Module Code & Module Title
CS5068NI– Cloud Computing & IoT

Motion Detection Alarm System

Assessment Type
50% Group Report

Semester
2024 Spring

Group Members

London Met ID	Student Name
22066998	Muzammil
22066988	Dikshit Phuyal
22066682	Kipa Shrestha
22085653	Sulav Neupane

Assignment Due Date: 10th May 2024
Assignment Submission Date:10 May 2024
Submitted to: Mr. Sugat Man Shakya
Word Count:

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgements

We would like to express our sincere gratitude to my module leaders Mr Sugat Man Shakya and Mr. Ayush Bhakta Pradhanang for providing us with this wonderful opportunity to work on this IoT project. Their guidance helped us gain knowledge on the topic related to IoT. Lastly, we would like to thank my classmates for making valuable comments and suggestions on this project. Despite their busy schedules, they managed their time and provided unique ideas for our project. Again, we want to thank everyone who helped us.

Abstract

The goal of this project is to develop a smart security device by utilising a PIR (Passive Infrared) motion sensor. Components such as Node MCU, LED, Buzzer, PIR sensor, etc. were used in the construction of this project. Our purpose is to improve home and workplace security systems. Ordinary security systems can fail to detect issues on time, making situations unsafe. Our device will be quicker and smarter. It can detect movement and promptly alert individuals with a message. Our objectives are to develop the device, allow connectivity, and test it to be sure it works well. With its ability to identify vulnerabilities immediately and allow quick response, we believe this device will increase public safety.

Table of Contents

Table of Figure

Table of Tables

1. Introduction

Internet of Things (IoT) is the network of physical objects embedded with sensors, software, and other technologies to connect and share data with other devices and systems over the Internet. An IOT ecosystem is made up of smart devices with internet access that gather, transmit, and act on data they obtain from the environment using embedded systems including processors, sensors, and communication hardware (Gillis, 2023).

PIR (Passive Infrared) sensors, sensors that allow us to sense motion, are used to detect whether a human has moved in or out of the sensor's range. They are small, less expensive, and easy to use so they are commonly found in gadgets used in homes and businesses. They are commonly used in security alarms, the aerospace industry and automatically activated lighting systems. PIR sensors are important for the development of motion detection technology, whether they are used to optimise manufacturing processes or as part of security systems (Adafruit, 2024).

1.1. Current Scenario

In today's security landscape, traditional systems often fail to provide real-time protection, leaving homes and workplaces exposed to threats. Systems like security cameras and alarms, don't work until an incident has occurred. Potential attackers can exploit vulnerabilities due to a lack of security coverage created by the delay in detection and response. In addition, these systems could not be flexible enough to change with the requirements of security. However, as IoT technology expands, there's a chance of completely changing safety measures. IoT devices with sensors and internet connections can offer real-time warnings and early detection.

1.2. Problem Statement and Project as a solution

Many people worry about their surroundings' security, especially when they are not present. Traditional security systems might face setup and monitoring challenges that make them vulnerable. Furthermore, existing motion sensors may lack the intelligence to distinguish between normal activities and possible threats.

To deal with these issues, we are creating a simple yet sophisticated motion sensor. Our device will be simple to use and install, making it suitable for people who have different levels of technical knowledge. It will connect to the internet, allowing for remote monitoring from any location. The sensor will be able to differentiate between normal movements and unusual behaviours that may signal a problem. Our technology guarantees safety and security in a variety of contexts by providing intelligent and dependable surveillance.

1.3. Aim and Objectives

1.3.1. Aim

The main goal of this project is to develop a system that can detect motion from the PIR sensor and display how far the object is from the ultrasonic sensor.

1.3.2. Objectives

- To design and build a prototype of the motion alarm system.
- To integrate the sensor module with IoT communication protocols.
- To develop a user-friendly interface for configuring and monitoring the sensor.
- To implement real-time motion detection algorithms.

2. Background

The PIR (Passive Infrared) motion sensor is a widely used technology in various applications, ranging from security systems to automated lighting control. This section provides an overview of the functionality, significance, and design considerations associated with PIR motion sensors.

2.1. System Overview

The PIR (Passive Infrared) motion sensor system is a cornerstone technology in the realm of motion detection which provides an environmentally friendly and simple way to detect movement in various situations. The PIR sensor works by sensing heat emitted by people or objects nearby. PIR sensors are suitable for situations where energy savings and low interference are top priorities since they passively detect infrared radiation.

This project required seven important devices to build a working PIR motion sensor system. The PIR motion sensor and breadboard were all connected to the Arduino, which functioned as the central control unit. The circuit installation was made easier by the breadboard, which included resistors and LEDs that were set up to light up when motion was detected. The breadboard was connected to the motion sensor and Arduino, which allowed for smooth integration into the system design and provided visual feedback when motion was detected.

The PIR motion sensor detected movement within its range and sent signals to Arduino, activating the busser and LED, delivering real-time alerts in response to detected motion events. These devices worked together to successfully create a PIR motion sensor system, improving automation and security capabilities across a range of situations.

2.2. Design Diagrams

2.2.1. Block Diagram

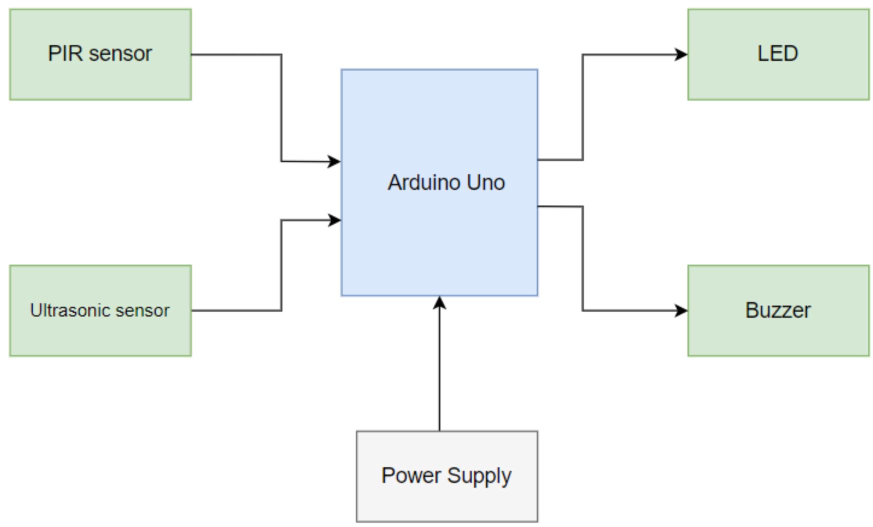


Figure 1: Block diagram.

The PIR sensor detects motion by sensing changes in infrared radiation, which is then processed by the Arduino to activate LED and buzzer, alerting based on detected motion.

2.2.2. System Architecture

The system architecture of PIR sensor motion is given below:

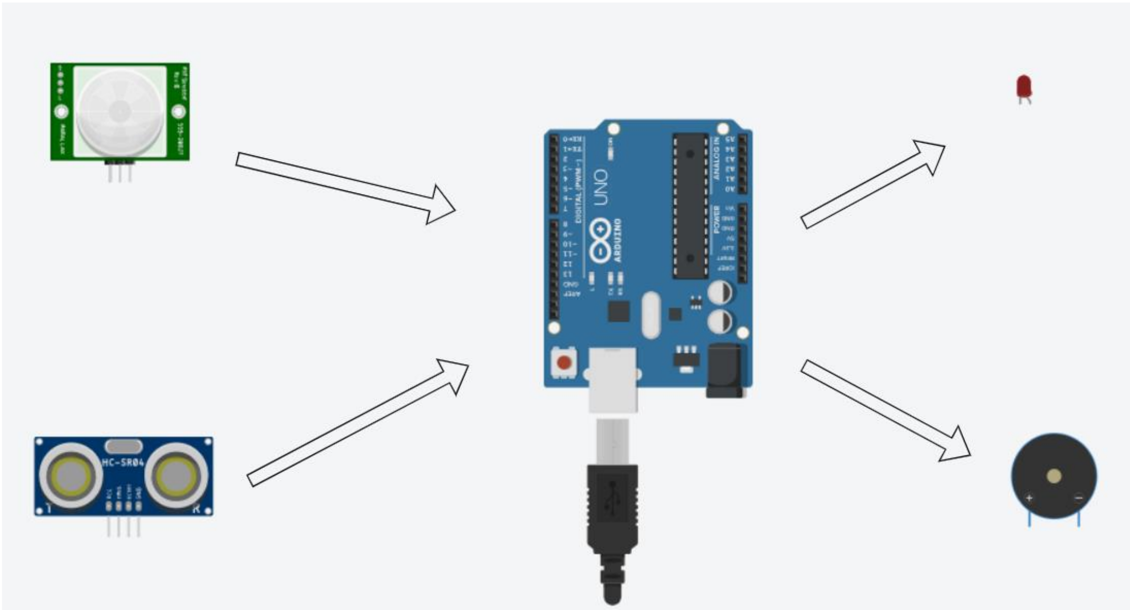


Figure 2: System Architecture of PIR Motion sensor.

2.2.3. Circuit Diagram

The figure below is the circuit diagram of PIR Sensor Motion.

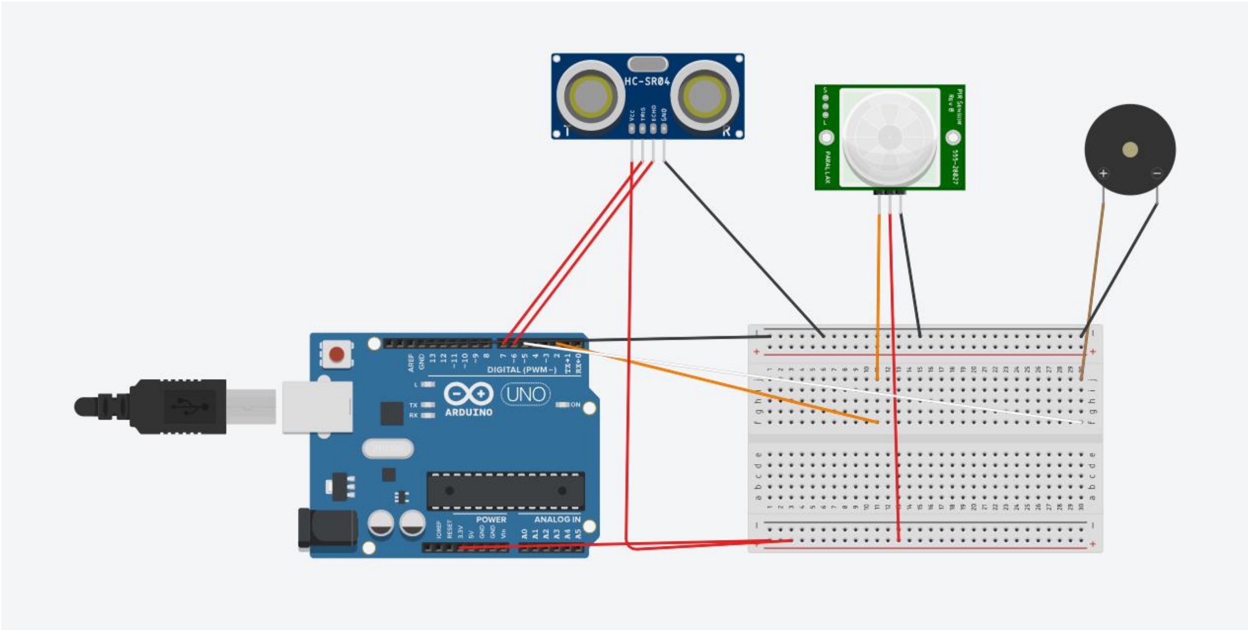


Figure 3: Circuit diagram of PIR Sensor Motion.

2.2.4. Schematics Diagram

The schematics diagram of PIR sensor motion is given below:

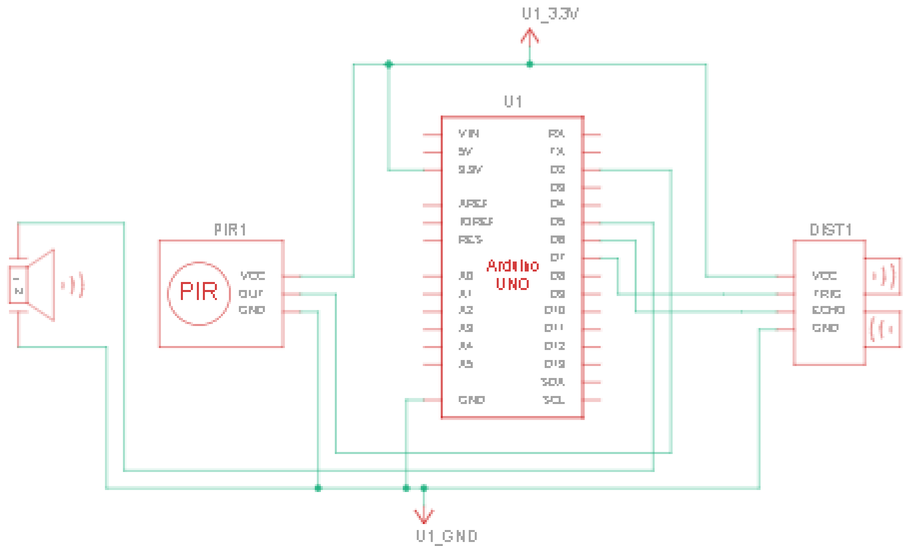


Figure 4: Schematic Diagram of PIR Motion Sensor.

2.2.5. Flow Chart

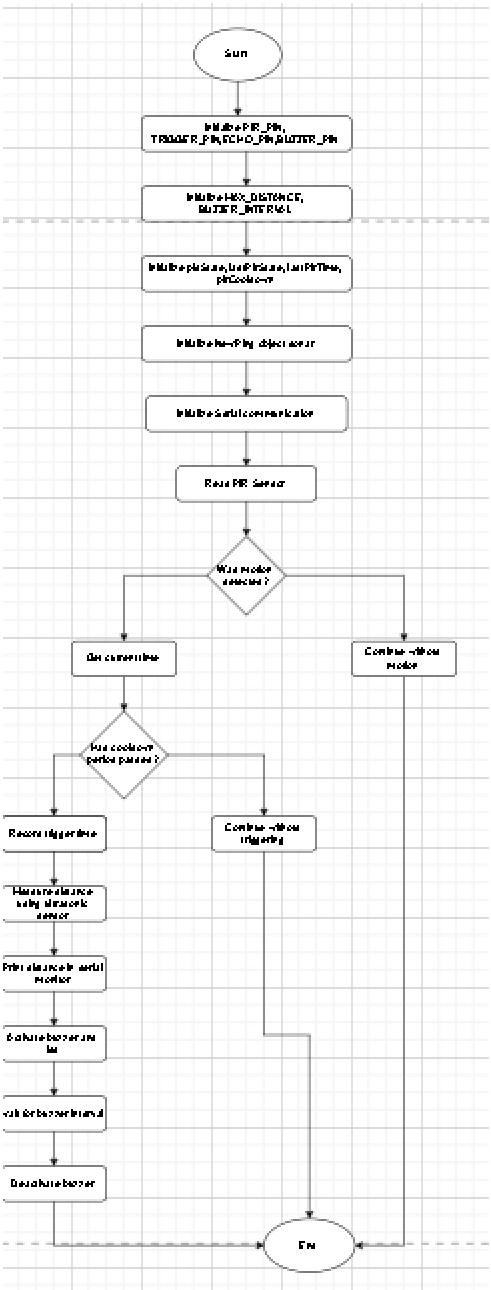


Figure 5: Flowchart

2.3. Requirement Analysis

2.3.1. Hardware Components

- **Node MCU**

  An open-source LUA-based firmware called NodeMCU was created for the ESP8266 Wi-Fi chip. NodeMCU firmware is included with the ESP8266 Development board/kit, also known as the NodeMCU Development board, in order to explore the capabilities of the ESP8266 chip (ElectronicWings, 2022).

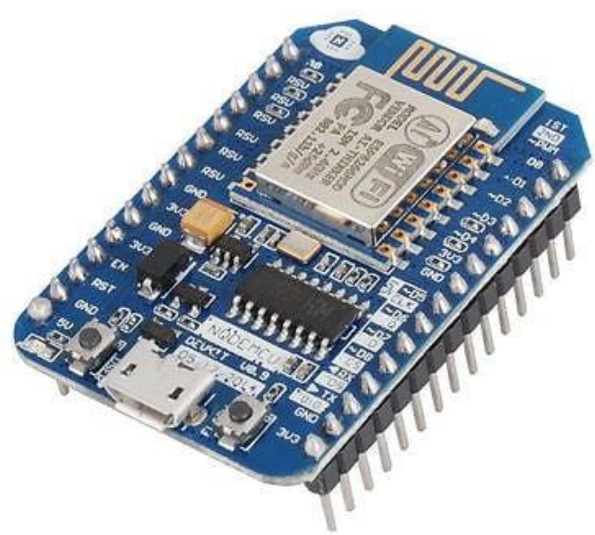


Figure 6: Node MCU.

- **PIR Sensor (HC-sr501)**

  It is low power, low cost, easy to interface and extremely popular. This PIR sensor itself is straightforward and works out of the box. Simply apply power 5V – 12V and ground. The sensor output goes HIGH when motion is detected and goes LOW when idle (no motion detected) (Last Minute Engineers, 2024).

Figure 7: HC-SR501 PIR Sensor.

- **Buzzer**

  It is a digital component that can be connected to digital outputs and emits a tone when the output is HIGH. This module can be used to provide sound feedback to your application just like the click sound of a button on a digital watch (Sensor Kit, 2024).

Figure 8: Buzzer.

- **Resistor**

  A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor (Tech Target Contributor, 2024).

Figure 9: Resistor (220 Ohm).

- **LED**

  A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it (Byjus, 2024).

Figure 10: LED.

- **Breadboard**

  A breadboard consists of plastic block holding a matrix of electrical sockets of a size suitable for gripping thin connecting wire, component wires or the pins of transistors and integrated circuits (ICs) (Schousek, 2018).

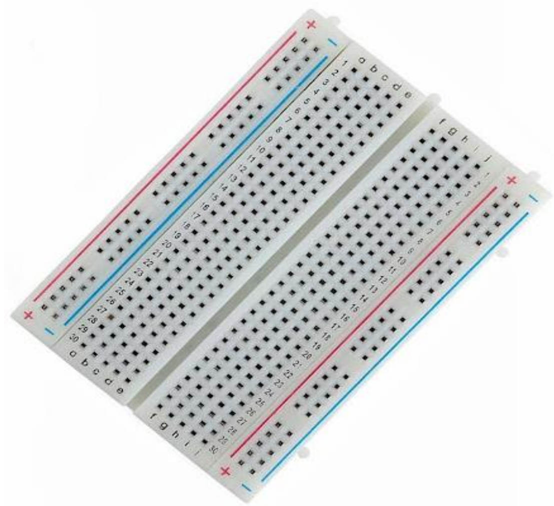


Figure 11: Breadboard.

- **Jumper Wire**

  Jumper wires used with breadboards that have connector pins at each end, allowing them to be used to connect two points without soldering (Hemmings, 2018).

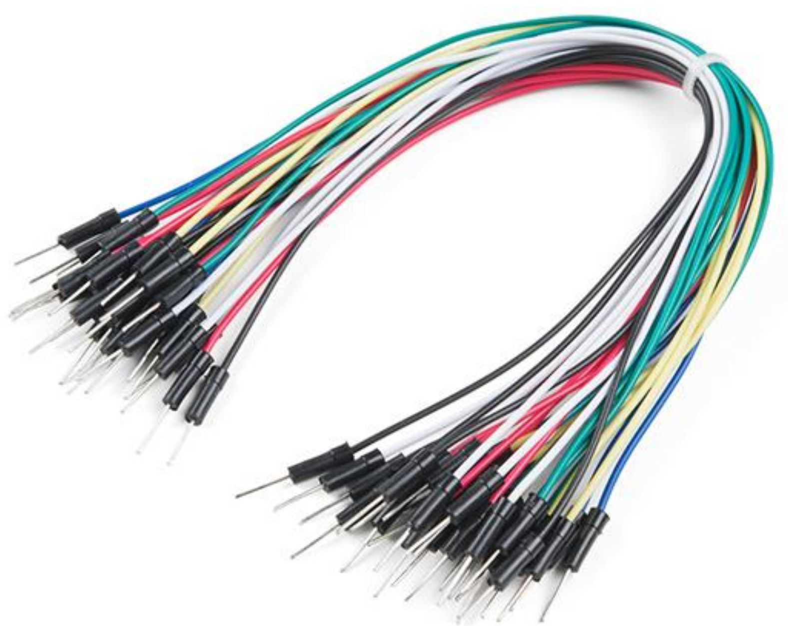


Figure 12: Jumper Wires.

- Ultrasonic Sensor
  An apparatus that uses ultrasonic sound waves to gauge an object's distance is called an ultrasonic sensor. An ultrasonic sensor transmits and receives ultrasonic pulses using a transducer to determine the proximity of an item (MaxBotix, 2023).

Figure 13: Ultrasonic Sensor

2.3.2. Software Components

- **Arduino IDE**

  The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board (Arduino, 2024).

- **MS Word**

  Microsoft Word is a word-processing program that allows for the creation of both simple and complex documents. It creates text documents that can be saved electronically or printed on paper (UA Little Rock, 2024).

- **Draw.io**

  It is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams.

3. Development

3.1. Planning and Design:

A security system based on passive infrared (PIR) sensors requires careful planning and design that involves multiple crucial processes. The first step is to specify the needs for the system, including the areas that need to be watched, the detection range, the sensitivity, and the power usage. The next step is component selection, with an emphasis on PIR sensors, relays, amplifiers, power supply circuitry, cameras, and processing computer hardware. Selecting a programming language, platform, and creating features like webcam detection and video recording are all part of software development. System dependability and sensor accuracy are confirmed by testing and evaluation. System architecture diagrams show how hardware is connected to one another, whereas hardware design entails drawing schematics for power supplies, amplifiers, and relay controls. Algorithms for user interface, relay control, and sensor inputs are defined as part of software design. Combination and examination Before finalizing documentation and maintenance plans to support continuous system operation and upkeep, make sure the system is functioning properly.

3.2. Resource collection:

PIR sensors for motion detection, differential amplifiers for signal processing, transformers and capacitors for the power supply, high-quality webcams compatible with the selected software platform, computer hardware with adequate processing power, software development tools like Microsoft Visual C++ and OpenCV, testing equipment like multimeters and oscilloscopes, documentation tools like word processing software and diagramming tools, and maintenance resources like spare parts and technical support contacts are all included in the extensive resource collection for the security system project. The quality and functionality of the system are ensured by the careful selection of reputable manufacturers or suppliers for each component, and the development and troubleshooting procedures are made easier by sufficient documentation and testing tools.

3.3. System Development

- Phase 1: Research and Component Selection

  The first phase involved comprehensive research into PIR sensor technology, including its operating principles, specifications, and integration requirements. This phase also encompassed identifying suitable microcontrollers and development boards capable of interfacing with the chosen PIR sensor while meeting project constraints such as power efficiency, processing capabilities, and available GPIO pins. Careful consideration was given to selecting components that would enable seamless integration and optimal performance of the PIR sensor system.

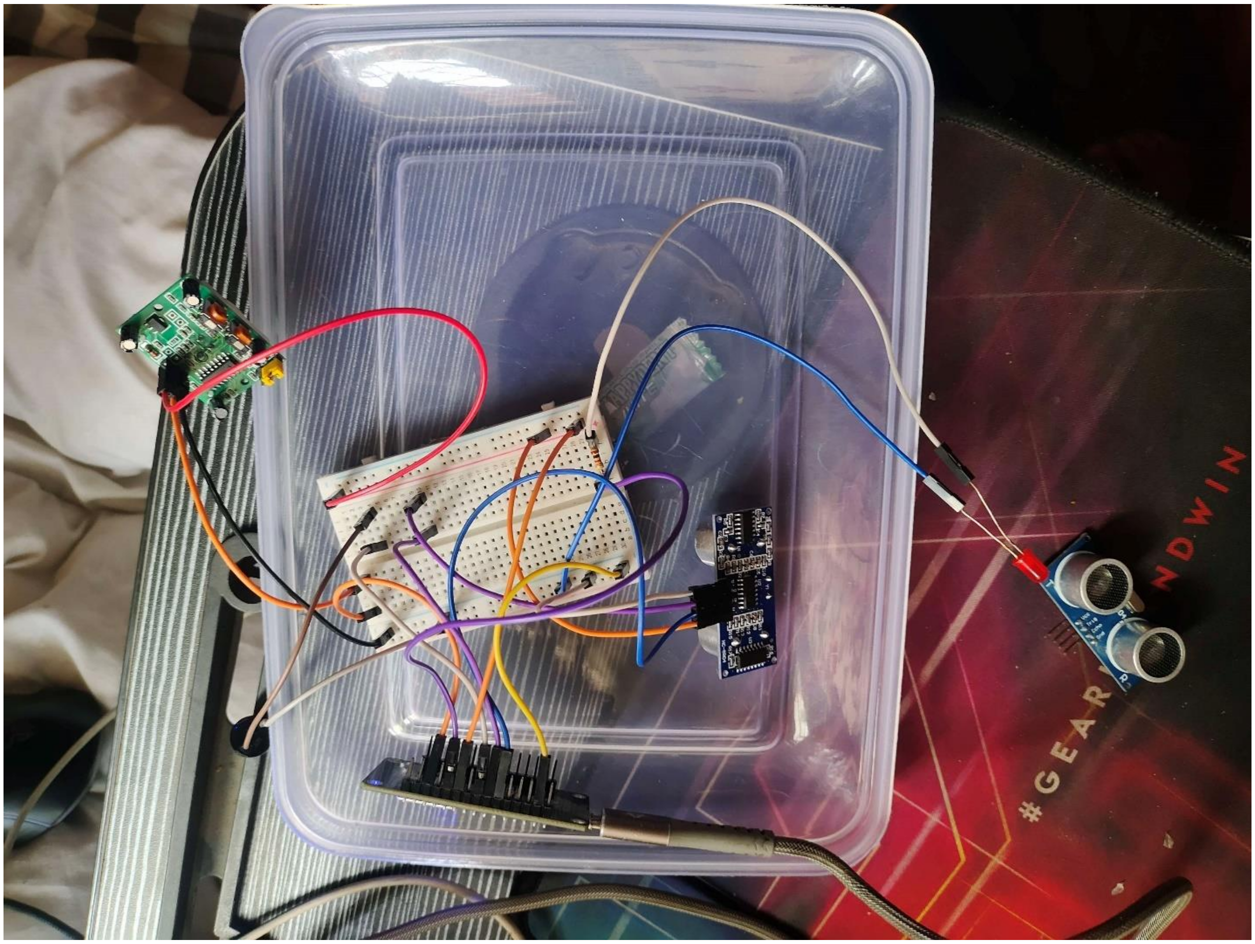
- Phase 2: Hardware Integration and Prototyping

  In Phase 2, the focus shifted towards hardware integration and prototyping. This involved physically connecting the PIR sensor to the selected microcontroller or development board, ensuring proper electrical connections and compatibility. Prototyping efforts aimed to establish basic functionality, including motion detection and signal processing, through the development of initial code snippets or firmware. Testing during this phase primarily focused on verifying hardware connections, detecting motion events, and assessing the reliability of sensor readings under different environmental conditions.
- Phase 3: Firmware Development and Optimization

  The third phase centred on firmware development and optimization to enable efficient operation of the PIR sensor system. This involved writing and refining code to handle sensor data, process motion detection events, and interface with other components or peripherals. Emphasis was placed on optimizing firmware algorithms to minimize false positives, reduce power consumption during idle periods, and maximize responsiveness to motion events. Iterative testing and debugging were conducted to identify and address any software-related issues, ensuring the robustness and reliability of the firmware.
- Phase 4: System Integration and Testing

  Phase 4 involved integrating the PIR sensor system with other relevant components or systems within the larger project framework. This may include interfacing with communication modules, data storage solutions, or user interfaces to enable comprehensive functionality and interaction. System integration efforts aimed to establish seamless communication and interoperability between different subsystems while maintaining overall system reliability and performance. Extensive testing was conducted to validate system behaviour, assess scalability, and identify potential integration challenges or compatibility issues.

- Phase 5: Deployment and Optimization
  The final phase focused on deploying the PIR sensor system in real-world environments and optimizing its performance based on practical feedback and usage scenarios. Deployment activities included installing the sensor system in target locations, configuring operational parameters, and conducting field testing to validate performance and reliability under diverse conditions. Feedback from real-world deployment informed ongoing optimization efforts, including fine-tuning sensor parameters, adjusting firmware algorithms, and implementing software updates to address any observed issues or improve overall system efficiency and effectiveness. Continued monitoring and optimization ensured that the deployed PIR sensor system remained responsive, reliable, and well-suited to its intended application context.



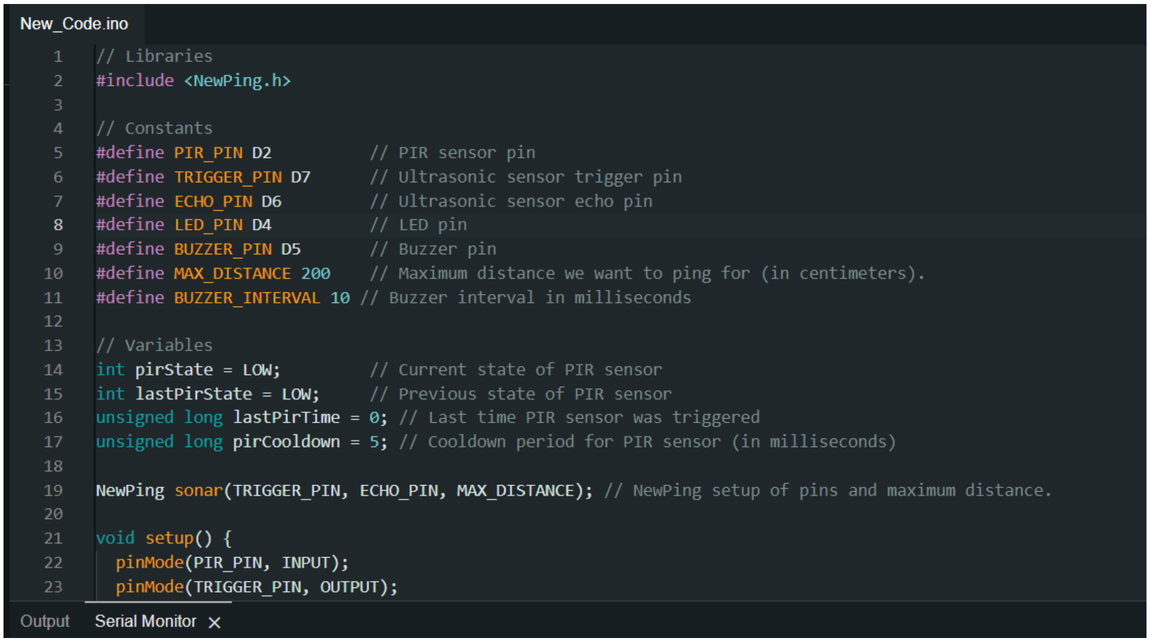
4. Results and Findings

Our project successfully created a motion sensor system using PIR sensors and Node MCU. We put the system to the test to see how fast and accurately it could detect motion. During testing, the system responded well to motion, activating functions such as activating LED or setting off alarms upon detecting movement. We also took measurements of the system's accuracy in motion detection and found that it performed well most of the time. The system demonstrated its ability to react quickly when needed by responding to motion events in seconds.

4.1.1. Test 1

Test	1
Objective	To compile the code and upload it.
Activity	Code was uploaded onto the Arduino by the help of Arduino IDE application.
Expected Result	The code would be compiled and run successfully.
Actual Result	The code was executed without any error.
Conclusion	Test was successful.

Table 1: To compile the code and upload it.




```
// Libraries
#include <NewPing.h>

// Constants
#define PIR_PIN D2          // PIR sensor pin
#define TRIGGER_PIN D7      // Ultrasonic sensor trigger pin
#define ECHO_PIN D6         // Ultrasonic sensor echo pin
#define LED_PIN D4          // LED pin
#define BUZZER_PIN D5       // Buzzer pin
#define MAX_DISTANCE 200    // Maximum distance we want to ping for (in centimeters).
#define BUZZER_INTERVAL 10 // Buzzer interval in milliseconds

// Variables
int pirState = LOW;         // Current state of PIR sensor
int lastPirState = LOW;     // Previous state of PIR sensor
unsigned long lastPirTime = 0; // Last time PIR sensor was triggered
unsigned long pirCooldown = 5; // Cooldown period for PIR sensor (in milliseconds)

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  pinMode(PIR_PIN, INPUT);
  pinMode(TRIGGER_PIN, OUTPUT);
```

Figure 14:Compiling the code

4.1.2. Test 2

Test	2
Objective	To test if the alarm system detects movement.
Activity	Placing an object in front of the PIR sensor.
Expected Result	The buzzer and LED would be activated when the object is brought near.
Actual Result	The buzzer and LED activated without any problem.
Conclusion	Test was successful.

Table 2:To test if the alarm system detects movement.

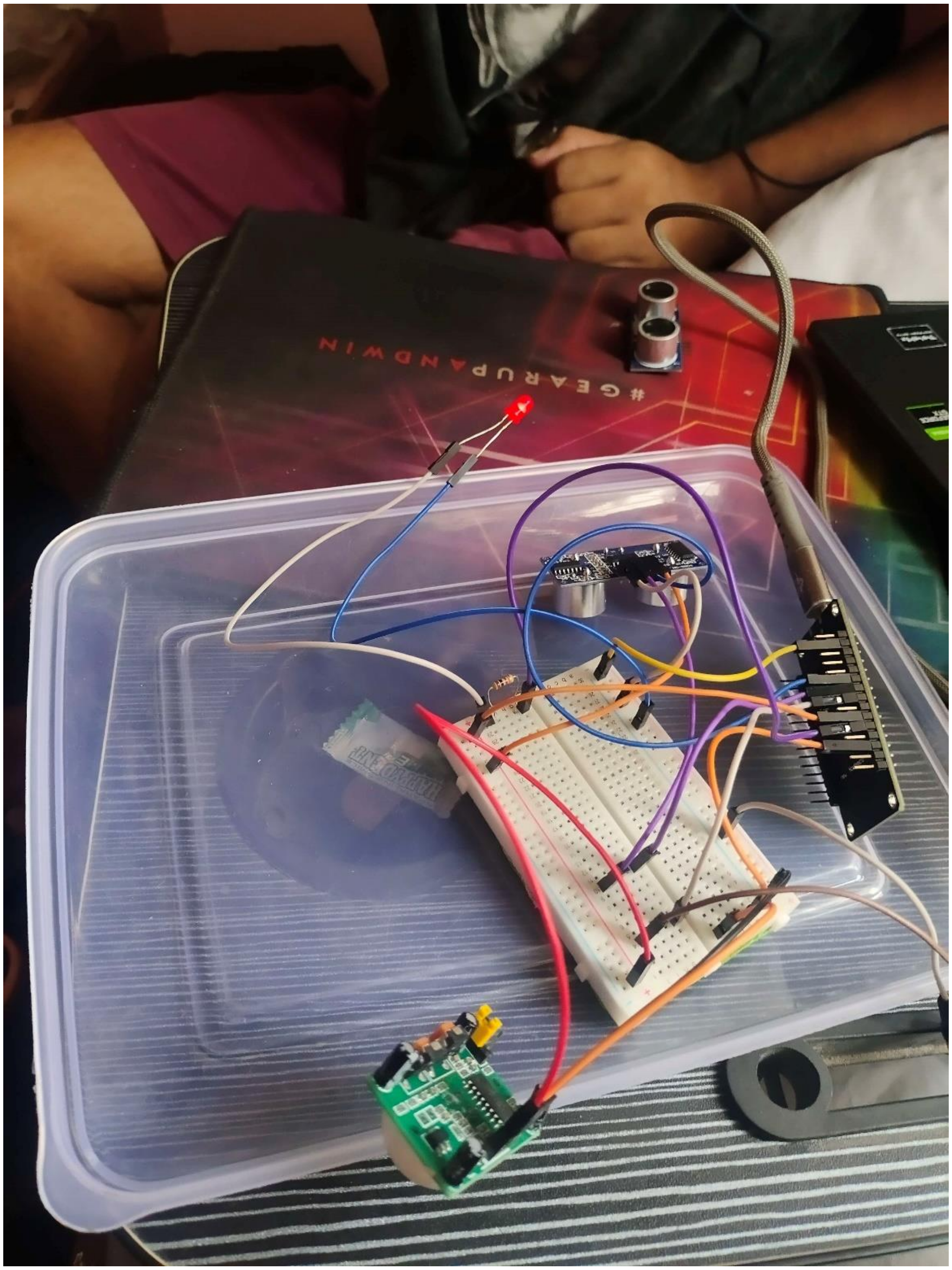


Figure 15: Movement was detected

4.1.3. Test 3

Test	3
Objective	To test if the ultrasonic sensor detects how far the object is.
Activity	Object was placed in the front of the sensor.
Expected Result	The distance between the object and sensor would be displayed in the serial monitor.
Actual Result	The distance was displayed in the serial monitor.
Conclusion	Test was successful.

Table 3:To test if the ultrasonic sensor detects how far the object is.

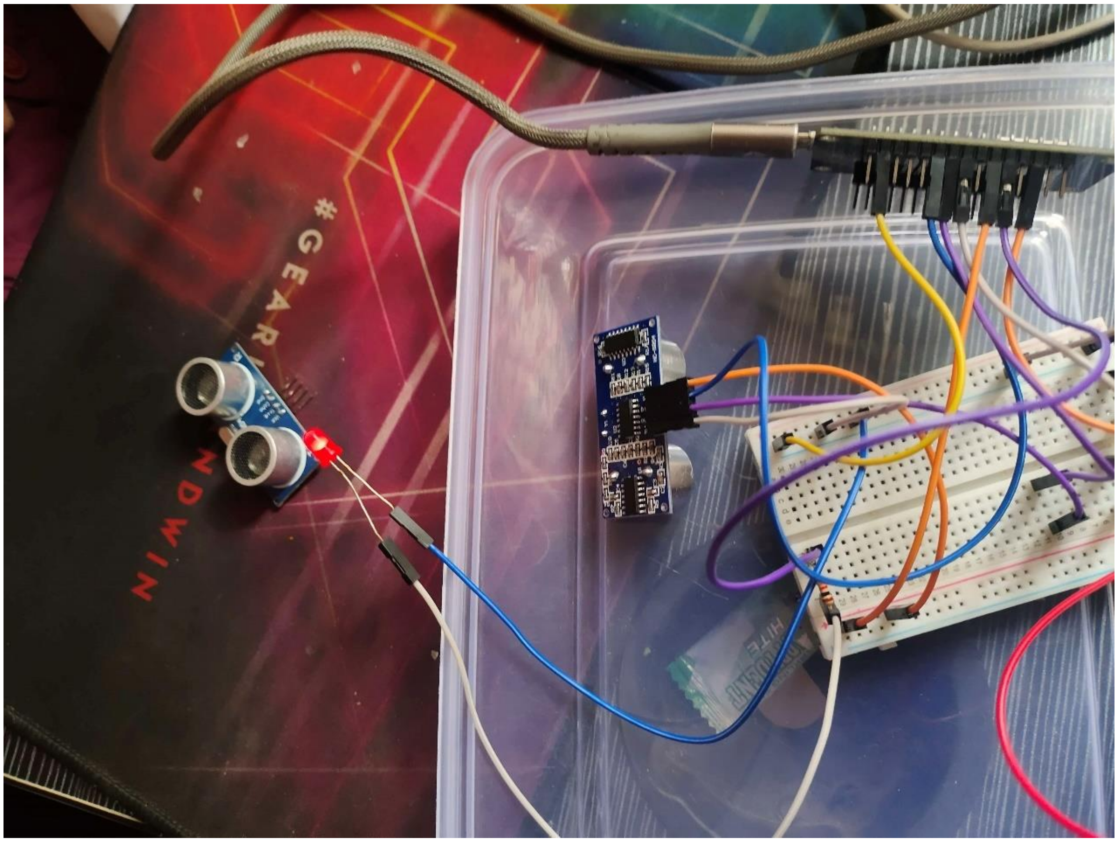


Figure 16: Motion was detected

```
11:52:33.767 -> Distance: 20 cm
```

Figure 17:Distance was measured

4.1.4. Test 4

Test	4
Objective	To test whether the project is working fully or not.
Activity	An object was placed in front of the sensor.
Expected Result	The buzzer and LED would activate and the serial monitor would display the distance between the object and sensor.

Actual Result	The buzzer and LED were activated and the distance was displayed.
Conclusion	Test was successful.

Table 4:To test whether the project is working fully or not.

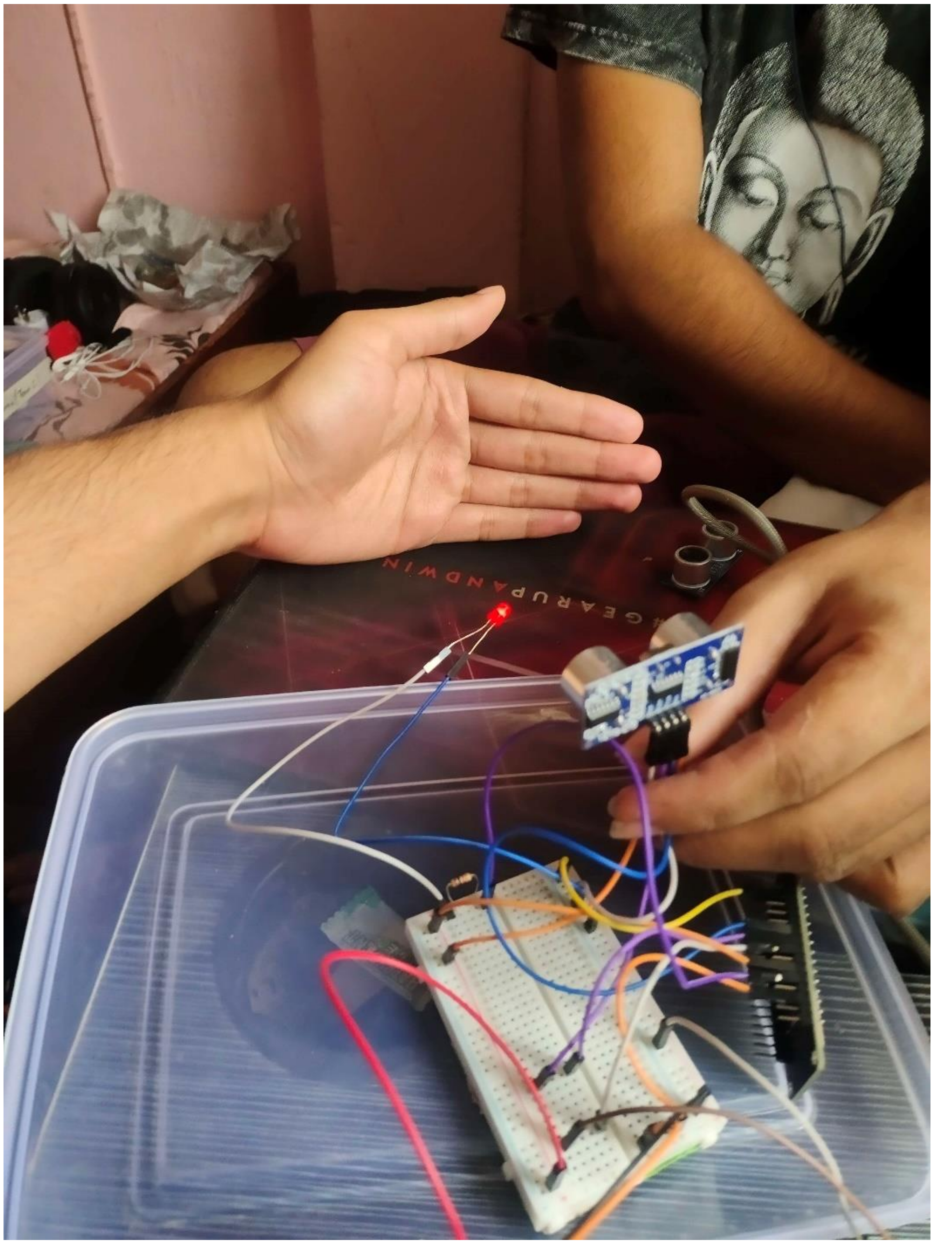


Figure 18: Final testing of project

Figure 19:Distance measured

5. Future Works

- Advanced notification system:

  To guarantee that users receive timely, accurate, and actionable warnings in response to security events, an advanced notification system within a security platform is essential. Users should be able to fully customize their notification preferences using such a system, including selecting communication channels and establishing escalation procedures according to the severity of the occurrences. The system may make sure that alerts are sent in a way that best fits users' operational demands and response capabilities by giving users the option to personalize how and when they receive notifications.

  Furthermore, the notification system's efficacy depends on its capacity to reduce false warnings and provide consumers with just pertinent information. This necessitates the incorporation of sophisticated algorithms that can differentiate between real security risks and safe environmental variables. While user feedback and historical data can gradually improve the system's accuracy, machine learning models can examine motion patterns and other sensor data to accurately identify anomalies. The sophisticated notification system improves the overall dependability and credibility of the security platform by giving users the confidence that they will only be informed of genuine security problems. This allows users to take prompt and decisive action in response to possible threats.

- Power efficiency optimization:

  Modern electronic systems must optimize power efficiency in their design and operation, especially in situations when minimizing energy usage without sacrificing functionality is necessary. This optimization includes a number of tactics used to lower power consumption in the system's various parts and subsystems. Using low-power parts and gadgets, including microcontrollers with enhanced power management capabilities and energy-efficient sensors, is one strategy. Furthermore, methods like as dynamic voltage and frequency scaling (DVFS) can dynamically modify the processors' operating voltage and clock frequency in response to workload demands, guaranteeing that energy is only used when required. Additionally, adaptive power gating and sleep modes for inactive or unused components can be enabled via sophisticated power management algorithms, greatly lowering overall power usage during periods of inactivity.

Optimizing software and algorithms to reduce computational complexity and increase energy efficiency is another way to optimize power efficiency. This can include methods to lessen computing burden and increase energy efficiency, like compiler optimizations, code profiling, and algorithmic optimizations. Reducing idle time and energy overhead can also be achieved by implementing effective programming paradigms like event-driven programming and asynchronous processing. Furthermore, based on power limitations and workload parameters, energy-aware scheduling algorithms can dynamically distribute system resources to maximize energy savings while maintaining optimal performance. Power efficiency optimization may dramatically increase battery life, lower energy costs, and lessen the environmental effect of electronic systems by using a comprehensive strategy that takes into account hardware, software, and system-level optimizations.

- Integration with Cloud Services:

  Modern system architecture is growing more and more dependent on integration with cloud services, which allows for improved data and service accessibility, scalability, and flexibility. Systems can transfer resource-intensive operations, such data processing and storage, to remote servers through seamless interaction with cloud platforms. This reduces the computational strain on local devices and enables optimal resource usage. With the help of this integration, users and numerous devices may collaborate and synchronize in real time, providing easy access to data and services from any location with an internet connection. Additionally, cloud services have sophisticated analytics and machine learning capabilities that let systems use big datasets for optimization, anomaly detection, and predictive analysis. This makes intelligent decision-making and automation possible.

Additionally, by integrating with cloud services, systems can benefit from a variety of specialized services and APIs, like payment processing, authorization, and authentication, without having to develop and maintain these features locally. This leverages the infrastructure and knowledge of cloud service providers to guarantee scalability and dependability while also speeding up the development process. Furthermore, strong security features like encryption, access control, and monitoring are provided by cloud-based solutions to protect sensitive data and guarantee adherence to privacy laws. Systems that embrace cloud integration can lower infrastructure costs and time-to-market, increase user value, and foster innovation, collaboration, and scalability while boosting competitiveness.

6. Conclusion

In conclusion, our project on the PIR motion sensor system has been a big step forward in making devices that can detect motion. We've put a lot of effort into planning and making sure our system works well. We have developed a system that is capable of accurately identifying motion by using special sensors that can detect heat from moving objects. We have demonstrated through testing that our technology performs as expected. It works well at sensing motion and is reliable. In the end, our project is a small piece of progress in the world of motion-sensing technology. It helps solve real problems and provides foundations for future advancements.

References

Adafruit, 2024. *learn.adafruit.com.* [Online]
Available at: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview
[Accessed 28 March 2024].

Arduino, 2024. *Software | Arduino.* [Online]
Available at: https://www.arduino.cc/en/software
[Accessed 25 April 2024].

Ashely, E., 2021. *What is Arduino UNO? | Arduino Uno Pinout & Program | DesignSpark.* [Online]
Available at: https://www.rs-online.com/designspark/what-is-arduino-uno-a-getting-started-guide#:~:text=Arduino%20UNO%20is%20a%20low,and%20motors%20as%20an%20output.
[Accessed 25 April 2024].

Byjus, 2024. *What is LED? - Definition, Working, Properties, Uses, Advantages.* [Online]
Available at: https://byjus.com/physics/light-emitting-diode/#:~:text=A%20light%2Demitting%20diode%20(LED,emitting%20light%20in%20the%20process.
[Accessed 25 April 2024].

ElectronicWings, 2022. *Introduction to NodeMCU.* [Online]
Available at: https://www.electronicwings.com/nodemcu/introduction-to-nodemcu
[Accessed 5 May 2024].

Gillis, A. S., 2023. *techtarget.com.* [Online]
Available at: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT
[Accessed 28 March 2024].

Hemmings, M., 2018. *What is a Jumper Wire?.* [Online]
Available at: https://blog.sparkfuneducation.com/what-is-jumper-wire
[Accessed 25 April 2024].

Last Minute Engineers, 2024. *How HC-SR501 PIR Sensor Works & How To Interface It With Arduino.* [Online]
Available at: https://lastminuteengineers.com/pir-sensor-arduino-tutorial/
[Accessed 25 April 2024].

MaxBotix, 2023. *How ultrasonic sesnors work.* [Online]
Available at: https://maxbotix.com/blogs/blog/how-ultrasonic-sensors-work#:~:text=What%20is%20an%20Ultrasonic%20Sensor,information%20about%20an%20object's%20proximity.
[Accessed 5 May 2024].

Schousek, T., 2018. *Breadboard - an overview | ScienceDirect Topics.* [Online] Available at: https://www.sciencedirect.com/topics/engineering/breadboard#:~:text=A%20breadboard%20consists%20of%20plastic,in%20rows%20of%20five%20sockets.
[Accessed 25 April 2024].

Sensor Kit, 2024. *Sensor Kit.* [Online] Available at: https://sensorkit.arduino.cc/sensorkit/module/lessons/lesson/04-the-buzzer
[Accessed 25 April 2024].

Tech Target Contributor, 2024. *What is a Resistor? - Definition from WhatIs.com.* [Online] Available at: https://www.techtarget.com/whatis/definition/resistor#:~:text=A%20resistor%20is%20an%20electrical,device%20such%20as%20a%20transistor.
[Accessed 25 April 2024].

UA Little Rock, 2024. *Information Technology Services.* [Online] Available at: https://ualr.edu/itservices/applications/v/microsoft-word/
[Accessed 25 April 2024].

Appendix

Code

```
// Libraries
#include <NewPing.h>

// Constants
#define PIR_PIN D2          // PIR sensor pin
#define TRIGGER_PIN D7      // Ultrasonic sensor trigger pin
#define ECHO_PIN D6         // Ultrasonic sensor echo pin
#define LED_PIN D4          // LED pin
#define BUZZER_PIN D5       // Buzzer pin
#define MAX_DISTANCE 200    // Maximum distance we want to ping for (in centimeters).
#define BUZZER_INTERVAL 10 // Buzzer interval in milliseconds

// Variables
int pirState = LOW;         // Current state of PIR sensor
int lastPirState = LOW;     // Previous state of PIR sensor
unsigned long lastPirTime = 0; // Last time PIR sensor was triggered
unsigned long pirCooldown = 5; // Cooldown period for PIR sensor (in milliseconds)

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  pinMode(PIR_PIN, INPUT);
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
```

```
  pinMode(LED_PIN, OUTPUT); // Set LED pin as output
  pinMode(BUZZER_PIN, OUTPUT); // Set buzzer pin as output
  Serial.begin(9600);
}

void loop() {
  // Read PIR sensor
  pirState = digitalRead(PIR_PIN);

  // If PIR sensor detects motion
  if (pirState == HIGH && lastPirState == LOW) {
    unsigned long currentMillis = millis();
    // Check if cooldown period has passed since last trigger
    if (currentMillis - lastPirTime > pirCooldown) {
      // Record the time of this trigger
      lastPirTime = currentMillis;
      // Trigger ultrasonic sensor to measure distance
      int distance = sonar.ping_cm();
      Serial.print("Distance: ");
      Serial.print(distance);
      Serial.println(" cm");
      // Turn on the LED
      digitalWrite(LED_PIN, HIGH);
      // Turn on the buzzer
      digitalWrite(BUZZER_PIN, HIGH);
      // Delay for a short period
      delay(500);
      // Turn off the LED
```

```
      digitalWrite(LED_PIN, LOW);
      // Turn off the buzzer
      digitalWrite(BUZZER_PIN, LOW);
    }
  }

  // Save the current state for next loop iteration
  lastPirState = pirState;
}
```

Individual Contribution Plan

The individual contributions of each member of the group to complete the project are mentioned below:

Name	**Task**
Muzammil	Responsible for coding and physical integration of the project
Dikshit Phuyal	Responsible for coding and physical integration of the project
Kipa Shrestha	Responsible for half of the report of this coursework
Sulav	Responsible for the other half of the report

Table 5:Individual Contribution Plan.