B. Tech. Major Project Report

On

**Simulation OF Password Based Door Locking System**

Submitted in partial fulfilment for the award of B.E degree

*by*

Gaurav Kumar Singh (1840169)

Akash Singh (1840164)

Ankit Kumar Sharma (1840170)

*Under the guidance of*

Dr. Manmohan Singh, Associate Professor



Department of Electrical Engineering

Sant Longowal Institute of Engineering and technology, Sangrur Punjab, 148106 (India)

Dec. 2020

**Department of Electrical Engineering**

**Sant Longowal Institute of Engineering and Technology**

**Sangrur -148106 (India)**

# UNDERTAKING

We declare that the work presented in the project entitled “**Password Based Door Locking System Using Arduino || Proteus Simulation**” submitted to the **Department of Electrical Engineering,**  work. We neither have plagiarized any part of the present project nor submitted the same work for the award of any other degree elsewhere.

In case this undertaking is found incorrect, the B.Tech. Project may be withdrawn unconditionally.

|  |  |  |
| --- | --- | --- |
| **(Gaurav Kumar Singh)**  **(1840169)** | **(Ankit Kumar Sharma)**  **(1840170)** | **(Akash Singh)**  **(1840164)** |



**Department of Electrical Engineering Sant Longowal Institute of Engineering and Technology**

**Sangrur -148106 (India)**

# CERTIFICATE

This is to certify that project entitled “**Simulation OF** **Password Based Door Locking System**” submitted by **Gaurav Kumar Sharma (1840169) ,Akash Singh (1840164), Ankit Kumar Sharma (1840170),** in partial fulfilment of the requirements for award of the degree of Bachelor of Technology (Electrical Engineering) to Sant Longowal Institute of Engineering and technology (India) has been carried out under my supervision and is an authentic record of students own work to the best of my knowledge and belief.

**Dr. Manmohan Singh**

Associate Professor

Date : 24th May 2022

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our mentor **Dr. Manmohan Singh, Associate Professor of Electrical Engineering Department** for guiding us thoughtfully and efficiently. We have got the opportunity to work under this project and he also imparted useful suggestions, motivation and instructions whenever necessary. We would also like to thank our mentor for being a great source of motivation and for providing encouragement throughout the length of this project.

We are also thankful to the **Dr. Surita Maini, Head, Department of Electrical Engineering** for providing the opportunity to learn the technical knowledge and practical aspects of the topic.

We are also thankful to the project panel members who made us go through a rigorous process of learning which further helped us to have enhanced understanding of this topic.

We offer our sincere thanks to all other persons who knowingly or unknowingly helped us to complete this project.

**INTRODUCTION**

Security is a major concern in our daily life. Everyone wants to defend themselves in the same way as much as possible. Access control forms an important link in the security chain. The I microcontroller-based door locker is an access control system that allows only 4, authorized persons to enter restricted areas. The system is fully controlled by a ATmegaA328P 8-bit microcontroller with 2K ROM for system memory. The I password is stored in the EPROM and can be changed at any time. The program has a keyboard for entering passwords. When the password is input, the relay is reset after being the same as the password stored in the memory, and the door is turned on. If you enter the wrong password more than three times, it means alarm is open. The default password is 1111, 1234, old password, new password You can change the password by pressing the # button. For example, if you change the password from the default to 4523 and try to dial 123411114523 #, you will hear beeps after dialing, which is for efficient operation. There are two transfers. One opens the door and the other closes the door. Button should be positioned inside the door so that someone inside can open and close the door. The password is stored in the EEPROM chip. EEPROM operates on I2C technology.

### **List of components and Softwares uses :**

1. Arduino Uno/Pro/Mini or Custom board using Atmega  328p Microcontroller
2. 16 x 2 LCD (Liquid Crystal Display), LM016L
3. 4 x 3 or 4 x 4 matrix keypad for Arduino
4. Servo motor
5. Software for simulation Proteus
6. Arduino IDE

**Literature Survey**

Other Door Locks System Available

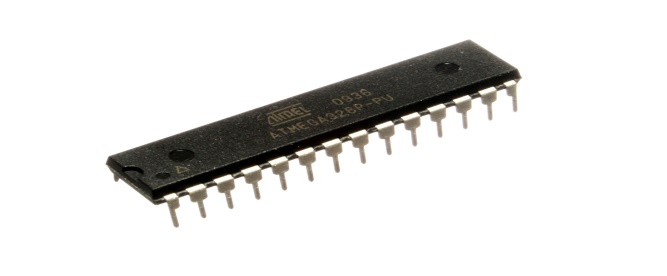
There are many security systems on the market today. This system is being studied into get an idea for the project it was built on. The systems on the market are now equipped with new technology and are more advanced, but the idea of ​​the still needs to be obtained through literature review. Literature review work helps to identify and develop skills for retrieving information from a variety of sources. These technologies are critical to solving the challenges that will or will face in the future.

PARADOX SECURITY SYSTEM  
This security system is manufactured by Paradox. There are many types of  
modules in the system, one of which is a wireless expansion module that acts as a receiver for a wireless source such as a cell phone. This gives an idea for this project using a cell phone. There are several ways to activate the control panel. For this project, the idea of ​​using wireless communication was applied to the construction of a system.

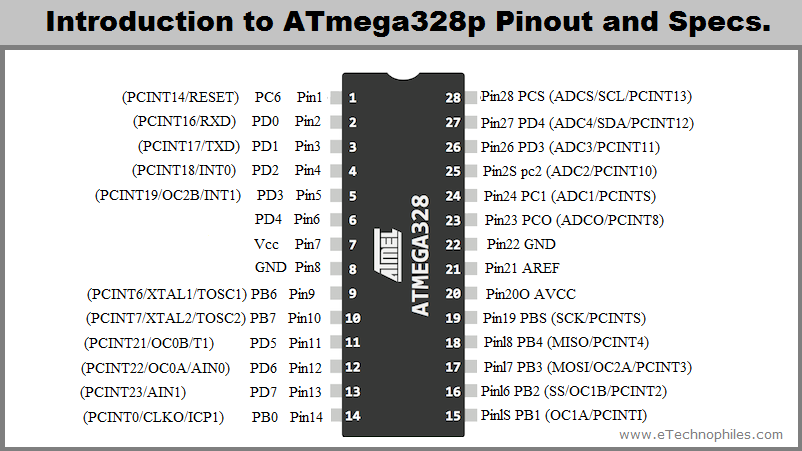
**About Components :**

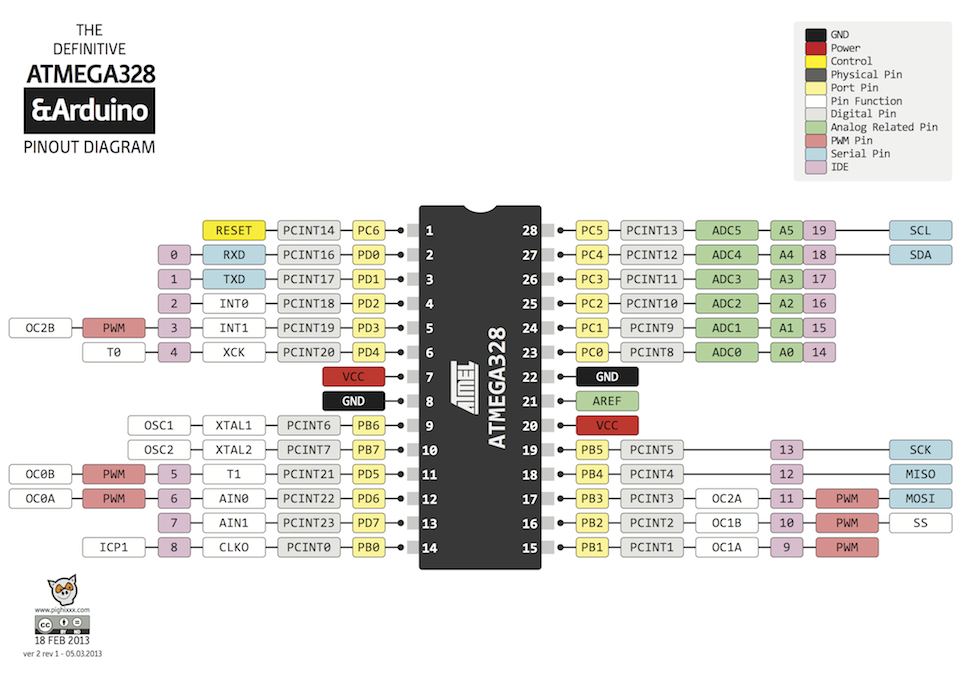
1. **ATmega328p Pinout, Datasheet and Specifications Introduction**

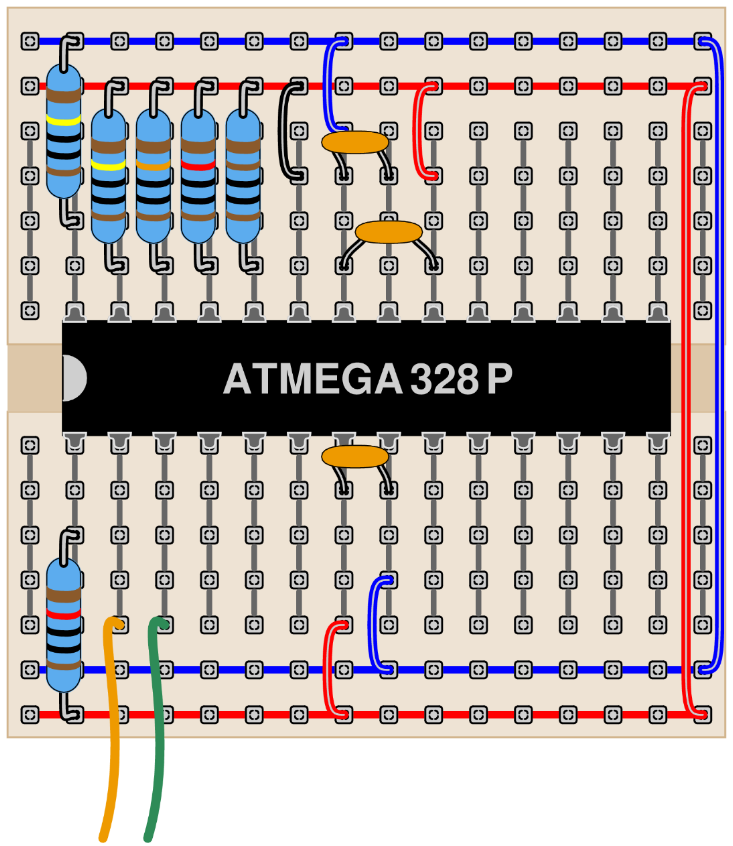
The ATmega328p is a single-chip, high-performance, efficient microcontroller made by Atmel in the megaAVR family. This is an 8-bit AVR RISC microcontroller. In this post, we will go into more detail about the ATmega328p pinout, datasheet, specs, and how to program it.



It consists of 32 KB read/write ISP flash memory, 2 KB static RAM (SRAM), 1 KB EEPROM, 23 general purpose I/O pins, clocked at 16 MHz, 32 general purpose working registers, and 3 flexible timers/counters (for comparison). mode. (two 8-bit and one 16-bit), internal and external interrupts, serial programmable UART, byte-oriented integrated circuit (IC) interface, SPI serial port, 6-channel 10-bit A/D converter, programmable watchdog timer internal generator and 5 programs for selecting energy saving mode. The device operates over a voltage range of 1.8-5.5 volts.







|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PIN NO.** | | **PIN NAME** | **PIN FUNCTION** | **PIN FUNCTION DISCRIPTION** | |
| 1 | | PC6 | RESET | This pin helps to reset the microcontroller. | |
| 2 | | PDO | Digital Pin(RX) | This is the input pin for serial communication | |
| 3 | | PD1 | Digital Pin (TX) | This is the output pin for serial communication | |
| 4 | | PD2 | Digital Pin | It is used as an external interrupt 0 | |
| 5 | | PD3 | Digital Pin (PWM) | It is used as an external interrupt 1 | |
| 6 | | PD4 | Digital Pin | It is used for external counter source Timer0 | |
| 7 | | Vcc | Positive Voltage | Positive supply of the system. | |
| 8 | GND | | Ground | The Ground of the system | |
| 9 | XTAL | | Crystal Oscillator | This pin should be connected to one pin of the crystal oscillator to provide an external clock pulse to the chip | |
| 10 | XTAL | | Crystal Oscillator | This pin should also be connected to the other pin of the crystal oscillator to provide an external clock pulse to the chip | |
| 11 | PD5 | | Digital Pin (PWM) | Pin 11 is used for external counter source Timer1 |
| 12 | PD6 | | Digital Pin (PWM) | Positive Analog Comparator i/ps |
| 13 | PD7 | | Digital Pin | Negative Analog Comparator i/ps |
| 14 | PB0 | | Digital Pin | Counter or Timer input source pin |
| 15 | PB1 | | Digital Pin (PWM) | Counter or Timer compare match A. |
| 16 | PB2 | | Digital Pin (PWM) | This pin act as a slave choice i/p. |
| 17 | PB3 | | Digital Pin (PWM) | This pin is used as a master data output and slave data input for the SPI interface. |
| 18 | PB4 | | Digital Pin | This pin act as a master clock input and slave clock output. |
| 19 | PB5 | | Digital Pin | This pin act as a master clock output and slave clock input for SPI. |
| 20 | AVcc | | Positive Voltage | Positive voltage for ADC (power) |

|  |  |  |  |
| --- | --- | --- | --- |
| 21 | AREF | Analog Reference | Analog Reference voltage for ADC (Analog to Digital Converter) |
| 22 | GND | Ground | The Ground of the system |
| 23 | PC0 | Analog Input | Analog input digital value (channel 0) |
| 24 | PC1 | Analog Input | Analog input digital value (channel 1) |
| 25 | PC2 | Analog Input | Analog input digital value (channel 2) |
| 26 | PC3 | Analog Input | Analog input digital value (channel 3) |
| 27 | PC4 | Analog Input | Analog input digital value (channel 4). This pin can also be used as a serial interface connection for data. |
| 28 | PC5 | Analog Input | Analog input digital value (channel 5). This pin is also used as a serial interface clock line. |

**ATmega328 Specification**

|  |  |
| --- | --- |
| Program Memory Type | Flash |
| Program Memory Size (KB) | 32 |
| CPU Speed (MIPS/DMIPS) | 20 |
| SRAM (B) | 2048 |
| Data EEPROM/HEF (bytes) | 1024 |
| Digital Communication Peripherals | 1-UART, 2-SPI, 1-I2C |
| Capture/Compare/PWM Peripherals | 1 Input Capture, 1 CCP, 6PWM |
| Timers | 2 x 8-bit, 1 x 16-bit |
| Number of Comparators | 1 |
| Temperature Range (°C) | -40 to 85 |
| Operating Voltage Range (V) | 1.8 to 5.5 |
| Pin Count | 32 |
| Low Power | YES |

**ATmega328p based Microcontroller boards:**

Adafruit METRO 328 - The Adafruit METRO 328 is a very convenient microcontroller. I have an ATmega328 on a core with 32KB Flash and 2KB RAM running at 16MHz.

Arduino Pro Mini 328 - The Arduino Pro Mini 328 is a microcontroller board based on the ATmega328 chip. It consists of 14 digital inputs/outputs (6 of which are PWM outputs), 6 analog inputs, a built-in 8 MHz resonator, a reset button and holes for attaching pin headers. The 6-pin connector connects to an FTDI cable or TTL serial adapter to power the board and can be used for programming purposes.

Arduino Uno R3 — The Arduino Uno is an open source microcontroller board based on the ATmega328P microchip. It consists of 6 analog input pins, 14 digital input/output pins (6 of which are PWM capable), 16 MHz ceramic crystal resonator, USBB port, ICSP connector, power connector and reset button.

Arduino Nano - The Arduino Nano has the same functionality but is smaller than the Arduino Uno. Another difference is that the Nano doesn't have a DC power jack and is powered by a MiniB USB cable instead of a standard cable.

ATmega328p Programming Methods:

Microcontroller programming can be a bit difficult as it requires special programmers and .hex files. Also, knowledge of the C programming language is required to write code.

To make it easier, Arduino created a .hex file that can be installed on AVR chips that allow us to program the board over a serial port. This ensures that in order to program a microcontroller (after installing the hex file) all you need is a serial connection between the microcontroller and the Computer which can be achieved using a USB to UART adapter. This hex file is called the Arduino Bootloader.

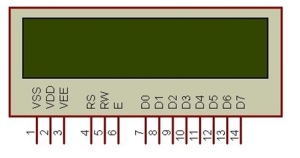
During the manufacturing process for electronic devices, microcontrollers are usually programmed after they have been mounted on the PCB. This programming method is called as In System Programming (ISP) and it requires the PCB to have certain header pins through which the microcontroller can be accessed for programming.

Most Arduino boards have a 2x3 pin header used to program the InSystem known as In-Circuit Serial Programming (ICSP). The Arduino ICSP header pins consist of three SPI pins (MOSI (D11), MISO (D12), SCK (D13)), VCC, GND and reset pins. By connecting the Atmega328p microcontroller to the ICSP pins, you can flash the Atmeg328p microcontroller using the Arduino bootloader.

1. **16 x 2 LCD (Liquid Crystal Display), LM016L :**

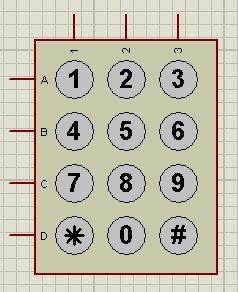
The most commonly used character-based LCDs are based on the Hitachi HD44780 controller or other controller compatible with the HD44580. We will be discussing the tricks you can do with this simple LCD that can give a character LCD, interaction with various microcontrollers, various interfaces (8-bit/4-bit), programming, and special tasks and applications a new look.

|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Name** | **Description** |
| Pin no. 1 | **VSS** | Power supply (GND) |
| Pin no. 2 | **VCC** | Power supply (+5V) |
| Pin no. 3 | **VEE** | Contrast adjust |
| Pin no. 4 | **RS** | 0 = Instruction input 1 = Data input |
| Pin no. 5 | **R/W** | 0 = Write to LCD Module 1 = Read from LCD module |
| Pin no. 6 | **EN** | Enable signal |
| Pin no. 7 | **D0** | Data bus line 0 (LSB) |



**3) Keypad for Arduino**

The keyboard is used when you need to use number buttons or when you need to use a lot of buttons to send commands, such as in some applications. Since you have to use 10 buttons, I recommend using a keyboard that saves a lot of time both in hardware and programming instead of using 10 separate buttons. So today we will take a closer look at how the keyboard works and how to interact with the Arduino with an Arduino in Proteus ISIS. Proteus also provides keyboard components in its database, making it easy to model in Proteus and save time. So, model first, then design your hardware.



1. **Servo motor :**

Servo motors are common motors used in engineering projects for precise circular motion. You can move servo motors to any angle you want, which is not possible with other motors like stepper motors or DC motors.

#### **PRINCIPLE OF SERVO MOTOR**

“Servo Motors are basically DC Motors with addition circuit that support in achieving precise position of the servo motor. In order to manage the rotation of the servo motor’s shaft, you require a special signal called Pulse Width Modulation signal.”

#### **APPLICATIONS OF SERVO MOTOR**

* This motor can be utilized to manage the robotic vehicle by controlling robot speed, create plenty torque to move and also start and stop the robot.
* Servo Motor is highly cost-effective.
* Servo Motor has compact size.
* Servo Motor has quick response and high speed.

#### **TYPES OF SERVO MOTOR**

These motors are classified into different types according to their application, such as brushless DC, continuous rotation, linear and constant position rotation, etc. A typical servo motor consists of three wires: power, control, and ground. The layout and dimensions of these motors depend on the application. The most common motor is the RC servo motor, which is used in applications such as robotics, reliability and convenience of microprocessor control.

Connecting Servo Motor To Atmega 328P:-

Servo Motor has three pins:

1. First Pin is **Vcc**.
2. Second Pin is **Control Pin**.
3. Third Pin is **GND**.

**5) Proteus**

Circuit design software developed by Lab center Electronics. It is used to design and simulate various circuits on a printed circuit board (PCB). Using proteus for any electronic circuit design makes your project cost-effective and error-prone by building circuits on proteus.

#### **Features of Proteus**

* There are 2 main parts of proteus first is used to design and draw different circuits and the second is for designing of PCB layout.
* First is ISIS that used to design and simulate circuits. And second is ARES that used for designing of a printed circuit board.
* It also provides features related to the three-dimensional view of design in PCB.

**6) Arduino Ide**

The Arduino Integrated Development Environment, or Arduino Software (IDE), contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a set of menus. It connects to the Arduino hardware to download programs and communicate.

**Code used :-**

#include <LiquidCrystal.h>

#include <Servo.h>

#include <Keypad.h>

Servo myservo;

int pos=0; // position of servo motor

LiquidCrystal lcd(A4, A5, A3, A2, A1, A0);

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]={0,1,2,3};

byte colPins[cols]={4,5,6};

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

char\* password="0123";

int currentposition=0

void setup()

{

displayscreen();

//Serial.begin(9600);

myservo.attach(9); //Servo motor connection

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

{

incorrect();

currentposition=0;

}

}

}

//------------------ Function 1- OPEN THE DOOR--------------//

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(" ");

for(pos = 180; pos>=0; pos-=5) // open the door

{

myservo.write(pos);

delay(5);

}

delay(2000);

delay(1000);

counterbeep();

delay(1000);

for(pos = 0; pos <= 180; pos +=5) // close the door

{ // in steps of 1 degree

myservo.write(pos);

delay(15);

currentposition=0;

lcd.clear();

displayscreen();

}

}

//--------------------Function 2- Wrong code--------------//

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");

delay(3000);

lcd.clear();

displayscreen();

}

//-------Function 3 - 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(" ");

}

//------------Function 4 - DISPLAY FUNCTION--------------------//

void displayscreen()

{

lcd.setCursor(0,0);

lcd.println("\*ENTER THE CODE\*");

lcd.setCursor(1 ,1);

lcd.println("TO OPEN DOOR!!");

}

//--------------Function 5 - Count down------------------//

void counterbeep()

{

delay(1200);

lcd.clear();

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:");

delay(1000);

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:");

delay(1000);

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

lcd.setCursor(4,1);

lcd.print("3");

delay(100);

lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

delay(1000);

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

lcd.setCursor(4,1);

lcd.print("2");

delay(100);

lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

delay(1000);

lcd.setCursor(4,1);

lcd.print("1");

delay(100);

lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN::");

delay(1000);

delay(40);

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!");

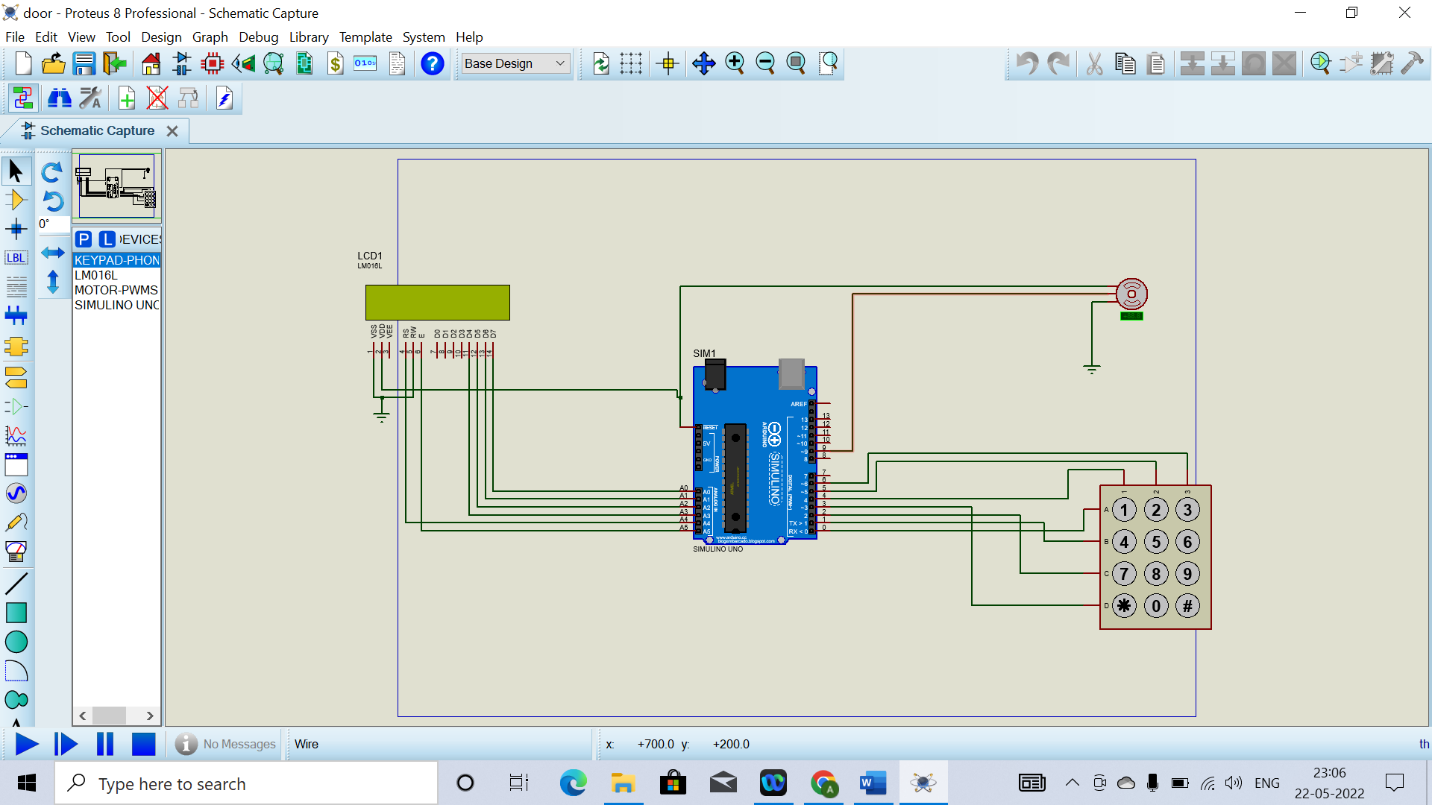
delay(440);

}

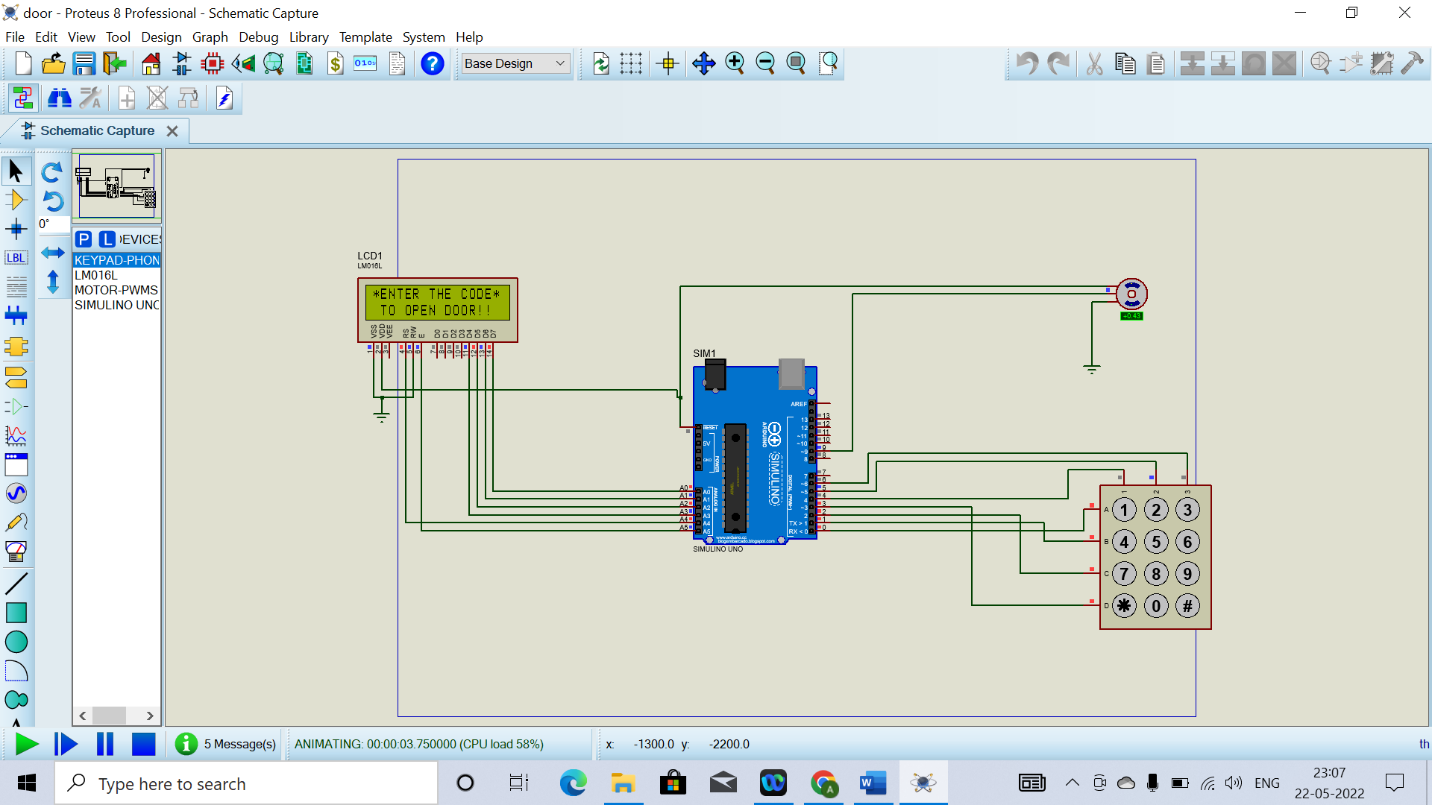
**Github Public repoistory Link :-**

<https://github.com/akkivanguu/Door-Locking-Syytem.git>

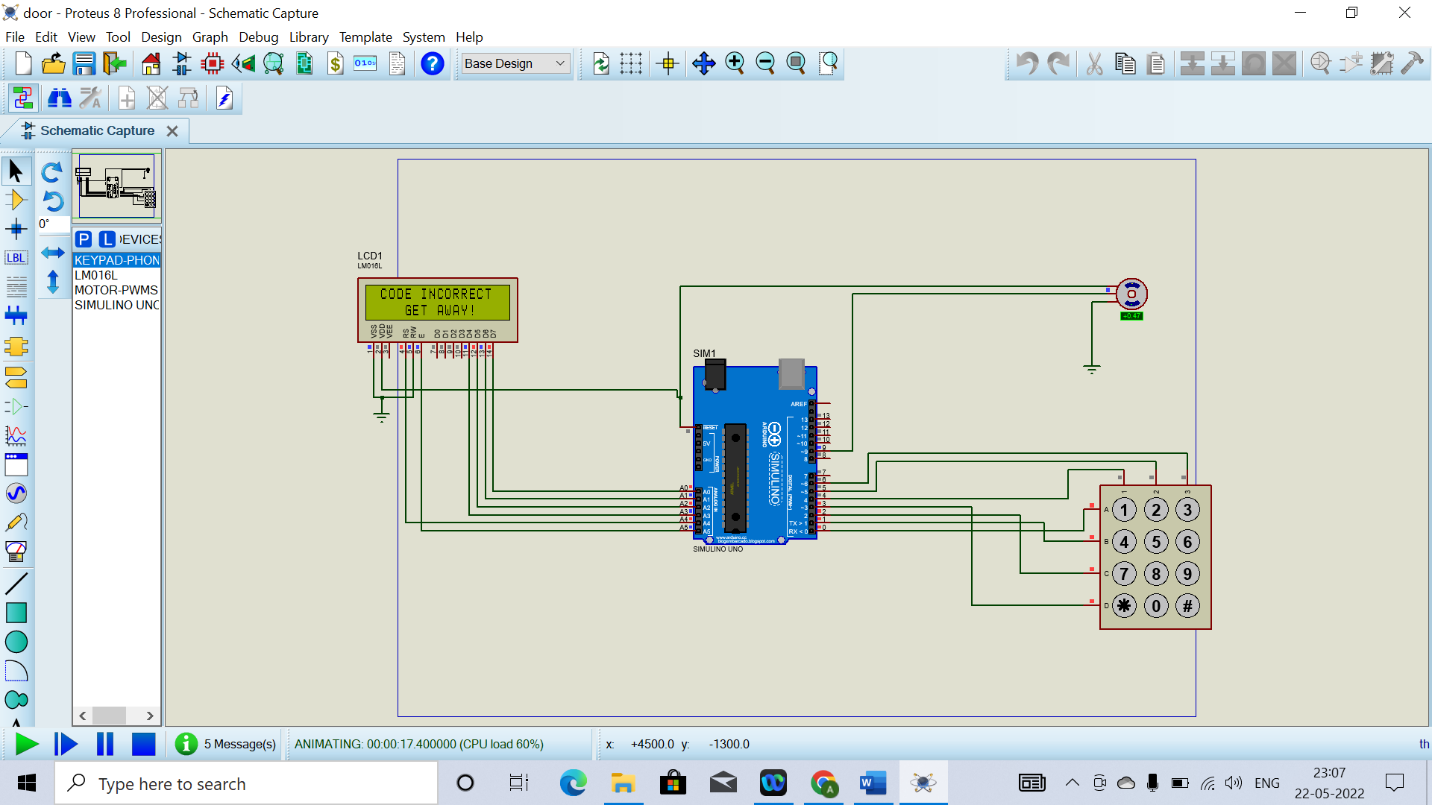
**Circuit Of Simulatuon :-**

****

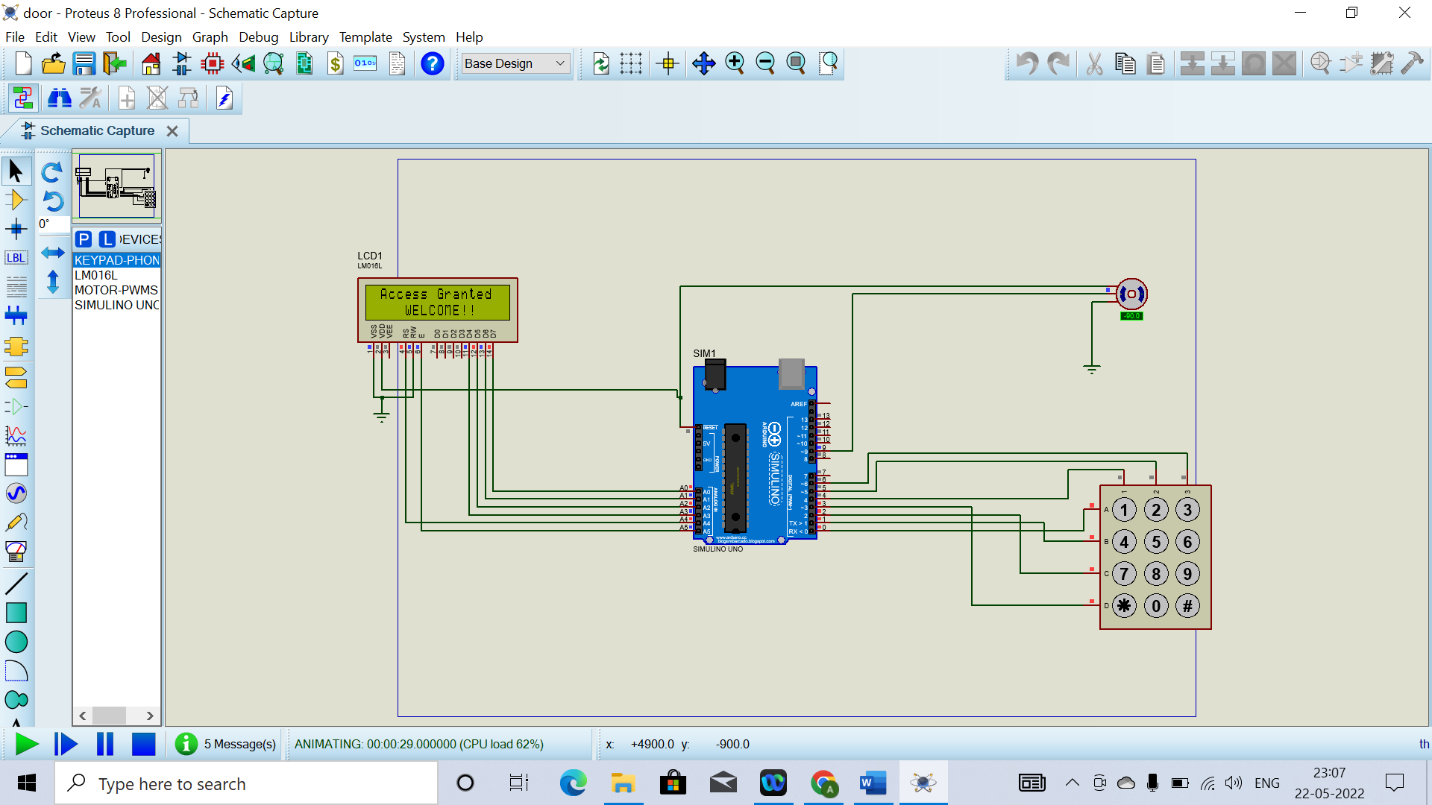
**i)Enter Code/Password**

****

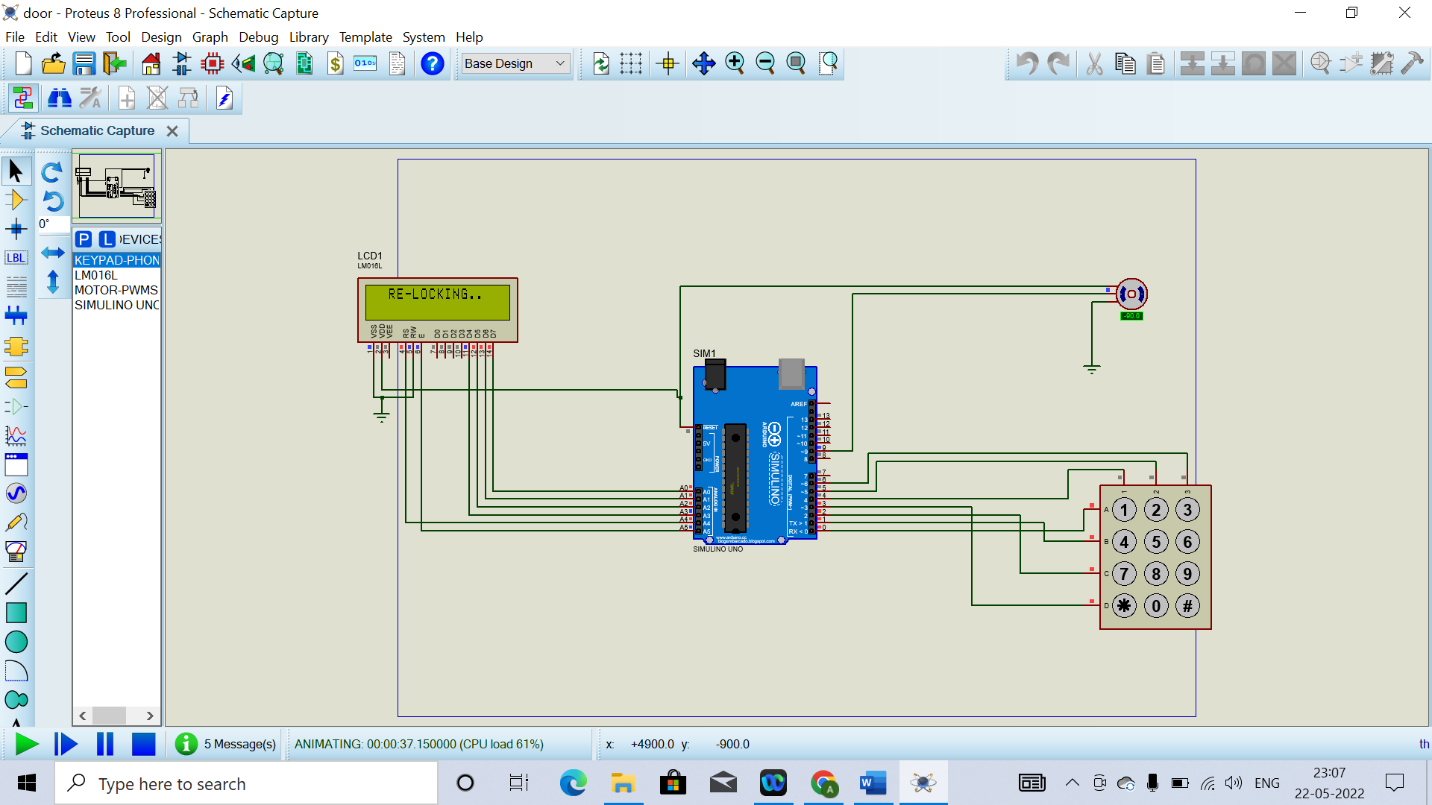
**ii) Password Incorrect**

****

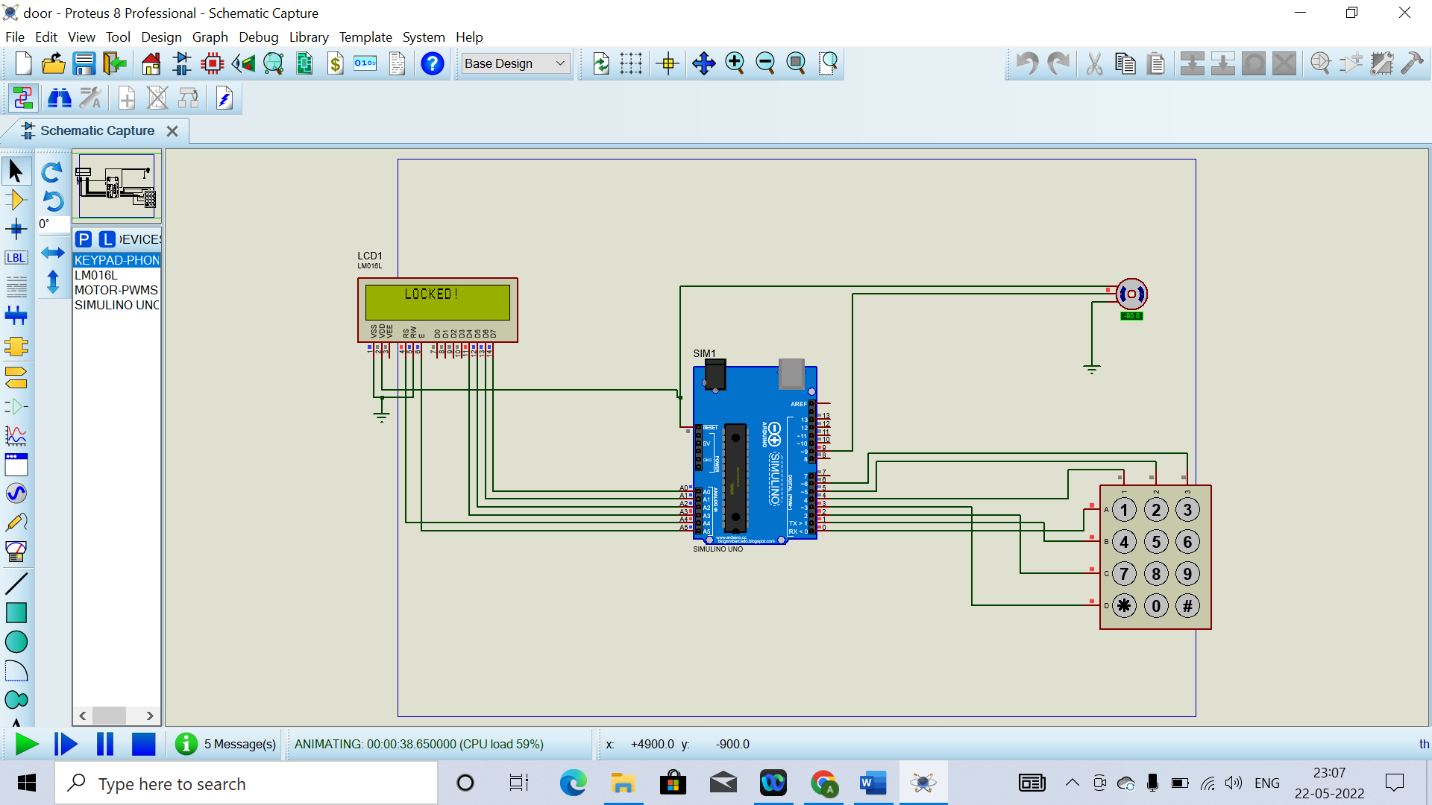
**iii)Password correct Access granted**

****

**iv) Relocking of Door**

****

**v) Now Door is again locked**

****

**Reference :-**

<https://opencircuit.shop/product/Atmega328P-PU-DIP-28>

<https://www.theengineeringprojects.com/2015/11/control-servo-motor-arduino-proteus.html>

<https://embeddedcenter.wordpress.com/ece-study-centre/display-module/lcd-16x2-lm016l/>

<https://www.theengineeringprojects.com/2015/12/interfacing-keypad-arduino-proteus-isis.html>