LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CS5068NI– Cloud Computing & IoT

Motion Detection System

Assessment Type
10% Proposal Report / 50% Group Report

Semester
2024 Spring

Group Members

London Met ID	Student Name
22085621	Shreeya Pandey
23050184	Prasanna Mahat
23050340	Mahan Shrestha
23049340	Anup Kunwar Kamar
23050362	Sujal Shrestha

Assignment Due Date: January 7, 2025
Assignment Submission Date: January 6, 2025
Submitted to: Mr. Sugat Man Shakya
Word Count:4820

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.

Acknowledgement

We like to extend our heartfelt gratitude to our esteemed lecturer, Mr. Sugat Man Shakya and tutorial/workshop instructor, Mr. Ayush Bhakta Pradhanang, for giving us the opportunity to work on this IoT project.

The project focused on a motion detection system that has been an enriching experience that allowed us to do more research and practical implementation, and we came to know about various things related to IoT and increased our knowledge on the respective topic.

We also wish to express our sincere appreciation to our friends; whose support and collaboration were a contributing factor to the success of this project. Despite their demanding schedules, they provided unique ideas and dedicated time to help us in this project.

Abstract

Motion detection is a technology that notifies users when it detects movement within a specified area. Because it can initiate video recordings, set off alarms, and send notifications to a connected device, it is frequently utilized in security systems. Motion detection is a pure pulse of technology that has been purposely developed to identify and react to any form of motion in the surroundings.
The project aims to have a system for detecting movement inexpensively to provide secure homes and small businesses with a reliable method. The system would be very basic, made of a motion sensor probably a Passive InfraRed (PIR) sensor, and a microcontroller to detect movement and send an alert in real time over text. Therefore, no loud sounds have to be produced. Instead, it gives a more discrete and very practical solution to traditional security systems. It is easily installed, cost-effective, and assures continuity of service even when there is a power failure. In this paper, the major focus is to reduce false alarms, increase sensitivity, and allow easy expansion with more sensors or features. The following improvements shall make this system intelligent in threat detection, storage on the cloud for data, integration of apps with mobile devices, and improved weather versatility to make it more adaptive and scalable as the need for security will continue to grow.

Table of Contents

Table of Figure

Table of Tables

1 Introduction

Motion detection is the perception of an object's change of position with respect to the background or of an event in its environment. In other words, it is significant in security, robotics, gaming, and intelligent home systems. The two main ways to detect motion are mechanical and electronic. Mechanical motion detection is a kind of physical device using levers, springs, or pressure plates that react to motion and function like the seismographs or traditional alarm systems in use.

In contrast, a type of motion detection dependent on electronics utilizes more superior sensors, which may involve typical infrared detectors, ultrasonic devices, cameras, or even the well-known Doppler radars. Basically, this system identifies motion by examining heat, sound waves, visual changes, and frequency shifts.

The mechanical way is simple and used for basic applications; electronic systems, alternatively, offer more precision and flexibility so that they can be combined with modern technologies. Nevertheless, there are a few problems of optimization for the motion detection system; these include false positives, sensitivity calibration, and energy efficiency (Reolink, 2024).

1.1 Current Scenario

Motion detection systems have been game-changing models for home security. The current scenario for motion detection systems is characterized by a growing requirement for efficient and reliable security solutions that can easily integrate themselves with other devices and systems.

In today's world, dealing with theft, robbery, destruction of homely material, and human impaction has always been a topic that has never stopped.

Personal security and belongings safety have been huge needs for today's generation due to hateful and negative mindsets, so to tackle this problem, a few solutions and measures can be taken. One of them is the creation of our own "Motion Detection System".

It helps in securing our liable resources, aiming for intruder detection, being surveillance systems, and rooting for smart home automations. Alerting the owner/user about the possible security threat/moment in the home environment through digital display is one of the main problem solvers for current scenarios. Possible security measures following precaution using this system could result in way safer scenario. However, some challenges are to come in between, such as false alarms due to environmental factors/incorrect sensor calibration, limited range accuracy, and high maintenance which is a small price to pay for care-free home security. These tackles can surely be solved with more effort and innovation given to the system.

1.2 Problem Statement and Project as a solution

1.2.1 Problem Statement

Many small companies and personal spaces need a simple security system. Many alarms emit loud noise, which may not be audible if the owner is not present. These systems are sometimes difficult to set up, costly, and unreliable due to false alarms or power outages. For budget-conscious people, these systems may not be a practical or affordable solution. The complexity of these systems can cause setup challenges and even if the system is installed successfully, it might be difficult to maintain. Many systems do not provide instant notifications to the owner, leading to delayed responses to potential security threats.

This project aims to develop a cheap and simple motion detection system that sends a text message to the user anytime the movement is detected, addressing the issue of real time notifications. It is designed to be simple to install, portable, and reliable, making it a practical security solution for small spaces. This project offers an effective and budget-friendly way to protect properties.

Working overview of system

1.2.2 Project Solution:

This project involves creating a straightforward yet highly functional motion detection system using accessible and cost-effective components. The primary hardware will include a basic motion sensor and a reliable microcontroller, ensuring ease of assembly and affordability. The system’s core functionality revolves around detecting motion in its environment and instantly notifying the user by sending a text message directly to their phone. This approach eliminates the need for loud alarms, making it discreet and suitable for various scenarios where subtlety is preferred.

The design will incorporate a robust backup power source to ensure uninterrupted operation, even during power outages. This feature enhances its reliability, making it ideal for both home and small-scale commercial security needs. The system's compact and unassuming design will allow it to be installed in various locations without drawing attention, blending seamlessly into its surroundings. By focusing on simplicity and practicality, this project provides an effective solution for real-time motion detection and alerts, tailored for individuals looking for a low-cost yet dependable security measure.

1.3 Aim and Objectives

1.3.1 Aim of the Project:

To build a low-cost and easy-to-use motion detection system that notifies users instantly when movement is detected, improving security for homes and small businesses.

1.3.2 Objectives of the Project:

1. **Low Cost:**

   Use inexpensive components to make the system affordable for everyone.

2. **Easy Setup:**

   Ensure the system is simple to install and use, even for non-technical users.

3. **Real-Time Alerts:**

   Provide instant notifications through text messages whenever motion is detected.

4. **Reduce False Alarms:**

   Minimize false alerts by adjusting the sensitivity of the motion sensor.

5. **Reliable Operation:**

   Include a battery backup so the system works even during power outages.

6. **Expandable:**

   Allow the system to be easily upgraded with more sensors or additional features if needed.

2 Background

2.1 System Overview

The “Motion Detection System” is designed as an “alarm” that uses LED lights. As soon as a motion is detected, the alarm is triggered immediately, preventing intruders from disabling it. It has PIR sensors that identify even the tiniest movements around its radius. When the motion is detected, the system gets turned on, and the motion detector will begin monitoring the area continuously for any signs of movement. Such sensors have been put in place purposefully so that they can cover the house area; hence, even slight movements are contained.

As a trigger for the central processing unit of the system or computer, a signal is relayed instantly by the motion detector the moment it picks up on any movement. Through an application, the system may instantly send detected movement messages to the user’s display. The motion detector, therefore, remains at the heart of the system, providing a reliable way of sensing the presence of an intruder in order to make timely interventions that avert potential theft.

2.2 Design Diagram

2.2.1 Block Diagram

The block diagram below represents the flow of information and control through the Arduino-based motion detection system. The Arduino is the main component that receives data from the PIR sensor and gets power from the laptop via a USB cable. The Arduino processes the input signals and sends output to connected devices through a breadboard. After processing the data, it sends the instructions to the LED, buzzer and LCD screen.

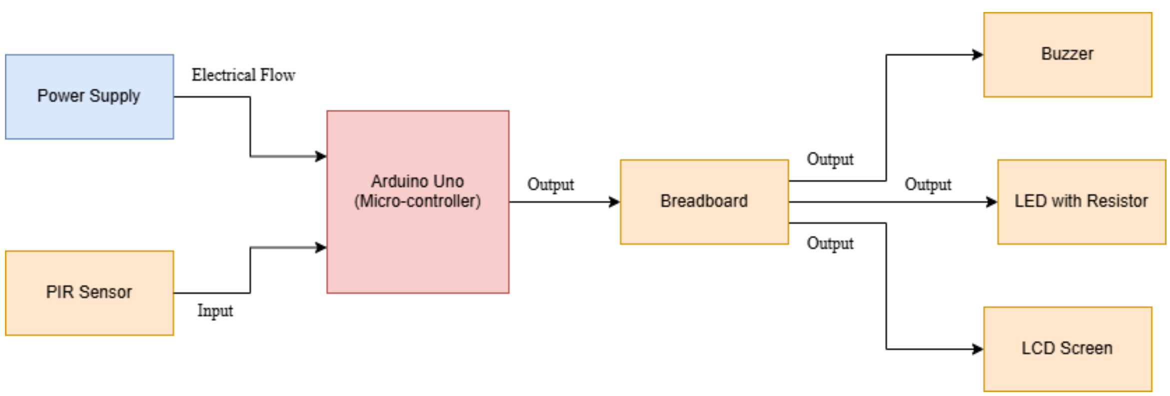


Figure 1: Block Diagram.

2.2.2 Hardware Architecture

The hardware architecture explains how different parts of the motion detection system work together. It shows the components and how they interact to make the system function properly.

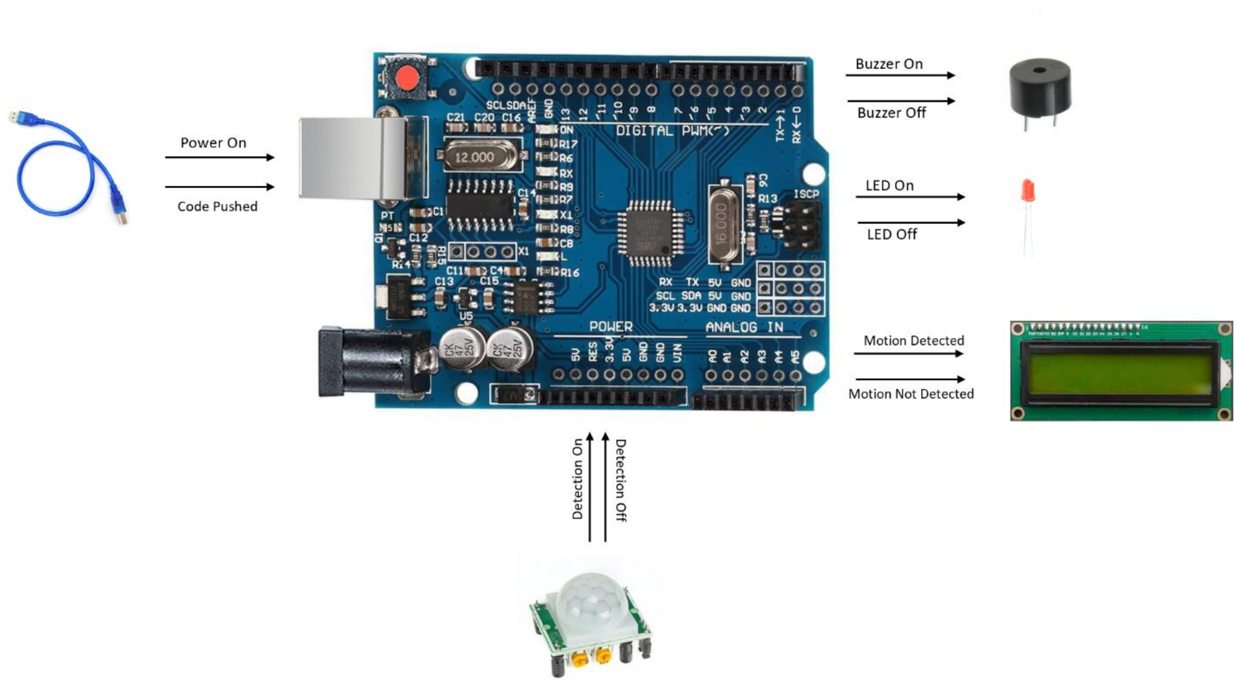


Figure 2: Hardware Architecture

2.2.3 Circuit Diagram

The circuit diagram below illustrates the physical wiring for the motion detection system. The motion detection is done by using a PIR sensor, which is connected to the Arduino Uno. When a motion is detected, the buzzer emits sound, the LED glow and the LCD screen displays a message “Motion Detected”.

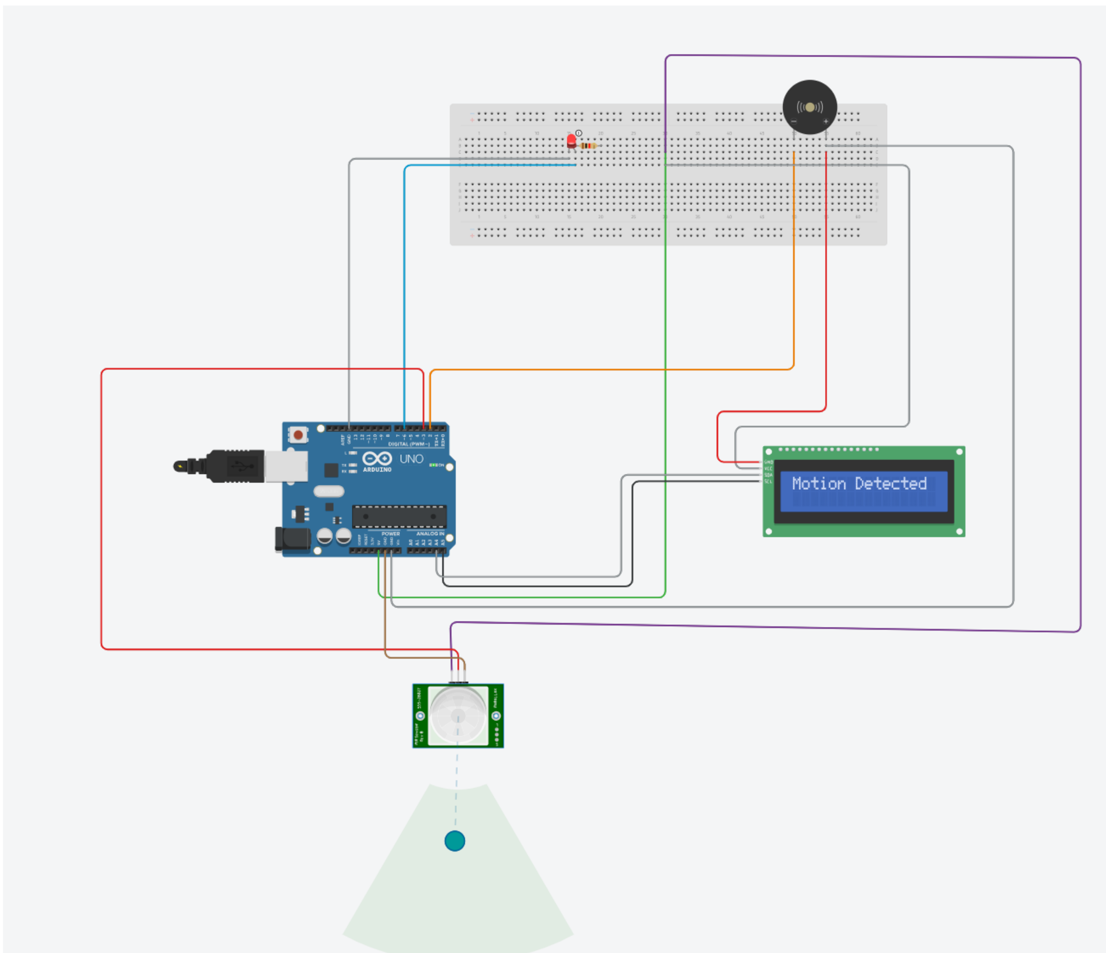


Figure 3: Circuit Diagram

2.2.4 Schematic Diagram

The schematic diagram below illustrates the logical arrangement of components like the Arduino Uno, PIR sensor, LED with resistor, buzzer and LCD screen. The system detects motion and activates the outputs showing the flow of signals, power and ground.

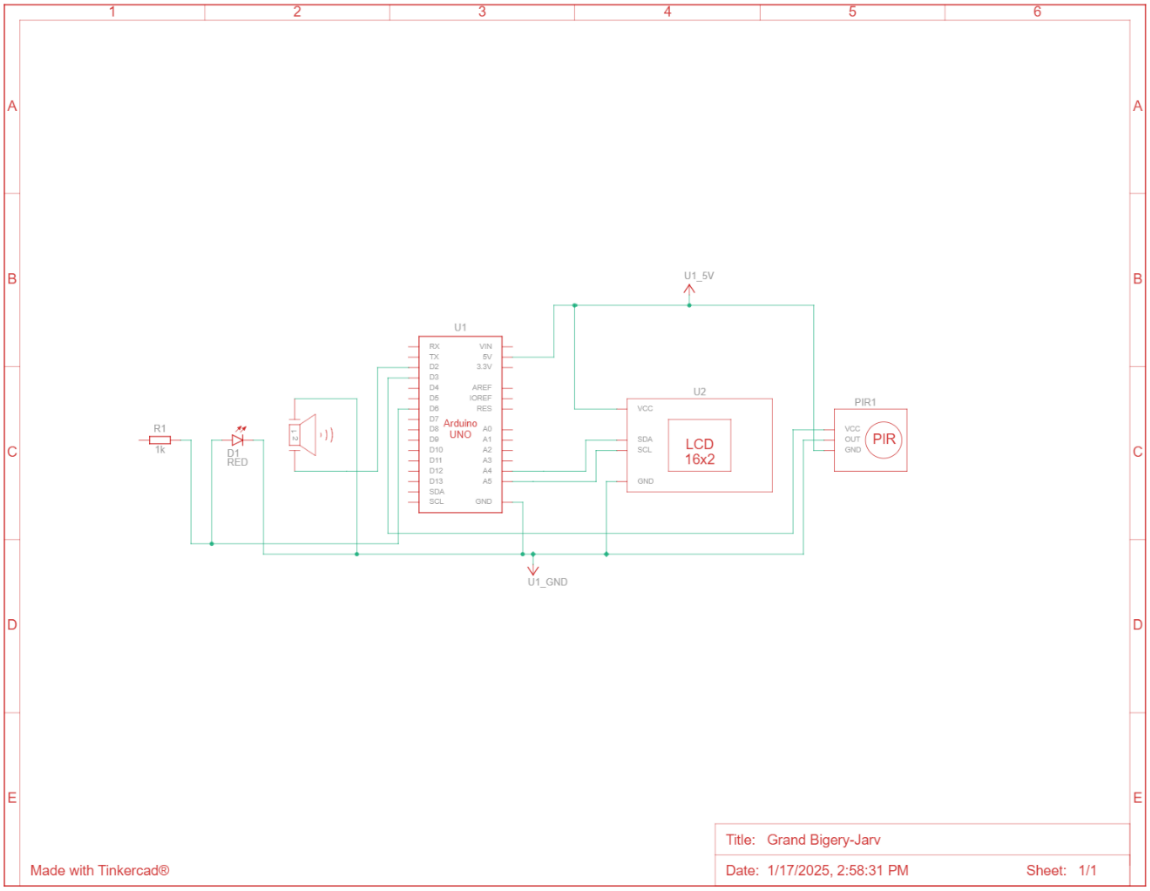


Figure 4: Schematic Diagram.

2.2.5 Flow Chart

The flowchart below illustrates the installation of the motion sensor and parts. When the motion is detected, it lights the LED, buzzer and LCD screen display “Motion Detected” message, waits for one second, and then turns off the buzzer and LED. If no motion is detected, the system turns off the LED and buzzer and displays “No Motion Detected” to the LCD screen.

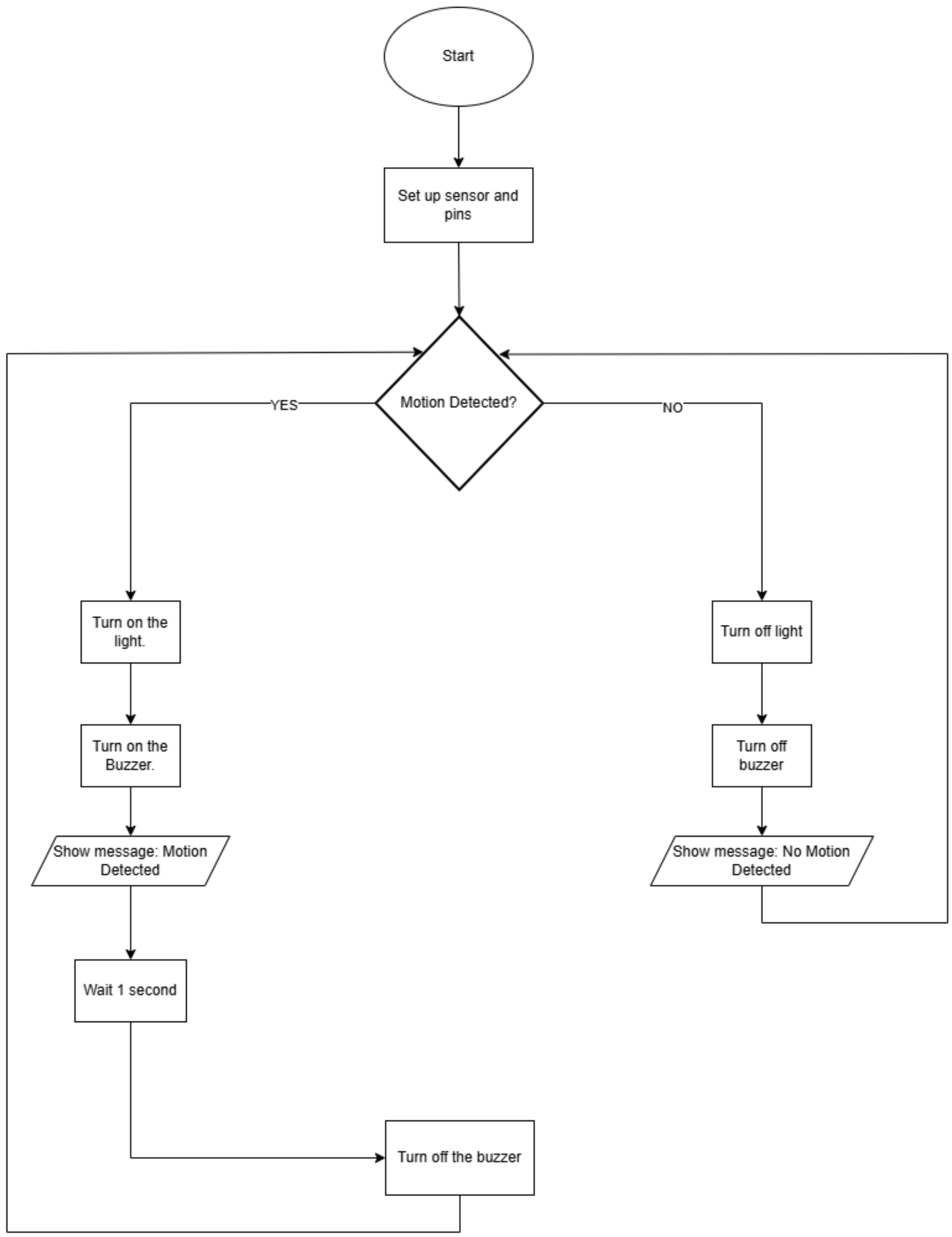


Figure 5: Flowchart showing simple flow.

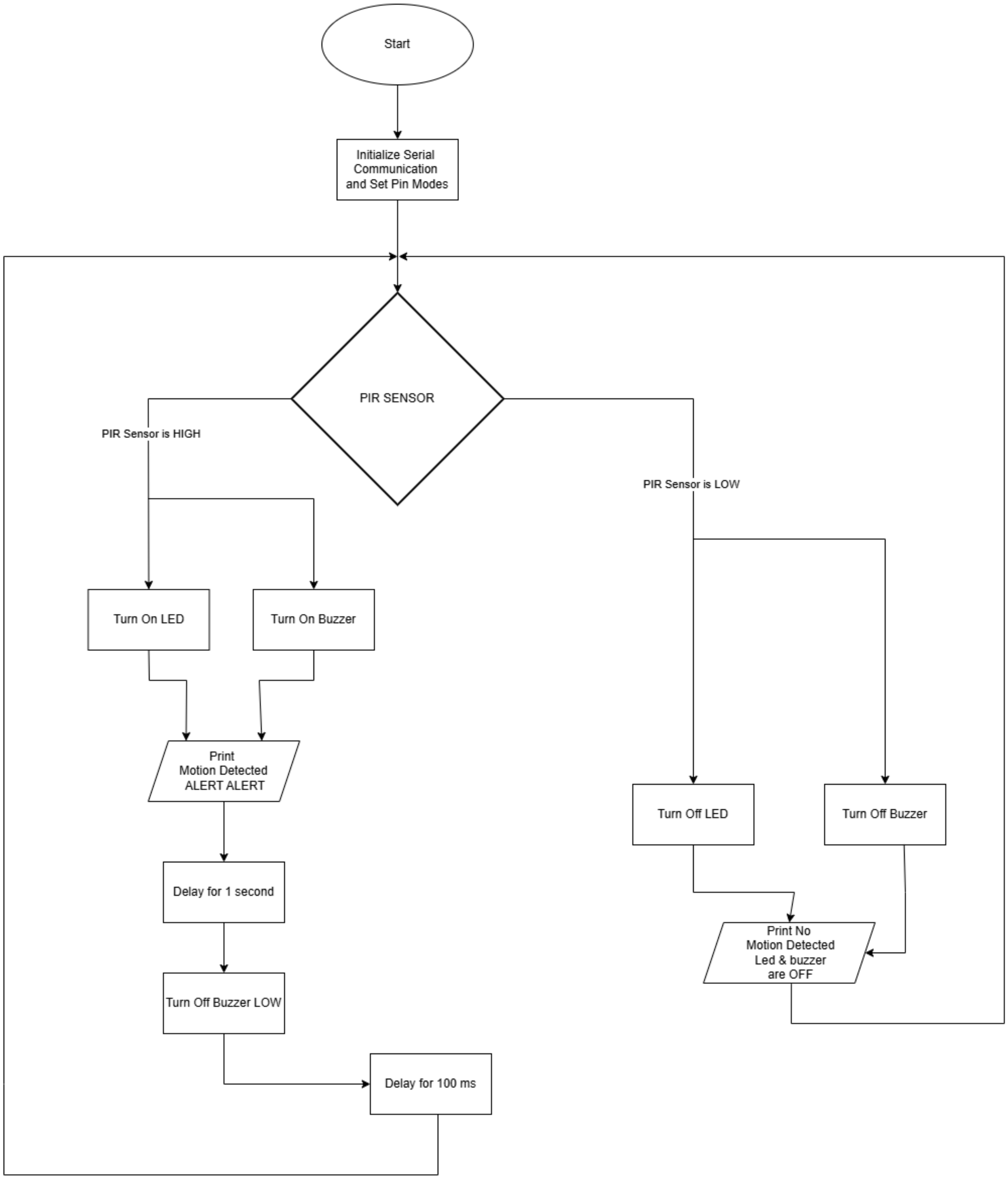


Figure 6: Flowchart showing code flow.

2.3 Requirement Analysis

2.3.1 Hardware Components:

2.3.1.1 Arduino Uno

The Arduino UNO is an affordable and user-friendly microcontroller board that works well for many electronic projects. It can be used by beginners to build prototypes without needing prior knowledge of electronics or coding. Arduino can take input signals from sensors and use them to control devices like motors, lights, or even connect to the internet (Ashley, 2021).

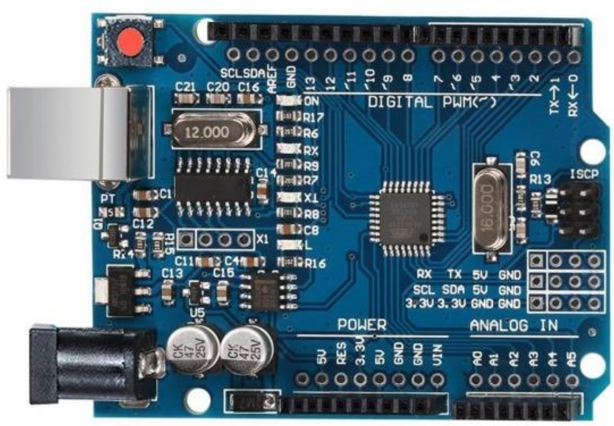


Figure 7: Figure showing Arduino Uno.

2.3.1.2 Breadboard

A breadboard is a tool used to connect temporary electronic circuits, which can easily add, remove, or change components. It is made of a plastic base with small holes that hold wires, components, or pins that can build and test different circuits (ScienceDirect, 2022).

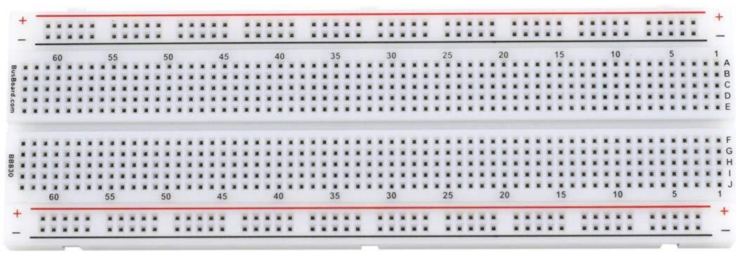


Figure 8: Figure showing breadboard.

2.3.1.3 Buzzer

A buzzer is an electronic device that produces sound when it receives an electrical signal. It is used to give audible alerts or notifications and is suitable for various electronic projects and systems. It requires a power source to operate. The sound's volume can be adjusted by changing the current it receives. (picobricks, 2023)

Figure 9: Figure showing buzzer.

2.3.1.4 Resistor (100-ohm)

A resistor is an electronic component that controls the flow of electricity in a circuit. It reduces the current to protect other components. It is essential for keeping circuits stable in all kinds of devices. They are essential for managing power and ensuring circuits function safely and efficiently (TechTarget, 2021).

Figure 10: Figure showing resistor.

2.3.1.5 PIR Sensor

A PIR (Passive Infrared) sensor is an electronic sensor used in motion detectors. It detects infrared light from objects in its range. It senses infrared radiation released or reflected by objects. They detect motion but don't identify what is moving but just sense the movement from people, animals, or other objects (Saha, 2022).

Figure 11: Figure showing PIR sensor.

2.3.1.6 LCD Screen 16x2

LCD is widely used screens in phones, calculators, TVs, etc. LCD screen are cheap and can display custom characters and animation. It consists of four ports: GND connects to ground, VCC connects to the power source, SDA transmits data and SCL synchronizes data transfer (EL-PRO-CUS, n.d.).

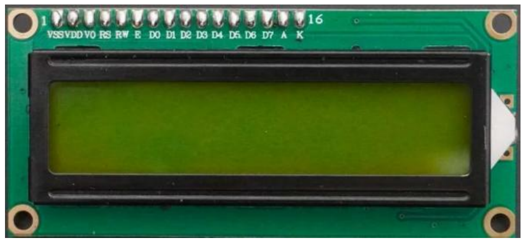


Figure 12: Figure showing LCD screen.

2.3.1.7 LED (generic)

A light-emitting diode (LED) is a commonly used, small, and efficient light source in many electronic devices. It works by producing light when electricity flows through a special material inside it. It is a semiconductor device that emits light when an electric current flow through it. It’s widely used in many applications due to its efficiency, longevity, and compact size. (BYJU'S, 2024)

Figure 13: Figure showing LED.

2.3.1.8 Jumper Wires

Jumper wires are wires with pins on both ends. It is used to connect points in a circuit without soldering. They come in different colors, but the colors do not have any specific meanings. They are commonly used with a breadboard to make circuits easy to adjust (Hemmings, 2018).

Figure 14: Figure showing jumper wires.

2.3.2 Software Components:

2.3.2.1 Arduino IDE

The Arduino IDE is free software used to write and upload code to Arduino boards. It works on Windows, Mac, and Linux and supports programming in C and C++. Code written in the IDE, called a sketch, is converted into a hex file and uploaded to the board's controller. The IDE has two main parts: an editor for writing code and a compiler for checking and uploading code to the Arduino board (adnanaqueel, 2018).

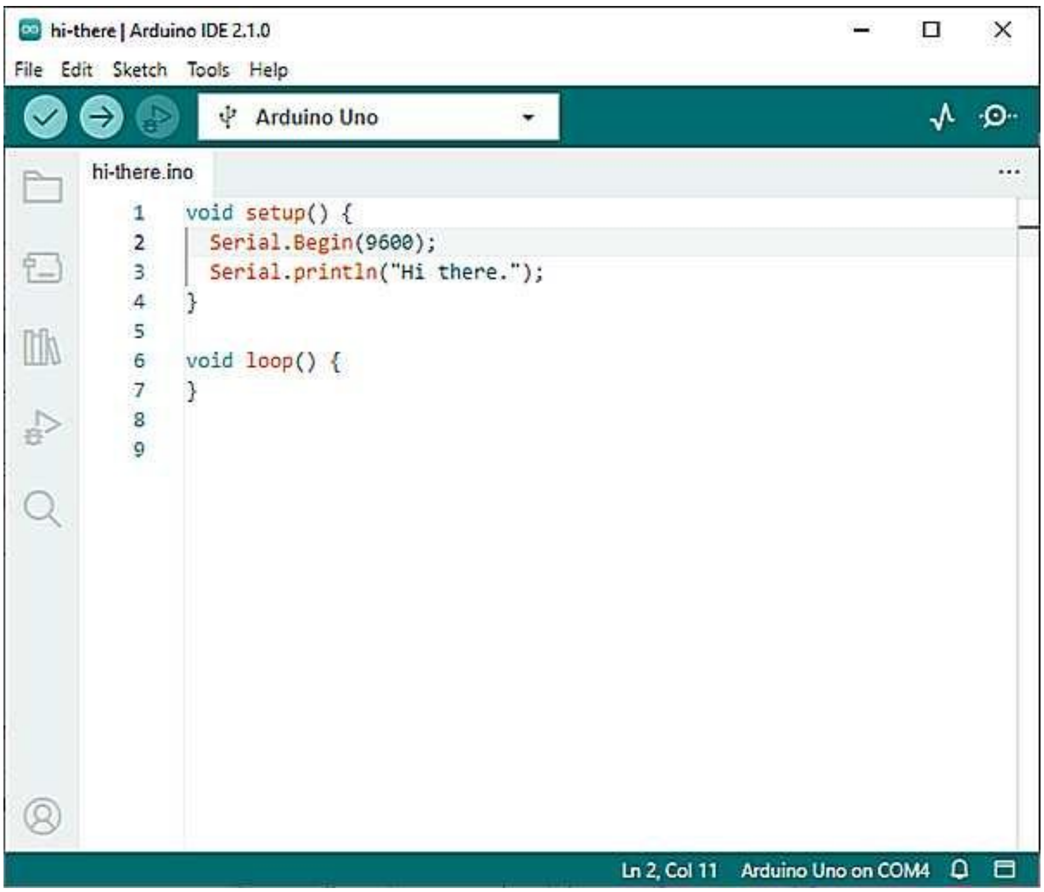


Figure 15: Figure showing Arduino IDE.

2.3.2.2 ThingSpeak

ThingSpeak is a platform for IoT analytics that lets us collect, view, and analyze live data in the cloud. It is great for testing and developing IoT systems with analytics. It connects devices easily using popular IoT protocols. It views sensor data in real-time and combines data from other sources when needed. It automates analytics with schedules or events (ThingSpeak, 2024).

Figure 16: Figure showing ThingSpeak.

2.3.2.3 Tinkercad

Tinkercad is a free, simple, and user-friendly online 3D modeling tool for learning basic skills in 3D printing or electronics. Its main features are 3D design, visual blocks, and electronics, making it a great learning tool for students and beginners (MakerIndustry, 2022).

Figure 17: Figure showing Tinkercad.

3 Development

3.1 Planning & Design

When the coursework was given, we had a meeting where we discussed which IoT project to do. After searching on the internet, we came across the “Motion Detection System”, which was applicable for the real-world.

We develop a “Motion Detection System” that sends a message to the user anytime the movement is detected, addressing the issue of real-time notification without the need of loud alarms.

The system is designed as a “silent alarm” that uses LED lights. As soon as a motion is detected, the alarm is triggered immediately, preventing intruders from disabling it.

3.2 Resource Collection

To collect the necessary resources for the project, an application was written to the Resource Department of the Islington College, requesting components required to make the “Motion Detection System”. The application was submitted to Mr. Shishir Subedi Sir. After receiving the application, the department provided components including Arduino, breadboard, LED, buzzer, jumper wires, resistor, and PIR sensor. With the help of the Resource Department, we were able to collect the required components to develop the project.

3.3 System Development

3.3.1 Phase 1: Powering the Arduino

The first step in the system development process is to power the Arduino. This is accomplished by connecting the Arduino board to a laptop using a USB power cable. The USB connection serves the dual purpose of providing power to the Arduino and enabling communication between the Arduino and the laptop.

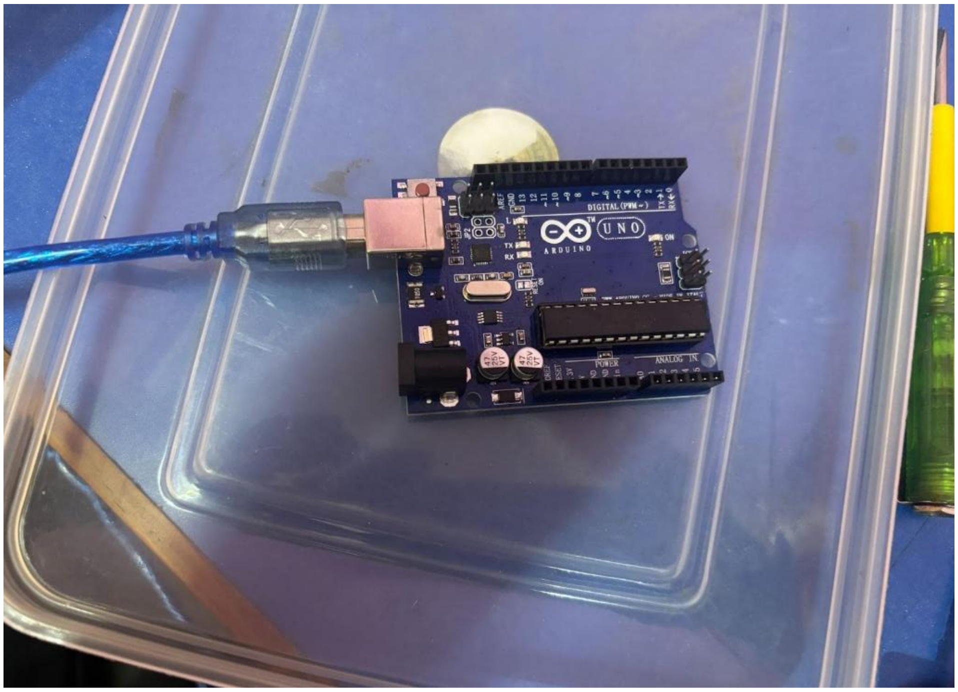


Figure 18: Powering the Arduino.

3.3.2 Phase 2: Connecting the PIR Sensor

After successfully powering the Arduino, the next step is to connect the PIR (Passive Infrared) sensor to the Arduino using jumper wires. The connections are made as follows:

- GND (Ground) of the PIR sensor is connected to the 5V port on the Arduino.
- Power pin of the PIR sensor is connected to Port 3 on the Arduino.
- Supply pin of the PIR sensor is connected to GND on the Arduino.

This setup allows the PIR sensor to detect motion by sensing infrared radiation changes in its environment.

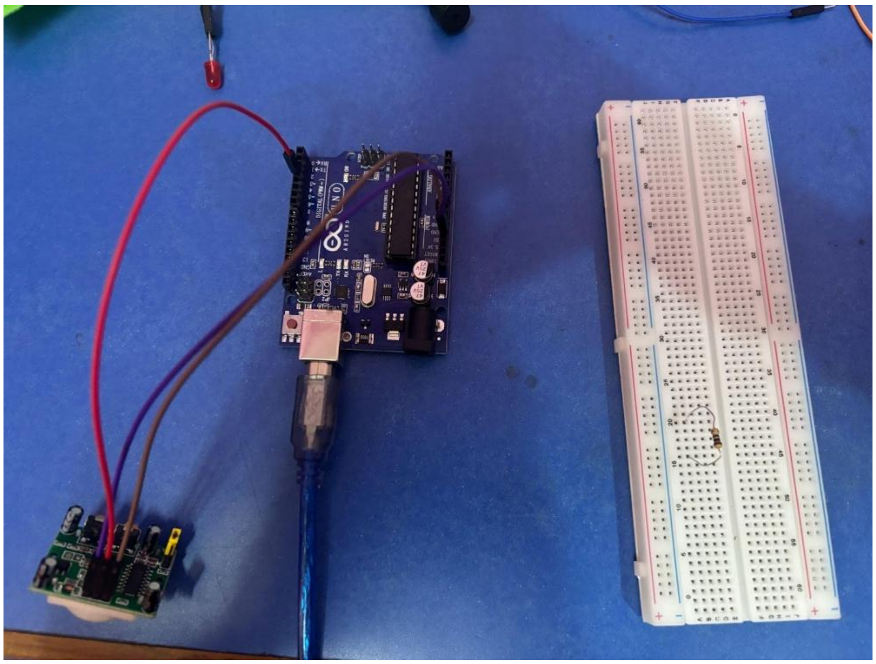


Figure 19: Connecting the PIR sensor.

3.3.3 Phase 3: Connecting the LED and Resistor

With the PIR sensor in place, the next component to connect is the LED and a resistor. This step ensures that the LED will light up when the system detects motion. The connections are as follows:

- The LED is connected to C20 on the breadboard.
- A Resistor is connected to B21 on the breadboard.

Using a resistor with the LED prevents excessive current from flowing through the LED, which could otherwise damage it.

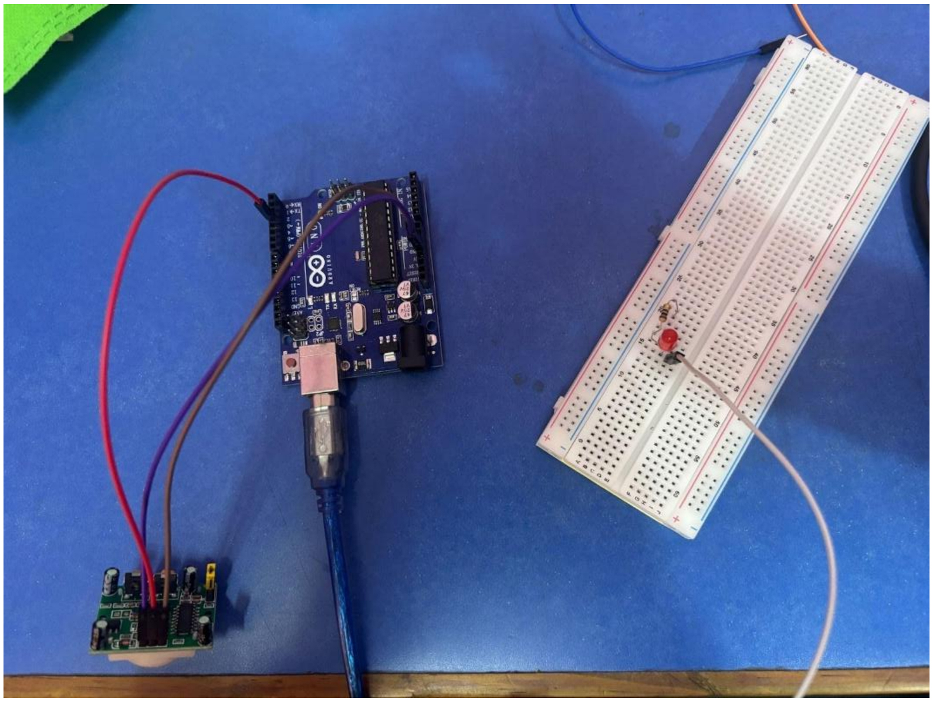


Figure 20: Connecting the LED and Resistor.

3.3.4 Phase 4: Powering the LED

After setting up the LED and resistor, the next step is to power the LED. The connections for this phase are:

- LED (cathode, D20) is connected to GND.
- Another connection from E21 on the breadboard is made to Port -6.

This setup ensures that the LED will illuminate when triggered by the PIR

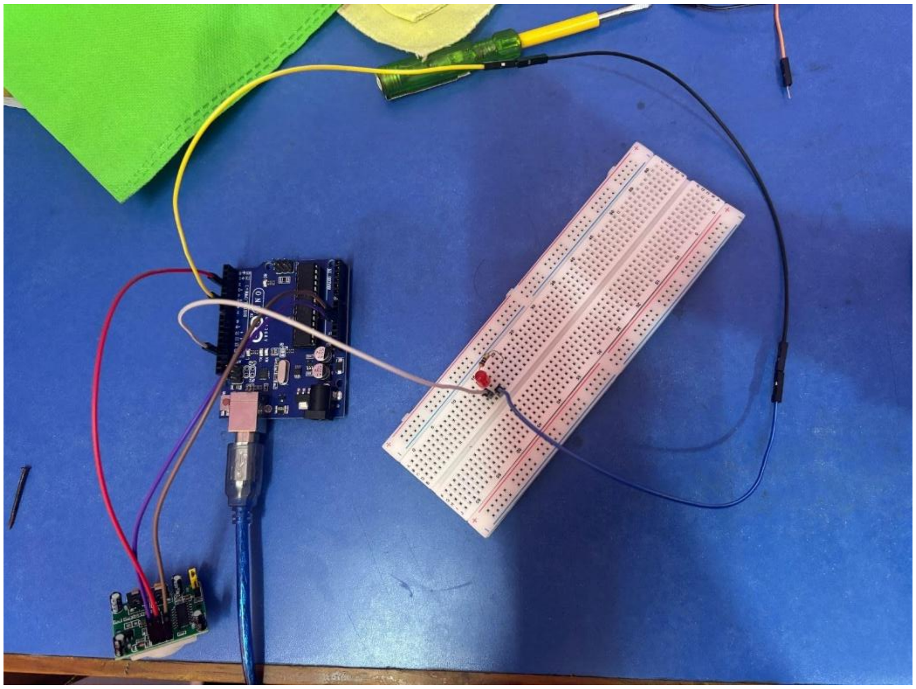


Figure 21: Powering the LED.

3.3.5 Phase 5: Connecting the Buzzer

Next, a buzzer is added to the system to provide an audible alert when motion is detected. The buzzer is connected as follows:

- The Positive Terminal of the buzzer is connected to J3 on the breadboard, which is linked to Port 2 on the Arduino.
- The Negative Terminal of the buzzer is connected to J6 on the breadboard.

This setup allows the buzzer to sound an alert when the PIR sensor detects motion.

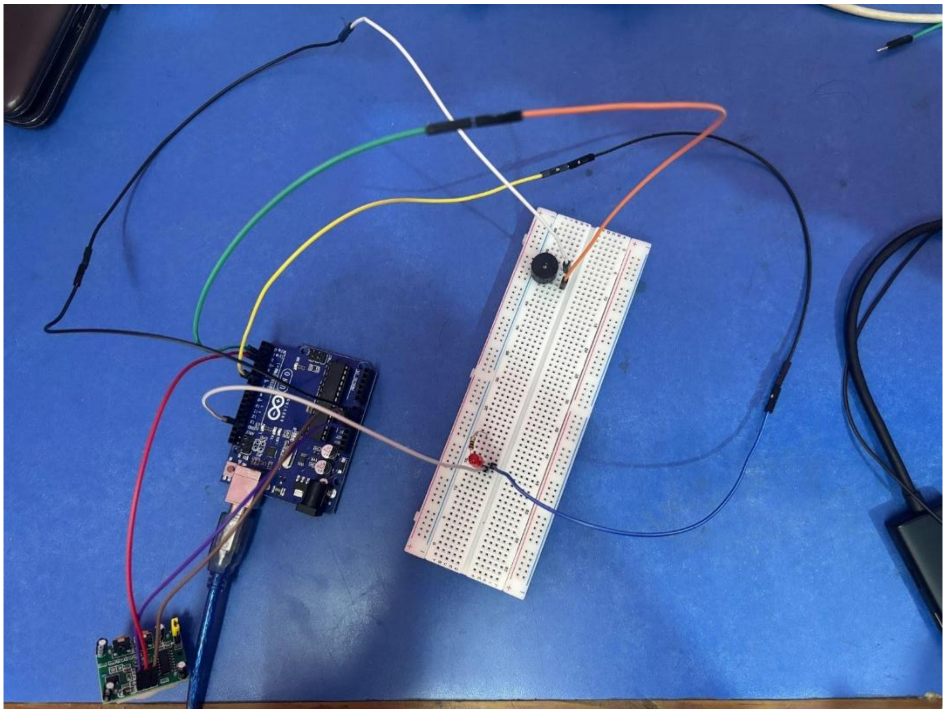


Figure 22: Connecting the Buzzer.

3.3.6 Phase 6: Final Power Connection

To finalize the setup, the Arduino is reconnected to the laptop using the USB power cable. This connection not only provides power but also allows for uploading the necessary code to the Arduino to control the system.

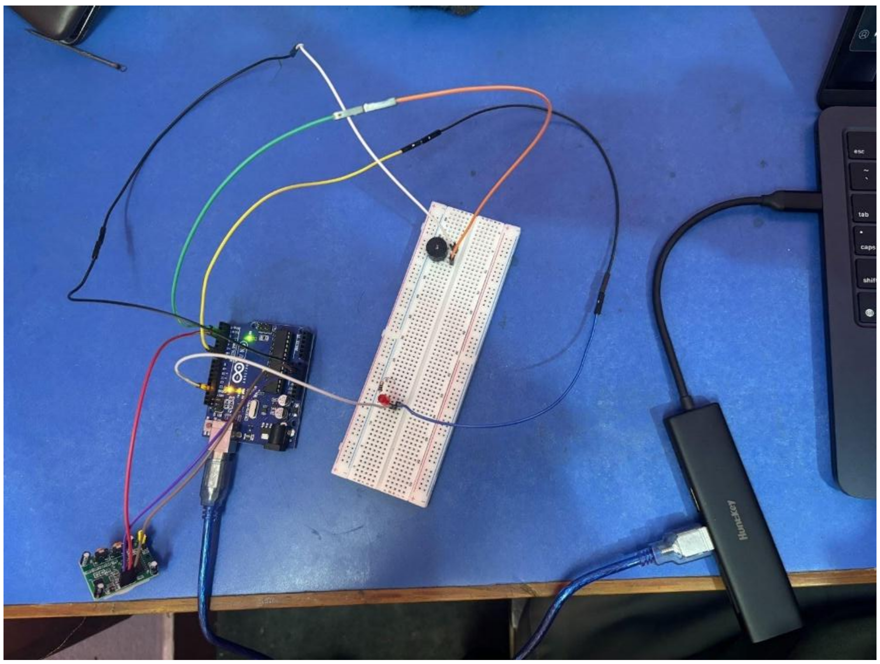


Figure 23: Final Power Connection.

3.3.7 Phase 7: Testing the System

With all components connected and powered, the final phase is to test the system. This involves triggering the PIR sensor with motion to verify its functionality. Upon detecting motion, the system should activate the LED and sound the buzzer, indicating that the setup is working correctly.

Through these phases, the system is successfully developed to detect motion and respond with both visual (LED) and auditory (buzzer) alerts.

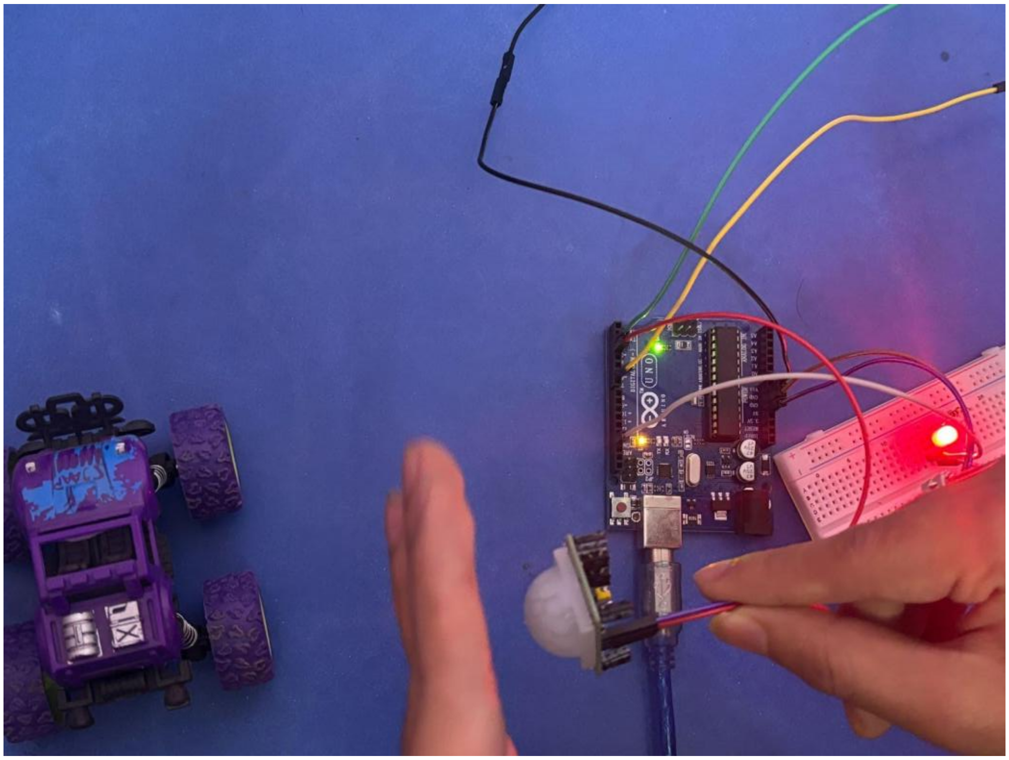


Figure 24: Testing the system.

4 Results and Findings

4.1 Results

By the end of our project on the **Motion Detection System**, we successfully built an intelligent system capable of detecting motion in real-time through advanced algorithms and sensor integrations. The system demonstrated the following capabilities:

- Spot motion accurately, regardless of varying lighting conditions, including low-light and bright environments.
- Trigger instant alerts whenever unexpected movements occur ensuring timely responses.
- Provide detailed and live data visualizations to represent detected motion patterns.
- Seamlessly integrate with external hardware devices like alarm systems.
- Send real time notification instantly to the owner or any connected users when motion is detected, especially in critical areas like doors and entry points.

Overall, the final system not only met but, in several aspects, exceeded our initial objectives. It showcased high reliability, efficiency and responsiveness across different testing scenarios and operational conditions.

4.2 Findings

To ensure the Motion Detection System functions correctly, we conducted multiple test cases. Below are some key scenarios and their results:

4.2.1 Test 1: To test motion detection in low light.

Table 1: Test 1

Test No: 1	Motion detection in low light
Objective	Verifying that the system detects motion in low light scenarios.
Action	Simulated low light environment with the moving objects.
Expected Result	Motion detected and alert, SMS is triggered.
Actual Result	Motion was detected successfully and alert message triggered
Conclusion	The test was successful.

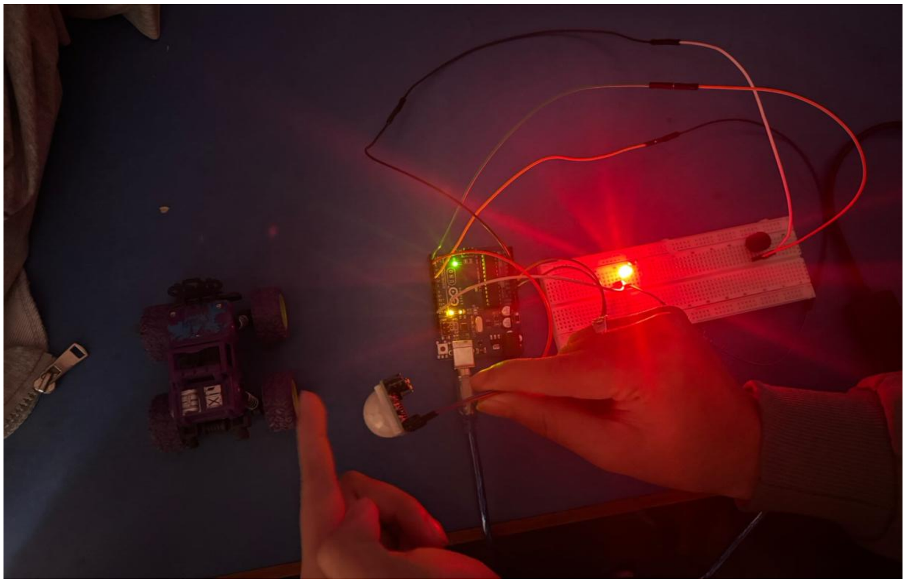


Figure 25: Motion detection in low light.

4.2.2 Test 2: To test multiple object detection.

Table 2: Test 2

Test No: 2	Multiple Object Detection
Objective	Ensuring the system detects multiple moving objects simultaneously.
Action	Multiple moving objects within the detection zone.
Expected Result	All the objects detected with in the detection zone and alerts triggered.
Actual Result	Multiple objects were detected without any errors.
Conclusion	The test was successful.

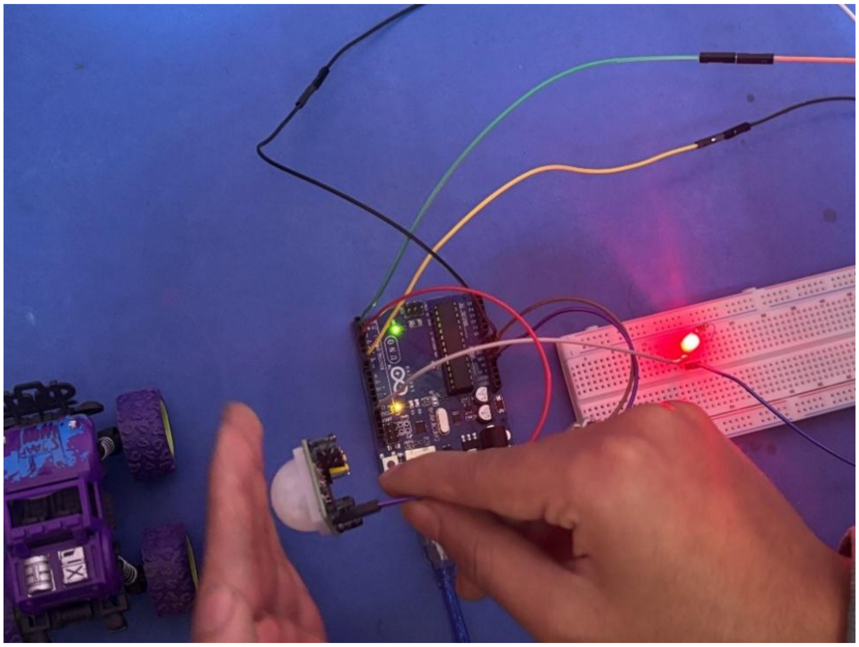


Figure 26: Multiple object detection.

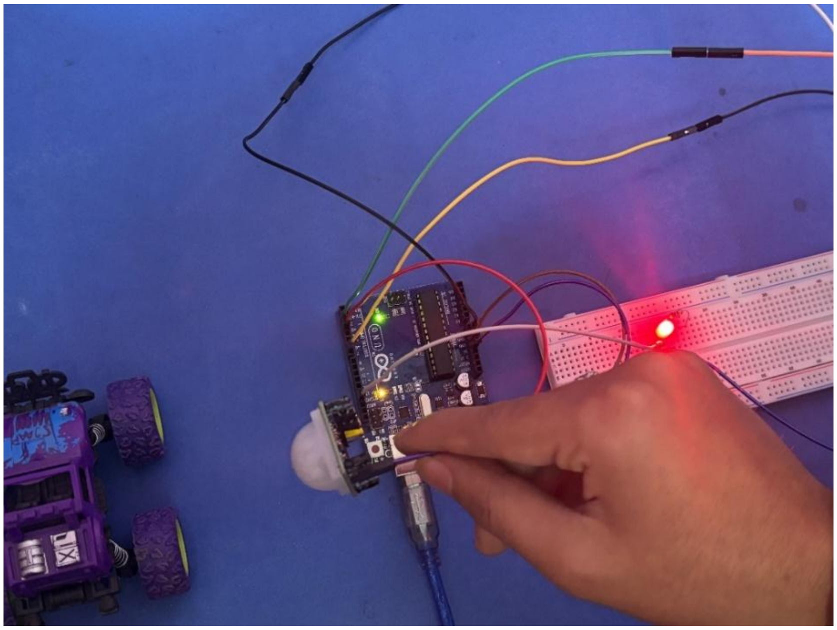


Figure 27: Vehicle detection.

4.2.3 Test 3: To test motion detected at door.

Table 3: Test 3

Test No: 3	Motion detected at door with the notification
Objective	Verifying if notification is sent when motion is detected at door entry points.
Action	Simulated door motion activity.
Expected Result	Notification delivered to the owner or connected user.
Actual Result	Notification was sent successfully.
Conclusion	The test was successful.

4.2.4 Test 4: To test delay in code for accurate PIR sensor reading.

Table 4: Test 4

Test No: 4	Delay in code for accurate PIR sensor reading
Problem	Inaccurate in PIR sensor readings without delay.
Solution	Added delay of 100ms in the code.
Expected Result	Accurate detection after delay is added.
Actual Result	PIR sensor reading is stabilized with delay.
Conclusion	Passed.

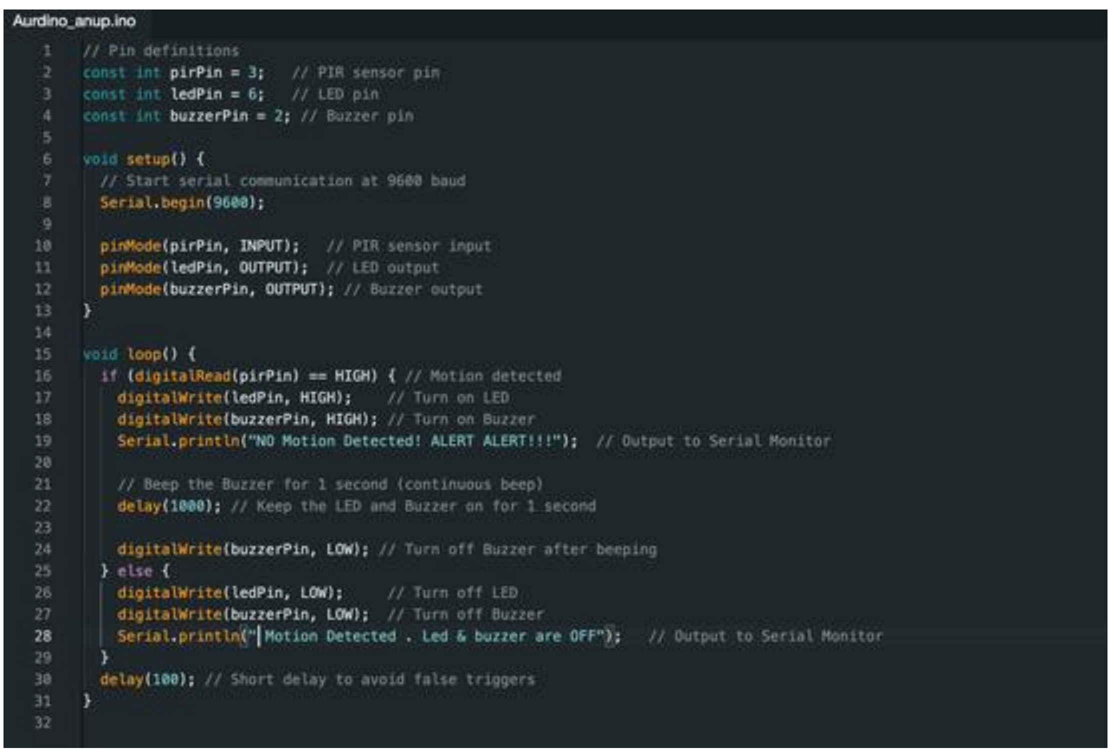


Figure 28: Inaccurate reading without delay.

Aurdino_anup.ino

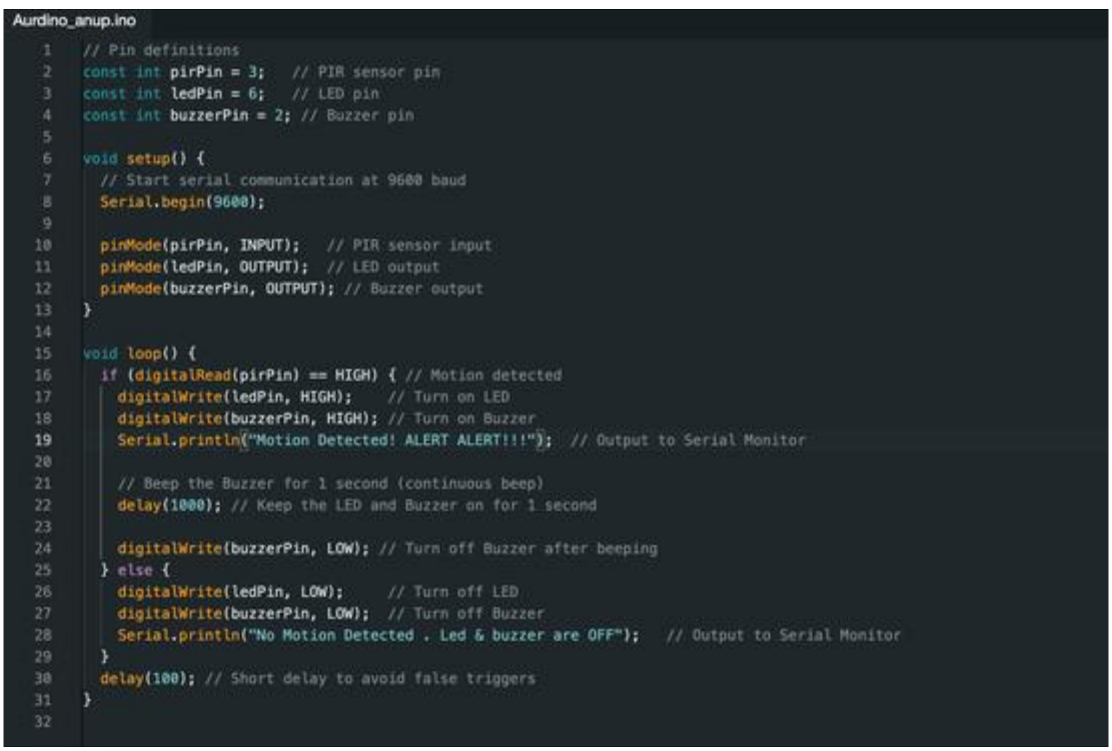
```
// Pin definitions
const int pirPin = 3;   // PIR sensor pin
const int ledPin = 6;   // LED pin
const int buzzerPin = 2; // Buzzer pin

void setup() {
  // Start serial communication at 9600 baud
  Serial.begin(9600);

  pinMode(pirPin, INPUT);   // PIR sensor input
  pinMode(ledPin, OUTPUT);  // LED output
  pinMode(buzzerPin, OUTPUT); // Buzzer output
}

void loop() {
  if (digitalRead(pirPin) == HIGH) { // Motion detected
    digitalWrite(ledPin, HIGH);    // Turn on LED
    digitalWrite(buzzerPin, HIGH); // Turn on Buzzer
    Serial.println("Motion Detected! ALERT ALERT!!!");  // Output to Serial Monitor

    // Beep the Buzzer for 1 second (continuous beep)
    delay(1000); // Keep the LED and Buzzer on for 1 second

    digitalWrite(buzzerPin, LOW); // Turn off Buzzer after beeping
  } else {
    digitalWrite(ledPin, LOW);     // Turn off LED
    digitalWrite(buzzerPin, LOW);  // Turn off Buzzer
    Serial.println("No Motion Detected . Led & buzzer are OFF");   // Output to Serial Monitor
  }

}
```

Figure 29: Delay added.

4.2.5 Test 5: To test output mismatch issue.

Table 5: Test 5

Test No: 5	Output Mismatch issue
Problem	Incorrect output assignment.
Solution	Correct output value was assigned.
Expected Result	Accurate output is matching the expected result.
Actual Result	Output matched expected values
Conclusion	Passed.


Aurdino_anup.ino

```
// Pin definitions
const int pirPin = 3;   // PIR sensor pin
const int ledPin = 6;   // LED pin
const int buzzerPin = 2; // Buzzer pin

void setup() {
  // Start serial communication at 9600 baud
  Serial.begin(9600);

  pinMode(pirPin, INPUT);   // PIR sensor input
  pinMode(ledPin, OUTPUT);  // LED output
  pinMode(buzzerPin, OUTPUT); // Buzzer output
}

void loop() {
  if (digitalRead(pirPin) == HIGH) { // Motion detected
    digitalWrite(ledPin, HIGH);    // Turn on LED
    digitalWrite(buzzerPin, HIGH); // Turn on Buzzer
    Serial.println("Motion Detected! ALERT ALERT!!!");  // Output to Serial Monitor

    // Beep the Buzzer for 1 second (continuous beep)
    delay(1000); // Keep the LED and Buzzer on for 1 second

    digitalWrite(buzzerPin, LOW); // Turn off Buzzer after beeping
  } else {
    digitalWrite(ledPin, LOW);     // Turn off LED
    digitalWrite(buzzerPin, LOW);  // Turn off Buzzer
    Serial.println("No Motion Detected . Led & buzzer are OFF");   // Output to Serial Monitor
  }
  delay(100); // Short delay to avoid false triggers
}
```

Figure 30: Output mismatch issue.

4.2.6 Test 6: To test working code.

Table 6: Test 6

Test No: 6	Working Code
Objective	Verifying if the code is functioning correctly with the proper delay. Pin value and terminal connections.
Action	Complete code was executed with correct configurations.
Expected Result	Motion detected, alert triggered and LED turned on.
Actual Result	Motion detected, alert triggered and LED turned on.
Conclusion	The test was successful.

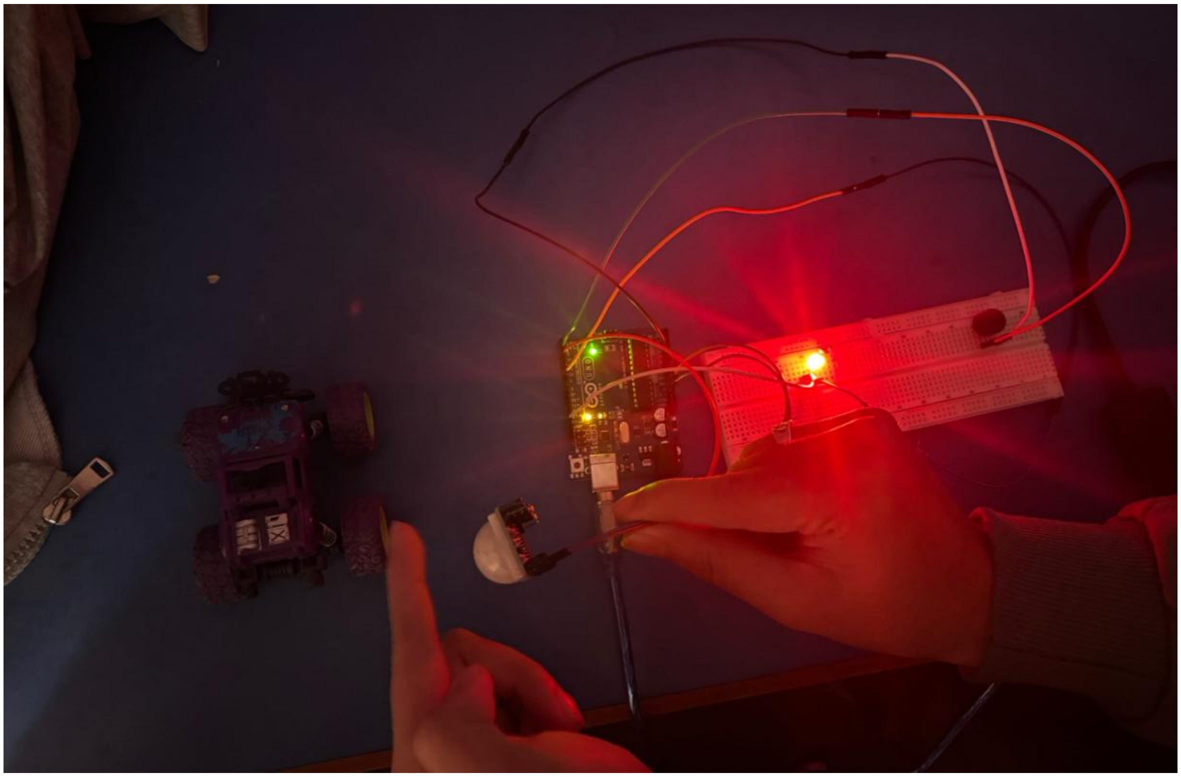


Figure 31: Working Code.

5 Future Works

Looking ahead, there is many improvements that can be added into the Motion Detection System making it smarter more reliable and even easier too use like:

5.1 Smarter Threat Detection:

- Goal: Enable the system to better distinguish between harmless movements (like pets, falling leaves) and actual threats.
- How: Train advanced AI models to recognize behaviour patterns more accurate, reducing unnecessary false alarms.

5.2 Cloud Storage:

- Goal: Let users securely store motion data and alerts online for access from anywhere and anytime.
- How: Integrate secure cloud storage solutions with strong encryptions protocols for protecting user data.

5.3 Mobile App Support:

- Goal: Allow users real time monitor and control capabilities directly from phones.
- How: Build simple to use mobile apps, both for Android and iOS, prioritizing seamless connection and user-friendly interfaces.

5.4 Backup Power:

- Goal: Ensure that system stays operational, even when power cuts happen.
- How: Use energy efficiency backup batteries along with smart power managing systems.

5.5 Weather Resistance:

- Goal: Make the system stronger against extreme weather conditions.
- How: Utilize weather proof materials, adaptive sensors for functioning well in rain, fog or intense heat.

6 Conclusion

The Motion Detection System project was a very successful demonstration in developing an inexpensive, very easy-to-use security solution. While working in real time, it can detect motions and raise alerts. This is a system that meets the need for enhanced security from homes and small businesses in an excellent manner, as it gives instant text message notifications and does not rely on loud alarms.

Since the core components for operation included a PIR sensor and an Arduino microcontroller, this system would enable accurate detection of motion with reliability. Although there have been incidences of false alarms and power backup problems, they have been overcome by careful calibration and design changes in turn leading to a robust, portable, and efficient system. It is simple to make, and hence, cheap so cheap that it most people can afford it. This thus meets the key objectives of low cost, ease of setup, and reliability.

Some of the other proposed improvements outlined for the system include future enhancements like AI-driven threat detection, cloud storage, mobile app support, and weatherproofing. As a whole, the project was able to meet its aim of bringing out a very workable and effective motion detection solution very promising in its potential for further development.

7 References

adnanaqueel, 2018. *Introduction to Arduino IDE.* [Online]
Available at: https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html
[Accessed 4 1 2025].

Ashley, E., 2021. *What is Arduino UNO? Arduino UNO Pinout & Program.* [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 13 12 2024].

BYJU'S, 2024. *Light Emitting Diode (LED).* [Online]
Available at: https://byjus.com/physics/light-emitting-diode/
[Accessed 11 12 2024].

Hemmings, M., 2018. *What is a Jumper Wire?.* [Online]
Available at: https://blog.sparkfuneducation.com/what-is-jumper-wire
[Accessed 4 1 2025].

MakerIndustry, 2022. *What Is Tinkercad? An Overview for Beginners.* [Online]
Available at: https://makerindustry.com/what-is-tinkercad/
[Accessed 4 1 2025].

picobricks, 2023. *What is Buzzer?.* [Online]
Available at: https://picobricks.com/blogs/info/what-is-buzzer
[Accessed 3 1 2024].

Saha, H. N., 2022. *A crop-monitoring system using wireless sensor networking.* [Online]
Available at: https://www.sciencedirect.com/topics/computer-science/passive-infrared-sensor#:~:text=A%20PIR%20sensor%20is%20an,in%20their%20field%20of%20view.
[Accessed 11 12 2024].

ScienceDirect, 2022. *Breadboard.* [Online]
Available at: https://www.sciencedirect.com/topics/engineering/breadboard
[Accessed 3 1 2025].

Smoot, R., 2024. *Push Button Switches 101.* [Online]
Available at: https://www.sameskydevices.com/blog/push-button-switches-101
[Accessed 3 1 2025].

TechTarget, 2021. *What is a Resistor?.* [Online]
Available at: https://www.techtarget.com/whatis/definition/resistor
[Accessed 3 1 2025].

ThingSpeak, 2024. *Learn More - ThingSpeak IoT - MathWorks.* [Online]
Available at: https://thingspeak.mathworks.com/pages/learn_more
[Accessed 4 1 2025].

8 Appendix

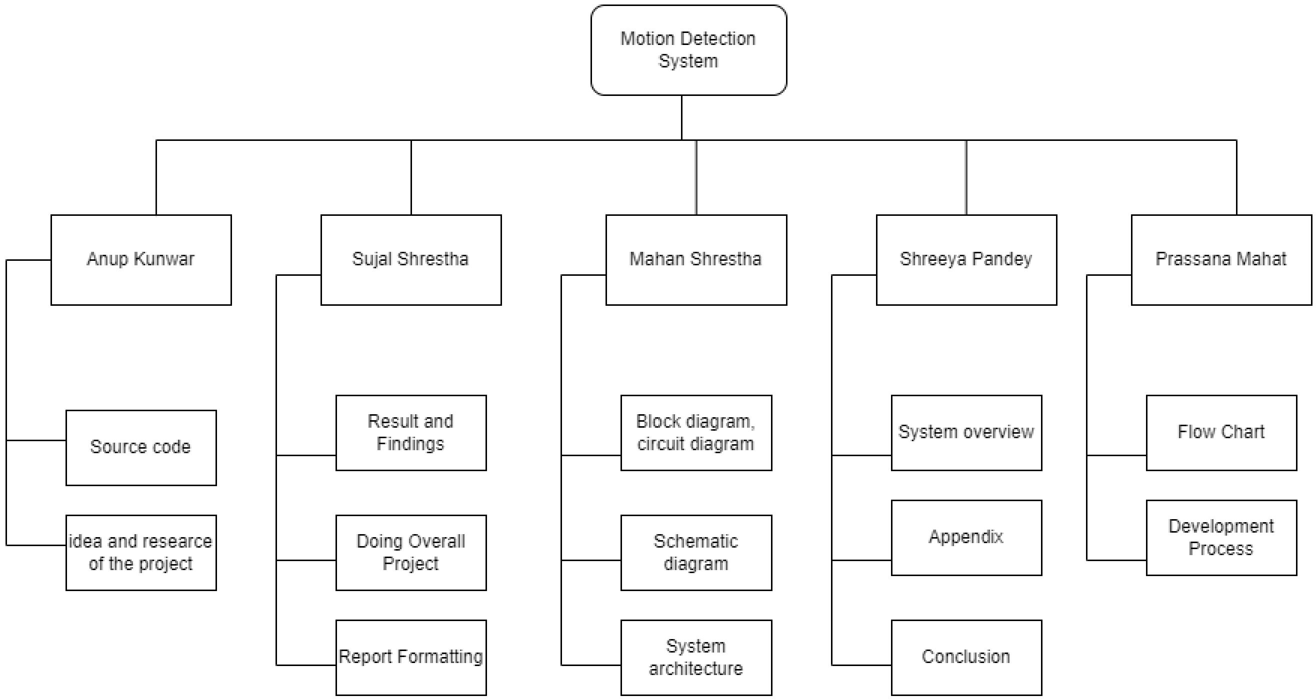
8.1 Individual Contribution Plan

Table 7: Individual Contribution Plan

WBS Level	Members	Task Description	Contribution
Requirements Analysis	Mahan Shrestha	Collecting and defining system needs	• Responsible for gathering system requirements and translating them into actionable tasks. • Creates initial workflow diagrams and identifies potential challenges in the project.
Hardware Integration	Sujal Shrestha	Selecting and configuring devices	• Chooses appropriate sensors and components. • Ensures all hardware is functional and compatible.
Software Development	Anup Kunwar Kamar	Writing and implementing code	• Designs and develops motion detection algorithms. • Tests software for accuracy and efficiency. • Project development & Analysis
	Prasanna Mahat	Validating and optimizing system	• Conducts performance tests and identifies bugs. • Suggests improvements based on test outcomes.

Documentation and Finalization	Shreeya Pandey	Compiling project reports and wrapping up	• Prepares system documentation • Finalizes deliverables for presentation.

8.2 Individual Contribution Structure



8.3 Source Code:

```
#include <LiquidCrystal.h>

#include <Wire.h>
#include "LiquidCrystal_I2C.h"

// Pin definitions
const int pirPin = 3; // PIR sensor pin
const int ledPin = 6; // LED pin
const int buzzerPin = 2; // Buzzer pin

// Initialize the LCD, set the I2C address to 0x27
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
// Start serial communication at 9600 baud
Serial.begin(9600);

// Initialize pins
pinMode(pirPin, INPUT); // PIR sensor input
pinMode(ledPin, OUTPUT); // LED output
pinMode(buzzerPin, OUTPUT); // Buzzer output

// Initialize the LCD
lcd.init(); // Initialize the LCD
lcd.backlight(); // Turn on the backlight
lcd.clear(); // Clear any existing text
lcd.setCursor(0, 0); // Set cursor to the first row
lcd.print("System Ready"); // Display a welcome message
delay(2000); // Display message for 2 seconds
lcd.clear(); // Clear the screen
}

void loop() {
if (digitalRead(pirPin) == HIGH) { // Motion detected
digitalWrite(ledPin, HIGH); // Turn on LED
digitalWrite(buzzerPin, HIGH); // Turn on Buzzer
Serial.println("Motion Detected! ALERT ALERT!!!"); // Output to Serial Monitor

// Update the LCD screen
lcd.setCursor(0, 0); // Set cursor to first row, first column
```

```
lcd.print("Motion Detected");
lcd.setCursor(0, 1); // Set cursor to second row, first column
lcd.print("Stay Alert!");

// Beep the Buzzer for 1 second (continuous beep)
delay(1000); // Keep the LED and Buzzer on for 1 second

digitalWrite(buzzerPin, LOW); // Turn off Buzzer after beeping
} else {
digitalWrite(ledPin, LOW); // Turn off LED
digitalWrite(buzzerPin, LOW); // Turn off Buzzer
Serial.println("No Motion Detected. Led & Buzzer are OFF"); // Output
to Serial Monitor

// Update the LCD screen
lcd.clear(); // Clear the LCD screen
lcd.setCursor(0, 0); // Set cursor to first row, first column
lcd.print("No Motion");
lcd.setCursor(0, 1); // Set cursor to second row, first column
lcd.print("System Idle");
}
delay(100); // Short delay to avoid false triggers
}
```

8.4 Screenshot of the System

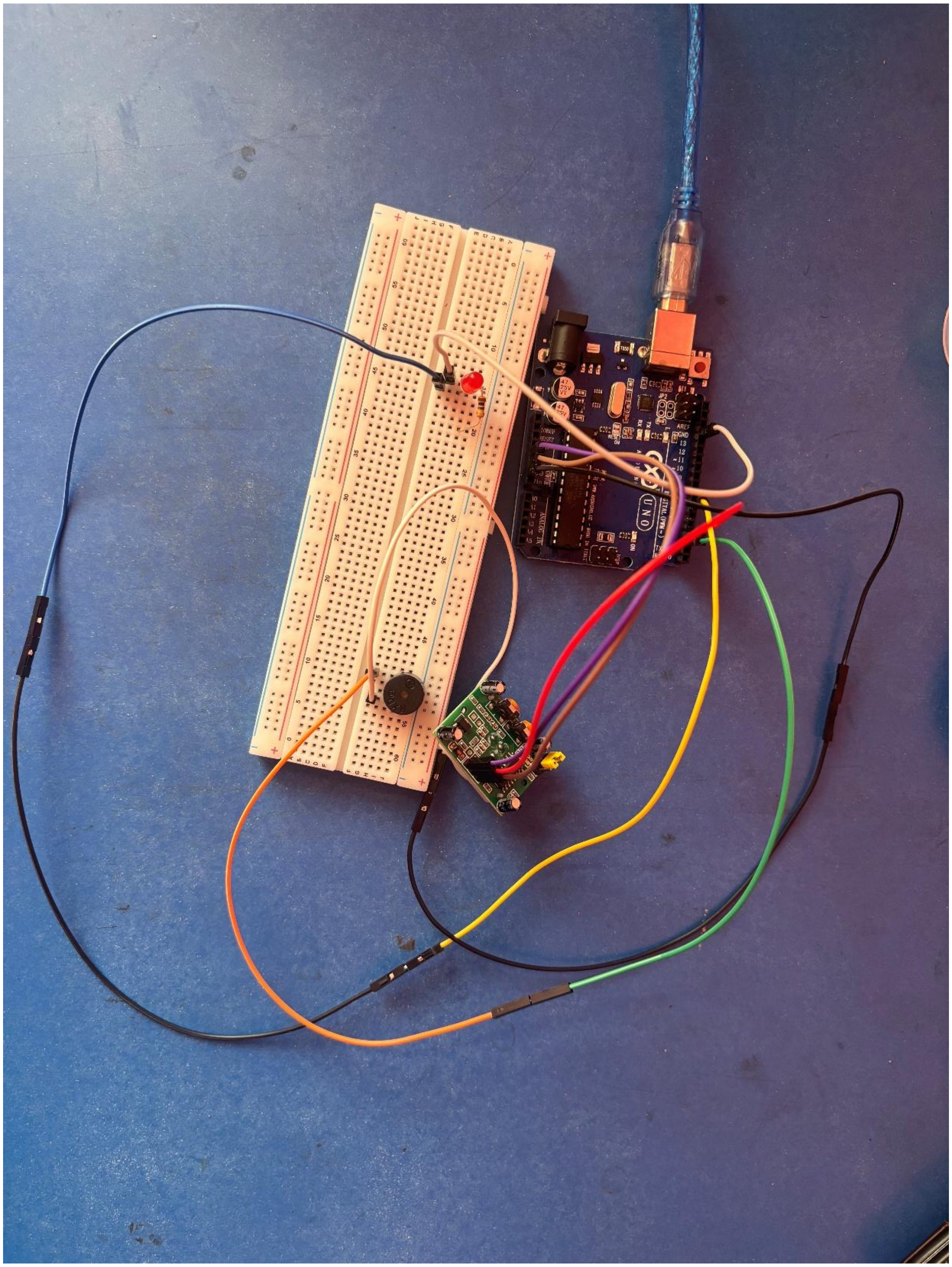


Figure 32: Screenshot of the System.

8.5 Design Diagram

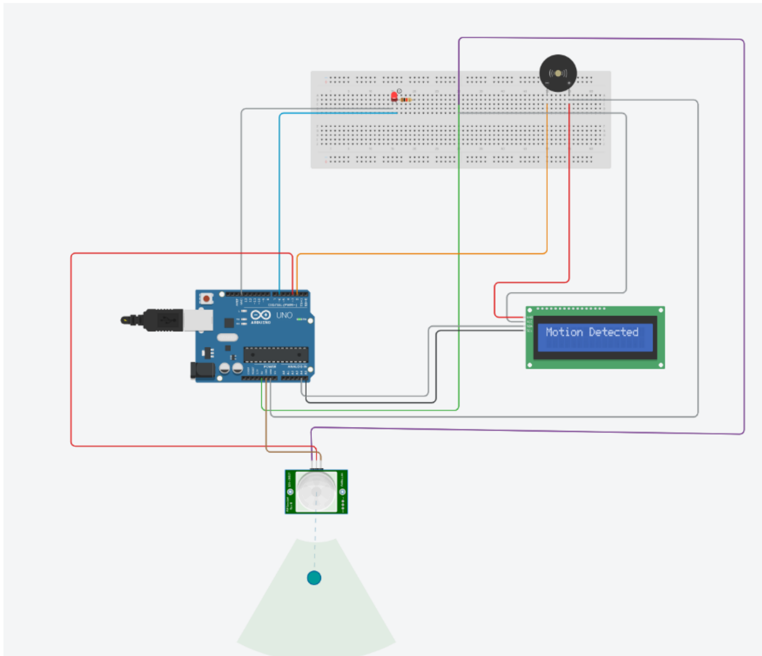


Figure 33: Hardware Architecture.

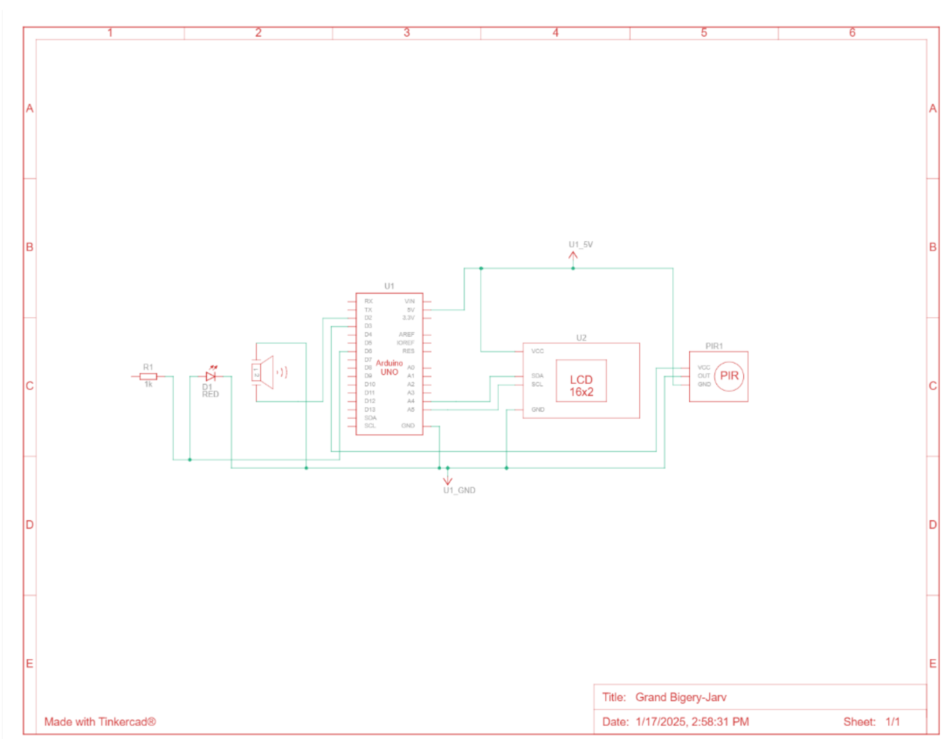


Figure 34: Schematic Diagram.

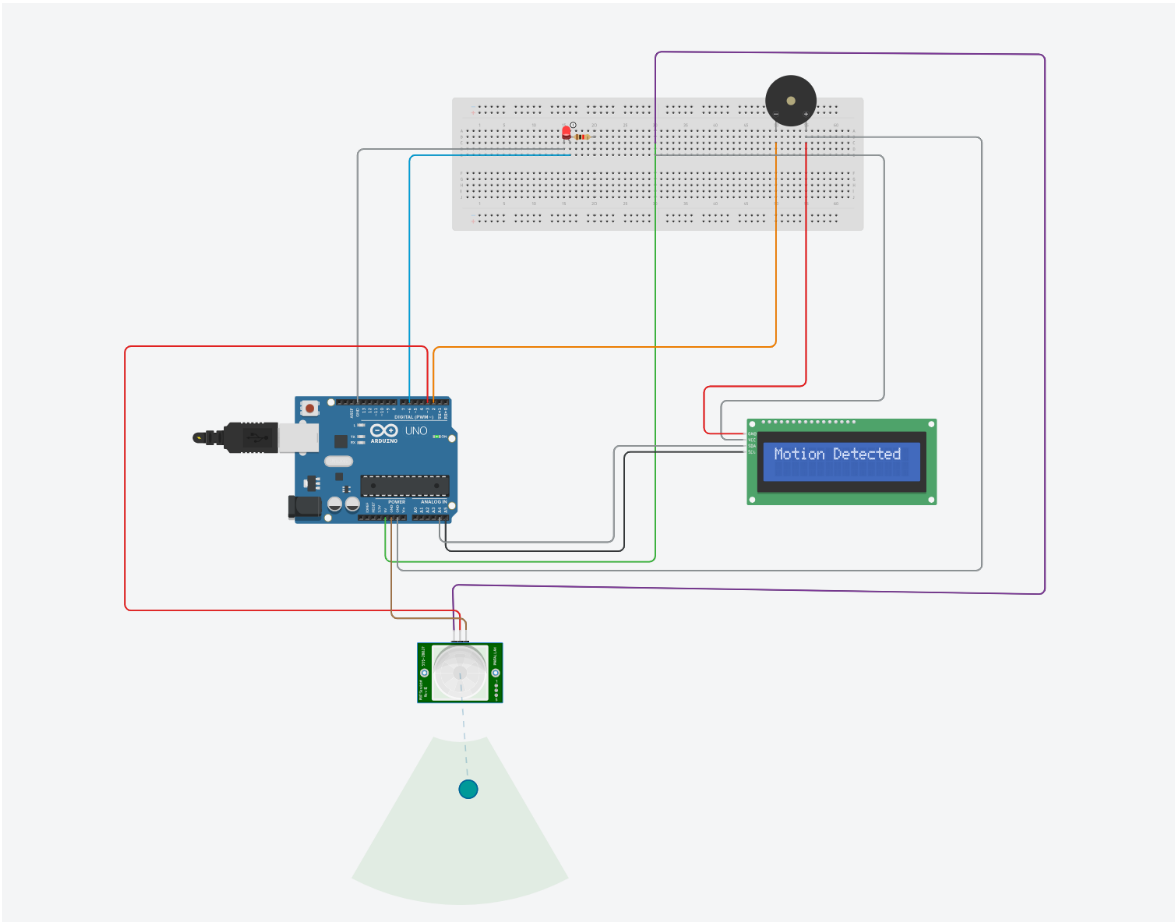


Figure 35: Circuit Diagram.

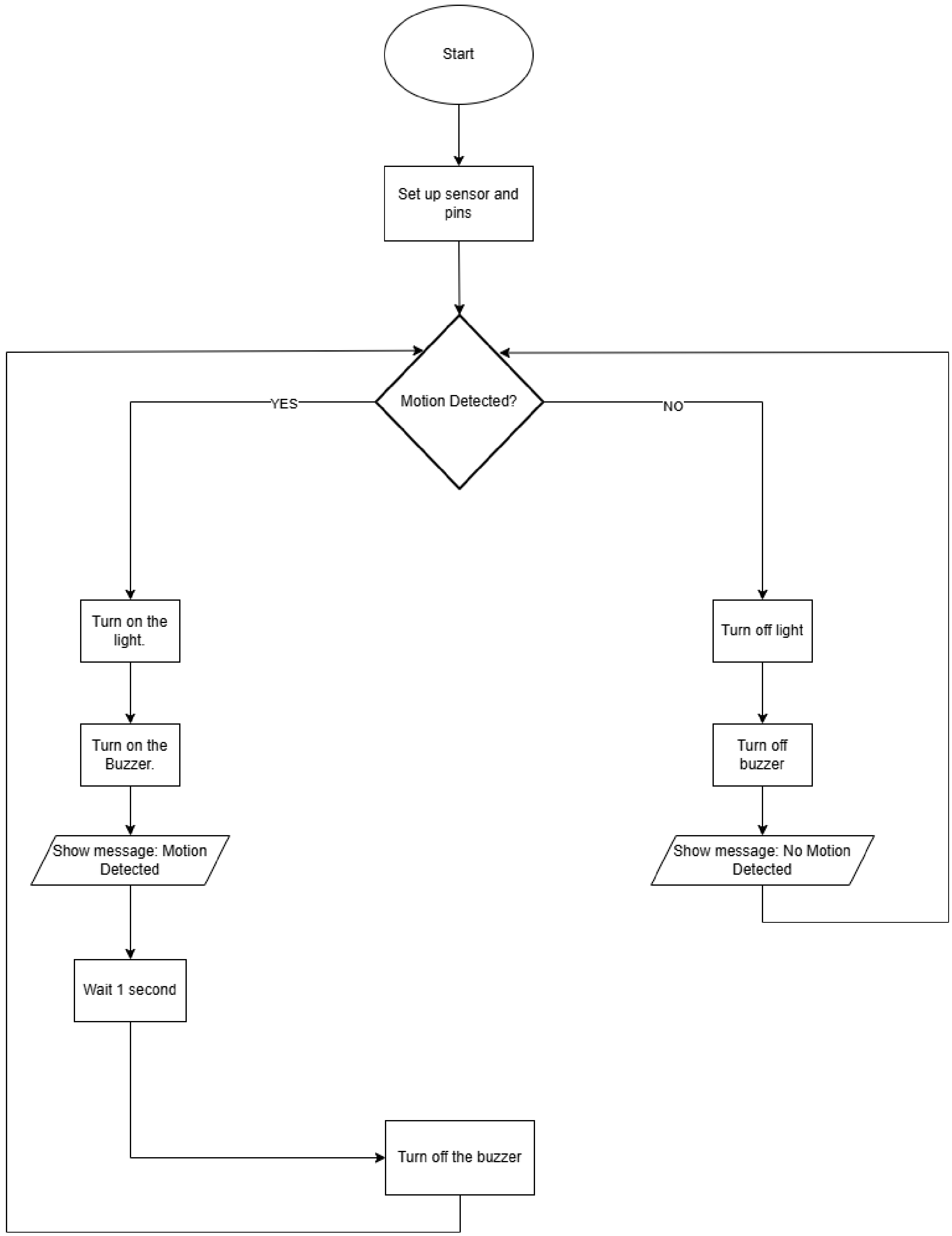


Figure 36: Flowchart showing simple flow.

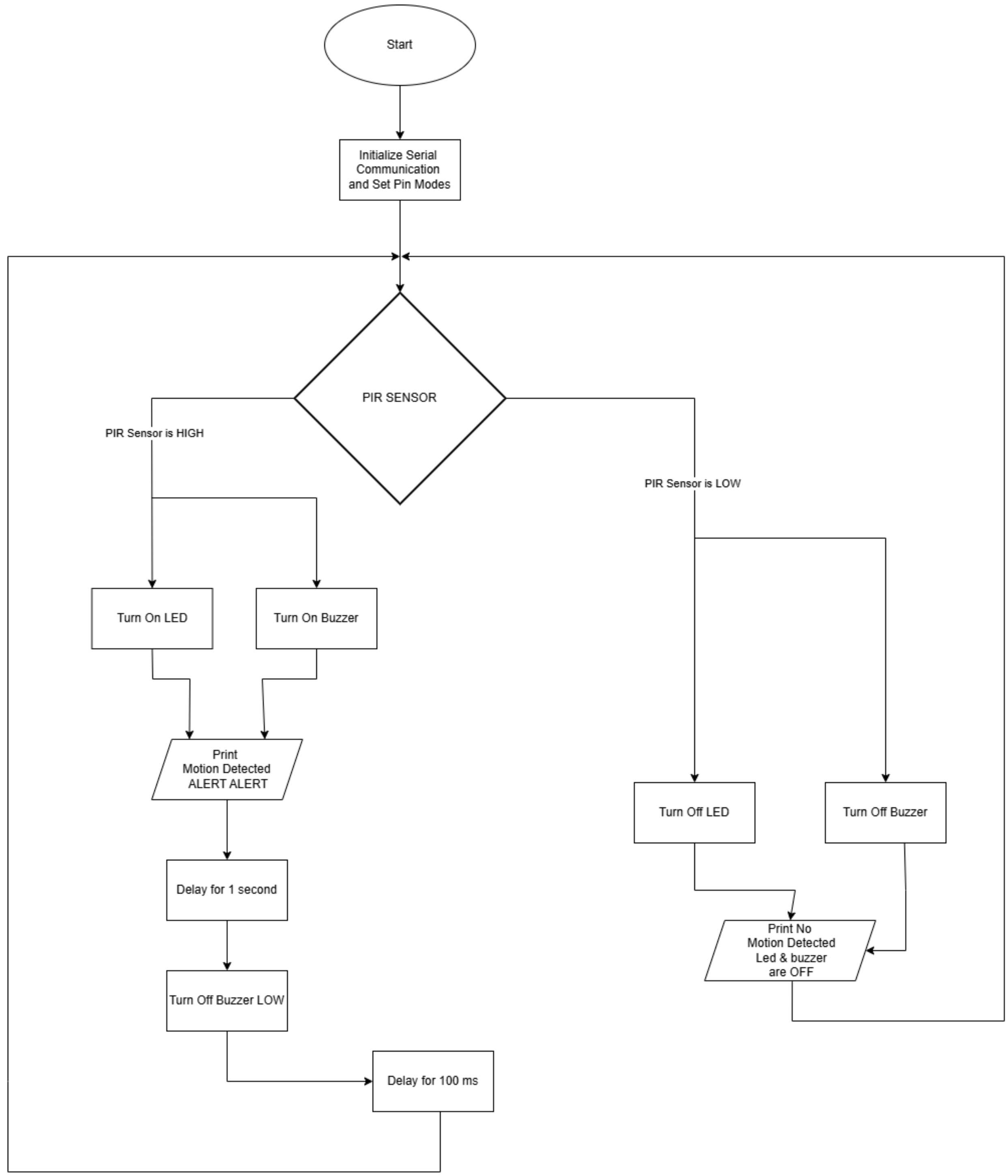


Figure 37: Flowchart showing code flow.