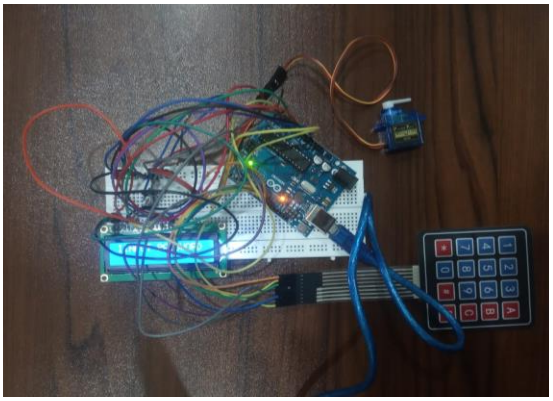
Code:

```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>

Servo myservo;

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

#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 = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
bool door = true;

byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts
of the keypad
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column
pinouts of the keypad

Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS,
COLS); //initialize an instance of class NewKeypad

void setup()
{
  myservo.attach(10);
  ServoClose();
  lcd.begin(16, 2);

}

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();
  }
}
```

Circuit:



THE END