

Password-Based Door Locking System

A Project Report

By

Group - 2

Authors-

Name: Samihah Binte Sharif

ID: **C211250**

Section: 5AF

Email: c211250@ugrad.iiuc.ac.bd

Name: Sanjida Akter

ID: **C211251**

Section: 5AF

Email: c211251@ugrad.iiuc.ac.bd

Submitted To:

Mr. Sahriar Reza

Assistant Lecturer

Department of CSE, IIUC

A project report submitted in partial fulfillment of the requirements for the course of Electrical Drives & Instrumentation Lab



Department of Computer Science & Engineering

International Islamic University Chittagong

Abstract

Physical keys are the most-used and well-known technology to lock or unlock a door. People are familiar with it. Despite the physical key locking system being well-proven, it is not without liabilities. Roaming outside of the home with physical keys is inconvenient and there is a risk of losing physical keys. Smart-locking system is a keyless technology that lets a person unlock without having to use a physical key. A smart lock is an electromechanical lock that is designed to perform locking and unlocking operations on a door when it receives such instructions from an authorized device and executes the authorization process using a cryptographic key.

OBJECTIVE

The goal of the project is to find and assess a suitable set of components for creating a password-based smart door locking system with Arduino UNO that provides high security and quick access. The following are the specific goals:

- Familiarity with a smart door locking system based on a microcontroller (Arduino).
- Using the components to build a simple and smart door locking system.

PROJECT WORKING

At first, Arduino is initialized in standby mode where it asks the user to enter the password to unlock the door. In this Password Security Lock System Using Arduino project, we have defined the default password in Arduino as “3301”. However, it can be changed via coding later.

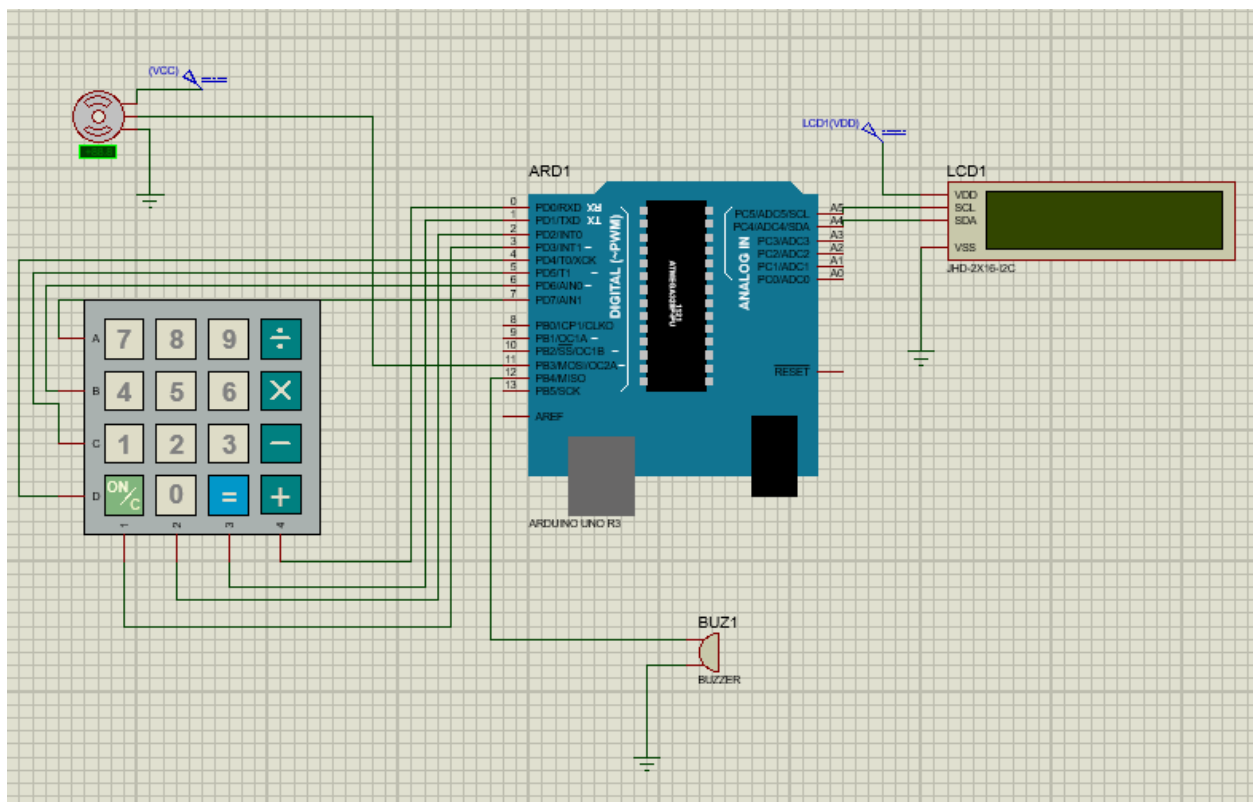
When users enter the “*” key it will allow the user to enter the password. When we enter a password, it will match it with the password stored in the Arduino. If it is correct, then it will show ‘Access Granted, Welcome’ and then rotate the servo motor to 180 degrees as the door is opened. It will then give 10 seconds time for entering via the door. After that the door will get locked automatically.

If the password is wrong, then it will show ‘Code Incorrect, Go Away’. However, if the password is incorrect 4 times in a row, the buzzer will beep to alert and no password can be entered for 30 seconds.

Similarly, in Arduino Password Security Lock System, if the user presses “#” it will ask for the code. Now, if the entered code is incorrect then it will get back to standby mode and if it is correct it will ask the user to enter the new password and confirm the password by retyping it. Here, the Arduino will compare the code; if it is incorrect then again it will get back to standby mode. But, if it is correct then it will change your current password.

IMPORTANT COMPONENTS

- Hardware
 1. Arduino UNO
 2. 4*4 Keypad
 3. SG90 Servo Motor
 4. 16*2 LCD
 5. Buzzer 5V
- Software
 1. Arduino IDE
 2. Proteus



CIRCUIT DIAGRAM ON PROTEUS

SOURCE CODE

```
#include <Keypad.h>
#include<Servo.h>
#include<Wire.h>
#include<EEPROM.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);

Servo myservo;

int pos=0; //LCD connections
int incorrectCount = 0;
int buzz = 12;

const byte rows=4;
const byte cols=4;
```

```
String scrollingMessage = "PRESS * TO OPEN DOOR, # TO CHANGE PASSWORD";
```

```
char key[rows][cols]={  
  {'1', '2', '3', 'A'},  
  {'4', '5', '6', 'B'},  
  {'7', '8', '9', 'C'},  
  {'*', '0', '#', 'D'}  
};
```

```
char keypressed;  
char code[]={ '3', '3', '0', '1' };  
char check1[sizeof(code)];  
char check2[sizeof(code)];
```

```
short a = 0, i = 0, s = 0, j = 0;
```

```
byte rowPins[rows]={7,6,5,4};  
byte colPins[cols]={3,2,1,0};
```

```
Keypad myKeypad = Keypad(makeKeymap(key),rowPins,colPins,rows,cols);
```

```
void setup() {  
  // put your setup code here, to run once:  
  lcd.init();           // initialize the lcd  
  lcd.backlight();  
  lcd.begin (16, 2);  
  lcd.setCursor(0, 0);  
  scrollMessage(0, scrollingMessage, 300, 16);
```

```
  pinMode(buzz, OUTPUT);  
  myservo.attach(11);  
  myservo.write(0);
```

```
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
  keypressed = myKeypad.getKey();           //Constantly waiting for a key to be pressed  
  if (keypressed == '*') {                  // * to open the lock  
    lcd.clear();  
  
    lcd.setCursor(0, 0);  
    lcd.print("ENTER CODE");
```

```

    lcd.setCursor(0, 1);
    lcd.print("THEN PRESS 'A'");
    delay(1000);
    lcd.clear();
    lcd.print("ENTER CODE");

    ReadCode();                                     //Getting code function

    if (a == sizeof(code))                         //The ReadCode function assign a value to a (it's correct
when it has the size of the code array)
        OpenDoor();                               //Open lock function if code is correct
    else {
        lcd.clear();
        lcd.setCursor(1, 0);
        lcd.print("CODE"); //Message to print when the code is wrong
        lcd.setCursor(6, 0);
        lcd.print("INCORRECT");
        lcd.setCursor(15, 1);
        lcd.print(" ");
        lcd.setCursor(4, 1);
        lcd.print("GET AWAY!!!");
        incorrectCount++;
        if(incorrectCount == 4){
            unauthorized();
            incorrectCount = 0;
        }
    }
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("ENTER THE CODE");
    lcd.setCursor(1, 1);

    lcd.print("TO OPEN DOOR");
    //Return to standby mode it's the message do display when waiting
}
if (keypressed == '#') {                          //To change the code it calls the changecode
function
    ChangeCode();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("ENTER CODE TO");
    lcd.setCursor(1, 1);

    lcd.print("OPEN DOOR");                        //When done it returns to standby mode
}
myservo.write(0);
}

```

```

void scrollMessage(int row, String message, int delayTime, int totalColumns){
    for(int i = 0; i<totalColumns; i++){
        message = " " + message;
    }
    message = message + " ";
    for (int position = 0; position < message.length(); position++) {
        lcd.setCursor(0, row);
        lcd.print(message.substring(position, position + totalColumns));
        delay(delayTime);
    }
}

void ReadCode() {
    //Getting code sequence
    i = 0; //All variables set to 0
    a = 0;
    j = 0;
    while (keypressed != 'A') { //The user press A to confirm
        the code otherwise he can keep typing
        keypressed = myKeypad.getKey();
        if (keypressed != NO_KEY && keypressed != 'A' ) { //If the char typed isn't A and
        neither "nothing"
            lcd.setCursor(j, 1); //This to write "*" on the LCD
            whenever a key is pressed it's position is controlled by j
            lcd.print("*");
            j++;
            if (keypressed == code[i] && i < sizeof(code)) { //if the char typed is correct a
            and i increments to verify the next caracter
                a++;
                i++;
            }
            else
                a--; //if the character typed is wrong
            a decrements and cannot equal the size of code []
        }
    }
    keypressed = NO_KEY;
}

void ChangeCode() {
    //Change code sequence
    lcd.clear();
    lcd.print("Changing code");
    delay(1000);
    lcd.clear();
    lcd.print("Enter old code");
    ReadCode(); //verify the old code first so you can change it
}

```



```

if (a == sizeof(code)) { //again verifying the a value
    lcd.clear();
    lcd.print("Changing code");
    GetNewCode1();           //Get the new code
    GetNewCode2();           //Get the new code again to confirm it
    s = 0;
    for (i = 0 ; i < sizeof(code) ; i++) { //Compare codes in array 1 and array 2 from two
previous functions
        if (check1[i] == check2[i])
            s++; //again this how we verify, increment s whenever
codes are matching
    }
    if (s == sizeof(code)) { //Correct is always the size of the array

        for (i = 0 ; i < sizeof(code) ; i++) {
            code[i] = check2[i]; //the code array now receives the new code
            EEPROM.put(i, code[i]); //And stores it in the EEPROM

        }
        lcd.clear();
        lcd.print("Code Changed");
        delay(2000);
    }
    else { //In case the new codes aren't matching
        lcd.clear();
        lcd.print("Codes are not");
        lcd.setCursor(0, 1);
        lcd.print("matching !!");
        delay(2000);
    }

}
else { //In case the old code is wrong you can't change it
    lcd.clear();
    lcd.print("Wrong");
    delay(2000);
}
}

void GetNewCode1() {
    i = 0;
    j = 0;
    lcd.clear();
    lcd.print("Enter new code"); //tell the user to enter the new code and press A
    lcd.setCursor(0, 1);
    lcd.print("and press A");
    delay(2000);
    lcd.clear();
}

```

```

lcd.setCursor(0, 1);
lcd.print("and press A");    //Press A keep showing while the top row print ***

while (keypressed != 'A') {    //A to confirm and quits the loop
    keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY && keypressed != 'A' ) {
        lcd.setCursor(j, 0);
        lcd.print("*");    //On the new code you can show * as I did or change it to
keypressed to show the keys
        check1[i] = keypressed;    //Store caracters in the array
        i++;
        j++;
    }
}
keypressed = NO_KEY;
}

void GetNewCode2() {    //This is exactly like the GetNewCode1 function
but this time the code is stored in another array
    i = 0;
    j = 0;

    lcd.clear();
    lcd.print("Confirm code");
    lcd.setCursor(0, 1);
    lcd.print("and press A");
    delay(3000);
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("and press A");

    while (keypressed != 'A') {
        keypressed = myKeypad.getKey();
        if (keypressed != NO_KEY && keypressed != 'A' ) {
            lcd.setCursor(j, 0);
            lcd.print("*");
            check2[i] = keypressed;
            i++;
            j++;
        }
    }
    keypressed = NO_KEY;
}

void OpenDoor() {    //Lock opening function open for 3s
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Access Granted");

```

```

    lcd.setCursor(4, 1);
    lcd.print("WELCOME!!");
    myservo.write(180);
}

void unauthorized(){
    digitalWrite(buzz, HIGH);
    delay(540);
    digitalWrite(buzz, LOW);
    delay(540);
    digitalWrite(buzz, HIGH);
    delay(540);
    digitalWrite(buzz, LOW);
    delay(540);
    digitalWrite(buzz, HIGH);
    delay(540);
    digitalWrite(buzz, LOW);
    delay(540);
    digitalWrite(buzz, HIGH);
    delay(540);
    digitalWrite(buzz, LOW);
    delay(540);
    digitalWrite(buzz, HIGH);
    delay(540);
    digitalWrite(buzz, LOW);
    delay(540);
    digitalWrite(buzz, HIGH);
    delay(540);
    digitalWrite(buzz, LOW);
    delay(540);
    digitalWrite(buzz, HIGH);
    delay(540);
    digitalWrite(buzz, LOW);
    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("WAIT FOR");

    for(int p = 30; p>0; p--){

        lcd.setCursor(2, 1);
        lcd.print(p);

        if(p<10){
            lcd.setCursor(3, 1);
            lcd.print(" ");
        }
        lcd.setCursor(5, 1);
        lcd.print("SECS");
    }
}

```

```
    delay(1000);  
  }  
}
```

CONCLUSION

As a result, the "Smart Door Locking System utilizing Arduino" is a modern take on the traditional door lock. The innovation generated by the lock system with no more direct touch between the user and the lock is the end of the topic of smart Lock utilizing Arduino. This method is both inexpensive and simple to set up. Finally, it was revealed that the project performed as expected and that it can be implemented. Because the Arduino UNO microcontroller is used in this project, the design is simple, and the project may be completed in less time than with other techniques. A safe locking/unlocking system based on a keypad and Arduino is proposed in this paper. Adding a password on the Arduino side improves the security of the system.

REFERENCES

- [1] <https://how2electronics.com/password-based-security-system-arduino-keypad/>
- [2] <https://iotprojectsideas.com/password-security-lock-system-using-arduino-keypad/>
- [3] [https://www.researchgate.net/publication/361265944_Password-Based Smart Door Locking System](https://www.researchgate.net/publication/361265944_Password-Based_Smart_Door_Locking_System)