



Documentație Proiect Sistem de Securitate pentru Control Access

Student:

Toth Antonio-Roberto, an II, CTI, seria B,
grupa 30228



Cuprins

1. Introducere
2. Listă abreviațiuni și simboluri
3. Componente
4. Descrierea realizării
5. Observații
6. Concluzii privind utilitatea practică a lucrării și aspecte economice
7. Listă referințelor bibliografice



1. Introducere

Proiectul de față urmărește dezvoltarea unui sistem de securitate pentru controlul accesului, bazat pe tehnologia RFID, combinată cu senzori de mediu. Acest sistem nu doar că verifică accesul autorizat, dar și monitorizează condițiile de mediu în timp real, asigurând un nivel suplimentar de siguranță și confort. În prezent, nevoia de securitate sporită este o preocupare majoră atât pentru companii, cât și pentru utilizatorii individuali. Tehnologia RFID este din ce în ce mai utilizată datorită fiabilității și eficienței sale în gestionarea accesului. Adăugarea senzorilor de temperatură, umiditate și gaz completează acest sistem, oferind o soluție completă și integrată pentru gestionarea accesului și monitorizarea mediului. Proiectul propus se distinge prin integrarea mai multor tehnologii și senzori într-un sistem unic și compact. Această integrare nu doar că asigură securitatea accesului, dar oferă și o monitorizare constantă a condițiilor de mediu, ceea ce poate fi deosebit de util în diverse scenarii, de la locuințe până la spații comerciale și industriale. Lucrarea este structurată pentru a oferi o descriere detaliată a componentelor utilizate și a modului în care acestea interacționează. Prima secțiune va prezenta componentele hardware. Secțiunea următoare va detalia configurația și programarea acestora, explicând codul și funcționalitatea fiecărei părți a sistemului. Lucrarea se va încheia cu o analiză a rezultatelor obținute și discuții privind posibile îmbunătățiri și aplicații practice ale sistemului dezvoltat.

2. Abreviațiuni și simboluri

Listă abreviațiuni:

RFID(Radio-Frequency Identification): Tehnologie utilizată pentru a citi datele stocate pe o etichetă(tag) folosind unde radio.

DHT(Digital Humidity and Temperature Sensor): Senzor folosit pentru măsurarea temperaturii și umidității.

LCD(Liquid Crystal Display): Afișaj utilizat pentru a arăta informații textuale.

OLED(Organic Light Emitting Diode): Afișaj folosit pentru a arăta informații grafice și textuale.

MQ135: Senzor pentru monitorizarea calității aerului.

I2C(Inter-Integrated Circuit): Protocol de comunicare serială utilizat pentru conectarea componentelor integrate.

SPI(Serial Peripheral Interface): Protocol de comunicare serială pentru scurtă distanță.

Servo: Servomotor folosit pentru a efectua mișcări precise de rotație.

Listă simboluri și definiții:

RST_PIN: Pinul de resetare utilizat de cititorul RFID



SS_PIN: Pinul de selectare a slave-ului utilizat de RFID
DHTPIN: Pinul la care este conectat senzorul DHT11
DHTTYPE: Definirea tipului de senzor DHT(DHT11)
SCREEN_WIDTH: Lățimea ecranului OLED în pixeli
SCREEN_HEIGHT: Înălțimea ecranului OLED în pixeli
OLED_RESET: Pinul de resetare pentru ecranul OLED
SCREEN_ADDRESS: Adresa I2C a ecranului OLED
BlueLed: Pinul la care e conectat LED-ul albastru
GreenLed: Pinul la care e conectat LED-ul verde
RedLed: Pinul la care e conectat LED-ul roșu
Buzzer: Pinul la care e conectat buzzerul
MQ135PIN: Pin analogic la care este conectat senzorul de calitate a aerului
lastSensorReadTime: Variabilă pentru a stoca ultima citire a senzorului
sensorInterval: Intervalul de timp între citirile senzorului(în milisecunde)

3. Componente

Placa Arduino Uno – centrul de comandă al sistemului. Placă de dezvoltare microcontroller care permite conectarea și controlul mai multor senzori și componente.

Senzor RFID(MFRC22) – utilizat pentru a citi etichetele RFID și pentru a verifica dacă accesul este permis sau nu.

Senzor de Temperatură și Umiditate(DHT11) – folosit pentru a măsura temperatura și umiditatea aerului.

Senzor de gaz(MQ135) – monitorizează calitatea aerului.

Display OLED(Adafruit SSD1306) – este folosit pentru a afișa valorile măsurate de senzorii de temperatură și umiditate și calitatea aerului.

Display LCD(LiquidCrystal I2C) – utilizat pentru a furniza informații despre statusul accesului.

Servomotor – controlează mecanismul de deschidere a ușii.

LED-uri (Roșu, Verde, Albastru) – indică starea accesului.(roșu – închis, verde – deschis, albastru - standby)

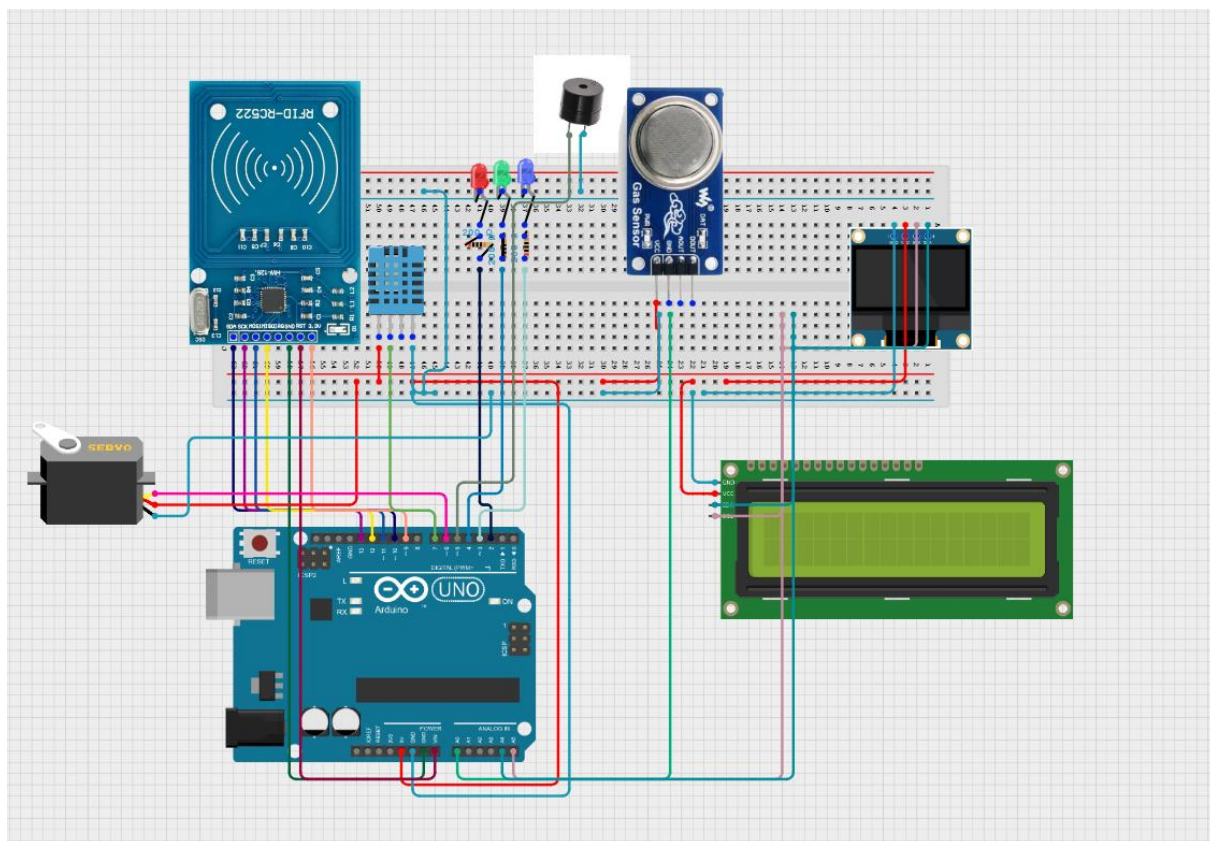
Buzzer – utilizat pentru a emite semnale sonore care indică accesul permis/respins.

Rezistor – pentru a proteja LED-urile de curent excesiv.



4. Descrierea realizării

Schema:





```

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

  // Initialize DHT sensor
  dht.begin();

  // Initialize OLED display
  if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
  display.display();
  delay(2000); // Pause for 2 seconds
  display.clearDisplay();

  // Initialize RFID reader
  SPI.begin();
  mfrc522.PCD_Init();

  // Initialize servo
  servo.attach(6);

  // Initialize LCD
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.print(" Access Control ");
  lcd.setCursor(0, 1);
  lcd.print("Scan Your Card>>");

  // Initialize LEDs and Buzzer
  pinMode(GreenLED, OUTPUT);
  pinMode(BlueLED, OUTPUT);
  pinMode(RedLED, OUTPUT);
  pinMode(Buzzer, OUTPUT);

  // Set Blue LED to HIGH and others to LOW for standby mode
  digitalWrite(BlueLED, HIGH);
  digitalWrite(GreenLED, LOW);
  digitalWrite(RedLED, LOW);
}

```

Funcția setup este apelată o singură dată pentru a inițializa toate componentele



sistemului. Se porneste senzorul DHT pentru a putea citi temperatura și umiditatea. Se configurează afișajul OLED, se inițializează RFID-ul, led-urile, buzzer-ul și LCD-ul.

```
void loop() {
    unsigned long currentMillis = millis(); // Get current time

    // Check if it's time to update the sensor readings (every 2 seconds)
    if (currentMillis - lastSensorReadTime >= sensorInterval) {
        lastSensorReadTime = currentMillis;

        // Read temperature and humidity from DHT11 sensor
        float temperature = dht.readTemperature(); // Celsius
        float humidity = dht.readHumidity(); // %

        // Check if the readings are valid
        if (isnan(temperature) || isnan(humidity)) {
            Serial.println(F("Failed to read from DHT sensor!"));
            temperature = 0;
            humidity = 0;
        }

        // Read air quality from MQ135 sensor
        int rawAirQuality = analogRead(MQ135Pin); // Raw MQ135 sensor value (0-1023)

        // Average multiple readings to smooth out noise
        int numReadings = 10;
        long totalAirQuality = 0;
        for (int i = 0; i < numReadings; i++) {
            totalAirQuality += analogRead(MQ135Pin);
            delay(50); // Short delay to give sensor time to stabilize
        }
        int averageAirQuality = totalAirQuality / numReadings; // Calculate average

        // Determine air quality status
        String airQualityStatus;
        if (averageAirQuality < 300) {
            airQualityStatus = "Poor";
        } else if (averageAirQuality >= 300 && averageAirQuality <= 700) {
            airQualityStatus = "Moderate";
        } else {
            airQualityStatus = "Good";
        }
    }
}
```



```

// Update the OLED display with temperature, humidity, air quality value, and category
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.print("Temp: ");
display.print(temperature);
display.println("C");

display.print("Humidity: ");
display.print(humidity);
display.println("%");

display.print("Air Quality: ");
display.print(averageAirQuality); // Display raw air quality value
display.setCursor(0, 24); // Move to next line for category
display.print("Category: ");
display.println(airQualityStatus); // Display air quality category (Poor/Moderate/Good)
display.display(); // Update the OLED screen
}

// Continuously check for RFID card presence
if (getUID()) {
  Serial.print("UID: ");
  Serial.println(UIDCard);
  lcd.clear();
  lcd.setCursor(2, 0);
  lcd.print("Permission");
  lcd.setCursor(0, 1);

  // Check if the detected UID matches the master tag
  if (UIDCard == MasterTag) {
    servo.attach(6); // Attach the servo
    lcd.print(" Access Granted!");
    digitalWrite(GreenLED, HIGH); // Green LED ON
    digitalWrite(BlueLED, LOW); // Blue LED OFF
    digitalWrite(RedLED, LOW); // Red LED OFF

    servo.write(70); // Open door
  }
}

```

Această funcție rulează continuu și gestionează citirea senzorilor, actualizarea afișajului OLED și verificarea prezenței cardului RFID.

La fiecare 2 secunde, se citesc datele de la senzorul de temperatură și umiditate (DHT11) și de la senzorul de calitate a aerului (MQ135).

Datele sunt afișate pe ecranul OLED, incluzând temperatura, umiditatea, valoarea calității aerului și categoria (Bună, Moderată, Slabă).

Dacă un card RFID este detectat, UID-ul acestuia este citit și comparat cu UID-ul principal (MasterTag). Dacă UID-ul se potrivește, accesul este acordat (ușa se



deschide, LED-ul verde se aprinde, buzzer-ul emite un sunet). Dacă UID-ul nu se potrivește, accesul este refuzat (LED-ul roșu se aprinde, buzzer-ul emite un sunet)

```
    for (int i = 0; i < 2; i++) {
        tone(Buzzer, 2000);
        delay(250);
        noTone(Buzzer);
        delay(250);
    }

    servo.detach(); // Detach the servo to save power and avoid jitter
} else {
    lcd.print(" Access Denied!");
    digitalWrite(BlueLED, LOW); // Blue LED OFF
    digitalWrite(GreenLED, LOW); // Green LED OFF

    tone(Buzzer, 1500);
    for (int i = 0; i < 6; i++) {
        digitalWrite(RedLED, HIGH); // Red LED ON
        delay(250);
        digitalWrite(RedLED, LOW); // Red LED OFF
        delay(250);
    }
    noTone(Buzzer);

}

// After processing, clear the LCD and go back to the waiting screen
lcd.clear();
lcd.print(" Access Control ");
lcd.setCursor(0, 1);
lcd.print("Scan Your Card>>");

// Set Blue LED to HIGH for standby mode again
digitalWrite(BlueLED, HIGH);
digitalWrite(GreenLED, LOW);
digitalWrite(RedLED, LOW);
}
```



```

boolean getUID() {
    if (!mfr522.PICC_IsNewCardPresent()) {
        return false;
    }

    if (!mfr522.PICC_ReadCardSerial()) {
        return false;
    }

    UIDCard = "";
    for (byte i = 0; i < mfr522.uid.size; i++) {
        UIDCard.concat(String(mfr522.uid.uidByte[i] < 0x10 ? " 0" : " "));
        UIDCard.concat(String(mfr522.uid.uidByte[i], HEX));
    }
    UIDCard.toUpperCase();
    UIDCard = UIDCard.substring(1);

    mfr522.PICC_HaltA();
    return true;
}

```

getUID() se ocupă de detectarea și citirea UID-ului cardului RFID.

Problemă nerezolvată: Multiple UID-uri.

Motivație:

1. Complexitatea administrării:
 - a. Într-un sistem cu mai mulți utilizatori actualizarea bazei de date poate deveni mai dificilă. Sunt necesare mecanisme suplimentare de adăugare/ștergere UID-uri.
2. Performanța sistemului:
 - a. Timpul necesar pentru verificarea UID-ului crește, având în vedere sistemul proiectat. Este necesară altă implementare pentru verificare.

5. Observații

1. Funcționare generală – proiectul funcționează conform așteptărilor în ceea ce privește verificarea tag-ului de acces; senzorii furnizează date precise.
2. Eficiență și feedback – buzzer-ul și led-urile oferă feedback vizual și auditiv.
3. Se poate îmbunătăți proiectul, folosind senzori mai preciși de temperatură, umiditate și calitate a aerului.



6. Concluzii

Concluzii Legate de Utilitatea Practică

1. Controlul Accesului:

- Proiectul oferă un sistem eficient și simplu de control al accesului bazat pe carduri RFID. Este ideal pentru utilizare în medii unde controlul accesului este esențial, cum ar fi clădirile de birouri, campusurile universitare, laboratoarele de cercetare, și alte spații restricționate.

2. Monitorizarea Mediului:

- Integrarea senzorilor de temperatură, umiditate și calitate a aerului adaugă valoare practică, permițând monitorizarea constantă a condițiilor de mediu în interiorul clădirilor.

3. Feedback Instantaneu:

- Afișajele OLED și LCD, împreună cu LED-urile și buzzer-ul, oferă feedback instantaneu utilizatorilor, îmbunătățind interacțiunea și experiența utilizatorului.

Aspecte Economice

1. Costuri Inițiale:

- Componenta principală a costului acestui proiect este hardware-ul necesar, incluzând senzorii (DHT11, MQ135), cititorul RFID, servo-ul, afișajele OLED și LCD, și microcontrolerul (de exemplu, un Arduino). Aceste componente sunt relativ accesibile și pot fi achiziționate la costuri moderate.

2. Costuri de Implementare și Întreținere:

- Instalarea inițială și configurarea sistemului necesită cunoștințe tehnice, ceea ce poate implica costuri suplimentare dacă sunt necesare servicii de instalare profesionale.
- Întreținerea sistemului poate include costuri pentru înlocuirea componentelor defecte, actualizarea software-ului, și gestionarea bazei de date a UID-urilor.



7. Bibliografie

<https://www.circuitstudio.com/> - pentru schema electrică

<https://www.create.arduino.cc/projecthub> - multiple idei de design și implementare

<https://github.com/curiores/ArduinoTutorials> - tutoriale pentru familiarizarea cu Arduino