IR Sensor-Based Object Detection and Buzzer Alert System Using Raspberry Pi

Introduction:

Infrared (IR) sensor-based object detection is a technology that is widely used in various fields such as automation, security, and robotics. This system helps in detecting the presence of objects without any physical contact. It is highly reliable and efficient, making it a preferred choice in multiple applications.

The fundamental principle behind an IR sensor is the emission and detection of infrared light. The sensor emits infrared rays, which, upon hitting an object, get reflected back. If the object is close enough, the reflected infrared signal is detected by the sensor's receiver module. Based on this detection, the sensor generates an output signal that can be processed further.

Components Required:

- Raspberry Pi (any model with GPIO support)
- IR Sensor Module (with digital output)
- Buzzer
- Resistors (if needed)
- Jumper Wires
- Breadboard
- Power Supply (5V for Raspberry Pi)

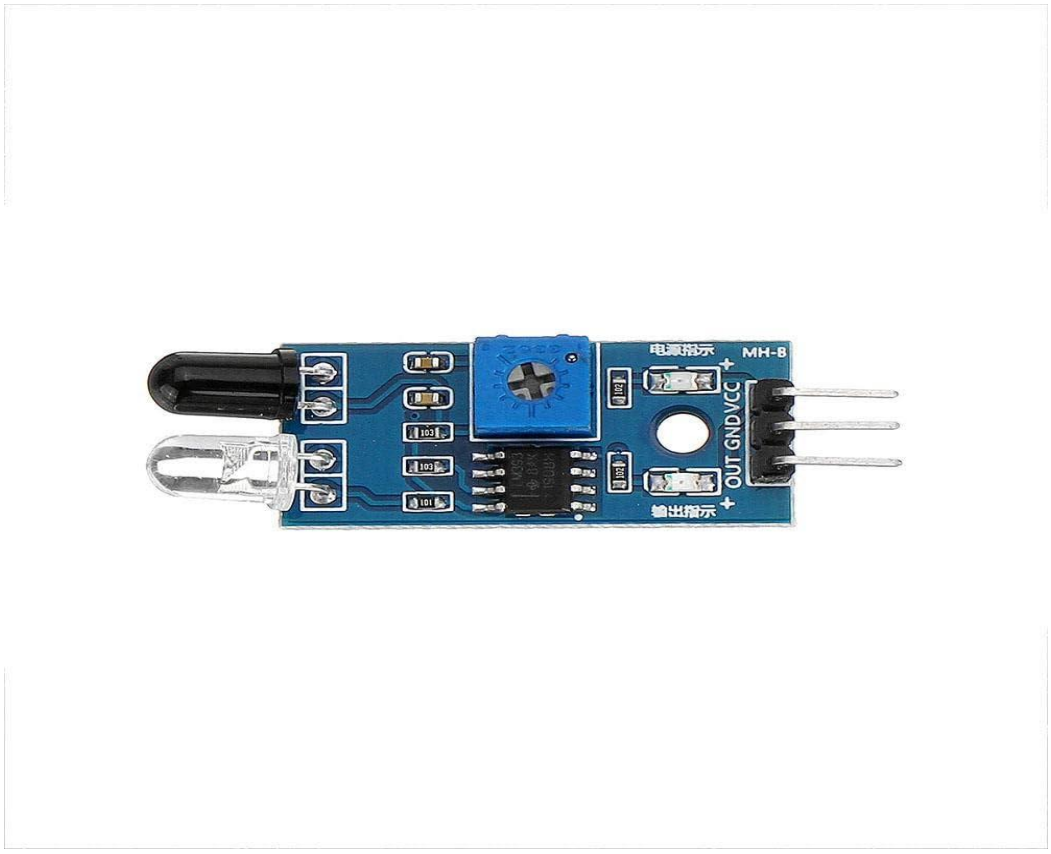
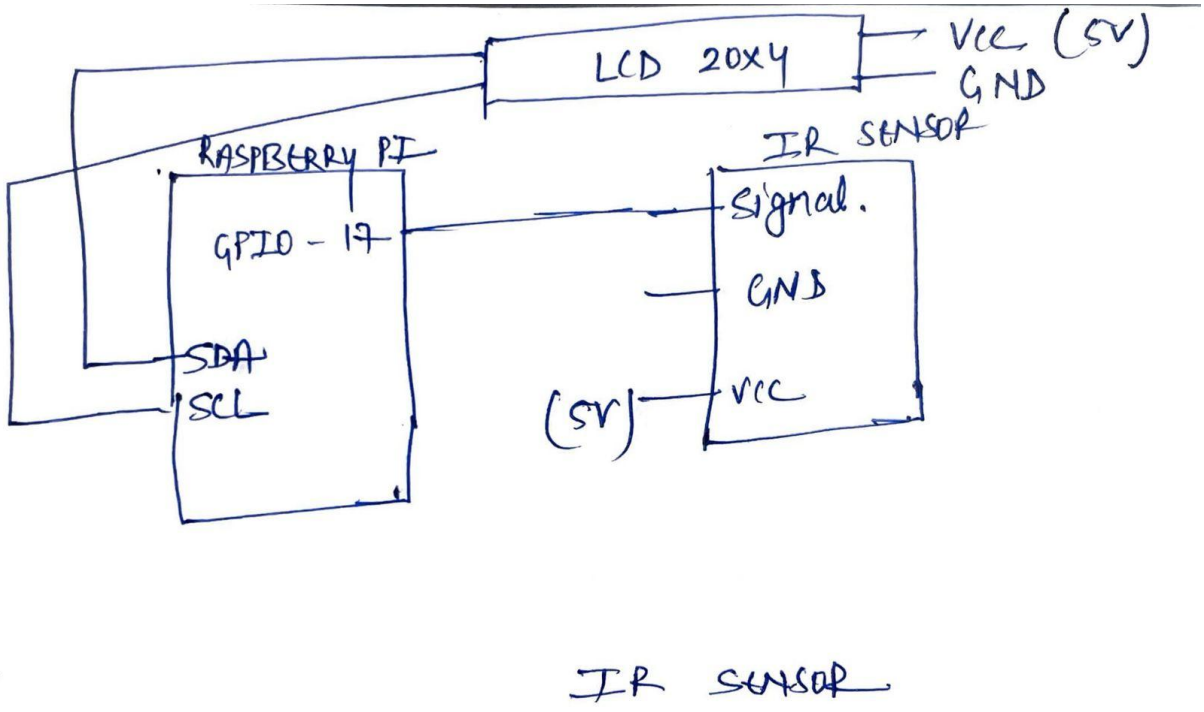


Fig 1. IR Sensor

Circuit Diagram and Connections:

- IR Sensor
    - VCC → 5V on Raspberry Pi
    - GND → GND on Raspberry Pi
    - OUT → GPIO 17 on Raspberry Pi
- Buzzer
    - Positive terminal → GPIO 27 on Raspberry Pi
    - Negative terminal → GND



Applications:

- Security Systems – Intruder detection and alarm systems.
- Automation – Object detection in industrial processes.
- Robotics – Obstacle detection for navigation.
- Parking Systems – Vehicle detection for smart parking.

Learnings:

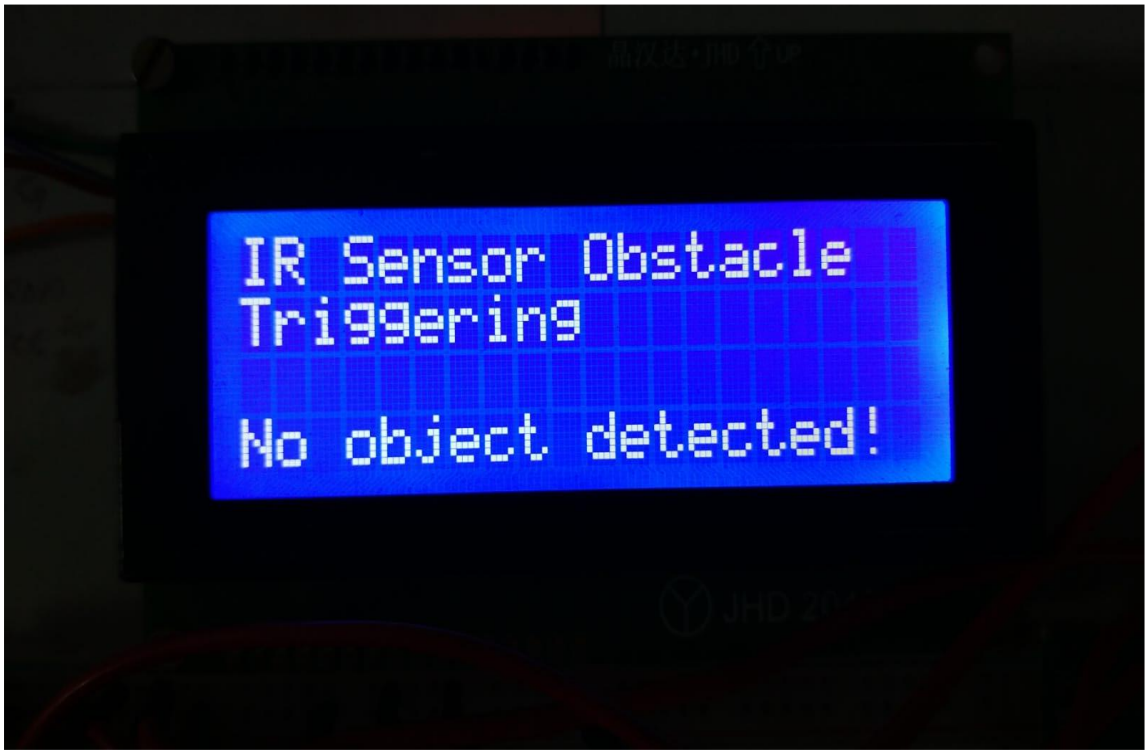
- Understanding IR sensor working and interfacing with Raspberry Pi.
- Controlling GPIO pins using Python.

- Implementing a real-time alert system with a buzzer.
- Using basic electronic components for automation.

Conclusion:

The IR Sensor-Based Object Detection and Buzzer Alert System is a simple yet useful project that can be further enhanced for security and automation applications. By integrating additional sensors and connectivity modules, it can be scaled into an advanced smart detection system.

Outcome:



Program:

```
import RPi.GPIO as GPIO
import time
from RPLCD.i2c import CharLCD
```

```
lcd = CharLCD('PCF8574', 0x27)

lcd.clear()

lcd.cursor_pos = (0, 0)  # First row
lcd.write_string(f'IR Sensor Obstacle')
lcd.cursor_pos = (1, 0)  # First row
lcd.write_string(f'Triggering')

IR_SENSOR_PIN = 17  # Change according to your wiring
LED_Pin = 26

GPIO.setmode(GPIO.BCM)
GPIO.setup(IR_SENSOR_PIN, GPIO.IN)
GPIO.setup(LED_Pin, GPIO.OUT)

try:
    while True:
        lcd.clear()
        lcd.cursor_pos = (0, 0)  # First row
        lcd.write_string(f'IR Sensor Obstacle')
        lcd.cursor_pos = (1, 0)  # First row
        lcd.write_string(f'Triggering')
        if GPIO.input(IR_SENSOR_PIN) == 0:  # Assuming LOW means blocked
            GPIO.output(LED_Pin, GPIO.HIGH)
            lcd.cursor_pos = (3, 0)  # First row
            lcd.write_string(f'Object detected!')

            print("Object detected!")
        else:
            GPIO.output(LED_Pin, GPIO.LOW)
```

```
            lcd.cursor_pos = (3, 0)
            lcd.write_string(f'No object detected!')

            print("No object detected")
        time.sleep(0.5)  # Small delay to prevent excessive CPU usage

except KeyboardInterrupt:
    GPIO.cleanup()  # Clean up GPIO on Ctrl+C exit
```