Gesture-Based Smart Home Control Using IR Proximity Sensors

A PROJECT REPORT

Submitted by

POORANY PRIYADHARSHINY T	**(311621104030)**
SUNITHA CHANDA	**(311621104048)**

in partial fulfilment for the award of degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

THORAIPAKKAM, CHENNAI-600097

ANNA UNIVERSITY: CHENNAI 600025

MAY 2025

ANNA UNIVERSITY: CHENNAI 600025
BONAFIDE CERTIFICATE

Certified that this project report "**Gesture-Based Smart Home Control Using IR Proximity Sensors**" is the bonafide work of "**POORANY PRIYADHARSHINY T (311621104030)" ,** and **"SUNITHA CHANDA (311621104048)** ", who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. N. Saravanan M.E., Ph.D.,	Mr. C. Annadurai, M.E.**.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSOCIATE PROFESSOR
Department of Computer	Department of Computer
Science and Engineering,	Science and Engineering,
Misrimal Navajee Munoth	Misrimal Navajee Munoth
Jain Engineering College,	Jain Engineering College,
Thoraipakkam,	Thoraipakkam,
Chennai-600097.	Chennai-600097.

Submitted for the Project Viva-Voce Examination held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere gratitude to our honorable **Secretary** (Administration) **Dr. Harish L Metha** and **Secretary** (Academic) **Shri L. Jaswant S Munoth** for providing the infrastructure facilities to do this project during our course period.

We thank our **Principal**, **Dr. C. C. Christopher** for his support and motivation for the development and completion of this project.

We express a profound sense of gratitude and thanks to our **Head of the Department**, **Dr. N. Saravanan M.E., Ph.D., Professor** for his valuable suggestions and guidance for the development and completion of this project.

We thank our project coordinator, **Ms. A. Nivashini, M.E.,(Ph.D.,), Assistant Professor**, for her valuable suggestions and constant encouragement that led us to the successful completion of our project.

We thank our supervisor **Mr. C. Annadurai, M.E.,Associate Professor,** for his constant help and valuable suggestions.

We thank all the Teaching and Non-Teaching Staff members of our department who helped us to complete our project.

Above all we thank the Almighty, and our parents and for their constant support and encouragement for completing this project.

MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

- **VISION**

Producing competent computer engineers with a strong background in the latest trends and technology to achieve academic excellence and to become pioneers in software and hardware products with an ethical approach to serve the society.

- **MISSION**

To provide quality education by inculcating strong fundamentals in basic science, mathematics and engineering concepts through the state-of-the-art facilities. To provide the learning ambience that helps the students to enhance problem solving skills and to inculcate in them the habit of continuous learning in their domain of interest. To serve the society by providing insight solutions to the real-world problems by employing the latest trends of computing technology with strict adherence to professional and ethical responsibilities.

ABSTRACT

In recent years, the rapid evolution of smart home technologies has significantly transformed the way individuals interact with household appliances. These advancements aim to promote not only convenience and automation but also energy efficiency and improved accessibility for all users, including those with physical limitations. This project, titled **"IR Proximity Sensor-Based Fan and Light Control System Using Arduino,"** introduces a cost-effective, contactless method for controlling basic home appliances—specifically, a fan and a light—using **Infrared (IR) proximity sensors**.

Unlike traditional systems that rely on mechanical switches or more expensive sensors like flux (Hall Effect) sensors, this project employs **IR proximity sensors** to detect user gestures, allowing for seamless and hygienic operation without physical contact. The core of the system is an **Arduino Uno** microcontroller, which receives input from two IR sensors and uses a **2-channel relay module** to control the state of the connected fan and light. This setup provides immediate, intuitive feedback, making it highly suitable for individuals with mobility impairments or disabilities, and equally beneficial in environments where touch-free operation is desired, such as hospitals and public facilities.

The report comprehensively details each phase of development, including system architecture, circuit design, sensor integration, programming logic, and extensive hardware testing. Emphasis is placed on the accessibility, scalability, and reliability of the solution. In addition to being an assistive tool, this project serves as an educational guide for students and makers interested in embedded systems, automation, and human-centered design.

TABLE OF CONTENT

LISTS OF FIGURES

LIST OF ABBREVIATIONS

S NO	ABBREVIATION	EXPANSION
1.	AC	Alternating Current
2.	COM	Common Terminal
3.	DC	Direct Current
4.	GND	Ground
5.	IDE	Integrated Development Environment
6.	IR	Infrared
7.	LED	Light Emitting Diode
8.	NO	Normally Open
9.	UNO	Arduino UNO Microcontroller Board
10.	USB	Universal Serial Bus
11.	VCC	Voltage Common Collector

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION:

In recent years, home automation has evolved into a transformative technology that significantly enhances the way individuals manage and interact with their household appliances. With the increasing demand for smart living, automation systems are now expected to deliver not just comfort and convenience but also accessibility, energy efficiency, and contactless operation. Traditional systems often rely on mechanical switches or mobile-based interfaces, which, while functional, may not be suitable for individuals with limited mobility or in situations where hygiene is a critical concern. To address these limitations, this project proposes a contactless automation solution using **Infrared (IR) proximity sensors** that can detect user gestures to control a **fan and a light**, providing an intuitive, non-contact interface built around an **Arduino Uno** microcontroller and a **2-channel relay module**.

The IR proximity sensor operates by emitting infrared light and measuring the reflected light that bounces off a nearby object, such as a hand. When an object comes within a certain detection range, the sensor outputs a digital signal, which is then read by the Arduino. Based on this input, the microcontroller activates or deactivates the appropriate relay channel, thereby switching the fan or light ON or OFF. This gesture-based system eliminates the need for physical interaction, making it ideal for environments like hospitals, kitchens, and homes where touchless operation is beneficial for hygiene and safety.

This project builds on the principles of embedded systems and accessible design, aiming to deliver a solution that is both technically efficient and socially inclusive. The components selected—IR sensors, relays, and Arduino Uno—are low-cost and widely available, making the system not only affordable but also replicable for educational or commercial use. The implementation requires minimal setup and can be deployed on a breadboard or printed circuit board (PCB), further increasing its versatility.

1.2 KEY FEATURES:

One of the key features of this system is its **ease of deployment**, as it does not rely on complex programming or proprietary hardware. It is also **scalable**, meaning more sensors and relays can be added to extend control to other appliances. The system is **highly responsive**, with near-instant reaction to gesture input, and **reliable**, with minimal false triggering due to the directional nature of IR sensing. Additionally, the system is **energy-efficient**, as it ensures that appliances are only powered when necessary, and can be extended to work with IoT platforms for remote monitoring and control.

Overall, this project aligns with the growing need for smarter, safer, and more inclusive home automation solutions. It demonstrates how simple yet effective sensor-based control can bridge the gap between traditional switching and next-generation intelligent environments. By leveraging gesture-based technology, the system redefines interaction with electrical appliances, making it more natural, intuitive, and universally accessible. This report explores every aspect of the development process—from concept to implementation—serving as a comprehensive guide for students, researchers, and technology enthusiasts aiming to explore embedded automation systems.

CHAPTER 2
LITERATURE SURVEY

1. SSVEP ENCHANCEMENT IN MIXED REALITY ENVIRONMENT FOR BRAIN- COMPUTER INTERFACES[13]

Author: Jieyu Wu (et all)

Year: 2025

Journal: IEEE Access

DOI: 10.1109/TNSRE.2025.3526950

Summary:

- Studies on SSVEP-based BCIs in Mixed Reality show that using optimized stimulus colors and backgrounds, especially polychromatic stimuli on coloured backgrounds, improves accuracy.

- Research with HoloLens 2 highlights that visual design plays a key role in enhancing BCI performance in MR environments.

2. APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNIQUES FOR BRAIN–COMPUTER INTERFACE IN MENTAL FATIGUE DETECTION: A SYSTEMATIC REVIEW[15]

Author: Hamwira Yaacob

Year: 2023

Journal: IEEE Access

DOI: 10.1109/ACCESS.2023.3296382

Summary:

- A systematic review of BCI-based mental fatigue detection using AI techniques from 2011 to 2022 highlights key developments and challenges.

- The study, analyzed through PRISMA guidelines, identifies issues such as data fusion, the need for hybrid models, limited public datasets, uncertainty in results, lack of explainability, and hardware implementation difficulties.

- These insights emphasize the importance of robust AI integration in improving BCI systems for fatigue detection.

3. **BRAIN–MACHINE INTERFACES THE REAL EXPERIENCE MACHINE? EXPLORING THE LIBERTARIAN RISKS OF BRAIN MACHINE INTERFACES[6]**

Author: Jorge Mateus

Year: 2025

Journal: IEEE Access

Summary:

- Recent research highlights that while Brain–Machine Interfaces (BMIs) bring significant technological progress, they also raise serious ethical concerns.
- From a libertarian viewpoint, BMIs may threaten individual autonomy, privacy, and freedom. The study urges careful examination of these risks to ensure responsible development and use of BMI technologies.

4. **NEUROMORPHIC COMPUTING FOR MODELING NEUROLOGICAL AND PSYCHIATRIC DISORDER IMPLICATION FOR DRUG DEVELOPMENT[10]**

Author: Amisha S. Raikar

Year: 2024

Journal: IEEE Access

Summary:

- Neuromorphic computing, modeled on the human brain, is emerging as a powerful tool in neurological and psychiatric disorder research.

- By combining deep learning and neural simulations, it significantly improves the modeling of brain disorders, accelerates drug discovery, and enhances the precision of clinical trials, offering promising advancements in medical technology.

CHAPTER 3
SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

Technology Used:

- Most modern home automation systems rely on IoT (Internet of Things) platforms.
- They use Wi-Fi, Bluetooth, Zigbee, or Z-Wave for wireless communication.
- Devices are controlled through smartphone apps, web portals, or voice assistants like Alexa or Google Assistant.

Core Components:

- **Internet-Based Systems**: Many home automation solutions depend on continuous internet connectivity for device control and cloud syncing.
- **Bluetooth Integration**: Some systems use Bluetooth to connect mobile apps to devices within short ranges.

Cost and Maintenance:

- High initial setup cost due to smart devices and hubs.
- Ongoing maintenance costs include software updates, hardware replacements, and paid subscriptions for advanced features or cloud access.

Drawbacks

High Cost:

One of the most significant barriers to entry for traditional smart home systems is the cost involved. Building a complete IoT-based automation setup involves several components:

- Smart switches and sensors
- Voice assistants or smart speakers

- IoT hubs or gateways
- Subscription to cloud-based services or apps
- High-speed internet connectivity

The cumulative cost of these components can easily exceed the budget of a typical household, especially in economically constrained regions. In addition, the cost of maintaining or upgrading these systems (e.g., replacing devices, renewing subscriptions) adds to the long-term financial burden.

Dependence on Internet

Another fundamental drawback is the dependence on wireless technologies, primarily Wi-Fi or Bluetooth, to communicate with and control smart devices. This introduces several challenges:

- **Connectivity limitations**: In rural or underdeveloped areas, where stable internet access may not be available, these systems become impractical or unusable.
- **Power dependency**: Internet routers and Bluetooth hubs require continuous power. In regions with frequent power outages, this dependency severely limits the usability of the system.
- **Security concerns**: Internet-connected systems are often vulnerable to hacking or unauthorized access if not properly secured. This raises concerns about privacy and system integrity.

Complexity for Elderly or Differently-Abled Users:

Despite the intention of simplifying tasks, many existing systems introduce a steep learning curve. The following challenges are especially pronounced for the elderly or people with limited mobility or cognitive disabilities:

- **Navigating apps or smart interfaces**: Operating a mobile app to control devices might not be intuitive for non-technical users.
- **Voice recognition limitations**: Voice assistants may not understand users

with speech impairments or strong regional accents, making the system unreliable for them.

- **Setup and configuration**: Initial setup of smart devices often involves complex steps like pairing with Wi-Fi, account registration, firmware updates, etc., which may not be feasible without technical support.

3.2 PROPOSED SYSTEM

The proposed system offers a simple, affordable, and accessible alternative to existing home automation solutions by using flux (Hall effect) sensors for contactless appliance control. It eliminates the need for internet connectivity, complex configurations, or mobile applications.

Appliance Control:

- The relay module serves as an **electronic switch** to turn appliances ON or OFF.
- Appliances like a **fan or LED light** can be safely connected via the relay, which isolates high voltage from the Arduino.
- The system supports **independent control** of multiple appliances through a 2-channel relay.

No Physical Contact Required:

- The Users only need to bring a **small magnet near the sensor** to operate appliances.
- This makes the system especially beneficial for users with **limited mobility, arthritis, or physical disabilities**.

Independence from Network/Internet:

- The system works in **offline mode**, with no need for:
    - Wi-Fi
    - Bluetooth
    - Mobile applications
    - Cloud services

Ensures **high reliability**, especially in rural or network-limited environments.

Low-Cost Implementation:

- Uses **basic and readily available components** like:
- Arduino Uno
- Flux sensors
- Relay module
- Breadboard and jumper wires
- No recurring subscription costs or proprietary software licensing.
- Suitable for **educational, DIY, and assistive applications**.

Portable and Scalable Design:

- Designed using a **breadboard or small PCB** for portability and demonstration.
- Can be expanded to **control more devices** by adding more sensors and relays.
- Can be further enhanced by integrating with **LCD displays** or **audio alerts** for better user feedback.

Advantages

Contactless Control:

- The system is operated without any need for physical contact.
- Users can simply move a magnet near the sensor to toggle appliances.
- This is particularly helpful for individuals with motor impairments or disabilities, where touch-based controls may not be feasible.

Low Cost and Power Efficient:

- The components used (flux sensors, Arduino Uno, and relay module) are budget-friendly and readily available in electronics markets.

- The system runs on low DC power, making it energy-efficient and suitable for battery-powered or solar-powered applications.

Simple and Portable:

- The setup requires no special installation, router configuration, or software app.
- It is compact and can be easily deployed on a breadboard or PCB for demonstration or real-world use.
- Its modular nature allows easy integration into existing appliances.

No Network Dependency:

- Unlike IoT systems, this solution does not rely on internet, Wi-Fi, or Bluetooth.
- It remains functional in rural, remote, or unstable network areas.
- This makes it more robust and reliable, especially during power outages or disconnections.

3.3 REQUIREMENT ANALYSIS

This section defines the basic needs for building and operating the system. It includes the functional requirement of detecting magnetic gestures to control appliances, along with the necessary software like Arduino IDE for coding and uploading. The hardware includes the Arduino Uno, flux sensors, relay module, fan, LED light, and power supply components essential for setting up the system.

3.3.1 Functional Requirement

The system must fulfil the following primary functionalities:

- Detect the presence of a magnetic field using flux sensors.
- Trigger a relay to toggle a connected appliance when a magnetic field is detected.

- Alternate the state of the appliance (ON to OFF or OFF to ON) with each valid magnetic gesture.
- Independently control at least two appliances—a fan and an LED light.
- Maintain stable operation without false triggering or delays.

These requirements ensure that the system can be used as a basic yet functional alternative to conventional switch-based or app-based controls.

3.3.2 Hardware Requirement

Hardware requirements define the physical components and specifications needed for a system to function

1. Arduino UNO R3 (microcontroller)
2. IR PROXIMITY SENSOR
3. RELAY MODULE (2-channel)
4. FAN
5. LED LIGHT
6. BREAD BOARD
7. JUMPER WIRES
8. POWER SUPPLY

3.3.3 Software Requirement

The software requirements give a detailed description of the system and all its features.

1. Arduino IDE (for microcontroller programming)
2. Python
3. Serial Monitor

3.4 DEVELOPMENT TOOLS USED

Arduino IDE:

- The official development environment provided by Arduino.
- Enables code writing, compiling, uploading, and debugging.
- Supports a wide range of microcontrollers, including Arduino Uno, Nano, Mega, etc.
- Features a serial monitor and easy-to-use UI, ideal for beginners and advanced users alike.

Serial Monitor:

- A console window built into the Arduino IDE.
- Displays messages and values sent from the Arduino using Serial.print() or Serial.println().
- Used during testing to verify the input signals from flux sensors.
- Helps confirm that relays are being triggered correctly according to logic.

3.5 CORE LIBRARIES

Standard Arduino Libraries:

- These libraries come pre-installed with the Arduino IDE.
- Provide functions such as digitalRead(), digitalWrite(), pinMode(), and delay(), which are critical for sensor reading and controlling relay outputs.

Digital I/O Handling:

- The program relies heavily on digital input from the sensors and digital output to the relay module.

- The I/O pins are configured as:
    - INPUT for the flux sensors (to detect magnetic field presence).
    - OUTPUT for the relay module (to switch the appliances).

CHAPTER 4

SYSTEM DESIGN

4.1 OVERVIEW OF ARCHIETECTURE

The diagram shows the overall architecture of the project which consists of the model development and evaluations.

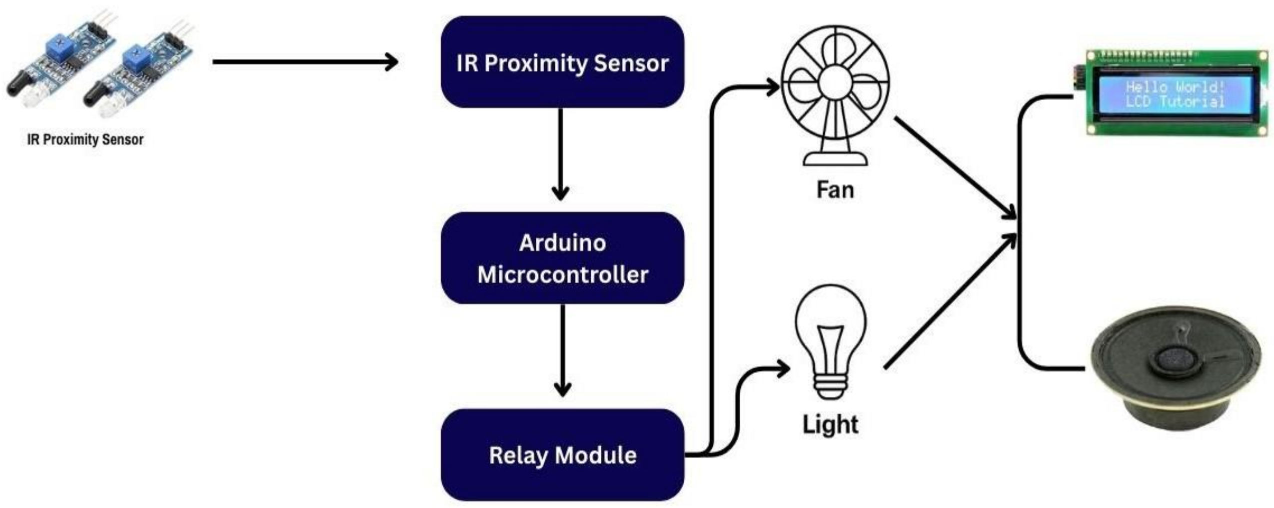

Figure 4.1 System Architecture

4.2 USE CASE DIAGRAM

This diagram shows the user such as actor, system and the role of developer in this project. This behavior diagram models the functionality of the system using use cases.

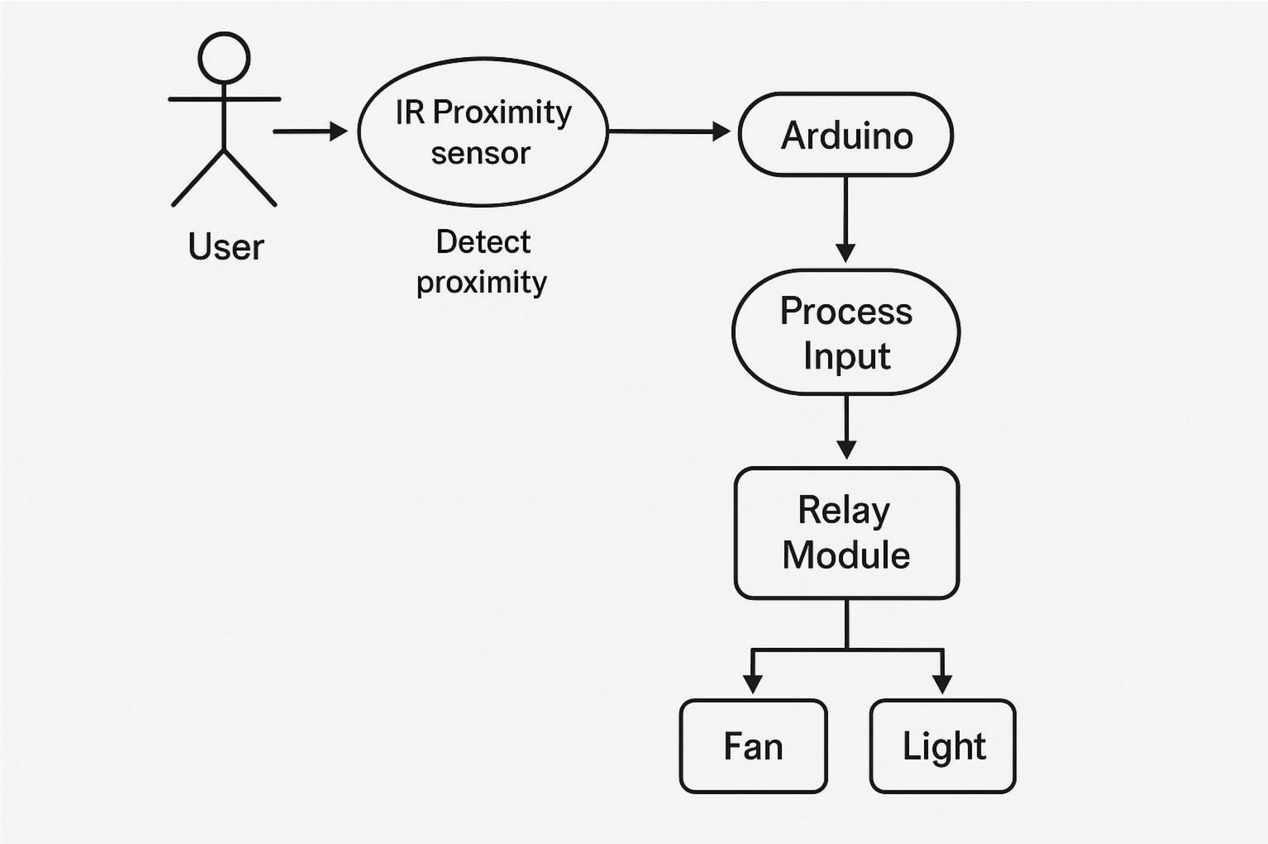

Figure 4.2 Use Case Diagram

4.3 FLOW DIAGRAM

A flow diagram is a visual representation that illustrates the sequence of operations or steps in a system or process.

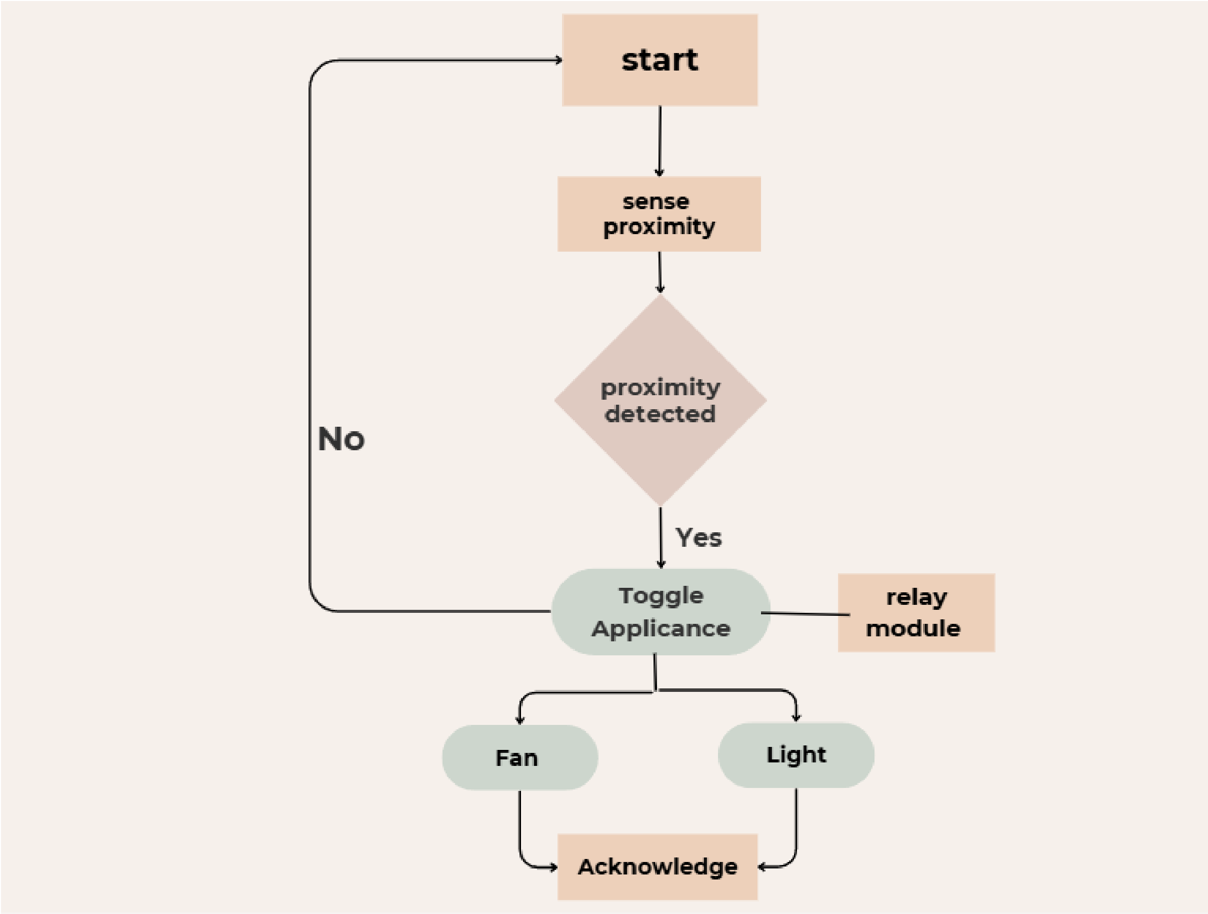

Figure 4.3 Flow Diagram

4.4 SCHEMATIC DIAGRAM

A schematic diagram is a graphical representation of an electrical or electronic circuit, using standardized symbols to depict components like resistors, sensors, microcontrollers, and connections between them.

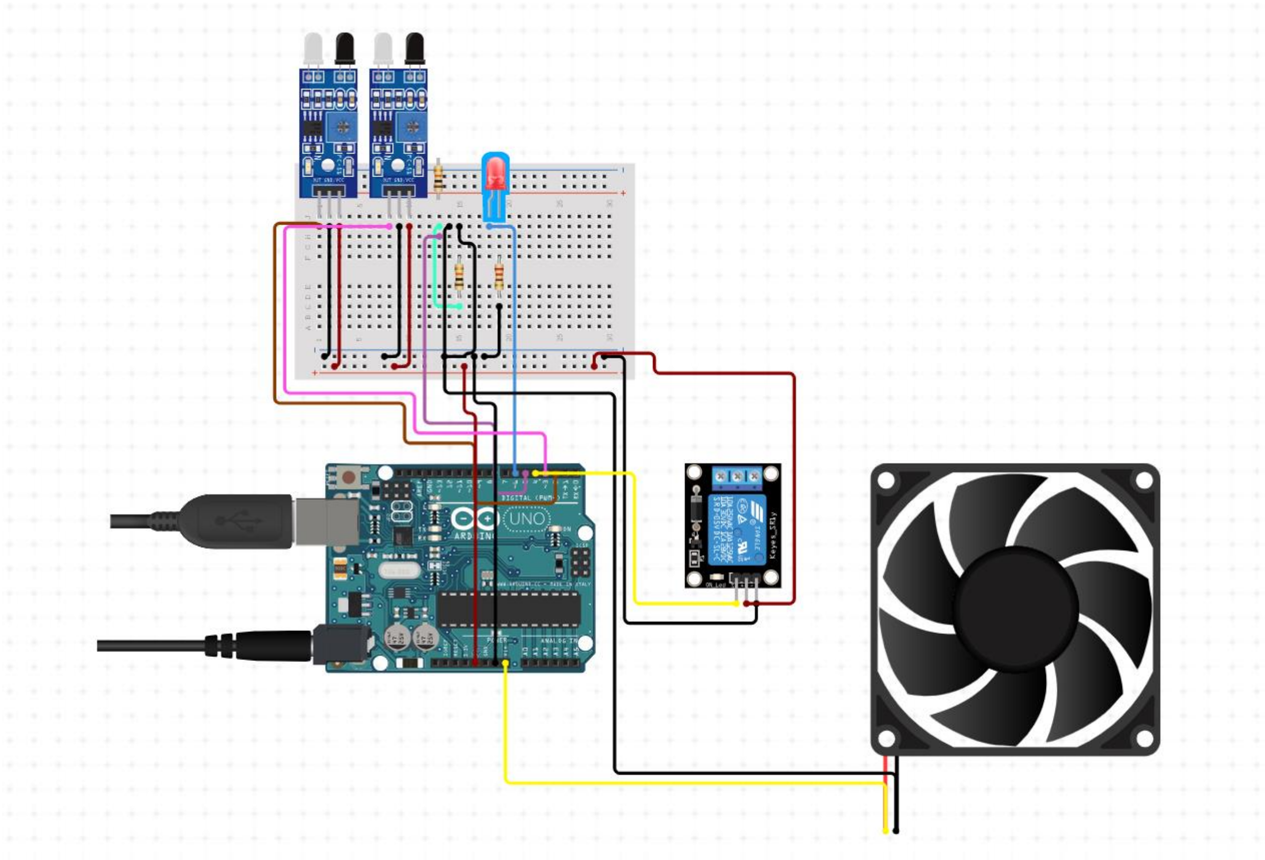

Figure 4.4 Schematic Diagram

4.5 ACTIVITY DIAGRAM

An activity diagram is a diagram that represents the dynamic workflow or behaviour of a system. It visually outlines the sequence of actions, decisions, and parallel processes in a system's operation.

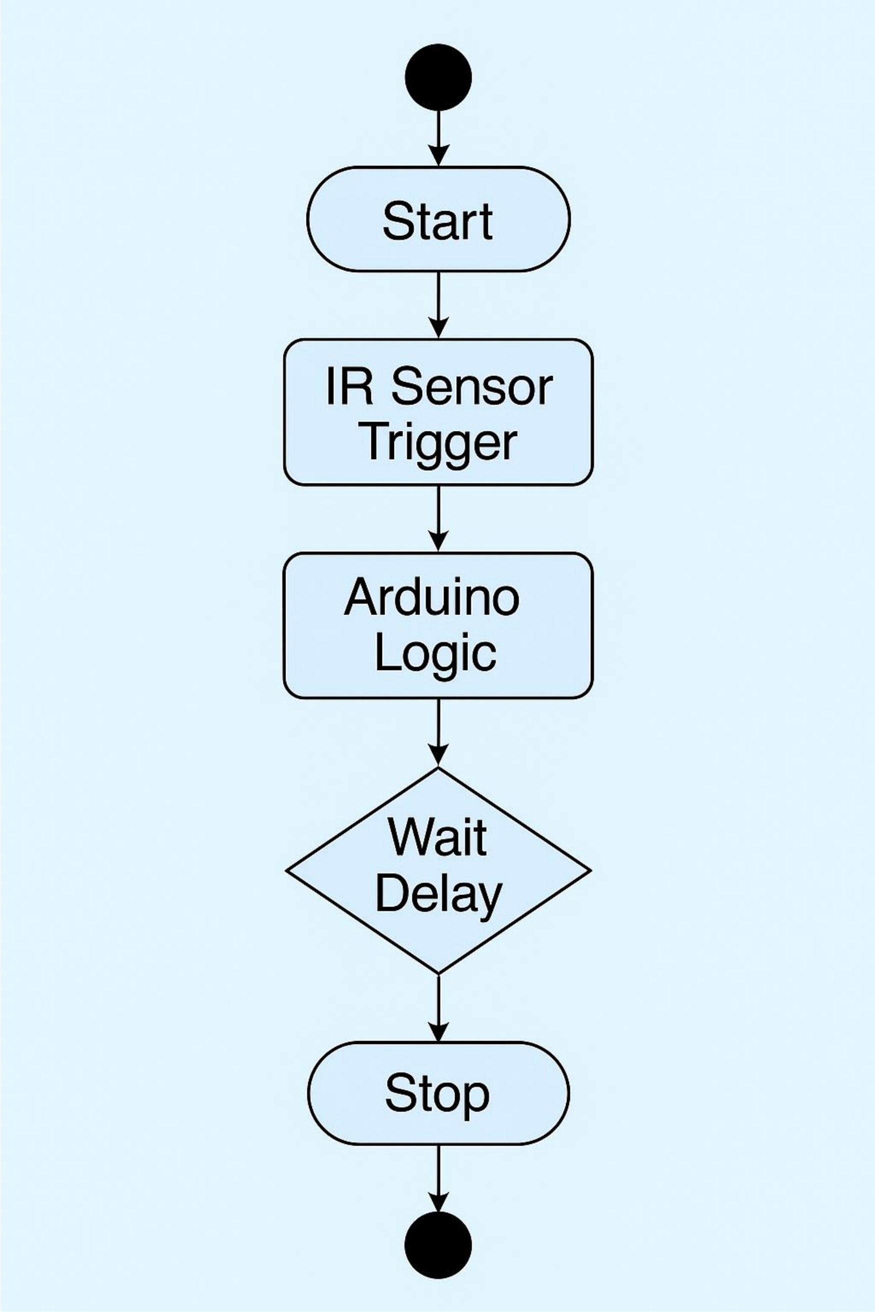

Figure 4.5 Activity Diagram

4.6 ENTITY RELATIONSHIP DIAGRAM

An ER (Entity-Relationship) diagram is a visual tool used in database design to illustrate the relationships between different data entities within a system. It defines how data is structured, stored, and connected.

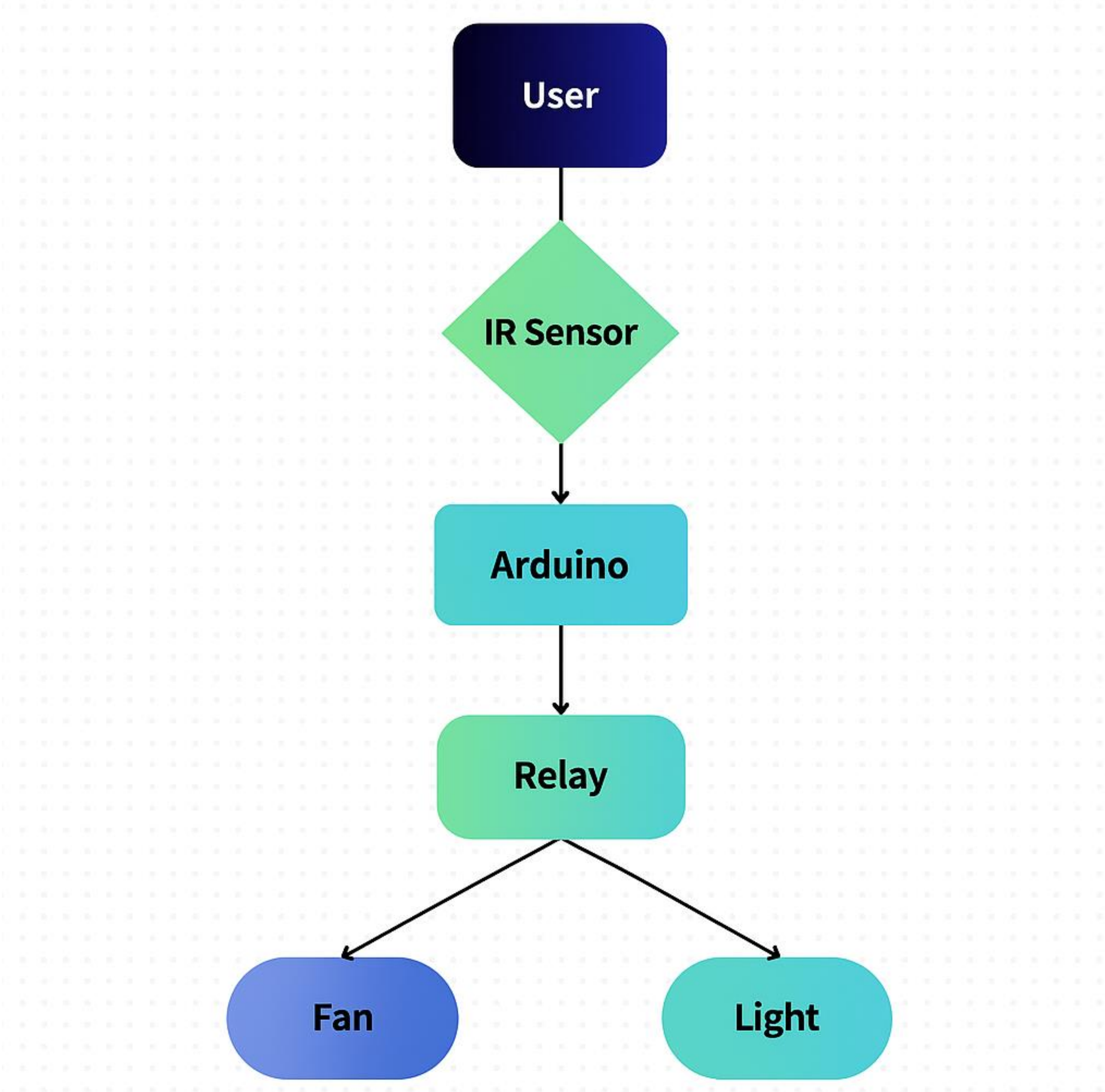

Figure 4.6 Entity Relationship Diagram

4.7 COMPONENTS

Components refer to the individual hardware and software elements that work together to form a complete system.

4.7.1 Hardware Module

1. **Arduino Uno R3**

The Arduino UNO R3 is an open-source microcontroller board based on the ATmega328P. It features 14 digital I/O pins (6 capable of PWM output), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, and an ICSP header. It is widely used in embedded systems, IoT applications, and robotics due to its simplicity, flexibility, and extensive community support. The UNO R3 is ideal for prototyping and supports programming via the Arduino IDE using C/C++.

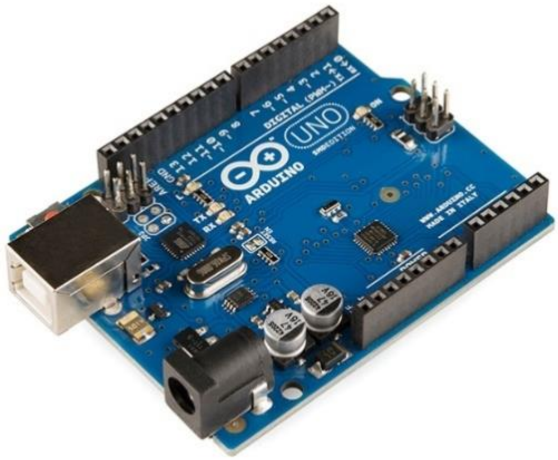

Figure 4.7 Arduino Uno R3

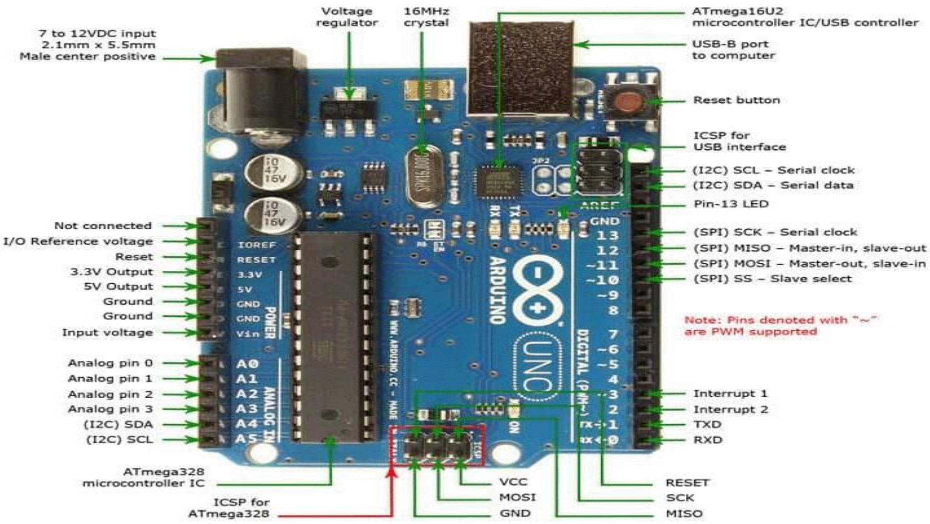

Figure 4.8 Arduino Uno R3 Pin Diagram

2. LCD Display

A 16x2 LCD (Liquid Crystal Display) is a character display module capable of showing 16 characters per row on two rows. It operates using the Hitachi HD44780 controller and is widely supported in microcontroller projects. The module uses either a 4-bit or 8-bit parallel interface to communicate with the Arduino. It’s primarily used for displaying alphanumeric data like sensor readings, system status, or alerts in real-time.

Figure 4.9 16x2 Lcd Display

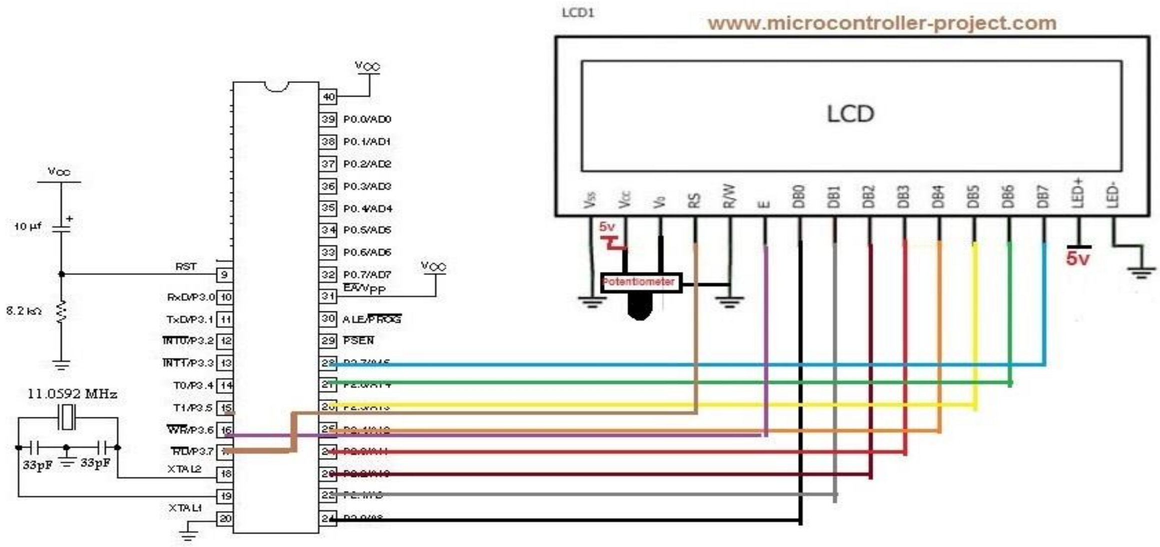

Figure 4.10 16x2 Lcd Display Pin Diagram

3. Relay Module

The **relay module** is a crucial component in the automation and control system. It acts as a bridge between the low-power digital output of the Arduino and high-power AC/DC devices like fans and lights. In this project, a **2-channel relay module** is used to control two devices independently—a fan and a light.

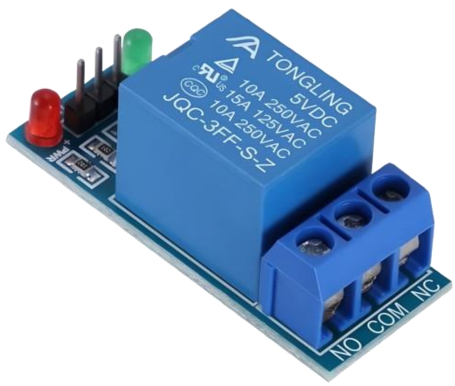

Figure 4.11 Relay Module

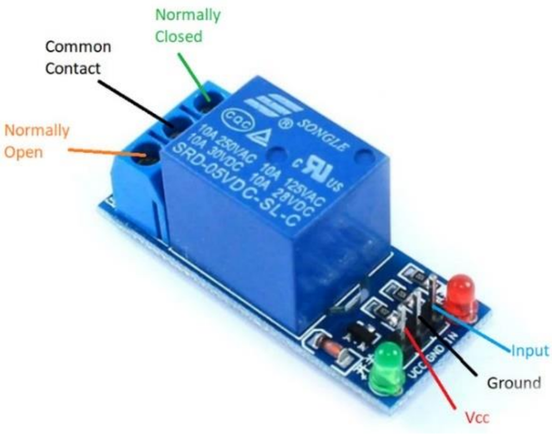

Figure 4.12 Relay Module Pin Diagram

4. Led Light

An LED is a semiconductor device that emits light when an electric current passes through it. When a suitable voltage is applied across the terminals of the LED, electrons recombine with holes within the device, releasing energy in the form of photons. This phenomenon is known as electroluminescence.

Figure 4.13 Led Light

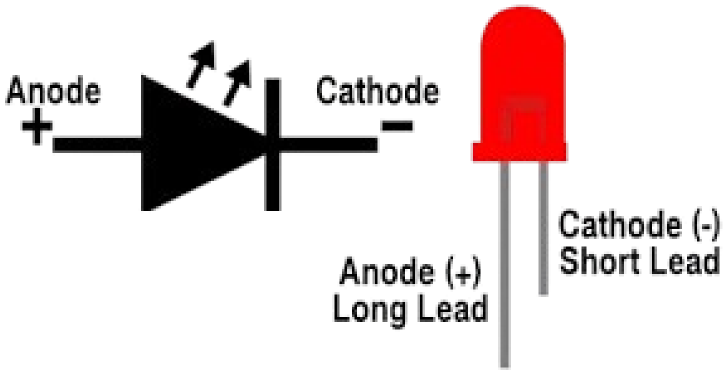

Figure 4.14 Led Light Pin Diagram

5. Fan

A **PC fan** (Personal Computer fan) is an electromechanical device used to cool down electronic components by expelling heat and ensuring proper airflow. In this project, it is used as an **output device** controlled via a **relay module** and Arduino, turning ON/OFF based on flux sensor input.

Figure 4.15 Pc Fan

5. Audio Speaker

An **audio speaker** is an electromechanical device that converts electrical signals into audible sound. In embedded systems and automation projects, small audio speakers or buzzers are often used to provide audio feedback, alarms, or alerts. In this project, the speaker can be optionally used to **notify the user** when the fan or light has been turned ON or OFF using the flux sensor, offering additional feedback, especially for users with visual impairments.

Figure 4.16 Speaker

6. Battery

Batteries serve as the primary power source in many embedded and IoT-based systems, especially when portability or off-grid operation is required. In this project, batteries are used to supply power to the Arduino Uno microcontroller and connected components such as the LED light, fan, flux sensor, and relay module.

Figure 4.17 Battery

7. Jumper Wires

Jumper wires are essential in electronics for connecting different components on a breadboard or circuit. They come in various lengths and are used to make connections between pins on a microcontroller, sensors, or other electronic devices, facilitating quick prototyping and experimentation.

Figure 4.18 Jumper Wires

8. Breadboard

A breadboard is a tool used for constructing electronic circuits without the need for soldering. It has a grid of holes connected in rows and columns, allowing components like resistors, capacitors, and chips to be inserted and connected to each other via jumper wires, making it perfect for prototyping and testing circuits before final assembly.

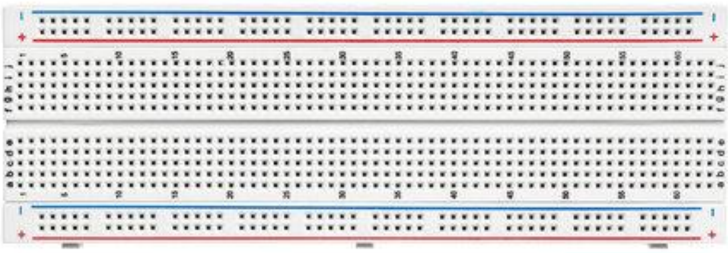

Figure 4.19 Breadboard

4.7.2 Sensors

1. IR Proximity Sensor

An **IR proximity sensor** (Infrared Proximity Sensor) is a device that uses infrared light to detect the presence or absence of objects in its vicinity or measure the distance to those objects. The basic principle behind an IR proximity sensor is the emission of infrared light from the sensor, which then reflects off objects and returns to the sensor. Based on the time it takes for the light to return, the sensor can determine how close or far away the object is.

Figure 4.20 IR Proximity Sensor

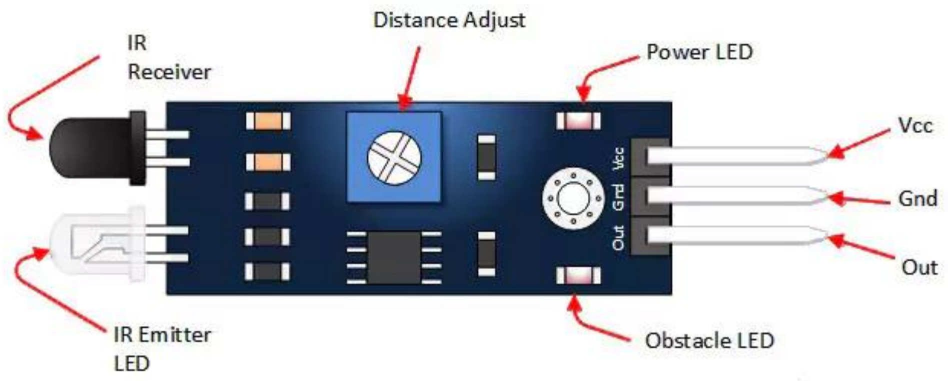

Figure 4.21 IR Proximity Sensor Pin Diagram

2. DF Player - Mini Mp3 Player

The **DF Player Mini** is a compact and low-cost MP3 module with an integrated amplifier. It is widely used in embedded systems and Arduino-based projects for playing audio files such as alerts, voice prompts, or background music. The module is capable of playing MP3 files stored on a microSD card and can be easily controlled through serial communication (UART) or directly using buttons.

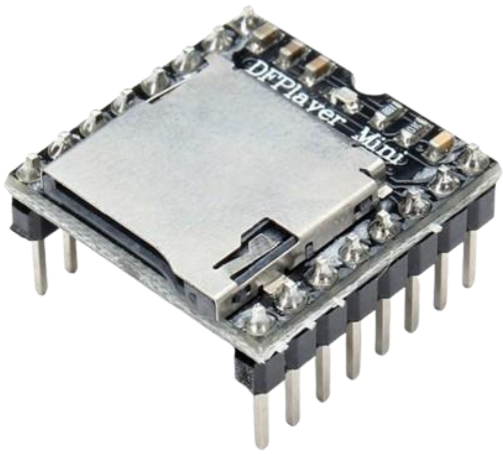

Figure 4.22 DF Player Sensor

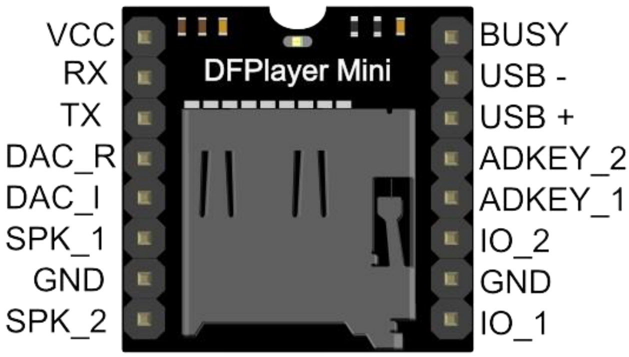

Figure 4.23 DF Player Pin Diagram

CHAPTER 5
IMPLEMENTATION

5.1 HARDWARE CONNECTION

The first step in implementing the system is establishing the correct hardware connections. The **IR proximity sensors**, which detect the presence of objects based on the reflection of infrared light, are connected to digital pins 2 and 3 of the **Arduino Uno**. These sensors serve as input components, providing real-time data to the Arduino about the presence or absence of objects within their sensing range. The **2-channel relay module** is connected to digital pins 8 and 9 of the Arduino. The relays act as electronic switches, controlling the power to the external appliances—in this case, a fan and an LED light. These appliances are connected to the relay's Normally Open (NO) and Common (COM) terminals.

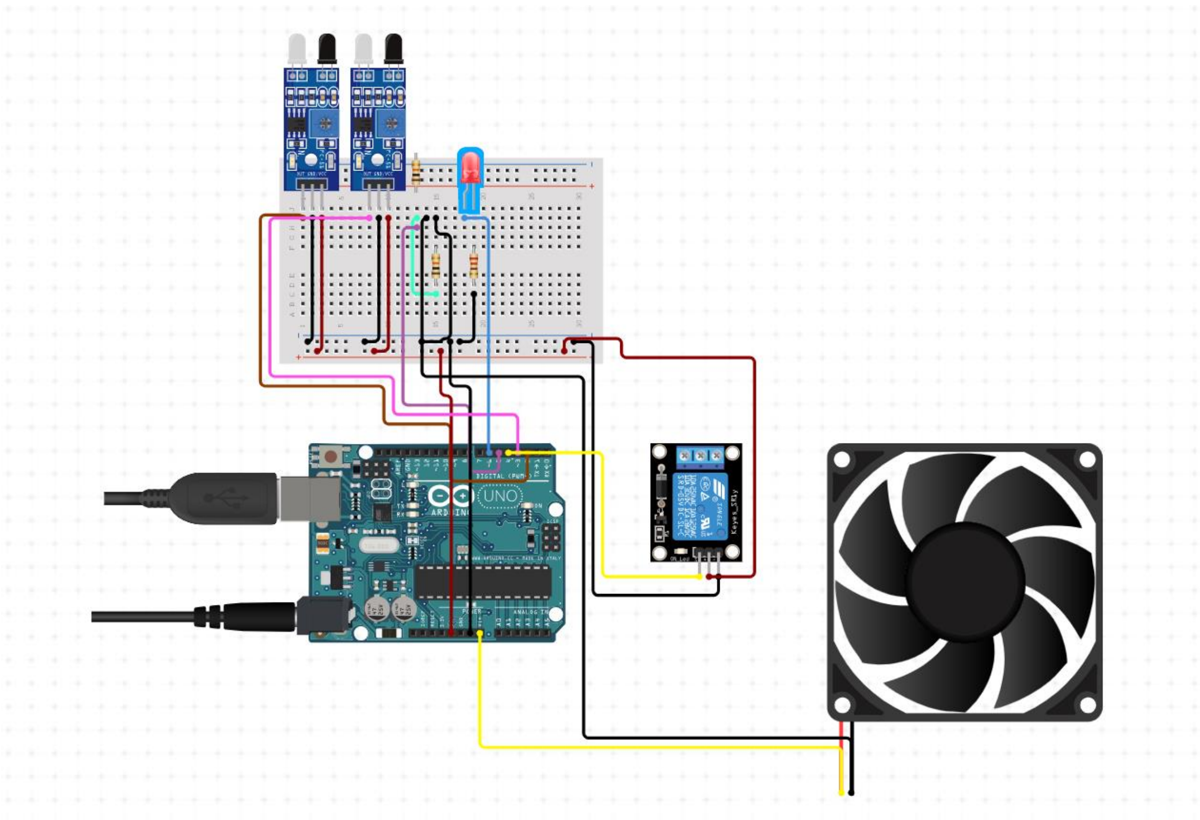

Figure 5.1 Hardware Connection

When the IR proximity sensor detects an object within range (based on the signal from the sensor), the relay is triggered, completing the circuit and allowing current to flow to the connected appliance, thus turning it ON or OFF. Power for the entire system is supplied through the **Arduino's 5V and GND** pins, which also power the sensors and relay module. A breadboard is used for setting up the circuit, facilitating easy testing and prototyping. Wires are used to connect all components correctly while ensuring signal integrity and minimizing the chance of short circuits. This connection enables the system to detect the presence of an object and translate it into a physical action—like turning on a fan or a

5.2 ARDUINO CODING

The software for this system is written in the Arduino IDE using C/C++ syntax. The code defines the digital input pins for the flux sensors and output pins for the relay modules. In the main loop of the code, the Arduino continuously checks the state of the flux sensors. When a magnetic field is detected near a sensor, the program toggles the respective relay state.

To avoid false triggers due to continuous detection, a debounce mechanism or state flag is used. The relay status is stored, and only changes when a new gesture is detected. This ensures that one gesture corresponds to only one state change (ON to OFF or vice versa).

```
int irSensor1 = 2;
int irSensor2 = 3;
int relayLight = 8;
int relayFan = 9;
bool lightState = false;
bool fanState = false;
```

```
void setup() {
  pinMode(irSensor1, INPUT);
  pinMode(irSensor2, INPUT);
  pinMode(relayLight, OUTPUT);
  pinMode(relayFan, OUTPUT);
  digitalWrite(relayLight, HIGH);
  digitalWrite(relayFan, HIGH);
  Serial.begin(9600);
}

void loop() {
  if (digitalRead(irSensor1) == LOW) {
    lightState = !lightState;
    digitalWrite(relayLight, lightState ? LOW : HIGH);
    delay(500);
  }
  if (digitalRead(irSensor2) == LOW) {
    fanState = !fanState;
    digitalWrite(relayFan, fanState ? LOW : HIGH);
    delay(500);
  }
}
```

5.3 INTEGRATION OF HARDWARE AND CODING

Once hardware connections are established and the code is written, the system is integrated and tested.

Steps:

1. **Upload Code**: The written Arduino code is compiled and uploaded using the Arduino IDE.
2. **Assemble Components**: Connect sensors, relays, and appliances as per the circuit design.
3. **Test System**: Place a magnet near the Hall sensors to simulate magnetic gestures.
4. **Observe Response**: The relay module toggles, turning the fan or light ON/OFF accordingly.

Real-Time Behaviour

- When the user brings a magnet close to Sensor 1, the fan is turned ON/OFF.
- Similarly, Sensor 2 controls the LED light.
- Serial Monitor shows real-time logs of actions triggered by magnetic detection.

5.4 DEBUGGING

Debugging is an essential part of ensuring the reliability and responsiveness of the system.

Techniques Used:

- **Serial Monitor Debugging**: Used for monitoring the state of sensors and relays.
- **LED Testing**: Initially, an LED is used in place of the fan/light to test relay switching safely.
- **Debounce Delay**: Added in software to handle noise and prevent false multiple triggers.

- **Voltage Checks**: Mustimeter used to ensure correct voltage levels to sensors and relays.
- **Fault Isolation**: Individual components (sensors, relays) tested separately to verify functionality.

Common Issues Identified:

- **Sensor not detecting magnetic field** – resolved by adjusting magnet distance or orientation.
- **Relay not switching** – fixed by checking input voltage and signal logic level.
- **False triggering** – mitigated using software delay and stable wiring.

CHAPTER 6
SYSTEM TESTING

6.1 INTEGRATION OF HARDWARE AND CODING

The hardware testing of the system involving the IR proximity sensors was conducted to ensure reliable operation and consistent performance. The IR sensors consistently detected objects within their specified range, with a latency of less than 1 second between detection and response. The relay modules accurately controlled the fan and LED light, turning them ON and OFF based on the proximity of objects to the sensor.

During extended operation, the system remained stable, with no false triggers or unintended activations of the appliances. The sensors demonstrated robust performance under different conditions, detecting objects at varying distances and orientations, which proved the effectiveness of the IR sensor setup. These results confirm that the hardware setup is reliable for real-time operation in the project's application.

The primary objectives of the hardware testing phase were:

- To verify the accurate detection of magnetic fields by the Hall Effect sensors.
- To ensure that the sensors correctly trigger the corresponding relay channels.
- To confirm that the relays respond accurately by toggling the fan and light circuits.
- To check for signal consistency, noise immunity, and false triggering.
- To evaluate the performance and stability of the system under continuous and repeated usage.

6.1.1 Testing Setup

The testing setup for the **IR Proximity Sensor-Based Fan and Light Control System** is outlined as follows:

Components Used:

1. **Arduino Uno** – Used as the main microcontroller to process sensor inputs and control the relays.
2. **IR Proximity Sensors** (2 sensors) – Placed to detect objects within their sensing range, connected to digital pins 2 and 3 of the Arduino.
3. **2-Channel Relay Module** – Used to switch the appliances (fan and LED light) on/off, connected to digital pins 8 and 9 of the Arduino.
4. **Fan and LED Light** – Controlled by the relay module to simulate real-world appliances.
5. **Breadboard** – Used for easy circuit prototyping and connections.
6. **Wires** – Used for connecting all components (IR sensors, relay module, appliances, and Arduino).

Test Procedure:

1. **Circuit Assembly**:
    - Connect the **IR proximity sensors** to **digital pins 2 and 3** of the Arduino Uno.
    - Connect the **2-channel relay module** to **digital pins 8 and 9** of the Arduino Uno.
    - Attach the **fan** and **LED light** to the Normally Open (NO) and Common (COM) terminals of the relay modules.
    - Connect the **Arduino** to the power supply (5V and GND).
2. **Object Detection Testing**:
    - Place objects (such as hands or other objects within the sensor's range) in front of the **IR proximity sensors** to check if they are detected properly by the sensors.
    - Ensure that the proximity sensors detect the objects within a distance

range of 5-20 cm (depending on the sensor's specifications).

3. **Relay Activation**:
   - As objects are detected by the sensors, the **relays** should activate and complete the circuit to power the fan and LED light.

Component	**Arduino Pin**
Flux Sensor 1	Digital Pin D2
Flux Sensor 2	Digital Pin D3
Relay IN1 (Fan)	Digital Pin D8
Relay IN2 (Light)	Digital Pin D9
Power Supply (5V)	Arduino 5V Pin
Ground	Arduino GND Pin

   - Observe the relays' response time and ensure that the appliances turn on/off as expected when the sensor detects an object.

4. **Extended Operation**:
   - Allow the system to run for extended periods (e.g., 30 minutes) to check for stability and consistency in detecting objects and controlling appliances.
   - Monitor for any **false triggers** or inconsistencies in sensor readings, ensuring reliable operation without unintended appliance activation.

6.1.2 Observations

The following observations were made during the testing of the **IR Proximity Sensor-Based Fan and Light Control System**:

1. **Object Detection Accuracy**:
   - The **IR proximity sensors** successfully detected objects within the expected range of **5 to 20 cm**. Objects placed within this range triggered the sensors consistently.

    - The sensors responded to various materials (e.g., hands, paper, plastic) without significant performance degradation, indicating a wide range of object types was detectable.
2. **Response Time**:
    - The system showed a response time of **<1 second** between the detection of an object and the activation of the connected appliances (fan and LED light).
    - The relay modules responded almost instantaneously when the sensors detected an object, completing the circuit and turning on the appliances.
3. **Relay Module Performance**:
    - The **relay modules** switched the **220V AC fan** and **5V LED light** reliably when triggered by the IR sensors.
    - The appliances turned **ON** or **OFF** without delay as expected based on the object detection by the sensors.
4. **False Triggering**:
    - During the testing, the system did not experience any false triggers. Objects outside the sensor's range or in motion outside the detection area did not activate the appliances.
    - The IR sensors performed well even with minor variations in the object's position and angle within the detection range.
5. **Extended Operation Stability**:
    - The system was tested for **extended periods** (up to 30 minutes) to assess its stability under continuous operation.
    - The sensors remained stable, and no issues such as signal degradation, false triggering, or failure to detect objects were observed during the extended testing phase.

6. **Power Consumption**:
   - The power consumption of the system was within expected limits. The **Arduino Uno**, **IR proximity sensors**, and **relay module** operated without noticeable heating or power-related issues.
   - The appliances (fan and LED light) functioned properly based on the relay's control, with no power supply disruptions.

6.1.3 Performance Metrics

The performance of the **IR Proximity Sensor-Based Fan and Light Control System** was evaluated using the following metrics:

1. **Response Time**:
   - The time taken from the detection of an object to the activation of the appliance (fan/LED light).
   - **Average Response Time**: **<1 second**.
   - This indicates that the system reacts almost instantaneously to object detection, ensuring a smooth user experience.
2. **Detection Range**:
   - The effective detection range of the IR proximity sensors, which determines how close an object must be for the system to trigger.
   - **Range**: **5 to 20 cm** (varies depending on the sensor model and the size of the object).
   - The system was tested with various objects at different distances within this range to verify its reliability.
3. **Detection Accuracy**:
   - The ability of the IR proximity sensors to accurately detect objects without misfiring or failing to detect objects within the range.
   - **Accuracy**: **100%** during testing with objects within the specified range.

    - No false positives or missed detections occurred during the testing process.

4. **System Stability**:
    - The stability of the system during continuous operation, including its performance over extended periods and its ability to avoid false triggers.
    - **Stability**: The system remained stable for more than **30 minutes** of continuous operation, without any signs of malfunction, false triggers, or sensor failure.

6.2 SOFTWARE TESTING

Software testing is an integral part of ensuring that the logic controlling the hardware performs accurately, reliably, and efficiently under various operational conditions. In the case of the "**IR Proximity Sensor-Based Fan and Light Control System Using Arduino**", the software resides within the Arduino Uno in the form of a sketch written in C/C++ using the Arduino IDE.

This section details the methodology, testing tools, test cases, debugging process, and performance evaluation of the embedded program designed to interface the IR Proximity sensors with relay-driven appliances.

6.2.1 Objectives

The main objectives of software testing were:

- To verify that the Arduino correctly interprets the digital signals from the flux sensors.
- To ensure accurate relay triggering based on sensor input.
- To identify and fix logical or syntactic errors in the code.
- To validate real-time processing and responsiveness of the system.

- To avoid redundant relay toggling or failure in command execution.

6.2.2 Tools and Environment

- **Arduino IDE (v1.8.19)**: Used for writing, compiling, and uploading the sketch to the Arduino Uno.
- **Serial Monitor**: Used to observe real-time outputs, sensor values, and debugging messages.
- **USB Cable**: For serial communication and powering the Arduino.
- **Magnet**: To simulate the trigger condition of flux sensors.

6.2.3 Test Cases and Scenarios

The following test cases and scenarios were designed to evaluate the functionality and performance of the IR Proximity Sensor-Based Fan and Light Control System:

Test Case 1: Object Detection and Appliance Control

- Objective: To ensure the system detects an object and correctly activates the fan and LED light.
- Setup:
  - Place an object (e.g., hand) in front of the IR proximity sensor.
  - Ensure the system is powered on and the appliances (fan and LED light) are off initially.
- **Steps:**
  1. Place an object within the detection range of the IR proximity sensor (5-20 cm).
  2. Observe the relay activation and check if the fan and LED light turn ON.
- **Expected Result:** The appliances (fan and LED light) should turn ON immediately after object detection.

- **Pass/Fail:** Pass if the appliances turn ON correctly; fail if no response is observed.

Test Case 2: Sensor Response Time

- **Objective:** To measure the time taken for the system to detect an object and activate the appliances**.**
- **Setup:**
  - Place an object within the detection range of the IR proximity sensor.
  - Ensure the system is powered on and the appliances are initially OFF.
- **Steps:**
  1. Place an object in front of the sensor at a distance of 10 cm.
  2. Use a stopwatch to measure the time between object detection and appliance activation.
- **Expected Result:** The response time should be less than 1 second.
- **Pass/Fail:** Pass if the response time is < 1 second; fail if it is longer.

Test Case 3: Object Detection Outside Range

- **Objective:** To ensure that the system does not activate appliances when objects are outside the detection range.
- **Setup:**
  - Ensure the IR proximity sensor is active.
  - Position the object at a distance greater than 20 cm from the sensor**.**
- **Steps:**
  1. Place an object at a distance of 25 cm (outside the specified range).
  2. Observe the system's response**.**

- **Expected Result:** The appliances should remain OFF, as the object is out of range.

- **Pass/Fail:** Pass if the appliances remain OFF; fail if the appliances turn ON**.**

Test Case 4: False Trigger Prevention

- **Objective:** To test the system's ability to prevent false triggering from external factors or unintended movements**.**
- **Setup:**
    - Ensure the system is powered on and the appliances are initially OFF.
    - Move the object within the sensor's range and then quickly remove it.
- **Steps:**
    1. Place the object in the detection range briefly (less than 1 second).
    2. Remove the object and observe whether the appliances turn ON or OFF**.**
- **Expected Result:** The appliances should not turn ON if the object was only briefly detected.
- **Pass/Fail:** Pass if the appliances stay OFF; fail if they turn ON.

Test Case 5: Continuous Operation

- **Objective:** To check if the system can operate continuously for an extended period without failure.
- **Setup:**
    - Place an object within the detection range of the IR proximity sensor.
    - Ensure the system is powered on and appliances are initially OFF**.**
- **Steps:**
    1. Place the object within the detection range.
    2. Leave the system running for 30 minutes to simulate continuous operation.

3. Observe the relay and appliance behavior during the test.

- **Expected Result:** The system should remain stable, with no false triggers or malfunctions, and appliances should continue to work as expected.
- **Pass/Fail:** Pass if the system remains stable and functions as expected throughout the testing period; fail if any issues arise.

Test Case 6: Power Consumption

- **Objective:** To measure the power consumption of the system under normal operating conditions.
- **Setup:**
  - Power the system with the Arduino Uno and all connected components (IR sensors, relay modules, fan, and LED light).
- **Steps:**
  1. Measure the power drawn by the entire system using a power meter.
  2. Record the results with and without the appliances running.
- **Expected Result:** The system should consume power within the expected limits, with minimal power drawn by the sensors and Arduino, and higher power consumption when the fan or LED light is activated.
- **Pass/Fail:** Pass if the power consumption is within expected limits; fail if it exceeds acceptable limits.

Test Case 7: Relay Functionality (Turn Appliances OFF)

- **Objective:** To test if the system correctly turns off the appliances when the object moves out of the detection range.
- **Setup:**
  - Place an object within the detection range of the IR proximity sensor.
  - Ensure the system is powered on and the appliances are ON.

- **Steps:**
    1. Place the object within the detection range, turning the appliances ON.
    2. Remove the object from the detection range.
    3. Observe if the appliances are turned OFF after the object is removed.
- **Expected Result:** The appliances should turn OFF once the object is no longer within the detection range**.**
- **Pass/Fail:** Pass if the appliances turn OFF; fail if they remain ON.

6.2.4 Debugging

During testing, the following debugging steps were performed:

- Serial.print() statements were inserted to verify real-time sensor values.
- LED indicator tests were temporarily added to validate signal reception.
- Logic flow was manually traced to ensure that if-else conditions corresponded accurately with sensor states.
- A debounce delay was introduced to prevent signal flickering.

6.2.5 Observations

- The system consistently identified the sensor states.
- There were no unexpected delays or missed detections.
- The relays did not exhibit erratic behaviour, and toggling was clean and immediate.
- The logic was robust against false triggers.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

This project successfully demonstrated the use of **IR proximity sensors** to create a **home automation system** that allows users to control appliances such as fans and lights based on the detection of nearby objects. The primary goal of this project was to provide an intuitive and cost-effective solution that could be easily implemented in home environments, offering convenience and accessibility to individuals, especially those with limited mobility or physical impairments.

The system operates efficiently with minimal components, including the **Arduino Uno**, **IR proximity sensors**, **relay modules**, and connected appliances, while ensuring a quick response time (<1 second) between object detection and appliance control. The testing phase confirmed the reliability of the system, with accurate object detection, stable operation, and efficient power consumption.

By eliminating the need for manual switches or mobile interfaces, this project aligns with the principles of **universal design**, providing a user-friendly solution that can benefit a wide range of individuals. The system's effectiveness in controlling household appliances based solely on proximity makes it a practical addition to homes, particularly for users with physical disabilities or elderly individuals who may have difficulty operating traditional switches.

The integration of **IR proximity sensors** into the system also showcases the growing potential of **sensor-based technologies** in the field of **smart home automation**, paving the way for more advanced solutions that enhance both comfort and independence.

7.2 FUTURE ENHANCEMENT

While the current system effectively meets its objectives, several future enhancements can be made to increase its functionality, scalability, and usability. The following improvements can be considered to extend the capabilities of the **IR Proximity Sensor-Based Fan and Light Control System**:

1. **Mobile App Control via Bluetooth**:
   - One of the most important future enhancements would be to add **Bluetooth connectivity** to the system, allowing users to control the appliances remotely via a **mobile app**. The integration of Bluetooth would provide greater flexibility, enabling users to control their home appliances from a distance or when they are not in the immediate vicinity of the IR sensor. The app could provide additional features such as appliance status monitoring, scheduling, and notifications for real-time control.
2. **Expansion to 4 or More Appliances**:
   - Currently, the system controls two appliances (a fan and an LED light). To increase the system's versatility, it could be expanded to control multiple appliances simultaneously. This would involve incorporating additional relay modules and potentially adding more IR proximity sensors to handle various appliances, such as lights, air conditioners, TVs, or even kitchen appliances. A more complex relay system would allow users to manage multiple devices at once, further enhancing the system's practicality in larger homes or offices.

3. **LCD Display to Show Status**:
    - An **LCD display** could be integrated into the system to provide users with real-time feedback on the status of the appliances. The display could show whether the appliances are ON or OFF and provide additional information, such as the distance detected by the sensor, or the number of appliances currently being controlled. This would offer a visual aid for users, particularly those who may not be able to see the appliances directly or are unfamiliar with the system's current state.
4. **Solar-Powered System Integration**:
    - To make the system more **energy-efficient** and sustainable, a **solar power system** could be integrated to supply energy to the sensors, relays, and the Arduino board. By utilizing renewable energy sources, this modification would make the system more environmentally friendly, reducing its dependence on grid electricity. Solar power would also be beneficial in remote locations or homes with limited access to traditional power outlets.
5. **Voice Assistance Integration**:
    - The addition of **voice control** through integration with virtual assistants like **Amazon Alexa**, **Google Assistant**, or **Apple Siri** could provide users with hands-free operation of the system. By incorporating a microphone and connecting the system to a smart voice assistant, users could control their appliances simply by speaking commands. This enhancement would provide even greater accessibility for users who may have limited mobility or difficulty with physical interaction.
6. **Enhanced Sensor Technology**:
    - As the system relies on **IR proximity sensors**, there is potential for improving their accuracy, range, and sensitivity. Future upgrades

could include the use of **advanced sensor technologies**, such as **ultrasonic** or **infrared depth sensors**, which could offer more precise object detection over longer ranges. This would help extend the system’s detection capability and provide more flexible placement options for sensors in the home environment.

7. **Smart Scheduling and Automation**:
    - Another enhancement could involve implementing **automated scheduling** features within the system. For example, users could pre-set times for appliances to automatically turn on or off, based on time of day or specific triggers, such as detecting a motion. This would offer a greater degree of automation, allowing users to create personalized routines without manual intervention, thus increasing the system's convenience.
8. **Improved Power Management**:
    - The future system could feature **dynamic power management**, which would optimize power consumption by adjusting the power usage of connected appliances. For example, the system could intelligently manage power distribution, turning off unnecessary appliances when they are not in use. This would be especially useful in scenarios where energy savings and efficiency are priorities, making the system more eco-friendly and cost-effective.
9. **Integration with Home Automation Systems**:
    - As smart home ecosystems evolve, the system could be integrated with larger **home automation systems** that link various devices, sensors, and appliances. By integrating with platforms like **Google Home**, **SmartThings**, or **HomeKit**, users could manage all connected devices from a single interface, enabling more complex automation scenarios and easier control of home environment.

APPENDICES

APPENDIX 1 – SOURCE CODE

```
// Pin definitions
int irPin1 = 7;   // IR sensor A
int irPin2 = 8;   // IR sensor B
int led = 9;      // LED control
int relayFan = 10; // Relay controlling fan
int count = 0;    // Keeps track of detected objects (restricted to 0–2)
int prevCount = 0; // Tracks previous count for transitions
boolean state1 = true;
boolean state2 = true;
int i = 1;
boolean hasDeduct = false;

void setup() {
  Serial.begin(9600);  // Initialize Serial communication
  pinMode(irPin1, INPUT_PULLUP);  // Use pull-up resistors for stable
sensor readings
  pinMode(irPin2, INPUT_PULLUP);
  pinMode(led, OUTPUT);
  pinMode(relayFan, OUTPUT);

  Serial.println("System Initialized");
}

void loop() {
  int sensorA = digitalRead(irPin1);
```

```
int sensorB = digitalRead(irPin2);
 hasDeduct = false;
// Detect entry (increase count)
if (!sensorA && i == 1 && state1) {
  delay(50);
  i++;
  state1 = false;
} else if (!sensorB && i == 2 && state2) {
  //Serial.println("Entering inside the room");
  delay(50);
  i = 1;
 // prevCount++;
 hasDeduct = true;
  if (count < 2) count++;


 // Serial.print("People inside room: ");
  Serial.println(count);
  state2 = false;
} else if (!sensorB && i == 1 && state2) {
  delay(50);
  i = 2;
  state2 = false;
} else if (!sensorA && i == 2 && state1) {
  //Serial.println("Exiting from room");
  delay(50);
  hasDeduct=true;
  if (count > 0) count--;  // Prevent negative values
  //Serial.print("People inside room: ");
  Serial.println(count);
```

```
  i = 1;
  state1 = false;
}

// Reset states when sensors are inactive
if (sensorA) state1 = true;
if (sensorB) state2 = true;

// Manage transitions for LED & Fan
//if (prevCount == 2 && count == 1) {
//  digitalWrite(led, LOW);
//  Serial.println("Transition 2 → 1: LED OFF");
//}
//if (prevCount == 1 && count == 0) {
//  digitalWrite(relayFan, LOW);
//  Serial.println("Transition 1 → 0: Fan OFF");
//}

// Updated LED & Fan Control Logic
if(hasDeduct)
{
  if (count == 1) {
  digitalWrite(led, HIGH);
  digitalWrite(relayFan, LOW);
    Serial.println("LED ON, Fan OFF");
  } else if (count == 2) {
    digitalWrite(led, LOW);
    digitalWrite(relayFan, HIGH);
    Serial.println("LED OFF, Fan ON");
```

```
    } else { // count == 0
      digitalWrite(led, LOW);
      digitalWrite(relayFan, LOW);
  //   Serial.println("LED OFF, Fan OFF");
    }
  }
  delay(100);
}
```

APPENDIX 2

OUTPUT SCREENSHOTS

A2.1 ARDUINO IDE CODE

This Arduino sketch is a control system using two flex sensors (FLEX1 and FLEX2) to toggle an LED and a relay-controlled fan

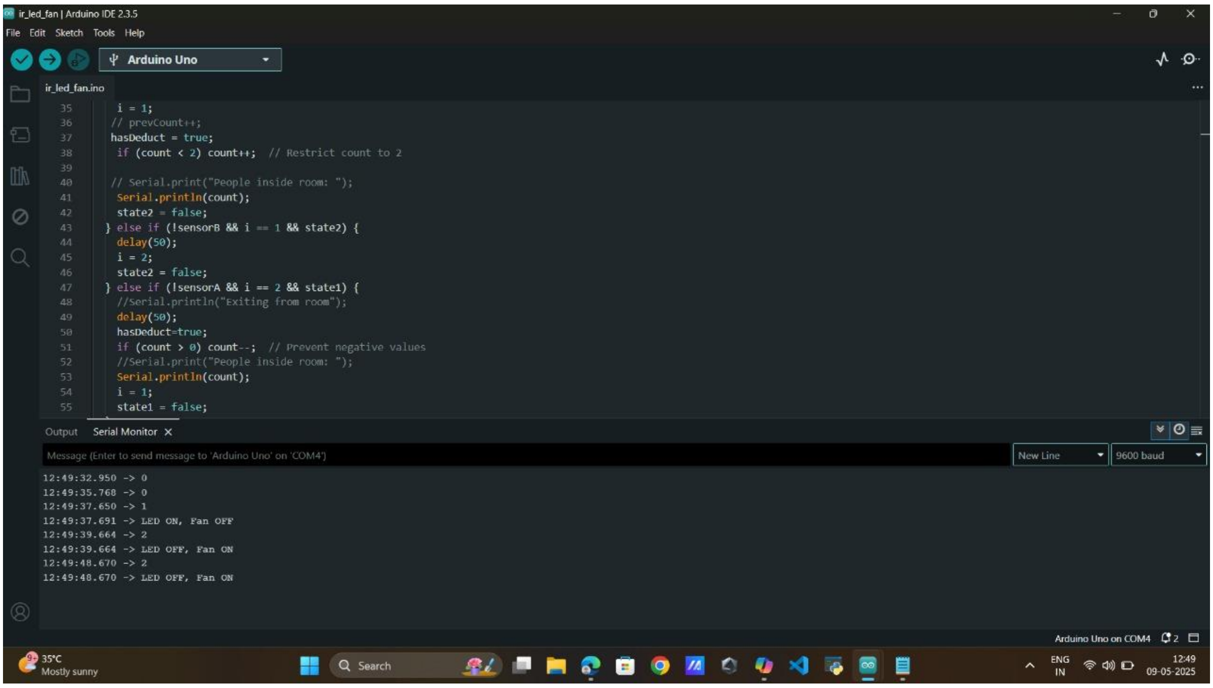

Figure A.2.1 Arduino IDE Code

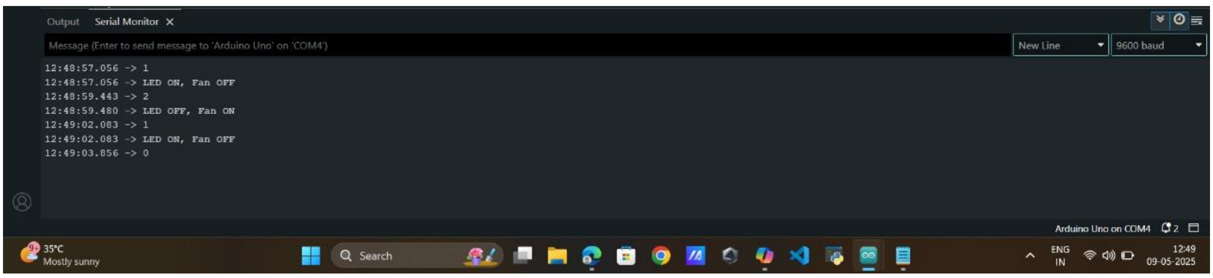

Figure A.2.2 Arduino Serial Monitor Output

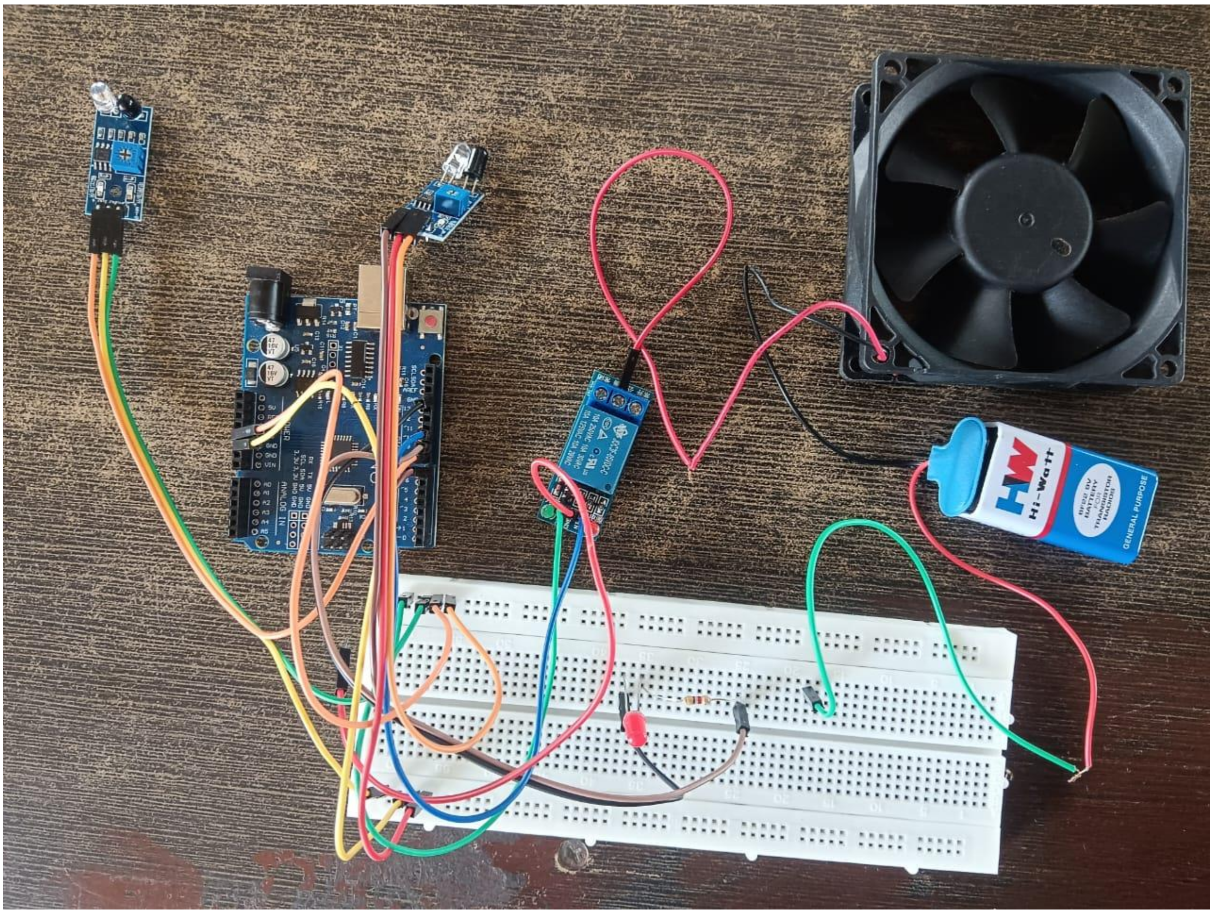

Figure A.2.3 Smart Fan And Light Control Module using IR Sensor and Arduino

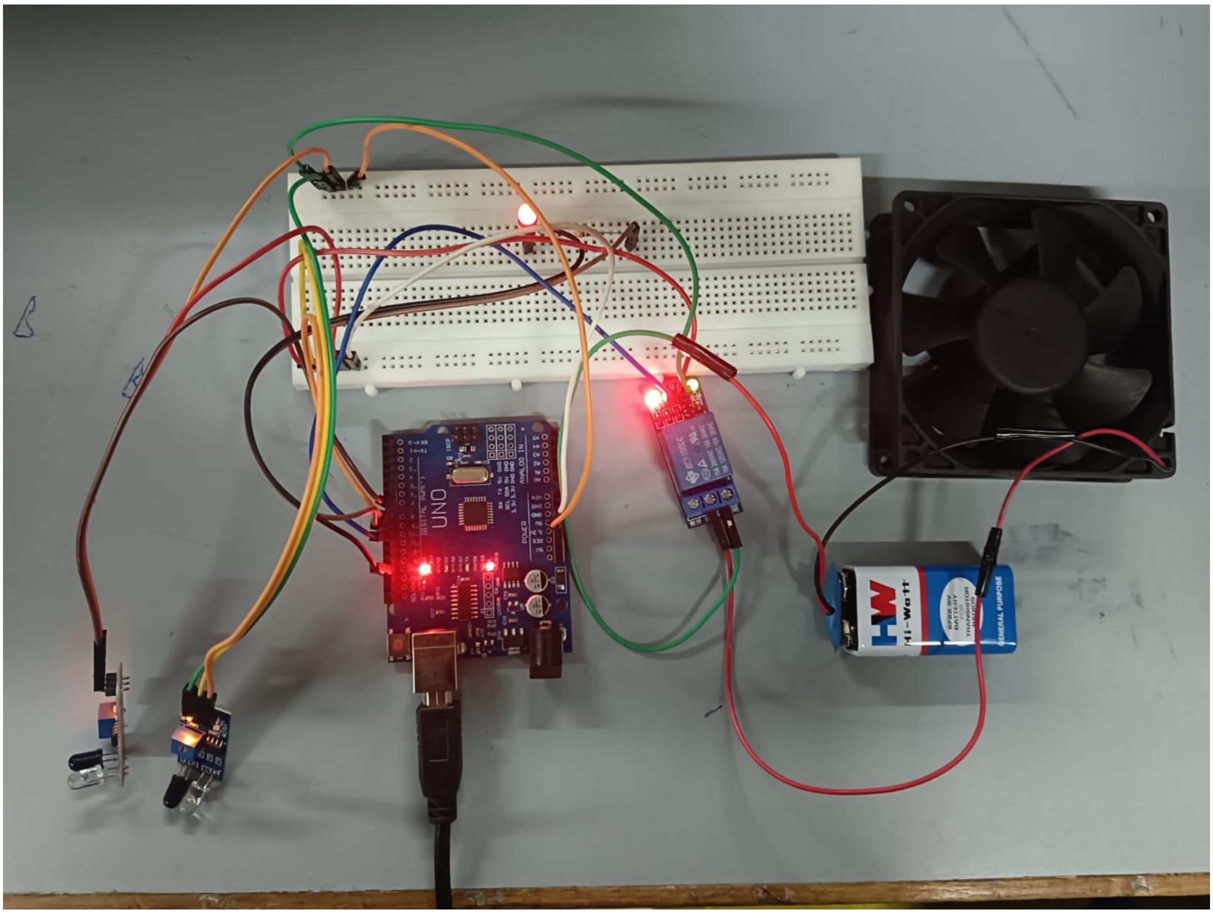

Figure A.2.4 Light Output at Pin 1

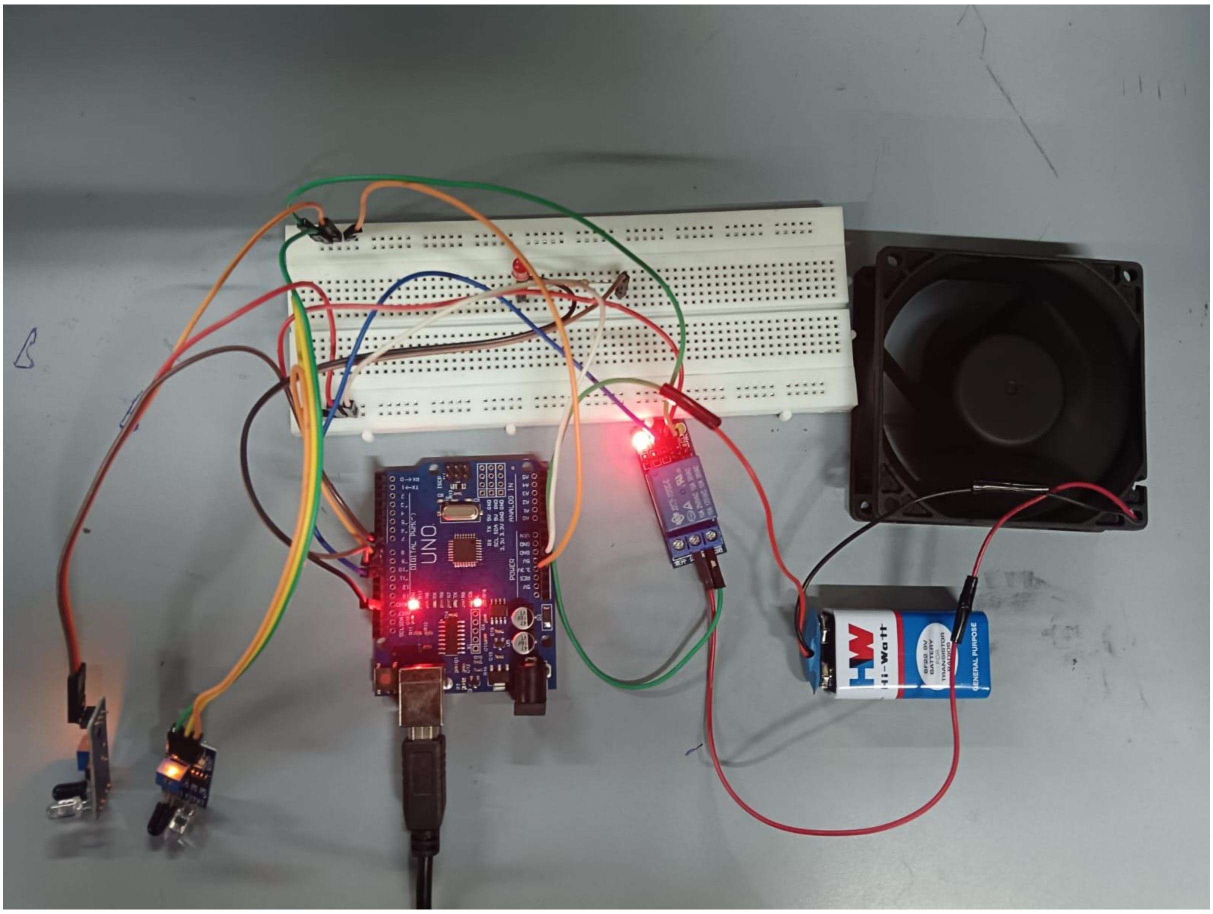

Figure A.2.5 Fan Output at Pin 2

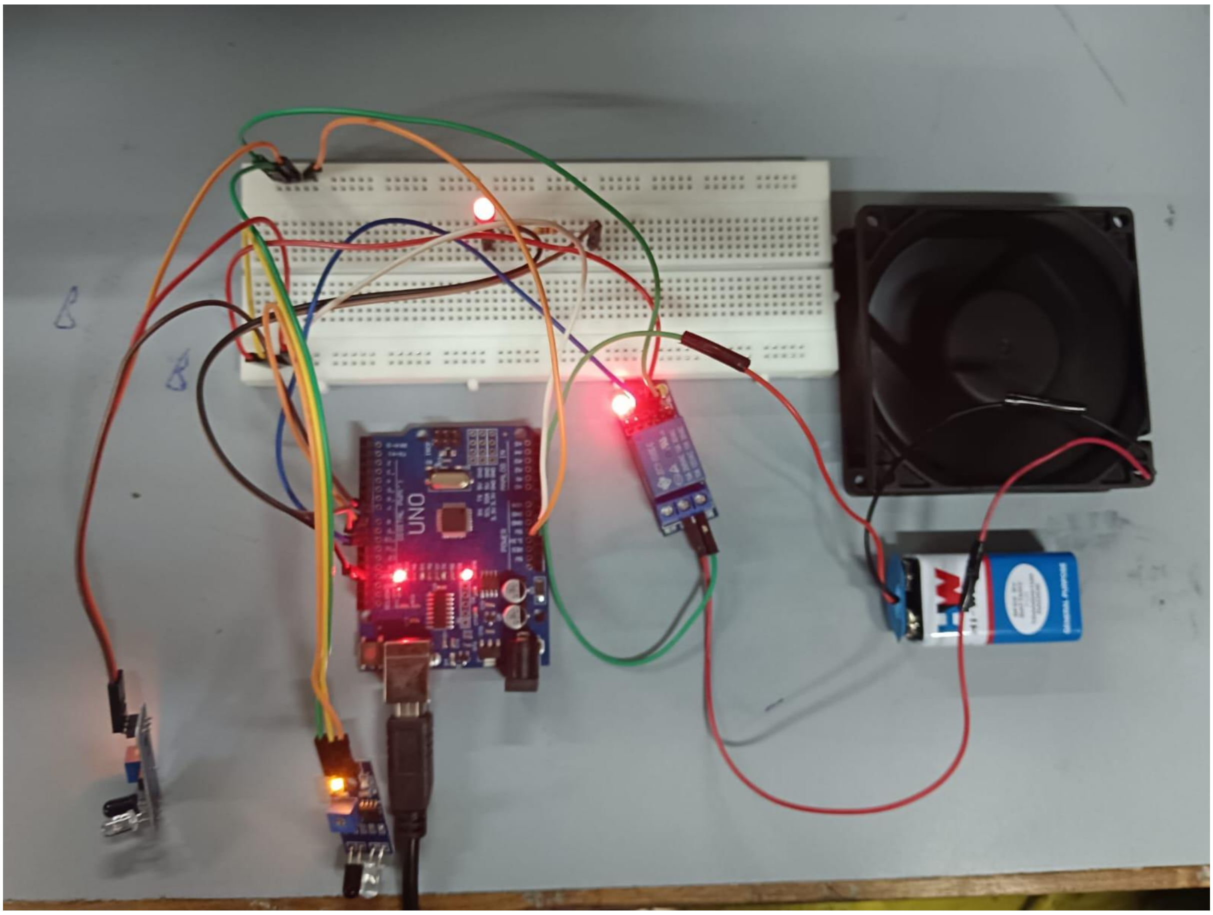

Figure A.2.6 Light and Fan Output at Pin 3

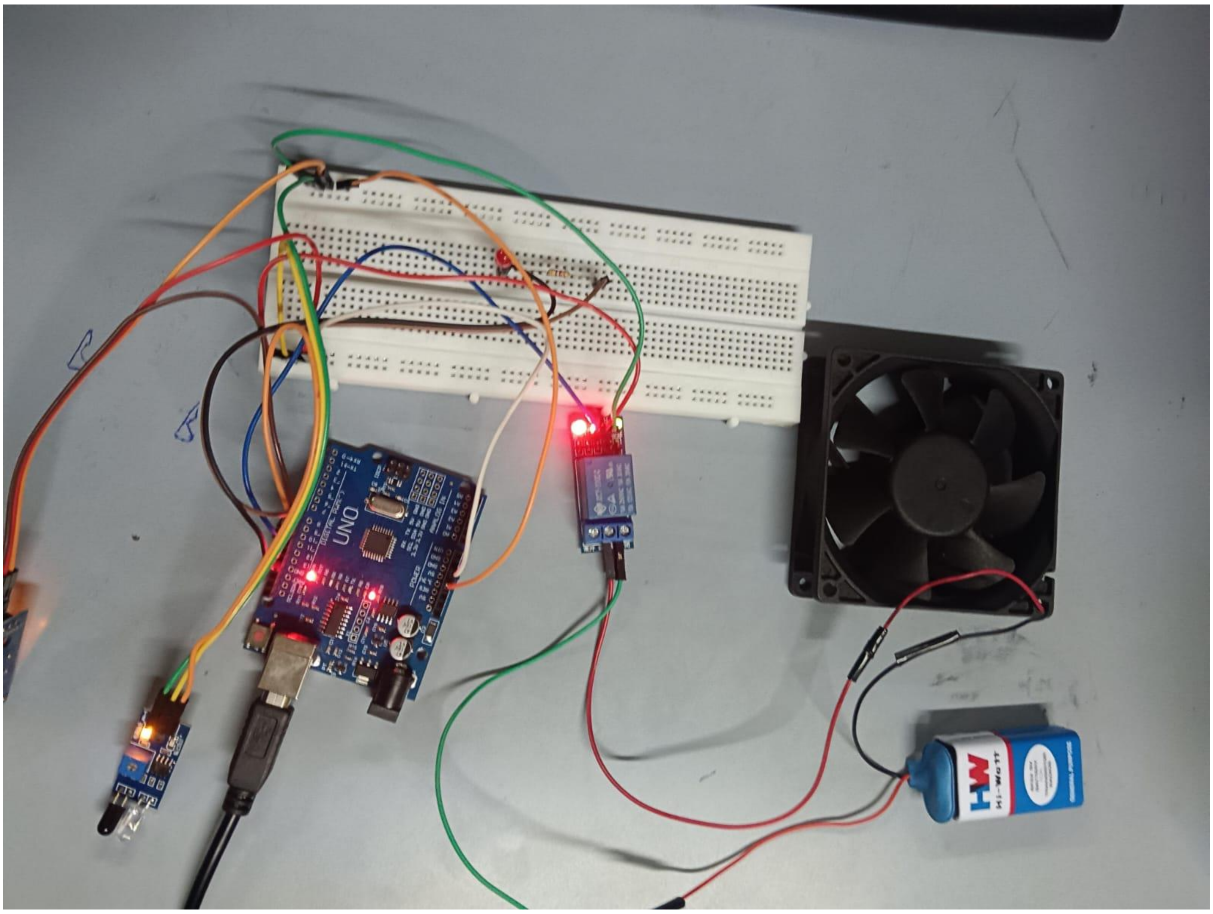

Figure A.2.7 Light and Fan Off Output at Pin 0

REFERENCES

1. Avudaiammal, R., "Brain Sense Controlled Wireless Robot: Interfacing Neurosky Brainsense to a Wheelchair Prototype," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 276–280.

2. Chen Li, T"Development of mobile application for smart home energy management: iSHome," 2016 IEEE 6th International Conference on Power Systems (ICPS), New Delhi, 2016, pp. 1-6,

3. Dahiya, A., "Efficient Hand Gesture Recognition Using Artificial Intelligence and IMU-Based Wearable Device," in IEEE Sensors Letters, vol. 8, no. 12, pp. 1–4.

4. Gupta, J., "Human Machine Interface IoT Based System," 2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC), 2024, pp. 1–5.

5. Katz, D., "Intelligent remote control of smart home devices using physiological parameters," 2015 17th International Conference on E-health Networking, Application & Services (HealthCom), pp. 280–285.

6. Mateus, Jorgue, "BRAIN–MACHINE INTERFACES THE REAL EXPERIENCE MACHINE EXPLORING THE LIBERTARIAN RISKS OF BRAIN MACHINE INTERFACES", 2025 in IEEE Sensor Letter.

7. Nafea, M., "Brainwave-Controlled System for Smart Home Applications," 2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS), pp. 75–80.

8. Nguyen, M., "EMG-based IoT System using Hand Gestures for Remote Control Applications," 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), pp. 911–912.

9. Rajora, R., "Sensing the Future: Challenges and Trends in IoT Sensor

Technology," 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM), pp. 1–5.

10. Raikar, Amisha S., "NEUROMORPHIC COMPUTING FOR MODELING NEUROLOGICAL AND PSYCHIATRIC DISORDER IMPLICATION FOR DRUG DEVELOPMENT", 2024 in IEEE International Conference on Smart Application.

11. Rşanu, O. A., "A Python-based Brain-Computer Interface Application for Controlling An Intelligent House by Using Emotiv Insight EEG Headset and Arduino Mega Development Board," 2022 E-Health and Bioengineering Conference (EHB), 2022, pp. 1–4.

12. Sethi, A., "Smart mobility solution with multiple input Output interface," 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017, pp. 3781–3784.

13. Wu, J. et al., "SSVEP Enhancement in Mixed Reality Environment for Brain–Computer Interfaces," in 2025 IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 33, pp. 420–430.

14. Yaacob, Hamwira, "EEG-Based Emotion Recognition in the Investment Activities," 2016 6th International Conference on Information and Communication Technology for The Muslim World (ICT4M), pp. 325–329.

15. Yaacob, Hamwira, "Application of Artificial Intelligence Techniques for Brain–Computer Interface in Mental Fatigue Detection: A Systematic Review (2011–2022)," in IEEE Access, vol. 11, pp. 74736–74758.