**Slide Title: ‘Wireless authentication of Smart Door Lock using RFID’**

**Slide Content**:

1.**Introduction** :

Presenting a Smart Door Lock System employing RFID technology for advanced security and convenience.

2.**Components Used** :

>Arduino Uno: Microcontroller for processing RFID data and controlling the lock mechanism.



>RFID Reader: Detects RFID tags/cards for user authentication.



>Relay: Facilitates the operation of the solenoid lock.



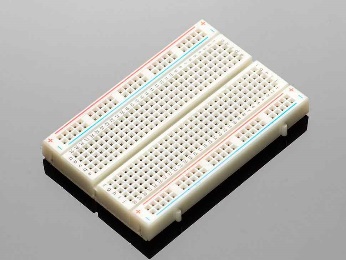
>12V Battery: Power source ensuring uninterrupted functionality.



>LEDs: Visual indicators for system status (e.g., locked, unlocked).



>Breadboard: Platform for prototyping and connecting electronic components.



>Solenoid: Actuator responsible for locking/unlocking the door.



>Jumping Wire: For connecting the equipments.



>Buzzer: For giving sound of signal.



3.**Functionality** :

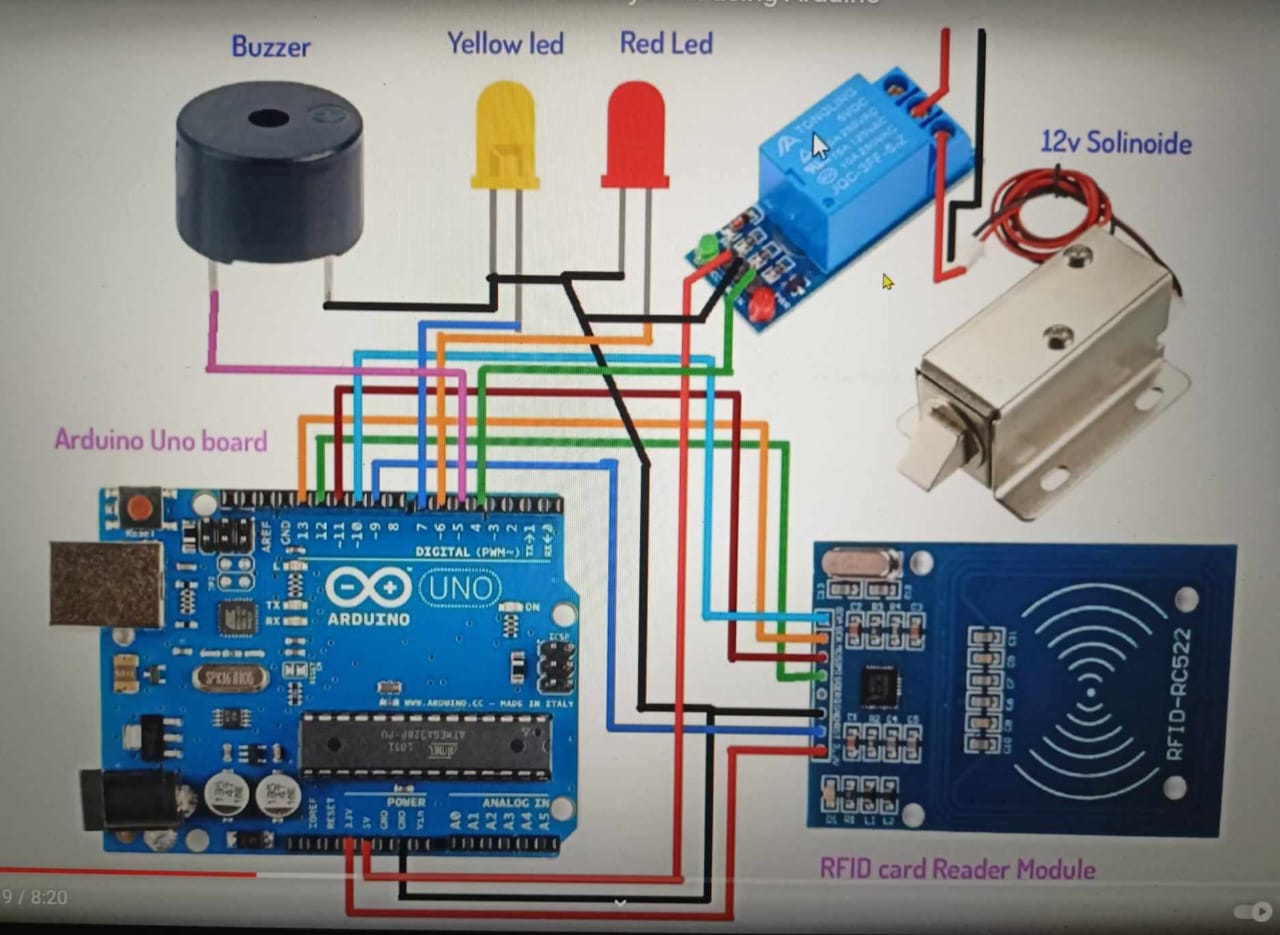
>RFID Authentication: Users present RFID tags/cards to the reader for access validation.

>Lock Control: Arduino triggers the relay to activate the solenoid lock upon successful authentication.

>LED Feedback: LEDs offer visual feedback on the lock status, indicating whether it's locked or unlocked.

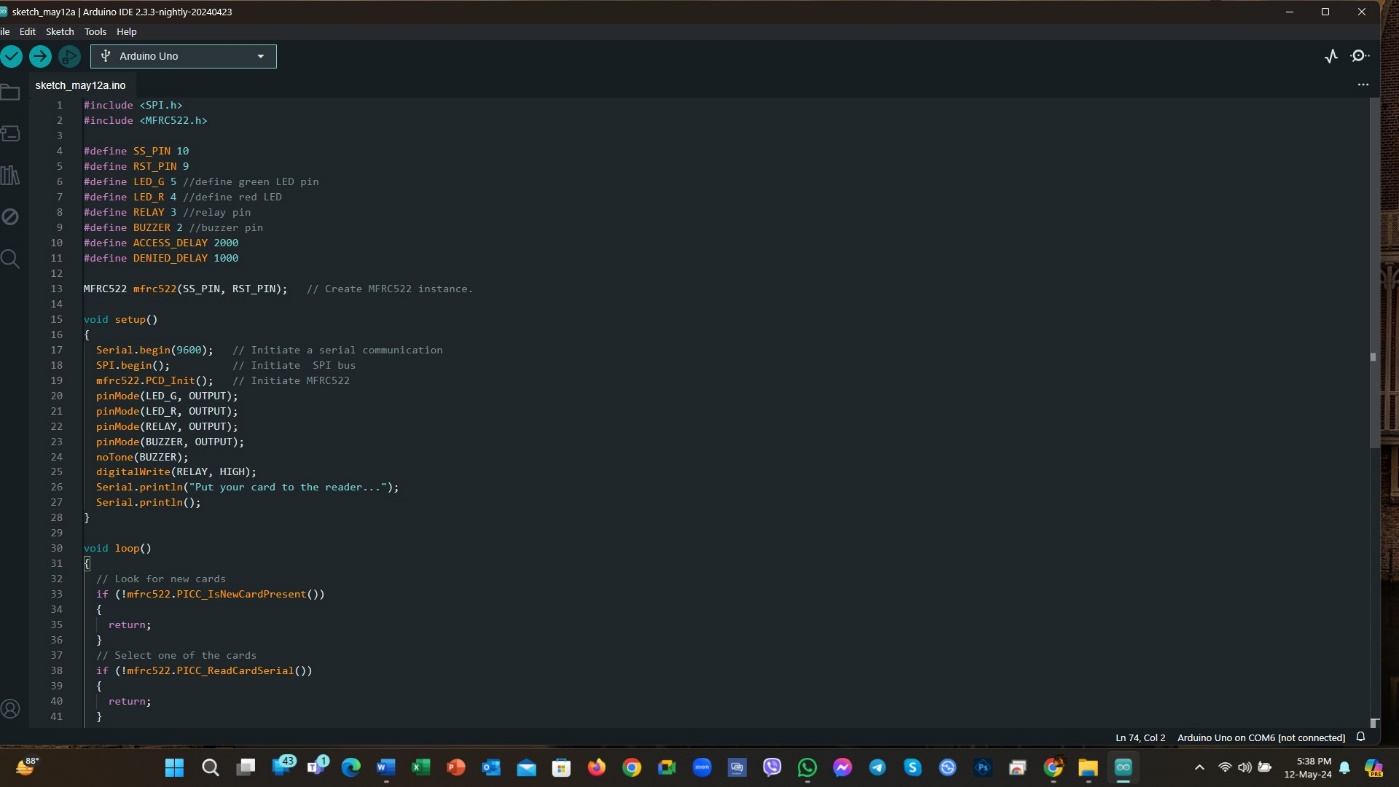
>Battery Power: System operates efficiently on a 12V battery, ensuring reliability during power outages.

4.**Circuit Diagram**:



5.**Software Requirements** :

> ARDUINO IDE SOFTWARE The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with various features like text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.



6.**Arduino Program**:

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

#define LED\_G 5 //define green LED pin

#define LED\_R 4 //define red LED

#define RELAY 3 //relay pin

#define BUZZER 2 //buzzer pin

#define ACCESS\_DELAY 2000

#define DENIED\_DELAY 1000

MFRC522 mfrc522(SS\_PIN, RST\_PIN);   // Create MFRC522 instance.

void setup()

{

  Serial.begin(9600);   // Initiate a serial communication

  SPI.begin();          // Initiate  SPI bus

  mfrc522.PCD\_Init();   // Initiate MFRC522

  pinMode(LED\_G, OUTPUT);

  pinMode(LED\_R, OUTPUT);

  pinMode(RELAY, OUTPUT);

  pinMode(BUZZER, OUTPUT);

  noTone(BUZZER);

  digitalWrite(RELAY, HIGH);

  Serial.println("Put your card to the reader...");

  Serial.println();

}

void loop()

{

  // Look for new cards

  if (!mfrc522.PICC\_IsNewCardPresent())

  {

    return;

  }

  // Select one of the cards

  if (!mfrc522.PICC\_ReadCardSerial())

  {

    return;

  }

  // Show UID on serial monitor

  Serial.print("UID tag :");

  String content = "";

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

  {

     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

     Serial.print(mfrc522.uid.uidByte[i], HEX);

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

     content.concat(String(mfrc522.uid.uidByte[i], HEX));

  }

  Serial.println();

  Serial.print("Message : ");

  content.toUpperCase();

  if (content.substring(1) == "43 38 08 15") //change here the UID of the card/cards that you want to give access

  {

    Serial.println("Authorized access");

    Serial.println();

    digitalWrite(RELAY, LOW);

    digitalWrite(LED\_G, HIGH);

    delay(ACCESS\_DELAY);

    digitalWrite(RELAY, HIGH);

    digitalWrite(LED\_G, LOW);

  }

  else

  {

    Serial.println(" Access denied");

    digitalWrite(LED\_R, HIGH);

    tone(BUZZER, 300);

    delay(DENIED\_DELAY);

    digitalWrite(LED\_R, LOW);

    noTone(BUZZER);

  }

}

7.**Assembly and Testing** :

When RFID tag placed on the RFID reader as it read the data and through reader its code send to the controller which access with the controller match and receives code with store code if the code is same then the security system is authorized to use and access the data. Change the tag ID in Access Control in to sketch with the ID you have noted down earlier and then connect arduino board with PC, upload the sketch into the board. After access control system the information is display on LCD and if the information is not correct the alarm will start ringing. If uploading is doing well, you will see the glowing of LED. It means the system is prepared to read the tag. Now, bring the tag near to RFID reader. If tag ID matches with the ID in the code, lock will open for five seconds. It closes manually after five seconds. Glowing of LED indicate that the lock is open. Glowing of caution LED means that you are using the wrong tag. The software for this project is written in Arduino indoctrination language. The Arduino UNO is program using Arduino IDE software. Atmega328 on Arduino UNO arrives with a boot loader that allows you to upload new code to it without the use of external hardware programmer. It used to communicate using STK500 protocol. You can also bypass the boot loader & program the microcontroller through in- circuit serial programming (ICSP) header, but with boot loader the program is fast & simple.

8.**Application** :

>Residential Security: Ideal for homes, providing a secure and convenient means of managing access.

>Office Access Control: Suitable for restricting access to authorized personnel in office environments.

9.**Conclusion** :

The methodology proposed in this chapter for evaluating security in commercial RFID systems has allowed for detecting relevant flaws in realworld developments, including the following Ability to clone animal identification information, Possibility of altering data of certain payment card, Extraction of private information from different transportation cards, Possibility of capturing tag-reader communications, Possibility of emulating both readers and tags. Most of the flaws detected were reported to the respective companies, and they have taken the proper measures to mitigate them: in some cases, the system was redesigned to increase security, but most companies had to replace the whole hardware with updated and more secure devices. The final conclusion is that although RFID systems can implement sophisticated security measures, certain developers have adopted the technology without taking such mechanisms into account. A methodology like the one proposed in this chapter can help to perform audits and determine the security level of an RFID system before taking it from a test environment to a real situation.

10.**Bills And Materials** :

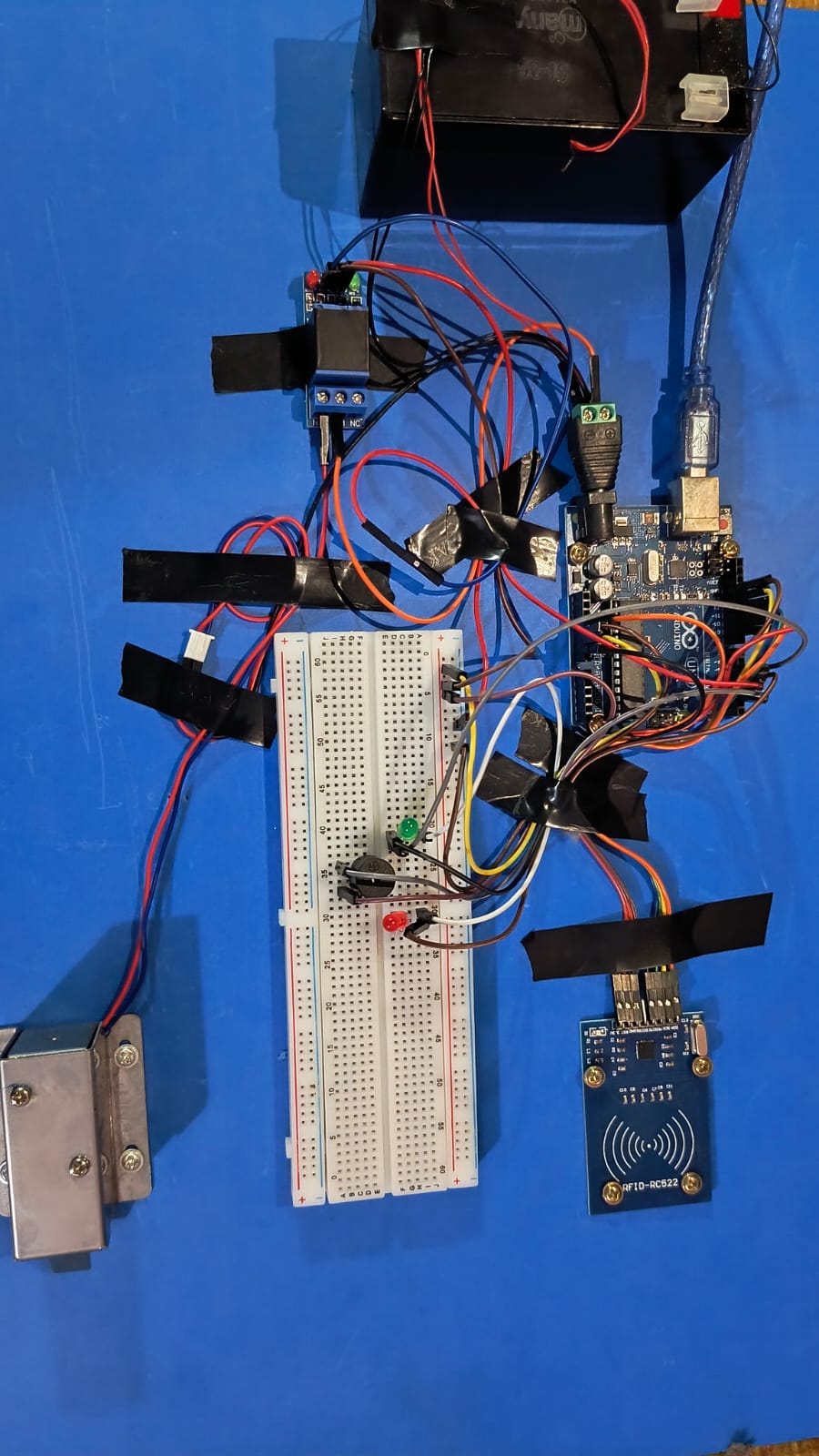
|  |  |
| --- | --- |
| **Components** | **Cost in Taka** |
| 1.Arduino Uno | 900 |
| 2.Relay | 300 |
| 3.Solinoid | 600 |
| 4.RFID Reader | 250 |
| 5.Jumping Wire | 100 |
| 6.12 V battery | 600 |
| 7.LED | 10 |
| 8.Buzzer | 20 |
| 9.Bread board | 120 |

Total Cost = 2900/=

11.**Future Scope** :

Tag detection not requiring human intervention reduces employment costs and eliminates human errors from data collection, As no line-of-sight is required, tag placement is less constrained, RFID tags have a longer read range than, e. g., barcodes, Tags can have read/write memory capability, while barcodes do not, An RFID tag can store large amounts of data additionally to a unique identifier, Unique item identification is easier to implement with RFID than with barcodes, Its ability to identify items individually rather than generically. Tags are less sensitive to adverse conditions (dust, chemicals, physical damage etc.), Many tags can be read simultaneously, RFID tags can be combined with sensors, Automatic reading at several places reduces time lags and inaccuracies in an inventory, Tags can locally store additional information; such distributed data storage may increase fault tolerance of the entire system, reduces inventory control and provisioning costs, Reduces warranty claim processing costs. RFID applications in the public sector. IT Asset Tracking Institutions with large IT assets with numerous data centers.

12**.Photoghraphy of the prototype** :



13.**Referance** :

• DhrubajyotiAdak, Manoj Kumar Pain, Uttam Kumar Dey” RFID Based Security System Using Arduino Module” International Journal of Scientific & Engineering Research, Volume 8, Issue 3, March-2017 ISSN 2229-5518

• Aditya A. Belhe,, Mukesh Mehta, Prince Raj Singh, MaqsoodChoudhary, SmitaChopade “RFID Based Secure Authentication ”International Journal of Advances in Electronics and Computer Science”,Volume2, Issue-12, Dec-2015 ISSN: 2393-2835

• Md. Abdul Aziz, Y.NaveenKumar, Ch.Pavan Kumar, P. Yaswanth Kumar “International Journal for Modern Trends in Science and Technology” Volume-2, Issue-4, April-2016 ISSN: 2455-3778

• Karan Khar , Aniket A. Kale , SupriyaRajankar“Arduino Based Door Access Control” Journal of Information, Knowledge and research in Electronics and Communication Engineering” Volume-4, Issue-2, Nov 2016- Oct 2017 ISSN: 0975-6779