

דו"ח מעבדת ארדואינו

קבוצה 8

318269362- 208498105 – 208786814 – 208752311

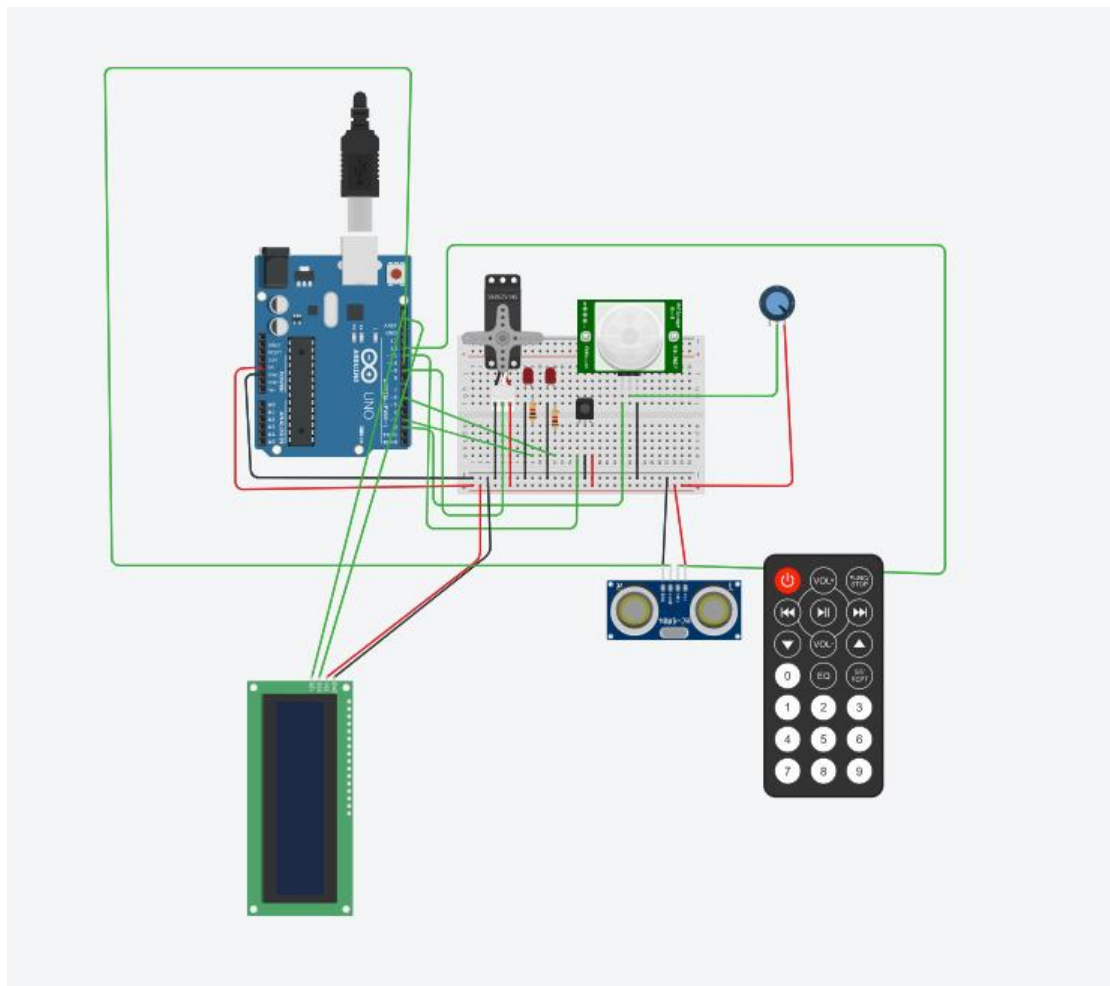
מטרת הפרויקט:

המטרה הייתה לייצר מערכת אוטומטית להאכלת חייית המחמד במטרה לעזור לבעלי חיות המחמד לדאוג להזנתם בשתי דרכים: הבעלים יכול להורות למערכת לבצע האכלה מרחוק באמצעות שימוש בשלט IR, וכן עם התקרבות בעל החיים לתחום החישה המערכת תבדוק את התקיימות התנאי המתאים להאכלה כלומר עבר X זמן (המדומה במערכת ע"י 20 שניות) וכן החיישן זיהה תנועה, כדי למנוע האכלה ע"י המערכת בצורה לא מבוקרת.

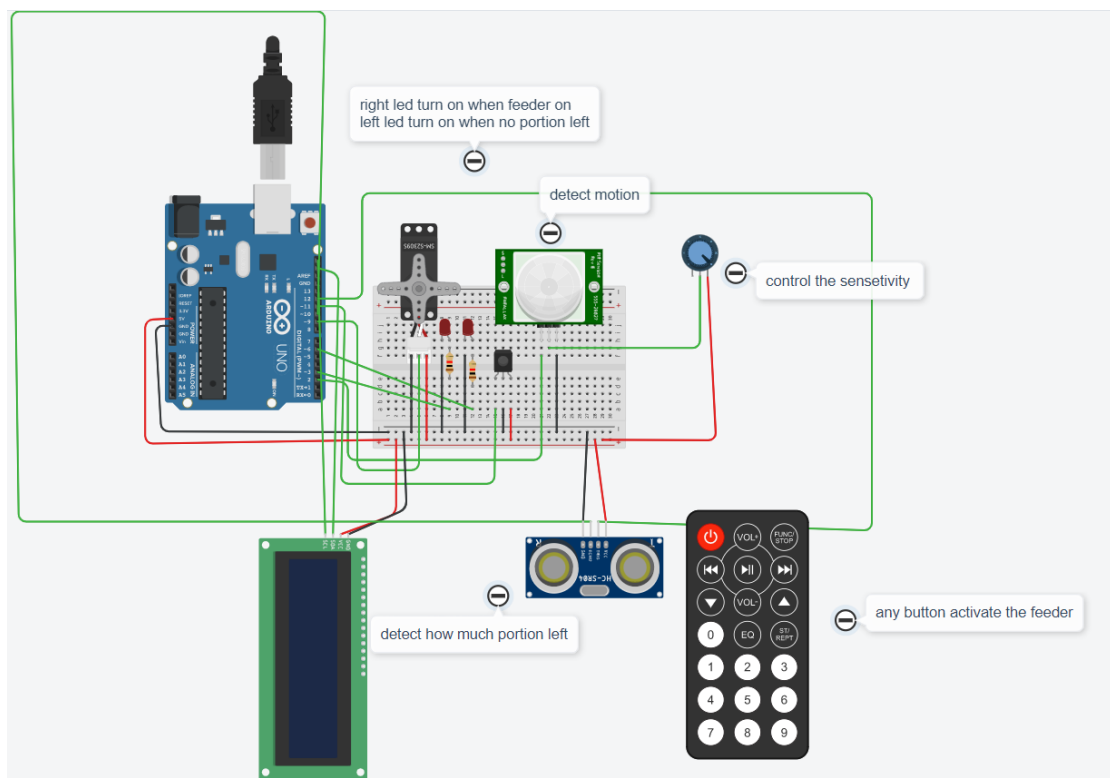
מסך LCD ילווה את התהליך בהצגת הודעות עדכון מצב המערכת מבחינת מספר המנות שנשארו במיכל.

נורת LED תהווה חיווי לביצוע האכלה ע"י המערכת.

תיאור התוכן: הפרויקט מומש באמצעות בקר ארדואינו אנו, הרכיבים הכלולים בפרויקט הינם: Ultrasonic Distance Sensor, IR שלט, PIR Sensor, נורת לד, מנוע סרבו, מסך LCD. תמונת האילוסטרציה למערכת שבנינו לקוחה מתוך תוכנת הסימולציה tinkercad בה עבדנו במהלך הפרויקט, להלן קישור המכיל את המודל ואת הקוד שבנינו: https://www.tinkercad.com/things/fwVoKeueyaC-ex3/editel?sharecode=KVbjcc5jS5rWSUqStWgkHvYBpD_heHHa4zN1WNSU490



תמונת תקריב של המודל:



הנחות:

- להדמיה בלבד הוספנו פוטנציומטר, שלפיו ניתן להתאים את רגישות חיישן התנועה. בחיישן האמיתי קיימת אופציה כזו לכן למודל האמיתי פוטנציומטר יהיה מיותר.
- כל ירידה של 5 ס"מ בגובה חיישן המרחק מהווה ירידת מנה אחת כאשר הכמות ההתחלתית היא 10 מנות.

מספר הפין	שם המשתנה	תצורה	הפעלה
11	חיישן IRrecv irrecv	INPUT	מקבל פקודות מהשלט להפעלה מרחוק ע"י המשתמש
12	trigPin	OUTPUT	שולח גלי קול במטרה לחישוב המרחק
10	echoPin	INPUT	מחשב את המרחק של האובייקט ע"י קבלת הזמן שגלי הקול חזרו לאחר שפגעו באובייקט
2	PIR	INPUT	חיישן תנועה - מזהה תנועה בתחום החישה של החיישן
6	נורת לד	OUTPUT	מחוזה על פעילות המערכת – האכלה
3	נורת לד	OUTPUT	מחוזה על חוסר במנות מזון
9	מנוע סרבו	OUTPUT	משמש להפעלת המערכת
SDA SCL	מסך LCD	OUTPUT	משמש להצגת מצב המערכת מבחינת כמות מנות שנותרו במיכל

תיאור משימות הבקרה:

בדיקת חומרה: במודל שלנו קיים חיישן אולטרה סוניק שמשמש אינדיקציה לכמות האוכל במיכל ע"י מרחקו מקו מסוים. ככל שהערך המתאר את המרחק מהקו קטן יותר – משמע, המיכל מלא יותר. אם המרחק מקסימלי הרי המיכל ריק וצריך למלא אותו במנת מזון נוספת.

כדי לבצע בדיקת התאמה של החומרה לדרישות, בדקנו אם דיוק החיישן מתאימה לדרישות המערכת. נדרש כי החיישן יהיה מדויק כמו סרגל במדידת המרחקים. לכן בוצעו 10 מדידות של המרחק בעבור 10 מרחקים שונים, ובבדיקה המדידה הן בסרגל והן בחיישן. מדד דיוק החיישן הינו השגיאה האבסולוטית ביניהם.

הבדיקות שבוצעו לבדיקת התאמתו של הערך שמחזיר החיישן לערך המרחק האמיתי:

שגיאה אבסולוטית	ערך שהתקבל מהחיישן (בס"מ)	ערך אמיתי (בס"מ)	בדיקה
28	1028	1000	1
18	1118	1100	2
51	1251	1200	3
9	1309	1300	4
96	1496	1400	5
94	1594	1500	6
57	957	900	7
-7	793	800	8
13.2	313.2	300	9
2.7	402.7	400	10

ביצענו מבחן t מזווג למדגמים תלויים בהנחת שוויון שוניות והנחה כי השגיאות מגיעות מההתפלגות הנורמלית, כיוון שמדובר על שני מדגמים שאמורים לייצג את אותו המרחק כאשר אחד מהמדגמים מהווה את המרחק האמיתי שנמדד בסרגל והשני את הערכתו לפי החיישן ששילבנו במערכת.

H0: ההפרשים בתוצאות המדידה = 0 כלומר ממוצע השגיאות הוא אפס, תוצאות המדידה בחיישן זהות לתוצאות המדידה בסרגל.

H1: אחרת.

ערך t סטטיסטי: $t = 3.1053$ ערך t קריטי $t = 2.100922$

הערך הסטטיסטי גדול מהערך הקריטי לכן נדחה את השערת האפס ונקבע כי המרחק שנמדד בחיישן שונה מהמרחק האמיתי שנמדד בסרגל בר"מ של 5%. בנוסף המבחן מובהק מאוד ובפרט רמת המובהקות (PVALUE) קטנה מאלפא (ושווה ל-0.0126) מה שתומך בהחלטתנו לדחות את השערת האפס.

לצורך ביצוע המבחן השתמשנו באתר <https://www.graphpad.com/quickcalcs/ttest2>. להלן פלטי המבחן:

Paired t test results

P value and statistical significance:

The two-tailed P value equals 0.0126

By conventional criteria, this difference is considered to be statistically significant.

Confidence interval:

The mean of Group One minus Group Two equals 36.190

95% confidence interval of this difference: From 9.826 to 62.554

Intermediate values used in calculations:

t = 3.1053

df = 9

standard error of difference = 11.654

Review your data:

Group	Group One	Group Two
Mean	1026.190	990.000
SD	426.659	401.248
SEM	134.921	126.886
N	10	10

בדיקת ולידציה: עבור בדיקת הלוגיקה של המערכת נעשו 8 מעברים בין מצבים שונים של המערכת. בכל הבדיקות המערכת פעלה כמצופה כמפורט בטבלה הבאה. בהתאם הסקנו כי המערכת תוכנתה על פי התכנון.

מספר בדיקה	מצב נתון	תגובה רצויה	תגובת המערכת
1	אין לחיצה על שלט + IR זיהוי תנועה + עבר מספיק זמן מאז האכלה הקודמת	האכלה מתבצעת	נורת LED נדלקה
2	אין לחיצה על שלט + IR זיהוי תנועה + לא עבר מספיק זמן מאז האכלה הקודמת	האכלה לא מתבצעת	נורת LED כבויה
3	אין לחיצה על שלט + IR אין זיהוי תנועה + עבר מספיק זמן מאז האכלה הקודמת	האכלה לא מתבצעת	נורת LED כבויה
4	אין לחיצה על שלט + IR אין זיהוי תנועה + לא עבר מספיק זמן מאז האכלה הקודמת	האכלה לא מתבצעת	נורת LED כבויה

5	לחיצה על שלט IR + זיהוי תנועה + עבר מספיק זמן מאז האכלה הקודמת	האכלה מתבצעת	נורת LED נדלקה
6	לחיצה על שלט IR + זיהוי תנועה + לא עבר מספיק זמן מאז האכלה הקודמת	האכלה לא מתבצעת	נורת LED כבויה
7	לחיצה על שלט IR + אין זיהוי תנועה + עבר מספיק זמן מאז האכלה הקודמת	האכלה מתבצעת	נורת LED נדלקה
8	לחיצה על שלט IR + אין זיהוי תנועה + לא עבר מספיק זמן מאז האכלה הקודמת	האכלה לא מתבצעת	נורת LED כבויה
9	חיישן מרחק לא מזהה עצם 50 ס"מ מולו	חוסר במנות אוכל במיכל	נורת LED אדומה נדלקה על המסך תופיע אינדיקציה לחוסר במנות
10	חיישן מרחק לא מזהה עצם 50 ס"מ מולו	יש מספיק מנות אוכל במיכל להפעלות נוספות	נורת LED אדומה כבויה על המסך יופיעו כמות המנות שנותרו

סיכום ומסקנות להמשך

במסגרת הפרויקט פיתחנו מערכת אוטומטית להאכלת חיית המחמד.

ביצענו בדיקות בקרה למערכת – מבחינת חומרה, חיישן המרחק של המערכת נבחן ונמצא לא מדויק. עם זאת, מבחינת לוגיקת המערכת, המערכת פעלה כשורה בכל המצבים האפשריים בסט הבדיקות. להמשך הפיתוח מומלץ לבחון חיבור של זמזם Piezo שיבצע זמזום התרעה בעת סיום האוכל במיכל כדי לסמן למשתמש שעליו למלא את המיכל במזון חדש.

נספח 1: הוראות מפעיל

המשתמש יחבר את המכשיר (הבקר) למקור מתח.

המשתמש ימלא את מיכל האוכל ב-10 מנות מסוג המתאים לבעל החיים לו מיועד המכשיר.

כמות המזון במיכל תתעדכן ותוצג על מסך ה-LCD.

במידה והמשתמש ירצה להפעיל את המכשיר מרחוק, הוא ילחץ על כל כפתור בשלט ה-IR. כל עוד עבר מספיק זמן מאז ההאכלה האחרונה, תתבצע האכלה ונורת ה-LED תידלק.

במידה שתזוהה תנועה בתחום החישה של חיישן PIR ובנוסף עבר מספיק זמן מאז ההאכלה האחרונה, תתבצע האכלה ונורת ה-LED תידלק.

אחרת, לא תתבצע האכלה ונורת ה-LED לא תידלק.

נספח 2: תיעוד קוד התוכנית

```
#include <Adafruit_LiquidCrystal.h>
#include <IRremote.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>

//lcd
LiquidCrystal_I2C lcd_1(0x27, 16, 2);
int portion= 10;
//

int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results; // Create a variable to store the results of
IR decoding
int onePress = 0;
long duration;
int distance;
const int trigPin = 12;
const int echoPin = 10;

Servo myservo;
int led = 6;
int led2 = 3;

int pir = 2;
int isOpen = 0;
int feedingAllowed = 1;
unsigned long Feeds = 0;
unsigned long lastFeed = 0;
```

```

void setup()
{

    //lcd

    lcd_1.begin(16, 2);
    lcd_1.setBacklight(1);

    //
    irrecv.enableIRIn(); // Start the receiver

    pinMode(pir, INPUT);
    pinMode(led, OUTPUT);
    pinMode(led2, OUTPUT);

    myservo.attach(9);
    Serial.begin(9600);
    myservo.write(90); // move the servo motor to position 100 to
demonstrate open motion
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}

void loop(){
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Reads the echoPin, returns the sound wave travel time in
microseconds
    duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    distance = duration * 0.034 / 2;
    lcd_1.setCursor(0, 0); // Sets the location at which subsequent text
written to the LCD will be displayed

    int portion = 10 - (round(distance)/5);
    if(portion < 0){
        portion = 0;
        digitalWrite(led2, HIGH);
        lcd_1.clear();
        lcd_1.print("Food thank empty.Refill"); // Prints string "portion"
on the LCD
    }
}

```



```

else{
    lcd_1.clear();
    lcd_1.print("portion: "); // Prints string "portion" on the LCD
    lcd_1.print(portion); // Prints the distance value from the sensor
    digitalWrite(led2, LOW);
}

int val = digitalRead(pir);

if ( millis() - lastFeed > 20000)
{
    feedingAllowed=1;
    results.value = 0;
    onePress=0;
}

if (irrecv.decode(&results)) {
    // Process IR remote button press
    if(&results.value!=0){
        onePress = 1;
        Serial.println("Button pressed!");
        results.value = 0;
    }

    irrecv.resume();
}

if ((val == HIGH)&&feedingAllowed==1)
{
    isOpen=1;
    Serial.println("motion detected!");
}

if ((isOpen==1||onePress == 1)&& feedingAllowed==1){ // If the
number 1 button is pressed
    digitalWrite(led, HIGH); // turn on the LED
    myservo.write(180); // move the servo motor to position 180 to
demonstrate open motion
    isOpen = 1; // set flag to indicate servo is open
    delay(1000); // wait for 1 seconds
    digitalWrite(led, LOW); // turn off the LED
}

```

```
    myservo.write(0); // move the servo motor back to position 0 to
close it
    delay(1000); // wait for 5 seconds
    myservo.write(90); // move the servo motor back to position 0 to
close it
    isOpen = 0; // set flag to indicate servo is closed
    onePress = 0; // reset the button press variable
    feedingAllowed=0;
    Feeds++;
    Serial.print("Feeds : ");
    Serial.println(Feeds);
    lastFeed = millis();
    Serial.print("last Feed : ");
    Serial.println(lastFeed );

}

delay(10);

}
```