Table of Contents

Abstract

With the advancement of IoT and automation, smart home systems have gained popularity for their ability to enhance security and convenience. This project focuses on designing a smart home automation system using Arduino Uno R3 and ESP C3 Mini, integrating RFID authentication, ultrasonic sensors, flame detection, motion sensing, temperature monitoring, and relay-based appliance control. The system includes real-time monitoring via Blynk, automatic control of home devices, and security features such as intruder alerts. This research aims to develop an affordable, reliable, and efficient smart home solution that ensures safety, energy efficiency, and automation.

Acknowledgment

I would like to sincerely thank my supervisors, Md. Omaer Faruq Goni and Oishi Jyoti, Assistant Professors in the Department of Electrical and Computer Engineering, for their invaluable advice, support, and encouragement throughout the project's development. Their knowledge and experience have been crucial in determining the direction of this research, and I am incredibly grateful for their perseverance and commitment, which have been crucial to the successful completion of this work.

CHAPTER 1

1.1 Introduction

Home automation has emerged as a significant technological advancement, integrating IoT and smart control systems to improve security, efficiency, and convenience. Traditional home management systems require manual intervention, leading to inefficiencies in energy consumption and security vulnerabilities. This project aims to design and implement a smart home automation system using Arduino Uno R3 and ESP C3 Mini, incorporating RFID-based security, motion detection, temperature-controlled automation, and fire safety measures. By leveraging wireless communication and IoT platforms like Blynk, users can remotely monitor and control appliances, ensuring optimal energy use and enhanced safety. The system includes various sensors such as ultrasonic, flame, PIR, and LDR sensors, combined with a relay module to automate household tasks. The main objective is to provide a cost-effective and user-friendly solution that enhances home security and minimizes human effort. The project not only addresses key concerns like unauthorized access and fire hazards but also contributes to energy conservation. Future enhancements may include voice-controlled integration, AI-based anomaly detection, and solar-powered automation. By implementing a smart, automated system, this research contributes to the ongoing evolution of home automation technologies, offering a scalable and adaptable model for modern households.

1.2 Background

Traditional home systems require manual control, making them inefficient and prone to security vulnerabilities. With the emergence of smart home technology, automation enables energy-efficient and secure home management. This project integrates IoT-based automation, real-time monitoring, and security enhancements to create a smart home solution that minimizes human intervention and maximizes convenience.

1.3 Motivation

The increasing demand for smart home systems is driven by concerns about security, energy efficiency, and convenience. Issues such as unauthorized access, high energy consumption, and lack of remote control capabilities highlight the need for an automated solution. This project aims to develop a system that provides intelligent control over home appliances while ensuring security and user accessibility.

1.4 Problem Description

Despite the availability of automation solutions, existing home systems face challenges such as:

- **Security risks** – Traditional locks and manual security measures are prone to breaches.
- **Energy inefficiency** – Lights and appliances are often left on unnecessarily, increasing electricity bills.
- **Lack of remote monitoring** – Users cannot control or monitor their home systems remotely.
- **Manual operation** – Home devices require direct physical interaction for control.

This project seeks to address these issues by integrating IoT-based control, smart authentication, and real-time monitoring.

1.5 Objectives

The key objectives of this research are:

1. **Develop an IoT-based smart home system** using Arduino Uno R3 and ESP C3 Mini.
2. **Implement security features** such as RFID authentication for door access.
3. **Integrate automation mechanisms** for lights, fans, and appliances using relay modules.
4. **Enhance fire safety** with a flame sensor that triggers alerts and automated responses.
5. **Enable remote monitoring and control** through the Blynk IoT platform.

1.6 Methodology

The project follows a structured approach:

1.6.1 Hardware Integration

- **Microcontrollers:** Arduino Uno R3, ESP C3 Mini.
- **Sensors:** RFID, ultrasonic, PIR, temperature, flame, LDR.
- **Actuators:** Servo motors, relay modules, buzzers.
- **Communication:** Wi-Fi (ESP C3 Mini) for IoT connectivity

1.6.2 Software Development

- **Programming**: Arduino IDE for firmware development.

- **IoT Connectivity**: Blynk for remote access.
- **Automation Logic**: Implement sensor-based automated responses.

1.6.3 Testing and Optimization

- **Test various home scenarios** to ensure seamless automation.
- **Optimize power consumption** for efficient energy management.

1.7 Limitations

Despite its benefits, the proposed system has some limitations:

- **Wi-Fi dependency** – Remote control features require a stable internet connection.
- **Limited security authentication** – RFID may not be as secure as biometric verification.
- **Environmental interference** – PIR and LDR sensors may be affected by external conditions.
- **Power constraints** – Multiple sensors and actuators require efficient power management.

1.8 Summary

This project presents an IoT-enabled smart home automation system that integrates security, automation, and remote monitoring. Using Arduino Uno R3 and ESP C3 Mini, the system enhances home security through RFID authentication, optimizes energy consumption through automated lighting and fan control, and improves safety with fire detection mechanisms. While the system has some limitations, its ability to automate daily tasks, enhance security, and provide real-time monitoring makes it a practical and cost-effective solution for modern homes.

CHAPTER 2

2.1 Technology and Literature Survey

Smart home automation has evolved with advancements in IoT, artificial intelligence, and embedded systems, leading to efficient and intelligent solutions for home management. Research in smart home technology primarily focuses on security, energy efficiency, and automation. Studies highlight the integration of RFID authentication, motion detection, temperature control, and fire

safety mechanisms to enhance home security and convenience. Existing systems, such as Google Nest and Amazon Alexa, demonstrate the potential of AI-driven smart home environments, but they rely heavily on cloud-based processing. Several research papers discuss the application of Arduino and ESP microcontrollers for cost-effective automation, enabling real-time monitoring via IoT platforms like Blynk and MQTT. Comparative studies indicate that relay-based control of appliances, coupled with sensor-driven automation, improves efficiency while reducing energy consumption. However, challenges such as connectivity issues, power management, and authentication security remain significant. This project builds upon previous research by integrating RFID-based security, real-time fire detection, and sensor-driven automation, ensuring a robust and scalable smart home solution.

2.2 Basic Operations

2.2.1 Security and Access Control

- RFID Authentication: Uses RFID tags to grant access to authorized users.
- Motion Sensors: Detects movement and triggers alarms or lights.
- Smart Locks: Uses RFID to secure entry points.

2.2.2 Energy Management

- Automatic Lighting: LDR and PIR sensors turn lights on/off based on brightness and movement.
- Temperature-based Fan Control: DHT11 sensor adjusts fan speed based on room temperature.
- Relay-based Appliance Control: Manages electrical devices remotely using IoT.

2.2.3 Fire Detection

Flame Sensors: Detects fire and triggers alarms and water pumps.

2.2.4 Waste Management

Smart Dustbin: Ultrasonic sensors detect presence and open the bin automatically.

2.2.5 Remote Monitoring and IoT Control

- Blynk & MQTT Platforms: Allows users to monitor and control appliances via smartphones.
- ESP Wi-Fi Modules: Enables cloud-based automation and alerts.

2.2.6 Automatic Lighting Control using LDR Sensor

The system integrates an LDR (Light Dependent Resistor) sensor to enable automatic lighting control based on ambient light conditions. The LDR sensor detects the surrounding light intensity and toggles four LED lights accordingly. When natural daylight is sufficient, the system ensures

that the LEDs remain off, conserving energy. Conversely, as night falls or ambient light diminishes, the system automatically switches on the LED lights. This feature enhances energy efficiency and user convenience by providing intelligent lighting control without manual intervention.

2.5 Required Hardware

1. **Arduino Uno R3:** The Arduino Uno R3 is a microcontroller board based on the ATmega328P, widely used for electronics and embedded systems projects. It features 14 digital I/O pins (6 supporting PWM), 6 analog inputs, and supports UART, I2C, and SPI communication. Powered by 5V USB or 7-12V external supply, it has 32KB flash memory, 2KB SRAM, and 1KB EEPROM for data storage. The board is programmed using the Arduino IDE via a USB Type-B connection. With its built-in LED on pin 13, it is ideal for prototyping, sensor interfacing, robotics, and IoT applications.

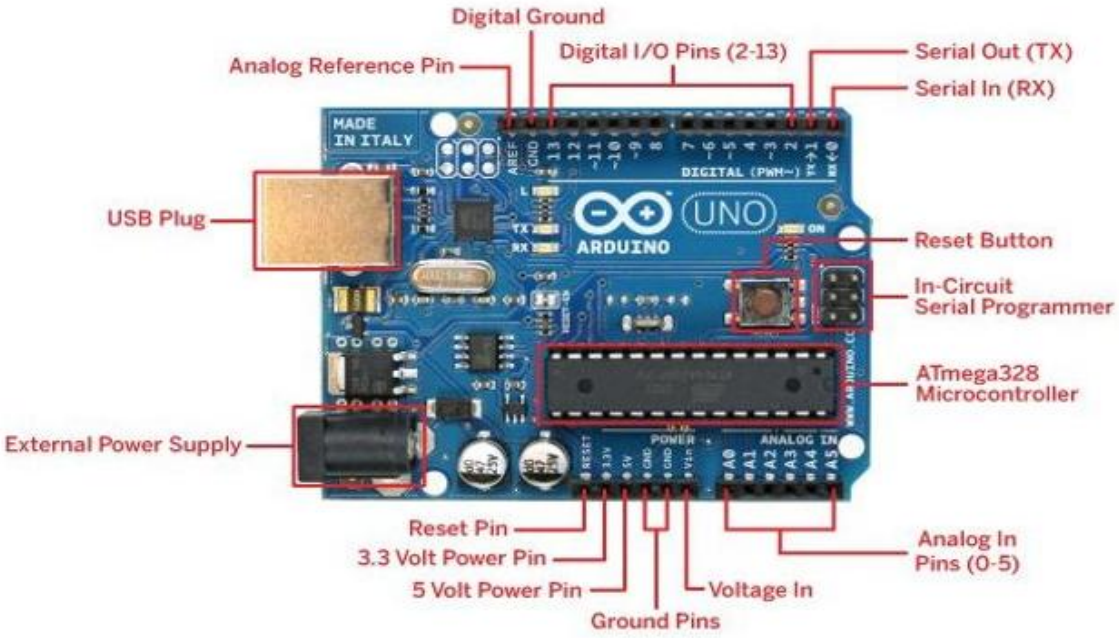

Figure: Arduino Uno R3

2. **ESP C3 Mini:** The ESP-C3 Mini is a compact and efficient Wi-Fi and Bluetooth Low Energy (BLE) module based on the ESP32-C3 chip, designed for IoT applications. It features a RISC-V 32-bit single-core processor, operating at up to 160 MHz, with built-in 2.4 GHz Wi-Fi and Bluetooth 5.0 (LE) for wireless connectivity. The module supports multiple GPIOs, SPI, I2C, UART, ADC, and PWM interfaces, making it versatile for embedded applications. It has low power consumption with deep sleep modes, making it ideal for battery-powered devices. The ESP-C3 Mini is commonly used in smart home

devices, wearables, and industrial automation due to its compact size, security features, and strong wireless capabilities.

The ESP-C3 Mini supports secure boot, flash encryption, and WPA3 for enhanced security. It integrates 4 MB flash memory, making it suitable for firmware storage. With low-power modes, it extends battery life, making it ideal for IoT, automation, and smart devices applications.

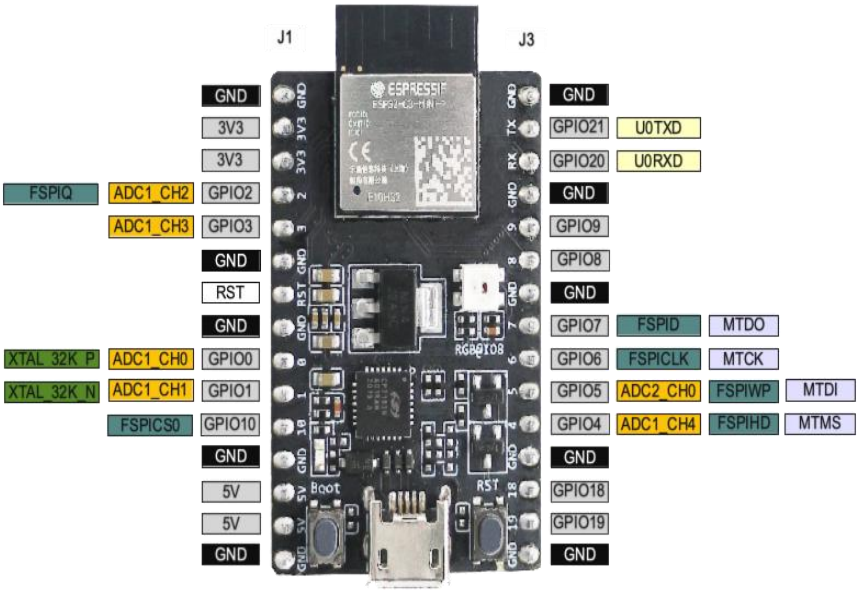

Figure: ESP-C3 Mini

3. **RFID Reader (MFRC522):** The RFID Reader (MFRC522) is a compact and efficient module used for reading RFID tags, commonly used in applications like access control and inventory management. It communicates with microcontrollers, like Arduino, via SPI or I2C protocols. The MFRC522 supports 13.56 MHz frequency, enabling it to read and write to compatible RFID tags. It provides quick and reliable data transfer and is easy to integrate with various platforms. However, it has a relatively limited reading range (about 5-10 cm) and can be sensitive to environmental interference, which may reduce performance in certain settings.

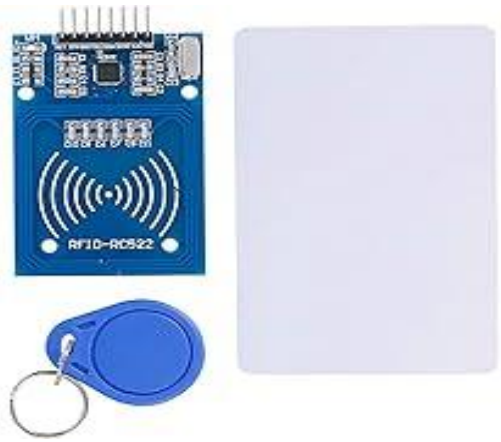

Figure: RFID Reader (MFRC522)

4. **Ultrasonic Sensor (HC-SR04):** The HC-SR04 ultrasonic sensor is used for measuring distances by emitting ultrasonic waves and calculating the time it takes for the waves to bounce back after hitting an object. It consists of two main components: a transmitter that sends out a high-frequency sound pulse and a receiver that detects the echo. The sensor calculates the distance by measuring the time between the transmission and reception of the sound wave, using the formula: distance = (time * speed of sound) / 2. This sensor is commonly used in robotics, obstacle detection, and proximity sensing applications.

Figure: Ultrasonic Sensor (HC-SR04)

5. **Flame Sensor:** A flame sensor detects the presence of a flame or fire by sensing infrared or ultraviolet light emitted from the flame. It is commonly used in industrial applications, heating systems, and boilers to ensure safety. When a flame is detected, the sensor signals the system to maintain ignition or shut off to prevent dangerous conditions, ensuring efficient operation and safety.

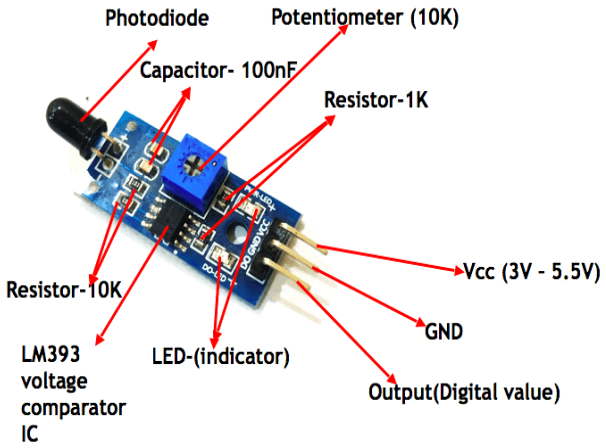

Figure: Flame Sensor

6. **PIR Motion Sensor**: A PIR (Passive Infrared) motion sensor detects motion by measuring changes in infrared radiation emitted by objects, typically human bodies. It consists of a sensor that detects heat variations in its surroundings. When a warm object moves within its range, the sensor triggers an output, commonly used in security systems, automatic lighting, and motion-triggered alarms.

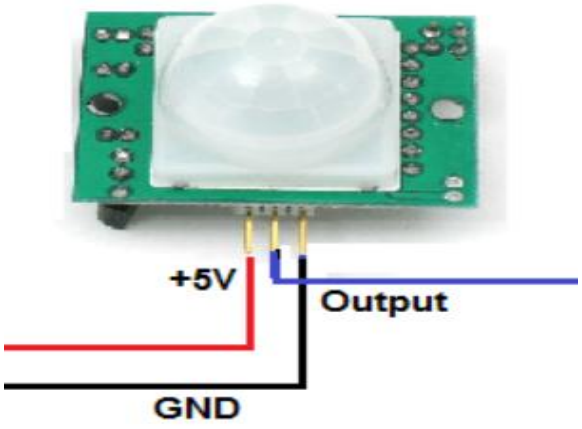

Figure: PIR Motion Sensor

7. **Temperature Sensor (DHT11):** The DHT11 is a digital temperature and humidity sensor that measures ambient temperature and relative humidity. It consists of a thermistor and a capacitive humidity sensor, providing data to a microcontroller via a single-wire digital interface. The sensor outputs a calibrated digital signal with temperature readings in Celsius and humidity levels as a percentage, making it suitable for applications like weather stations, HVAC systems, and environmental monitoring.

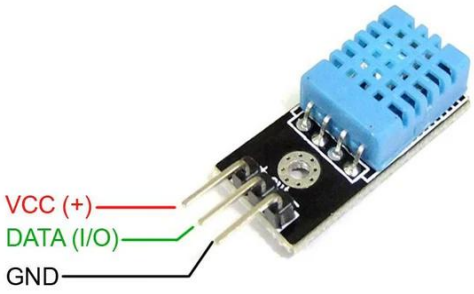

Figure: Temp. Sensor (DHT11)

8. **LDR Sensor**: An LDR (Light Dependent Resistor) sensor changes its resistance based on the amount of light falling on it. In low light, the resistance is high, while in bright light, the resistance decreases. This allows it to detect light levels and is commonly used in light-sensitive applications like automatic street lighting, light meters, or controlling devices based on ambient light conditions. It helps in energy-efficient systems by adjusting based on light intensity.

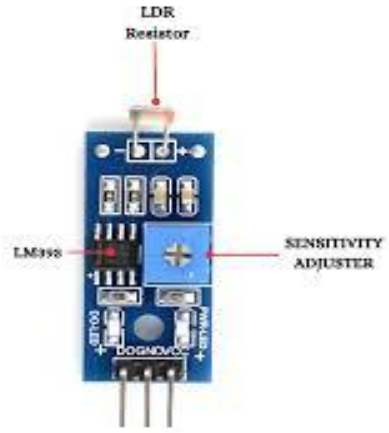

Figure: LDR Sensor

9. **L298N Motor Driver:** The L298N motor driver controls the movement of the delivery bot by regulating the speed and direction of its motors. It features a dual H-Bridge design, supports two DC motors per board, and can handle up to 2A per channel while operating at 5V–35V. PWM support allows precise speed adjustments, while its built-in heat sink prevents overheating. It receives control signals from the ESP32 via PWM and drives four TT DC motors for smooth movement.

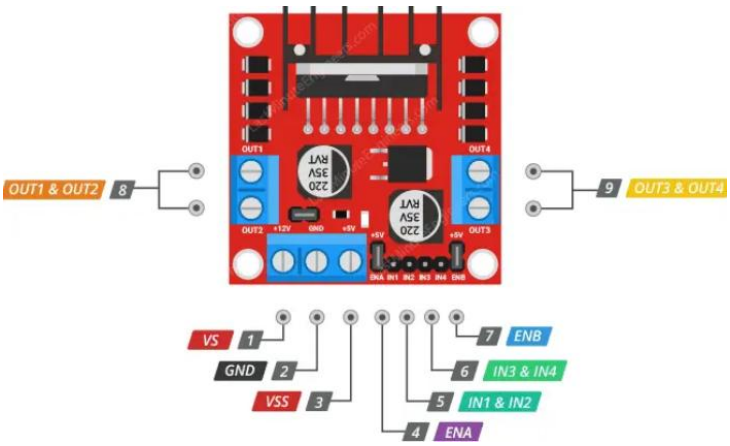

Figure: Pin Diagram of L298 Motor Driver

10. **Servo Motor (SG90):** Two servo motors are used to control the pan-tilt mechanism. The SG90 servo, a lightweight option with 180° rotation, operates on 4.8V–6V, while the MG995 is a more powerful alternative with a torque rating of 13kg/cm. These servos, controlled by PWM signals from the ESP32, allow precise adjustments to the camera's viewing angle.

Figure: Pin Diagram of Servo Motor (SG90)

11. **4-Channel Relay Module**: A 4-channel relay module is used to control multiple high-voltage devices like motors, lamps, and home appliances through low-voltage circuits. It consists of four independent relays, each with a switch that can be activated by a microcontroller or other digital control sources, like Arduino. When a control signal is applied to a relay, it switches from its default state (either open or closed) to allow or block the flow of electricity. The module typically supports both normally open (NO) and normally closed (NC) configurations, providing flexibility in controlling external devices safely.

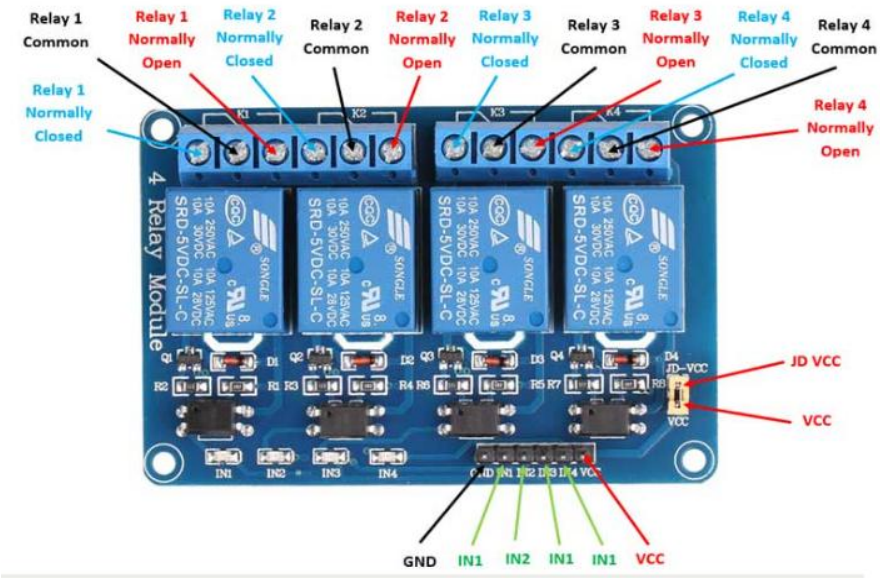

Figure: 4 Channel Relay Module

12. **Buzzer:** A buzzer is an electronic device that produces sound when activated. It operates using piezoelectric or electromagnetic principles, commonly used in alarms, timers, and notifications for alerts or warnings in various systems.

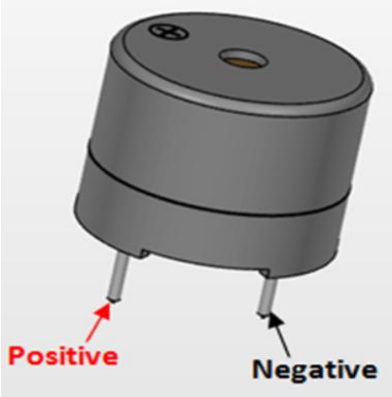

Figure: Pin Diagram of Buzzer

13. **16×2 LCD display and I2C module:** A 16×2 LCD display is a widely used alphanumeric display that can show 16 characters per row across 2 rows. It operates using an HD44780 controller, which allows communication via parallel (4-bit or 8-bit) or serial interfaces. To reduce wiring, an I2C (Inter-Integrated Circuit) module is often used, which converts parallel communication into a 2-wire (SDA and SCL) serial interface, enabling easy connection with microcontrollers like Arduino. The I2C module includes a built-in PCF8574 chip, which manages data transmission efficiently. This setup simplifies coding,

conserves GPIO pins, and enhances system scalability, making it ideal for embedded projects.

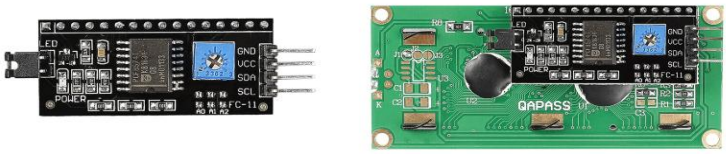

Figure: I2C Module and 16×2 LCD display

14. **Water Pump (5V):** A 5V pump is a small DC-powered device that moves liquids using an electric motor. It operates at 5 volts, making it ideal for low-power applications like water circulation, mini fountains, and DIY projects.

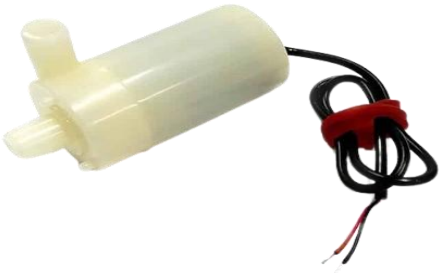

Figure: 5V Water Pump

15. **Jumper Wires:** Jumper wires connect various electronic components, enabling signal and power transmission. These include male-to-male, male-to-female, and female-to-female configurations, supporting low-current connections up to 2A. They play a crucial role in establishing reliable links between the microcontroller, sensors, and actuators.

2.6 Required Software

Blynk (IoT Platform): Blynk is used for remote monitoring and control of the bot. It allows users to check the bot's location, receive authentication alerts, and monitor its status through a mobile application or web interface.

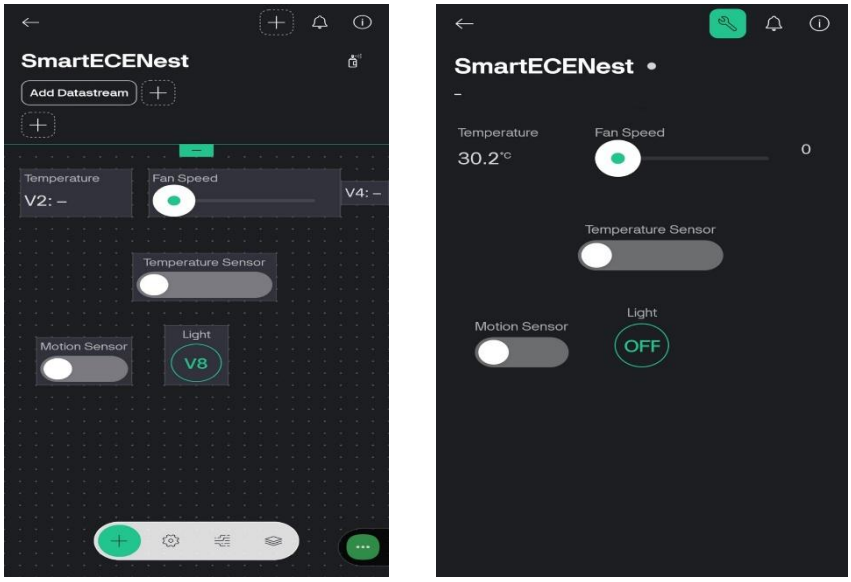

Figure: Blynk Software

CHAPTER 3

3.1 Hardware connection

1. Dustbin Servo Motor Connections:

Component	Arduino Pin
Dustbin Servo (Signal)	6
Dustbin Servo (VCC)	5V
Dustbin Servo (GND)	GND

2. Door Servo Motor Connections:

Component	Arduino Pin
Door Servo (Signal)	3
Door Servo (VCC)	5V
Door Servo (GND)	GND

3. RFID Module Connections:

RFID Pin	Arduino Pin
VCC 3.3V	3.3v
GND GND	GND
RST 9	9
SDA (SS) 10	10
SCK 13	13
MOSI 11	11
MISO 12	12

4. Ultrasonic Sensor (HC-SR04) Connections:

Ultrasonic Sensor Pin	Arduino Pin
VCC	5V
GND	GND
TRIG	4
ECHO	5

5. Thief Alarm (Buzzer) Connections:

Component	Arduino Pin
LED/Buzzer (Signal)	2
LED/Buzzer (VCC)	5V
LED/Buzzer (GND)	GND

6. LCD I2C Module Connections:

LCD I2C Pin	Arduino Pin
VCC	5V
GND	GND
SDA	A4
SCL	A5

7. DHT11 Sensor Connection:

Component	ESP-32 C3 Pin
DHT11 (**VCC)**	**3.3V**

DHT11 (**GND)**	**GND**
DHT11 (**Data)**	**GPIO 2**

8. L298N Motor Driver (Fan Control) Connection:

Component	**ESP-32 C3 Pin**
IN1 (Motor Direction)	**GPIO 4**
IN2 (Motor Direction)	**GPIO 5**
ENA (Speed Control - PWM)	**GPIO 6**
Motor (**VCC)**	**External 12V**
Motor (**GND)**	**GND**

9. PIR Motion Sensor Connection:

Component	**ESP-32 C3 Pin**
PIR (**VCC)**	**5V**
PIR (**GND)**	**GND**
PIR (**OUT)**	**GPIO 7**

10. LED Connections:

Component	**ESP-32 C3 Pin**
LED **VCC**	**5V**
LED **GND**	**GND**
LED Control (Physical LED)	**GPIO 8**
LED Control (Blynk LED)	**GPIO 9**

11. Fire Sensor Connection:

Component	**ESP-32 C3 Pin**
Fire Sensor **VCC**	**5V**
Fire Sensor **GND**	**GND**
Fire Sensor **OUT**	**GPIO 18**

12. Buzzer Connection:

Component	**ESP-32 C3 Pin**

Buzzer **VCC**	**5V**
Buzzer **GND**	**GND**
Buzzer Control	**GPIO 3**

13. Water Pump Connection:

Component	**ESP-32 C3 Pin**
Pump **VCC**	**5V (or external 12V)**
Pump **GND**	**GND**
Pump Control	**GPIO 0**

3.2 Functionality of the Project

LCD Display (User Feedback Interface): The LCD display is used to provide visual feedback to the user regarding the system's status. It displays messages such as "Scan your card", "Wrong card!", "Door is open", and "Door is locked", ensuring that the user knows the current state of the system. It helps the user interact with the RFID system, providing clear instructions and information about the status of the door lock and the dustbin lid. This enhances the user experience by providing real-time updates on the system's operations.

RFID Module (Access Control): The RFID module is used to control access to the home or specific areas by scanning an RFID card. The module reads the unique UID (Unique Identifier) of an RFID card and compares it to a pre-set UID stored in the system. If the card matches, it activates the servo motor controlling the door lock, either unlocking or locking the door. The system also provides feedback on an LCD screen, notifying the user whether the door is locked or open. If an incorrect card is scanned, the system displays a message indicating "Wrong card!" and denies access.

Ultrasonic Sensor (Smart Dustbin Functionality): The ultrasonic sensor is employed to detect objects or hands within a specific range (in this case, 15 cm) above the dustbin. It uses sound waves to measure the distance to an object and triggers the servo motor to open the dustbin lid when an object is detected within the specified distance. Once the lid is opened for a few seconds, it automatically closes, ensuring that the user does not need to touch the lid. This feature enhances hygiene by allowing users to dispose of waste without physical contact with the dustbin.

Servo Motor (Door Locking and Dustbin Lid Control): The servo motors are responsible for physically locking or unlocking the door and controlling the opening and closing of the dustbin lid. The first servo motor is connected to the door lock mechanism, where it moves to either lock or unlock the door based on the RFID card authentication. The second servo motor controls the

dustbin lid, which opens when the ultrasonic sensor detects a hand or object within 15 cm, and closes automatically after a few seconds. These servos are controlled programmatically based on sensor input, providing an automated experience.

Theif PIN (Intruder Alert): The Theif_PIN is used to activate a security feature when unauthorized access is detected. If the correct RFID card is scanned and the door is unlocked, the Theif_PIN is turned off, indicating that the user has authorized access. However, if the incorrect card is scanned, the Theif_PIN is activated, signaling a potential security breach. This feature is designed to help monitor and secure the premises, providing an alert whenever unauthorized access is attempted.

DHT11 Sensor (Temperature Monitoring & Fan Control): The DHT11 sensor measures the ambient temperature of the room and sends this data to the ESP C3 Mini microcontroller. If the temperature exceeds a certain threshold, it automatically controls the fan's speed to maintain a comfortable environment. Additionally, the user can manually adjust the fan speed through the Blynk app, providing flexibility for the user to optimize comfort as needed. The system automatically adjusts the fan based on predefined temperature ranges to ensure efficient cooling when the temperature rises above a set point.

L298N Motor Driver (Fan Speed Control): The L298N motor driver controls the fan's motor speed by receiving signals from the ESP C3 Mini. When the DHT11 sensor detects high temperatures, the motor driver adjusts the fan's speed based on the fan speed value received. The motor driver uses Pulse Width Modulation (PWM) to regulate the power supplied to the fan's motor, thereby controlling the fan's speed and helping to maintain an optimal indoor environment. Users can also control the fan speed manually through the Blynk app.

PIR Motion Sensor (Motion Detection & LED Control): The PIR motion sensor detects motion within a predefined area. When motion is detected, the system triggers a response, which, in this case, turns on an LED light. The user can enable or disable the motion detection functionality via the Blynk app. If motion is detected while the feature is active, the system automatically turns on the LED. If no motion is detected, the LED will be turned off, enhancing security and energy efficiency by ensuring lights are only on when needed.

Fire Sensor (Fire Detection & Alarm Activation): The fire sensor detects any smoke or fire hazard in the environment. Upon detecting a fire, the system triggers an alarm by activating a buzzer and starts the water pump to potentially suppress the fire. Additionally, it sends a fire alert notification via the Blynk app, allowing the user to take immediate action. The system continues monitoring the fire sensor to ensure safety, deactivating the pump and buzzer once the fire is no longer detected.

Laser Sensor (Intruder Detection): The laser sensor is used for security purposes to detect any intrusion. When the sensor detects an interruption in the laser beam, indicating a potential intruder, it triggers the system's alarm and activates a buzzer. This feature helps in securing the premises

and provides an immediate alert to the user through the Blynk app, notifying them of a possible theft. This system adds an additional layer of security by monitoring the area around the entrance or windows.

Water Pump (Fire Suppression): The water pump is activated in case of a fire detection, providing an automatic water flow to suppress the fire. When the fire sensor detects smoke or fire, it sends a signal to the microcontroller, which then triggers the pump. The water pump helps prevent fire escalation by delivering water to the affected area. In the context of the Smart Home Automation system, the pump is turned on automatically to ensure the safety of the home environment, and the system deactivates it once the fire is no longer detected.

Door Sensor (Intruder Detection & Security): The door sensor detects the status of the door, whether it is open or closed. It works in conjunction with the laser sensor to enhance the security system. If the door is opened, the sensor reads the state and triggers the system to monitor for possible intruders. If an interruption in the laser sensor's beam is detected, which indicates that someone might be trying to break in, the door sensor's status helps to verify that the door is in use and adds context to the intruder detection. The system then sends an alert to the user about possible unauthorized entry, helping to maintain home security.

Automatic LED Control using LDR Sensor:

The LDR sensor enables automatic lighting by detecting ambient light levels and controlling LEDs accordingly. If daylight is sufficient, the LEDs remain off; when darkness increases, the system activates the LEDs. The microcontroller monitors the LDR's output, triggering the LEDs when light falls below a predefined threshold. This enhances energy efficiency, eliminates manual operation, and allows remote control via the Blynk app, supporting the project's smart home automation objectives.

Buzzer (Alert & Alarm): The buzzer is an essential component for alerting users to potential threats or emergencies. It is activated in situations such as fire detection or intruder detection. When the fire sensor detects fire or smoke, the buzzer emits a loud alarm to alert the user and anyone in the vicinity of the potential danger. Similarly, if the laser sensor detects an intruder, the buzzer is triggered to indicate unauthorized entry. Additionally, the buzzer serves as a feedback mechanism, ensuring that the user is alerted in case of any dangerous situation.

4-Channel Relay (Light Control via Blynk App): The 4-channel relay module in this project is used to control a light connected to a 220V power supply. The relay acts as a switch that allows or interrupts the flow of electricity to the connected light. The relay module receives control signals from the microcontroller (in this case, the ESP C3 Mini) via its GPIO pins, which are controlled through the Blynk app. By toggling the relay through the app, you can turn the light on or off remotely. The relay allows for safe switching of high-voltage devices, such as the light, without directly interacting with the high voltage, as the relay acts as an intermediary switch. When the control pin of the relay is activated, it closes the circuit and powers the light, and when deactivated,

it opens the circuit and cuts the power to the light. This integration enables convenient home automation with remote control of lights through the Blynk app, making it easy to manage your home lighting from anywhere.

Blynk App (Remote Control & Monitoring): The Blynk app serves as the user interface for remote control and monitoring of all the components in the system. It allows the user to adjust the fan speed, control the LED, enable or disable motion detection, and receive real-time alerts for events like fire or intrusion. The app communicates with the ESP32 via Wi-Fi to provide a seamless experience where the user can control the system remotely from anywhere. Notifications for emergencies are also sent directly to the app, ensuring the user is always informed of critical events.

This project is integrated with Blynk, allowing remote control of various components given below:

Blynk Virtual Pin	**Function**
V2	Displays Temperature
V4	Fan Speed Control (Manual)
V5	Toggle Fan Auto Mode
V7	Enable/Disable PIR Sensor
V8	Manual LED ON/OFF
Fire Alert	Sends Fire Notification
Thief Alert	Sends Intruder Notification

3.3 Future Work

Enhanced Security Features: To further improve security, advanced features like facial recognition, fingerprint scanning, or even the integration of smart cameras could be added. This would provide multi-layered authentication for accessing the home, ensuring that only authorized individuals can enter. The system could also trigger alerts or notifications in case of any unauthorized access attempts, ensuring that the home remains secure.

Energy Efficiency and Power Management: The system can be expanded to monitor and optimize energy consumption by integrating smart plugs and appliances that can be controlled

remotely. By analyzing the energy usage patterns of various devices, the system could intelligently manage power consumption—turning off lights, fans, or other devices when not in use. This could reduce overall electricity costs and promote sustainable living.

Mobile App Enhancements: The current Blynk mobile app can be enhanced with additional features such as geofencing, which would automatically trigger certain actions based on the user's location. Other possible upgrades include scheduling for devices, push notifications for system alerts, and more detailed control options for users to interact with their smart home devices from anywhere

Integration with More Smart Appliances: The smart home system could be expanded to include more connected appliances, such as refrigerators, washing machines, and air conditioners. These appliances could be remotely controlled or even operate autonomously based on conditions set by the user. This expansion would make the system more comprehensive, offering increased automation and control over various household functions.

Scalability for Larger Spaces and Multi-Home Use: To accommodate larger homes or multiple residences, the system could be made scalable. More devices, sensors, and relays could be added to support the automation of larger spaces such as offices or commercial buildings. Multi-home support would allow the system to be used in various locations, providing remote control and monitoring of multiple smart homes from a single app.

CHAPTER 4

4.1 Cost Estimation

Here's the total cost estimation for the Smart Home Automation project:

- Breadboard×3: 400 Tk
- Arduino Uno R3: 700 Tk

- Esp32 C3 Mini: 500 Tk
- RFID Reader (MFRC522): 360 Tk
- 3.7V Battery×3: 240 Tk
- Battery Casing: 40 Tk
- L298N Motor Driver×2: 300 Tk
- 16×2 LCD Display and I2C Module: 360 Tk
- Sonar Sensor: 100 TK
- PIR Motion Sensor: 90 Tk
- Flame Sensor: 60 Tk
- Temperature Sensor (DHT11): 120 Tk
- LDR Sensor×2: 100 Tk
- 4-Channel Relay Module: 270 Tk
- Buzzer & LED Indicators: 100 Tk
- Light and Holder×2: 350 Tk
- 5V Water Pump: 25 Tk
- Small Motor and Fan: 60 Tk
- Connecting Wires: 300 Tk
- Water pipe: 15 Tk
- Home Design: 1300 Tk

Total Cost: 5790 Taka.

4.2 Challenge faced

Hardware Compatibility: Ensuring all components (Arduino, ESP32, sensors, etc.) worked together smoothly required careful wiring and pin configuration.

Power Supply: Managing power for multiple components, especially with motors and sensors, needed an external power supply to avoid overloading the Arduino.

Sensor Calibration: Calibration of sensors like LDR, DHT11, and flame sensor for accurate readings required fine-tuning in different conditions.

Wireless Communication: Setting up ESP32 C3 mini for reliable Wi-Fi communication was tricky, but resolved by proper configuration and testing.

Data Handling: Managing sensor data without delays required careful timing and using interrupts for real-time processing.

LCD Display Issues: Ensuring proper communication with the 16x2 LCD and I2C module required troubleshooting wiring and code.

Relay Control: Using a 4-channel relay to control actuators needed additional components like transistors to handle higher voltage safely.

RFID Accuracy: Achieving reliable RFID authentication was challenging but resolved by improving tag shielding and antenna placement.

Motion Detection: Preventing false motion sensor triggers involved adjusting sensitivity and adding time delays for reliability.

Debugging: Troubleshooting the system with multiple components was challenging but handled by isolating issues and using serial monitoring.

User Interface: Designing a simple and intuitive control system was essential, and a user-friendly interface was created for ease of use.

Safety Concerns: Ensuring sensors functioned safely in various environments involved adding checks for false alarms and proper enclosures for protection.

4.4 Conclusion

The Smart Home Automation project successfully integrates various electronic components and sensors to enhance the functionality and convenience of home management. By utilizing the Arduino Uno R3 and ESP32 C3 Mini, we were able to create a system capable of wirelessly controlling devices and gathering real-time data. The RFID Reader and Sonar Sensor provide an efficient access control mechanism, ensuring that only authorized individuals can interact with the system. The integration of the Flame Sensor, Motion Sensor, Temperature Sensor (DHT11), and LDR Sensor ensures the system is capable of monitoring environmental conditions and automating actions accordingly, such as turning on fans, lights, or activating alarms in response to detected events.

The 16×2 LCD Display with an I2C module provides an easy-to-read interface, presenting real-time data for the user. The 4-Channel Relay Module and L298N Motor Driver allow for seamless control of electrical devices like lights, motors, and pumps, while the buzzer and LED indicators offer immediate feedback to the user.

The project demonstrates the effectiveness of combining multiple sensors and microcontrollers in creating a smart home system that is not only efficient but also adaptable to different environmental conditions and user needs. Overall, this system provides a secure, energy-efficient, and convenient solution for modern homes, with the potential for future enhancements such as remote control and advanced automation features.

4.5 Picture of Smart Home Automation Project (Smart_ECE_Nest)

Figure: Front View

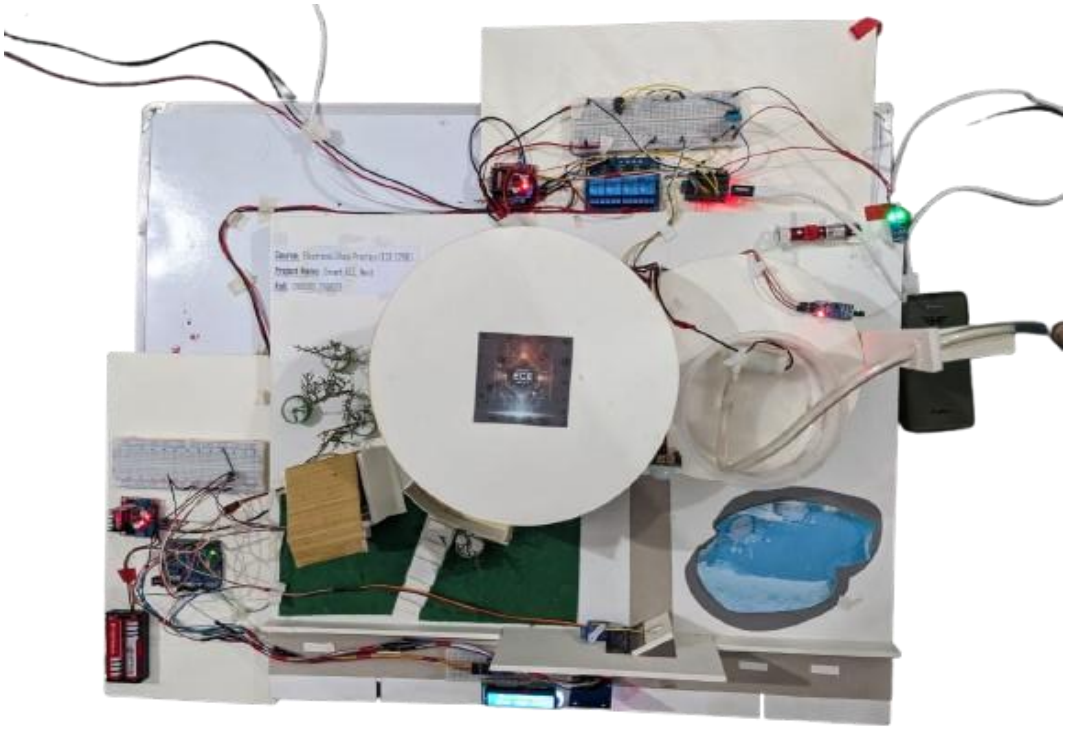

Figure: Top View

4.6 Appendices

Arduino Code:

```
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>

// RFID Module Pins
#define SS_PIN 10
#define RST_PIN 9

// Ultrasonic Sensor Pins
#define TRIG_PIN 4
#define ECHO_PIN 5

// Servo Motor Pins
#define SERVO_DOOR 3
#define SERVO_DUSTBIN 6

#define Theif_PIN 2

// RFID Card UID
String UID = "63 7E 65 1A";
byte lock = 0;

// Object Creation
Servo doorLock;
Servo dustbinLid;
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 rfid(SS_PIN, RST_PIN);

void setup() {
    Serial.begin(9600);

    // Setup RFID
    SPI.begin();
    rfid.PCD_Init();

    // Setup LCD
    lcd.init();
    lcd.backlight();

    // Setup Servos
    doorLock.attach(SERVO_DOOR);
    dustbinLid.attach(SERVO_DUSTBIN);

```

```
    doorLock.write(70);  // Initially lock the door
    dustbinLid.write(0); // Initially, dustbin lid is closed

    // Setup Ultrasonic Sensor
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);

    pinMode(Theif_PIN, OUTPUT);
    digitalWrite(Theif_PIN, LOW);
}

void loop() {
    checkRFID();     // Check for RFID card
    checkDistance(); // Check distance for smart dustbin
}

// RFID Function
void checkRFID() {
    lcd.setCursor(2, 0);
    lcd.print("SmartECENest");
    lcd.setCursor(1, 1);
    lcd.print("Scan your card");

    if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
        return;

    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("Scanning");
    Serial.print("NUID tag is: ");
    String ID = "";

    for (byte i = 0; i < rfid.uid.size; i++) {
        lcd.print(".");
        ID.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
        ID.concat(String(rfid.uid.uidByte[i], HEX));
        delay(300);
    }
    ID.toUpperCase();

    if (ID.substring(1) == UID && lock == 0) {
        doorLock.write(10);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Door is locked");
        digitalWrite(Theif_PIN, LOW);
        delay(1500);
        lcd.clear();
        lock = 1;
    } else if (ID.substring(1) == UID && lock == 1) {
```

```
 95.         doorLock.write(140);
 96.         lcd.clear();
 97.         lcd.setCursor(0, 0);
 98.         lcd.print("Door is open");
 99.         digitalWrite(Theif_PIN, HIGH);
100.         delay(1500);
101.         lcd.clear();
102.         lock = 0;
103.     } else {
104.         lcd.clear();
105.         lcd.setCursor(0, 0);
106.         lcd.print("Wrong card!");
107.         delay(1500);
108.         lcd.clear();
109.     }
110. }
111.
112. // Ultrasonic Sensor & Dustbin Function
113. void checkDistance() {
114.     long duration;
115.     float distance;
116.
117.     // Trigger ultrasonic sensor
118.     digitalWrite(TRIG_PIN, LOW);
119.     delayMicroseconds(2);
120.     digitalWrite(TRIG_PIN, HIGH);
121.     delayMicroseconds(10);
122.     digitalWrite(TRIG_PIN, LOW);
123.
124.     // Read echo response
125.     duration = pulseIn(ECHO_PIN, HIGH);
126.     distance = (duration * 0.0343) / 2; // Convert duration to cm
127.
128.     Serial.print("Distance: ");
129.     Serial.print(distance);
130.     Serial.println(" cm");
131.
132.     // If an object (hand) is detected within 15 cm, open dustbin lid
133.     if (distance > 0 && distance < 15) {
134.         Serial.println("Hand detected! Opening lid...");
135.         dustbinLid.write(5); // Open dustbin
136.         delay(3000);  // Keep it open for 3 seconds
137.         Serial.println("Closing lid...");
138.         dustbinLid.write(100);  // Close dustbin
139.         delay(1000);
140.     }
141.
142.     delay(500);
143. }
144.
```

Esp32 C3 Mini Code:

```
#define BLYNK_TEMPLATE_ID "TMPL6alS5GpQ4"
#define BLYNK_TEMPLATE_NAME "SmartECENest"
#define BLYNK_AUTH_TOKEN "lStDIK3UP_T4ikAkrXqp0uWz7IP08xd8"

#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// Define Wi-Fi credentials
char ssid[] = "Love_Ruet";
char pass[] = "passworddibona";

// Define Blynk Auth Token
char auth[] = BLYNK_AUTH_TOKEN;

// Define the DHT pin and type
#define DHTPIN 2
#define DHTTYPE DHT11

// Define L298N Motor Driver Pins
#define IN1 4
#define IN2 5
#define ENA 6

// Define PIR Motion Sensor Pin
#define PIR_PIN 7

// Define LED Pin
#define LED_PIN 8
#define Blynk_LED 9

#define FIRE_SENSOR_PIN 18
#define LASER_PIN 10
#define Buzzer_PIN 3
#define Pump_PIN 0

#define Door_PIN 19

// Initialize the DHT sensor
DHT dht(DHTPIN, DHTTYPE);

// Global variables
```

```
45. bool enableFanControl = true;  // Fan control by DHT11
46. int manualFanSpeed = 0;  // Manual fan speed from slider
47. bool motionSensorEnabled = true;  // PIR sensor control (V7)
48. bool manualLedOn = false;  // LED manual control flag
49.
50. // Handle Fan Control Toggle (V5)
51. BLYNK_WRITE(V5) {
52.     enableFanControl = param.asInt();
53.     Serial.println(enableFanControl ? "Fan control ENABLED" : "Fan
control DISABLED");
54.
55.     if (!enableFanControl) {
56.         analogWrite(ENA, manualFanSpeed);
57.     }
58. }
59.
60. // Handle Fan Speed Slider (V4)
61. BLYNK_WRITE(V4) {
62.     manualFanSpeed = param.asInt();
63.     if (!enableFanControl) {
64.         analogWrite(ENA, manualFanSpeed);
65.         Serial.print("Manual Fan Speed: ");
66.         Serial.println(manualFanSpeed);
67.     }
68. }
69.
70. // Enable/Disable PIR Motion Sensor (V7)
71. BLYNK_WRITE(V7) {
72.     motionSensorEnabled = param.asInt();
73.     Serial.println(motionSensorEnabled ? "Motion Sensor ENABLED" :
"Motion Sensor DISABLED");
74. }
75.
76. // Handle Manual LED Button (V8)
77. BLYNK_WRITE(V8) {
78.     int buttonState = param.asInt();
79.     if (buttonState == 1) {
80.         digitalWrite(Blynk_LED, LOW);
81.         Serial.println("LED turned ON manually");
82.         manualLedOn = true;
83.     } else {
84.         digitalWrite(Blynk_LED, HIGH);
85.         Serial.println("LED turned OFF manually");
86.         manualLedOn = false;
87.     }
88. }
89.
90. void setup() {
91.     Serial.begin(115200);
92.     Serial.println("Smart Home System Initializing...");
```

```
 93.
 94.     Blynk.begin(auth, ssid, pass);
 95.     dht.begin();
 96.
 97.     pinMode(IN1, OUTPUT);
 98.     pinMode(IN2, OUTPUT);
 99.     pinMode(ENA, OUTPUT);
100.     pinMode(PIR_PIN, INPUT);
101.     pinMode(LED_PIN, OUTPUT);
102.     pinMode(Blynk_LED, OUTPUT);
103.     pinMode(FIRE_SENSOR_PIN, INPUT);
104.     pinMode(LASER_PIN, INPUT);
105.     pinMode(Buzzer_PIN, OUTPUT);
106.     pinMode(Pump_PIN, OUTPUT);
107.     pinMode(Door_PIN, INPUT);
108.
109.     digitalWrite(IN1, LOW);
110.     digitalWrite(IN2, HIGH);
111.     analogWrite(ENA, 0);
112.
113.     digitalWrite(LED_PIN, LOW);
114.     digitalWrite(Pump_PIN, LOW);
115.     digitalWrite(Buzzer_PIN, LOW);
116. }
117.
118. void loop() {
119.     Blynk.run();
120.
121.     // Read temperature
122.     float temperature = dht.readTemperature();
123.     if (!isnan(temperature)) {
124.         Serial.print("Temperature: ");
125.         Serial.print(temperature);
126.         Serial.println("°C");
127.         Blynk.virtualWrite(V2, temperature);
128.     }
129.
130.     // Read humidity
131.     float humidity = dht.readHumidity();
132.     if (!isnan(humidity)) {
133.         Serial.print("Humidity: ");
134.         Serial.print(humidity);
135.         Serial.println("%");
136.         Blynk.virtualWrite(V1, humidity);
137.     }
138.
139.     // Fan control
140.     if (enableFanControl) {
141.         int fanSpeed = 0;
142.         if (temperature >= 25 && temperature <= 40) {
```

```
            fanSpeed = map(temperature, 25, 40, 0, 255);
        } else if (temperature > 40) {
            fanSpeed = 255;
        } else {
            fanSpeed = 0;
        }

        analogWrite(ENA, fanSpeed);
        Serial.print("Fan Speed: ");
        Serial.println(fanSpeed);
        Blynk.virtualWrite(V4, fanSpeed);
    }

    // Motion Detection - Only works if V7 is ON
    if (motionSensorEnabled) {
        int motionState = digitalRead(PIR_PIN);
        if (motionState == HIGH) {
            digitalWrite(LED_PIN, LOW);
            Serial.println("Motion detected! LED ON");
        } else {
            digitalWrite(LED_PIN, HIGH);
            Serial.println("No motion detected. LED OFF");
        }
    }

    int fireState = digitalRead(FIRE_SENSOR_PIN);
    if (fireState == LOW) {  // Fire detected
        Serial.println("🔥 FIRE ALERT! 🔥");
        digitalWrite(Buzzer_PIN, HIGH);
        delay (1000);
        digitalWrite(Pump_PIN, HIGH);
        Blynk.logEvent("fire"); // Send alert
        delay (1000);
    } else {
        digitalWrite(Pump_PIN, LOW);
        digitalWrite(Buzzer_PIN, LOW);
    }

    int doorState = digitalRead(Door_PIN);
    if (doorState == LOW){
    int theifState = digitalRead(LASER_PIN);
    if (theifState == HIGH) {  // Fire detected
        Serial.println("Thief Detected!");
        digitalWrite(Buzzer_PIN, HIGH);
        Blynk.logEvent("theif"); // Send alert
    } else {
        digitalWrite(Buzzer_PIN, LOW);
    }
    }

```

```
193.     delay(2000);  // Reduce delay for faster response
194. }
195.
```

References:

[1] **A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Ayyash, S. S. He, and M. A. E. Hossain,** "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 4th Quart., 2015.

[2] **M. A. Khan, A. K. M. Z. Haider, and A. S. M. H. Gazi,** "Smart Home Automation System Using Arduino and ESP32," *2018 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2018, pp. 1-5.

[3] **M. A. Hossain, M. A. R. Mamun, M. R. Khan, and M. S. Islam,** "Design and Implementation of Smart Home Automation System Using Arduino," *2016 International Conference on Robotics, Automation, Control and Embedded Systems (RACE)*, Dhaka, Bangladesh, 2016, pp. 68-72.

[4] **S. A. B. S. S. S. A. A. Younis,** "Home Automation with IoT," *2016 5th International Conference on Electrical Engineering (ICEE)*, Karachi, Pakistan, 2016, pp. 295-299.

[5] **M. A. K. A. A. S. S. V. Krishnan,** "Design and Implementation of Arduino Based Home Automation System with Smart Access Control Using RFID," *2015 IEEE International Conference on Computational Intelligence and Computing Research*, Chennai, India, 2015, pp. 1-4.

[6] **V. S. Nagarajan, K. R. S. S. Y. Chaitanya, and M. S. M. R. S. K. Reddy,** "Arduino Based Home Automation System," *2014 IEEE International Conference on Control and Computing (ICCC)*, Pune, India, 2014, pp. 150-154.

[7] **P. Sharma, R. R. Tiwari, and R. K. Sharma,** "Internet of Things Based Home Automation with Smart Security System," *2017 IEEE International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2017, pp. 964-968.

[8] **P. R. Pradhan, B. K. Sahu, and S. K. Sahu,** "Arduino Based Home Automation System Using IoT," *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2017, pp. 1-5.

[9] **R. G. Rojas, A. G. Santana, and A. G. Perez,** "IoT Based Smart Home Automation Using Arduino," *2016 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Yogyakarta, Indonesia, 2016, pp. 1-4.

[10] **S. R. S. A. S. M. M. J. E. C. Young,** "Home Automation Using Arduino and Smart Devices," *2019 International Conference on Computing and Network Communications (CoCoNet)*, Chennai, India, 2019, pp. 213-217.

[11] **C. P. Y. G. G. Kumar,** "Design and Implementation of Arduino-based Home Automation System with Wireless Control," *2014 IEEE International Conference on Communication and Signal Processing*, Melmaruvathur, India, 2014, pp. 758-762.

[12] **A. T. S. G. K. G. Ali,** "Smart Home Automation Using Arduino and IoT," *2019 International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, India, 2019, pp. 228-233.

[13] **G. T. R. G. A. R. M. R. P. Kumar,** "Smart Home Automation Using Arduino and Internet of Things (IoT) Based Technologies," *2017 IEEE 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Bangalore, India, 2017, pp. 1-6.

[14] **S. S. A. K. S. M. V. Kumar,** "Design and Development of Smart Home System using Arduino," *2015 IEEE International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Coimbatore, India, 2015, pp. 1-6.

[15] **S. B. M. A. G. R. Y. A. K. G. T. B. S. Shah,** "Design and Implementation of a Smart Home Automation System Using Arduino and IoT," *2018 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2018, pp. 2116-2120.

[16] **S. Kumar, S. K. Agarwal, and M. K. Srivastava,** "IoT-Based Smart Home Automation Using Arduino and ESP32," *2019 International Conference on Signal Processing and Communication (ICSC)*, Kochi, India, 2019, pp. 104-109.

[17] **M. G. A. S. Kumar, and P. G. M. Singh,** "Design of Smart Home Automation System Using Sensors and Arduino," *2017 IEEE 2nd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2017, pp. 303-307.

[18] **M. B. N. D. M. A. B. S. A. F. B. S. G. K. R. Y. Y. Jain,** "Smart Home Automation System Using Arduino and IoT," *2018 IEEE International Conference on Research in Intelligent and Computing in Engineering (RICE)*, Coimbatore, India, 2018, pp. 87-92.

[19] **A. B. S. S. A. R. B. Patel,** "Smart Home Automation with IoT Using Arduino," *2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, Coimbatore, India, 2017, pp. 102-107.