بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيم

اسلامیہ یونیورسٹی بہاولپور

The Islamia University
of Bahawalpur

SEMESTER PROJECT 2023

Project Title :

Access Control System with Arduino and Servo Motor

Semester: 5th Semester

Department: bS Information Security

Session: 2021-2025

Islamia University of Bahawalpur,Bahawalnagar Campus

Represented to **Sir Qaiser Mahmood_**

-Mphil COmputer Science

Course: Wireless and Mobile Security

Muhammad Shamil Abbas

" Believe in yourself and all that you are. Know that there is something inside you that is greater than any obstacle."

"We would like to thanks to Sir Qaiser Mehmood who gave this opportunity that we become able to complete this project with the help of Allah Almighty"

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيم

Access Control System with Arduino and Servo Motor

Project Implementation , design & Documented
by **MUHAMMAD SHAMIL ABBAS**

## Historical Context and IoT Relevance

Arduino was created in 2003 by a group of students at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy to simplify electronics for non-experts.A group of students, including Massimo Banzi and David Cuartielles, aimed to create an accessible platform for rapid prototyping of electronic projects..Its open-source philosophy and user-friendly programming language and IDE made it accessible to a wide audience. Arduino's modularity, strong community, and global reach have had a significant impact on the maker and IoT communities. It continues to evolve and inspire both open-source and commercial products, supporting a wide range of applications and projects worldwide.The Arduino platform, pioneered by a group of visionary scientists and engineers, has played a significant role in democratizing electronics and fostering innovation. Its contribution to the world of technology

is undeniable. In the context of the Internet of Things (IoT), Arduino has emerged as a foundational tool for developing and prototyping connected devices. By simplifying hardware and software development, Arduino has empowered creators worldwide to craft smart, interconnected solutions that are transforming our modern world.

##Introduction

The "Access Control System with Arduino and Servo Motor" is a sophisticated project that showcases the integration of various electronic components to create a secure access control solution. This project employs an Arduino microcontroller, a 16x2 LCD display, a 4x4 Matrix Keypad, a servo motor, a buzzer, and LEDs to grant or deny access based on a pre-defined 4-digit password. When access is granted, the servo motor moves to a specific angle, visually indicating authorized entry. In the event of unauthorized access, the servo returns to its initial position, and an alarm sounds. This system is designed to enhance security and provide a user-friendly interface for both educational and practical applications.

##Components

The following components are integral to the success of this project:

Arduino Board: The project is built on an Arduino platform, such as the Arduino Uno, serving as the central control unit.

16x2 LCD Display: An LCD display with an I2C module is utilized for displaying system messages and feedback to the user.

4x4 Matrix Keypad: A keypad enables users to input a 4-digit password for access control.

Servo Motor: A servo motor is responsible for physically controlling access. It rotates to a specified angle when access is granted and returns to its initial position when denied.

Buzzer: A buzzer provides audible feedback to indicate access status.

Green and Red LEDs: These LEDs are used to signal access granted and access denied, respectively.

Jumper Wires: These wires are crucial for connecting the components and facilitating communication between them.

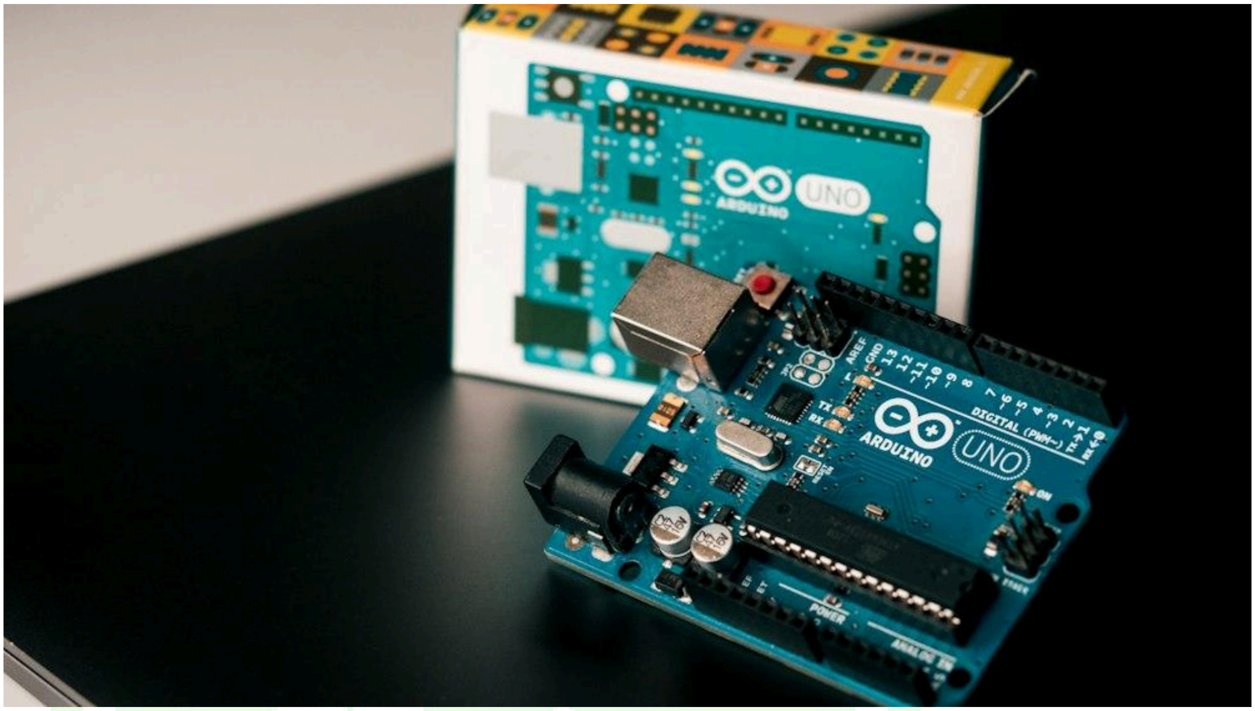
Power Source for Servo: In some cases, an external power source may be necessary to ensure the servo receives sufficient power.

##**Procedure**

Creating a keypad lock system using an Arduino, an S90 servo motor, an I2C 16x2 LCD, and some additional components like a buzzer and two LEDs (green and red) is a fun project. Here's a step-by-step procedure with example code to check and integrate each component.

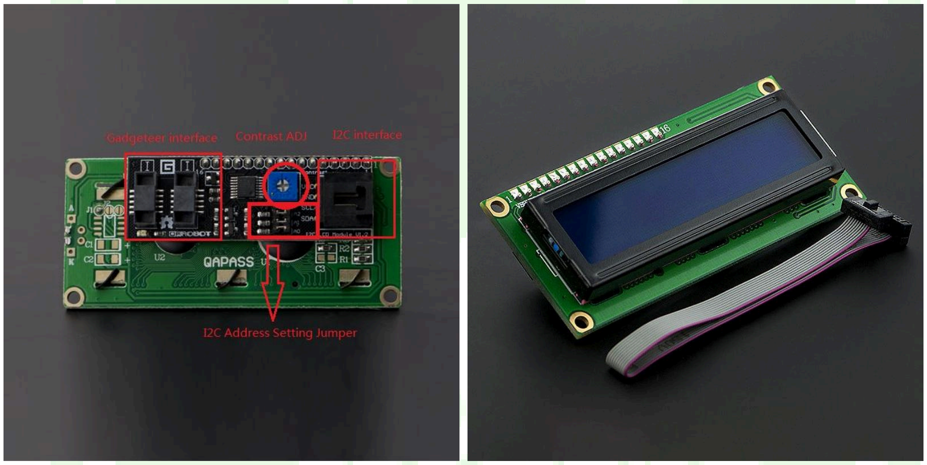
## Components Needed:

1. Arduino Uno

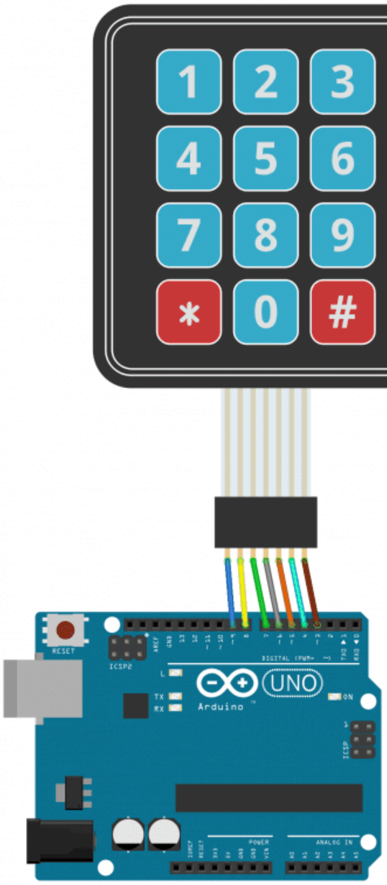


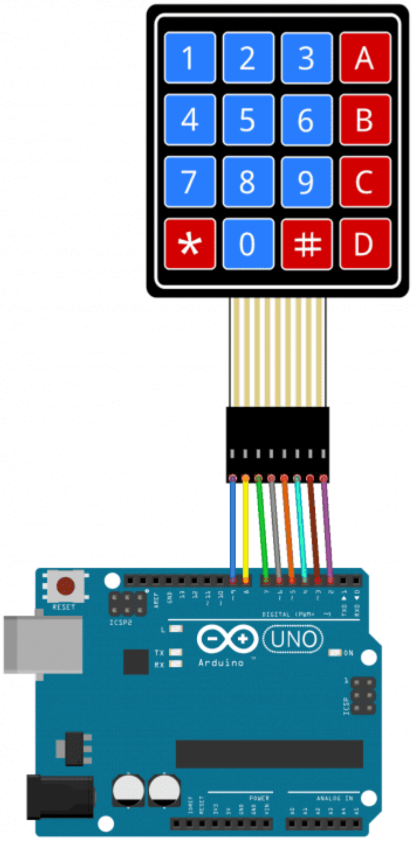
2. S90 Servo Motor

3. I2C 16x2 LCD

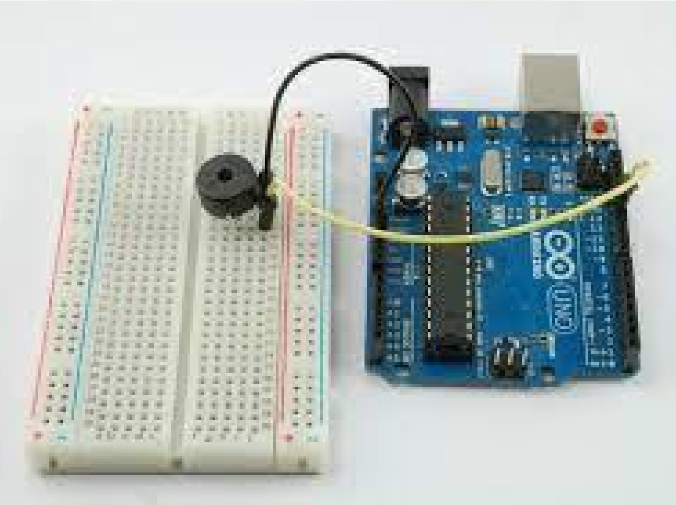


4. Keypad



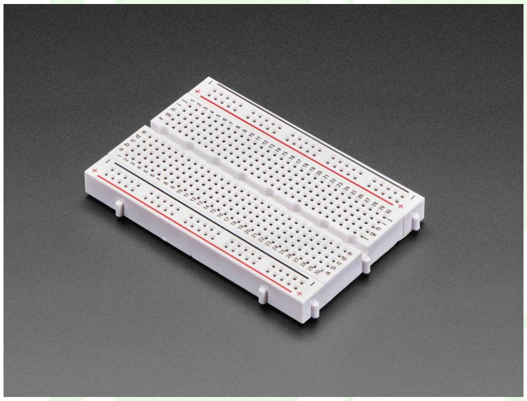


5. Buzzer

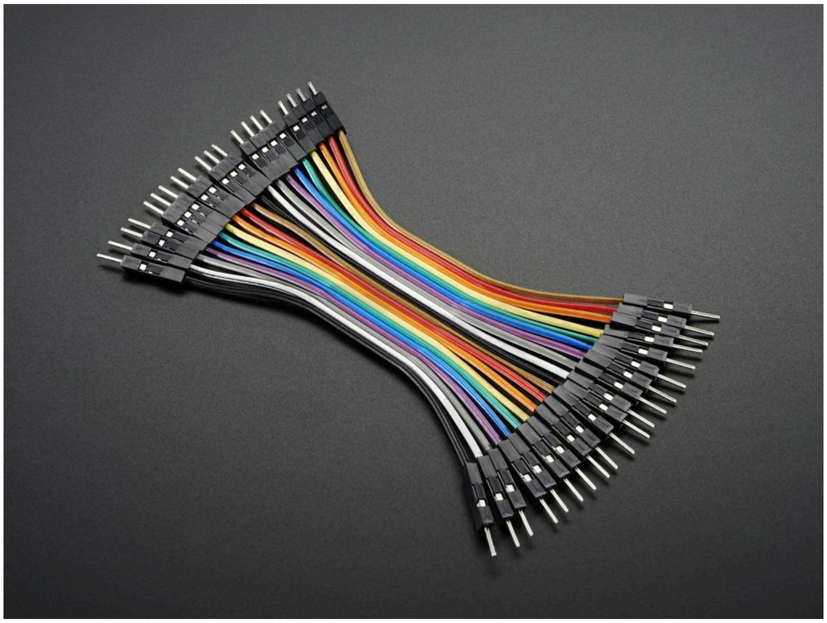


6. Green and Red LEDs

7. Small Breadboard



8. Jumper Wires



Step 1: Set Up the Circuit

- Connect the I2C LCD to the Arduino.

- Connect the servo motor to the Arduino.
- Connect the keypad, buzzer, green LED, and red LED to the Arduino.
- Ensure all components are powered appropriately.

Step 2: Install Required Libraries

- To use the I2C LCD, install the "LiquidCrystal_I2C" library from the Arduino Library Manager.
- To control the servo, no additional libraries are required.

Step 3: Check the LCD

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

LiquidCrystal_I2C lcd(0x27, 16, 2);  // Address may vary; use an I2C
scanner to find the correct one.
void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("LCD is working!");
}
void loop() { // Nothing to do here for now
}
```

Step 4: Test the Keypad

- Install the "Keypad" library.

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

LiquidCrystal_I2C lcd(0x27, 16, 2);
const byte ROWS = 4;
const byte 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 );
void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Enter Password:");
}
void loop() {
  char key = keypad.getKey();
  if (key) {
    lcd.setCursor(0, 1);
    lcd.print(key);   // Store key and check for correct password logic
  }
}
```

Step 5: Check the Buzzer

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

LiquidCrystal_I2C lcd(0x27, 16, 2);
 int buzzerPin = 10;

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
   lcd.print("Enter Password:");
   pinMode(buzzerPin, OUTPUT);
 }
void loop() {
  tone(buzzerPin, 1000);
  delay(500);
  noTone(buzzerPin);
  delay(500);
}
```

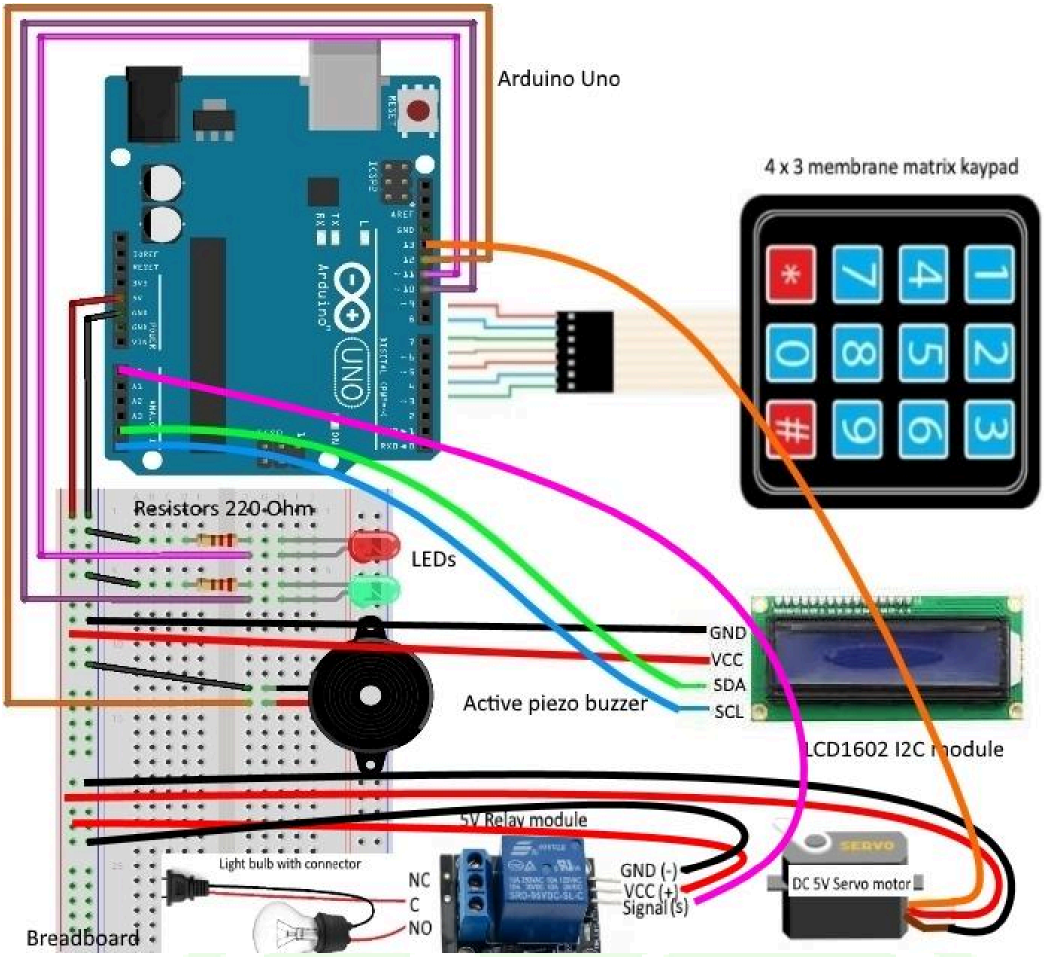
Step 6: Check the LEDs

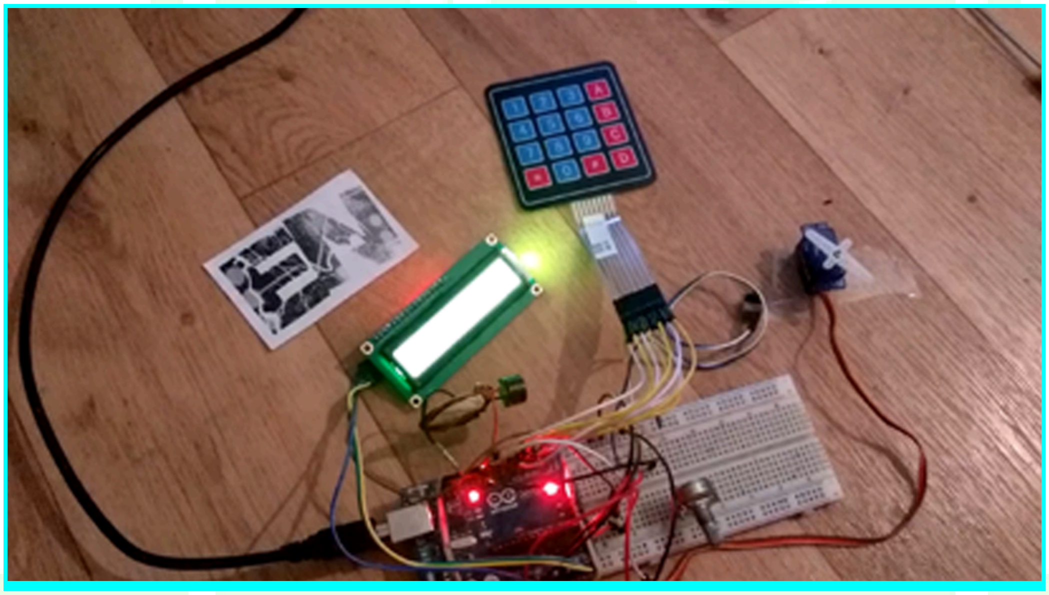
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
int greenLedPin = 11;
int redLedPin = 12;

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Leds Testing");
  pinMode(greenLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);
}

void loop() {
  digitalWrite(greenLedPin, HIGH);
  delay(1000);
  digitalWrite(greenLedPin, LOW);
  delay(1000);
  digitalWrite(redLedPin, HIGH);
  delay(1000);
  digitalWrite(redLedPin, LOW);
  delay(1000);
}
```

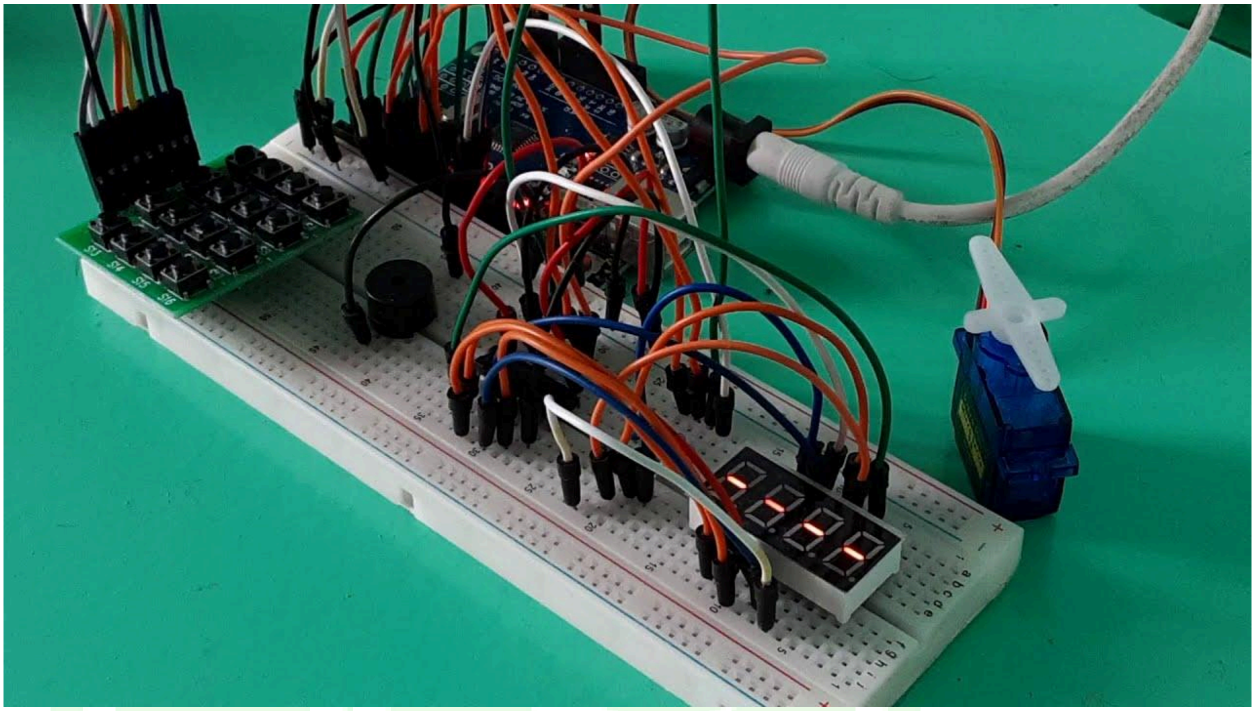

Arduino Uno
4 x 3 membrane matrix kaypad
Resistors 220 Ohm
LEDs
GND
VCC
SDA
SCL
Active piezo buzzer
LCD1602 I2C module
5V Relay module
Light bulb with connector
NC
C
NO
GND (-)
VCC (+)
Signal (s)
DC 5V Servo motor
Breadboard



Step 7: Integrate All Components

Component	Arduino Pin	Breadboard	Notes
I2C LCD (SDA)	A4 (analog 4)	Connect to Breadboard for Power and Ground	SDA is data, connect GND, VCC, and SDA
I2C LCD (SCL)	A5 (analog 5)	Connect to Breadboard for Power and Ground	SCL is clock, connect GND, VCC, and SCL
Keypad Rows (4 rows)	6, 7, 8, 9	Connect to Breadboard for Ground	Rows are connected to Arduino pins 6, 7, 8, and 9
Keypad Columns (3 cols)	2, 3, 4	Connect to Breadboard for Pull-up Resistors	Columns are connected to Arduino pins 2, 3, and 4
Green LED	10	Connect to Breadboard for Resistor and Ground	Anode (+) to Arduino pin 10, Cathode (-) to Ground
Red LED	11	Connect to Breadboard for Resistor and Ground	Anode (+) to Arduino pin 11, Cathode (-) to Ground
Buzzer	12	Connect to Breadboard for Resistor and Ground	Connect Buzzer positive (+) to Arduino pin 12, negative (-) to Ground
Servo Motor	13	Connect to Breadboard for Power and Ground	Servo signal to Arduino pin 13, connect GND, VCC for power
Breadboard Power	5V, GND	Connect to Arduino for Power | 5V to 5V, GND to GND for powering components	5V to 5V, GND to GND for powering component

These connections should help you set up the keypad lock system with the mentioned components. Make sure to add resistors where needed for the LEDs and set up the servo and other components according to the code and the components' datasheets.



- Combine the codes and logic to create your keypad lock system. You'll need to add the password logic and servo control for the door lock/unlock functionality. Use the Servo library to control the servo motor.
- Make sure to replace **"YourPassword"** with your desired password.

—FULL CODE—

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

LiquidCrystal_I2C lcd(0x27, 16, 2);  // I2C address 0x27, 16x2 LCD
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Four columns

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}; // Connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 3}; // Connect to the column pinouts of the keypad

int greenLED = 10;  // Digital pin for the green LED
int redLED = 11;    // Digital pin for the red LED
int buzzerPin = 12; // Digital pin for the buzzer
Servo servo;        // Create a servo object

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
char password[] = "1234"; // Set your 4-digit password here
```

```
boolean accessGranted = false;

void setup() {
  lcd.init();                    // Initialize the LCD
  lcd.backlight();                 // Turn on the backlight
  lcd.setCursor(0, 0);
  lcd.print("Enter code:");

  pinMode(greenLED, OUTPUT); // Set green LED pin as an
output
  pinMode(redLED, OUTPUT);   // Set red LED pin as an output
  pinMode(buzzerPin, OUTPUT); // Set buzzer pin as an output

  servo.attach(13); // Attach the servo to digital pin 13
  servo.write(0);   // Initialize the servo at 0 degrees

  Serial.begin(9600);
}

void loop() {
  char inputPassword[5] = "    "; // Initialize an empty input password
  int digitCount = 0;

  while (digitCount < 4) {
    char key = keypad.getKey();
    if (key) {
      lcd.setCursor(digitCount, 1);
      lcd.print("*"); // Display an asterisk for each digit entered
      inputPassword[digitCount] = key;
      digitCount++;
      delay(100); // Debounce
    }
  }
  if (strcmp(inputPassword, password) == 0) {
    lcd.clear();
    lcd.print("Access Granted");
    digitalWrite(greenLED, HIGH); // Turn on the green LED
    accessGranted = true;
    // Slowly rotate the servo to 90 degrees (or any desired angle)
    for (int pos = 0; pos <= 90; pos++) {
      servo.write(pos);
      delay(15); // Adjust the delay to control the speed
    }
    delay(3000); // Additional delay
    lcd.clear();
    lcd.print("Enter code:");
    digitCount = 0; // Reset the digit count for the next attempt
    memset(inputPassword, ' ', sizeof(inputPassword)); // Clear the input password
    digitalWrite(greenLED, LOW); // Turn off the green LED
  } else {
    if (accessGranted) {
      // Slowly rotate the servo back to 0 degrees
      for (int pos = 90; pos >= 0; pos--) {
        servo.write(pos);
        delay(15); // Adjust the delay to control the speed
      }
      accessGranted = false;
    }
    lcd.clear();
    lcd.print("Access Denied");
    digitalWrite(redLED, HIGH); // Turn on the red LED
    digitalWrite(buzzerPin, HIGH); // Turn on the buzzer
    delay(3000); // Buzzer duration
    digitalWrite(buzzerPin, LOW); // Turn off the buzzer
  delay(5000); // Additional delay
    lcd.clear();
    lcd.print("Enter code:");
    digitCount = 0; // Reset the digit count for the next attempt
    memset(inputPassword, ' ', sizeof(inputPassword)); // Clear the input password
    digitalWrite(redLED, LOW); // Turn off the red LED
  }
}
```

## Code Overview

The Arduino code underpinning this project governs various crucial aspects, including:

Initialization: The code initializes the LCD, keypad, servo motor, and relevant pins.

Password Definition: A 4-digit password is predefined as a variable.

User Interaction: The system awaits user input via the keypad.

Access Control: When the correct password is entered, access is granted. The servo motor rotates to 90 degrees, and the green LED illuminates.

Unauthorized Access: In the case of an incorrect password, access is denied. The servo reverts to 0 degrees, the red LED signals denial, and the buzzer emits a brief alarm.

## Key Benefits

1. Enhanced Security:

The system bolsters security by enforcing a password-based access control mechanism, effectively preventing unauthorized entry. making it suitable for locking doors, cabinets, or other access points.

2. User-Friendly Interface:

The integration of a keypad and LCD display ensures a seamless user experience, allowing individuals to input their passwords and receive clear access status feedback.

3. Customization:

The system is highly adaptable, allowing for easy adjustments to the password or the incorporation of advanced features, including remote monitoring or integration into larger IoT projects.

4. Learning Opportunity:

Building this project helps you learn and practice electronics, coding, and the integration of different components, enhancing your knowledge and skills.

5. Low-Cost Solution:

This project can be implemented with relatively low-cost components, making it an affordable option for DIY security.

6. **Control of Electronic Locks:**

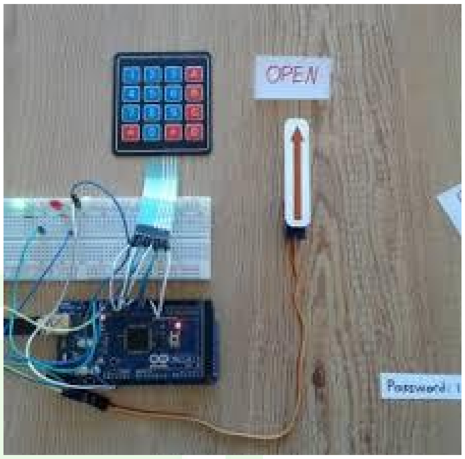
By replacing the servo with an electronic lock or relay, you can adapt the system to control various types of locks, making it suitable for a wide range of applications.

7. **Hands-On Experience:**

You gain hands-on experience with Arduino, a popular microcontroller platform, which can be useful for various future projects and applications.

8. Educational Tool:

This project serves as a valuable educational resource, aiding newcomers in grasping fundamental concepts related to electronics and programming.



9. IoT Integration:

With suitable IoT extensions, this system can seamlessly integrate into modern smart homes and offices, contributing to advanced control and monitoring capabilities.

Overall, this project combines practicality and educational value, making it a rewarding endeavor for individuals interested in electronics, programming, and DIY security solutions.

## Conclusion

The "Access Control System with Arduino and Servo Motor" project stands as a testament to the sophistication and versatility of the Arduino platform. It underscores the simplicity and educational value of Arduino while offering a glimpse into its potential to contribute to advanced security solutions in our increasingly interconnected and smart modern environments.

"Thank you for considering this project! Remember, every step you take in learning and creating brings you closer to your goals. Keep building, keep learning, and never stop exploring the possibilities. The journey is as important as the destination."

⭐

"The only way to do great work is to love what you do." - Steve Jobs