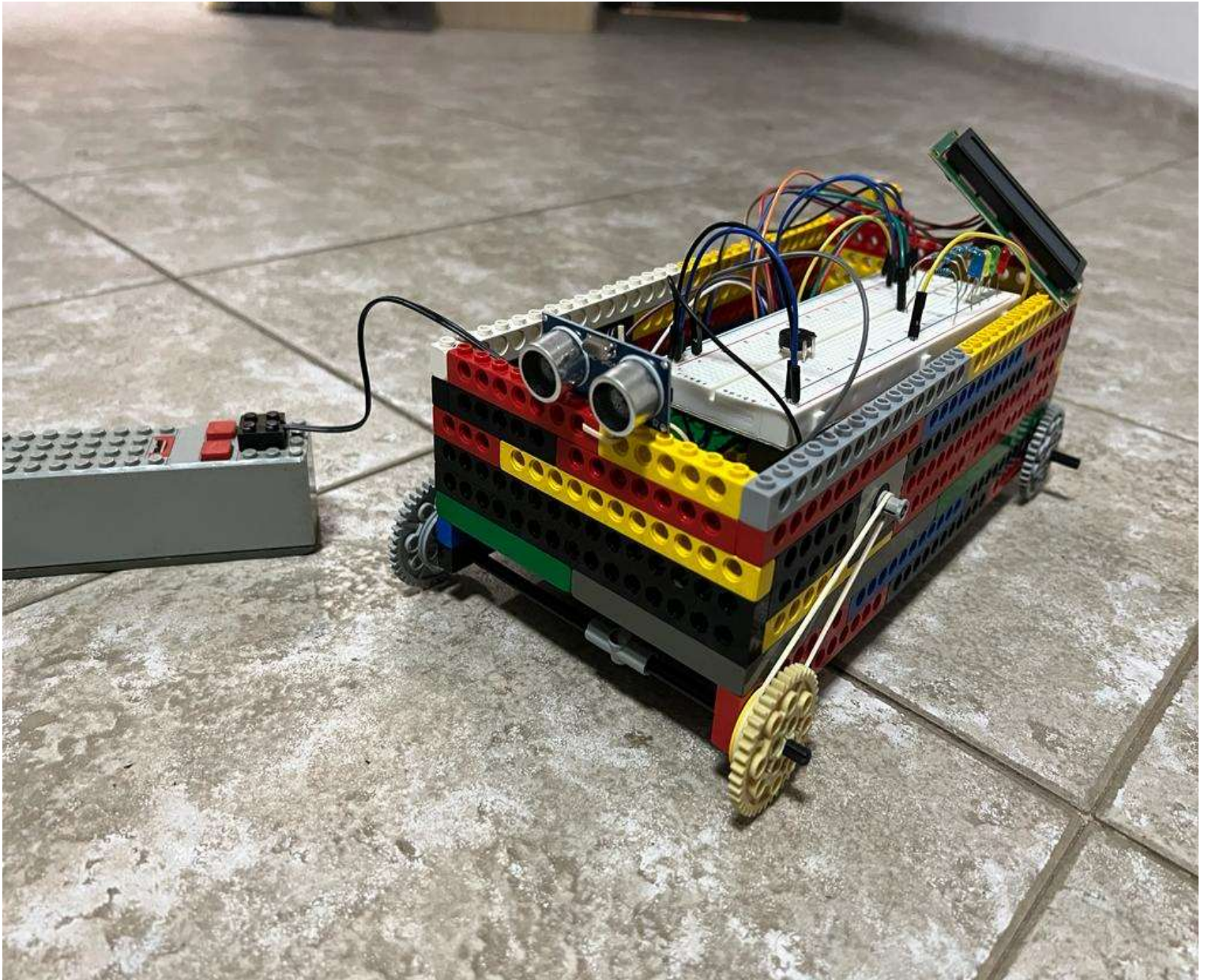


פרויקט בארדואינו: מד מרחק חיישני חוורס לרכב

Arduino Project- Distance meter for cars



מגשים:

ניב טורן

יוסף סוהיל פריזאדה

## תיאור הפרויקט:

הפרויקט מנצל את יכולת החיישן היפרסוני, השולח מצדו הראשון של החיישן גל קול אל עבר אובייקט מסוים. ברגע שהגל פוגע באובייקט הוא חוזר אל עבר החיישן השני היודע למדוד באיזה מרחק האובייקט נמצא מהחיישן.

המידע שנקלט מתוך החיישן היפרסוני מתורגם באמצעות הארדואינו אל עבר שני האמצעים הוויזואליים- מחד נמצא מסך הLED שמדפיס את המרחק שהתקבל בסנטימטרים, ומאידך נמצאות הנורות שמתארות את הקרבה אל עבר האובייקט.

ככל שהרכב מתקרב לאובייקט מסוים, מוצגות בפניו שני אמצעים- הראשון מבניהם הוא מסך הLED שמצביע לנהג מה המרחק שהרכב קרוב, והשני מבניהם הם הנורות שמצביעות אם המרחק הוא מרחק " בטוח" ( אותו הגדרנו בתור 10 סנטימטר) המאפשר לנהג הרכב לכניסה בטוחה של הרכב לחנייה עם סיכון נמוך לתאונה מזערית- ככל שכמה שיותר נורות נדלקות, הרכב קרוב לאובייקט. בזמן שנדלקות כל נורות הרכב- זהו המרחק הבטוח בין הרכב לאובייקט שכדאי לשמור.

## תיאור הרעיון ומימוש:

אנו לא מתארים את הפרויקט שלנו בתור פרויקט רגיל, אנו מתארים אותו בתור חזון- **ליצור עתיד בטוח יותר וזול יותר!**

מערכת חישני רוורס היא מערכת עזר לנהג בעת נסיעה לאחר שפותחה במתרה להתריע על פגיעה אפשרית של הרכב באובייקטים הנמצאים במסלול המסיעה. המערכת הינה אקטיבית המופעלת באמצעות חיווי אור. תפקידה של המערכת הוא להתריע על מצבי הסיכון שעלולים להיווצר בין מפגש בין כלי הרכב אל אובייקטים בסביבה, ויכול להוריד משמעותית את התאונות שנגרמות עקב נסיעה מאחור, ויכול גם למנוע פצועים בגוף ובנפש. מרבית התאונות נגרמות שקוראות לאחר נסיעה לאחר נגמרות בגלל שדה ראייה לקוי לאחר באזורים שנתיב הנסיעה אינו נראה במראה הפנימית ובמראות הצד, במקום הנקרא "שטחים מתים"

אנו מאמינים בחשיבות המערכת הזאת, שיכולה ליצור אלטרנטיבה זולה לשוק- לרכבים ישנים שאינה מותקנת בהם מערכת חישני רוורס, בעלות זולה ביחס לשוק ויכולה למנוע תאונות שנגרמות לאחר נסיעה לאחר.

## **רכיבי הארדואינו הנדרשים:**

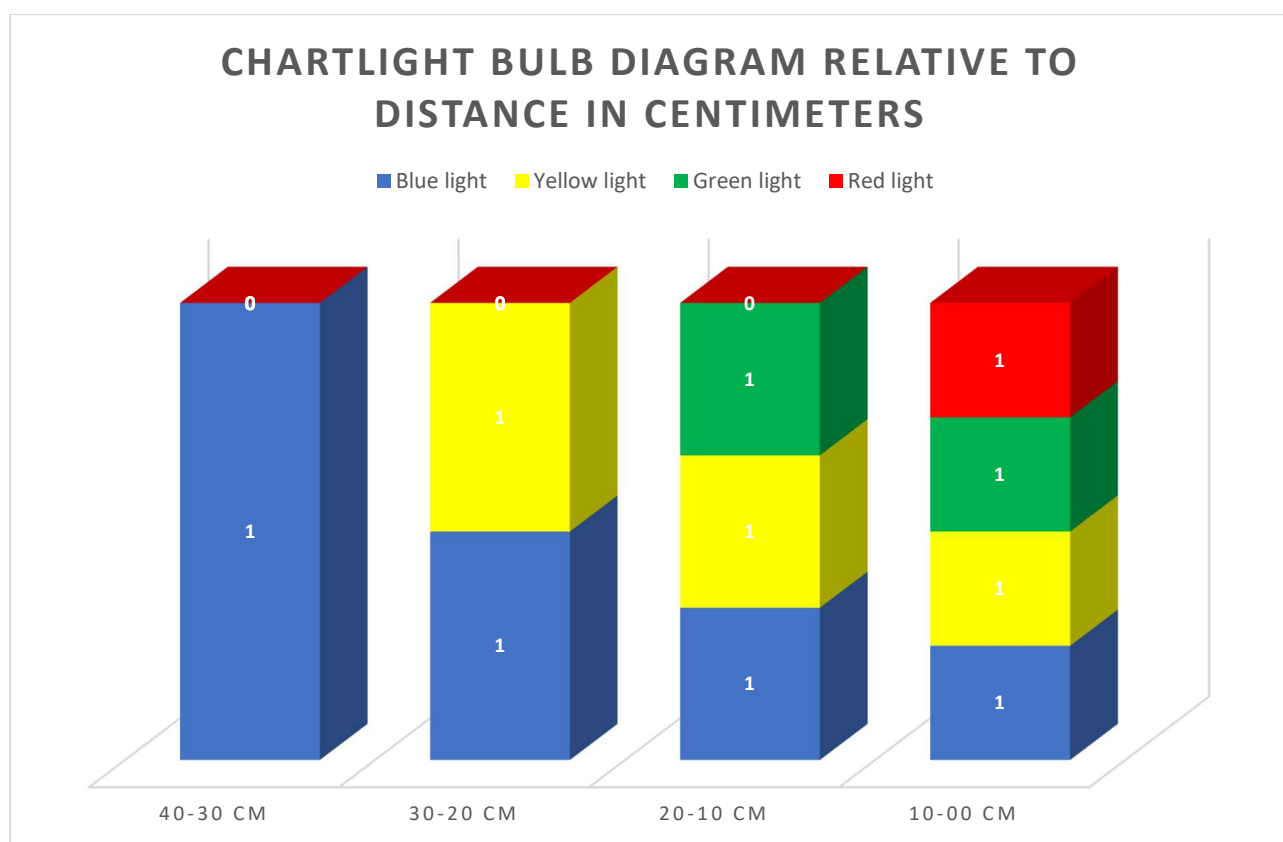
1. 1 Arduino Uno
2. 1 LCD 16X2
3. 1 Ultrasonic sensor
4. 4 light leds
5. 5 OHM Resistor
6. 1 Breadboard
7. 25 Jumper wires

## מדריך למשתמש:

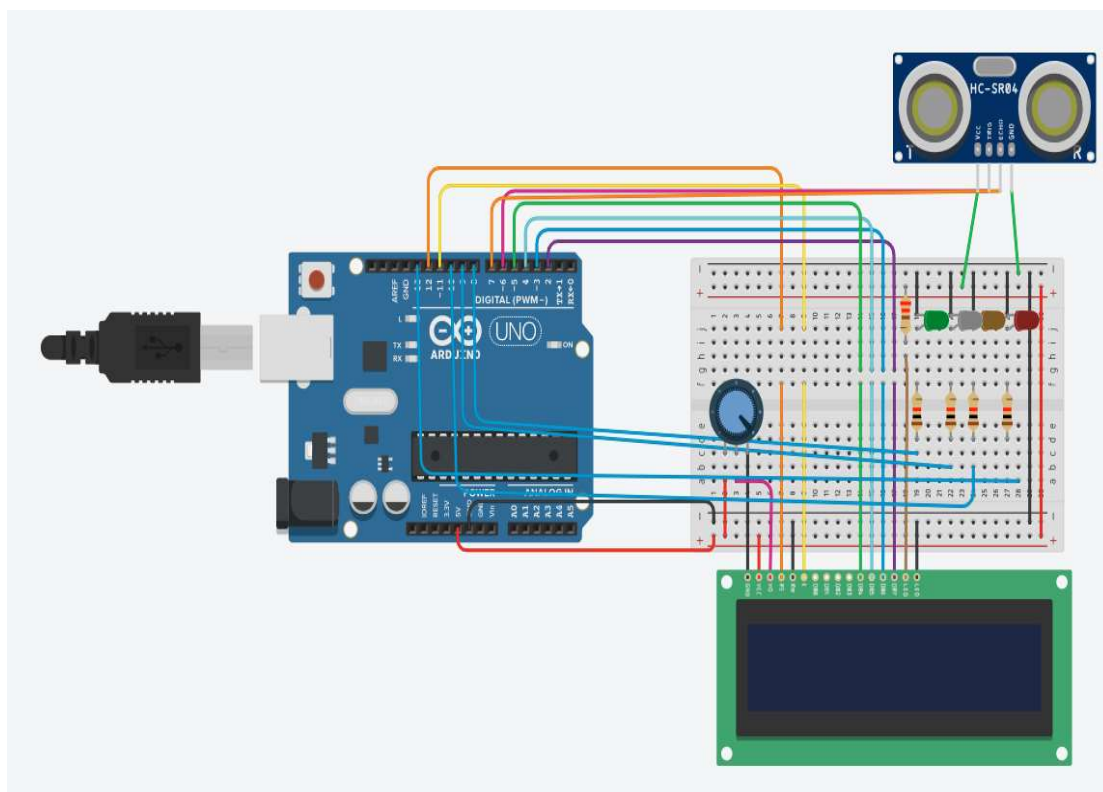
ראשית נציין כי זה רק אב-טיפוס של המערכת, וישתנה בעתיד.

בעת שילוב ידית ההילוכים יש לסובב את בורר המערכת ולהפעיל את חיישני הרוורס. לאחר מכן יש לנוע ברוורס והמערכת תפעל בהתאם. ברגע שהרכב ינוע אל עבר אובייקט מסויים, יפעלו ויודלקו הנורות, ובצד הלד יופיע המרחק בין הרכב לאובייקט. לאחר הנסיעה לאחור יש לסובב את בורר המערכת לכיוון הנגדי.

## דיאגרמה של האלגוריתם המתארת איך הוא עובד:



סרטוט הפרויקט:



קישור לסימולציה:

<https://www.tinkercad.com/things/ePJ8AMiwgk1-fantabulous-allis-trug/editel?sharecode=-m1HrrnptAniXOTDJxFI5QP-QtrGughffhwBsZT3Em8>



```
1  #include <LiquidCrystal.h>
2
3  const int trigPin = 6;
4  const int echoPin = 7;
5
6  const int LED1 = 8;
7  const int LED2 = 9;
8  const int LED3 = 10;
9  const int LED4 = 13;
10
11  int distance = 0;
12  int duration = 0;
13
14  const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
15  LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
16
17  void setup()
18  {
19      pinMode(trigPin, OUTPUT);
20      pinMode(echoPin, INPUT);
21
22      lcd.begin(16, 2);
23
24      pinMode(LED1, OUTPUT);
25      pinMode(LED2, OUTPUT);
26      pinMode(LED3, OUTPUT);
27      pinMode(LED4, OUTPUT);
28      Serial.begin(9600);
29  }
30
31  void loop()
32  {
33      // send a pulse on the trigger pin
34      digitalWrite(trigPin, HIGH);
35      delayMicroseconds(1000);
36      digitalWrite(trigPin, LOW);
37
38      // measure the duration of the pulse on the echo pin
39      long duration = pulseIn(echoPin, HIGH);
40
41      // convert the duration to distance using the speed of sound
42      long distance = (duration / 2) / 29.1;
43
44      lcd.setCursor(1, 1);
45  }
```

```
46  lcd.clear();
47  lcd.print("Distance = ");
48  //lcd.setCursor(10,10);
49  lcd.print(distance);
50  lcd.print(" cm");
51  delay(500);
52
53  if (distance <= 10)
54  {
55      digitalWrite(LED1,HIGH);
56  }
57  else
58  {
59      digitalWrite(LED1,LOW);
60  }
61  if (distance <= 20)
62  {
63      digitalWrite(LED2,HIGH);
64  }
65  }
66  else
67  {
68      digitalWrite(LED2,LOW);
69  }
70  if (distance <= 30)
71  {
72      digitalWrite(LED3,HIGH);
73  }
74  else
75  {
76      digitalWrite(LED3,LOW);
77  }
78  if (distance <= 40)
79  {
80      digitalWrite(LED4,HIGH);
81  }
82  else
83  {
84      digitalWrite(LED4,LOW);
85  }
86  }
```