RFID BASED SMART ACCESS SYSTEM USING RELAY

The domain of the Project

Embedded Systems and IoT

Mentor

MEHAK MAJEED
(Junior Engineer , ATFAAL Innovations)

By

Ms.GADIGE SUVARNA LAKSHMI
(B.Tech, 4th year pursuing)

Period of the project

November 2025 to December2025

SURE ProEd

Declaration

We, the undersigned, hereby declare that the project entitled "**RFID BASED SMART ACCESS SYSTEM USING RELAY**" has been carried out by us under the mentorship of MEHAK MAJEED and with the support of SURE Trust during the period June 2025 to December2025.

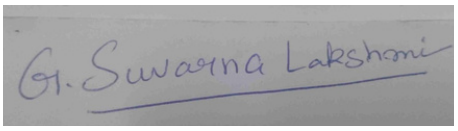
This project has been undertaken for the benefit of gaining hands-on experience in industry-relevant technologies, enhancing our practical knowledge in IoT and embedded systems, and access control solutions, and preparing us for relative prospective employment opportunities.

We further declare that this work is a result of our genuine effort and has not been copied or reproduced from any other source.

Name

Ms. GADIGE SUVARNA LAKSHMI

Signature



Mentor

MEHAK MAJEED

(Junior Engineer—ATFAAL innovations)

Signature

Seal & Signature

Prof. Radhakumari
Executive Director & Founder
SURE ProEd

Table of contents

Executive Summary

Security and controlled access are critical requirements in modern institutions such as offices, educational campuses, laboratories, and residential buildings. Conventional locking systems based on mechanical keys are prone to security risks, misuse, and lack flexibility in access management. To address these challenges, this project proposes an **RFID-based Smart Access Control System** using ESP32, Arduino UNO, relay, and a solenoid door lock.

The proposed system provides secure, automated, and time-restricted access by allowing only authorized RFID cards to unlock the door during predefined time windows. The ESP32 functions as the main controller, responsible for reading RFID card data, verifying authorized users, synchronizing real-time clock information using **Wi-Fi and NTP**, and displaying system status on an LCD screen. Audible feedback is provided using a buzzer for both successful and unsuccessful access attempts.

To safely control the high-power 12V solenoid lock, the ESP32 communicates with an Arduino UNO through UART communication. The Arduino activates a relay module, which switches the 12-volt power supply required to operate the solenoid lock. This approach ensures electrical isolation and protects low-voltage control circuits from high-current loads.

The system automatically unlocks the door upon authentication and relocks it after a predefined delay, ensuring minimal human intervention. The design is cost-effective, reliable, scalable, and energy-efficient, making it suitable for real-world deployment. The project demonstrates effective integration of embedded systems, IoT technology, and access control mechanisms,

.

Introduction

Background and Context

In the modern world, security and access management have become increasingly important due to the growth of institutions, workplaces, and residential infrastructures. Traditional access control methods such as mechanical locks and physical keys are widely used but have several drawbacks. These systems are vulnerable to issues like key loss, duplication, unauthorized access, and lack of accountability, which can compromise security.

With advancements in technology, there has been a shift toward automated and electronic access control systems. These systems aim to provide improved security, convenience, and reliability by using digital authentication methods instead of physical keys. Technologies such as RFID, embedded systems, and wireless communication have enabled the development of smarter and more efficient access control solutions.

Problem Statement

Conventional access control systems that rely on mechanical keys and manual supervision suffer from several limitations such as unauthorized key duplication, loss of keys, lack of access tracking, and absence of time-based control. These systems do not provide sufficient security in environments where restricted and monitored access is required. Additionally, manual locking mechanisms depend heavily on human intervention, which can lead to errors, security breaches, and inefficiency.

There is a growing need for a secure, automated, and reliable access control system that can authenticate users effectively, restrict access based on predefined conditions such as time, and control high-power locking mechanisms safely. The system should be cost-effective, easy to implement, and capable of integrating modern technologies to enhance security while minimizing manual effort.

Scope of the Project

The proposed system is designed to:

- Design and implementation of a smart access control system using embedded technology
- Authentication of users through contactless RFID cards
- Restriction of access based on predefined time windows
- Automation of door locking and unlocking using a solenoid lock and relay
- Safe control of high-power devices using relay isolation
- Integration of microcontrollers and wireless communication for real-time operation
- Display of access status through LCD and buzzer alerts

The system focuses on providing controlled access to restricted areas by authenticating users through contactless identification and enforcing predefined access conditions. The project aims to reduce dependency on traditional mechanical locking systems and minimize human intervention in access management.

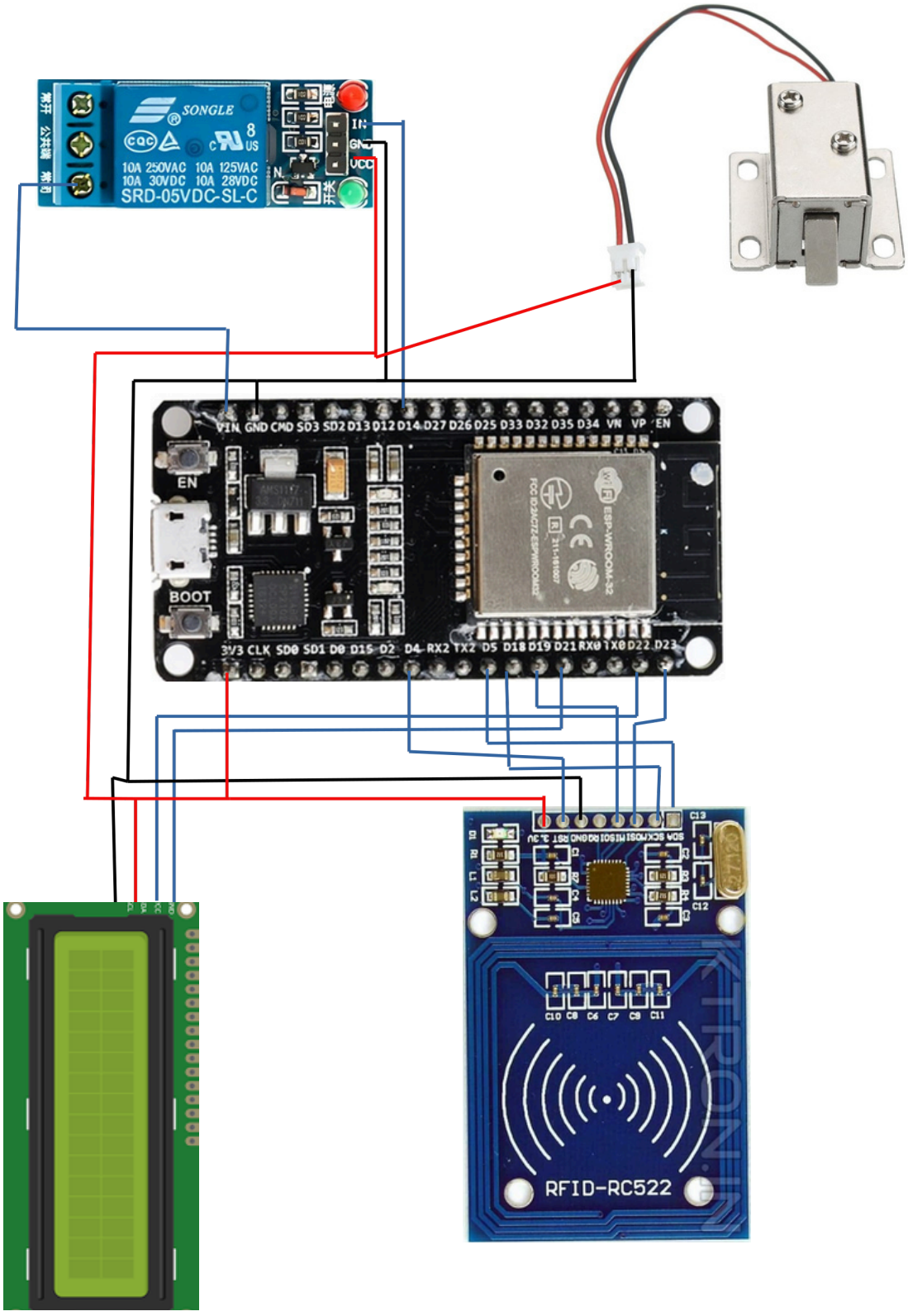
Innovation Component

The innovative aspects of this project are:

- Integration of RFID authentication with time-based access control, ensuring entry is allowed only during authorized time windows
- Use of ESP32 with Wi-Fi and NTP to obtain accurate real-time data without requiring an external RTC module
- Dual-controller architecture using ESP32 and Arduino UNO to safely separate control logic and high-power load handling
- Implementation of UART communication between ESP32 and Arduino for reliable command transfer
- Safe control of a 12V solenoid lock using a relay, providing electrical isolation and protecting low-voltage components
- Automatic door locking mechanism after a predefined delay to enhance security and prevent unauthorized entry
- Real-time user feedback through LCD display and buzzer alerts for access status
- Cost-effective design using easily available and low-power components

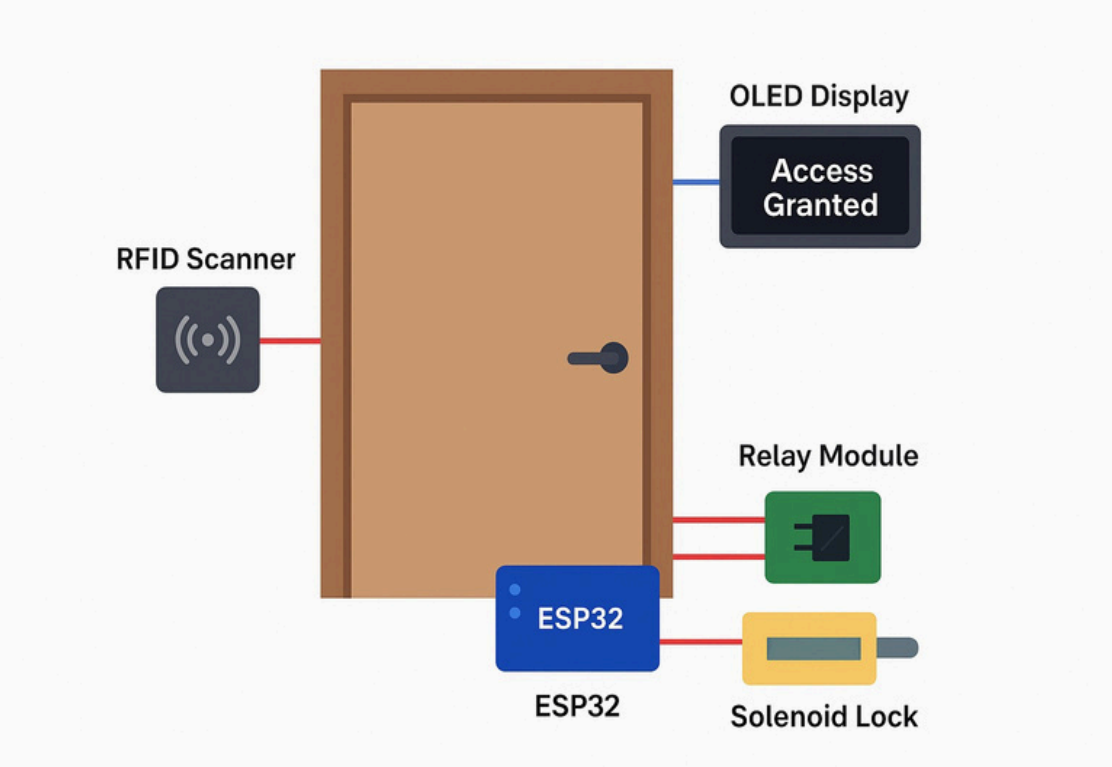
System Diagram

Fig.1.a. Electrical Representation of RFID based Smart access system using relay



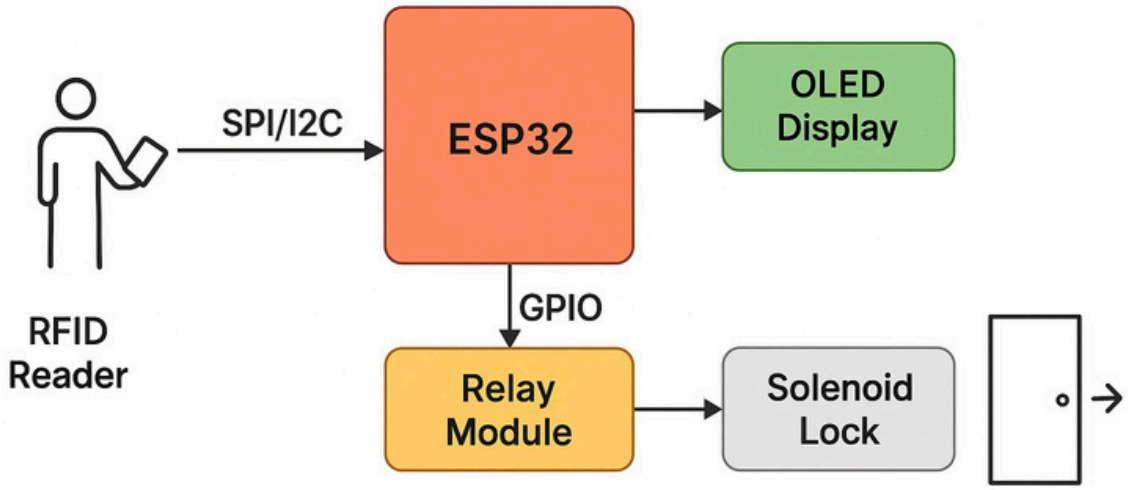
Electrical Representation: This diagram illustrates the power supply and wiring connections of the system. The ESP32, OLED, and RFID reader are powered through regulated DC supply, while the relay and solenoid lock operate on 5V with isolation. Common ground connections and protection devices such as flyback diodes are included to ensure safe operation.

Fig.1.b. Mechanical Representation of RFID based smart access System using Relay



Mechanical Representation: This diagram highlights the physical arrangement of the system components. The RFID reader and LCD display are placed at the user access point, while the ESP32 and relay are secured inside an enclosure. The solenoid lock is mechanically fixed on the door or cabinet to control entry. User interaction occurs through RFID scanning, and access is physically granted by the actuation of the solenoid.

Fig.1.c. Electronic Representation of RFID based Smart access system using relay



Electronic Representation: This diagram shows the logical flow of signals between different components. The ESP32 acts as the central controller, receiving input from the RFID reader, displaying locking status on the LCD display through I2C communication, and sending control signals to the relay module. The relay in turn drives the solenoid lock, enabling or restricting access.

Project Objectives

The goal of the project "**RFID-Based Smart Access Control System**" is to design and implement a secure, automated, and IoT-enabled access control solution that overcomes the limitations of traditional key-based locking systems and manual access monitoring. The proposed system ensures that only authorized users are allowed access, enforces time-based restrictions, reduces human intervention, and improves overall security, accountability, and efficiency.

Detailed Objectives

1. Develop a Smart and Automated Access Control System

- Design a system capable of controlling access to restricted areas such as laboratories, offices, hostels, classrooms, and secure rooms.
- Replace traditional mechanical locks with an automated electronic access mechanism.
- Minimize dependency on manual supervision and human intervention.

2. Implement RFID-Based User Authentication

- Integrate an RFID reader (MFRC522) to scan contactless RFID cards.
- Read and extract the unique identification number (UID) from each RFID card.
- Validate scanned UIDs against a predefined list of authorized users stored in the ESP32.
- Prevent access to unauthorized users by rejecting invalid or unregistered RFID cards.

3. Enforce Time-Based Access Control

- Implement time-restricted access policies to allow entry only during authorized hours.
- Use Wi-Fi-enabled real-time clock synchronization to ensure accurate time validation.
- Prevent misuse of access outside permitted time windows, even for authorized users.

4. Provide Real-Time User Feedback

- Display system status messages such as “Scan Your Card,” “Access Granted,” “Access Denied,” and “Time Restricted” on the LCD/OLED display.
- Provide audible alerts using a buzzer to indicate successful or failed access attempts.
- Improve user awareness and interaction with immediate feedback.

5. Enable Secure Physical Access Control

- Control a 12V solenoid door lock using a relay module to physically lock and unlock the door.
- Ensure automatic relocking of the door after a predefined delay to maintain security.
- Provide electrical isolation between control circuitry and high-power devices for safe operation.

6.Integrate IoT for System Synchronization and Monitoring

- Connect the ESP32 to a Wi-Fi network for internet connectivity.
- Use Network Time Protocol (NTP) for accurate and automatic time synchronization.
- Enable future expansion for cloud-based monitoring and remote access management.

7. Ensure System Safety, Reliability, and Efficiency

- Protect low-voltage components from high current and voltage using relay isolation.
- Ensure stable and continuous operation under real-world conditions.
- Optimize power usage for energy-efficient operation.

8. Develop a Cost-Effective and Scalable Solution

- Use low-cost, easily available hardware components to keep the system affordable.
- Design the architecture to support scalability for multiple access points.
- Provide a strong foundation for future enhancements such as:
  - Biometric authentication
  - Mobile application integration
  - Cloud-based access logging
  - AI-based security systems

Methodology and Results

The methodology of the proposed RFID-Based Smart Access Control System involves a systematic approach that integrates hardware components, embedded programming, and IoT technologies to achieve secure and automated access control. The system is designed to authenticate users, enforce time-based access, and control physical locking mechanisms safely and efficiently.

1. Hardware Setup

The system consists of the following hardware components:

1. ESP32 Development Board:

- Acts as the main controller of the system
- Handles RFID authentication and decision-making
- Manages Wi-Fi connectivity and NTP time synchronization

2. Implement RFID-based User Authentication

- Integrate an RFID reader (MFRC522) to scan user ID cards.
- Validate scanned UIDs against a registered database stored on the ESP32.
- Prevent unauthorized users from accessing resources.

3. Enable Secure Physical Access Control

- Use a relay-driven solenoid lock to physically control access to resources.
- Ensure automatic relocking after a fixed duration (5 seconds) to maintain the security.

4. Provide Real-Time User Feedback

- Display user identity (UID and name) and access status (Granted/Denied) on the LCD screen.
- Improve user experience by providing instant confirmation of access decisions.

5.Power Supply (12V DC)

- Dedicated power source for solenoid door lock
- Provides sufficient current required for solenoid operation

2. Software Setup

The software is developed and tested using:

- Arduino IDE – For coding, compiling, and uploading programs to ESP32.
- Programming Languages: Embedded C/C++ for microcontroller logic.

- Libraries Used
  - SPI Library:Enables SPI communication between ESP32 and RFID reader
  - Wire Library:Enables I2C communication between ESP32 and LCD display.
  - NTPClient Library:Used to obtain real-time clock information from an NTP server
- **Communication Protocols**
  - SPI Protocol is used for communication between ESP32 and RFID reader.
  - I2C Protocol is used for communication between ESP32 and LCD display.
  - UART Protocol is used for communication between ESP32 and Arduino UNO.
  - Wi-Fi Protocol is used for internet connectivity and time synchronization.

3. Workflow of the System

The step-by-step working of the RFID-Based Smart Access Control System is as follows:

1. System Initialization.

- RFID reader (MFRC522), LCD display, buzzer, and UART communication are configured.
- ESP32 connects to the Wi-Fi network.
- Network Time Protocol (NTP) client is initialized to obtain accurate real-time data.

2. User Authentication

- The user presents an RFID card near the RFID reader.
- The MFRC522 reads the card and extracts the unique identification number (UID).

3. Access Decision

The ESP32 compares the scanned UID with the list of authorized UIDs stored in memory.

- If the UID is valid,in time window,gives access
- ESP32 sends an UNLOCK command to the Arduino via UART.
- If the UID is invalid or the time condition fails,Access is denied.
- The solenoid lock remains closed.

4. Physical Door Control

- Upon receiving the UNLOCK command, the Arduino activates the relay.
- The relay supplies 12V power to the solenoid lock.
- The solenoid unlocks the door for a predefined duration and is locked

5. User Feedback

- The LCD displays messages such as “Access Granted,” “Access Denied,” or “Time Restricted.” and buzzes sound.

4. Results

Access Granted:
Registered users (User A, User B, User C, User D) were able to access the system successfully.

- Each authorized RFID card was correctly recognized by the system.
- Upon successful authentication and valid time window:
- The solenoid lock unlocked for the predefined duration.
- The access event was processed successfully without delay.

Access Denied:

- Unknown or unregistered RFID cards were denied access.
- The solenoid lock remained in the locked state.
- Audible alert (buzzer) was triggered for failed access attempts.
- Unauthorized access attempts were consistently blocked.

Time-Based Restriction:

- Authorized users attempting access outside the allowed time window were denied.
- The system correctly identified time violations using real-time clock synchronization.

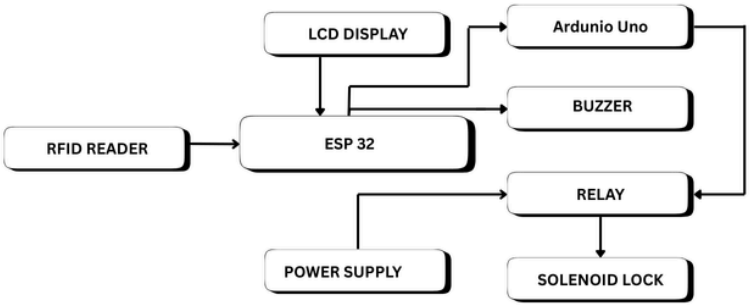
System Reliability:

- The system handled multiple consecutive access attempts without failure.
- UART communication between ESP32 and Arduino remained stable.
- Relay and solenoid operation was consistent across all tests.
- No system crashes or unexpected behavior were observed during testing

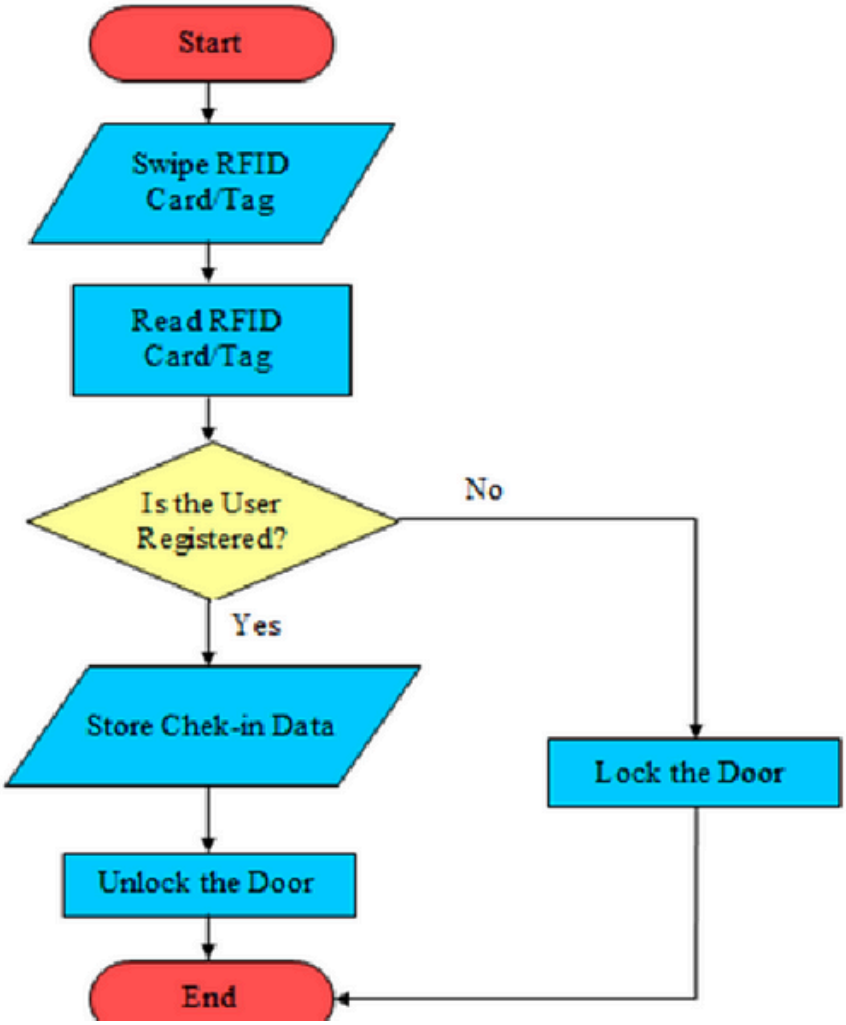
Key Outcomes

- Successfully demonstrated secure RFID-based access control.
- Verified time-based and user-based access restriction.
- Implemented real-time LCD/OLED feedback for transparency.
- Confirmed reliable relay and solenoid lock operation.
- System operated correctly under multiple test cases such as:
  - Valid access
  - Invalid access
  - Time-restricted access
  - Continuous access attempts

BLOCK DIAGRAM:



FLOW CHART :



5. Pictures

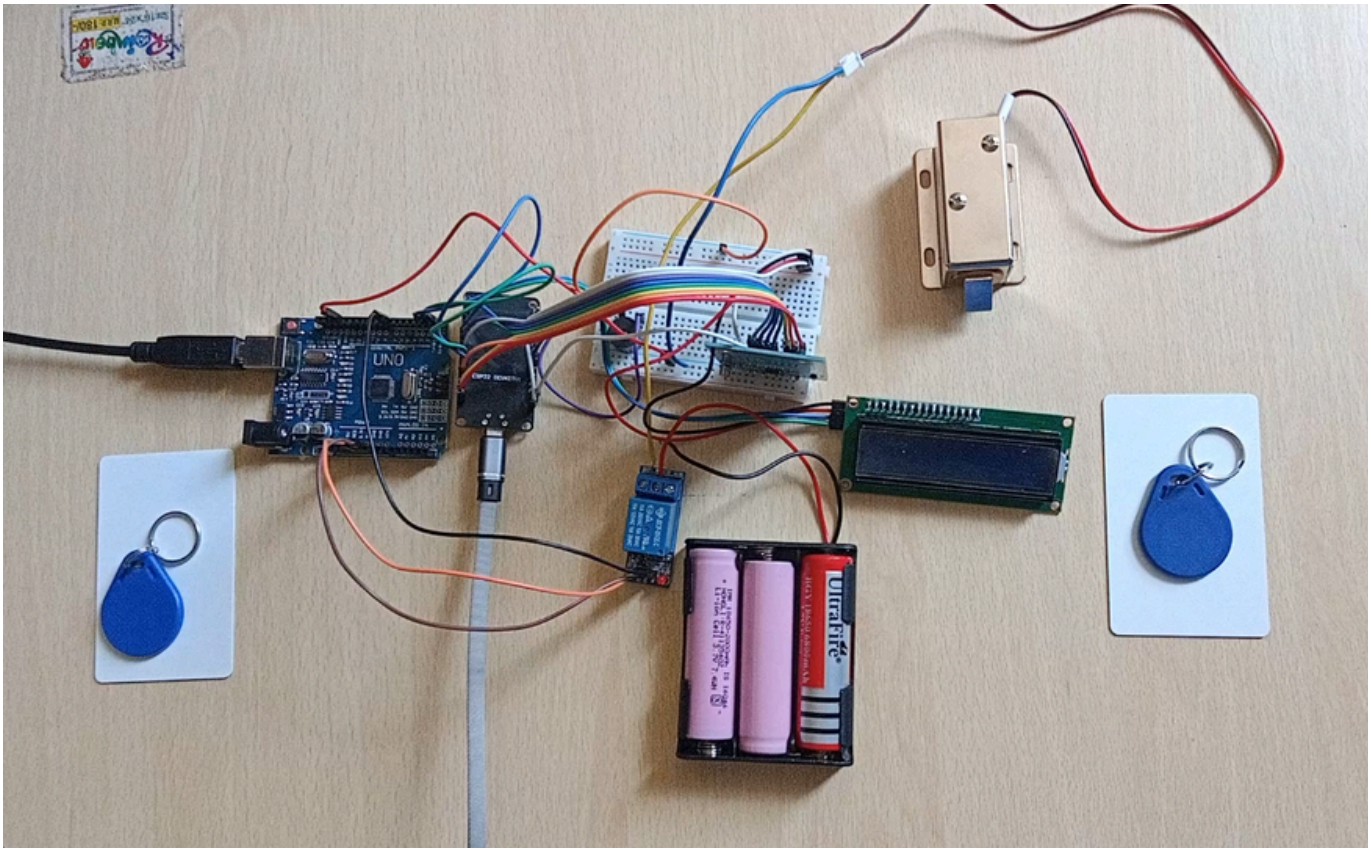


Fig.2. CIRCUIT CONNECTIONS

COMPONENTS :

Fig.2.a. ESP32 MICRO CONTROLLER

Fig.2.b. LCD DISPLAY

Fig.2.c. RELAY MODULE

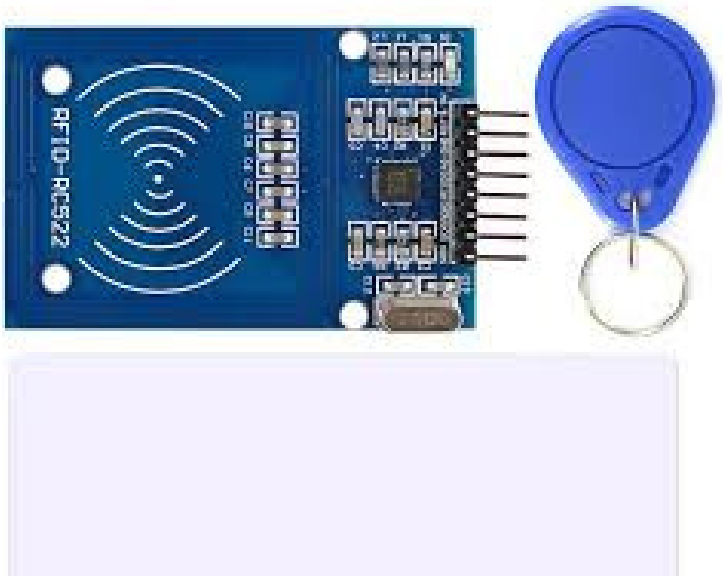
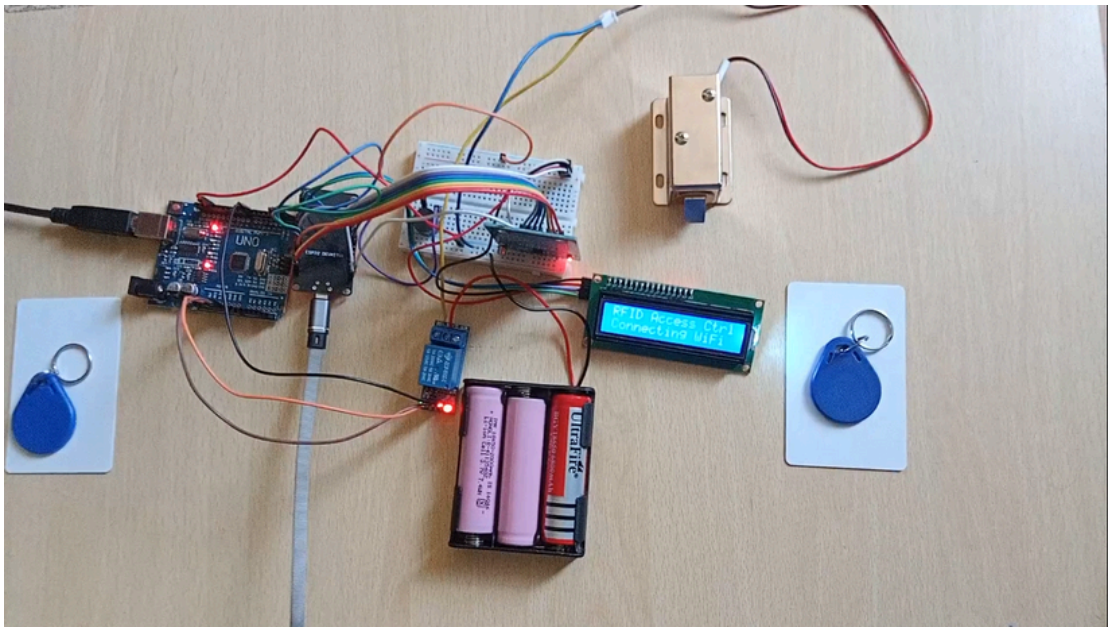
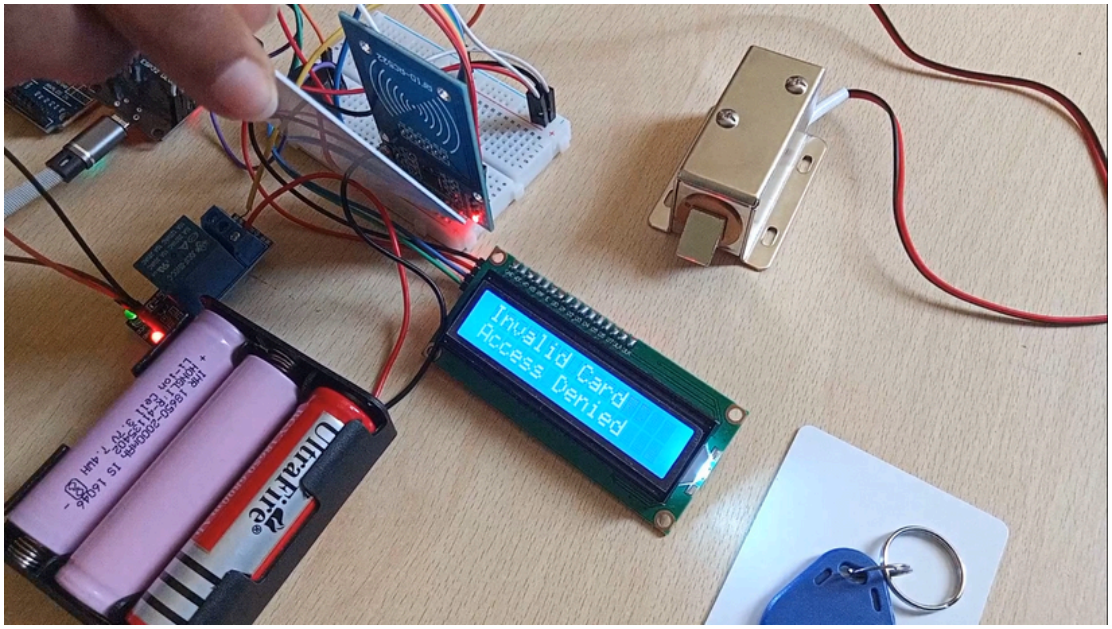


Fig.2.d. RC522 RFID MODULE

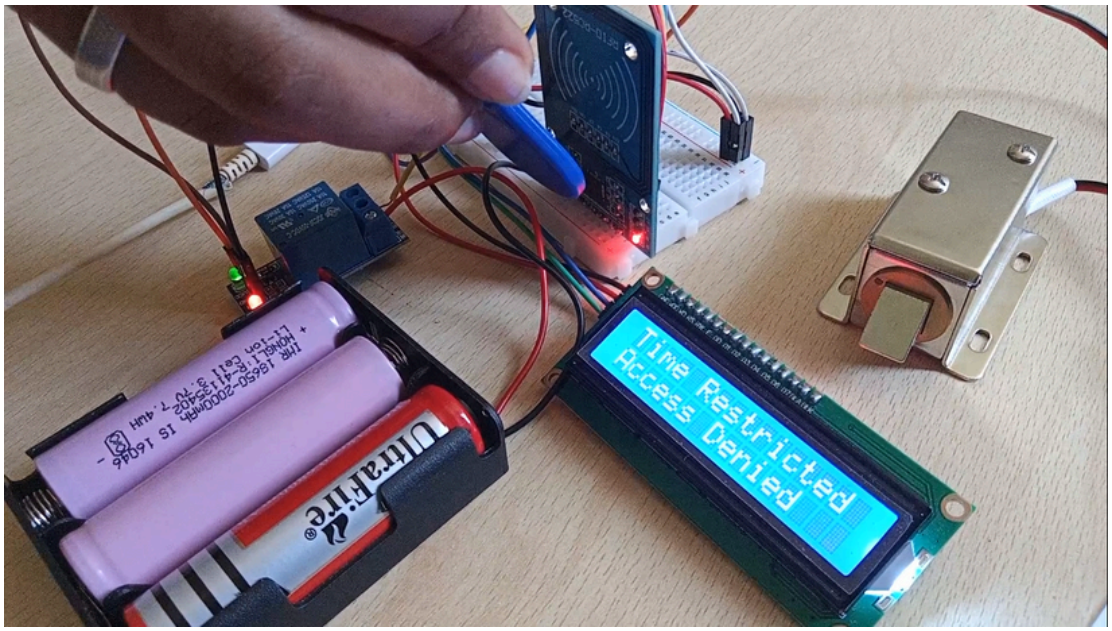
Fig.2.e. SOLENOID LOCK

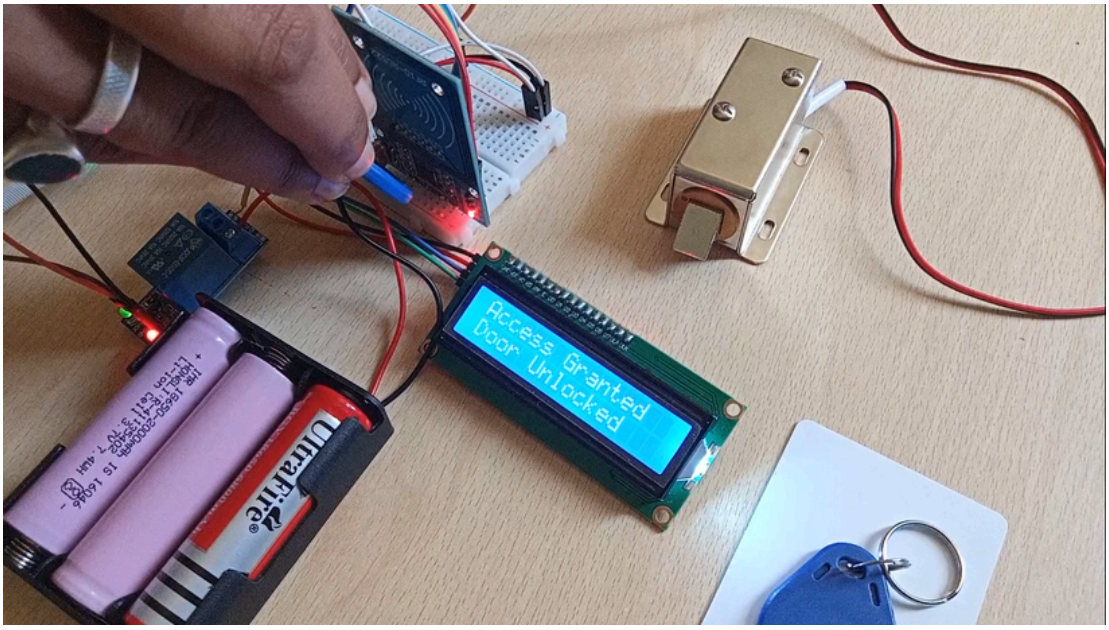
FINAL OUPUT:


UNO
RFID Access Ctrl
Connecting WiFi
UltraFire


Invalid card
Access Denied
UltraFire

FINAL OUPUT:


Time Restricted
Access Denied
UltraFire


Access Granted
Door Unlocked
UltraFire

Observations & Analysis

RFID Reader Performance:

- The MFRC522 RFID reader reliably detected RFID cards within a range of 2–4 cm.
- Detection failed beyond this range, ensuring controlled and secure scanning.

Relay Module Operation:

- The relay module switched ON and OFF cleanly during lock and unlock operations.
- Electrical isolation provided safe switching of the 12V solenoid lock without affecting low-voltage control circuits.

Solenoid Lock Behavior:

- The solenoid lock responded quickly to relay activation and unlocked the door reliably.
- Continuous activation beyond 5 seconds caused slight heating, which was mitigated by implementing automatic relocking logic.

UART Communication Stability:

- UART communication between ESP32 and Arduino UNO was stable and error-free.
- LOCK and UNLOCK commands were transmitted without delay or data loss.

LCD Display Visibility:

- The LCD display clearly showed access messages under indoor lighting conditions.
- Visibility slightly reduced under bright ambient light, but messages remained readable.

Time-Based Access Accuracy

- NTP-based time synchronization provided accurate real-time access validation.
- Authorized users were correctly denied access outside permitted time windows.

Error Handling and Safety:

- Invalid RFID cards were consistently rejected.
- After power loss and restart, the system returned to a safe locked state, preventing unauthorized access.

Social / Industry relevance of the project

The RFID-Based Smart Access Control System is highly relevant in both social and industrial contexts, as it addresses real-world challenges related to security, automation, controlled access, and accountability. The system provides a reliable and cost-effective solution for managing access to restricted areas without manual intervention.

1.Social Relevance

Educational Institutions:

- Colleges and universities can deploy the system to secure laboratories, classrooms, libraries, and staff rooms.
- Ensures that only authorized students and staff can access restricted areas.
- Prevents misuse of expensive laboratory equipment and infrastructure.

Hostels and Residential Buildings

- Provides secure and automated entry for hostel residents.
- Eliminates the need for manual registers and physical keys.
- Enhances safety for students and residents.

Public Buildings and Community Center:

- Can be used in community halls, study centers, and training institutes.
- Ensures controlled access to facilities.
- Useful in rural development centers supported by government or NGOs

Public Resource Management:

- Controls access to medicine storage rooms, operation theaters, and staff-only areas.
- Prevents unauthorized entry and improves patient safety.

2. IndustryRelevance

Corporate Offices:

- Manages secure access to offices, meeting rooms, server rooms, and confidential departments.
- Ensures that only authorized employees can enter sensitive areas.
- Reduces security risks and improves organizational control.

Industrial Environments:

- Restricts access to machinery rooms, control panels, and hazardous zones.
- Prevents unauthorized personnel from operating dangerous or high-value equipment.
- Enhances workplace safety and compliance.

Research & Development Labs:

- Controls access to specialized instruments and research facilities.
- Ensures only trained and authorized personnel can operate sensitive equipment.
- Improves accountability and reduces equipment misuse.

Warehouses and Storage Facilities:

- Secures access to inventory storage areas.
- Prevents theft and unauthorized handling of goods.
- Improves inventory management and safety

3. Benefits Across Domains

- **Security**: Prevents unauthorized access to restricted areas.
- **Automation**: Eliminates manual key-based systems and registers.
- **Accountability**: Access attempts can be monitored and audited.
- **Efficiency**: Reduces human error and administrative workload.
- **Scalability**: Can be expanded for multiple doors or facilities.
- **Cost-Effectiveness**: Uses low-cost components suitable for large-scale deployment.
- **Transparency**: Provides real-time status and system feedback

Learning and Reflection

The development of the RFID-Based Smart Access Control System was a valuable learning experience that provided practical exposure to embedded systems, IoT technologies, and real-world security applications. This project helped bridge the gap between theoretical concepts and hands-on implementation.

1. Technical Learnings

Embedded Systems & IoT Integration

- Learned to program and configure the ESP32 microcontroller for multi-functional tasks such as RFID authentication, Wi-Fi connectivity, time synchronization, and communication with Arduino.
- Gained practical understanding of how embedded systems interact with peripherals in real-time applications.

RFID Authentication

- Gained experience in interfacing the MFRC522 RFID reader with ESP32.
- Learned how to read UID values from RFID cards and validate them against registered users.
- Implemented secure authentication logic to prevent unauthorized access.

Display Interfacing

- Implemented 16×2 I2C LCD display interfacing using the LiquidCrystal_I2C library.
- Displayed real-time system messages such as Scan Your Card, Access Granted, Access Denied, and Time Restricted.
- Improved user interaction through visual feedback.

Time-Based Access Control (IoT Concept)

- Learned how to use Wi-Fi connectivity and Network Time Protocol (NTP) for accurate real-time clock synchronization.
- Implemented time-restricted access logic without using an external RTC module.
- Understood the importance of time-based security in access control systems.

System Testing & Debugging

- Learned to troubleshoot hardware connections, power supply issues, and grounding problems.
- Debugged RFID detection issues and optimized response time.
- Verified system behavior under different test cases such as valid access, invalid access, and time-restricted access.
- Improved system reliability through repeated testing and error handling

2. Project Management

- Developed skills in dividing project tasks among team members such as:
  - Hardware setup and wiring
  - Embedded coding for ESP32 and Arduino
  - Testing and debugging
  - Documentation and report preparation
- Improved time management skills by planning milestones and completing tasks within given deadlines.
- Learned to balance multiple project components such as hardware integration, software development, and testing simultaneously.
- Enhanced team collaboration through regular discussions, idea sharing, and troubleshooting sessions.
- Improved communication skills by coordinating work and resolving technical issues collectively.

3. Reflections on the Experience

- The project demonstrated the real-world relevance of embedded and IoT-based solutions in solving practical security challenges like access control.
- Provided valuable hands-on exposure to technologies such as RFID authentication, microcontrollers, relay control, and time-based automation that are widely used in industry.
- Highlighted the importance of system integration, ensuring seamless interaction between hardware, software, communication protocols, and power systems.
- Helped each team member gain confidence in applying classroom concepts to real-time, industry-relevant problems.

Main Code and File Structure

Software Implementation

Project Directory Structure:

To maintainclarity and modularity, the project is organized into a structured folder format.The main Arduino sketch resides at the root level, required external libraries are grouped in a libraries folder, and a README.txt file documents pin connections and setup procedure.

Main Code

(a). Libraries and Configuration

```
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#include <NTPClient.h>

// ===== PIN DEFINITIONS =====
#define SS_PIN   5   // RFID SDA
#define RST_PIN  22  // RFID RST
#define BUZZER_PIN 26

// ===== RFID AND LCD =====
MFRC522 rfid(SS_PIN, RST_PIN);
LiquidCrystal_I2C lcd(0x27, 16, 2);

// ===== WIFI CREDENTIALS =====
const char* ssid = "realmeC25_Y";
const char* password = "gadige@777";

// ===== NTP CONFIGURATION =====
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800,
60000); // GMT+5:30 IST
```

```
// ===== AUTHORIZED RFID CARDS =====
const byte validUID1[] = {0xE3, 0x05, 0xAF, 0x13};
const byte validUID2[] = {0xFF, 0xA1, 0xDF, 0xC4};

void setup() {
 Serial.begin(115200); // Debugging
  Serial2.begin(9600);  //  UART  to  Arduino  (TX2=17,
RX2=16 on ESP32 by default)

 SPI.begin();
 rfid.PCD_Init();

 pinMode(BUZZER_PIN, OUTPUT);
 digitalWrite(BUZZER_PIN, LOW);

 lcd.init();
 lcd.backlight();
 lcd.setCursor(0, 0);
 lcd.print("RFID Access Ctrl");

 // WiFi Setup
 WiFi.begin(ssid, password);
 lcd.setCursor(0, 1);
 lcd.print("Connecting WiFi");
 while (WiFi.status() != WL_CONNECTED) {
 delay(500);
 Serial.print(".");
 }
 Serial.println("\nWiFi Connected");
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("WiFi Connected");
 delay(1000);
 lcd.setCursor(0, 0);
 lcd.print("SCAN YOUR CARD");
 timeClient.begin();
}
```

```
void loop() {
 timeClient.update();
 int hour = timeClient.getHours();
 bool inWindow1 = (hour >= 9 && hour < 11);
 bool inWindow2 = (hour >= 15 && hour < 16);
if(!rfid.PICC_IsNewCardPresent()                    ||
!rfid.PICC_ReadCardSerial()) {
 delay(100);
 return;
 }
 Serial.print("Card UID: ");
 for (byte i = 0; i < rfid.uid.size; i++) {
 Serial.print(rfid.uid.uidByte[i], HEX);
 Serial.print(" ");
 }
 Serial.println();
 bool isAuthorized = checkUID(rfid.uid.uidByte);
 if (isAuthorized) {
 if (inWindow1 || inWindow2) {
 Serial.println("Access Granted");
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Access Granted");
 lcd.setCursor(0, 1);
 lcd.print("Door Unlocked");
 digitalWrite(BUZZER_PIN, HIGH);
 delay(500);
 digitalWrite(BUZZER_PIN, LOW);
  Serial2.println("UNLOCK");  // 🔹 Send command to
Arduino
 delay(3000);
 Serial2.println("LOCK"); // 🔹 Lock again after 3 sec
 lcd.setCursor(0, 1);
 lcd.print("Door Locked ");
 lcd.setCursor(0, 0);
 lcd.print("SCAN YOUR CARD");
 }
```

```
else {
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Time Restricted");
 lcd.setCursor(0, 1);
 lcd.print("Access Denied");

 digitalWrite(BUZZER_PIN, HIGH);
 delay(1000);
 digitalWrite(BUZZER_PIN, LOW);
 }
 } else {
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Invalid Card");
 lcd.setCursor(0, 1);
 lcd.print("Access Denied");

 digitalWrite(BUZZER_PIN, HIGH);
 delay(1000);
 digitalWrite(BUZZER_PIN, LOW);
 }

 rfid.PICC_HaltA();
 rfid.PCD_StopCrypto1();
 delay(500);
}

bool checkUID(byte *uid) {
 if (memcmp(uid, validUID1, 4) == 0) return true;
 if (memcmp(uid, validUID2, 4) == 0) return true;
 return false;
}
```

Conclusion and Future Scope

Conclusion

The **RFID-Based Smart Access Control System** developed in this project successfully demonstrates the application of embedded systems and IoT technologies to address real-world security and access management challenges. The system was designed to provide a secure, automated, and efficient alternative to traditional key-based and manual access control mechanisms, which are often prone to misuse, loss, and human error.

The project effectively integrates RFID-based authentication to uniquely identify users and ensure that access is granted only to authorized individuals. By using the ESP32 microcontroller, the system is capable of handling multiple tasks such as RFID card detection, access decision-making, Wi-Fi connectivity, and real-time clock synchronization. The use of **Network Time Protocol (NTP)** enables accurate time-based access control without requiring additional hardware, making the system both cost-effective and reliable.

A key strength of the system is its safe and robust physical access control mechanism. The combination of an Arduino UNO, relay module, and 12V solenoid lock ensures secure door locking and unlocking while protecting low-voltage control components from high-current loads. The implementation of automatic relocking logic further enhances system safety by preventing prolonged solenoid activation and overheating.

The system also provides clear and immediate user feedback through an LCD display and buzzer, improving usability and transparency. Users are informed of access status such as Access Granted, Access Denied, or Time Restricted, which enhances trust and ease of interaction with the system. The stable UART communication between the ESP32 and Arduino ensures reliable command transmission and consistent system behavior.

The project also successfully implements safe physical door control using a combination of Arduino UNO, relay module, and 12V solenoid lock. This dual-controller architecture provides electrical isolation, protects low-voltage components, and ensures reliable operation. The automatic relocking mechanism enhances security and prevents overheating of the solenoid lock.

Future Scope

1. Biometric Authentication Integration

- Fingerprint or facial recognition modules can be added along with RFID.
- Enhances security by providing multi-factor authentication.
- Prevents misuse of lost or stolen RFID cards.

2. Cloud-Based Access Logging

- Integrate cloud platforms to store access logs securely.
- Enables remote monitoring and auditing of access history.
- Useful for institutions and industries requiring compliance and reporting.

3. Mobile Application Control

- Develop a mobile application to manage access permissions.
- Allows administrators to grant or revoke access remotely.
- Users can receive real-time notifications for access events.

4. Multi-Door and Multi-User Expansion

- Extend the system to control multiple doors using a centralized controller.
- Support large-scale deployment in campuses, offices, and industries.
- Enable role-based access for different user categories.

5. AI-Based Security Enhancements

- Integrate AI algorithms to analyze access patterns.
- Detect suspicious behavior or repeated unauthorized attempts.
- Improve predictive security and anomaly detection.

6. Emergency and Safety Features

- Add emergency unlock mechanisms for fire or evacuation scenarios.
- Integrate with fire alarms or safety systems.
- Ensure compliance with safety regulations.

7. Power Optimization and Backup

- Implement low-power modes to reduce energy consumption.
- Add battery backup or solar power support for uninterrupted operation.
- Improve system reliability during power failures.

8. Web-Based Admin Dashboard

- Develop a web interface for managing users and access policies.
- Visualize access logs and system status in real time.
- Simplifies system administration.

9. Integration with Smart Building Systems

- Connect the system with smart lighting, HVAC, and surveillance cameras.
- Enable full smart-building automation.
- Enhance energy efficiency and security coordination.