Fi/Bluetooth for further use. The 555 timer in this setup provides precise timing control for the ultrasonic pulse, while the ESP32 handles the logic, distance computation, and output display.

Results and Discussion

Upon testing the circuit with objects at various distances, the output (LED or buzzer) reliably activates when an object is within the designated range, successfully indicating the presence of an object. The analog setup achieves proximity detection with basic components, showing that the circuit effectively detects objects without a microcontroller.

During testing, it was observed that the accuracy of detection could be influenced by environmental factors such as temperature, which can affect the speed of sound, and by the precision of the timing components. Fine-tuning the comparator threshold improved reliability, but the circuit's overall accuracy remains limited compared to digital alternatives with microcontrollers. The simplicity of this analog circuit is beneficial for low-cost and low-complexity applications, though it may not be ideal for tasks requiring precise distance measurement or adaptable detection ranges. This approach is sufficient for entry-level object detection tasks, such as basic automation alerts, where exact distance measurement is not critical.

Conclusion

The project successfully demonstrates that an ultrasonic proximity detection circuit can be constructed without using a microcontroller, achieving basic object detection capabilities. The use of a 555 timer and comparator enables the circuit to trigger an output when an object is detected within a specified range. This analog approach provides a low-cost and straightforward solution for proximity detection applications, though it may lack the precision of microcontroller-based designs. The project shows that this design is feasible for simple automation systems and warning indicators, highlighting its effectiveness for entry-level detection applications.

Applications

Robotics: Object avoidance systems.
Automation Systems: Detecting presence or absence of objects on conveyor belts.
Security Systems: Triggering alarms when an intruder is detected within a certain distance.
Vehicle Parking Sensors: Indicating proximity to obstacles without the need for complex control systems.