Title of Minor Project

**Arduino RFID-Based Smart Lock**

**Date: 05/12/2023**

**Submitted by: Sakshi Kumari, Indubhusan Sahoo, Ragini Singh**

**Institution: Sri Sri University**

**Course/Department: B.sc. Comp Science,5th sem, FOS**

### ****Table of Contents****

1.Introduction

2.Project Objective

3.Materials and Tools

4.Circuit Diagram

5.Step-by-Step Instructions

6.Code Explanation

7.System Features

8.Non-functional Requirements

9.Other Requirements

10.Testing and Troubleshooting

11.Conclusion

12.References

### ****Introduction****

#### **Background:**

The integration of Radio-Frequency Identification (RFID) technology in security systems has become increasingly prevalent. This project leverages Arduino, a versatile microcontroller, to develop an RFID-based smart lock. The system allows for secure access control by identifying RFID tags and granting or denying access based on pre-defined permissions.

#### **Project Relevance:**

The significance of RFID-based smart locks lies in their potential applications for enhancing security in various settings, including homes, offices, and DIY projects. This documentation serves as a guide for individuals interested in understanding and implementing RFID technology for access control.

### ****Abstract****

### *****Overview:***** The RFID-Based Smart Lock project utilizes Arduino as the central component to create a secure access control system. The project employs RFID technology for user identification and access authorization. This documentation comprehensively outlines the steps, materials, and code involved in building this RFID-based smart lock, providing a valuable resource for both beginners and enthusiasts in the field of electronics and security systems.

### ****Project Objective****

#### **Purpose:**

The primary goal of this project is to design and implement a functional RFID-based smart lock system using Arduino, providing a practical application of RFID technology in the realm of access control.

#### **Scopes:**

* To familiarize users with RFID technology and its implementation.
* To demonstrate the integration of RFID readers with Arduino for data processing.
* To create a secure access control system using RFID cards and Arduino.
* To facilitate learning in basic electronics, programming, and security system design.

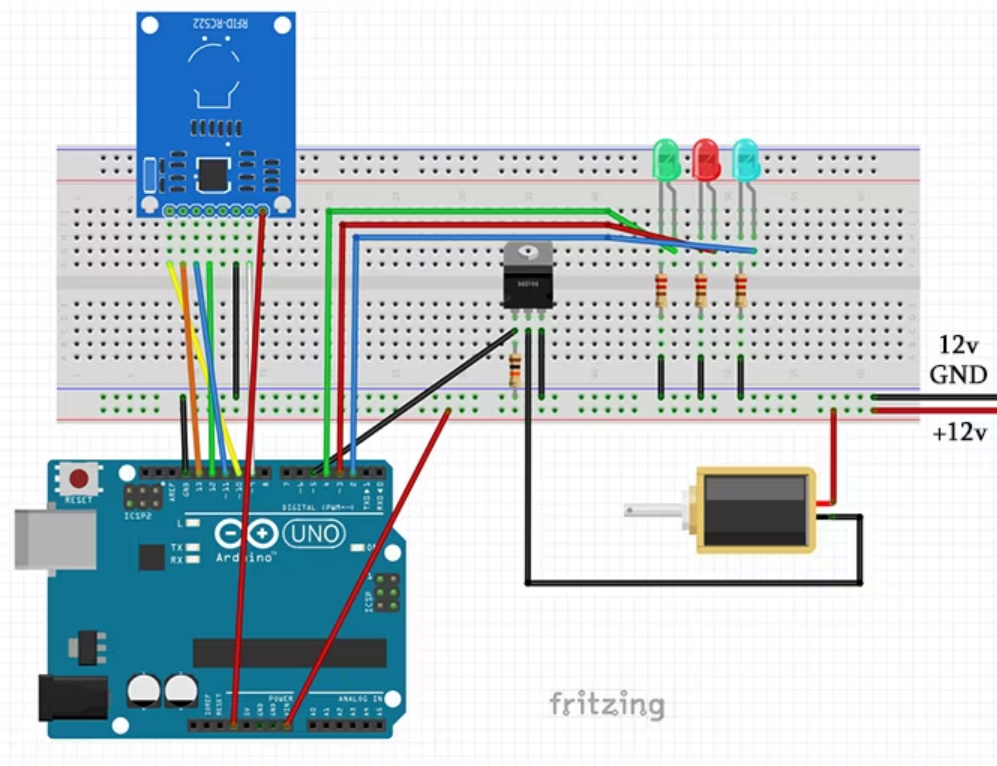
### ****Overall Description****

The RFID-based smart lock consists of an Arduino board, an MFRC522 RFID reader module, LEDs, resistors, a MOSFET, and a solenoid. The MFRC522 module reads RFID data, which is processed by the Arduino. The system controls access by activating the solenoid based on RFID tag authentication. LEDs provide visual feedback on system status.

### ****Materials and Tools****

* Arduino Board (e.g., Arduino Uno)
* MFRC522 RFID Reader Module
* LEDs (Red, Blue, Green)
* Resistors (220 ohm, 10k ohm)
* Logic-level N channel MOSFET
* 12V Solenoid
* 12V Power Supply
* Breadboard
* Jumper Wires

**Circuit Daigram**

****

### ****Step-by-Step Instructions****

1. **Setting Up the MFRC522 Module:**
   * Install the MFRC522 library in the Arduino IDE.
   * Connect the RFID reader to Arduino following the provided pinout.
   * Test the RFID reader using the DumpInfo sketch.
2. **Adding Components:**
   * Connect LEDs and resistors to pins 2, 3, and 4 for visual feedback.
   * Connect the MOSFET to pin 5 with a 10k ohm resistor to ground.
   * Connect the solenoid to the MOSFET and ground.
3. **Power Connections:**
   * Connect the 12V power supply to the breadboard.
   * Connect Arduino GND to ground rail and MFRC522 GND to the ground rail.
4. **Testing the Setup:**
   * Power up the circuit and ensure the blue LED indicates system operation.
   * Use the master card to enter admin mode and manage access rights.
   * Test access by presenting authorized RFID cards; observe LED feedback.

**Code Explanation**

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

#define red 3

#define green 4

#define blue 2

#define relay 5

int delayTime = 10000;

void setup() {

Serial.begin(9600);

SPI.begin();

mfrc522.PCD\_Init();

pinMode(red, green, blue, relay, OUTPUT);

}

void loop() {

if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial()) {

String uid = getUID();

uid == "MASTER\_UID" ? adminMode() : (authenticate(uid) ? accessGranted() : accessDenied());

}

}

bool authenticate(String uid) {

return true; // Custom authentication logic (always true for overview)

}

void accessGranted() {

digitalWrite(green, HIGH);

digitalWrite(relay, HIGH);

delay(delayTime);

}

void accessDenied() {

digitalWrite(red, HIGH);

delay(2000);

reset();

}

void adminMode() {

digitalWrite(blue, HIGH);

reset();

}

void reset() {

digitalWrite(red, LOW);

digitalWrite(green, LOW);

digitalWrite(blue, LOW);

digitalWrite(relay, LOW);

}

String getUID() {

String uid = "";

for (byte i = 0; i < mfrc522.uid.size; i++)

uid.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "") + String(mfrc522.uid.uidByte[i], HEX));

return uid;

}

#### **Explanation:**

* The code initializes the RFID reader, LED pins, and MOSFET pin.
* The **accessControl()** function manages card detection and calls other functions accordingly.
* The **authenticateCard()** function can be customized for specific access control logic.
* The **grantAccess()** and **denyAccess()** functions control LED feedback and lock activation.
* The **adminMode()** function indicates when the master card is detected, allowing access to admin features.

### ****System Features****

* **RFID Card Authentication:** The system authenticates RFID cards for access control.
* **LED Feedback:** Visual feedback through LEDs indicates system status and access outcomes.
* **Admin Mode:** Master card enables admin mode for managing access rights.
* **Customizable Access Logic:** The project allows customization of access control logic through code modifications.

### ****Non-functional Requirements****

* **Reliability:** The system should reliably authenticate and control access based on RFID cards.
* **Responsiveness:** LED feedback and lock activation should occur promptly upon RFID card detection.
* **Security:** The system should provide secure access control, minimizing unauthorized entry.

### ****Other Requirements****

* **Power Supply:** A stable 12V power supply is required for the solenoid and overall system operation.
* **Physical Structure:** The project can be housed in a protective enclosure to prevent damage and enhance aesthetics.
* **Master Card:** A designated master RFID card is needed to enter admin mode and manage access rights.

### ****Testing and Troubleshooting****

#### **Testing Steps:**

1. Power up the circuit and verify the blue LED indicates system operation.
2. Use the master card to enter admin mode and manage access rights.
3. Test access by presenting authorized RFID cards; observe LED feedback.
4. Ensure the lock stays open for the specified duration during authorized access.

#### **Common Issues and Troubleshooting:**

* **Issue:** RFID cards not detected.
  + **Solution:** Check wiring and ensure proper placement of RFID cards.
* **Issue:** LED feedback not matching expected outcomes.
  + **Solution:** Review code logic and LED pin assignments.
* **Issue:** Solenoid not triggering.
  + **Solution:** Check MOSFET wiring and ensure proper power supply for the solenoid.

### ****Conclusion****

The RFID-Based Smart Lock project demonstrates the integration of RFID technology with Arduino for secure access control. By providing a detailed guide, this project enables users to understand RFID functionality, Arduino programming, and basic security system design. The project's features, customization options, and non-functional requirements contribute to its educational value and potential for real-world applications.

**References:**

* [Ian Buckley's DIY Smart Lock Article](https://chat.openai.com/c/link-to-article)
* [MFRC522 Library Documentation](https://chat.openai.com/c/link-to-library-docs)