Raspberry Pi Motion Sensor Alarm using PIR Sensor

A MINI PROJECT REPORT

By

Name	Roll No
V.Gayathri Krishna	322103211049
V.Pravallika	322103211050
Y.Vennala	322103211052
T.V.Harshitha	322103211053
G.Yasasri	322103211054

Under the esteemed guidance of

Mrs.Ch Sirisha

Assistant Professor

ECE Department

Department of Information Technology

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

[Approved by AICTE NEW DELHI, Affiliated to ANDHARA UNVIERSITY]

[Accredited by National Board of Accreditation (**NBA**) for B.Tech. CSE, ECE & IT – Valid from 2019-22 and 2022-25]

[Accredited by National Assesment and Accreditation Council(**NAAC**)– Valid from 2022-27]

Kommadi , Madhurawada, Visakhapatnam–530048

2025–2026

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING FOR WOMEN

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the mini project report titled "Interfacing 16*2 LCD display with Raspberry Pi" is a bonafide work of following III B.Tech. students in the Department of Information Technology, Gayatri Vidya Parishad College of Engineering for Women affiliated to Andhra University, during the academic year 2024-2025 Semester-6.

Name	Roll No
V.Gayathri Krishna	322103211049
V.Pravallika	322103211050
Y.Vennala	322103211052
T.V.Harshitha	322103211053
G.Yasasri	322103211054

Internal Guide

Mrs Ch.Sirisha

Assistant Professor

Abstract

This project focuses on the development of a simple yet effective motion detection alarm system using a Raspberry Pi and a Passive Infrared (PIR) sensor. The primary goal is to detect human movement within a specified area and trigger an audible alert through a buzzer, providing a basic yet functional security mechanism. The PIR sensor, connected to the Raspberry Pi's GPIO pins, senses infrared radiation changes caused by motion, and the system responds using Python-based control logic. The entire setup is beginner-friendly, cost-efficient, and ideal for learning purposes, offering practical exposure to sensor integration, real-time data processing, and embedded programming. It serves as a foundation for further exploration into home automation, IoT applications, and smart surveillance systems, demonstrating the versatility of the Raspberry Pi in building interactive and responsive hardware solutions.

TABLE OF CONTENTS

Raspberry Pi Motion Sensor Alarm using PIR Sensor

1. AIM

To develop a real-time motion detection system using Raspberry Pi and a PIR sensor.

2. COMPONENTS REQUIRED

Hardware Required:

- Raspberry Pi 3
- PIR Sensor
- Breadboard and Jumper Wires
- Buzzer

Software Required:

- Raspbian OS
- Python (3.x recommended)
- Thonny

3. THEORY

The core of this project revolves around interfacing a Raspberry Pi with a Passive Infrared (PIR) sensor and a buzzer to detect human motion and trigger an audible alert through GPIO pins. Understanding how the Raspberry Pi communicates with sensors and output devices is essential for comprehending the functioning of this real-time security system.

Raspberry Pi and GPIO

The Raspberry Pi is a compact, powerful single-board computer equipped with General Purpose Input/Output (GPIO) pins. These pins serve as the primary interface between the Raspberry Pi and external hardware components like sensors, actuators, and displays. Each GPIO pin can be individually programmed to act as an input (to receive data from a sensor) or an output (to control devices like buzzers and LEDs). This flexibility makes the Raspberry Pi ideal for building embedded systems and IoT-based projects.

PIR Sensor (Passive Infrared Sensor)

A PIR sensor is used to detect motion by sensing changes in infrared radiation emitted by surrounding objects, especially human bodies. It has three primary pins: VCC (power), GND (ground), and OUT (digital output). When the sensor detects motion, it sends a HIGH signal through the output pin, which can be read by the Raspberry Pi through a GPIO pin configured as input.

Working Principle

The PIR sensor constantly monitors the infrared levels in its field of view. When a warm object, like a human, moves across this area, it causes a sudden change in infrared levels. This change triggers the sensor to output a HIGH signal. The Raspberry Pi reads this signal and executes the corresponding action—in this project, it activates a buzzer connected to another GPIO pin set as output. Once motion is no longer detected, the signal returns to LOW, and the buzzer is turned off.

Pin Functions and Connections

- **PIR Sensor**:
  - **VCC** → 5V on Raspberry Pi
  - **GND** → Ground
  - **OUT** → GPIO (input mode)
- **Buzzer**:
  - **Positive terminal** → GPIO (output mode)
  - **Negative terminal** → Ground

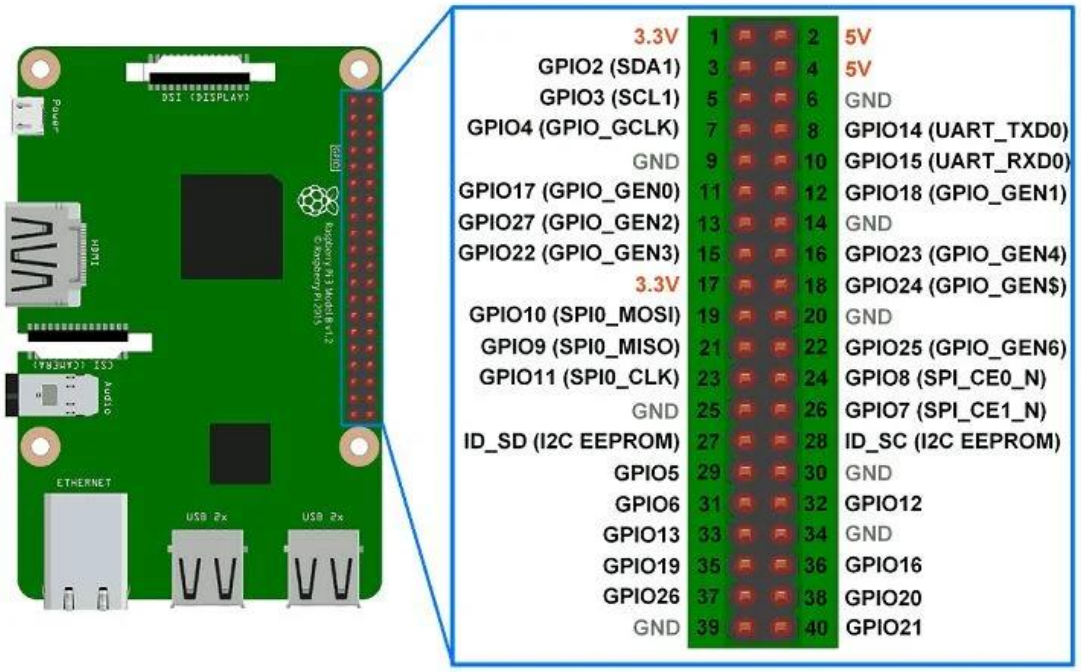
The buzzer serves as an audible alert, indicating when motion is detected. It is activated by setting the corresponding GPIO pin HIGH and turned off by setting it LOW.

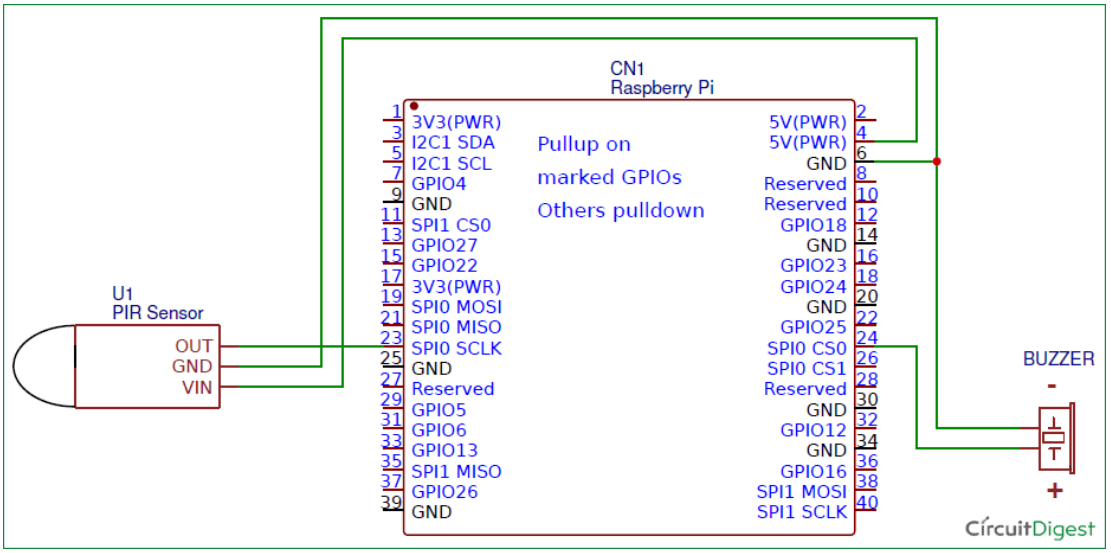
Software Interface

The Python programming language is used to control the hardware through Raspberry Pi's GPIO pins. The RPi.GPIO or gpiozero library simplifies the interaction by allowing users to easily set pin modes and define behaviors based on sensor input. The script continuously monitors the PIR sensor's output, and upon detecting a HIGH signal (motion), it activates the buzzer for the duration of motion detection.

Thonny is a beginner-friendly Python Integrated Development Environment (IDE) designed to simplify the coding experience, especially for students and novice programmers. Developed at the University of Tartu in Estonia, Thonny comes with Python built-in, so users can start coding right after installation without additional configuration. It features a clean and simple interface, a built-in debugger, and a variable explorer that allows users to track variable values step by step, making it easier to understand how code executes. Thonny also includes a package manager for easy installation of external libraries and supports MicroPython, making it ideal for IoT and embedded system projects using boards like Raspberry Pi Pico or ESP32. Overall, Thonny provides an accessible environment for learning and developing Python applications, from basic scripting to hardware-based projects.

4.CIRCUIT CONNECTION AND RASPBERRY PIN DESCRIPTION:





As shown in the above schematic diagram for Raspberry Pi and PIR sensor based motion detector, the positive pin of PIR sensor is connected with the pin 4 (5v) and ground pin of PIR sensor is connected with Pin 6 (Ground) of Raspberry Pi (You can find here the Pin Diagram of Raspberry Pi). The output pin of PIR sensor is connected with the GPIO 23 of Raspberry pi which is used to give input to Raspberry Pi. The GPIO pin 24 which is declared here for output is connected with positive of buzzer, and ground of buzzer is connected with the ground (pin 6) of raspberry pi.

5. PROCEDURE:

Step 1: Hardware Setup

1. Power off the Raspberry Pi before making any connections to avoid potential damage.
2. Connect the PIR sensor and buzzer to the Raspberry Pi GPIO pins as per the following configuration:

Component	Function	Raspberry Pi GPIO Pin
PIR Sensor VCC	Power supply	5V
PIR Sensor GND	Ground	GND
PIR Sensor OUT	Motion signal output	GPIO 23
Buzzer (+)	Audible alert activation	GPIO 24
Buzzer (–)	Ground	GND

Step 2: Software Setup

1. Set up Raspberry Pi OS on a microSD card and boot your Raspberry Pi.
2. Open the terminal on the Raspberry Pi.
3. Install required libraries (if not already installed):

```
sudo apt update
sudo apt install python3-rpi.gpio
```

Step 3: Source Code
Open a Python editor (like Thonny or nano) and enter the following code:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN)
GPIO.setup(24, GPIO.OUT)
try:
    time.sleep(2)
    while True:
        if GPIO.input(23):
          GPIO.output(24, True)
          time.sleep(1)
          GPIO.output(24, False)
          print("Motion Detected")
          time.sleep(5)
       time.sleep(0.1)
except:
    GPIO.cleanup()
```

Save the file as pirsensor.py.

Run the code.

Step 4: Modifications

- The buzzer duration and sensitivity can be adjusted using time.sleep() and the PIR sensor's onboard potentiometers.
- You can integrate an LCD display or camera module to display real-time data or capture images when motion is detected.

6. OBSERVATION

- "The PIR sensor successfully detected human motion by identifying infrared radiation changes."
- "Upon motion detection, the Raspberry Pi activated the buzzer as an alert."
- "The system loop efficiently checked for motion and responded with minimal delay."

7. ANALYSIS

- The interfacing between the Raspberry Pi and PIR sensor highlights the use of GPIO pins for real-time sensor communication in embedded systems.
- By using Python's RPi.GPIO library, the system processes input signals from the PIR sensor and triggers an output device (buzzer), offering a simple yet effective automation application.

- The project demonstrates core IoT concepts—sensing, processing, and actuation—in a modular way, making it suitable for integration with more advanced systems.
- The PIR sensor and buzzer functioned reliably during testing, with immediate feedback upon motion detection.

8. PRECAUTIONS

- Always power off the Raspberry Pi before connecting or disconnecting GPIO pins.
- Ensure correct GPIO pin mapping to prevent hardware damage or logical errors.
- Use resistors if necessary to protect components when connecting to GPIO pins.
- Test the PIR sensor range and delay time to match your desired application area.
- Avoid placing the sensor near heat sources or moving fans to prevent false detections.
- Ensure secure and proper wiring to avoid accidental disconnections during operation.

9.OUTPUT

CODE SCREENSHOT LCD DISPLAY PHOTO

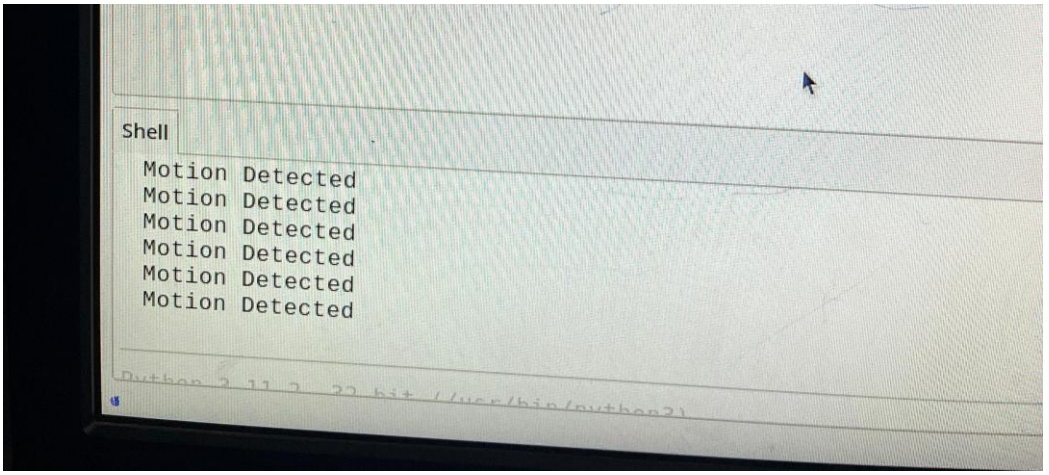


Fig: Motion detected is displayed in Thonny

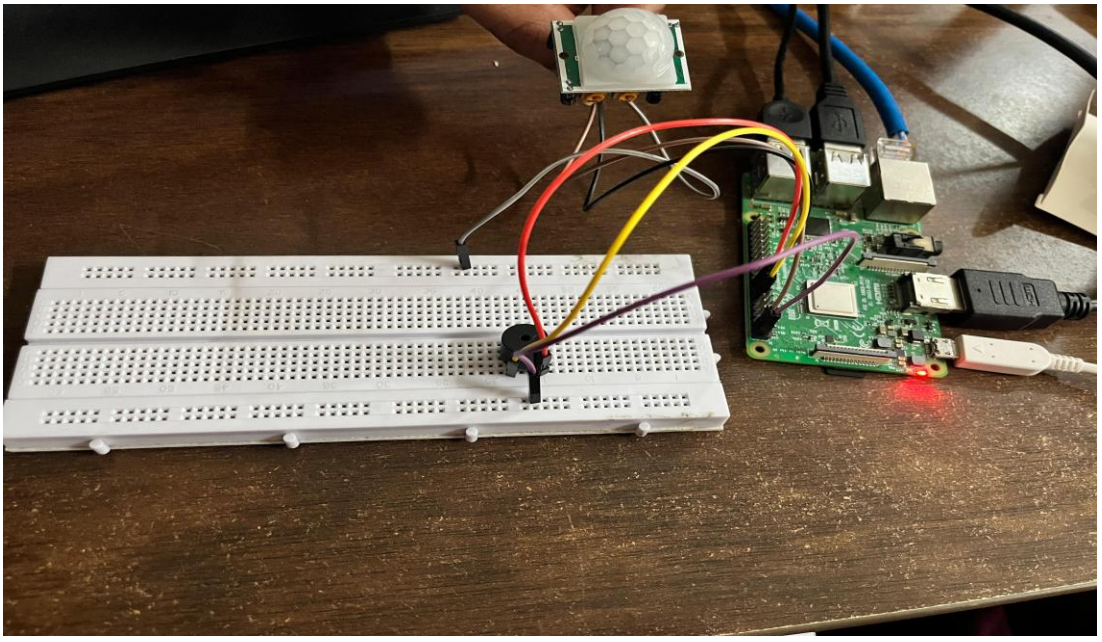


Fig: Interfacing Raspberry Pi with PIR Sensor and Buzzer

10.CONCLUSION

- This project successfully demonstrates the interfacing of a PIR motion sensor and a buzzer with a Raspberry Pi using Python. By leveraging the Raspberry Pi's GPIO pins and the simplicity of Python scripting, a real-time motion detection alarm system was implemented effectively.
- The system reliably detected motion and triggered an audible alert, validating its utility in basic security and monitoring applications.
- This project provides a practical introduction to embedded systems and IoT by bridging theoretical knowledge with real-world implementation. It lays the groundwork for more complex systems involving wireless alerts, camera integration, or smart home automation features.
- The simplicity and scalability of the design make it ideal for educational use, prototyping, and extension into larger projects.

11.REFERENCES

- Adafruit PIR Motion Sensor Guide– https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor
- Project link (for reference material and visuals) – https://search.app/NkKda3PdJrH3txt77