1. INTRODUCTION

1.1 PROJECT DESCRIPTION

A. Background of Study and Motivation

In our daily lives, safety is a major concern. Every person requires a sense of safety. Our security pattern includes an access control system for doors. Traditional locks are no longer as secure as they once were; anyone can gain access by breaking these locks. We need to create a system that will assist 24 hours a day, seven days a week. Only authorized individuals have access to restricted areas thanks to a password-based door lock system. Arduino is in charge of the entire system. A keypad can be used to enter the password. The door opens if the password matches the password entered in Arduino. This password-based bolt structure will provide clients with a more secure and low-effort locking-opening mechanism. Mechanical door locks will be replaced by electronic door locks in the future, thanks to the security door lock automation system.

B. Project Objectives

The goal of this project is to research and analyze a suitable collection of components for developing a smart door lock using Arduino that provides excellent security and quick access. The following are the specific project goals: • Familiarity with a smart door locking system based on a microcontroller. • Using Arduino to create a simple and smart door locking system.

C. A brief outline of the report

This project is divided into 5 chapters. Chapter 1 present the background of study and motivation of this project. Chapter one also presents objectives and a brief outline. Chapter 2 provides the literature review of this project. Chapter 3 introduces the project methodology and modeling like working principle, process of work, component, implementation, testing and cost analysis. Chapter 4 presents the results and discussions of this project, also the simulation and experimental results. Finally, Chapter 5 Conclude the project.

2. LITERATURE REVIEW

In this current system of the project for door locking, there are a few points of view. A traditional door and key locking system that integrates the latest smart locking technology. Modern existence is widely reliant on technical improvements such as the ability to open doors and control modern gadgets and devices. People want their appliances to make them feel comfortable and protected. People's requirements are the key reason for the creation of smart locks. Some of these systems are discussed in this section.

● Internet of Things

Smart buildings have recently become the foundation for the Internet of Things (IoT). Connecting devices in homes to the internet increases internet usage, making homes more comfortable, provident, pleasant, and secure. The proposed approach focuses on a key security feature of smart home technologies: the door lock mechanism. The door lock system sets up security by allowing the owner to check the buildings using an Arduino UNO-controlled, Bluetooth-connected system. Installing the built android application on devices such as tablets, smartphones, laptops, and other computers allows users to open or close the door lock by entering login credentials such as username and password, which are confirmed in a database over the internet. If the credentials are incorrect, the buzzer sounds and an SMS notice is issued to the building's owner, increasing security. Using other wireless connectivity, this concept can be extended out to commercial sectors such as ATMs, vending machines, and so on.

● Fingerprint Locking System

Only individuals whose fingerprints are pre-stored in the memory can use fingerprint recognition technology. Even in case of a complete power outage or battery loss, stored fingerprints are kept. This eliminates the need to keep track of keys or memorize a password or PIN combination. Because there are no keys or combinations that can be copied or stolen, or locks that can be picked, it can only be accessed when an authorized person is present. As a result, the fingerprint-based lock is an excellent solution to commonly encountered problems.

● Knock-Pattern Using Arduino and GSM Communication

This technique uses a 'Secret Knocking Pattern,' which is only known by the owner of the safe, luggage, or other object or item on which the device is installed. The knocking pattern must be applied only at a certain spot known only to the owner for the lock to open. Only when the secret knock has been unlocked can the secret pattern be changed. Because there is no key to copy, this method completely removes the possibility of duplication.

● Keyless Entry System Based on Arduino Board with Wi-Fi Technology

An Arduino circuit board, a Wi-Fi module, and the PHP programming language are used to allow access to a closed door in this keyless entry system. The proposed method is explained, which involves using an Arduino Uno board and a Wi-Fi shield to unlock the door without a key. Unlike earlier systems, which have a restricted range, the internet connection allows the device to unlock the door from anywhere.

● RFID Based Access Control System

In the recommended system, a magnetic door lock is controlled by an RFID reader, which starts the authentication and validation of the user or, in other words, regulates access. Furthermore, the systems keep track of each user's access and exit records via a log report for each access. To avoid unforeseen occurrences, the administrator of the central subsystem can revoke any user's validity at any time.

3. METHODOLOGY & MODELING

A. Introduction

In this project, we used an Arduino and a keypad to create a password-based security system. Thefts and frauds are becoming more common by the day, therefore security is becoming a serious worry. As a result, a smart lock with a digital code can simply secure our home, business, locker, and other valuables. It only unlocks a door when the correct password is supplied. Due to a password-based door lock mechanism, only authorized personnel are permitted access to the restricted areas. The Arduino is in charge of the project's overall operation. A 4×4 keypad can be used to input the necessary password**.**

B. Working Principle

There are two cases for this experiment. The purpose of this experiment is to implement a door-locking mechanism that opens or closes the lock on the door automatically with password.

Case 1: The lock will open and close

When a password is entered via keypad, the system checks the password and finds out if it is right or wrong. If the password matches with the stored password in the microcontroller chip, the microcontroller sends the signal to the LCD display for showing “The door is open” as well as the microcontroller sends the signal to Servo Motor. Then the motor is rotated by 0° to 180° and opens the lock, allowing the door to be unlocked. Later, by pressing ‘#’ button, the Servo motor is rotated again by 180° to 0° closing the door.

Case 2: The lock will not open

If the wrong password is entered, the system shows “Password doesn’t match – Please try again” and Servo Motor is not rotated. A bit of time the system automatically starts again from the beginning**.**

C. Important Components

- Hardware:
  1. Arduino UNO
  2. 4×4 Matrix Keypad
  3. SG90 Micro Servo Motor
  4. 16×2 LCD Display
  5. Potentiometer
- Software:
  1. Arduino IDE
  2. WOKWI Web Application

SOFTWARE REQUIREMENTS SPECIFICATION

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as computer system requirements and are often used as a guide line as opposed absoluted rule. Most software defines two sets of system requirements: minimum that can defineded recommended, with increasing demand for higher power and resources in newer versions of software ,system requirements tend to increase over time. Industry analysts suggest the trend plays a bigger part in upgrades to existing computer system than technological.

TOOLS AND TECHNNOLOGIES USED:

Arduino UNO:

Arduino UNO is an open source microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

The Arduino platform has become quite popular with people just starting with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

4×3 Matrix Keypad:

This 4x3 matrix keypad has 12 built-in pushbutton contacts connected to row and column lines. A microcontroller can scan these lines for a button-pressed state. In the keypad library, the Propeller sets all the column lines to input and all the row lines to input. Then, it picks a row and sets it high. After that, it checks the column lines one at a time. If the column connection stays low, the button on the row has not been pressed. If it goes high, the microcontroller knows which row (the one it set high), and which column, (the one that was detected high when checked.

The keypad library supports pretty much any number of rows and columns. So, the program has to tell us that our keypad has 4 rows and 4 columns, which I/O pins the lines are connected to, and what value each button represents. The rows, cols, and values arrays store that information. The rows array will be used to tell the keypad library that the top row is connected to P7, the second row to P6 and so on. Likewise, the cols array lists the leftmost column as connected to P3, the next over connected to P2 and so on.

The values array stores the value we want the program to give us for each button press. For example, if the top-left button is pressed, we want the number 1, and if the next one is pressed, we want the number two. If the top right button is pressed, we want the ASCII code for the 'A' character, which is 65

SG90 Micro Servo Motor:

Micro Servo Motor SG90 is a tiny and lightweight servo motor with high output power. Servo can rotate approximately 90 degrees, and works just

like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with 3 horns (arms) and hardware.

16×2 LCD Display:

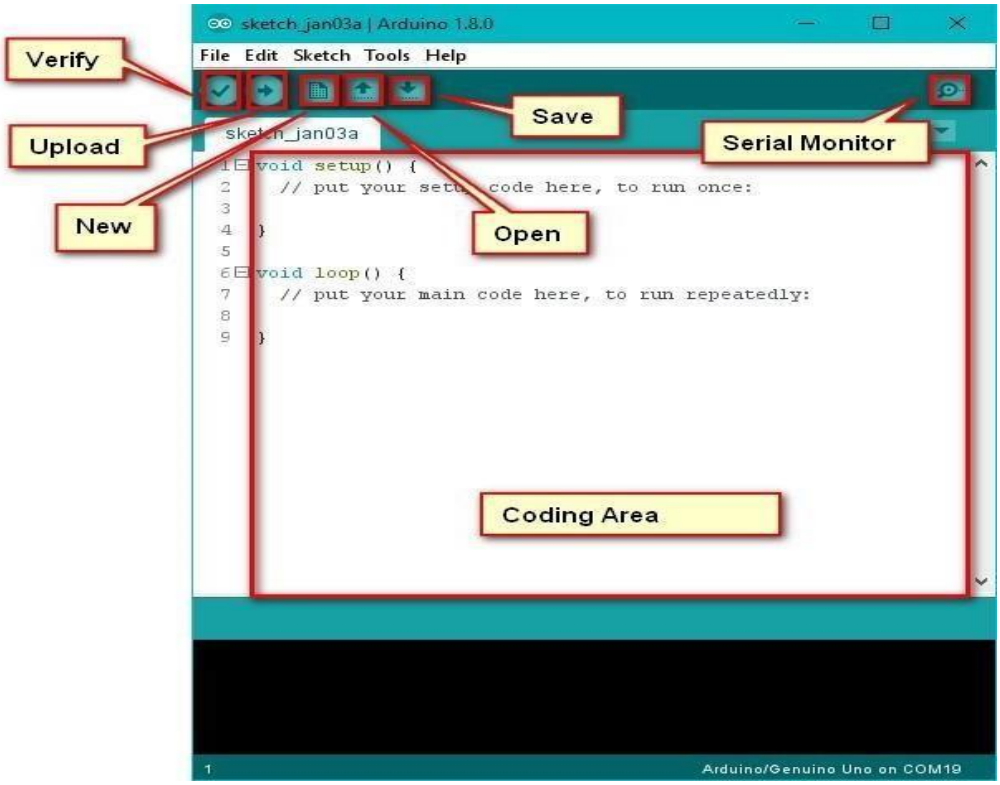
The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

Jumper Wires:

A jumper wire is an electrical wire, or group of them in a cable, with a connector or pin at each end which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components.

We are also required to use the jumper wires so that the internal connection between have the different components of the Arduino could be established.

Arduino IDE:



The open – source Arduino Software (IDE) makes it easy to write code and upload to the board. This software can be used with any Arduino board. Programs written using with a Arduino software (IDE) are called sketches.

These Sketches are written in the text editors and are saved with the file extension .ino the editor has features for cutting/pasting and for searching/replacing text.The message locate gives feedback while saving and exporting and also displays errors.The console displayed text output by the Arduino software (IDE), including complete error messages and others information.

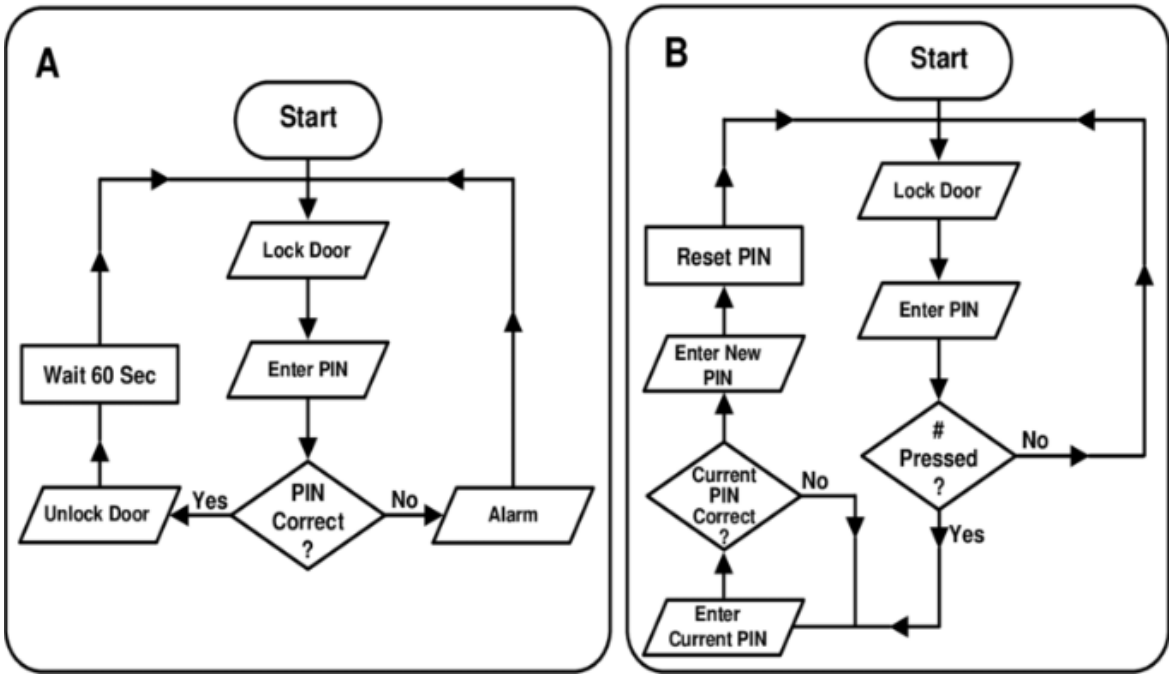
The bottom righthand corner of the window displays the configured board and serial ports the toolbar buttons allow you to verify and upload programs, create, open, and save many sketches, and open the serial monitor.

4. Analysis Design:

4.1 System Design

In this project we are combining two things together. Hardware and software, as we are coding in the microcontroller and the software we need to be aware of the developing process of the application and the hardware. As we have used SDLC method demonstrate the developing steps of the hardware and the software. So, for system design, we have use certain criteria that could combine both software and hardware in an orderly manner. In a chapter we will describe the tools and skill required to develop system with combinations of software and hardware. To illustrate our system design we have used various diagrams.

5.1.1 Data flow diagram

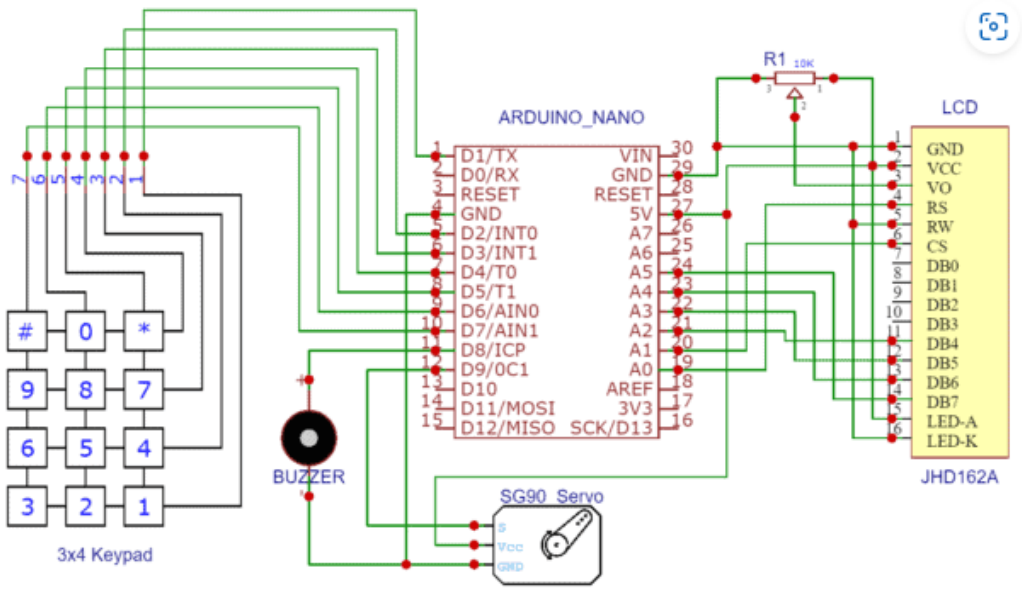


Flowchart of proposed IOT System

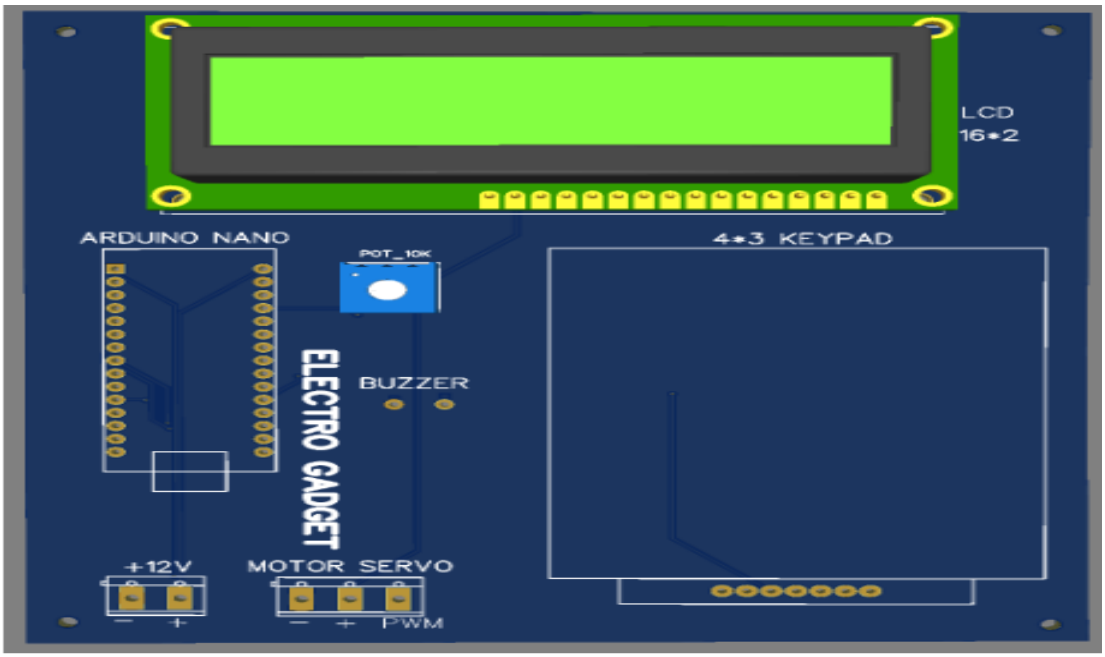
The Data flow diagram is a kind of diagram explains the flow of the application that is to be developed. In other words, all the information that is processed is through flow for all diagram. It represents certain kind of symbols that shows activity that is performed at the each step and it justify all the steps accordingly that are followed in a certain sequence of one after the other.

4.1.2 Circuit Diagram

Circuit diagram are the illustrative diagram to show the circuits that help users understand which element is connected to where and how. It is a graphical representation of wires, as cables and board together. As our project is based on the circuit, where all the elements of the project are interconnected. Following circuit design shows the circuit diagram, where device is connected to the circuit board, which is further connected to the utilities.

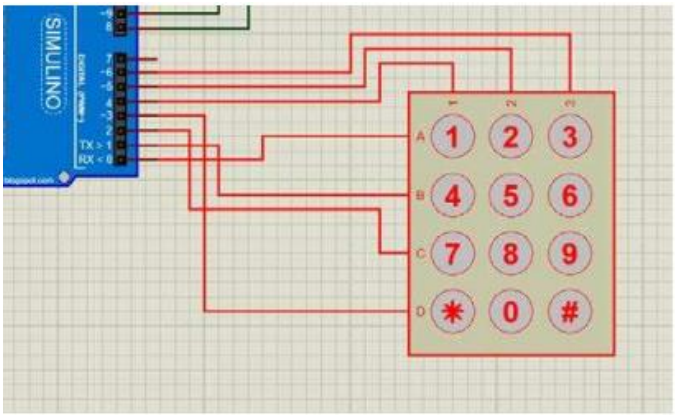


DESIGN

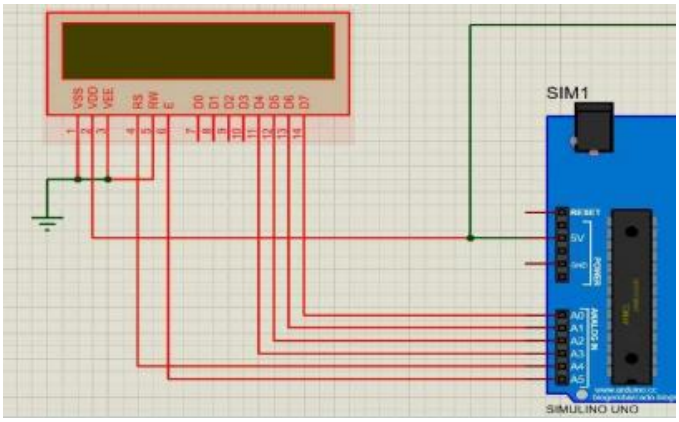


5. IMPLEMENTATION

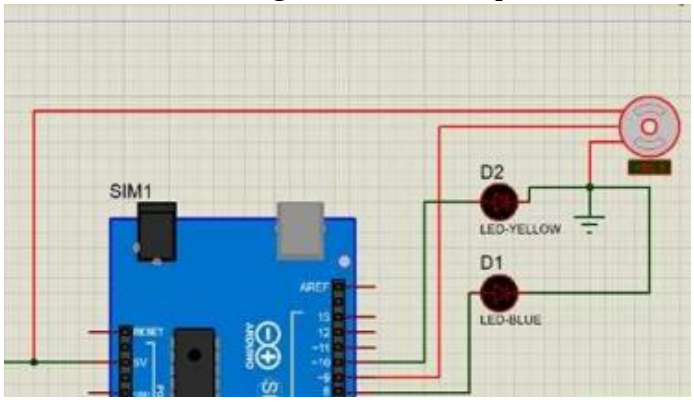
At first, the Keypad-Phone component was connected with the Arduino UNO R3. We connected all the necessary pins of the Keypad to the Arduino board. All 7 pins were connected using connecting wires.



Then we connected the 16x2 Alphanumerical Display to the Arduino Board. This allowed us to view the inserted values and messages. All the pins were connected using connecting wires



Finally, we connected the PWM Powered Servo Motor with the Arduino Board. The motor is used to open and close the locking mechanism of the door. If the Motor rotates 90°, then the lock will open. If it stays in the same position, the lock will stay closed. LED Bulbs were also to further differentiate if the locking mechanism is open or closed



6.1 SOURCE FILES

```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>
Servo myservo;
int pos=0; // LCD Connections
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);
const byte rows=4;
const byte cols=3;

char key[rows][cols]={
{'1','2','3'},
{'4','5','6'},
{'7','8','9'},
{'*','0','#'}
};
byte rowPins[rows]={1,2,3,4};
byte colPins[cols]={5,6,7};
Keypad keypad= Keypad(makeKeymap(key),rowPins,colPins,rows,cols);
char* password="4567";
int currentposition=0;
int redled=10;
int greenled=11;
int buzz=8;
int invalidcount=12;



void setup()
{

displayscreen();
Serial.begin(9600);
pinMode(redled, OUTPUT);
pinMode(greenled, OUTPUT);
pinMode(buzz, OUTPUT);
myservo.attach(9); //SERVO ATTACHED//

lcd.begin(16,2);

}

void loop()
{
if( currentposition==0)
{
displayscreen();

}
int l ;
```

```
char code=keypad.getKey();
if(code!=NO_KEY)
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("PASSWORD:");
lcd.setCursor(7,1);
lcd.print(" ");
lcd.setCursor(7,1);
for(l=0;l<=currentposition;++l)
{

lcd.print("*");
keypress();
}

if (code==password[currentposition])
{
++currentposition;
if(currentposition==4)
{

unlockdoor();
currentposition=0;

}

}

else
{
++invalidcount;
incorrect();
currentposition=0;

}
if(invalidcount==5)
{

++invalidcount;
torture1();

}
if(invalidcount==8)
{
torture2();
}

}
// LOOP ENDS!!!//
}

//********OPEN THE DOOR FUNCTION!!!!**********//
```

```
void unlockdoor()
{
delay(900);

lcd.setCursor(0,0);
lcd.println(" ");
lcd.setCursor(1,0);
lcd.print("Access Granted");
lcd.setCursor(4,1);
lcd.println("WELCOME!!");
lcd.setCursor(15,1);
lcd.println(" ");
lcd.setCursor(16,1);
lcd.println(" ");
lcd.setCursor(14,1);
lcd.println(" ");
lcd.setCursor(13,1);
lcd.println(" ");
unlockbuzz();

for(pos = 180; pos>=0; pos−=5) // goes from 180 degrees to 0 degrees
{
myservo.write(pos); // tell servo to go to position in variable 'pos'
delay(5); // waits 15ms for the servo to reach the position
}
delay(2000);



delay(1000);
counterbeep();

delay(1000);

for(pos = 0; pos <= 180; pos +=5) // goes from 0 degrees to 180 degrees
{ // in steps of 1 degree
myservo.write(pos); // tell servo to go to position in variable 'pos'
delay(15);


currentposition=0;

lcd.clear();
displayscreen();

}
}

//**********WRONG CODE FUNCTION********//

void incorrect()
{
delay(500);
```

```
lcd.clear();
lcd.setCursor(1,0);
lcd.print("CODE");
lcd.setCursor(6,0);
lcd.print("INCORRECT");
lcd.setCursor(15,1);
lcd.println(" ");
lcd.setCursor(4,1);
lcd.println("GET AWAY!!!");

lcd.setCursor(13,1);
lcd.println(" ");
Serial.println("CODE INCORRECT YOU ARE UNAUTHORIZED");
digitalWrite(redled, HIGH);
digitalWrite(buzz, HIGH);
delay(3000);
lcd.clear();
digitalWrite(redled, LOW);
digitalWrite(buzz,LOW);
displayscreen();
}
//************** CLEAR THE SCREEN!!!*************//
void clearscreen()
{
lcd.setCursor(0,0);
lcd.println(" ");
lcd.setCursor(0,1);
lcd.println(" ");
lcd.setCursor(0,2);
lcd.println(" ");
lcd.setCursor(0,3);
lcd.println(" ");
}
//**************KEYPRESS********************//
void keypress()
{



digitalWrite(buzz, HIGH);
delay(50);
digitalWrite(buzz, LOW);
}
//********DISPALAY FUNCTION!!!*************//
void displayscreen()
{

lcd.setCursor(0,0);
lcd.println("*ENTER THE CODE*");
lcd.setCursor(1 ,1);

lcd.println("TO _/_ (OPEN)!!");
}
//*************** ARM SERVO***********//
```

```
void armservo()
{

for (pos=180;pos<=180;pos+=50)
{
myservo.write(pos);
delay(5);
}
delay(5000);

for(pos=180;pos>=0;pos−=50)
{
myservo.write(pos);
}

}
//**********UNLOCK BUZZ**********//
void unlockbuzz()
{

digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(80);
digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(200);
digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(80);
digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(80);
}

//**********COUNTER BEEP**********//
void counterbeep()
{
delay(1200);


lcd.clear();
digitalWrite(buzz, HIGH);

lcd.setCursor(2,15);
lcd.println(" ");
lcd.setCursor(2,14);
lcd.println(" ");
lcd.setCursor(2,0);
delay(200);
lcd.println("GET IN WITHIN:::");
```

```
lcd.setCursor(4,1);
lcd.print("5");
delay(200);
lcd.clear();
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN:");
digitalWrite(buzz,LOW);
delay(1000);
//2
digitalWrite(buzz, HIGH);
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN:");
lcd.setCursor(4,1); //2
lcd.print("4");
delay(100);
lcd.clear();
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN:");
digitalWrite(buzz,LOW);
delay(1000);
//3
digitalWrite(buzz, HIGH);
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN:");
lcd.setCursor(4,1); //3
lcd.print("3");
delay(100);
lcd.clear();
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN:");
digitalWrite(buzz,LOW);
delay(1000);
//4
digitalWrite(buzz, HIGH);
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN:");
lcd.setCursor(4,1); //4
lcd.print("2");
delay(100);
lcd.clear();
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN:");
digitalWrite(buzz,LOW);
delay(1000);
//
digitalWrite(buzz, HIGH);
lcd.setCursor(4,1);
lcd.print("1");
delay(100);
lcd.clear();
lcd.setCursor(2,0);
lcd.println("GET IN WITHIN::");
digitalWrite(buzz,LOW);
```

```
delay(1000);
//5
digitalWrite(buzz, HIGH);
delay(40);
digitalWrite(buzz,LOW);
delay(40);
digitalWrite(buzz, HIGH);
delay(40);
digitalWrite(buzz,LOW);
delay(40);
digitalWrite(buzz, HIGH);
delay(40);
digitalWrite(buzz,LOW);
delay(40);
digitalWrite(buzz, HIGH);
delay(40);
digitalWrite(buzz,LOW);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("RE-LOCKING");
delay(500);
lcd.setCursor(12,0);
lcd.print(".");
delay(500);
lcd.setCursor(13,0);
lcd.print(".");
delay(500);
lcd.setCursor(14,0);
lcd.print(".");
delay(400);
lcd.clear();
lcd.setCursor(4,0);
lcd.print("LOCKED!");
delay(440);
}
//*********TORTURE1**********//
void torture1()
{
delay(1000);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("WAIT FOR ");
lcd.setCursor(5,1);
lcd.print("15 SECONDS");
digitalWrite(buzz, HIGH);
delay(15000);
digitalWrite(buzz, LOW);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("LOL..");
lcd.setCursor(1,1);
lcd.print(" HOW WAS THAT??");
delay(3500);
lcd.clear();
```

```
}
//*****TORTURE2*****//
void torture2()
{
delay(1000);
lcd.setCursor(1,0);
lcd.print(" ");
lcd.setCursor(2,0);
lcd.print("EAR DRUMS ARE");
lcd.setCursor(0,1);
lcd.print(" PRECIOUS!! ");
delay(1500);
lcd.clear();
lcd.setCursor(1,0);
lcd.print(" WAIT FOR");
lcd.setCursor(4,1);
lcd.print(" 1 MINUTE");
digitalWrite(buzz, HIGH);
delay(55000);
counterbeep();
lcd.clear();
digitalWrite(buzz, LOW);
lcd.setCursor(2,0);
lcd.print("WANT ME TO");
lcd.setCursor(1,1);
lcd.print("REDICULE MORE??");
delay(2500);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("Ha Ha Ha Ha");
delay(1700);
lcd.clear();
}
```

7. TESTING

In software development, the last and very important part is to test the software, we have software and the hardware, so we have to test the software and the hardware as well. We will be conducting major testing in software development. IOT testing is a type of testing to check IOT devices. Today there is increasing need to deliver better and faster services. There is a huge demand to access, create, use and share data from any device. The trust is to provide greater insight and control, over various inter connected IOT Devices, Hence IOT testing framework is important.

- Black box testing
- Functionality testing

8. CONCLUSION

Thus "Smart Door Locking System using Arduino" is a modern successor of the conventional door locking system. The conclusion of the discussion of smart Lock using Arduino is the innovation created from the lock system with no more direct contact between the user and the lock. This system is very cost-effective and easy to install. In conclusion, it was discovered that the project performed according to specification and can be implemented. The use of the Arduino UNO microcontroller in this project allows for design simplicity, hence, the project can be achieved in lesser time compared to other techniques previously employed. This work proposes a secure locking/unlocking system based on a keypad and Arduino. Adding password to the Arduino side increase the system security. The system also has a feature for locking itself after some delayed time. This system could be used to prevent houses, companies, institutions from stealing or losing the ordinary key..

9. FUTURE ENHANCEMENTS

The security level can be increased by adding a biometric fingerprint scanner. We can interface sensors like Fire, LPG, PIR motion detector to microcontroller in case of any accident so that door will open automatically. We can interface camera to the micro controller so that it could capture the picture of the thief who is trying to breach the security. This simple circuit can be used at places like home to ensure better safety. With a slight modification, this project can also be used to control the switching of loads through password. It can also be used at organizations to ensure authorized access to highly secured places.

10. REFERENCES

- The working principle of an Arduino, Abuja, Electronics, Computer andComputation(ICECCO), 2014 11th International Conference, IEEE
- http://arduino.cc/tutorial
- http://instructables.com
- Component details http://en.wikipedia.org/