**A REPORT ON**

**HOME SECURITY USING IOT TECHNOLOGY**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

OF

**PROJECT BASED LEARNING (FIRST YEAR ENGINEERING)**

##### SUBMITTED BY

**EBHA MOLLICK** Exam No: F190220151

**JAGDISH SINGH** Exam No: F190220187

**MOHIT SHARMA** Exam No: F190220251

**RAMKISHAN GUPTA** Exam No: F190220358



## DEPARTMENT OF ASGE

## ARMY INSTITUTE OF TECHNOLOGY

DIGHI HILLS, ALANDI ROAD, PUNE 411015

## SAVITRIBAI PHULE PUNE UNIVERSITY

## 2023 -2024



**CERTIFICATE**

This is to certify that the project report entitles

**“HOME SECURITY USING IOT TECHNOLOGY”**

Submitted by

**EBHA MOLLICK** Exam No: F190220151

**JAGDISH SINGH** Exam No: F190220187

**MOHIT SHARMA** Exam No: F190220251

**RAMKISHAN GUPTA** Exam No: F190220358

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Prof. Anjali Hudedamani** and it is approved for the partial fulfillment of the requirement of , First year course on Project Based learning of Savitribai Phule Pune University.

**(Prof. Anjali Hudedamani)** **(Prof. Dr S A Kulkarni)**

Guide Head,

Department of ASGE Department of ASGE

**(Dr. B.P.Patil)**

Principal,

Army Institute of Technology, Dighi, Pune – 411015

Place : Pune

Date :

**ACKNOWLEDGEMENT**

First and foremost, we would like to express our gratitude to our Mentor, Prof. Anjali Hudedamani, who was a continual source of inspiration. She pushed us to think imaginatively and urged us to do this homework without hesitation. Her vast knowledge, extensive experience, and professional competence enabled us to successfully accomplish this project. This endeavour would not have been possible without her help and supervision. We could not have asked for a finer mentor in our studies. This initiative would not have been a success without the contributions of each and every individual. We were always there to cheer each other on, and that is what kept us together until the end.

To all the individuals and resources that have contributed to our project, whether through guidance, feedback, or assistance, we offer our sincerest thanks. Your contributions have enriched our project and made it more robust.

As we conclude this project, we carry forward the lessons learned and the bonds forged, knowing that the support and guidance we have received will continue to inspire and guide us in future endeavours.

**NAME OF THE STUDENTS**

EBHA MOLLICK

JAGDISH SINGH

MOHIT SHARMA

RAMKISHAN GUPTA

**ABSTRACT**

This project presents a home security system leveraging IoT (Internet of Things) technology to protect homes from theft. The system aims to deter potential burglars by simulating the presence of occupants inside the house, thus reducing the risk of break-ins. The system also includes scheduling capabilities for appliances to create a more realistic illusion of occupancy.

The problem addressed by this project is the increasing rate of home burglaries and the need for effective and affordable security solutions. Traditional security measures such as alarms and cameras can be expensive and may not always be effective in deterring burglars. This project offers a cost-effective alternative by using IoT devices to create a smarter, more responsive home security system.

The solution proposed in this project involves the use of IoT devices such as ESP32, Arduino, smart lights, and smart plugs to create a simulated occupancy effect. When the system detects suspicious activity, such as movement around the house when no one is supposed to be home, it activates lights and appliances to give the impression that someone is inside. Additionally, the system allows users to schedule the activation of lights and appliances to mimic their daily routines, further enhancing the illusion of occupancy.

In conclusion, this project demonstrates the effectiveness of IoT technology in enhancing home security. By creating a system that can simulate occupancy and respond to potential threats, this project offers a practical and affordable solution to improve home security and deter burglars.

**[P3]**

**TABLE OF CONTENTS**

LIST OF ABBREVATIONS i

LIST OF FIGURES ii

LIST OF TABLES iii

# CHAPTER TITLE PAGE NO.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No.** | | | | **Title of Chapter** | | | | **Page No.** |
| **01** | | | | **Introduction** | | | | 6-10 |
|  | 1.1 | | | Overview | | | | 6 |
|  | 1.2 | | | Motivation | | | | 6 |
|  | 1.3 | | | Problem Definition and Objectives | | | | 7 |
|  |  | | | 1.3.1 | | | Problem Definition | 7 |
|  |  | | | 1.3.2 | | | Objectives | 7 |
|  | 1.4 | | | Project Scope & Limitations | | | | 7-8 |
|  |  | | | 1.4.1 | | Project Scope | | 7 |
|  |  | | | 1.4.2 | | Limitations | | 8 |
|  | 1.5 | | | Methodologies of Problem solving | | | | 9 |
| **02** | | | | Literature Survey | | | | 11-20 |
| **03** | | | | Increase Home Security Using IoT | | | | 21-27 |
|  | | 3.1 | | Diagram | | | | 21 |
|  | | 3.2 | | Working | | | | 21 |
|  | | 3.3 | | Code | | | | 22 |
|  | | 3.4 | | Team Organization | | | | 27 |
|  | |  | | 3.4.1 | Task Network | | | 27 |
| **04** | | | | Results | | | | 28 |
|  | | | 4.1 | Outcomes | | | | 28 |
|  | | | 4.2 | Images | | | | 28 |
| **05** | | | | Conclusions | | | | 30-31 |
|  | | 5.1 | | Conclusions | | | | 30 |
|  | | 5.2 | | Future Work | | | | 30 |
|  | | 5.3 | | Applications | | | | 31 |

1. **INTRODUCTION**

**1.1 OVERVIEW**

This project explores the innovative use of IoT devices to develop a smart home security system that can deter burglars by creating the illusion of occupancy. By leveraging IoT devices such as motion sensors, smart lights, and smart plugs, this system can simulate the presence of occupants inside the house, even when it is empty. The key feature of this system is its ability to schedule the activation of lights and appliances based on the user's daily routine. This scheduling not only enhances the illusion of occupancy but also allows for greater customization and control over the security measures.

Additionally, the system can be easily integrated with existing home automation systems, allowing for seamless control and monitoring of the entire home environment.

Overall, this project aims to showcase the potential of IoT technology in improving home security. By providing a cost-effective and efficient alternative to traditional security systems, this system offers homeowners peace of mind knowing that their homes are protected, even when they are not there.

**1.2 MOTIVATION**

The motivation behind this project stems from the pressing need for a more accessible and efficient home security solution. Traditional security systems can be prohibitively expensive and often require extensive installation and maintenance. Additionally, these systems may not always be effective in deterring burglars or providing timely alerts to homeowners.

By leveraging IoT technology, we aim to address these challenges and provide a smarter, more responsive security solution. IoT devices are becoming increasingly affordable and easier to integrate, making them an ideal choice for modernizing home security.

Moreover, the flexibility of IoT devices allows for greater customization and control, empowering homeowners to tailor their security measures to suit their specific needs. This project seeks to harness the potential of IoT technology to enhance home security, providing homeowners with a cost-effective and reliable solution that offers peace of mind and greater protection for their homes.

**1.3 PROBLEM DEFINITION AND OBJECTIVES**

1.3.1 Problem Definition

The primary problem addressed by this project is the need for an effective and affordable home security solution. Traditional security systems can be expensive to install and maintain, making them inaccessible to many homeowners. Additionally, these systems may not always be effective in deterring burglars or providing timely alerts to homeowners.

Another key issue is the lack of flexibility in traditional security systems. They are often limited in their ability to adapt to different situations or user preferences. This can lead to a false sense of security and leave homes vulnerable to break-ins.

1.3.2 Objectives

The main objective of this project is to develop a smart home security system using IoT technology that addresses these challenges. The system will be designed to simulate the presence of occupants inside the house, even when it is empty, by scheduling the activation of lights and appliances based on the user's daily routine.

Additionally, the system will be user-friendly, with a simple interface that allows users to easily set up and manage their security settings. It will also be scalable, allowing for easy integration with existing home automation systems.

Overall, the objectives of this project are to provide homeowners with a cost-effective and efficient home security solution that enhances the safety and security of their homes.

**1.4 PROJECT SCOPE & LIMITATIONS**

1.4.1 Project Scope

The scope of this project includes the design and implementation of a smart home security system using IoT technology. The system will focus on simulating occupancy to deter burglars, achieved through the scheduling of lights and appliances based on the user's daily routine.

The project will involve the selection and integration of IoT devices such as motion sensors, smart lights, and smart plugs. These devices will communicate with a central hub or controller, which will manage the scheduling and activation of the devices.

The system will also include a user interface, which will allow homeowners to easily set up and manage their security settings. The interface will provide real-time feedback on the status of the system and allow users to customize their security settings as needed.

1.4.2 Limitations

Some potential limitations of the home security system using IoT technology include:

1. Reliability: The system's reliability depends on the stability of the internet connection and the proper functioning of all connected devices. Any disruptions in connectivity or device malfunction could compromise the system's effectiveness.
2. Security Concerns: IoT devices are vulnerable to hacking and security breaches. If the system is not adequately secured, it could be accessed by unauthorized individuals, potentially leading to privacy violations or security threats.
3. Cost: Implementing a comprehensive IoT-based home security system can be expensive, especially if it involves purchasing multiple devices and sensors. Additionally, there may be ongoing costs for maintenance and subscription services.
4. Compatibility Issues: Integrating different IoT devices from various manufacturers can be challenging due to compatibility issues. Ensuring that all devices work seamlessly together may require additional effort and technical expertise.
5. Limited Range: Some IoT devices, such as motion sensors or cameras, may have limited range, which could restrict their effectiveness in larger homes or outdoor areas.
6. Power Outages: The system may be vulnerable to power outages, especially if it relies on electricity to function. Implementing backup power sources, such as batteries, could mitigate this limitation.
7. Privacy Concerns: IoT devices collect and transmit data, raising concerns about privacy. Users should be aware of how their data is being used and take steps to protect their privacy.

Despite these limitations, with proper planning, implementation, and maintenance, an IoT-based home security system can provide significant benefits in terms of security, convenience, and energy efficiency.

**1.5 METHODOLOGIES AND PROBLEM SOLVING**

This project employs a systematic approach to problem-solving, incorporating several key methodologies to address the challenges of creating a smart home security system using IoT technology.

1. Research and Analysis: The project begins with thorough research and analysis of existing IoT-based home security systems. This includes studying the latest trends, technologies, and best practices in the field. The goal is to identify key features and functionalities that can be incorporated into the new system.

2. Requirement Gathering: The next step is to gather requirements from stakeholders, including homeowners, security experts, and IoT enthusiasts. This involves conducting surveys, interviews, and focus groups to understand their needs, preferences, and concerns regarding home security.

3. System Design: Based on the research and requirements gathered, the system is designed using a combination of hardware and software components. This includes selecting appropriate IoT devices, designing the system architecture, and developing algorithms for simulating occupancy and scheduling appliances.

4. Prototyping: A prototype of the system is developed to test its feasibility and functionality. This involves assembling the selected IoT devices, integrating them with the software, and conducting initial tests to ensure that the system can simulate occupancy effectively.

5. Testing and Validation: The prototype is subjected to rigorous testing to identify and fix any bugs or issues. This includes testing the system's ability to detect suspicious activity, simulate occupancy, and communicate with the user interface.

6. User Interface Design: A user-friendly interface is designed to allow homeowners to easily set up and manage their security settings. The interface provides real-time feedback on the status of the system and allows users to customize their security settings as needed.

7. Integration and Deployment: Once the system is tested and validated, it is integrated into the home environment and deployed for real-world use. This involves installing the IoT devices, configuring the system, and providing training to homeowners on how to use the system effectively.

8. Monitoring and Maintenance: After deployment, the system is monitored regularly to ensure that it is functioning correctly. Any issues that arise are addressed promptly through maintenance and updates to the system.

1. **LITERATURE SURVEY**

**2.1 IOT BASED HOME AUTOMATION USING ESP8266 & GOOGLE HOME**

The integration of Internet of Things (IoT) technology into homes has the potential to transform ordinary houses into smart homes, offering increased automation and safety. This paper aims to develop a voice/text-controlled home application that enables users to control their appliances remotely. Users can send text messages or voice instructions to turn appliances ON or OFF based on their needs, even when they are not physically present. The application also allows users to schedule the appliances' status and receive details regarding previous schedules. The implementation of this application utilizes low-cost, high-performance technology, making it accessible to a wide range of users. The system is particularly beneficial for the elderly and individuals with disabilities who may find it challenging to physically operate appliances. Keywords— IoT, Sinric Pro, Google Home, Google Assistant, Relay, Smart control, NodeMCU, Blynk, Alexa, IFTTT

2.1.1 Introduction

Automation, which involves controlling devices or processes with minimal human intervention, has made significant advancements in home environments, leading to the concept of home automation. Today, accessing and managing electrical equipment in homes has become more convenient and efficient, thanks to technological advancements. Homeowners can now control various aspects of their homes with just a few clicks, such as opening and closing doors or turning on the fan. Several technologies, including Bluetooth, ZigBee, Infrared Remote (IR) controllers, and GSM-based technology, have been utilized to construct home automation systems. However, these technologies have limitations, such as limited spectrum availability and range issues, which can lead to electrical mishaps. Hence, there is a need for a reliable method to remotely control home appliances. This study demonstrates how an IoT-based home automation system can be implemented using Wi-Fi technology to address these challenges.

2.1.2 System Methodology

A. Home Automation Equipment: This study recommends using inexpensive, easy-to-implement, and energy-efficient options. Several low-cost and low-power solutions for home automation with NodeMCU, IFTTT, Blynk, Sinric Pro, Google Home, and Amazon Alexa have been suggested. The recommendations from NodeMCU, Sinric Pro, and Google Home are intended to save money at home. The proposed components include:

- NodeMCU: A board based on the ESP8266 chip with Wi-Fi connectivity (802.11 b/g/n) and 17 digital pins.

- Sinric Pro: A home automation platform that allows controlling Raspberry Pi, ESP8266, ESP32, or Arduino boards, which can be linked to Amazon Alexa or Google Home for free.

- Google Home Mini: A smart speaker integrated with Google's personal assistant services through voice recognition.

- Echo Dot: An Amazon-powered smart speaker integrated with the Amazon Alexa mobile app.

- Relay Module: An electromechanical device used to switch circuits, allowing any electrical appliance in the house to be turned on or off.

B. Procedure Sinric Pro, NodeMCU, and App Google Home: The setup involves creating an account in Sinric Pro, adding three switch-type devices to the platform, turning on notifications, including Sinricpro libraries in the Arduino IDE, adding libraries for creating a time system using an NTP client, connecting the NTPClient of Advanced and the Sinricpro of MultiSwitch\_advance programs, changing the password and WIFI SSID to match the home wifi network, and placing the credentials of the APP\_KEY, APP\_SECRET, and deviceID.

C. Working Procedure: The system operates by recognizing voice commands or instructions from Google Home/Alexa, which change the status of the Sinric Pro linked device. The Node MCU ESP8266 microcontroller, equipped with a Wi-Fi module, determines which sets should be on. When the load is switched from '0' to '1', the relay is activated, turning on the appliance. Manual switches are connected for manual control, and the NodeMcu input voltage is 5v.

D. Google Home Control: The Google Home app can be used for virtual switch control and Google Assistant control, offering voice-activated control over linked devices.

E. Hardware Setup: The central processing unit is the ESP8266, powered by a mobile phone charger for a 5V input. The ESP8266 is connected to relay modules, which control the appliances based on the NodeMCU's high and low signals.

2.1.3 System Design and Simulation

The circuit uses the 4-channel relay module to control appliances, with manual switches connected to control relays manually. The relevant relay activates when the control pins receive a LOW signal and deactivates with a HIGH signal. The simulation uses an AC load with a DC source for Arduino pins.

2.1.4 Hardware Development and Testing

A. Power Supply: A mobile phone charger is repurposed as a power supply for the NodeMCU, providing a 5V regulated DC supply. This approach promotes cost-effectiveness and ease of integration into various projects.

B. Load: Electrical appliances requiring a 220/240V 50Hz power source are connected to the system for remote management.

C. Manual Control: Manual switches are integrated as a backup option and for manual control of the load. This offers users flexibility and control, even in situations of technical problems or outages.

D. Google Home Control: Google Home offers voice and virtual switch control, adding convenience to the project.

E. Hardware Setup: The central processing unit is the ESP8266, connected to relay modules for controlling appliances based on signals from the NodeMCU.

2.1.5 Result and Discussion

The prototype design and Google Home setup are shown, with successful testing of light control using voice commands. The data table shows successful control of bulbs and fans using manual, virtual, and voice commands.

2.1.6 Conclusion

This paper presents the implementation of a cloud-based home automation system, allowing remote control of electrical equipment. The system is user-friendly and can be easily installed in real houses, enhancing home automation and control over appliances.

**2.2 DESIGNING AND IMPLEMENTATION OF HOME AUTOMATION SYSTEM BASED ON REMOTE SENSING TECHNIQUE WITH ARDUINO UNO MICROCONTROLLER**

This paper presents a design and implementation of a remote sensing-based home automation system using the Arduino Uno Kit as the main controller. The system allows users to control home devices remotely using a smartphone with remote sensors and internet technologies. It offers two operational scenarios: a manual scenario, where users control devices from anywhere using a smartphone, and an automatic scenario, where devices are monitored and controlled automatically. The system utilizes the Matlab GUI platform as an interface for easy control and monitoring. Overall, the proposed system is simple, cost-effective, and flexible, making it suitable for smart home applications.

2.2.1 Introduction

The focus on saving energy while maintaining a luxurious lifestyle has led to increased interest in developing optimal smart home automation systems. These systems aim to improve people's lifestyles by offering flexibility, affordability, and ease of implementation and use. Modern smart home concepts are based on home automation systems that utilize various technologies, including Bluetooth, GSM, RF ZigBee, etc., for remote controlling household devices. However, many of these systems have limitations such as limited control range or the need for a PC as an interface.

This paper proposes a remote sensing home device management scheme that overcomes these limitations. The system is designed to be flexible, easy to use, and simple to implement with both hardware and software interfaces. The system consists of a PC home server and an Arduino Uno microcontroller board connected through a USB cable. The PC runs the Matlab-GUI platform for management, while the Arduino Uno controls the devices and communicates with the sensors. This system architecture allows users to remotely access and control home appliances through a smartphone, offering greater convenience and flexibility.

2.2.2 System Overview

The proposed home automation system consists of two main hardware components: the PC home server and the Arduino Uno microcontroller board. The Arduino Uno board provides various digital and analog inputs, a serial interface, and digital and PWM outputs. It is connected to the PC through a USB cable and can be programmed using free software. The system architecture is illustrated in Figure 1, showing how the PC home server and Arduino Uno work together to control and monitor home appliances remotely.

The system offers two scenarios: a manual scenario and an automatic scenario. In the manual scenario, users can control devices remotely using the Matlab-GUI software interface on their smartphone. They can turn devices ON or OFF by pressing buttons in the interface. In the automatic scenario, the Arduino Uno kit manages the home devices based on signals from sensors, allowing for automatic control without user intervention. The Matlab-GUI interface provides a user-friendly way to monitor and control the system from anywhere in the world.

2.2.3 Hardware Implementation

To validate the proposed system's performance, a hardware implementation was carried out. The manual scenario implementation allows users to remotely control devices using ON or OFF buttons in the Matlab-GUI software interface. The automatic scenario implementation involves sensors sending signals to the Arduino Uno kit, which then controls the home devices based on predefined conditions.

For example, in the automatic scenario, the temperature sensor TMP36 is used to control the cooling system in the house. If the temperature exceeds a certain threshold, the fan is turned ON to reduce the temperature. Similarly, the light-dependent resistor (LDR) is used to detect the light state in the house, allowing the Arduino Uno kit to turn ON the lights when it gets dark and turn them OFF when it gets light.

2.2.4 Conclusion

This paper presented a remote sensing scheme for a home automation system using the Arduino Uno Kit. The system offers two operational scenarios: a manual scenario where users can control devices remotely using a smartphone, and an automatic scenario where devices are controlled based on signals from sensors. The system is implemented using the Matlab-GUI platform, providing a user-friendly interface for monitoring and control. Overall, the system is simple, flexible, and cost-effective, making it suitable for smart home applications.

This paper presents the design and implementation of a remote sensing-based home automation system using the Arduino Uno Kit as the main controller. The system allows for two operational scenarios: manual control using a smartphone and automatic management between remote sensors and the Arduino Uno Kit. A Matlab GUI platform is designed to provide a user-friendly interface for controlling and monitoring the system. The proposed system is cost-effective, flexible, and simple to implement, making it suitable for smart home applications.

**2.3 STIMULATION OF INTELLIGENT ROOM LIGHTING ILLUMINANCE CONTROL**

Lighting plays a significant role in energy consumption within commercial buildings, offering a key area for enhancing energy efficiency. This paper presents a design for energy-efficient lighting in buildings, discussing the current state of lighting technology, energy-saving strategies, and advanced smart lighting systems. The integration of visible light for potential advancements is also explored. A Simulink model using the Fuzzy tool in MATLAB for an energy-efficient lighting system, based on fuzzy rules to generate appropriate illuminance considering thermal and visual comfort, is highlighted. The intelligent lighting control is integrated with automated window blind control, with a Simulink model designed for smart lighting using Fuzzy logic to control indoor lighting and blinds based on room temperature and outdoor light.

2.3.1 Introduction

The use of incandescent lamps in the late 19th century marked the beginning of artificial lighting advancements. With lighting accounting for about 20% of a home's annual power usage and about 40% for commercial buildings, there is a growing focus on efficient lighting schemes to conserve energy. Control technology in electric light dimming, outdoor light utilization, and light fixture positioning can significantly improve lighting energy efficiency.

Modern lighting sources, like LEDs, offer higher reliability and efficiency compared to older lamps, with luminous efficacy close to 100 lm/w for LEDs compared to around 20 lm/w for incandescent lamps. This makes LEDs a preferred choice for energy-efficient lighting. While previous work has focused on basic lighting control, using fuzzy logic can improve the accuracy and efficiency of lighting systems, especially in terms of time control and user comfort.

2.3.2 Review and Methodology

The literature review covers various lighting systems and control designs for energy management. For instance, Lu et al. discussed smart daylight harvesting, which predicts and controls window transparency based on incoming daylight. Kaur et al. proposed a fuzzy logic-based dimming control circuit for natural light harvesting, integrating outdoor light sensing and room occupancy. Gunay et al. developed an algorithm for adaptive window blinds and lighting control, ensuring sufficient lighting without causing discomfort.

The methodology includes components such as occupancy sensors, radar motion sensors, window blinds, lux sensors, and dimming drivers. These components work together to optimize lighting based on available daylight and user comfort. The simulation design scheme involves creating a Simulink model for an intelligent lighting system, integrating fuzzy logic for control.

2.3.3 Results

The Simulink model was evaluated using constant values for motion detection, occupancy, and outdoor light levels. The model demonstrated that indoor illumination increases with sunlight intensity, and window blind position adjusts accordingly. The fuzzy logic control for LED output showed that LED brightness decreases as outdoor light intensity increases, considering room temperature as a disturbance factor.

2.3.4 Conclusion

The paper presents a design for controlling indoor illuminance with automatic window blinds and LED lights using fuzzy logic. The simulation results demonstrate the effectiveness of the proposed system in adjusting to changing outdoor light levels while considering user comfort. Future work could involve hardware implementation using controllers like Arduino interfacing with fuzzy logic for improved energy efficiency and user comfort.

In conclusion, the proposed lighting system design offers a promising approach to enhancing energy efficiency and user comfort in buildings, showcasing the potential of fuzzy logic in intelligent lighting control.

**2.4 DEVELOPMENT OF AN IOT- BASED DAYLIGHT RESPONSIVE LIGHTING CONTROL & MONITORING SYSTEM FOR INTERIOR ENVIRONMENTS**

Energy derived from electrical sources significantly impacts our day-to-day activities. Reducing energy consumption is crucial to ending the energy crisis and lowering carbon emissions. We are developing an Internet of Things (IoT)-based light intensity control and monitoring system for rooms. The system's controller receives light intensity information from an LDR sensor, which in turn gathers ambient light data. The system adjusts light intensity based on the required percentage using the BOLT IoT platform. Data collected by the sensors is transferred to the BOLT IoT platform for further analysis.

2.4.1 Introduction

In recent years, the global climate has been significantly affected by the widespread adoption of electrical appliances, leading to increased energy consumption. Smart homes are addressing energy conservation by incorporating energy-saving technologies. Lighting consumes a substantial portion of energy in households, and its daily usage continues to rise. Smart house technology was introduced by the American Association in 1984.

Installing a smart house with advanced technologies like machine learning, artificial intelligence, and IoT can decrease consumption, making it suitable for consumers. IoT allows users to control devices remotely over the cloud, web, or LAN, enabling typical processing tasks at a low cost and with less memory.

Integrating natural light in interior environments is popular for its aesthetic appeal, promoting productivity, and reducing energy consumption. Daylight sensors can regulate artificial lighting intensity based on available natural light. Shading systems can control sunlight entering a space, reducing glare and heat gain.

The intensity of light needed varies for different rooms (e.g., reading room, kitchen, sitting room). Controlling light intensity using IoT technology to achieve the intended lux level is referred to as IoT-based luminance level control. This system aims to provide energy-efficient, cost-effective, and user-friendly lighting control options.

The device comprises sensors like LDR installed in the area to be controlled. These sensors detect light quantity and send information to a central gateway device, which calculates the required light quantity and sends control signals to lighting devices.

The Bolt module serves as the IoT technology basis, offering Wi-Fi capabilities and cloud connectivity for remote device management. This technology enables users to control devices from anywhere with an active internet connection using a laptop, phone, or other devices.

2.4.2 Proposed System Configuration

To maintain total lux and reduce electricity consumption during the day, we use automation for light brightness intensity control, with manual control available for extreme lighting requirements. The block diagram illustrates the overall automation and manual control of light brightness intensity.

This system uses Pulse Width Modulation (PWM) control. For automatic control, the main switch and automatic control switch button on the IoT platform are turned on, and the Arduino collects LDR sensor data. The PWM value is sent to the Arduino PWM digital I/O pins, controlling the dimmer module, which adjusts AC voltage based on light percentage in the environment. For manual control, the Bolt device controls the PWM signal, allowing users to adjust brightness using a sliding bar in the application.

2.4.3 Hardware Architecture

A. BOLT IoT Device

The BOLT module has Wi-Fi connectivity, cloud platform facilities, and is based on the esp8266 Wi-Fi chipset. It collects LDR sensor information, processes it internally, and displays data on its cloud platform, accessible from any internet-connected device. The BOLT IoT cloud platform allows dashboard customization, device health monitoring, and running AI and ML algorithms.

B. Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328P microchip, developed by Arduino.cc. It gathers sensor data from the LDR and PWM values from BOLT, processes it, and transmits PWM values to the dimmer module.

C. LDR (Light Dependent Resistor)

An LDR exhibits resistance changes when exposed to light, with resistance depending on light intensity. Connected to an Arduino Uno A0 pin, it detects ambient light intensity, reflecting changes in surrounding light levels.

D. AC Dimmer Module

The AC dimmer module controls alternating current voltage up to 600V/8А. It adjusts bulb brightness gradually, producing a relaxing lighting environment. The dimmer module is efficient with moderate to high brightness dimmable LED lamps but less efficient with dimmable lamps with low brightness levels.

2.4.4 Results

The system maintains overall lux in the interior by adjusting light intensity based on LDR sensor data. Light intensity can also be controlled manually through the IoT system, increasing or decreasing lux using a PWM sliding bar in the application. The system is energy-efficient and can be controlled remotely, offering valuable insights for optimizing lighting performance and comfort.

2.4.5 Conclusion

The proposed IoT-based smart lighting monitoring and control system effectively adjusts lighting based on natural light presence or absence, reducing energy consumption. The system provides constant light output and uniform illumination distribution, contributing to energy savings and environmental sustainability. Future advancements in IoT-based lighting control systems are expected to further improve sustainability and intelligence in interior environments. The collected data from this study can be exported for further analysis, making it a valuable resource for future research.

1. **INCREASE HOME SECURITY USING IOT**

**3.1 DIAGRAM**

**3.2 WORKING**

1. Library Inclusions: The code includes necessary libraries such as Arduino.h, WiFi.h (for ESP8266 and ESP32), SinricPro.h, and SinricProSwitch.h.
2. WiFi Setup: The setupWiFi() function connects the ESP module to the specified WiFi network using the SSID and password provided.
3. Relay Setup: The setupRelays() function sets up the GPIO pins connected to the relays as OUTPUT pins and initializes them to HIGH (OFF state).
4. Flip Switch Setup: The setupFlipSwitches() function sets up the GPIO pins connected to the flip switches as INPUT or INPUT\_PULLUP pins based on the activeLow configuration. It also initializes a map (flipSwitches) to manage the flip switches' state and debounce.
5. SinricPro Setup: The setupSinricPro() function initializes the SinricPro library with the APP\_KEY and APP\_SECRET provided. It also sets up the onPowerState callback for each switch device to handle requests to switch the relays' state (ON/OFF).
6. onPowerState Callback: The onPowerState callback function is called whenever a request is received from the SinricPro server to switch the state of a switch device. It updates the corresponding relay's state (ON or OFF) based on the request.
7. Handle Flip Switches: The handleFlipSwitches() function checks the state of each flip switch. If a flip switch state changes, it debounces the input and sends an event to the SinricPro server to update the corresponding switch device's state. This allows manual control of the relays via the flip switches.
8. Loop: The loop() function continuously calls SinricPro.handle() to handle incoming SinricPro events (such as requests to switch the relay states) and handleFlipSwitches() to manage the flip switches' state.

**3.3 CODE**

/\*

\* ADVANCED example for: how to use up to N SinricPro Switch devices on one ESP module

\* to control N relays and N flipSwitches for manually control:

\* - setup N SinricPro switch devices

\* - setup N relays

\* - setup N flipSwitches to control relays manually

\* (flipSwitch can be a tactile button or a toggle switch and is setup in line #52)

\*

\* - handle request using just one callback to switch relay

\* - handle flipSwitches to switch relays manually and send event to sinricPro server

\*

\* - SinricPro deviceId and PIN configuration for relays and buttins is done in std::map<String, deviceConfig\_t> devices

\*

\* If you encounter any issues:

\* - check the readme.md at https://github.com/sinricpro/esp8266-esp32-sdk/blob/master/README.md

\* - ensure all dependent libraries are installed

\* - see https://github.com/sinricpro/esp8266-esp32-sdk/blob/master/README.md#arduinoide

\* - see https://github.com/sinricpro/esp8266-esp32-sdk/blob/master/README.md#dependencies

\* - open serial monitor and check whats happening

\* - check full user documentation at https://sinricpro.github.io/esp8266-esp32-sdk

\* - visit https://github.com/sinricpro/esp8266-esp32-sdk/issues and check for existing issues or open a new one

\*/

#ifdef ENABLE\_DEBUG

#define DEBUG\_ESP\_PORT Serial

#define NODEBUG\_WEBSOCKETS

#define NDEBUG

#endif

#include <Arduino.h>

#if defined(ESP8266)

#include <ESP8266WiFi.h>

#elif defined(ESP32) || defined(ARDUINO\_ARCH\_RP2040)

#include <WiFi.h>

#endif

#include "SinricPro.h"

#include "SinricProSwitch.h"

#include <map>

#define WIFI\_SSID "SFLY1"

#define WIFI\_PASS "90009000"

#define APP\_KEY "642a6b59-fabf-48e3-a934-94b3a3771eca"

#define APP\_SECRET "1987a729-43e1-4231-a3b7-1a0c1a88a2ed-adb1ea20-0f02-4749-a4f6-4738c8c8964a"

// comment the following line if you use a toggle switches instead of tactile buttons

#define BAUD\_RATE 115200

#define DEBOUNCE\_TIME 250

#define RELAYPIN\_1 2

#define RELAYPIN\_2 4

typedef struct { // struct for the std::map below

int relayPIN;

int flipSwitchPIN;

bool activeLow;

} deviceConfig\_t;

// this is the main configuration

// please put in your deviceId, the PIN for Relay and PIN for flipSwitch

// this can be up to N devices...depending on how much pin's available on your device ;)

// right now we have 4 devicesIds going to 4 relays and 4 flip switches to switch the relay manually

std::map<String, deviceConfig\_t> devices = {

//{deviceId, {relayPIN, flipSwitchPIN, activeLow}}

{"6627a9647c9e6c6fe8733cc2", { 2, 13, true }},

{"6627a9c37c9e6c6fe8733d9b", { 4, 12, true }},

};

typedef struct { // struct for the std::map below

String deviceId;

bool lastFlipSwitchState;

unsigned long lastFlipSwitchChange;

bool activeLow;

} flipSwitchConfig\_t;

std::map<int, flipSwitchConfig\_t> flipSwitches; // this map is used to map flipSwitch PINs to deviceId and handling debounce and last flipSwitch state checks

// it will be setup in "setupFlipSwitches" function, using informations from devices map

void setupRelays() {

for (auto &device : devices) { // for each device (relay, flipSwitch combination)

int relayPIN = device.second.relayPIN; // get the relay pin

pinMode(relayPIN, OUTPUT);

digitalWrite(relayPIN, HIGH); // set relay pin to OUTPUT

}

}

void setupFlipSwitches() {

for (auto &device : devices) { // for each device (relay / flipSwitch combination)

flipSwitchConfig\_t flipSwitchConfig; // create a new flipSwitch configuration

flipSwitchConfig.deviceId = device.first; // set the deviceId

flipSwitchConfig.lastFlipSwitchChange = 0; // set debounce time

flipSwitchConfig.lastFlipSwitchState = false; // set lastFlipSwitchState to false (LOW)

int flipSwitchPIN = device.second.flipSwitchPIN; // get the flipSwitchPIN

bool activeLow = device.second.activeLow; // set the activeLow

flipSwitchConfig.activeLow = activeLow;

flipSwitches[flipSwitchPIN] = flipSwitchConfig; // save the flipSwitch config to flipSwitches map

if(activeLow) {

pinMode(flipSwitchPIN, INPUT\_PULLUP); // set the flipSwitch pin to INPUT\_PULLUP

}

else {

pinMode(flipSwitchPIN, INPUT); // set the flipSwitch pin to INPUT

}

}

}

bool onPowerState(String deviceId, bool &state) {

Serial.printf("%s: %s\r\n", deviceId.c\_str(), !state ? "on" : "off");

int relayPIN = devices[deviceId].relayPIN; // get the relay pin for corresponding device

digitalWrite(relayPIN, !state); // set the new relay state

return true;

}

void handleFlipSwitches() {

unsigned long actualMillis = millis(); // get actual millis

for (auto &flipSwitch : flipSwitches) { // for each flipSwitch in flipSwitches map

unsigned long lastFlipSwitchChange = flipSwitch.second.lastFlipSwitchChange; // get the timestamp when flipSwitch was pressed last time (used to debounce / limit events)

if (actualMillis - lastFlipSwitchChange > DEBOUNCE\_TIME) { // if time is > debounce time...

int flipSwitchPIN = flipSwitch.first; // get the flipSwitch pin from configuration

bool lastFlipSwitchState = flipSwitch.second.lastFlipSwitchState; // get the lastFlipSwitchState

bool activeLow = flipSwitch.second.activeLow;

bool flipSwitchState = digitalRead(flipSwitchPIN); // read the current flipSwitch state

if(activeLow) flipSwitchState = !flipSwitchState;

if (flipSwitchState != lastFlipSwitchState) { // if the flipSwitchState has changed...

#ifdef TACTILE\_BUTTON

if (flipSwitchState) { // if the tactile button is pressed

#endif

flipSwitch.second.lastFlipSwitchChange = actualMillis; // update lastFlipSwitchChange time

String deviceId = flipSwitch.second.deviceId; // get the deviceId from config

int relayPIN = devices[deviceId].relayPIN; // get the relayPIN from config

bool newRelayState = !digitalRead(relayPIN); // set the new relay State

digitalWrite(relayPIN, newRelayState); // set the trelay to the new state

SinricProSwitch &mySwitch = SinricPro[deviceId]; // get Switch device from SinricPro

mySwitch.sendPowerStateEvent(newRelayState); // send the event

#ifdef TACTILE\_BUTTON

}

#endif

flipSwitch.second.lastFlipSwitchState = flipSwitchState; // update lastFlipSwitchState

}

}

}

}

void setupWiFi() {

Serial.printf("\r\n[Wifi]: Connecting");

#if defined(ESP8266)

WiFi.setSleepMode(WIFI\_NONE\_SLEEP);

WiFi.setAutoReconnect(true);

#elif defined(ESP32)

WiFi.setSleep(false);

WiFi.setAutoReconnect(true);

#endif

WiFi.begin(WIFI\_SSID, WIFI\_PASS);

while (WiFi.status() != WL\_CONNECTED)

{

Serial.printf(".");

delay(250);

}

Serial.printf("connected!\r\n[WiFi]: IP-Address is %s\r\n", WiFi.localIP().toString().c\_str());

}

void setupSinricPro() {

for (auto &device : devices) {

const char \*deviceId = device.first.c\_str();

SinricProSwitch &mySwitch = SinricPro[deviceId];

mySwitch.onPowerState(onPowerState);

}

SinricPro.begin(APP\_KEY, APP\_SECRET);

}

void setup() {

Serial.begin(BAUD\_RATE);

setupRelays();

setupFlipSwitches();

setupWiFi();

setupSinricPro();

}

void loop() {

SinricPro.handle();

handleFlipSwitches();

}

**3.4 TEAM ORGANIZATION**

* + 1. **Task Network**

1. Research and Analysis

- Identify key requirements for home security using IoT.

- Investigate existing IoT-based home security solutions.

- Analyze potential threats and vulnerabilities in home security systems.

2. System Design

- Define the architecture for the IoT-based home security system.

- Specify the hardware and software components needed.

- Design the user interface for remote monitoring and control.

3. Hardware Acquisition and Setup

- Procure necessary hardware components (sensors, microcontrollers, IoT devices).

- Install sensors and devices in strategic locations within the house.

- Ensure proper connectivity and integration of hardware components.

4. Software Development

- Develop the software for controlling appliances remotely.

- Implement algorithms for simulating occupancy and scheduling appliances.

- Integrate the software with the IoT platform for monitoring and control.

5. Testing and Validation

- Conduct thorough testing of the entire system.

- Validate the effectiveness of the simulated occupancy feature.

- Ensure the system can effectively deter theft by appearing occupied.

6. Documentation and Presentation

- Document the design, development, and testing processes.

- Prepare user manuals and installation guides.

- Create a presentation to showcase the project's features and functionality.

7. Deployment and Maintenance

- Deploy the IoT-based home security system in the target environment.

- Provide ongoing maintenance and support for the system.

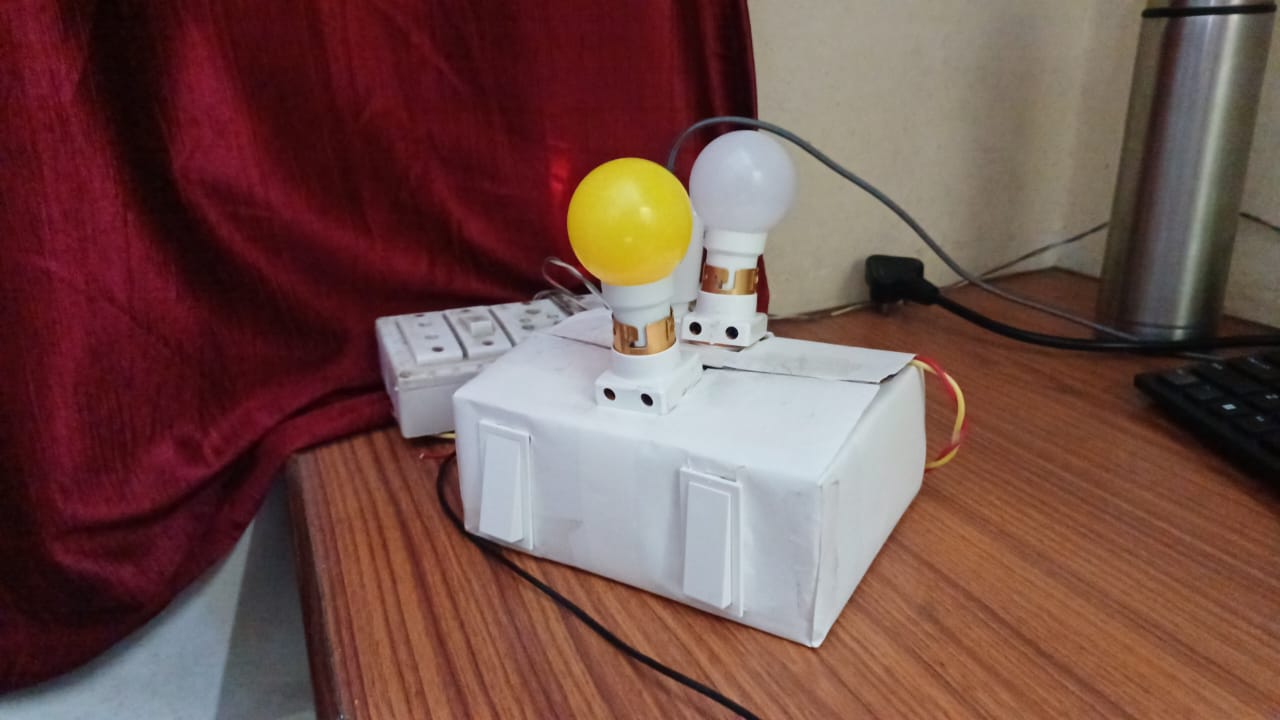
- Monitor system performance and address any issues that arise.

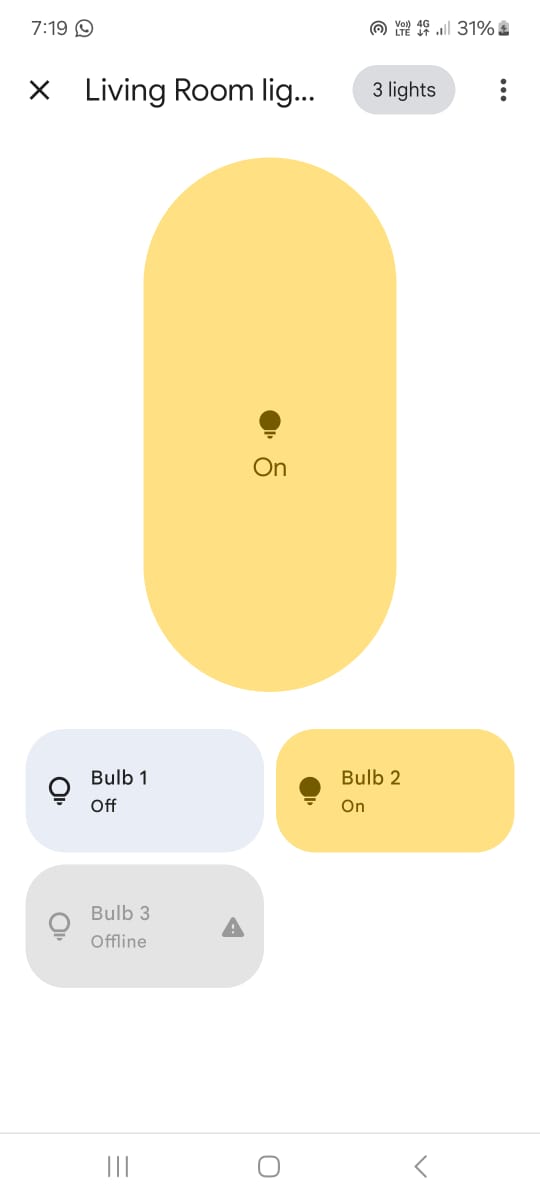
1. **RESULTS**

**4.1 OUTCOMES**

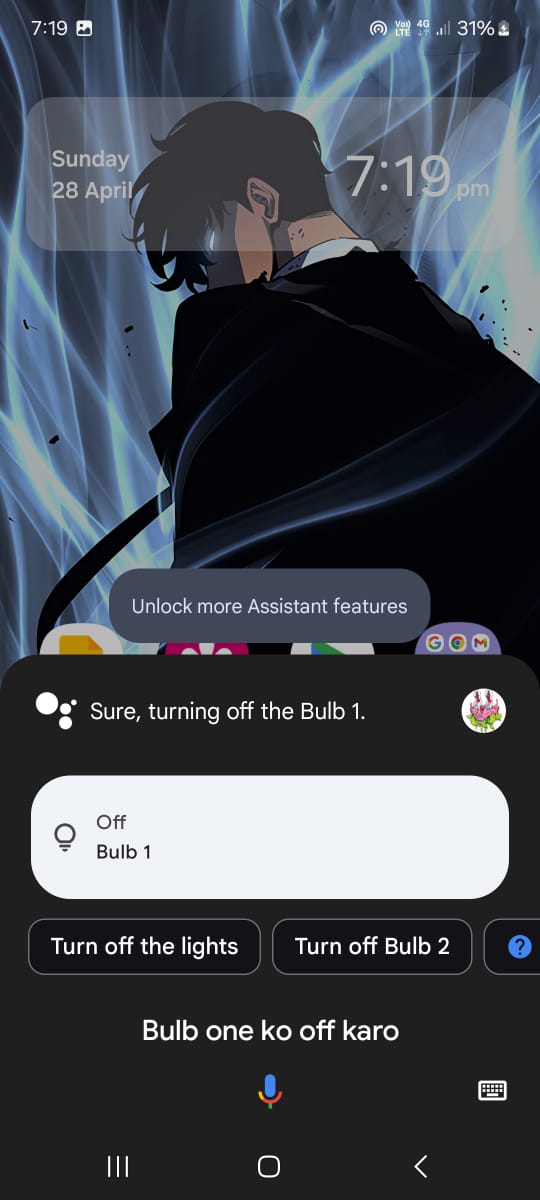
1. Enhanced Security: By simulating occupancy and scheduling appliances, the system creates the illusion that someone is present in the house, deterring potential burglars and enhancing the overall security of the home.
2. Remote Monitoring and Control: The system allows users to remotely monitor and control their home appliances, lights, and security devices using a mobile application or web interface, providing convenience and peace of mind.
3. Energy Efficiency: By scheduling appliances and lights to turn on and off at specific times, the system helps reduce energy consumption, contributing to a more sustainable living environment.
4. Cost-Effective Solution: The project offers a cost-effective home security solution compared to traditional security systems, as it utilizes existing IoT devices and can be easily implemented without the need for expensive hardware.
5. Scalability and Flexibility: The modular design of the system allows for scalability, enabling users to add or remove devices as needed. Additionally, the system can be easily customized to suit individual preferences and requirements.

**4.2 IMAGES**

****



Mobile Control



Voice Control

1. **CONCLUSIONS**

**5.1 CONCLUSION**

In conclusion, the home security system using IoT technology offers an innovative solution to enhance home security and protect against theft. By simulating occupancy and scheduling appliances, the system effectively deters potential intruders by creating the illusion that someone is present in the house.

Through the integration of IoT devices such as motion sensors, smart plugs, and smart bulbs, the system can intelligently control lighting, television, and other appliances to mimic the patterns of a typical household. This not only enhances security but also provides convenience and energy efficiency benefits to the user.

While the system has shown promising results in improving home security, there are certain limitations and challenges that need to be addressed. These include the reliance on internet connectivity, potential vulnerabilities in IoT devices, and the need for continuous monitoring and updates to ensure optimal performance.

Looking ahead, the future scope of the project includes advancements in AI integration, increased automation, integration with smart assistants, and improved sensor technologies. These advancements aim to further enhance the effectiveness and efficiency of the home security system, providing users with a comprehensive and reliable solution to protect their homes.

**5.2 FUTURE SCOPE**

The future scope of the home security system using IoT technology is promising, with several potential advancements and expansions:

1. Enhanced AI Integration: Future systems could incorporate advanced artificial intelligence (AI) algorithms to improve threat detection and response. AI could analyze patterns of behavior and identify anomalies to enhance security measures.
2. Increased Automation: Automation could be further enhanced to include more devices and actions. For example, the system could automatically adjust lighting, temperature, and blinds based on occupancy and time of day.
3. Integration with Smart Assistants: Integration with popular smart assistants like Amazon Alexa or Google Assistant could allow for voice-controlled security features, providing a more convenient user experience.
4. Advanced Sensor Technologies: Future systems could leverage advancements in sensor technologies to improve accuracy and efficiency. For example, more advanced motion sensors could distinguish between different types of movement, reducing false alarms.
5. Energy Efficiency: The system could be further optimized for energy efficiency by intelligently managing the use of appliances and lighting based on occupancy and natural light levels.
6. Remote Monitoring and Control: Improved connectivity and mobile applications could allow for more robust remote monitoring and control, giving users greater flexibility and peace of mind.
7. Scalability and Customization: Future systems could be more scalable and customizable, allowing users to easily add or modify components to suit their specific security needs.
8. Integration with Smart Home Ecosystems: Integration with other smart home devices and ecosystems could provide a more comprehensive home automation and security solution.

Overall, the future of IoT-based home security systems is likely to focus on improving functionality, usability, and integration with other smart home technologies to provide a seamless and efficient security solution for homeowners.

**5.3 APPLICATION**

The home security system using IoT technology has several practical applications:

1. Vacation Security: When homeowners are away on vacation, the system can simulate occupancy, making it appear as though someone is at home, thus deterring potential burglars.
2. Remote Monitoring: Homeowners can monitor the security of their homes remotely through a smartphone or computer, receiving alerts in case of any suspicious activity.
3. Energy Efficiency: The system can be integrated with smart home devices to control lighting and appliances, leading to energy savings by ensuring that they are only on when needed.
4. Customizable Security Settings: Users can customize the system to suit their needs, such as adjusting the sensitivity of motion sensors or setting specific schedules for appliance control.
5. Integration with Other Smart Home Systems: The system can be integrated with other smart home systems, such as smart locks or cameras, to create a comprehensive home security solution.
6. Enhanced Safety: In addition to deterring burglars, the system can also enhance the safety of the home by alerting homeowners to potential hazards, such as smoke or carbon monoxide leaks.