דו"ח תרגיל 3 – ARDUINO

קבוצה 22 - ת.ז : 206560740, 208687897, 318402930 , 313293219

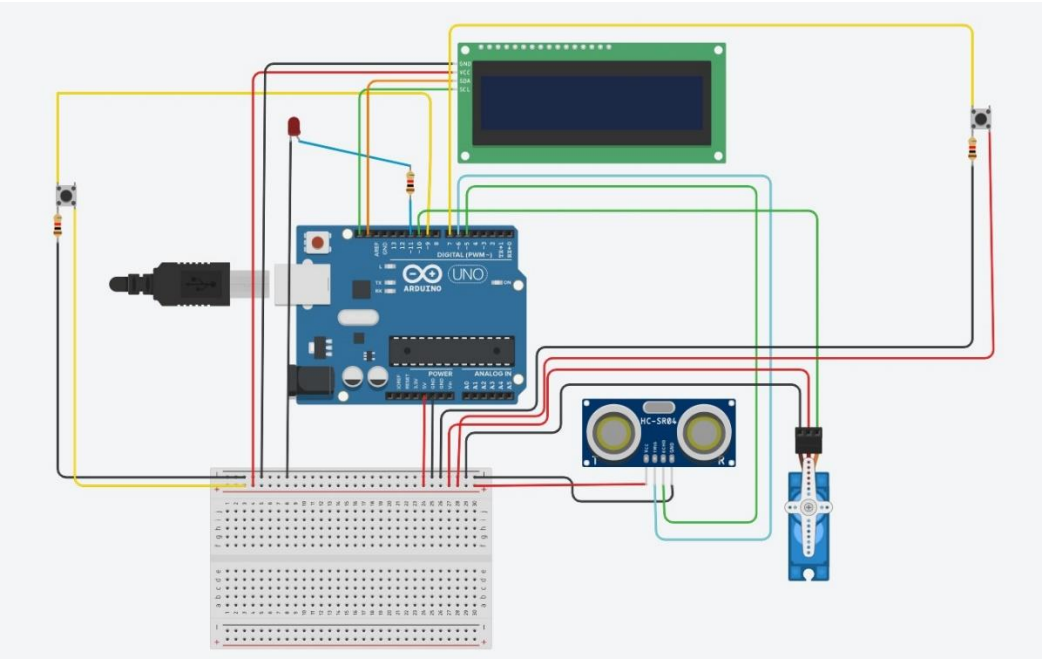
1. מטרת המערכת ותיאור משימת הבקרה:

במלון היוקרה של אוניברסיטת בן גוריון, כל אורח נהנה מחוויית שהייה ייחודית וטכנולוגית. אנו דואגים לפרטיות ונוחות מקסימלית בעזרת מערכות חכמות המאפשרות לכם לשלוט במתקני החדר בלחיצת כפתור וללא הפרעות מיותרות.

המערכת כוללת:

- **מסך LCD -** ממוקם מחוץ לחדר ומציג בזמן אמת את מצב החדר (תפוס/פנוי) כדי למנוע הפרעות מצד צוות המלון.
- **חיישן אולטראסוניק** מזהה נוכחות בחדר ומעדכן את המסך בהתאם. כשהחיישן מזהה שיש מישהו בחדר, המסך יראה שהחדר תפוס.
- **נורת LED** נדלקת לפי הצורך בלחיצת כפתור, מספקת תאורה מותאמת אישית לכל רגע.
- **מנוע SERVO -** שולט על הווילונות, נפתח ונסגר בהתאם לפקודות שהתקבלו מהכפתור, מבטיח שליטה מלאה על כמות האור הנכנסת לחדר.

2. תיאור התכן הכולל:



הפרויקט מומש באמצעות בקר ארדואינו. התכן (המתואר בטבלה למטה) נעשה בעזרת תוכנת הסימולטור TINKERCARD מבית AutoDesk. תיעוד הקוד מפורט בנספח למסמך זה.

3. תיאור הרכיבים:

מספר הפין	שם המשתנה	תצורה	הפעלה
10	מנוע Servo	OUTPUT	אחראי על פתיחת וסגירת הווילון. מקבל פקודות בהתאם ללחיצה על כפתור.
11	נורת LED	OUTPUT	נדלקת או נכבית בהתאם ללחיצה ידנית על כפתור.
6	סנסור Ultrasonic-TrigPin	OUTPUT	שולח אות לחיישן לזיהוי מרחק.
5	סנסור Ultrasonic-EchoPin	INPUT	מקבל את האות החוזר מהחיישן לקביעת המרחק.
7	כפתור להפעלת המנוע	INPUT	מאפשר הפעלה ועצירה ידנית של המנוע.
9	כפתור הפעלת הנורה	INPUT	מאפשר הדלקה וכיבוי ידני של נורת הLED
SDA	מסך LCD- SDA		חיבור להעברת נתונים.
SCL	מסך LCD- SCL		חיבור להעברת שעון.

4. תיאור הבדיקות שבוצעו:

מטרת הבדיקות הייתה לוודא שהמערכת פועלת בצורה תקינה ועומדת בדרישות התכן. הבדיקות כללו בדיקות חומרה ובדיקות לוגיקה של המערכת.

בדיקות שבוצעו:

בדיקת לוגיקה-

1. **בדיקת חיישן מרחק (אולטראסוניק) ומסך LCD:** בדקנו האם כאשר החיישן מזהה נוכחות בטווח של 10 ס"מ ומטה, מודפס במסך ה LCD שהחדר תפוס ולהיפך.ביצענו בדיקה זו על מנת לוודא שהחיישן והמסך מזהים נוכחות בחדר בצורה מדויקת ומעודכנת ומוציאים פלט כפי שכתבנו בקוד. לצורך בדיקה זו קיבענו מרחק של 10 ס"מ וביצענו 14 תצפיות, בכל תצפית קירבנו והרחקנו עצם לחיישן ובדקנו מה הפלט על המסך ומתי הוא מופיע בהתאם למרחק 10 ס"מ.

בדיקת חומרה-

2. **בדיקת מנוע Servo:** בדקנו את משך סיבוב המנוע לצורך ווידוא שהרכיב עובד כראוי ומסתובב במשך 3 שניות כל פעם. ביצענו בדיקה זו לצורך בדיקת אמינות החומרה וכדי להבטיח פעולה מדויקת ויעילה של הווילונות. ביצענו 10 דגימות באמצעות סטופר ומדדנו את

משך סיבוב המנוע. ביצענו על התצפיות מבחן סטטיסטי כשהשערת האפס הינה שהזמן מתפלג אחיד רציף עם תוחלת של 3 שניות.

5. תיאור תוצאות הבדיקות:

בדיקת חיישן מרחק (אולטראסוניק) ומסך LCD:

מספר תצפית	מרחק חיישן (ס"מ)	פלט על מסך LCD	האם הפלט נכון?
1	מעל 10	Room Empty	V
2	מתחת ל- 10	Room Occupied	V
3	מעל 10	Room Empty	V
4	מתחת ל- 10	Room Occupied	V
5	מעל 10	Room Empty	V
6	מתחת ל- 10	Room Empty	X
7	מעל 10	Room Empty	V
8	מתחת ל- 10	Room Occupied	V
9	מעל 10	Room Empty	V
10	מתחת ל- 10	Room Occupied	V
11	מעל 10	Room Empty	V
12	מתחת ל- 10	Room Occupied	V
13	מעל 10	Room Empty	V
14	מתחת ל- 10	Room Occupied	V
15	מעל 10	Room Empty	V

בדיקת מנוע Servo

מספר תצפית	משך סיבוב המנוע (מאיות השניה:שניות)
1	3:07
2	2:57
3	3:01
4	2:59
5	3:10
6	2:58
7	3:03
8	2:55
9	2:55
10	3:00

תוצאות הבדיקה הסטטיסטית:

```
T-Statistic: 0.31943828249996997
P-Value: 0.7566843129174481
אין דחיית השערת האפס: אין הבדל משמעותי בין משך הזמן שנמדד לתוחלת של 3 שניות.
```

נקבל את השערת האפס ברמת מובהקות 0.05 ונאמר כי אין סטייה משמעותית בין משך סיבוב המנוע לבין 3 שניות.

המערכת פועלת בצורה אמינה ועקבית, עם תוצאות סטטיסטיות המראות על תקינות ואמינות גבוהה של רכיבי החומרה. המבחנים הראו שאין בעיות משמעותיות בתקשורת בין הרכיבים ובביצוע הפקודות השונות, מה שמעיד על תכנון נכון וביצוע מדויק של המערכת.

6. מסקנות מהעבודה:

- השימוש בבקר ארדואינו ובסימולטור Tinkercard איפשר לנו לפתח ולבדוק את המערכת בצורה מהירה ויעילה. השימוש בטכנולוגיות אלו הקל על תהליך הפיתוח ותרם רבות להצלחת הפרויקט.
- תהליך הבדיקות שביצענו היה חיוני להצלחת המערכת. הבדיקות בסימולטור ולאחר מכן בחומרה האמיתית הבטיחו שכל הרכיבים פועלים כמצופה ושאין בעיות בתקשורת בין הרכיבים לבקר הארדואינו.

שדרוגים אפשריים של מערכת הבקרה:

- **הוספת חיישני תאורה חיצונית:** כדי להתאים את תאורת החדר לתאורה החיצונית באופן אוטומטי.
  ע"י : התקנת חיישני תאורה שימדדו את רמת התאורה החיצונית ויתאימו את עוצמת תאורת ה LED בחדר בהתאם. זה יכול לשפר את חוויית השהייה על ידי חיסכון באנרגיה והתאמה אישית של תאורת החדר.

- **הוספת חיישני טמפרטורה ולחות:** כדי לשמור על טמפרטורה ולחות נוחות בחדר.
  ע"י : התקנת חיישני טמפרטורה ולחות שינטרו את התנאים בחדר ויחברו למערכת מיזוג האוויר או למכשירי האוורור כדי לשמור על תנאים נוחים בהתאם להעדפות האורחים.

7. נספחים:

א. קוד התוכנה-

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

// Define pin numbers for the ultrasonic sensor
const int trigPin = 6;
const int echoPin = 5;

// LCD I2C configuration
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Define pin numbers for the LED, button, servo, and piezo buzzer
#define led  11          // LED connected to digital pin 11
#define inPin 9          // Button connected to digital pin 9
#define servoPin 10      // Servo connected to digital pin 10
#define servoButtonPin 7// Button connected to digital pin 8 for servo control


Servo myServo;  // Create servo object to control a servo

int buttonState = 0;          // Variable to store the current state of the
LED button
int lastButtonState = 0;      // Variable to store the previous state of the
LED button
```

```
int ledState = LOW;           // Variable to store the state of the LED (LOW
means off)

int servoButtonState = 0;     // Variable to store the current state of the
servo button
int lastServoButtonState = 0; // Variable to store the previous state of the
servo button
int servoState = 0;           // Variable to store the state of the servo (0 =
right, 1 = left)

unsigned long previousMillis = 0;
const long interval = 500;    // Interval for checking distance
String lastStatus = "";       // Variable to store the last status displayed
on the LCD

// Function to get distance from ultrasonic sensor
long getDistance() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1; // Convert duration to distance in cm
  return distance;
}

void setup() {
  // Initialize the serial monitor for debugging
  Serial.begin(9600);

  // Initialize LCD I2C
  lcd.begin(16, 2);
  lcd.print("Initializing...");
  delay(2000);
  lcd.clear();

  // Set up the ultrasonic sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Set up the button, LED
  pinMode(inPin, INPUT_PULLUP); // Set LED button pin as input with internal
pull-up resistor
  pinMode(servoButtonPin, INPUT_PULLUP); // Set servo button pin as input with
internal pull-up resistor
  pinMode(led, OUTPUT);         // Set LED pin as output

  digitalWrite(led, ledState);  // Initialize the LED state


  // Attach the servo to the servo pin
  myServo.attach(servoPin);
  myServo.write(90); // Initialize the servo to 90 degrees (middle position)
```

```
}

void loop() {
  // Button and LED logic
  buttonState = digitalRead(inPin); // Read the state of the LED button value

  // Check if the LED button state has changed
  if (buttonState != lastButtonState) {
    // If the LED button state has changed to LOW, toggle the LED
    if (buttonState == LOW) {
      ledState = !ledState; // Toggle the LED state
      digitalWrite(led, ledState); // Update the LED with the new state
    }
    delay(50); // Debounce delay
  }

  // Save the current LED button state as the last state for the next loop
  lastButtonState = buttonState;

  // Servo button logic
  servoButtonState = digitalRead(servoButtonPin); // Read the state of the
servo button value

  // Check if the servo button state has changed
  if (servoButtonState != lastServoButtonState) {
    // If the servo button state has changed to LOW, toggle the servo position
and piezo buzzer
    if (servoButtonState == LOW) {
      if (servoState == 0) {
        myServo.write(180); // Move the servo to 180 degrees (left)
        delay(3000); // Wait for 500 milliseconds
        myServo.write(90);
        servoState = 1;
      } else {
        myServo.write(0); // Move the servo to 0 degrees (right)
        delay(3000); // Wait for 500 milliseconds
        myServo.write(90);
        servoState = 0;
      }
    }
    delay(50); // Debounce delay
  }

  // Save the current servo button state as the last state for the next loop
  lastServoButtonState = servoButtonState;

  // Measure the distance using the ultrasonic sensor
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    long distance = getDistance();
    String status;

    // Determine the status based on the distance
```

```
    if (distance > 10) {
      status = "Room Empty";
    } else {
      status = "Room Occupied";
    }

    // Update the LCD only if the status has changed
    if (status != lastStatus) {
      lastStatus = status;
      lcd.clear();
      lcd.print(status);
    }
  }
}
```

ב. קוד פיתון לבדיקת ההשערה הסטטיסטית:

```
import numpy as np
from scipy import stats
from scipy.stats import t

# הזמנים שנמדדו במאיות שניות
times = [187, 177, 181, 179, 190, 178, 183, 175, 175, 180]

# התוחלת שהשערת האפס טוענת
mu = 180

# ביצוע מבחן t
t_statistic, p_value = stats.ttest_1samp(times, mu)

# הצגת התוצאות
print(f'T-Statistic: {t_statistic}')
print(f'P-Value: {p_value}')

# פרשנות התוצאה
alpha = 0.05
if p_value < alpha:
    print("דחיית השערת האפס: יש הבדל משמעותי בין משך הזמן שנמדד לתוחלת של 3 שניות.")
else:
    print("אין דחיית השערת האפס: אין הבדל משמעותי בין משך הזמן שנמדד לתוחלת של 3 שניות.")
```

ג. קובץ הוראות למפעיל (READ ME):

README

Arduino Project - Smart Hotel Room Control

Introduction

This project aims to create a smart hotel room control system using Arduino. The system provides a comfortable and private experience for guests by allowing them to control room facilities with the press of a button. The system includes a motorized curtain, an LED light, and a real-time room occupancy display using an ultrasonic sensor and an LCD screen.

System Components

- Arduino Uno
- Ultrasonic Sensor (HC-SR04)
- LCD Screen (I2C)
- Servo Motor
- LED Light
- Push Buttons
- Breadboard and Jumper Wires

Usage

1. **Power the Arduino:** Connect the Arduino to a power source.
2. **Observe the LCD screen:** The screen will display "Room Empty" after initialization.
3. **Control the curtain:** Use the button connected to pin 7 to open and close the curtain.
4. **Control the LED:** Use the button connected to pin 9 to toggle the LED on and off.
5. **Check room occupancy:** The LCD screen will display "Room Occupied" or "Room Empty" based on the readings from the ultrasonic sensor.

Requirements

- Arduino IDE
- LiquidCrystal_I2C library
- Servo library
- Power source for Arduino
- **Basic electronic components** (breadboard, jumper wires, resistors)

Authors

- Carolina Fain
- Ohad Snir
- Hagar Shifman
- Yonatan Seleznev