



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

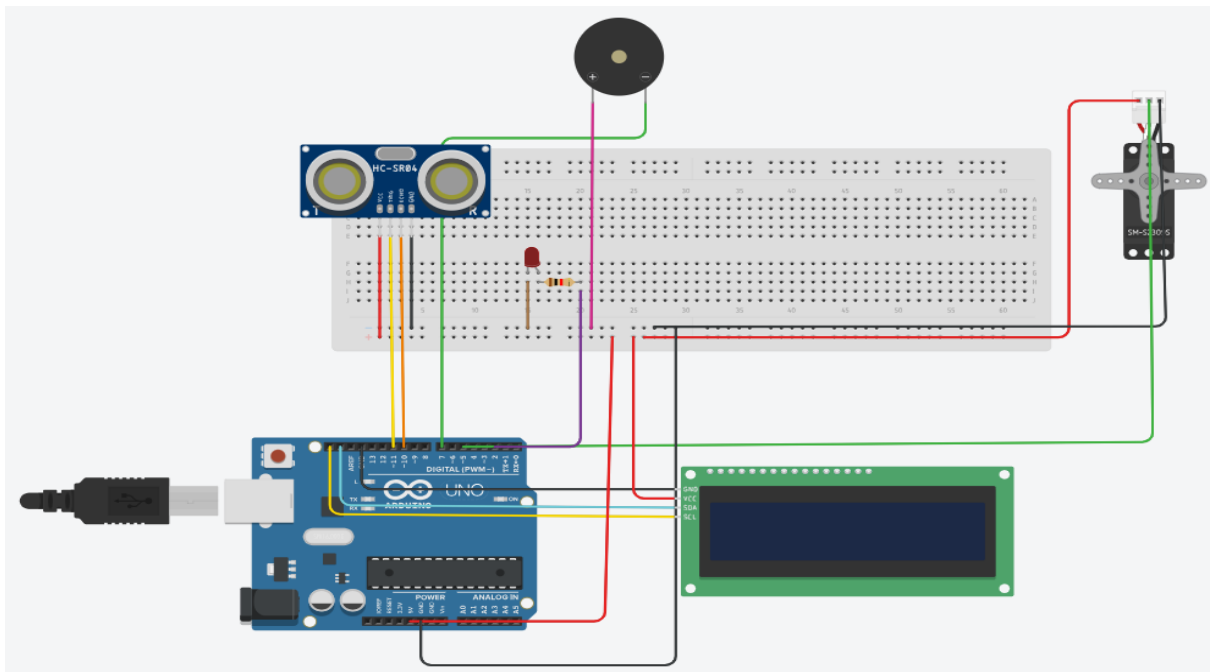
מספר קבוצה	שם התרגיל			תאריך הגשה
10	מטלה 3 - ARDUINO			29.02.24
204408728	206993420	207207119	316553981	315552489

1. מטרת המערכת ותיאור משימת הבקרה

במרפסת החיצונית הגדולה של חברת X שבה עובדי המשרד שוהים בזמני הפסקות, אוכל ובכל זמן בו הם רוצים הפוגה כלשהי ישנה מכת עורבים אשר שוהים מעל גג משרדי החברה ומנסים לקחת מהמזון של עובדי המשרד וכן מרעישים ומפריעים לשלוות עובדי המשרד. בניסיון לטפל בבעיה המטרידה פנו אלינו למציאת פתרון רלוונטי למצב הבעייתי. אנחנו בנינו מערכת בדמות דחליל שמטרתה להתריע על קרבתם של עורבים ולהבריחם באמצעות תזוזת הדחליל והשמעת רעש בעל תדר ייחודי. המערכת עובדת בצורה הבאה - המערכת תזהה עורב שחצה את הגבול המוגדר (70 ס"מ), תירשם הודעת מרחק בצג ה-LCD ומיד תושמע אזעקה. בהתאמה תופעל נורת אזהרה מהבהבת וכן יופעל מנוע הסרבו אשר יגרום לתזוזת הדחליל במטרה להרתיע ולהרחיק העורב.

2. תיאור התכן

הפרוייקט מומש באמצעות ערכת ARDUINO. הרכיבים הכלולים בפרוייקט הינם: מסך LCD, נורת LED, זמזם Piezo, גלאי מרחק, Ultrasonic Distance Sensor ומנוע Micro servo. התכן (איור 1, טבלה 1) נעשה בעזרת תוכנת הסימולטור Thinkercad. את התכן ניתן לראות בקישור: [קישור לסימולטור](#). תיעוד הקוד מפורט בנספח למסמך זה.



איור 1: שרטוט חשמלי של הפרוייקט



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

להלן הגדרת המשתנים:

מספר הפין	שם המשתנה	תצורה	הפעלה
7	piezo	output	מפעיל את הבאזר (אזעקה)
2	ledRed	output	נדלקת כאשר הגוף נמצא בטווח
9	servoPin	output	מפעיל את הזזת הסרבו
11	trigPin	output	שולח אות קולי לחישוב המרחק
10	echoPin	input	קולט את האות שנשלח חזקה לאולטרה - סוני כדי לחשב את המרחק

3. בדיקות ולידציה של המערכת

בוצעו שתי בדיקות ולידציה של המערכת – בדיקת התאמה של החומרה לדרישות ובדיקה של הלוגיקה של המערכת.

בדיקה מס' 1 – בדיקת התאמה של החומרה לדרישות

הבדיקה הראשונה בחנה האם דיוק חיישן המרחק (Ultrasonic distance) מתאים לדרישות המערכת. נדרש כי חיישן המרחק יהיה מדויק כמו חיישן מרחק אחר מערכה של חברינו לתואר. לכן, בוצעו 10 מדידות של החיישן במרחקים שונים. המרחק בכל מדידה נבדק ע"י קריאת החיישנים השונים. (טבלה 2) מדד דיוק החיישן הינו השגיאה האבסולוטית בין קריאות החיישנים. ההשערות הן:

$$H_0: \mu_D = 0$$

$$H_1: \mu_D \neq 0$$

בוצע מבחן t למדגמים בלתי תלויים תוך הנחת שוויון שוניות לבדיקה האם ממוצע השגיאות הינו אפס. המבחן בוצע ברמת מובהקות $\alpha = 5\%$.

להלן תוצאות המדידה תוך חישוב שגיאת חיישן מרחק:

מספר בדיקה	מרחק רצוי	מרחק נמדד	שגיאה אבסולוטית
1	55	54	1
2	36	36	0
3	89	88	1
4	43	43	0
5	93	94	1
6	42	42	0
7	55	57	2
8	77	77	0
9	68	68	0
10	52	54	2



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

להלן תוצאות המבחן:

t-Test: Two-Sample Assuming Equal Variances

מרחק נמדד	מרחק רצוי	
61.3	61	Mean
394.0111111	397.3333333	Variance
10	10	Observations
	395.6722222	Pooled Variance
	0	Hypothesized Mean Difference
	18	df
	-0.033723953	t Stat
	0.486734217	P(T<=t) one-tail
	1.734063607	t Critical one-tail
	0.973468435	P(T<=t) two-tail
	2.10092204	t Critical two-tail

תוצאות הבדיקה הסטטיסטית:

ניתן לראות כי קיבלנו $P\text{value}=0.97$ ולכן נקבל את השערת האפס בר"מ 5% ונטען כי ממוצע השגיאות הינו אפס, ולכן החיישן האולטראסוני מדויק.



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

בדיקה שנייה - בדיקת הלוגיקה של המערכת

עבור בדיקת הלוגיקה של המערכת נבדקו הרכיבים במצבים הבאים:

א. במרחק גדול מ-70 ס"מ - וידאנו את תקינות חיישן המרחק בכך שבעת הצבת אובייקט במרחק גדול מהנדרש, לא נקבל אותות אזהרה – הנורה לא תידלק, לא תשמע אזעקה ומנוע Servon לא יפעל.

ב. במרחק קטן מ-70 ס"מ - וידאנו כי המערכת פועלת כראוי:

- מנוע Servo - סיבוב לזווית הנדרשת שהוזנה לקוד
 - מנורת Led - הבהוב בעת הגעת העורב עד לבריחתו
 - באזר Piezo - צפצוף בעת הגעת העורב עד לבריחתו
 - מסך LCD - הדפסת הודעה בעת הגעת העורב עד לבריחתו, יחד עם מרחקו
- ג. במרחק גדול מ-70 ס"מ - בדקנו שכל הרכיבים הרלוונטיים למערכת האזהרה לא פועלים יותר. בכל הבדיקות המערכת פעלה כמצופה. בהתאם הסקנו כי המערכת תוכנתה על-פי התכנון.

4. מסקנות והמלצות

פיתחנו מערכת התראה עבור רווחת עובדי המשרד. חיישן המרחק של המערכת נמדד בצורה תקינה והמערכת להברחת העורבים פועלת כפי שתוכננה עבור מרחקים שונים. בנוסף, למדנו במסגרת העבודה שכאשר עובדים עם המיקרו בקר ומספר חיישנים נוספים, כדאי להתחיל מראש בבדיקת כל חיישן בנפרד ולהבין כיצד הוא עובד במקום לחבר את כל המערכת ורק אז לבצע את הבדיקות מה שמקשה על התהליך משמעותית. עבור פרויקטי המשך, אנו נמליץ לבחון מעבר למצלמה על פני חיישן מרחק ושימוש בעיבוד תמונה במטרה להרתיע על עורבים בלבד.



5. נספחים

נספח א' - הוראות הפעלה למפעיל Readme

1. צרוב את התוכנה "CrowDetection" לבקר הארדואינו.
2. לאורך ההרצה יודפס המרחק המזוהה על-ידי חיישן המרחק עבור כל רגע נתון על גבי צג ה-LCD.
3. כאשר תזוזה תנועה במרחק של עד 70 ס"מ מחיישן המרחק - ירשם בצג ה-LCD המרחק המתאים של העורב (DISTANCE X), תידלק ותהבהב נורת הלבד ומנוע הסרבו יחל להסתובב עם כיוון השעון ונגד כיוון השעון לסירוגין (בכל חצאי סיבוב - 180 מעלות).
4. לאחר התרחקות העורב מהמרחב, האזעקה תיפסק, הנורה תיכבה ומנוע הסרבו יפסיק להסתובב. בשלב זה המערכת עדיין פועלת לזיהוי עורבים נוספים.

נספח ב' – תיעוד הקוד

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

Servo servo;
LiquidCrystal_I2C lcd_1(0x27, 16, 2); // Adjust the address (0x27) if needed

int piezo = 7;
int servoPin = 9;
int angle = 0;
bool Direction = true;
bool Is_LED_OFF = true;
int ledRed = 2;
int trigPin = 11;
int echoPin = 10;
long duration, distance;

unsigned long ultraSonicPreviousMillis = 0;
const long ultraSonicInterval = 60; // Interval in milliseconds for the
UltraSonic function

unsigned long byecrowPreviousMillis = 0;
const long byecrowInterval = 1000; // Interval in milliseconds for the
Bye_Crow function

unsigned long servoPreviousMillis = 0;
const long servoInterval = 15; // Interval in milliseconds for the Serv
function

void setup() {
```



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

```
lcd_1.begin(16, 2); // Initialize the LCD
lcd_1.backlight();
lcd_1.blink_on();

pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(piezo, OUTPUT);
servo.attach(servoPin);
Serial.begin(9600);
}

void loop() {
  UltraSonic();
  Distance_Calculation();
  lcd();
  if (distance < 70) {
    Bye_Crow();
    ServoFunction(); // Call ServoFunction only when something is detected
    within 50 cm
  } else {
    return_normal();
  }
}

void UltraSonic() {
  unsigned long currentMillis = millis();

  if (currentMillis - ultraSonicPreviousMillis >= ultraSonicInterval) {
    ultraSonicPreviousMillis = currentMillis;

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
  }
}

void Distance_Calculation() {
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
}

void lcd() {
  Serial.print("Distance: ");
  Serial.print(distance);
}
```



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

```
Serial.println(" cm");

String disp = String(distance);

lcd_1.clear();
lcd_1.print("Distance :");
lcd_1.setCursor(0, 1);
lcd_1.print(disp);
lcd_1.print(" cm");
}

void Bye_Crow() {
    unsigned long currentMillis = millis();

    if (currentMillis - byecrowPreviousMillis >= byecrowInterval) {
        byecrowPreviousMillis = currentMillis;

        blink(ledRed);
        buzz(piezo);
        ServoFunction();
    }
}

void return_normal() {
    noTone(piezo);
    servo.write(90);
    digitalWrite(ledRed, LOW);
}

void blink(int led) {
    if (Is_LED_OFF) {
        digitalWrite(led, HIGH);
        Is_LED_OFF = false;
    } else {
        digitalWrite(led, LOW);
        Is_LED_OFF = true;
    }
}

void buzz(int piezoName) {
    tone(piezoName, 2000);
}

void ServoFunction() {
    unsigned long currentMillis = millis();
```



הפקולטה למדעי ההנדסה - המחלקה להנדסת תעשייה וניהול
אוטומציה וייצור ממוחשב 364-1-3321
אוניברסיטת בן גוריון בנגב

```
if (currentMillis - servoPreviousMillis >= servoInterval) {  
    servoPreviousMillis = currentMillis;  
  
    if (Direction) {  
        angle = 0; // Start from 0 degrees  
        Direction = false;  
    } else {  
        angle = 180; // Move to 180 degrees  
        Direction = true;  
    }  
    servo.write(angle);  
}  
}
```