A blue text on a white background

Description automatically generatedA black and grey logo with a graduation cap and book

Description automatically generated

**GOVERNMENT OF TAMILNADU**

DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

**IoT AND ITS APPLICATIONS**

**HOST INSTITUTION**

xxxxx

COIMBATORE – 04

**TRAINING PARTNER**

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

DATE:

|  |  |
| --- | --- |
| Name | Roll No |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Table Of Contents**

|  |  |  |
| --- | --- | --- |
| **S no** | **Title** | **Page no** |
| 1 | Introduction |  |
| 2 | Abstract |  |
| 3 | Hardware and Software Requirements |  |
| 4 | Block Diagram And  working |  |
| 5 | Code |  |
| 6 | Output |  |
| 7 | Cloud Output  &Applications |  |
| 8 | Conclusion |  |
| 9 | References |  |

**IOT BASED RFID TAG DOOR MANAGEMENT SYSTEM**

**INTRODUCTION:**

In an era where technology continually transforms our daily lives, the Internet of Things (IoT) has emerged as a groundbreaking innovation, connecting various devices and systems to create smarter, more efficient environments. One remarkable application of IoT technology is in access control and security. Among the many IoT-based solutions, the integration of the ESP32 microcontroller with RFID (Radio Frequency Identification) technology for door management stands out. This combination offers a robust and versatile approach to secure and efficient access control.

The ESP32, a powerful and cost-effective microcontroller with built-in Wi-Fi and Bluetooth capabilities, serves as the backbone of this RFID Tag Door Management System. By leveraging the ESP32’s wireless connectivity and processing power, this system provides seamless and real-time control over door access, making it ideal for a range of environments—from residential and commercial buildings to industrial facilities.

This ESP32-based RFID system not only enhances security with reliable, contactless RFID tag identification but also simplifies the management of access points. It enables centralized control, remote monitoring, and real-time updates, all while maintaining a high level of efficiency and scalability.

**ABSTRACT:**

* The primary objective of this project is to design and implement a door management system that leverages the ESP32's connectivity to enhance security, efficiency, and user convenience. The system employs RFID tags for secure and contactless access, with the ESP32 managing communication between RFID readers and a central control unit. This setup allows for real-time monitoring and control of access points, offering a streamlined solution for both residential and commercial environments.
* Overall, this ESP32-based RFID Tag Door Management System represents a significant advancement in access control technology, combining the reliability of RFID with the versatility of IoT to deliver a sophisticated, user-friendly solution for modern security challenges.
* Top of Form
* Bottom of Form

**PROPOSED SYSTEM:**

**System Architecture:**

* + **ESP32 Microcontroller**:
    - **Role**: Acts as the central control unit, managing communication between the RFID reader, the RFID tags, and the network.
    - **Features**: Equipped with built-in Wi-Fi and Bluetooth capabilities for seamless connectivity and remote access.
  + **RFID Reader**:
    - **Role**: Reads the unique identification data from RFID tags.
    - **Features**: Connects to the ESP32 to transmit tag information for processing and validation.
  + **RFID Tags**:
    - **Role**: Serve as the unique identifiers for users or authorized personnel.
    - **Features**: Each tag has a distinct identifier that is read by the RFID reader to grant or deny access.
* **Relay Module:** Acts as a switch to control the street lights, turning them on or off based on the ESP32's commands.
* **Power Supply:** Provides the necessary power for the ESP32, sensors, and relay module.
* **Wi-Fi Connectivity:** The ESP32 connects to the internet via Wi-Fi to send data to the cloud server and receive control commands.
* **ThingzMate Cloud Platform**: Provides a cloud-based interface for remote monitoring and control. The ESP32 uploads the fan's operating status to the ThingzMate platform, allowing users to monitor the system from anywhere via a mobile device or web interface.

**COMPONENTS REQUIRED:**

**SOFTWARE**

* Arduino Ide.
* Thingzmate.

**HARDWARE:**

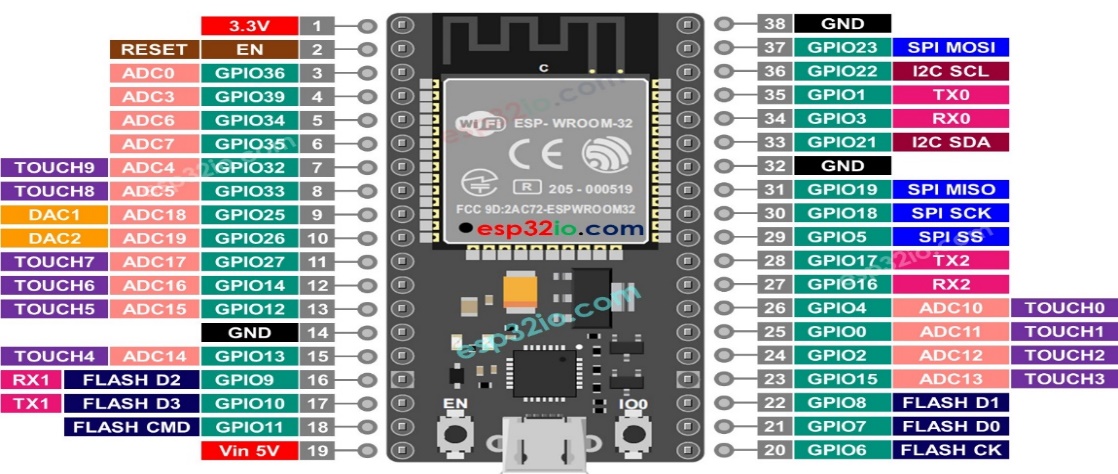
1. ESP32 Development Board
2. RFID Reader Module (e.g., MFRC522)
3. RFID Tags
4. Relay Module (for controlling the door)
5. Power Supply
6. Connecting Wires
7. Breadboard (optional, for prototyping)

**ESP32 MICROCONTROLLER:**

The ESP32 is a highly versatile microcontroller developed by Espressif Systems, designed for a wide range of applications, particularly in the Internet of Things (IoT) space. It is renowned for its combination of high performance, integrated wireless connectivity, and a rich set of features, all at a low cost. The ESP32 is commonly used in projects that require both Wi-Fi and Bluetooth capabilities, making it suitable for smart home devices, sensor networks, and wearable technology.

**Key Features:**

* **Dual-Core Processor**: Features a dual-core Tensilica LX6 microprocessor running up to 240 MHz.
* **Connectivity**: Includes Wi-Fi (802.11 b/g/n) and Bluetooth (Classic and BLE).
* **Memory**: Typically comes with 520 KB of SRAM and supports external flash memory.
* **I/O Pins**: Offers numerous GPIO (General Purpose Input/Output) pins with various functionalities.



**2.RFID SENSOR:**



1. **RFID Tags:** These are small devices attached to the items being tracked. They contain a microchip that stores data and an antenna to communicate with the RFID reader.

* + Passive Tags: Do not have their own power source. They are powered by the electromagnetic field generated by the RFID reader.
  + Active Tags: Have their own power source (usually a battery) and can transmit data actively over a longer range.
  + Semi-Passive Tags: Have a battery to power the chip but rely on the reader’s field for communication.
* **RFID Reader:** The device that emits radio waves and reads the data from RFID tags. It consists of an antenna and a transceiver.
* **Antenna:** The component that sends and receives radio waves between the RFID reader and tags.

**2.Data Transmission:**

* **Communication:** When an RFID tag enters the range of an RFID reader, the reader sends a radio signal that powers the tag (for passive tags) or communicates with it (for active and semi-passive tags).
* **Data Exchange:** The tag responds with its unique identifier and possibly other stored data. The reader captures this data and sends it to a backend system for processing.

**3.Frequency Bands:**

* Low Frequency (LF): 125 kHz – 134.2 kHz. Shorter read ranges and lower data transfer rates but better performance in challenging environments.
* High Frequency (HF): 13.56 MHz. Commonly used for contactless payment systems and access control.
* Ultra High Frequency (UHF): 856 MHz – 960 MHz. Longer read ranges and higher data transfer rates, suitable for logistics and supply chain management.
* Microwave: 2.45 GHz or 5.8 GHz. Used for high-speed applications and longer read ranges.

**RELAY MODULE:**

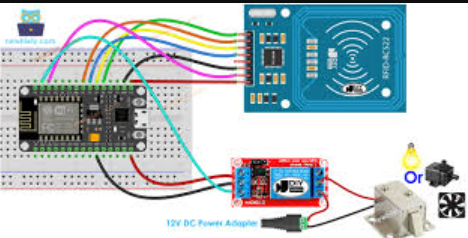
* A relay consists of an electromagnet, a movable armature, a set of contacts, and a spring.
* When a low-power signal (typically from a GPIO pin of a microcontroller) is applied to the relay's coil, it generates a magnetic field that pulls the armature, closing or opening the contacts.
* This action either completes or interrupts the circuit connected to the high-power device (in this case, street lights).



**POWER SUPPLY:**

* Power supply is a reference to a source of electrical power.
* A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU.
* The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.
* This power supply section is required to convert AC signal to DC signal and also to reduce the amplitude of the signal.
* The available voltage signal from the main is 230V/50Hz which is an AC voltage, but the required is DC voltage with the amplitude of +5V and +12V for varies applications

**CIRCUIT DIAGRAM:**

****

**CODE:**

#include <SPI.h>

#include <MFRC522.h>

#include <WiFi.h>

#include <HTTPClient.h>

// Pin definitions

#define SS\_PIN    5  // ESP32 pin GPIO5 for RFID reader

#define RST\_PIN   22 // ESP32 pin GPIO22 for RFID reader

#define RELAY\_PIN   2  // ESP32 pin GPIO2 for LED

MFRC522 rfid(SS\_PIN, RST\_PIN);  // Create MFRC522 instance

// WiFi credentials

const char\* ssid = "MI11X";         // Replace with your WiFi SSID

const char\* password = "Bharath123"; // Replace with your WiFi password

// ThingzMate settings

const String serverName = "https://console.thingzmate.com/api/v1/device-types/kommakahome2/devices/thingzkit2/uplink"; // Replace with your ThingzMate endpoint URL

const String AuthorizationToken = "Bearer de49c0ef7f6e04ccb2a7ee4ce36ec719"; // Replace with your ThingzMate Authorization Token

// Define UIDs for access control

byte uidOn[] = {0xE3, 0x68, 0x5B, 0x28}; // UID for turning LED on

byte uidOff[] = {0xE3, 0x19, 0xCE, 0x13}; // UID for turning LED off

void setup() {

  Serial.begin(115200);        // Initialize Serial Monitor

  SPI.begin();                 // Initialize SPI bus

  rfid.PCD\_Init();             // Initialize MFRC522 RFID reader

  pinMode(RELAY\_PIN, OUTPUT);    // Set RELAY\_PIN as an output

  digitalWrite(RELAY\_PIN, LOW);  // Ensure the LED is off initially

  // Connect to WiFi

  WiFi.begin(ssid, password);

  Serial.println();

  Serial.print("Connecting to ");

  Serial.println(ssid);

  // Wait until connected to WiFi

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

    delay(500);

    Serial.print(".");

  }

  Serial.println("Connected to WiFi");

  Serial.print("IP Address: ");

  Serial.println(WiFi.localIP());

}

void loop() {

  if (rfid.PICC\_IsNewCardPresent()) { // Check if a new card is present

    if (rfid.PICC\_ReadCardSerial()) { // Read the card's UID

      boolean accessGranted = false;

      String statusMessage;

      // Check UID and perform corresponding action

      if (compareUID(rfid.uid.uidByte, uidOn)) {

        accessGranted = true;

        Serial.println("DOOR OPEN");

        digitalWrite(RELAY\_PIN, HIGH); // Turn on the LED

        statusMessage = "DOOR OPEN";

      }

      else if (compareUID(rfid.uid.uidByte, uidOff)) {

        accessGranted = true;

        Serial.println("ACCESS DENIED");

        digitalWrite(RELAY\_PIN, LOW); // Turn off the LED

        statusMessage = "ACCESS DENIED";

      }

      else {

        Serial.print("Unknown UID: ");

        for (int i = 0; i < rfid.uid.size; i++) {

          Serial.print(rfid.uid.uidByte[i] < 0x10 ? "0" : "");

          Serial.print(rfid.uid.uidByte[i], HEX);

        }

        Serial.println();

        statusMessage = "Unknown UID";

      }

      // Send data to ThingzMate

      sendDataToThingzMate(statusMessage);

      rfid.PICC\_HaltA(); // Halt PICC

      rfid.PCD\_StopCrypto1(); // Stop encryption on PCD

    }

  }

}

void sendDataToThingzMate(String message) {

  if (WiFi.status() == WL\_CONNECTED) {

    HTTPClient http;

    http.begin(serverName); // Specify the URL

    http.addHeader("Content-Type", "application/json"); // Add content-type header

    http.addHeader("Authorization", AuthorizationToken); // Add authorization header

    // Create JSON payload

    String payload = "{\"message\":\"" + message + "\"}";

    int httpResponseCode = http.POST(payload); // Send POST request

    if (httpResponseCode > 0) {

      String responsePayload = http.getString();

      Serial.println(httpResponseCode);

      Serial.println(responsePayload);

    } else {

      Serial.print("Error code: ");

      Serial.println(httpResponseCode);

    }

    http.end(); // Free resources

  } else {

    Serial.println("Error: Not connected to WiFi");

  }

}

// Function to compare the UID read from the RFID reader with a valid UID

boolean compareUID(byte \*uid1, byte \*uid2) {

  for (int i = 0; i < 4; i++) {

    if (uid1[i] != uid2[i]) {

      return false;

    }

  }

  return true;

}

**WORKING :**

**1. RFID Tag Scanning**

* **Step 1**: The user presents their RFID tag/card to the RFID reader placed near the door.
* **Step 2**: The RFID reader scans the tag and reads its unique ID.

**2. Data Processing**

* **Step 3**: The RFID reader sends the tag's ID to the micro-controller (like an ESP32).
* **Step 4**: The micro-controller processes the ID to determine if it is authorized to open the door. This involves comparing the scanned UID with a list of authorized IDs stored in the micro-controller.

**3. Door Control**

* **Step 5**: If the UID is authorized, the micro-controller sends a signal to the door mechanism (e.g., an electronic lock or relay) to unlock the door.
* **Step 6**: If the UID is not authorized, the system denies access and can provide feedback (e.g., a buzzer sound).

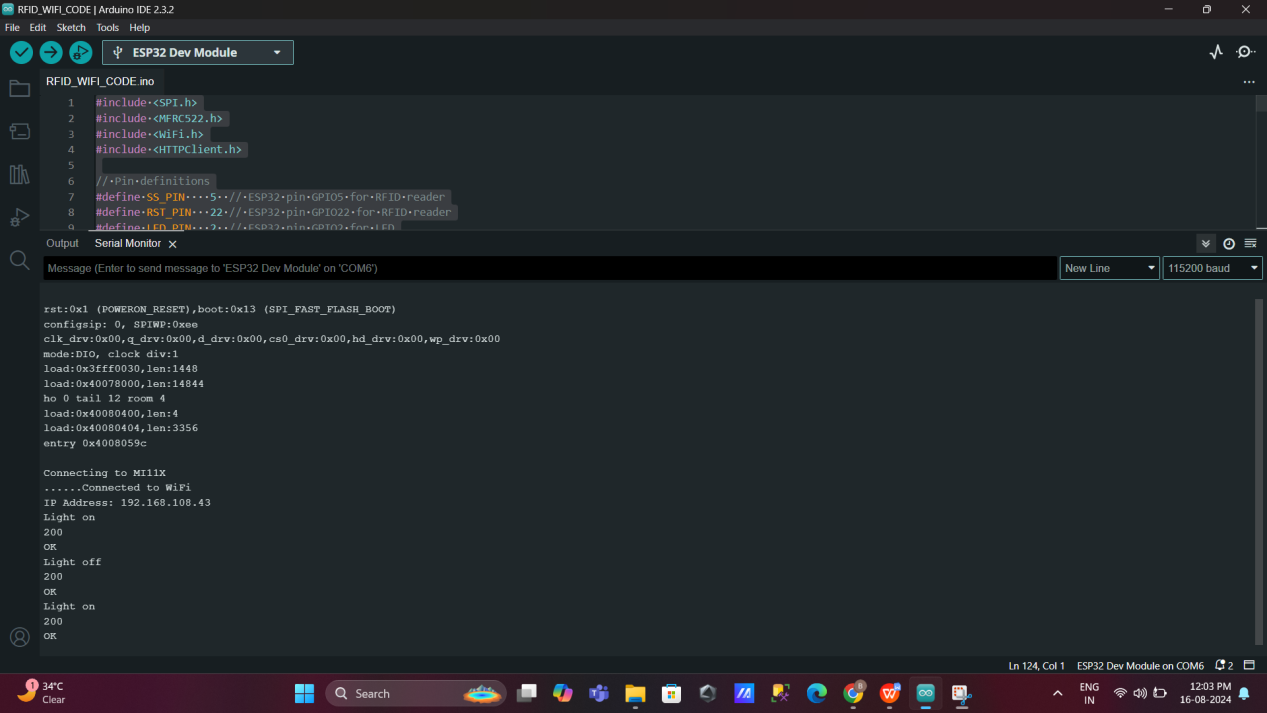
**4. Cloud Integration**

* **Step 7**: The micro-controller sends data to the cloud service, including information about the access attempt (authorized or unauthorized), time of the event, and other relevant details.
* **Step 8**: The cloud service stores this data and can provide a web or mobile interface for monitoring and control. It can also send notifications or alerts based on specific events (e.g., unauthorized access attempts).

**5. Remote Management**

* **Step 9**: Users or administrators can access the cloud service to monitor the door's status, view access logs, and manage authorized RFID tags remotely.
* **Step 10**: Administrators can update access permissions or perform other management tasks through the cloud interface.

**OUTPUT:**



**STEPS TO FOLLOW:**

**·** Check that Serial.begin(9600); is present in your setup() function.

· Match the baud rate in the Serial Monitor to 9600.

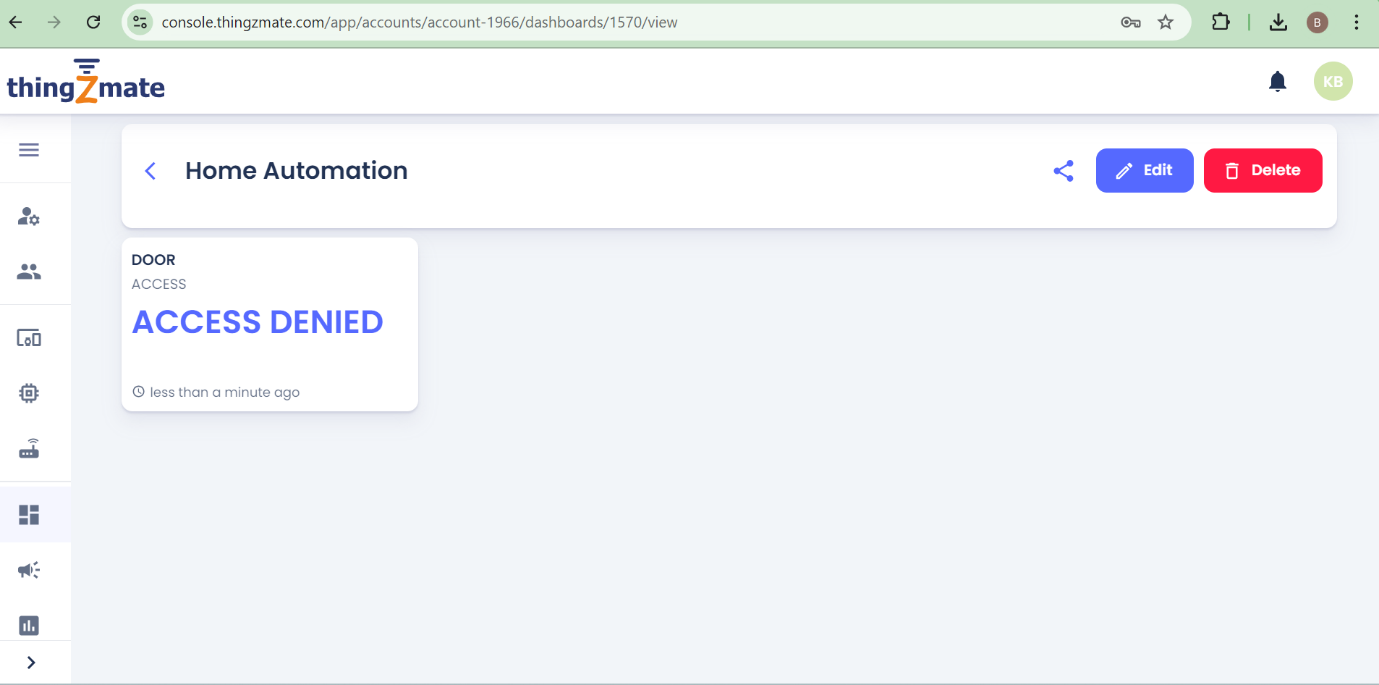
· Verify the COM port in the Serial Monitor is correctly selected.

· Ensure no other application is using the same COM port.

· Restart your IDE and try again.

**IoT Integration:**

* **Cloud Integration:** Describe how data is sent to a cloud service for monitoring (e.g., using MQTT or HTTP).
* **CLOUD OUTPUT:**



* **Dashboard:** Discuss creating a user interface for monitoring and controlling the street lights remotely.

**Testing and Results:**

* **Test Cases:** Describe various scenarios tested, such as different lighting conditions and motion detection.
* **Results:** Provide screenshots or data showing the system's response to the test cases.

**APPLICATIONS:**

1.Residential Buildings: Secure entry to homes or apartments.

2.Commercial Buildings: Access control for offices and restricted areas.

3.Educational Institutions: Manage access to classrooms and administrative offices.

4.Hotels: RFID key cards for guest room access and amenities.

5.Gyms and Fitness Centers: Control access to facilities and locker rooms.

6. Public Transportation: Secure entry to restricted areas in stations or airports.

7. Warehouses: Secure storage areas and track inventory.

8. Data Centers: Control access to server rooms and sensitive infrastructure.

9. Hospitals: Manage access to operating rooms and medicine storage.

10. Laboratories: Secure access to labs and sensitive equipment.

11. Conferences and Exhibitions: RFID badges for attendee access and tracking.

**CONCLUSION:**

**1. Functionality and Purpose**

**RFID Technology**: RFID (Radio Frequency Identification) tags are used to uniquely identify and authenticate users. An RFID reader scans these tags and communicates with a micro-controller to determine access permissions.

**Relay Module**: Acts as an intermediary between the low-power control signal from the micro-controller and the high-power electric door lock. The relay module enables the micro-controller to control high-voltage devices safely.

**2. System Components**

**Microcontroller**: Such as ESP32 or Arduino, manages the RFID reader, processes tag data, and controls the relay module.

**RFID Reader**: Reads the RFID tags and provides the UID (Unique Identifier) to the micro-controller.

**Relay Module**: Switches the electric lock mechanism on or off based on the control signal from the micro-controller.

**Electric Door Lock**: Controlled by the relay module to lock or unlock the door.

**3. Integration and Wiring**

**RFID Reader to Microcontroller**: Connects through SPI (Serial Peripheral Interface) for communication.

**Relay Module to Microcontroller**: Connects via a GPIO pin to control the relay. The relay module also requires power (VCC) and ground (GND) connections.

**Relay Module to Electric Lock**: Connects through the relay’s COM (Common) and NO (Normally Open) terminals, with the lock’s power supply integrated into the relay circuit.

**4. Software and Control**

**Code Implementation**: The code running on the micro-controller reads the RFID tag, checks if it matches the authorized tags, and activates or deactivates the relay accordingly. This controls the door lock based on the tag’s UID

**Control Logic**: Typically involves activating the relay to unlock the door when an authorized tag is detected ,and deactivating it after a certain period to re-lock the door.

**5. Considerations**

**Power Requirements**: Ensure that the relay module and electric lock are compatible in terms of voltage and current ratings. The micro-controller must be able to safely drive the relay.

**Safety**: Use relays with appropriate isolation to protect the micro-controller and other low-voltage components from high-voltage spikes.

**Reliability**: Test the system thoroughly to ensure that the RFID reader, relay module, and electric lock operate reliably under all expected conditions.

**6.** **Potential Enhancements**

**Wireless Connectivity**: Integration with cloud platforms or wireless communication modules (e.g., WiFi, Bluetooth) to allow remote monitoring and control.

**Data Logging and Alerts**: Implement features for logging access events and sending alerts for unauthorized access attempts or system malfunctions.

**Enhanced Security**: Use encrypted communication and advanced authentication mechanisms to further secure the access control system

**References:**

* Cite all references used in the project, including websites, papers, and books.
* Referred the IOT websites and esp32 websites for the circuit connections.