EMBEDDED SYSTEM PROJECT REPORT

SMART LOCK SYSTEM

BATCH 8:

810021106085-Syed Rabiyathul Basariya S M

810021106086-Tarun M R

810021106088-Thiruveni R

810021106089-Tony Chacko Thomas

INTRODUCTION

Advancements in IoT technology have revolutionized home security systems, offering innovative solutions for convenience and safety. One such project is the IoT smart RFID door lock system, which utilizes the NodeMCU ESP8266 microcontroller board. This project integrates RFID technology with IoT capabilities, allowing users to unlock doors remotely using their smartphones or RFID cards.

By leveraging the power of NodeMCU ESP8266, a versatile and cost-effective IoT platform, this project enables real-time monitoring and control of door access. Through a user-friendly interface, individuals can manage access permissions, receive notifications on door status, and enhance security measures.

With its potential for customization and scalability, the IoT smart RFID door lock system presents an exciting opportunity for DIY enthusiasts, tech enthusiasts, and homeowners seeking modern solutions for home security and automation

PROBLEM STATEMENT

Conventional Key-based Systems: Traditional door lock systems reliant on keys pose security risks such as key duplication, loss, or theft, leading to unauthorized access.

Remote Monitoring and Control: In today's busy lifestyle, the ability to monitor and control door access remotely is crucial for homeowners, property managers, and businesses.

Integration with IoT: With the rise of IoT technology, there's a growing demand for smart solutions that can be integrated into existing home automation systems for enhanced functionality and convenience.

User-friendly Interface: Many existing electronic door lock systems lack user-friendly interfaces and may require complex setups, hindering widespread adoption and usability.

LITERATURE SURVEY

IoT smart RFID door lock system project integrates RFID technology with NodeMCU ESP8266, offering a comprehensive approach to access control. RFID technology provides a secure and reliable means of identification, ensuring that only authorized individuals can gain entry. Each RFID tag carries a unique identifier, minimizing the risk of unauthorized access and enhancing security measures.

NodeMCU ESP8266 complements this by adding IoT capabilities to the system. It enables remote monitoring and control of the door lock through a Wi-Fi connection, allowing users to manage access permissions, receive real-time notifications about door status, and automate locking/unlocking processes.This integration with IoT not only enhances convenience for users but also opens up possibilities for further customization

By combining RFID technology with NodeMCU ESP8266, the project achieves a balance between security, affordability, scalability, and user-friendliness, making it a robust solution for modern access control needs in both residential and commercial settings.

HARDWARE

1. NodeMCU ESP8266: Microcontroller board with built-in Wi-Fi capabilities.

2. RFID Reader: Reads RFID tags/cards for user authentication.

3. Servo Motor: Actuates the physical lock/unlock mechanism of the door.

4. Relay Module: Optionally used to control high-power devices like electronic locks.

5. Power Supply: Provides stable power (usually 5V) to the system.

6. Wiring and Connections:

    Connect RFID Reader to NodeMCU (SPI or UART).

    Connect Servo Motor to NodeMCU GPIO pins.

    Connect Relay Module if used.

    Connect power supply to NodeMCU and components.

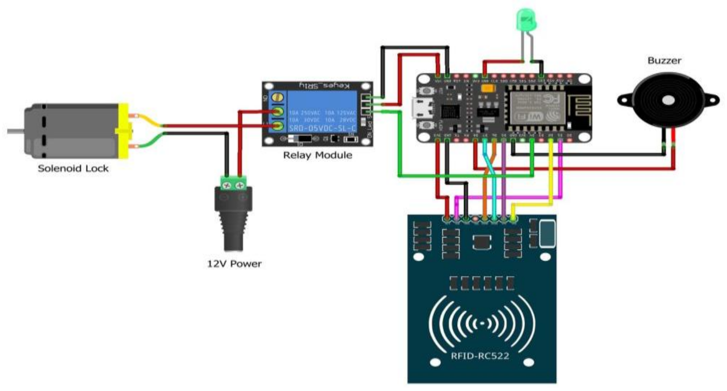
7. Optional Components:

    LEDs: For status indication.

    Buzzer: For alarms or notifications.

    Door sensors: For detecting door status (open/close).

CIRCUIT



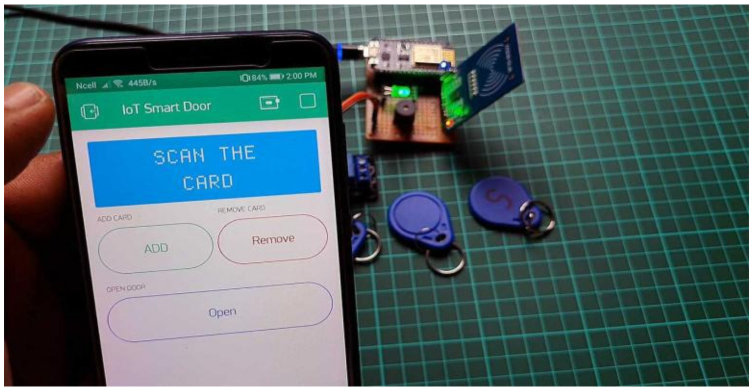
SOFTWARE

```
#define BLYNK_TEMPLATE_ID "TMPLmNMgxxyz"

#define BLYNK_DEVICE_NAME "IoT Smart Door"

#define BLYNK_FIRMWARE_VERSION        "0.1.0"

#define BLYNK_PRINT Serial

//#define BLYNK_DEBUG

#define APP_DEBUG

//#define USE_SPARKFUN_BLYNK_BOARD

#define USE_NODE_MCU_BOARD

//#define USE_WITTY_CLOUD_BOARD

//#define USE_WEMOS_D1_MINI

#include "BlynkEdgent.h"

#include <SPI.h>

#include <MFRC522.h>

#define SS_PIN 4  // sda pin D2

#define RST_PIN 5 // RST (flash) pin D2

#define relay 2 //Relay Pin D4

#define BUZZER 15 //buzzer pin D8

MFRC522 mfrc522(SS_PIN, RST_PIN);        // Create MFRC522 instance.

SimpleTimer timer;

int button1 = 0;

int button2 = 0;

int button3 = 0;

WidgetTerminal terminal(V2);

void setup()

{

  Serial.begin(115200);
  delay(100);
  BlynkEdgent.begin();
  pinMode(relay, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  SPI.begin();                // Init SPI bus
  mfrc522.PCD_Init();       // Init MFRC522 card
  timer.setInterval(3000L, iot_rfid);
}
void loop() {
  timer.run(); // Initiates SimpleTimer
  BlynkEdgent.run();
}
void iot_rfid()
{
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }
  Serial.print("Card UID:");
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], DEC);
  }
  Serial.println();
  byte piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
  //    Serial.print("PICC type: ");
  //Serial.println(mfrc522.PICC_GetTypeName(piccType));
  if (        piccType != MFRC522::PICC_TYPE_MIFARE_MINI
            &&        piccType != MFRC522::PICC_TYPE_MIFARE_1K
            &&        piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    //Serial.println("This sample only works with MIFARE Classic cards.");
    return;
  }
  if ( ((mfrc522.uid.uidByte[0] == 129) && (mfrc522.uid.uidByte[1] == 163) &&
(mfrc522.uid.uidByte[2] == 220) && (mfrc522.uid.uidByte[3] == 121)) && (button1 == 1) )
  {
    Serial.println("Access Granted to Mr. Alsan");
    Blynk.virtualWrite(V2, "Access Granted to Mr. Alsan" );
    digitalWrite(relay, LOW);
    delay(5000);
    digitalWrite(relay, HIGH);
  }
  else if (( (mfrc522.uid.uidByte[0] == 134) && (mfrc522.uid.uidByte[1] == 96) &&
(mfrc522.uid.uidByte[2] == 128) && (mfrc522.uid.uidByte[3] == 248)) && (button2 == 1) )
  {
    Serial.println("Access Granted to Mrs. Aashika");
    Blynk.virtualWrite(V2, "Access Granted to Mrs. Aashika" );
    digitalWrite(relay, LOW);

    delay(5000);

    digitalWrite(relay, HIGH);

  }

  else if ( ((mfrc522.uid.uidByte[0] == 57) && (mfrc522.uid.uidByte[1] == 234) &&
(mfrc522.uid.uidByte[2] == 176) && (mfrc522.uid.uidByte[3] == 109)) && (button3 == 1) )

  {

    Serial.println("Access Granted to Dr. Smith");

    Blynk.virtualWrite(V2, "Access Granted to Dr. Smith" );

    digitalWrite(relay, LOW);

    delay(5000);

    digitalWrite(relay, HIGH);

  }

 else {

    Serial.println("Unregistered user");

    Blynk.virtualWrite(V2, "Unregistered user Trying to Access your Door Lock " );

    Serial.println("Access denied");

    Blynk.virtualWrite(V2, "Access denied");

    digitalWrite(BUZZER, HIGH);

    delay(2000);

    digitalWrite(BUZZER, LOW);

  }

}

BLYNK_WRITE(V3)

{

  button1 = param.asInt(); // assigning incoming value from pin V3 to a variable

}
```

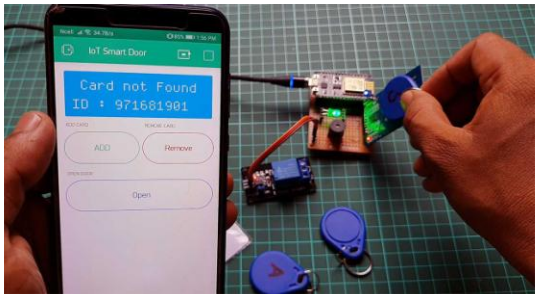
```
BLYNK_WRITE(V4)
{
  button2 = param.asInt(); // assigning incoming value from pin V4 to a variable
}
BLYNK_WRITE(V5)
{
  button3 = param.asInt(); // assigning incoming value from pin V5 to a variable
}
```

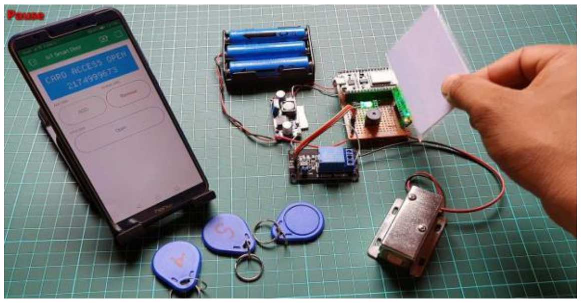
RESULT

INPUT:



OUTPUT





CONCLUSION

ADVANTAGES :

1. Remote Access: You can control and monitor your door lock from anywhere with an internet connection, adding convenience and security.

2. Integration: It can be integrated with other IoT devices or smart home systems, allowing for automation and customization based on user preferences.

3. Enhanced Security: With features like real-time alerts and activity logs, you can keep track of who enters and exits your property, enhancing security.

DISADVANTAGES :

1. Security Risks: IoT devices, including NodeMCU-based systems, can be vulnerable to hacking or cyber attacks if not properly secured, potentially compromising the safety of your home or property.

2. Reliability: Reliability issues may arise due to factors such as internet connectivity issues, power outages, or software bugs, leading to unexpected behavior or malfunctions in the door lock system.

3. Complexity: Setting up and configuring an IoT-based door lock system using NodeMCU may require technical expertise, which could be challenging for users with limited technical knowledge or experience.

LIMITATIONS :

1. Limited Range: The range of NodeMCU's Wi-Fi connectivity is typically limited to the range of the Wi-Fi router, which may restrict its use in larger properties or areas with poor Wi-Fi coverage.

2. Power Source Dependency: NodeMCU requires a stable power source to operate, which means that power outages or battery failures could render the door lock system temporarily non-functional.