Tinkercad Link: https://www.tinkercad.com/things/42jR8zw3rga-door-lock

NAME: REGINA D | PROJECT – 2 |

KEYPAD-BASED SERVO LOCK SYSTEM SIMULATION IN TINKERCAD

OBJECTIVE

To design and implement a keypad-based servo lock system using an Arduino, a 4x4 keypad, an LCD display, a servo motor, and a buzzer. This system ensures secure access control by allowing entry only when the correct password is entered.

1. SOFTWARE REQUIRED: Tinkercad

2. TINKERCAD

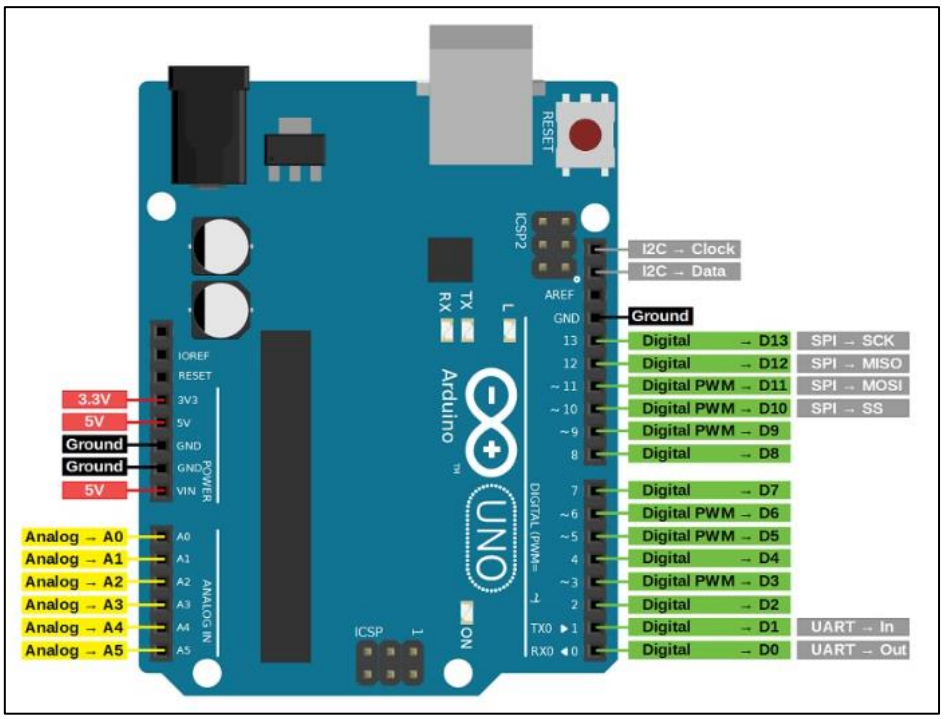
Tinkercad is a free, web-based simulation and design tool that allows users to create and test electronic circuits, 3D models, and code-based projects. This project is done by electronics circuit simulation.

3. COMPONENTS

S. No	NAME	QUANTITY
1.	Arduino Uno R3	1
2.	4X4 Keypad	1
3.	LCD 16X2 Display	1
4.	Positional Servo Motor	1
5.	Piezo Buzze	1

3.1 ARDUINO UNO R3

The Arduino Uno R3 is a microcontroller board based on the ATmega328P. It operates at 5V, has 14 digital I/O pins (6 PWM), 6 analog inputs, 32 KB Flash memory, and a 16 MHz clock speed. It is widely used for IoT, robotics, automation, and prototyping due to its beginner-friendly and open-source nature.



3.1. 1 TECHNICAL SPECIFICATIONS

S. No	FEATURE	SPECIFICATION
1	Microcontroller	ATmega328P
2	Operating Voltage	5V
3	Input Voltage (Recommended)	7-12V
4	Input Voltage (Limits)	6-20V
5	Digital I/O Pins	14 (6 PWM)
6	Analog Input Pins	6
7	Flash Memory	32 KB (0.5 KB used by bootloader)
8	SRAM	2 KB
9	EEPROM	1 KB
10	Clock Speed	16 MHz
11	Communication	UART, I2C, SPI
12	USB Interface	Type B USB
13	Reset Button	Yes
14	Dimensions	68.6 mm × 53.4 mm

3.1.2 PIN DESCRIPTION

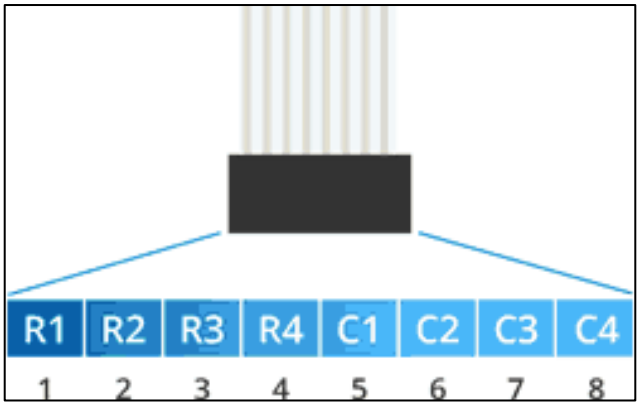
1. **Vin:** This is the input voltage pin of the Arduino board used to provide input supply from an external power source.
2. **5V:** This pin of the Arduino board is used as a regulated power supply voltage and it is used to give supply to the board as well as onboard components.
3. **3.3V:** This pin of the board is used to provide a supply of 3.3V which is generated from a voltage regulator on the board
4. **GND:** This pin of the board is used to ground the Arduino board.
5. **Reset:** This pin of the board is used to reset the microcontroller. It is used to Resets the microcontroller.
6. **Analog Pins:** The pins A0 to A5 are used as an analog input and it is in the range of 0-5V.
7. **Digital Pins:** The pins 0 to 13 are used as a digital input or output for the Arduino board.
8. **Serial Pins:** These pins are also known as a UART pin. It is used for communication between the Arduino board and a computer or other devices. The transmitter pin number 1 and receiver pin number 0 is used to transmit and receive the data resp.
9. **External Interrupt Pins:** This pin of the Arduino board is used to produce the External interrupt and it is done by pin numbers 2 and 3.
10. **PWM Pin:** This pin of the board is used to convert the digital signal into an analog by varying the width of the Pulse. The pin numbers 3,5,6,9,10 and 11 are used as a PWM pin.
11. **SPI Pins:** This is the Serial Peripheral Interface pin, it is used to maintain SPI communication with the help of the SPI library. SPI pins include:
12. **SS:** Pin number 10 is used as a Slave Select
13. **MOSI:** Pin number 11 is used as a Master Out Slave In
14. **MISO:** Pin number 12 is used as a Master In Slave Out
15. **SCK:** Pin number 13 is used as a Serial Clock

16. **LED Pin:** The board has an inbuilt LED using digital pin-13. The LED glows only when the digital pin becomes high.
17. **AREF Pin:** This is an analog reference pin of the Arduino board. It is used to provide a reference voltage from an external power supply.

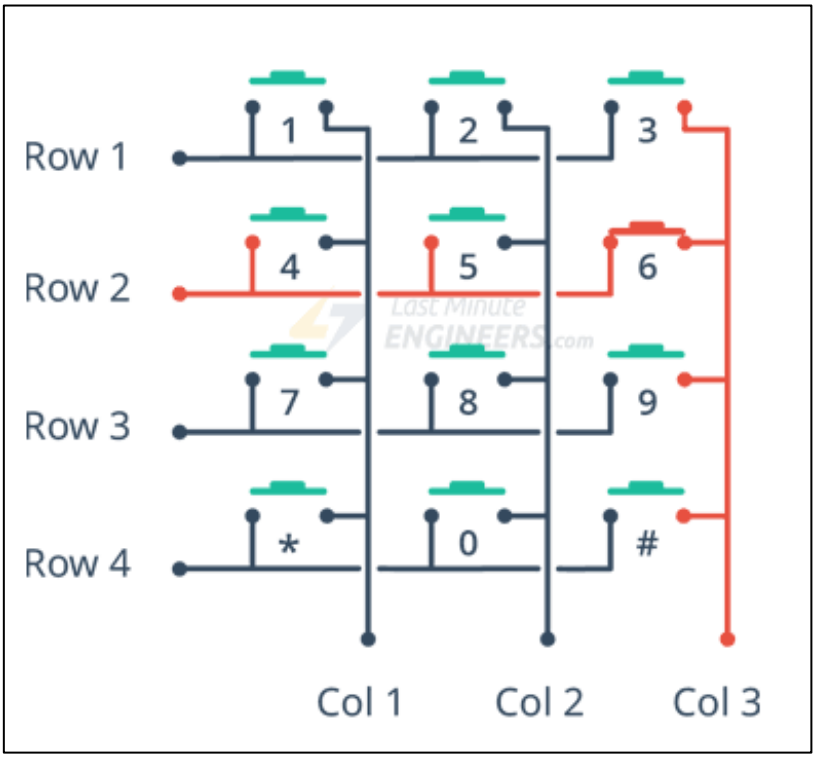
3.2 4x4 KEYPAD

A keypad is used as an input device to read and process the user's keystrokes. In the A 4x4 keypad contains four rows and four columns. Between the rows and columns, switches are positioned. By pressing the corresponding key establishes a connection between the row and column are connected of that switch.

3.2.1 PIN DIAGRAM



3.2.2 DIAGRAM



3.2.3 WORKING PRINCIPLE

1. Each row is connected to an input pin, and each column is connected to an output pin.
2. Input pins are pulled HIGH by enabling internal pull-up resistors.
3. The microcontroller then sequentially sets the pin for each column LOW and then checks to see if any of the row pins are LOW. Because pull-up resistors are used, the rows will be high unless a button is pressed.
4. If a row pin is LOW, it indicates that the button for that row and column is pressed.
5. The microcontroller then waits for the switch to be released. It then searches the keymap array for the character that corresponds to that button.

3.3 LCD 16X2 DISPLAY WITH ATTACHED PCF8574 (I2C MODULE

LCD module is applied to display characters, letters, numbers, real time clock and temperature. 1602 LCD display with I2C serial interface is compatible with for Arduino and Raspberry pi, display function: LCD display with I2C serial interface can be applied to display real time clock, temperature, humidity and so on.

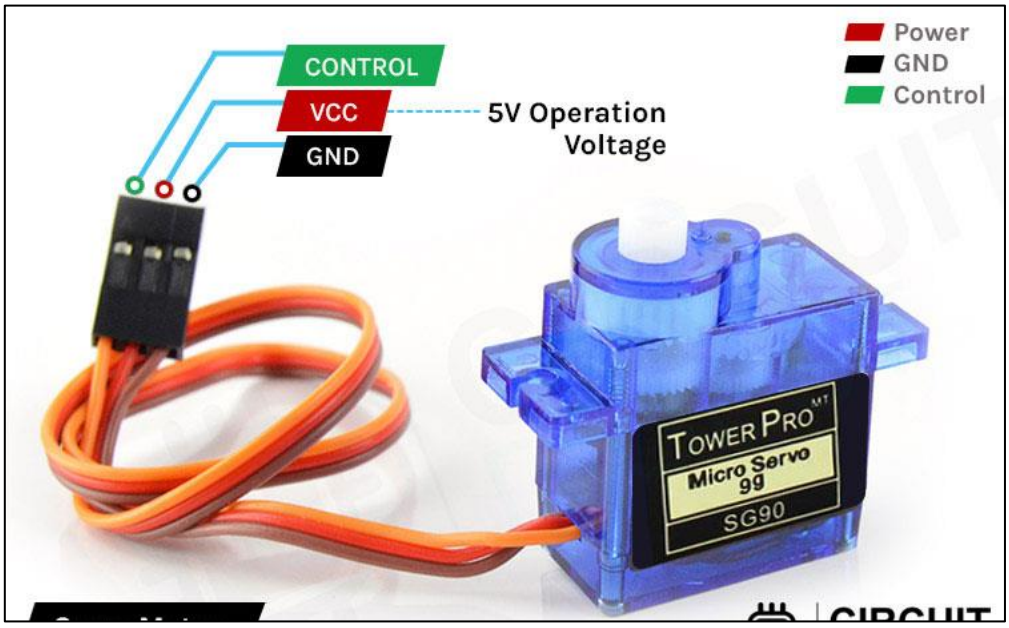
3.3.1 SPECIFICATIONS

1. **LCD screen** can display 2-lines x 16-characters.
2. **Working voltage:** 5V.
3. **Potentiometer:** To adjust backlight and contrast.
4. **I/O ports:** GND, VCC, SDA (serial data line), SCL (serial clock line),

3.4 SERVO MOTORS

A positional servo motor is a type of servo motor designed for precise angular positioning within a limited range, typically 0° to 180°, though some models extend up to 270° or 360°. It operates using pulse-width modulation (PWM) control, where the duty cycle of the input signal determines the desired angular position.

3.4.1 PIN DIAGRAM



3.4.2 WORKING PRINCIPLE

1. The control signal (PWM – Pulse Width Modulation) sent from the microcontroller (Arduino in this case) determines the angle of rotation.
2. The internal potentiometer continuously monitors the current position and adjusts it to match the desired angle.
3. When a signal corresponding to a specific angle (e.g., 180°) is received, the motor rotates and holds its position.
4. The motor stays in place until a new signal is given.

3.4.3 HOW IT WORKS IN THE PROJECT

- When the correct password is entered, the Arduino sends a PWM signal to rotate the servo to 180° (unlock position).
- After a delay, the Arduino sends another PWM signal to return the servo to 0° (locked position).
- The motor holds its position due to its closed-loop control system.

3.4.4 SPECIFICATIONS

Specification	Details
Torque	1.8 kg-cm (at 4.8V)
Speed	0.10 sec/60°
Weight	9 grams
Gear Type	Plastic
Rotational Range	180°
Features	Easy to use, Good quality product

3.5 ACTIVE PIEZO BUZZER

The Active Piezo Buzzer is a compact and efficient sound-emitting device that operates within a DC voltage range of 3V to 24V. It is designed to produce a continuous tone when powered, making it ideal for alarms, notifications, and warning systems. With a sound output of 85 dB, it ensures clear and audible alerts in various applications.

3.5.1 WORKING PRINCIPLE

1. When an electric signal (voltage) is applied, the piezoelectric material expands and contracts, causing the metal plate to vibrate.
2. These vibrations create audible sound waves.
3. The frequency of the signal determines the pitch of the sound.

3.5.2 HOW IT WORKS IN THE PROJECT

1. When a key is pressed, the buzzer beeps briefly to confirm the input.
2. If the wrong password is entered, the buzzer beeps multiple times as an alert.
3. The Arduino controls the buzzer by turning it ON and OFF using digitalWrite() and delay().

3.5.3 SPECIFICATIONS

S. No	Specification	Details
1	Operating Voltage	DC 3V – 24V
2	Sound Level	85 dB
3	Type	Active Buzzer (self-driving)
4	Usage	Alarm systems, alerts, notifications
5	Ease of Use	Simple two-wire connection

4. PROGRAM

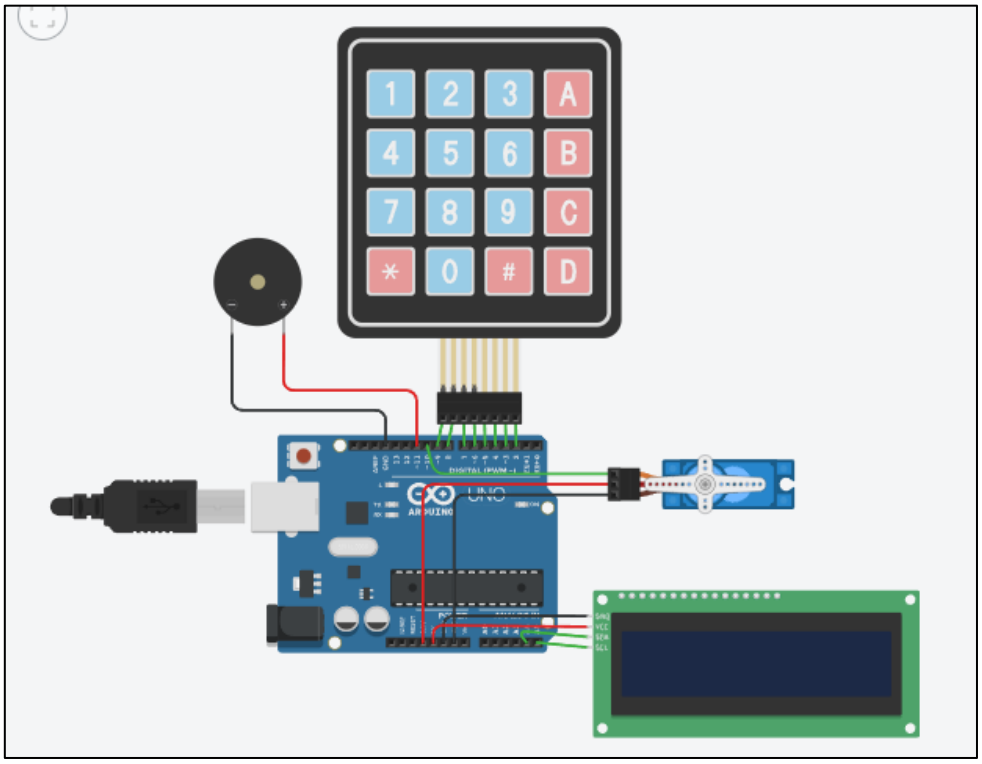
```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <Keypad.h>

#include <Servo.h>


LiquidCrystal_I2C lcd(0x27, 16, 2); //lcd(address, col, row);


Servo servo_lock; //creating object

const int servo_pin= 10;

const int buzzer_pin= 11;


const int rows= 4;

const int cols=4;


char keys[rows][cols]=

{

  {'1', '2', '3', 'A'},

  {'4', '5', '6', 'B'},

  {'7', '8', '9', 'C'},

  {'*', '0', '#', 'D'}

};


byte rowpins[rows]= {9,8,7,6};

byte colpins[cols]= {5,4,3,2};

Keypad keypad= Keypad(makeKeymap(keys), rowpins, colpins, rows, cols);
```

```
const char original_password[9]="A123456B";
char entered_password[9];

void setup()
{
  Wire.begin();
  lcd.init(); //lcd,begin(cols, rows);
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("ENTER PASSWORD");

  pinMode(buzzer_pin, OUTPUT);
  servo_lock.attach(servo_pin);
  servo_lock.write(0);
}

void loop()
{
  int i = 0;
  lcd.setCursor(0, 1);

  while (i < 8)
  {
    char key = keypad.getKey();

    if (key)  {
```

```
digitalWrite(buzzer_pin, HIGH);
    delay(20);
    digitalWrite(buzzer_pin, LOW);

    entered_password[i] = key;
    lcd.setCursor(i, 1);
    lcd.print('*');
    i++;
  }
 }
entered_password[8]='\0';
 if(strcmp(entered_password, original_password)==0)
 {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("DOOR OPENED");
  servo_lock.write(90);
  delay(20);
  servo_lock.write(180);
  delay(3000);
  servo_lock.write(180);
  delay(20);
  servo_lock.write(90);
  delay(20);
  servo_lock.write(0);
 }
```

```
  else
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("ACCESS DENIED");
    delay(500);
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("ENTER PASSWORD");
}
```

5. CIRCUIT DIAGRAM



5.1 PROCEDURE

Step 1: Connecting the LCD (16x2 I2C)

1. VCC - 5V (Arduino)
2. GND - GND (Arduino)
3. SDA - A4 (Arduino)
4. SCL - A5 (Arduino)

Step 2: Connecting the 4x4 Keypad

1. R1, R2, R3, R4 – 9, 8, 7, 6 pins (Arduino)
2. C1, C2, C3, C4 –5, 4, 3, 2 pins (Arduino)

Step 3: Connecting the Servo Motor

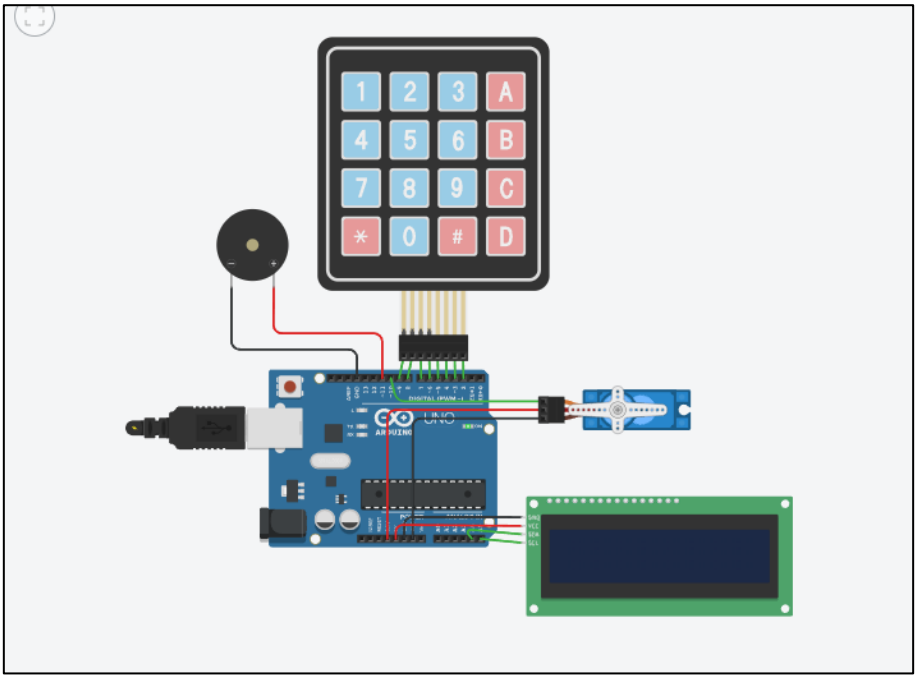
1. Signal (Yellow/Orange wire) – Pin 10 (Arduino)
2. VCC (Red wire) – 5V (Arduino)
3. GND (Black/Brown wire) – GND (Arduino)

Step 4: Connecting the Buzzer

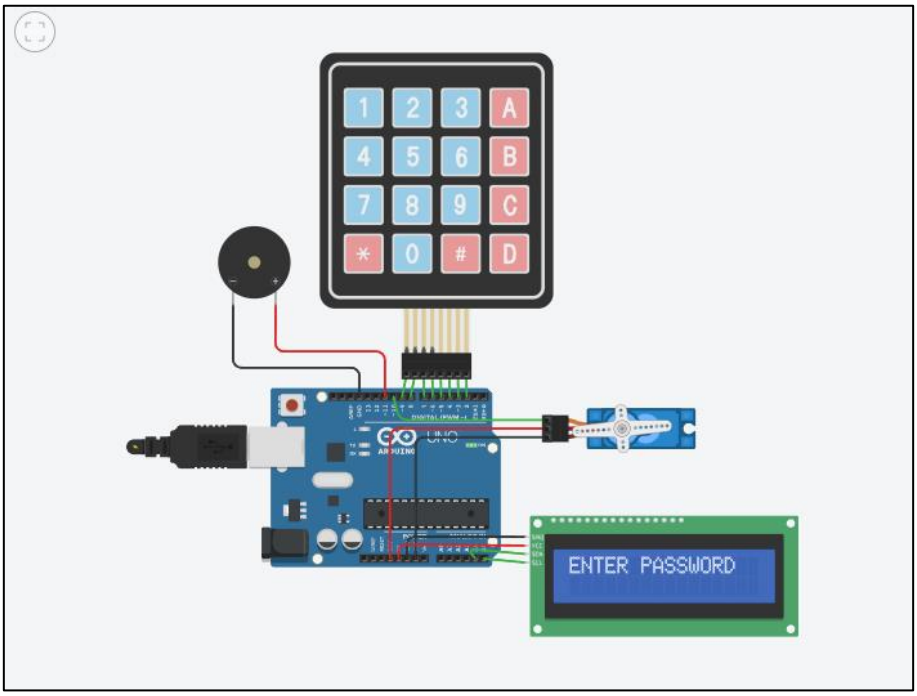
1. Positive (+) – Pin 11 (Arduino)
2. Negative (-) – GND (Arduino)

6. OUTPUT

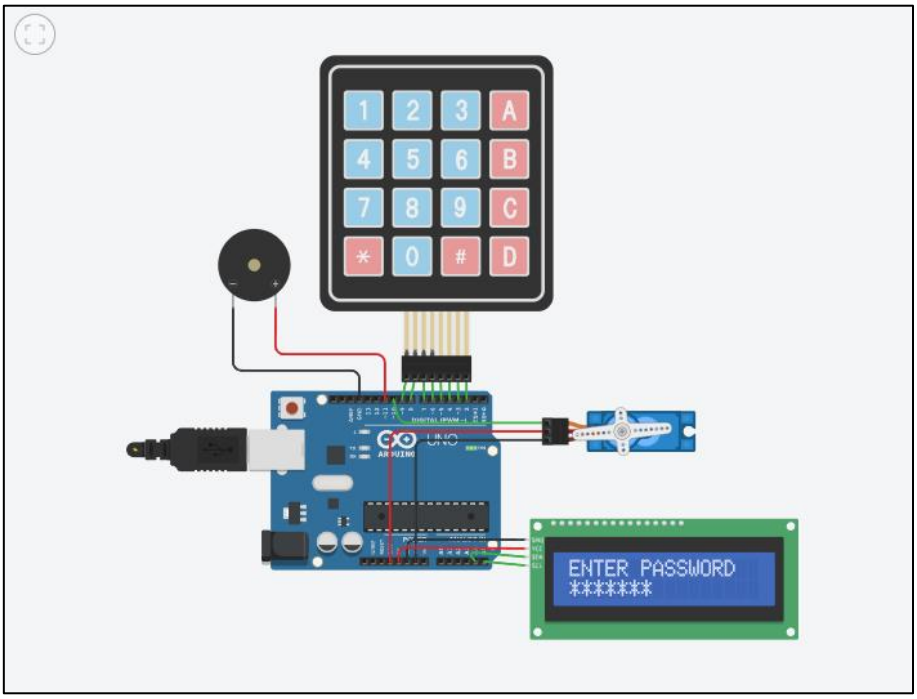
1. Run the simulation.



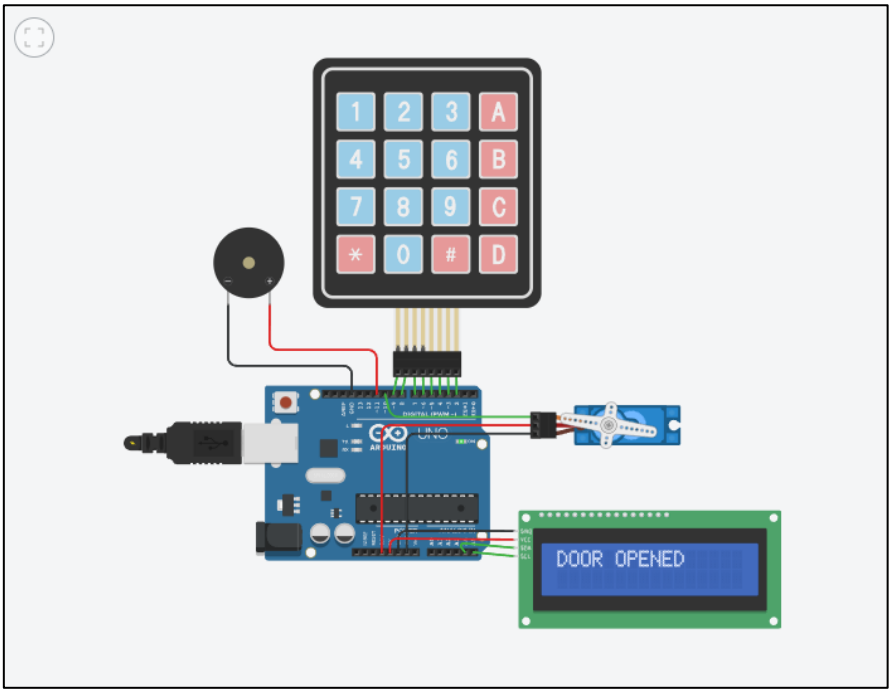
2. The LCD display shows “ENTER PASSWORD”.



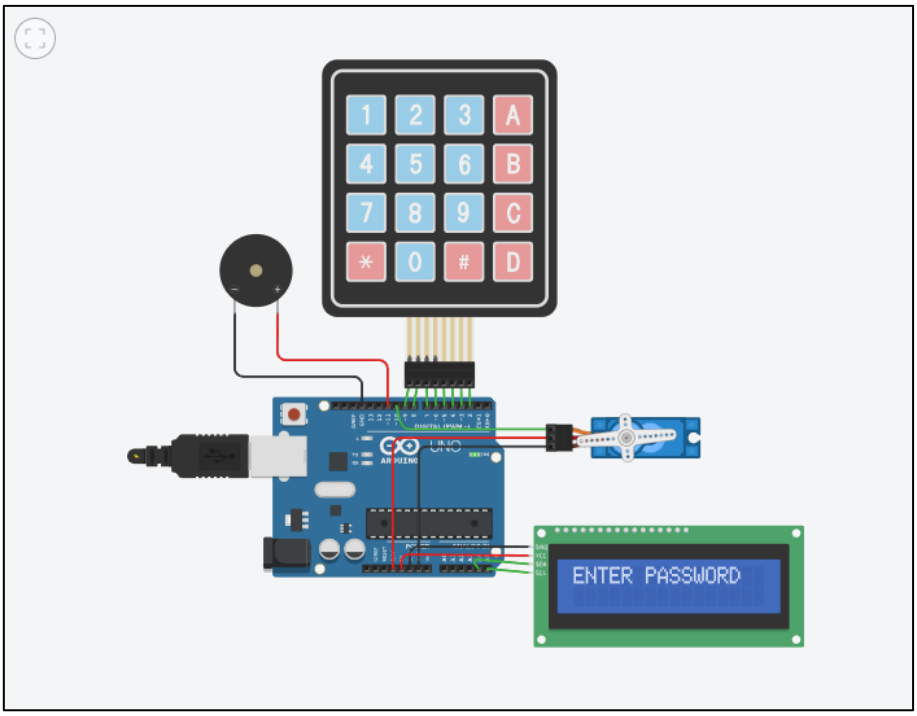
3. Type the password using keypad. Each input element is represented using asterisk (*).



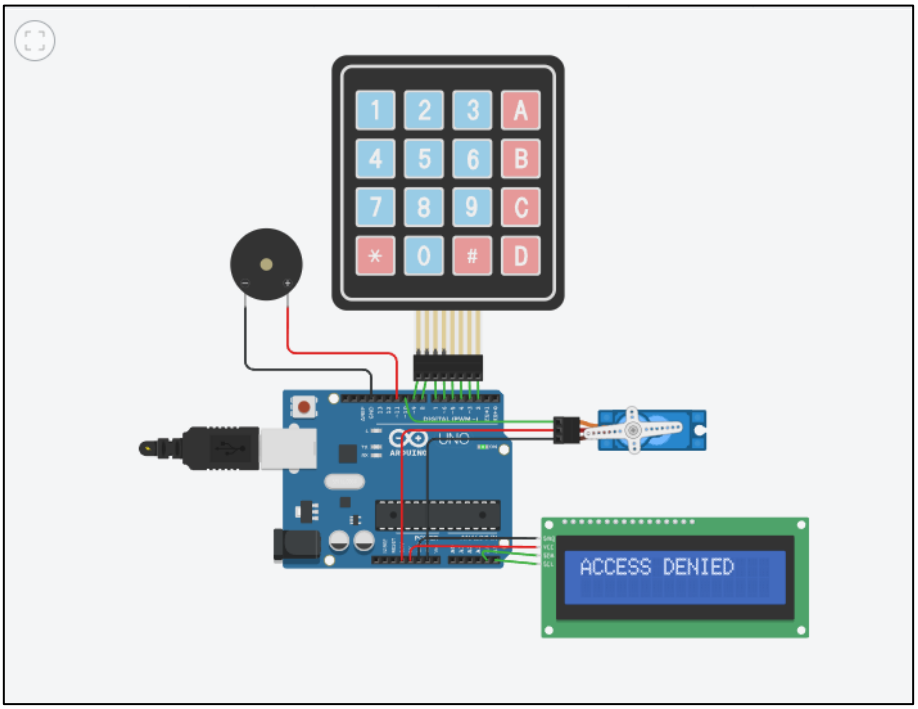
4. If password is correct, the LCD display shows “DOOR OPENED”. At that instant, the servo motor rotates 180°, unlocking the door.



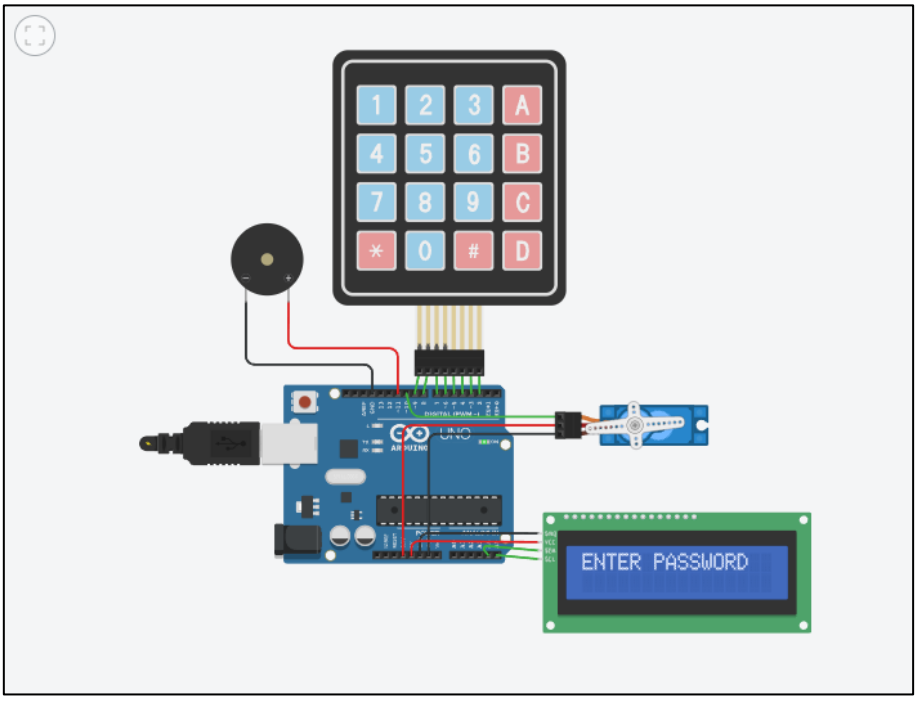
5. After 3 seconds the servo motor returns to 0°, locking the door. Then, the lcd displays “ENTER PASSWORD” again.



6. If password is incorrect, the LCD displays “ACCESS DENIED”.



7. After some time the LCD displays “ENETER PASSWORD” again.



7. APPLICATION

1. Home security
2. Office/ industries
3. Safety locks