RFID DOOR LOCK SYSTEM

by

SHIVA SHRI HARI AL U (20BEC1061)

PRANESH K (20BEC1102)

SATYAM SINGH (20BEC1072)

HARISH D (20BEC1034)

A project report submitted to

Dr. M. Jagannath

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECE3003 – Microcontroller and its applications

in

B.Tech. Electronics and Communication Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

VIT CHENNAI

Vandalur-Kelambakkam Road

Chennai – 600127

JUNE 2021

BONAFIDE CERTIFICATE

Certified that this project report entitled "**RFID DOOR LOCK SYSTEM"** is a bonafide work of **SHIVA SHRI HARI AL U (20BEC1061) PRANESH K (20BEC1102) SATYAM SINGH (20BEC1072) HARISH D (20BEC1034**) who carried out the project work under my supervision and guidance.

Dr. M. Jagannath

Associate Professor Senior

School of Electronics Engineering (SENSE),

VIT Chennai

Chennai-600127.

ABSTRACT

The major threat being faced by today's generation is insecurity over their belongings. It could be things, places, even data. So, in this project, we replicated a security system that is based on Radio Frequency Identification ID he/she will be allowed to access or go through if he/she has his/her ID registered in the system. We have also added two units the indoor and the outdoor unlike the generic systems even if someone tries to short the motor it won't be possible.

Radio Frequency Identification is a type of Near field communication that refers to a technology whereby digital data encoded in RFID tags or smart labels (defined below) are captured by a reader via radio waves. RFID is similar to barcoding in that data from a tag or label are captured by a device that stores the data in a database. RFID, however, has several advantages over systems that use barcode asset tracking software.

This project can be implemented through various microcontrollers. We demonstrated it using ARUDUINO UNO and used a RFID card for passkey. This project is done in ARDUINO UNO because in a normal microcontroller we cannot store the entry and exit of any RFID, but it can be stored in the cloud and accessed through an ethernet shield.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. M. Jagannath,** Associate Professor Senior, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. A. Sivasubramanian,** Dean of the School of Electronics Engineering, VIT Chennai, for extending the facilities of the school towards our project and for her unstinting support.

We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

HARISH D
(20BEC1034)

SATYAM SINGH
(20BEC1072)

SHIVA SHRI HARI AL U
(20BEC1061)

Pranesh K
(20BEC1102)

TABLE OF CONTENTS

1.INTRODUCTION

1.1 OBJECTIVES AND GOALS

- Design RFID based security system.
- Detect the RFID card using the RFID reader on the outside unit and encrypted data is transferred the data to an inside unit.
- Display whether the door is open for the scanned RFID and if the door is open for a long-time alert user.

1.2 BENEFITS

- Faster operation and efficiency.
- More secure due to 2 units
- Impossible to crack/open from outside by shorting the motor.

1.3 FEATURES

- RFID scanner scans digital codes assigned for each user and identifies them as key or not.
- The inside unit is used to check the passkey of the person along.
- The microcontroller used is ARDUINO UNO which is easy to program with Arduino IDE.

2. RELATED WORKS:

There are many related works. we can enlist them as follows:

1. Attendance based on RFID System with LCD Display

2. RFID Technology for authentication of passport details

3. RFID-based project for Library Automation System.

4. RFID Door lock and Alert System Using Arduino

3. RFID DOOR LOCK SYSTEM

3.1 BRIEF DESCRIPTION OF WORKING PRINCIPLE

The aim of this project is to design a Smart RFID-based door lock System using ARDUINO UNO, in which the when the card is scanned the door is open. The working of the project is explained here.

When this circuit is powered ON, the RFID reader/writer module detects the ID card, it will send the unique card number to the indoor unit which will be encrypted and sent to the indoor unit where it is decrypted.

With the help of suitable programming, we need to compare the decrypted received card number with the numbers that are already stored in the ARUDIUNO IDE.

Once, if any of these numbers are matched with the decrypted received card number, then the door is opened and if the door is open for a long time, then the buzzer is turned on.

3.2 BLOCK DIAGRAM

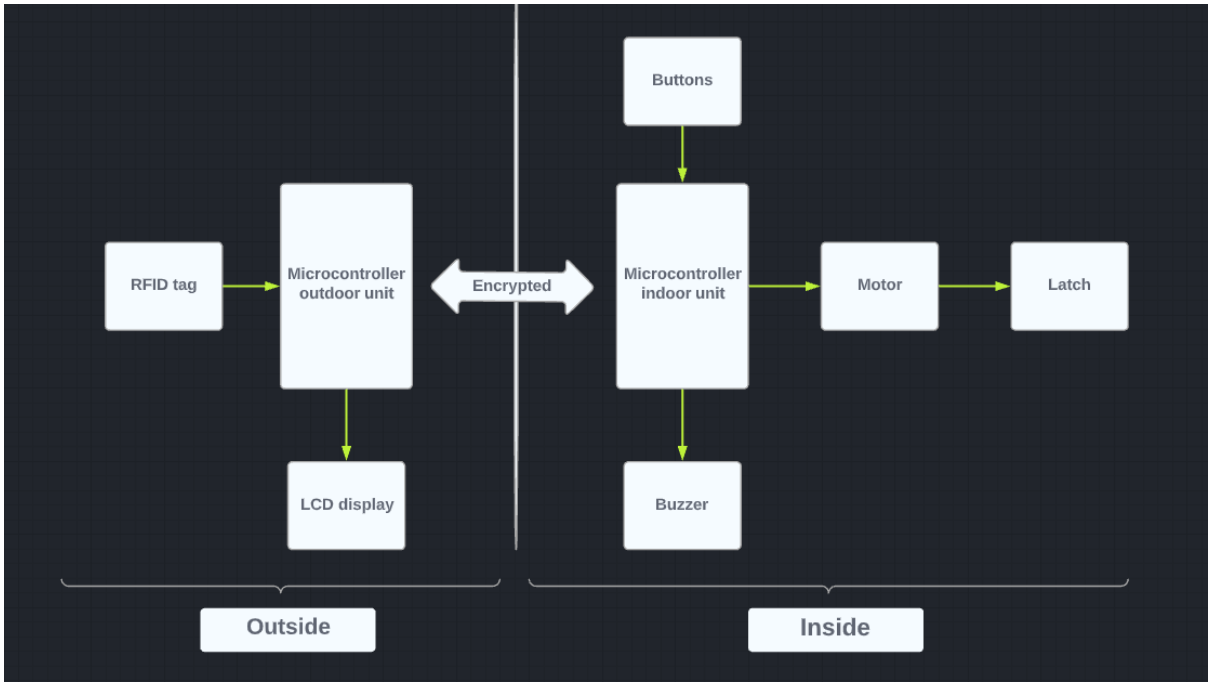

Figure 1: Block Diagram

3.3 REQUIRED EQUIPMENT

Hardware Specifications:

- Arduino uno board x 2.
- RFID Reader and Writer Module.
- RFID Tag faulty and correct.
- Jumper Cable.
- Power Cable.
- Bread board.

Software Specifications:

- Arduino IDE.
- Proteus
- Keil IDE.

3.4 HARDWARE ANALYSIS

The hardware consists of the 2 Arduino UNO boards, RFID reader/writer module, servo motor which represents the door latch, buzzer, and buttons. As the outdoor acts as the RFID reader and the encrypted data is sent to the indoor unit and the indoor unit decrypts the RFID tag data and compares with the key and if the key is matched the door is opened and if it is open for a long the buzzer goes on and goes off until the door closes.

3.4.1 Arduino Uno

Figure:2

The Arduino Uno (Figure:2) is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins,

and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. The word "UNO" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes pre-programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

3.4.2 RC522 RFID Reader

The RC522 RFID module based on the MFRC522 IC from NXP is one of the most inexpensive RFID options that you can get online for less than four dollars. It usually comes with an RFID card tag and key fob tag having 1KB memory. The RC522 RFID Reader module is designed to create a 13.56MHz electromagnetic field that it uses to communicate with the RFID tags (ISO 14443A standard tags). The reader can communicate with a microcontroller over a 4-pin Serial Peripheral Interface (SPI) with a maximum data rate of 10Mbps. It also supports communication over I2C and UART protocols. The operating voltage of the module is from 2.5 to 3.3V, but the good news is that the logic pins are 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter. [2]

3.5 SOFTWARE ANALYSIS

The microcontroller ATMEGA328 used in this project is programmed using Arduino IDE. Proteus is used to simulate the software.

3.5.1 CIRCUIT SOFTWARE REQUIREMENTS

- **Arduino.**
- **Proteus**

3.5.1.1 Arduino

Arduino is an open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the "Arduino language". In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command-line tool (arduino-cli) developed in Go. **[3]**

Key Features

- The **Project** window shows application source files and selected software components. Below the components, you will find corresponding library and configuration files.
- The operating voltage is 5V
- The recommended input voltage will range from 7v to 12V
- The input voltage ranges from 6v to 20V
- Digital input/output pins are 14
- Analog i/p pins are 6
- DC Current for each input/output pin is 40 mA
- DC Current for 3.3V Pin is 50 mA
- Flash Memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK Speed is 16 MHz

3.5.1.2 ASSEMBLY LEVEL CODING- SOFTWARE

OUTDOOR UNIT:

```
; P0.0 GOES TO RX INTERPT

; p2 is for LCD DATA

; P1.0 RS

; P1.1 R/W

; P1.2 E



ORG 0000H

LJMP MAIN

ORG 0023H

LJMP SERIAL

ORG 0003H

ACALL IX0

RETI



ORG 0030H

MAIN:   ;R0, R1, R7

SETB TCON.0

MOV IP, #00000001B

MOV TMOD, #21H

MOV TH1, #-3D ; 9600 baud rate
```

```
MOV SCON, #50H
SETB TR1
MOV TH0, #00H
MOV TL0, #00H
MOV P2, #00H  ; making p2 as output port
MOV P1, #0f0H
MOV A, #38H ;INTI
ACALL LCD_COMMAND
MOV A, #0EH
ACALL LCD_COMMAND
ACALL LCD_LOCK



READY:
MOV R0, #30H ; Temp address for storing the recived key
MOV R1, #05H
MOV R7, #77H ; ENCRY KEY
MOV IE, #10010001B
POLL:
CJNE R1, #00H, POLL
CLR IE.7 ; All bits are recived from RFID
MOV R0, #30H
MOV R1, #05H
SEND:
```

```
MOV A, @R0
XRL A, R7   ; Encrypting r7
MOV SBUF, A
CLR TI
HERE: JNB TI, HERE
INC R0
DJNZ R1, SEND
MOV R0, #30H
MOV R1, #05H
CLR TI
SETB IE.7
SJMP POLL

SERIAL:
MOV A, SBUF
MOV @R0, A
INC R0
DEC R1
CLR RI
cpl p1.7
RETI

IX0:
cpl p1.6
MOV A, #01H
```

```
ACALL LCD_COMMAND
MOV A, #02H
ACALL LCD_COMMAND
ACALL LCD_OPEN
IX0L:
JNB P3.2, IX0L
MOV A, R7
ADD A, #05H
MOV R7, A
MOV A, #01H
ACALL LCD_COMMAND
MOV A, #02H
ACALL LCD_COMMAND
ACALL LCD_LOCK
MOV R0, #30H
MOV R1, #05H
RET


LCD_COMMAND:
MOV P2, A
CLR P1.0 ;CLR RS
CLR P1.1 ;WRITE MODE
SETB P1.2
ACALL LCD_DELAY
```

```
CLR P1.2
ACALL LCD_DELAY
RET


LCD_DATA:
MOV P2, A
SETB P1.0 ;CLR RS
CLR P1.1 ;WRITE MODE
SETB P1.2
ACALL LCD_DELAY
CLR P1.2
ACALL LCD_DELAY
RET


LCD_LOCK:
MOV A, #'D'
ACALL LCD_DATA
MOV A, #'O'
ACALL LCD_DATA
MOV A, #'O'
ACALL LCD_DATA
MOV A, #'R'
ACALL LCD_DATA
```

```
MOV A, #' '
ACALL LCD_DATA
MOV A, #'L'
ACALL LCD_DATA
MOV A, #'O'
ACALL LCD_DATA
MOV A, #'C'
ACALL LCD_DATA
MOV A, #'K'
ACALL LCD_DATA
MOV A, #'E'
ACALL LCD_DATA
MOV A, #'D'
ACALL LCD_DATA
RET

LCD_OPEN:
MOV A, #'D'
ACALL LCD_DATA
MOV A, #'O'
ACALL LCD_DATA
MOV A, #'O'
ACALL LCD_DATA
MOV A, #'R'
```

```
ACALL LCD_DATA
MOV A, #' '
ACALL LCD_DATA
MOV A, #'O'
ACALL LCD_DATA
MOV A, #'P'
ACALL LCD_DATA
MOV A, #'E'
ACALL LCD_DATA
MOV A, #'N'
ACALL LCD_DATA
RET


LCD_DELAY:
MOV TH0, #0FDH
MOV TL0, #00H
SETB TR0
LLD: JNB TF0, LLD
CLR TR0
CLR TF0
RET


END
```

INDOOR UNIT:

```
ORG 0000H
LJMP MAIN
ORG 0013H
ACALL IX1
RETI
ORG 0023H
LJMP SERIAL


ORG 0030H
MAIN: ;r0, r1, r2, r3, r4, r7-encyp key
MOV IE, #10010100B
SETB TCON.0
MOV TMOD, #21H
MOV TH1, #-3D ; 9600 baud rate
MOV SCON, #50H
SETB TR1
MOV R0, #30H ; starting address of incoming byte - 1
MOV R2, #05H ; number of bytes
MOV IP, #00000100B ; External interpt 1 will have higher priority
MOV P1, #0FFH
MOV R1, #40H
CLR P2.0
```

```
;STORING THE VALUE OF THE KEY IN 20H
MOV 40H, #'Q' ; QWERT, 12345, abcde are passwords
MOV 41H, #'W'
MOV 42H, #'E'
MOV 43H, #'R'
MOV 44H, #'T'
MOV 45H, #'a'
MOV 46H, #'b'
MOV 47H, #'c'
MOV 48H, #'d'
MOV 49H, #'e'
MOV 4AH, #'1' ;pra is the password
MOV 4BH, #'2'
MOV 4CH, #'3'
MOV 4DH, #'4'
MOV 4EH, #'5'
MOV R7, #77H
LJMP HERE ; to bypass search program at start
;SEARCH PROGRAM
SEARCH:
MOV R0, #30H ; starting address of entered key
MOV R2, #05H
MOV R5, #40H
MOV R6, #03H
```

```
MOV R1, #40H
LSEARCH:
MOV A, @R0
XRL A, R7    ;decrypting the data
SUBB A, @R1
JNZ NEXT
INC R0
INC R1
DJNZ R2, LSEARCH
ACALL OPEN
SJMP EXIT
; add code to check against multiple key
NEXT:
MOV A, R5
ADD A, #05H
MOV R5, A
MOV R1, A
MOV R0, #30H
MOV R2, #05H
DJNZ R6, LSEARCH

EXIT:
MOV R0, #30H
MOV R1, #40H
MOV R2, #05H
```

```
; SEARCH PROGRAM ENDS
CLR P2.0
HERE:
JB P2.7, SKIP
ACALL OPEN
SKIP:
CJNE R2, #00H, HERE
SJMP SEARCH


SERIAL:
MOV A, SBUF
MOV @R0, A
INC R0
DEC R2
CLR RI
RETI


IX1:        ; Door is open
SETB TR0
JNB TF0, JUMP
CLR TR0
CLR TF0
INC R4
```

```
CJNE R4, #50D, JUMP
SETB P2.0
JUMP:
        JNB P3.3, IX1
CLR TR0
CLR P2.0
MOV R3, #01H   ;To cut short delay if called in the middle of delay
MOV TH0, #0FH
MOV TL0, #0FH
MOV R4, #00H
SETB TR0
RET


OPEN:
CLR IE.4
NOP
NOP
NOP
NOP
NOP
CLR P2.6 ; Pause u1
CLR P2.1
MOV R3, #35D ; Using r0 register to loop
REPEAT:
```

```
MOV TH0, #00H
MOV TL0, #00H
SETB TR0
LOOP: JNB TF0, LOOP
CLR TR0
CLR TF0
DJNZ R3, REPEAT
CLR P2.0
SETB P2.1
CLR TR0
CLR TF0
;change key
CLR IE.7
MOV A, R7
ADD A, #05H
MOV R7, A
MOV R4, #15D
LS: NOP
DJNZ R4, LS
MOV R4, #00H
SETB P2.6
SETB IE.7
SETB IE.4
RET
```

END

3.6 CIRCUIT BLOCK DIAGRAM

CIRCUIT DIAGRAM

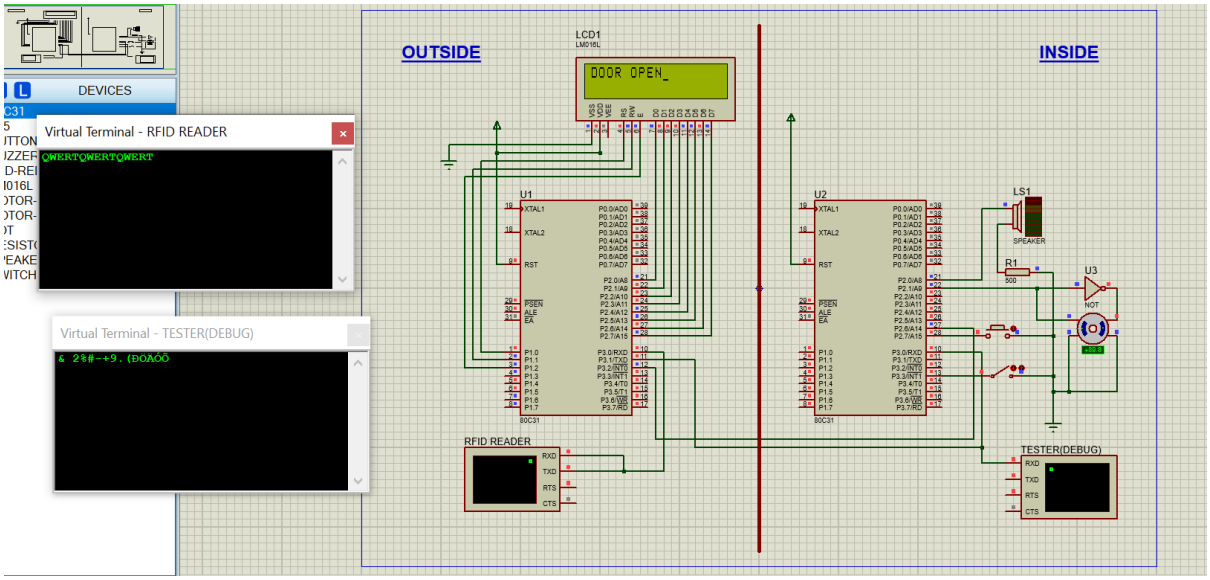

Figure 3: Circuit diagram

3.7 RFID DOOR-LOCK SYSTEM USING ARDUINO UNO

ALGORITHM

- Necessary header files are included at the beginning of the program.
- Define different functions such as one for sending data to the indoor unit, one for setting up the RFID scanner, one for checking the validity of the RFID tags, encrypting data, decrypting data are added.
- In the “loop” function we check whether the RFID numbers is given as input and send it the indoor unit where the decryption and checking takes place and the motor is controlled.

3.8 PROGRAM CODES

Arduino code:
INDOOR UNIT:

```
#include<Wire.h>
#include <Servo.h>
#define button A1
#define latch A0
#define buzzer 7

//byte key =
String MasterTag = "8D8C4CD3";
String MasterTag1 = "9CAE54A1";
String tagID = "";
byte cardUID[4];
byte key = 77;
Servo doorlock;
char c;
int i = 0;
bool door = false;


void setup() {
  Wire.begin(8);
  Wire.onReceive(receiveEvent);
  //Wire.onRequest(requestEvent);
  Serial.begin(9600);
  doorlock.attach(9);
  doorlock.write(0);
  pinMode(2, OUTPUT);
  digitalWrite(2, HIGH);
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
}


void loop() {
  // put your main code here, to run repeatedly:
  if(door)
    door_open();
  int a = analogRead(button);
```

```
  if(a > 900)
    door_open();
  delay(100);
}

void decrypt() {
  Serial.println("Decrypt");
  for(int i = 0; i<4;i++) {
    Serial.print(cardUID[i], HEX);
    cardUID[i] = cardUID[i] ^ key;
  }
  Serial.println();
  for(int i = 0; i<4; i++) {
    tagID.concat(String(cardUID[i], HEX));
  }
  tagID.toUpperCase();
  Serial.println(tagID);
  Serial.println("Decrypt - over");
  checkTag();
}

void receiveEvent(int a) {
  while (Wire.available()) {
    c = Wire.read();
    if(c != '\0') {
      Serial.print(c);
      cardUID[i] = (int)c;
      i++;
    } else if (c == '\0')
      decrypt();
    Serial.println();
  }
}

void door_open() {
  Serial.println(tagID.equals(MasterTag));
  digitalWrite(2, LOW);
  doorlock.write(180);
  delay(2500);
  int a = analogRead(latch);
  long int start_t = millis();
  while(a > 900) {
    a = analogRead(latch);
```

```
    if ((millis() - start_t) > 3000)
      digitalWrite(buzzer, HIGH);
  }
  digitalWrite(buzzer, LOW);
  doorlock.write(0);
  digitalWrite(2, HIGH);
  key += 5;
  door = false;
}

void checkTag() {
  if (tagID.equals(MasterTag) || tagID.equals(MasterTag1)) {
      door = true;
    }
    else {
      delay(4000);
    }
    tagID = "";
    i = 0;
}
```

OUTDOOR UNIT:

```
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>



#define RST_PIN 9
#define SS_PIN 10

byte key = 77;
byte readCard[4];
String tagID = "";

// Create instances
MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
  // Initiating
```

```
  SPI.begin(); // SPI bus
  mfrc522.PCD_Init(); // MFRC522
  Serial.begin(9600);
  Wire.begin();
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.print(" Access Control ");
  lcd.setCursor(0, 1);
  lcd.print("Scan Your Card>>");
  pinMode(2,INPUT);
}

void access_granted() {
  Serial.println("Granted");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Door opened");
  key += 5;
  return;
}

void encrypt() {
  for(int i = 0; i<4;i++) {
    Serial.print(readCard[i], HEX);
    readCard[i] = readCard[i] ^ key;
  }
  Serial.println();
  for(int i = 0; i<4; i++)
    Serial.print(readCard[i], HEX);
  Serial.println();
}

void door_reset() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Access Control ");
  lcd.setCursor(0, 1);
  lcd.print("Scan Your Card>>");
  return;
}
int b = 1;
void loop()
```

```
{
  while (getID())
  {
    Serial.println(tagID);
    sendTag();
  }
  int a = digitalRead(2);
  if(a == 0 && a != b)
    access_granted();
  else if (a == 1 && a != b)
    door_reset();
  b = a;
}

void sendTag() {
  encrypt();
  Wire.beginTransmission(8);
  for(int i = 0; i<4; i++)
    Wire.write((char)readCard[i]);
  Wire.write('\0');
  Wire.endTransmission();
}
//Read new tag if available
boolean getID()
{
  // Getting ready for Reading PICCs
  if ( ! mfrc522.PICC_IsNewCardPresent()) { //If a new PICC placed to RFID reader continue
  return false;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) { //Since a PICC placed get Serial and continue
  return false;
  }
  tagID = "";
  for ( uint8_t i = 0; i < 4; i++) { // The MIFARE PICCs that we use have 4 byte UID
  readCard[i] = mfrc522.uid.uidByte[i];
  tagID.concat(String(mfrc522.uid.uidByte[i], HEX)); // Adds the 4 bytes in a single String variable
  }
  tagID.toUpperCase();
  mfrc522.PICC_HaltA(); // Stop reading
```

return true;}

3.9 How RFID based DOOR LOCK system Works?

1. Arduino code is written as per the hardware and ports used.
2. Connections are given as per the circuit diagram.
3. Program is compiled and RFID tags are scanned.
4. Once scanned the ARDUINO outdoor unit encrypts the data and sends it to the indoor unit.
5. The indoor unit decrypts the data and compares with the keys stored and if keys match then the door is opened.
6. If the door is open for a long time the buzzer goes on and stops when the door is closed and another button is used to manually open the door from the inside
7. If the data doesn't match the latch doesn't move.

3.9.1 Circuit connection

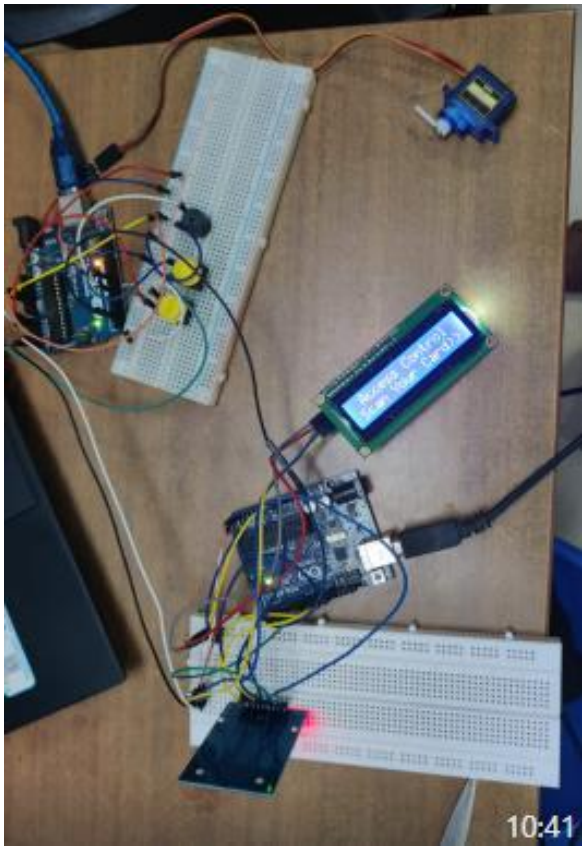

Figure 4

3.9.2 SCANNING THE VALID RFID TAGS

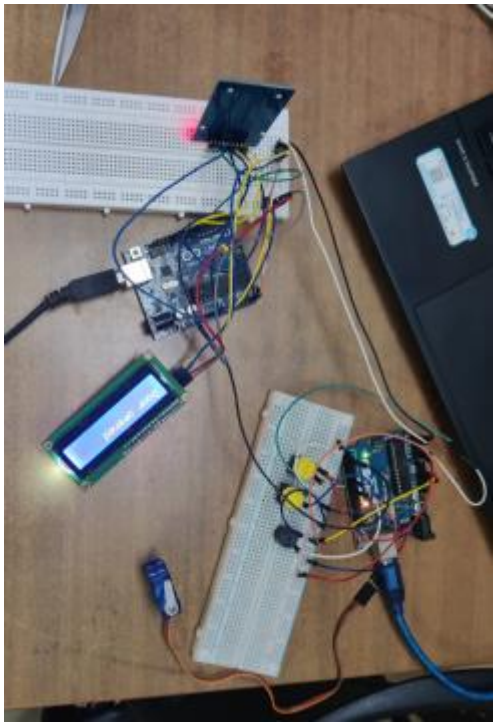

Figure 5

3.9.2 OPENING USING THE INDOOR UNIT BUTTON

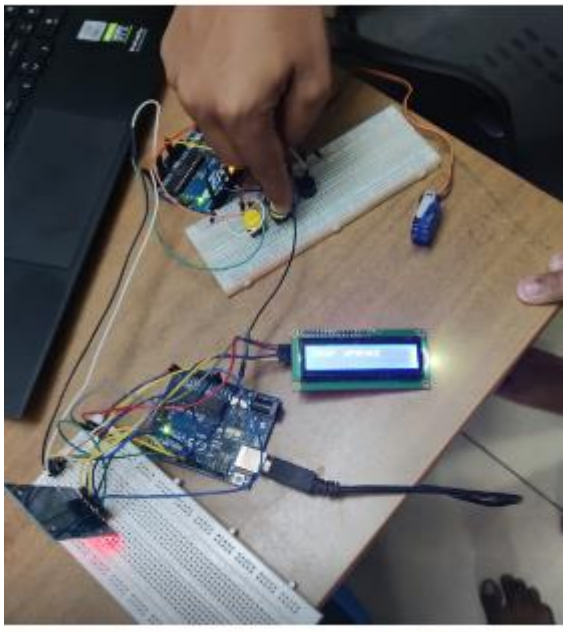

Figure 6

4. LIMITATIONS OF PROJECT CIRCUIT

- In this project the distance between the RFID scanner and the RFID tag is limited. As the technology works only for short-range.
- Materials like metal & liquid can impact signal.
- Since it is not as secure as biometric, the system is prone to manipulation.

5. APPLICATION AND ADVANTAGES:

- This system can be used for security and surveillance for students in school, college, university, offices and storage units. It also can be used to take attendance for workers in working places with further modifications.
- Its ability to uniquely identify each person based on their RFID tag type of ID card make the process of security and surveillance easier, faster, and secure as compared to the conventional method.
- The security is much higher and people without the proper id will not be able hack the door lock since the motor is controlled by the indoor unit and they need to know the encryption key to open predict and open the door which is practically impossible.
- No longer there is any need for papers and registers, and you can save papers and adapt to an environmentally friendly mode of living. **[8]**

6. CONCLUSION

In conclusion, the objective to build an RFID-based DOOR LOCK system was successfully achieved. In terms of performance and efficiency, this project has provided a convenient method of security compared to the usual method of door lock system. A lesser amount of manpower is required compared to the usual method. By using two units the security has been increase more compared the conventional electrical locks. All the future work is expected without spending extra cost, makes sure of safety and security too.

7. REFERENCES:

[1]https://www.etechnophiles.com/introduction-to-atmega328p-pinout-datasheet-specifications/

[2]https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/

[3]https://en.wikipedia.org/wiki/Arduino

[4]https://www.elprocus.com/atmega328-arduino-uno-board-working-and-its-applications/

[5] https://developers.google.com/apps-script/overview

[6] https://www.pushingbox.com

[7]https://docs.google.com/spreadsheets/d/1Wz1HMGF6LXie0TmkUwHbBpcQOxFyiq8m_q7B0ZF23t8/edit#gid=0

[8]https://www.electronicsforu.com/electronics-projects/rfid-based-attendance-system#