



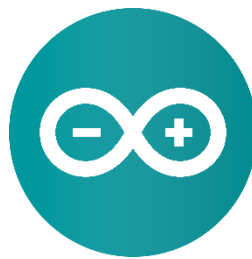
2ECOE53 ARDUINO FOR ENGINEERS
SPECIAL ASSIGNMENT

Arduino Door Lock System with Servo Motor

Submitted By

20BCE016 Arunima Barik

20BCE079 Gaurav Golchha



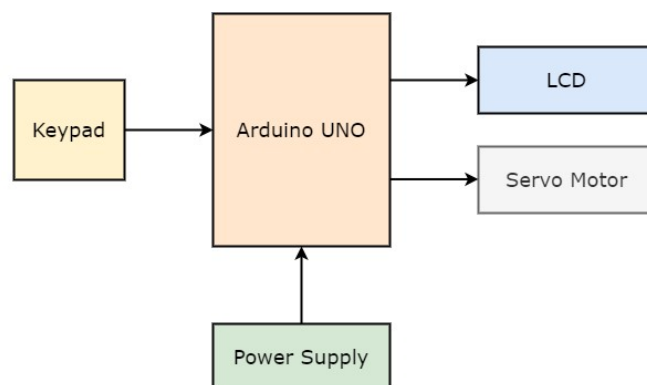
About the project

The Arduino Uno Door Lock System, featuring a servo motor, offers a straightforward and reliable solution for enhancing security in your home or workspace. This project provides a basic yet effective approach to door access control by implementing a password-protected mechanism.

Features:

1. **Password Protection:** With this system, access to your space is protected by a password. Users must enter the correct combination on a keypad to unlock the door, adding a layer of security.
2. **Servo Motor Control:** The servo motor operates the locking mechanism, ensuring smooth and precise door locking and unlocking.

Block Diagram

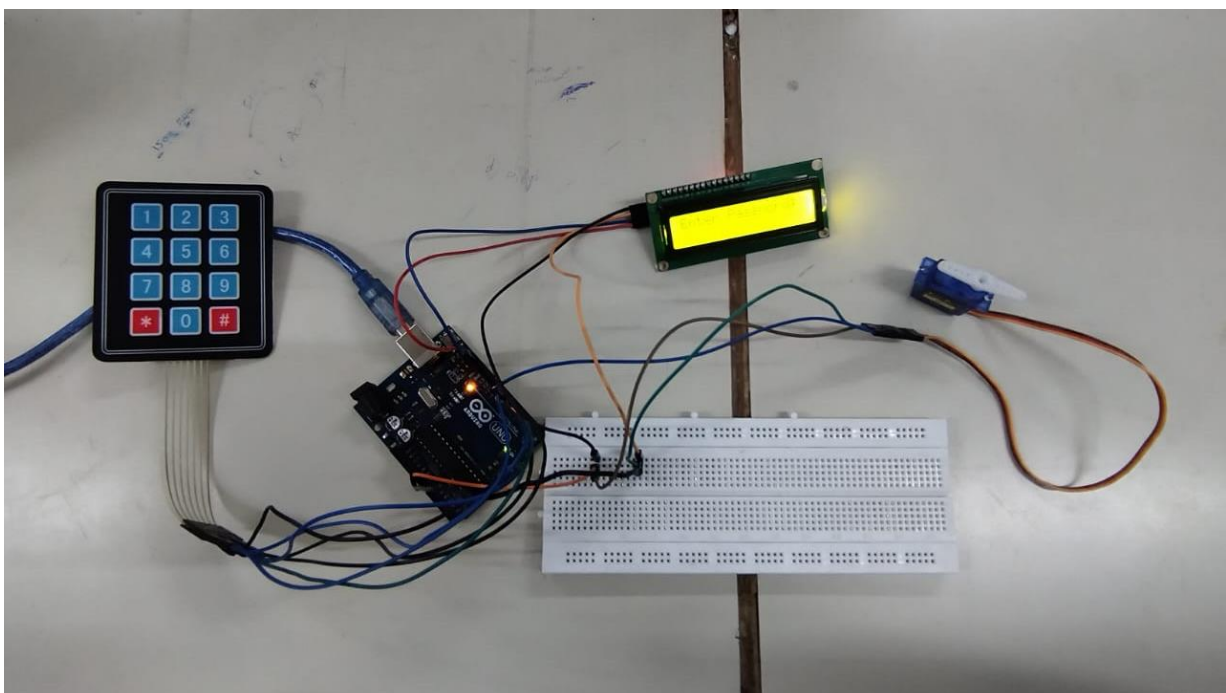
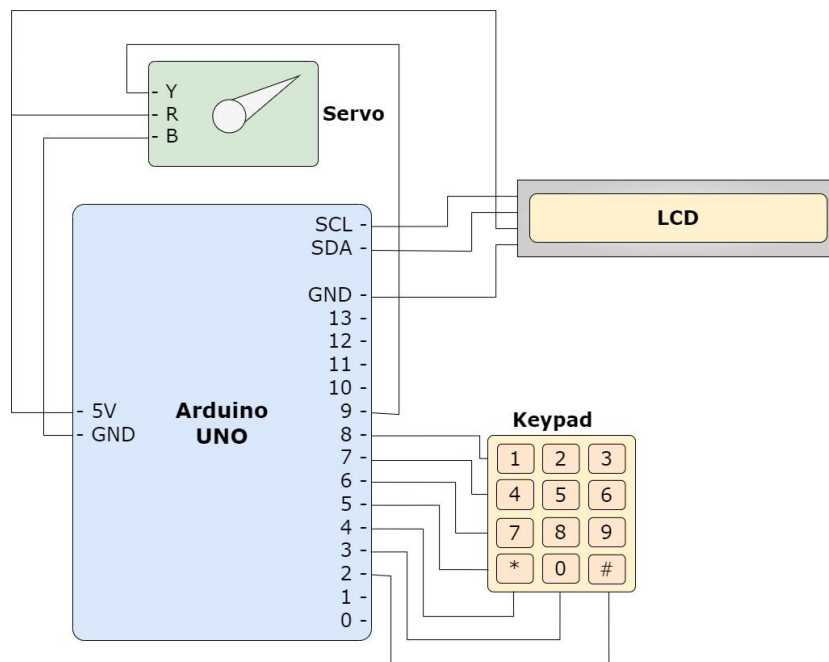


Components used:

1. **Arduino Uno Board:** Manages the operation of the servo motor, read input from the keypad and controls the LCD display. It processes the entered password and executes the locking/unlocking actions.
2. **Servo Motor:** Physically controls the locking mechanism of the door. It rotates to engage or disengage the lock when the correct password is entered.
3. **LCD (16 X 2 | 5V):** Displays relevant information, including system status, prompts for entering the password and notifications such as "Locked" or "Unlocked".

4. Keypad (4 X 3): Input the password required to unlock the door.
5. Jumper Wires: Establish electrical connections between the various components.
6. Breadboard: Used for prototyping and connecting the components without soldering.

Circuit Diagram



Code

```
#include <Servo.h>
#include <Keypad.h>
#include <Wire.h> // Include the Wire library for I2C communication
#include <LiquidCrystal_I2C.h> // Include the I2C LCD library

Servo doorServo;
LiquidCrystal_I2C lcd(0x27, 16, 2); // Change the address (0x27) to match your LCD's
address

const int servoPin = 9;
const int numRows = 4;
const int numCols = 3;

char keypadKeys[numRows][numCols] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte rowPins[numRows] = {8, 7, 6, 5}; //connect to the row pins (R0-R3) of the keypad
byte colPins[numCols] = {4, 3, 2}; //connect to the column pins (C0-C2) of the keypad

char password[] = "1234"; // Change this to your desired password
char enteredPassword[5]; // Store the entered password
int passwordIndex = 0;

const int unlockAngle = 90; // Angle to unlock the door
const int lockedAngle = 0; // Angle to lock the door

Keypad keypad = Keypad(makeKeymap(keypadKeys), rowPins, colPins, numRows, numCols);

void setup() {
  doorServo.attach(servoPin);
  lcd.init(); // Initialize the LCD
  lcd.backlight(); // Turn on the backlight
  lcd.setCursor(0, 0);
  lcd.print("Enter Password:");
  lcd.setCursor(0, 1);
  lcd.print(">");
}

void loop() {
  char key = getKey();

  if (key != NO_KEY && key != '#') {
    enteredPassword[passwordIndex++] = key;
    lcd.setCursor(passwordIndex, 1);
    lcd.print('*');
  }
}
```

```

if (key == '#') {
    enteredPassword[passwordIndex] = '\0'; // Null-terminate the entered password
    if (strcmp(enteredPassword, password) == 0) {
        unlockDoor();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Unlock");
        while(getKey() != '*');
        lcd.print("Locking");
        lockDoor();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Enter Password:");
    } else {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Wrong Password");
        delay(2000); // Display "Wrong Password" for 2 seconds
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Enter Password:");
    }
    clearPassword();
}
}

char getKey() {
    char key = keypad.getKey();
    return key;
}

void unlockDoor() {
    doorServo.write(unlockAngle);
}

void lockDoor() {
    doorServo.write(lockedAngle);
}

void clearPassword() {
    passwordIndex = 0;
    memset(enteredPassword, 0, sizeof(enteredPassword));
    lcd.setCursor(0, 1);
    lcd.print("                "); // Clear the password display
    lcd.setCursor(0, 1);
    lcd.print(">");
}

```

Applications

1. Home Security: Enhance the security of your home by implementing this password-protected door lock on your front or back door.
2. Small Office Access Control: Improve access control in a small office or workspace, allowing authorized personnel entry while keeping unauthorized individuals out.
3. Guest Room or Private Space: Install it on a guest room door or a private space within your home to control access and provide additional security.
4. Temporary Access Control: Use it in situations where you need temporary access control, like safeguarding a room during renovations or securing a storage area.

Future expansion

1. Remote Control: Add remote control capabilities, allowing users to lock or unlock the door via a smartphone app or a web interface. This would involve integrating wireless communication modules like Wi-Fi or Bluetooth.
2. Biometric Authentication: Enhance security by adding biometric authentication methods such as fingerprint or facial recognition scanners.
3. Multiple User Profiles: Create a system that supports multiple user profiles with different access permissions. This could be beneficial for households or offices with multiple occupants.
4. Smart Home Integration: Integrate the door lock system with smart home platforms like Amazon Alexa or Google Home for seamless control and automation.

Limitations

1. **Security Concerns:** May not be suitable for high-security applications. More robust encryption and authentication methods would be required for critical security needs.
2. **Maintenance:** Mechanical components like the servo motor may require periodic maintenance to ensure smooth operation.
3. **Scalability:** Expanding the system to support a larger number of users or doors may require a significant redesign and additional hardware.