|  |  |
| --- | --- |
| **Iotronics Techlab Pvt Ltd.** | |
| **Aim:** | To design and implement a door lock system |
| **Requirements:** | AutomationKit, Mechatronics kit, Wireless & iot kit and Power and Control |
| **IDE:** | Arduino IDE |
| **Connection Diagram:** |  |
| **Working** | 1. The user inputs a password through the 4x4 keypad. 2. The Arduino Uno reads the input from the keypad. 3. If the entered password matches the predefined one, it triggers the servo motor to unlock the door. 4. If the password is incorrect, nothing happens. 5. The system resets after a predefined timeout period. |
| **Procedure:** | **1.Connect the Servo Motor:**   * Connect the control wire of the servo motor to a PWM pin on the Arduino Uno (e.g., pin 9). * Connect the power and ground wires of the servo motor to the 5V and GND pins on the Arduino Uno, respectively.   **2.Connect the 4x4 Keypad:**   * Connect the rows and columns of the keypad to digital pins on the Arduino Uno. * Refer to the datasheet of the keypad for pin mappings.   **3.Writing the Code:**   * Open the Arduino IDE. * Write code to read input from the 4x4 keypad. * Define a password for the door lock system. * Compare the entered password with the predefined one. * If the password matches, trigger the servo motor to unlock the door.   **4.Uploading and Testing:**   * Upload the code to the Arduino Uno. * Test the system by entering the correct and incorrect passwords. * Ensure that the servo motor behaves as expected based on the input. |
| **CODE:** | #include <Keypad.h>  #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  #include <Servo.h>  Servo myservo;  LiquidCrystal\_I2C lcd(0x27, 16, 2); // Change the address if necessary  #define Password\_Lenght 7 // Give enough room for six chars + NULL char  int pos = 0;    // variable to store the servo position  char Data[Password\_Lenght]; // 6 is the number of chars it can hold + the null char = 7  char Master[Password\_Lenght] = "123456";     //Change PASSWORD here  byte data\_count = 0, master\_count = 0;  bool Pass\_is\_good;  char customKey;  const byte ROWS = 4;  const byte COLS = 3;  char keys[ROWS][COLS] = {    {'1', '2', '3'},    {'4', '5', '6'},    {'7', '8', '9'},    {'\*', '0', '#'}  };  bool door = true;  byte rowPins[ROWS] = {64, 62, 68, 66}; //connect to the row pinouts of the keypad  byte colPins[COLS] = {65, 63, 69, }; //connect to the column pinouts of the keypad  Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS); //initialize an instance of class NewKeypad  void setup()  {    Wire.begin();    lcd.init();    lcd.backlight();    myservo.attach(14);    ServoClose();  }  void loop()  {    if (door == 0)    {      customKey = customKeypad.getKey();      if (customKey == '#')      {        lcd.clear();        lcd.setCursor (3,0);        lcd.print("LOCKING");        delay (700);        lcd.setCursor (10,0);        lcd.print(".");        delay (800);        lcd.setCursor (11,0);        lcd.print(".");        delay (800);        lcd.setCursor (12,0);        lcd.print(".");        delay (500);        lcd.clear();        delay (500);        ServoClose();        lcd.setCursor (5,0);        lcd.print("LOCKED");        delay (1700);        door = 1;      }    }    else Open();  }  void clearData()  {    while (data\_count != 0)    { // This can be used for any array size,      Data[data\_count--] = 0; //clear array for new data    }    return;  }  void ServoOpen()  {    for (pos = 180; pos >= 0; 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);                       // waits 15ms for the servo to reach the position    }  }  void ServoClose()  {    for (pos = 0; pos <= 180; pos += 5) { // goes from 180 degrees to 0 degrees      myservo.write(pos);              // tell servo to go to position in variable 'pos'      delay(15);                       // waits 15ms for the servo to reach the position    }  }  void Open()  {    lcd.setCursor(1, 0);    lcd.print("INPUT PASSWORD");    customKey = customKeypad.getKey();    if (customKey) // makes sure a key is actually pressed, equal to (customKey != NO\_KEY)    {      lcd.setCursor(0, 1);      lcd.print("\*\*\*\*\*\*");     //To hide your PASSWORD, make sure its the same lenght as your password      Data[data\_count] = customKey; // store char into data array      lcd.setCursor(data\_count, 1); // move cursor to show each new char      lcd.print(Data[data\_count]); // print char at said cursor      data\_count++; // increment data array by 1 to store new char, also keep track of the number of chars entered    }    if (data\_count == Password\_Lenght - 1) // if the array index is equal to the number of expected chars, compare data to master    {      if (!strcmp(Data, Master)) // equal to (strcmp(Data, Master) == 0)      {        lcd.clear();        ServoOpen();        lcd.print(" ACCESS GRANTED");        lcd.setCursor(0,1);        lcd.print("press # to close");        door = 0;      }      else      {        lcd.clear();        lcd.setCursor(1,0);        lcd.print("ACCESS DENIED!");        delay(1500);        door = 1;      }      clearData();    }  } |
| **Result/Output** |  |