ROOM OCCUPANCY COUNTER

An Embedded System and IoT-Based Solution for Real-Time Occupancy Tracking

Anupriya S

B.E. in Electronics and Communication Engineering

Institution: UpSkill Campus

Internship Project Report

ABSTRACT

The *Room Occupancy Counter* is a compact embedded system that monitors the number of people in a room using two infrared (IR) sensors and an Arduino Uno microcontroller. As individuals enter and exit, the sensors detect movement, updating the room's occupancy count in real-time on a 16x2 LCD screen. This project provides an effective solution for managing room capacity in spaces like classrooms and offices, with potential for future expansion into IoT-enabled remote monitoring for enhanced resource management and safety. Additionally, the system is designed for low power consumption, ensuring energy efficiency. The simple yet scalable design also allows for easy modifications, such as adding network capabilities for remote data access. Overall, this project highlights the practicality and versatility of embedded systems in modern occupancy management solutions.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to everyone at UpSkill Campus for their support and guidance throughout my internship. Their expertise and encouragement have been invaluable in helping me develop practical skills in embedded systems and IoT. This project on the *Room Occupancy Counter* has enhanced my understanding of embedded technology and strengthened my skills in project development, which will greatly contribute to my future career in electronics and communication engineering.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The rapid growth of technology has transformed how we manage and interact with physical spaces, particularly in settings like schools, offices, and public venues. One of the emerging applications in this field is room occupancy monitoring, which enables facilities to track the number of people in a room at any given time. Such systems are increasingly essential for ensuring safety, optimizing space utilization, and complying with occupancy limits. The *Room Occupancy Counter* project addresses these needs by providing a simple, efficient solution to monitor room occupancy in real-time using embedded technology. With its combination of infrared sensors and a microcontroller, this project delivers practical and immediate information on room occupancy, making it valuable in various settings.

In spaces where maintaining regulated occupancy levels is critical, knowing the exact number of people present can significantly impact planning and safety. Conference rooms, libraries, classrooms, and waiting areas are some examples where controlling crowd density is essential for both security and functionality. Traditional manual counting methods are often prone to errors, especially in high-traffic environments, and can also be labor-intensive. An automated occupancy counter overcomes these limitations by providing a consistent and reliable count of entries and exits, ultimately enhancing space management. Additionally, the data gathered by such a system can inform decisions on facility use and capacity planning, adding a layer of intelligence to space management.

The primary aim of the *Room Occupancy Counter* project is to create a low-cost, efficient system that accurately counts individuals entering and exiting a room and displays the real-time count on an LCD screen. This project uses an Arduino Uno microcontroller as the control unit, paired with two infrared (IR) sensors positioned strategically at the entrance. As people cross the entry and exit points, the sensors detect the movement, allowing the system to update the room occupancy count. The data is then displayed on a 16x2 LCD, providing a user-friendly interface that clearly shows the current occupancy count. This straightforward approach ensures both accessibility and practicality for users and administrators alike.

One of the key design considerations in this project is ensuring reliable and accurate counting, which requires effective handling of sensor signals and movement detection. The infrared sensors detect motion by registering changes in infrared light caused by a person's presence. Each sensor is positioned to capture either entry or exit, enabling the Arduino to distinguish between these movements based on the sequence in which the sensors are triggered. The Arduino code manages these signals with logic that counts entries and exits while avoiding double-counting or missed counts, particularly during high-traffic periods. This approach ensures accurate, real-time tracking of occupancy.

To further enhance accuracy, the system's design incorporates specific delays to prevent false signals, such as when multiple people pass through quickly. Sensor placement is also optimized to reduce interference from other objects, ensuring the system only detects relevant movement. These measures help maintain accuracy and make the system adaptable to various environments where different factors, like speed of movement or room size, may influence performance. The result is a reliable and robust system that can function effectively in diverse applications, from small meeting rooms to larger public spaces.

CHAPTER 2

EMBEDDED SYSTEMS AND IOT

2.1 EMBEDDED SYSTEMS

Embedded systems are specialized computing systems that perform dedicated functions within larger systems. These are often integrated into devices to provide specific control functions, making them indispensable in modern electronics.

2.1.1 KEY CHARACTERISTICS OF EMBEDDED SYSTEMS

- Single-Functioned: Designed to perform a specific task.
- Real-Time Operation: Can process inputs and outputs quickly, often in real-time.
- Embedded Software: Runs specific software tailored for particular tasks.
- Resource Constraints: Optimized for minimal resources such as memory, power, and processing power.

2.1.2 EXAMPLES OF EMBEDDED SYSTEMS

- Automobiles: Engine control units (ECUs) that manage fuel injection, airbags, and ABS systems.
- Consumer Electronics: Televisions, washing machines, and microwave ovens.
- Industrial Machines: Robotics, automated assembly lines, and control systems.
- Healthcare Devices: Medical devices like pacemakers and infusion pumps.

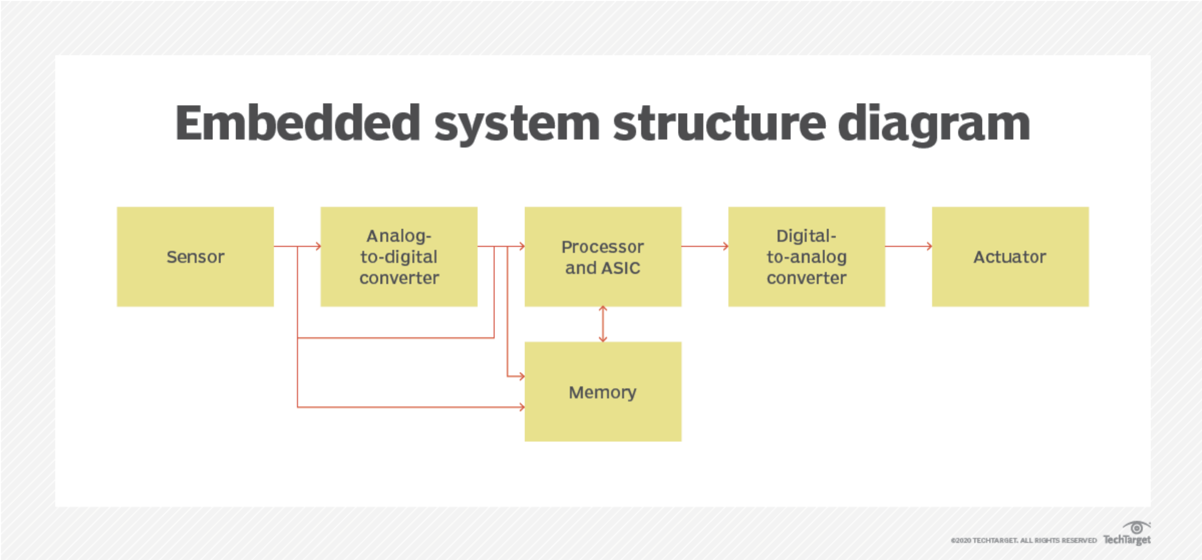

Fig 2.1.1 Embedded system structure diagram

In terms of hardware, a basic embedded system consists of the following elements:

- Sensors. These components convert physical sense data into an electrical signal.
- Analog-to-digital converters. A-D converters change an analog electrical signal into a digital one.
- Processors. These process digital signals and store them in memory.
- Digital-to-analog converters. D-A converters change the digital data from the processor into analog data.
- Actuators. These components control the mechanical motion of the embedded system by converting electrical signals into physical actions.

The sensor reads external inputs, the converters make that input readable to the processor, and the processor turns that information into useful output for the embedded system.

2.2 INTERNET OF THINGS (IOT)

The Internet of Things (IoT) refers to the interconnected network of devices, sensors, and software, enabling these entities to collect, exchange, and act on data. IoT extends the concept of embedded systems by adding connectivity, allowing devices to communicate with each other and with central servers.

Fig 2.2.1 Internet of Things (IoT)

2.2.1 KEY CHARACTERISTICS OF IOT

- Connectivity: Devices are connected to the internet or other networks.
- Data Collection: Sensors gather data from the environment.
- Data Processing: Data is processed locally or in the cloud to derive meaningful insights.
- Automation and Control: Devices can be controlled remotely, and actions can be automated based on data insights.

2.2.2 EXAMPLES OF IOT APPLICATIONS

- Smart Homes: Devices like smart thermostats, security cameras, and lighting systems.
- Wearables: Fitness trackers and smartwatches.
- Industrial IoT: Smart factories, predictive maintenance, and supply chain monitoring.
- Healthcare: Remote patient monitoring and connected medical devices.
- Agriculture: Precision farming and smart irrigation systems.

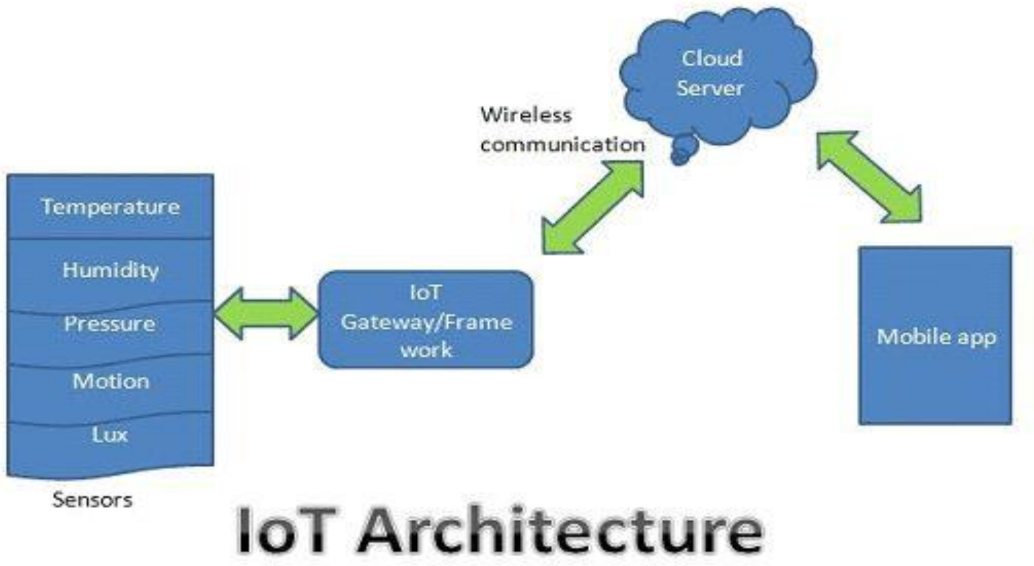

Fig 2.2.3.1 IoT Architecture

2.2.3 EXPLANATION OF IOT ARCHITECTURE

1. **Sensors:** Sensors are everywhere, sensors sense data from the atmosphere or place. the eg. temperature sensor senses temperature from the room and shares it through IoT gateway.

2. **IoT Gateways & frameworks:** Gateways act as a carrier between the internal network of sensor nodes with the external Internet or World Wide Web.

3. **Cloud Server:** The data transmitted through the gateway is stored & processed securely within the cloud server i.e. in data centres. This processed data is then used to perform intelligent actions that make all our devices Smart Devices. In the cloud, all analytics and decision making happen considering user comfort.

4. **Mobile Applications:** The intuitive mobile apps will help end-users to control & monitor their devices (ranging from the room thermostat to vehicle engines) from remote locations.

CHAPTER 3
ROOM OCCUPANCY COUNTER

The *Room Occupancy Counter* is a practical and efficient embedded system designed to monitor and display the number of people in a room in real-time. The project addresses the need for effective space management by offering a simple, cost-effective solution that can be implemented in various public and private spaces. Utilizing two infrared (IR) sensors positioned at the entryway and a microcontroller to process signals, this system provides a reliable count of individuals entering and exiting a room, updating the total count on an LCD screen. This setup is especially useful for spaces that require controlled access, such as conference rooms, libraries, and classrooms, where knowing the number of people present is crucial for safety, security, and resource management. Additionally, the real-time data provided by the system allows for quick response in case of emergencies or overcrowding, enhancing occupant safety. The system's compact design makes it easy to install, while its minimal power requirements ensure it can operate efficiently over extended periods. With potential upgrades, this device can integrate with IoT networks, allowing remote monitoring and centralized control. Furthermore, the collected occupancy data can provide valuable insights into usage trends, supporting informed decision-making for facility managers.

3.1 COMPONENTS

1. Arduino Uno
2. 16x2 LCD Display
3. Two Infrared (IR) Sensors
4. Jumper Wires
5. Power supply

3.2 BLOCK DIAGRAM

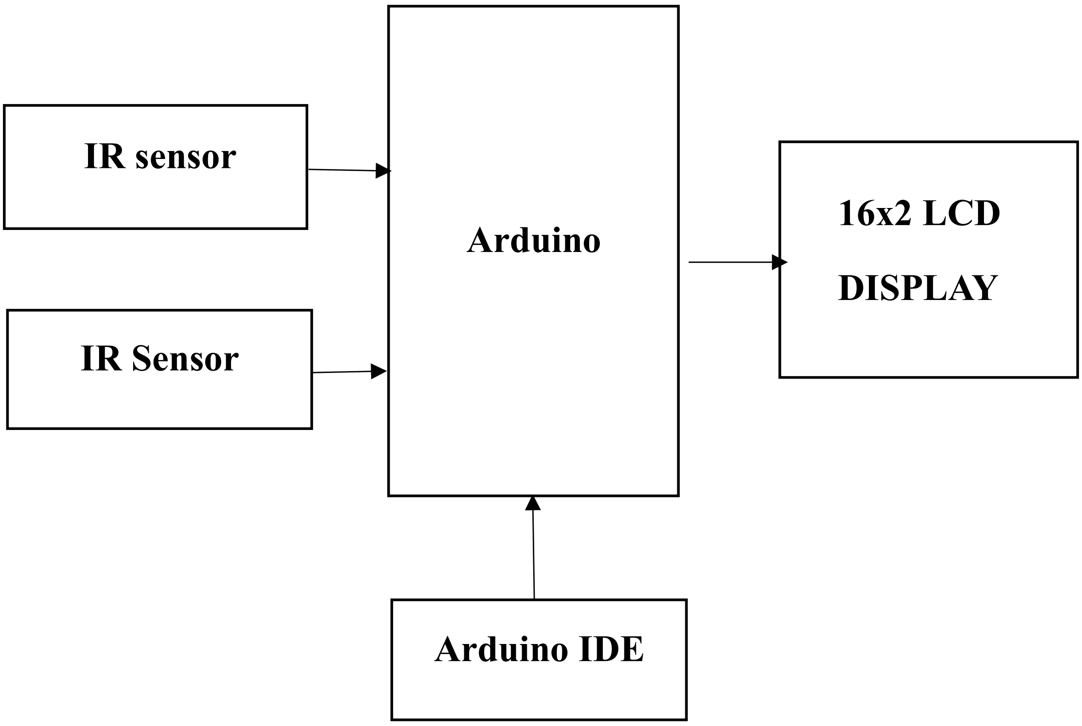

Fig 3.2.1. Block diagram

3.3 BLOCK DIAGRAM EXPLANATION

In the block diagram, each component has a specific role in maintaining an accurate occupancy count. The two IR sensors are positioned on either side of the entry, allowing the system to detect directional movement and distinguish between entry and exit events. When someone enters the room, the first sensor activates, and the Arduino increases the occupancy count. Similarly, if someone exits, the second sensor activates, and the count is reduced. The updated count is then displayed on the 16x2 LCD screen, which provides real-time data for anyone managing or monitoring the room's occupancy. The Arduino is programmed with logic to prevent double-counting by including a delay between sensor activations, ensuring reliable results even during peak hours. This simple yet effective design allows the system to operate autonomously, requiring minimal user interaction.

3.4 CIRCUIT DIAGRAM

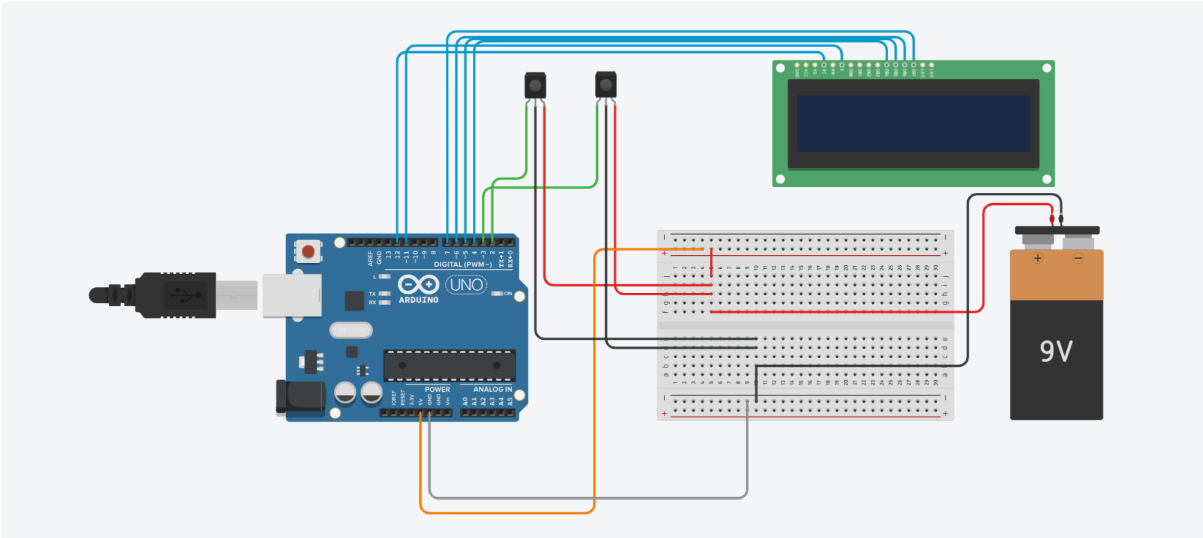

Fig 3.4.1 CIRCUIT DIAGRAM

3.5 WORKING PRINCIPLES

The *Room Occupancy Counter* operates based on movement detection and signal processing principles, using infrared (IR) sensors to capture entry and exit events. Each IR sensor is strategically placed at the room's entrance to detect individuals crossing through. The entry sensor detects people entering the room, while the exit sensor captures those leaving. When a person crosses the entry sensor, it sends a high signal (digital “1”) to the Arduino, which interprets this as an entry event, increasing the occupancy count by one. Conversely, when a person crosses the exit sensor, the Arduino receives a similar high signal, identifying the event as an exit and decreasing the count by one. This enables the system to continuously track the current number of people in the room.

The Arduino’s code is specifically designed to ensure accuracy even in busy conditions. For example, delay functions are incorporated to prevent “bouncing” or double-counting when people move quickly or if multiple people pass through simultaneously. These delays give the system time to distinguish between separate individuals, reducing the likelihood of error. Additionally, by managing sensor signals through conditional logic, the Arduino effectively discards any false readings or duplicate signals that might occur in quick succession, maintaining an accurate tally.

To deliver instant and reliable feedback, the Arduino processes each entry and exit signal in real time. This real-time signal processing ensures that every entry and exit is captured as soon as it happens, without lag. The updated occupancy count is immediately displayed on a 16x2 LCD screen, allowing users or room managers to monitor the occupancy status at a glance. This visual feedback is particularly valuable in situations where rapid decision-making might be needed, such as managing crowd limits during peak times or in emergencies.

The system's reliance on IR sensors, which detect disruptions in infrared light when a person passes, is both effective and efficient for monitoring human movement. This setup is sensitive enough to respond to the presence of people but not overly sensitive to minor disturbances, such as air movement or changes in light levels, making it highly suitable for occupancy counting. The simplicity of this approach, combined with the reliability of the Arduino microcontroller, ensures a system that is both user-friendly and robust.

The modular design of the *Room Occupancy Counter* also allows for potential scalability. Additional sensors can be integrated if required, and the Arduino can be programmed to handle more complex occupancy logic, such as counting people in larger spaces with multiple entry and exit points. This modularity further highlights the flexibility of the system, ensuring that it can be adapted to a wide range of applications beyond simple room occupancy tracking.

3.6 COMPONENTS EXPLANATION

3.6.1. ARDUINO UNO

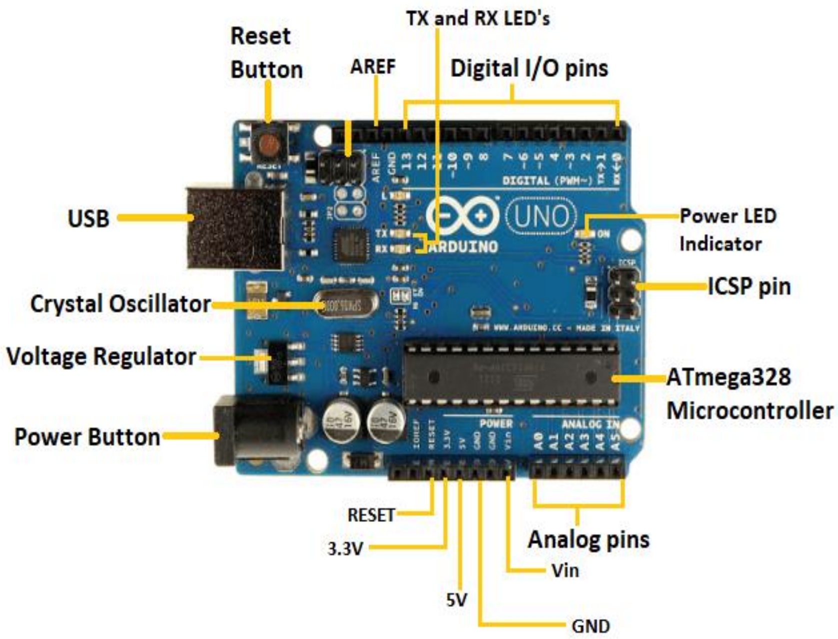

Fig 3.6.1.1. Arduino Uno

The Arduino Uno microcontroller serves as the system's "brain," running code that interprets input from the IR sensors and updates the occupancy count on the LCD display. The Arduino Uno is based on the ATmega328P microcontroller and is known for its versatility, ease of programming, and compatibility with a wide range of sensors, displays, and modules. It efficiently processes signals from the sensors and updates the display without any noticeable delay. The Arduino's programmability allows precise handling of signals, ensuring that entries and exits are counted accurately. Additionally, its low power consumption makes it suitable for continuous operation, which is essential for a real-time occupancy counter. The board's digital and analog input/output pins also allow for future expansion, such as adding additional sensors or connecting to network modules for remote monitoring. The Arduino's built-in USB interface simplifies code uploading and debugging, making development fast and accessible.

3.6.2 16x2 LCD DISPLAY

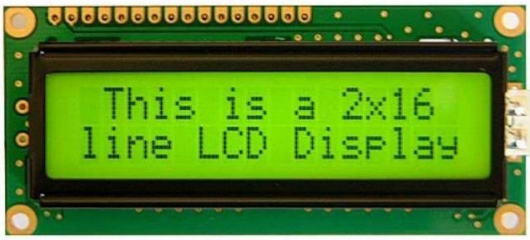

Fig 3.6.2.1 16x2 LCD Display

This display is used to present the current occupancy count visually. The 16x2 LCD screen can display two lines of up to 16 characters, which is ideal for showing entry, exit, and total occupancy counts. The LCD is directly connected to the Arduino and is programmed to update in real-time with changes in the count, providing instant feedback to anyone monitoring the room. Its backlighting ensures visibility in various lighting conditions, making it easy to read. This simple and effective display is well-suited to this project, as it is both low-cost and compatible with the Arduino Uno. Additionally, the LCD’s power requirements are low, allowing it to run efficiently without burdening the system’s power supply. The display can be easily configured to show customizable messages, making it adaptable to other projects with different display needs. Its robust design also ensures durability in continuous-use applications, supporting the *Room Occupancy Counter* for long-term reliability. The LCD module's compatibility with standard libraries makes integration and programming straightforward, minimizing development time. Furthermore, the use of a 16x2 LCD screen allows for efficient space management, as it keeps the interface compact while still delivering essential data. Its low power consumption ensures it won't significantly drain the system, which is ideal for battery-powered projects or continuous usage. The clarity and simplicity of the LCD's design also promote ease of use, making it a user-friendly option for displaying real-time occupancy data in an intuitive and accessible format.

3.6.3 IR SENSOR

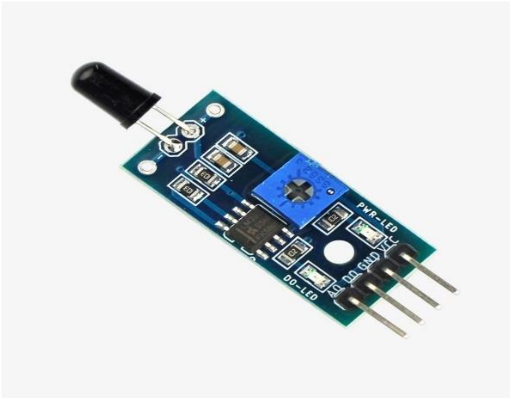

Fig 3.6.3.1 IR Sensor

Infrared (IR) sensors are fundamental to the system's operation, as they detect the movement of people entering and exiting the room. These sensors emit infrared light, which is invisible to the human eye but can detect the presence of objects or people when the emitted light is interrupted. Each IR sensor is positioned strategically—one at the entry point and the other at the exit. When a person crosses the IR beam, the sensor detects a change in infrared levels and sends a high signal to the Arduino, which then processes this signal as either an entry or an exit based on which sensor is triggered first. IR sensors are ideal for this purpose because they provide reliable detection, are unaffected by ambient light, and can distinguish between the direction of movement based on placement, ensuring accurate occupancy counting. Additionally, IR sensors are low-cost, easy to integrate with the Arduino, and require minimal power to operate, making them a cost-effective solution for long-term use. The sensors' non-contact nature also makes them ideal for hygienic applications, as they do not require physical interaction, reducing wear and tear. Furthermore, the sensors' quick response time ensures near-instantaneous updates to the occupancy count, providing real-time monitoring without delays.

3.6.4 BREADBOARD

The breadboard provides a temporary, organized platform for connecting the components without soldering. It allows for easy placement and rearrangement of wires, sensors, and other components, making it ideal for prototyping and testing the circuit. By connecting the Arduino, IR sensors, LCD, and power source through the breadboard, you ensure stable signal paths and power distribution while maintaining flexibility. The breadboard allows for quick adjustments in component positioning and wiring, especially when optimizing the sensor placement for accurate entry and exit detection.

3.6.5 JUMPER WIRES

Jumper wires connect all the components on the breadboard, including the Arduino, IR sensors, LCD, and power supply. These wires ensure that signals are transmitted accurately and that power flows consistently across the system. They come in different types and lengths (male-to-male, female-to-female, and male-to-female), allowing for precise connections. High-quality jumper wires are essential to prevent signal interference or disconnections, especially in a real-time counting system where reliable signal transmission is crucial.

3.6.6 POWER SUPPLY

The power supply ensures continuous operation by providing the necessary voltage for the Arduino, IR sensors, LCD, and other components. Typically, the system can be powered by a USB connection from a laptop or adapter, or by an external battery if the system needs to be portable. The Arduino Uno and associated components typically operate at 5V, so a single power source can efficiently support the entire setup. Consistent power ensures that the system runs smoothly 24/7, making it suitable for long-term occupancy monitoring applications.

3.7 PROGRAM

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int entryPin = 6;

int exitPin = 7;

int entryCount = 0;

int exitCount = 0;

int currentCount = 0;

void setup() {

  pinMode(entryPin, INPUT);

  pinMode(exitPin, INPUT);

  lcd.begin(16, 2);

}

void loop() {

  int entry = digitalRead(entryPin);

  int exit = digitalRead(exitPin);

  if (entry == HIGH) {

    entryCount++;

    currentCount++;
```

```
    delay(500);  // Prevents double-counting
  }
  else if (exit == HIGH) {
    exitCount++;
    currentCount--;
    delay(500);  // Prevents double-counting
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Entries: ");
  lcd.print(entryCount);
  lcd.setCursor(0, 1);
  lcd.print("Exits: ");
  lcd.print(exitCount);
  lcd.setCursor(8, 1);
  lcd.print("Curr: ");
  lcd.print(currentCount);
  delay(1000);
}
```

3.8 KEY FEATURES FOR ROOM OCCUPANCY COUNTER

- **Real-Time Monitoring:** The system provides real-time occupancy data on the LCD display.
- **Simplicity and Efficiency:** Minimal components make the system easy to build and cost-effective.
- **Low Power Consumption:** The Arduino and sensors are designed to run on low power, ensuring sustainability.
- **Scalability:** The system can be expanded to connect with other IoT systems for remote monitoring.
- **Accurate Directional Detection:** IR sensors detect the direction of movement, ensuring precise entry and exit tracking.
- **User-Friendly Interface:** The LCD display provides an easy-to-read output for users, making it intuitive to understand occupancy levels at a glance.
- **Compact Design:** The system is designed to be space-efficient, ideal for environments where space is limited.
- **Hygienic and Non-Contact Operation:** The IR sensors enable contactless monitoring, making the system more hygienic and reducing wear and tear.
- **Cost-Effective:** The low-cost components used in the system make it affordable for both small and large-scale implementations.
- **Easy to Install:** The system's simple design allows for easy setup and integration into existing spaces without the need for complex installation processes.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

The *Room Occupancy Counter* is an effective embedded system project that combines practical design with real-time monitoring. The system provides an accessible solution for managing room occupancy, enhancing safety, and optimizing space usage. By using simple components like the Arduino Uno, IR sensors, and an LCD, the project achieves efficient functionality with room for further development.

4.2. FUTURE WORK

Future improvements could include IoT connectivity for remote occupancy monitoring using Wi-Fi modules, and data analytics to track occupancy patterns. Additionally, expanding the system with advanced sensors or integrating machine learning could improve its adaptability for larger-scale applications.