

# פנינת נוי

7/2024

רחל גלטשטיין  
"דרכי חנה"

## תוכן עניינים

2	1. מבוא
2	1.1. רקע:
2	1.2. בחירת הפרויקט:
2	2. תקציר הפרויקט
2	2.1. תיאור פעולת המערכת:
4	2.2. תרשים מלבנים:
4	2.3. מפרט טכני:
5	2.4. תמונה של הפרויקט:
5	3. שרטוט חשמלי
6	4. רכיבי הפרויקט
6	4.1. בקרס Arduino Uno:
8	4.2. מסך LCD:
10	4.3. מסך מגע:
12	4.4. משאבת מים:
13	4.5. דוחף זרם:
14	4.6. בר לדים:
16	4.7. חיישן מרחק:
17	5. תיעוד הפרויקט
17	5.1. תהליך ביצוע הפרויקט:
17	5.2. בעיות בתהליך הפרויקט:
18	6. תוכנה
18	6.1. תרשים זרימה:
18	6.2. פירוט פונקציות:
19	6.3. קוד התכנית:
23	7. סיכום ומסקנות
23	8. פיתוחים עתידיים
23	9. ביבלוגרפיה

## 1. מבוא

### 1.1. רקע:

פינת נוי ומרגוע בגינה היא דבר חשוב לעין ערוך.

כאשר מישהו חש במצב לחץ/קושי/דאגה/עצבות הוא יכול לצאת לגינה, לשמוע את קולות פכפוך המים ולרחף באיזה ענן מתוק עם אורות משלהבים הנוסכים רוגע ושלווה.

פינת חמד כזאת היא אידיאלית לשיחות נפש מלב אל לב וכן יצירת אווירה פסטורלית ומענגת.

### 1.2. בחירת הפרויקט:

היות ואני מאד חובבת טבע, בעיקר מים וצמחייה וכן אוהבת מאד השקעה בפרטים קטנים, מיד ידעתי שאני רוצה שהפרויקט שלי יהיה משהו שמשלב את הכול.

וכך חשבתי על רעיון של פינת נוי לגינה.

מכיוון שאני אוהבת להתנסות בהרבה דברים, וגם החיישנים מאד גירו אותי, החלטתי לעשות גם את המפל וגם תאורה צבעונית שנדלקת כשמישהו מתקרב.

## 2. תקציר הפרויקט

### 2.1. תיאור פעולת המערכת:

על המשתמש להגדיר 2 דברים תחילה:

\* גווני תאורת האווירה שידלקו כאשר יהיה מישהו בקרבת המפל.

\* כמות המים שתזרום בעת זרימת המים במפל.

## מסך ראשי



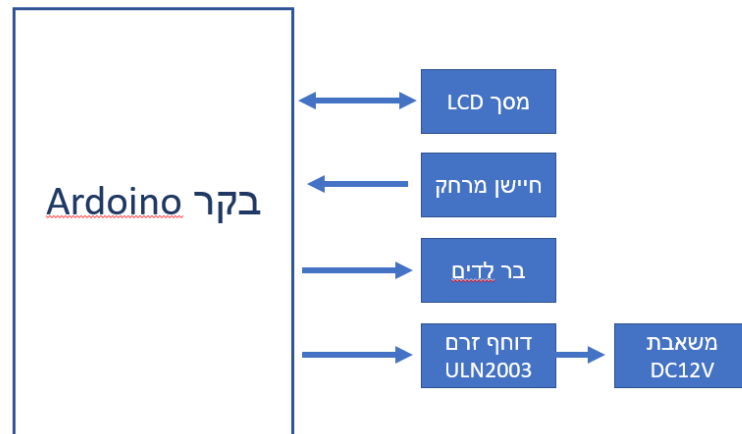
### מסך בחירת גווני תאורה



### מסך הגדרה עבור המפל דלוק/מכובה



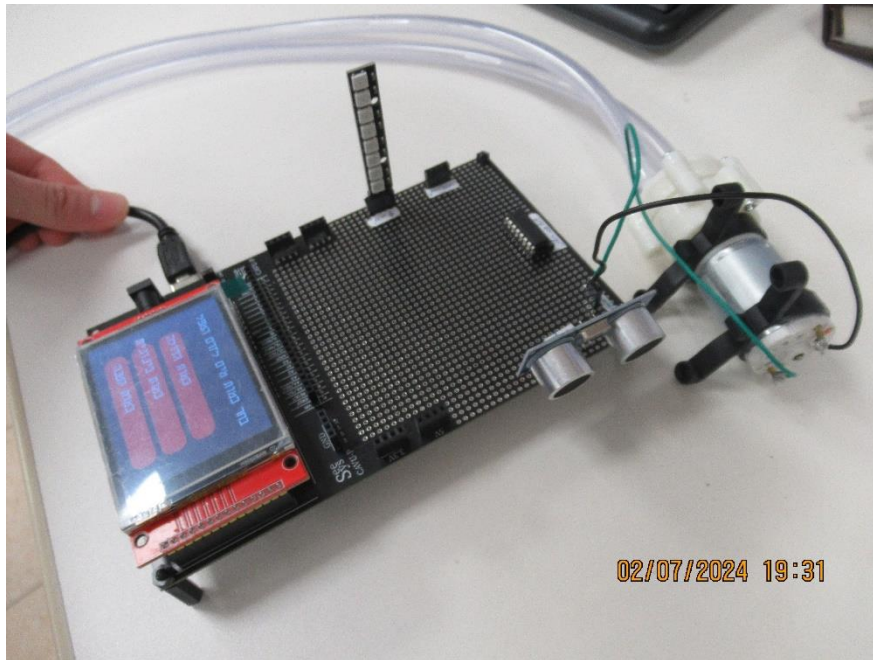
## 2.2 תרשים מלבנים:



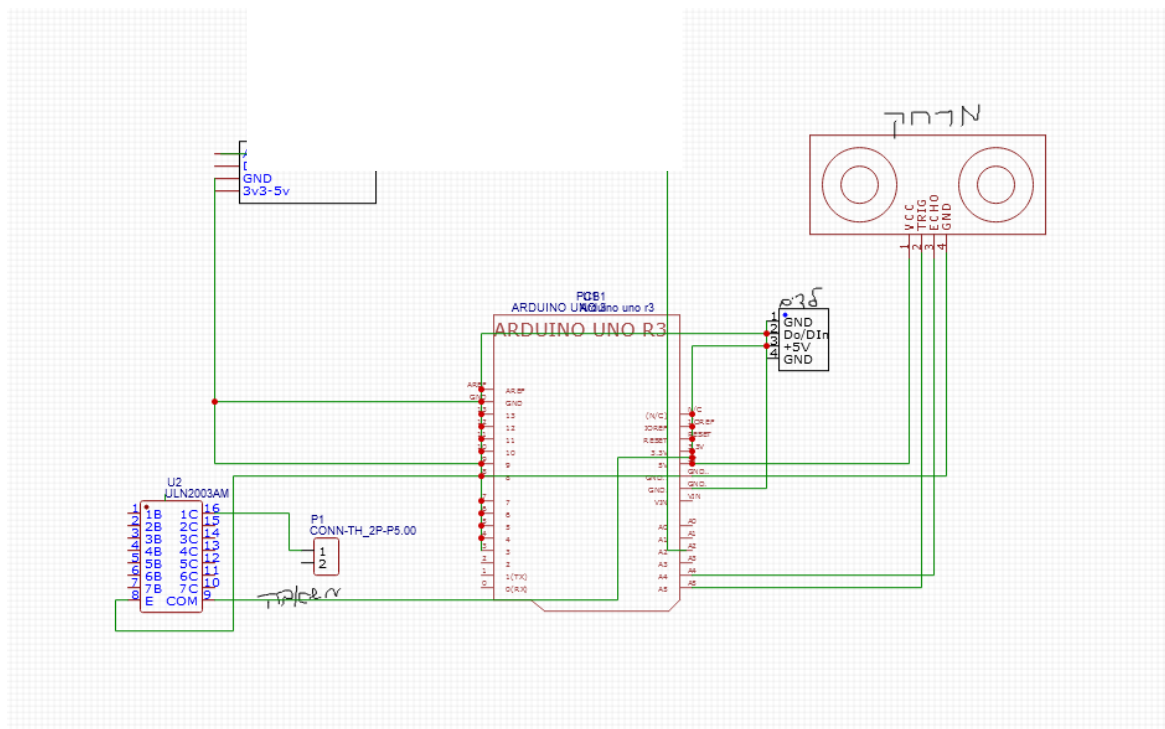
## 2.3. מפרט טכני:

- Arduino Uno
- מסך מגע גרפי 320 x 240 הכולל ממיר ADC 12 סיביות.
- משאבת מים
- דוחף זרם
- חיישן מרחק
- בר לדים

## 2.4. תמונה של הפרויקט:



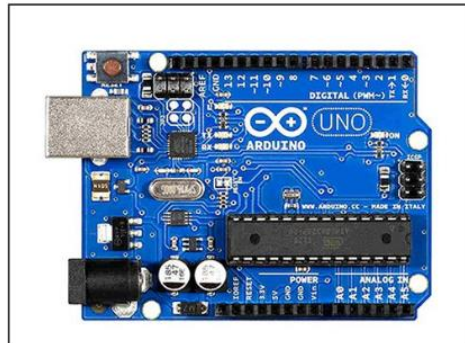
## 3. שרטוט חשמלי



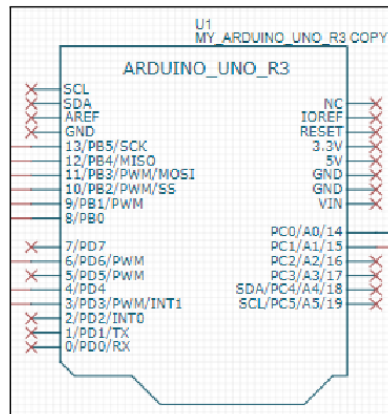
## 4.רכיבי הפרויקט

### 4.1.בקר Arduino Uno:

#### תמונה



#### שרטוט חשמלי



#### מאפיינים עיקריים

##### \* מיקרו-בקר (Microcontroller)

- ATmega328P: זהו המיקרו-בקר המרכזי של הלוח, אחראי על ביצוע הקוד. הוא מכיל זיכרון פלאש לאחסון התוכנה, זיכרון SRAM וזיכרון EEPROM.

##### \* כניסות ויציאות דיגיטליות (Digital I/O Pins)

- 14 פינים דיגיטליים (0-13): ניתן להשתמש בהם ככניסות או כיציאות. פינים מסוימים תומכים ב-PWM (פולס ברוחב מודולציה).

##### \* כניסות אנלוגיות (Analog Input Pins)

- 6 פינים אנלוגיים (A0-A5): מאפשרים קריאה של מתח אנלוגי וממירים אותו לערך דיגיטלי (ADC) באיכות של 10 ביטים.

##### \* יציאות מתח (Power Pins)

- VIN: כניסת מתח חיצונית (7-12V).

- V5: יציאת מתח של 5V.

- V3.3: יציאת מתח של 3.3V.

- GND: קרקע.

**\* מחבר USB Connector**

- מאפשר חיבור למחשב לצורך תכנות ותקשורת טורית.

**\* מחבר חשמל (Power Jack)**

- מאפשר חיבור של ספק כוח חיצוני.

**\* מייצב מתח (Voltage Regulator)**

- ממיר את המתח הנכנס מ-VIN למתח יציב של 5V.

**\* כפתור איפוס (Reset Button)**

- מאפשר לאפס את המיקרו-בקר ולהתחיל את התוכנית מהתחלה.

**\* מחבר ICSP Header**

- מאפשר תכנות ישיר של המיקרו-בקר.

**\* LED Built-in**

- LED מחובר לפין דיגיטלי 13: מאפשר בדיקות מהירות של יציאה דיגיטלית.

**תכנות ה-Arduino**

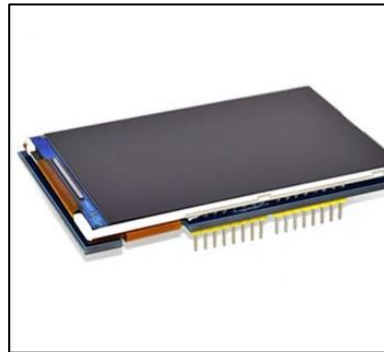
ה-Arduino מתוכנת בעזרת Arduino IDE (סביבת פיתוח משולבת) כותבים את

התוכנית בשפת Arduino ומעלים את הקוד לבקר דרך חיבור ה-USB.

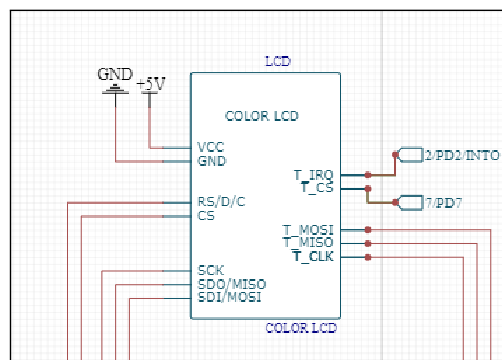


## 4.2. מסך LCD:

### תמונה



### שרטוט חשמלי



### מאפיינים עיקריים

- רזולוציה: 320X240 פיקסלים
- עומק צבע: 16 או 18 ביט
- ממשק: SPI
- מספר התקנים: עד 3
- פינים:
  - איפוס: אתחול מעבד התצוגה
  - AD/RS: קביעת סוג המידע הנשלח (נתון או בקרה).
  - MOSI/SDI: קו כניסת נתונים טורי
  - MISO/SDO: קו יציאת נתונים טורי
  - SCK: קו שעון

CS: הדק אפשר תקשורת

### קביעת צבע פיקסל

ניתן לקבוע את צבע פיקסל יחיד באמצעות שליחת מידע של 16 או 18 ביט. במצב 16 ביט, כל בית מוקצה לצבע אחר (אדום, ירוק או כחול). במצב 18 ביט, ניתן לקבל טווחי צבע.

### תהליך שליחת מידע

1. איפוס התצוגה: שליחת אות Resetl-0

2. קביעת מצב RS:

○ RS=0: שליחת כתובת פיקסל

○ RS=1: שליחת מידע צבע

3. שידור נתונים: שליחת 16 או 18 ביטים (בהתאם למצב ה-RS) דרך קו ה-MOSI

4. העלאת קצה SCK: סימון סיום העברת נתונים

5. הורדת CSL-0: התחלת תקשורת

6. העלאת CSL-1: סיום תקשורת

### 4.3. מסך מגע:

#### תמונה

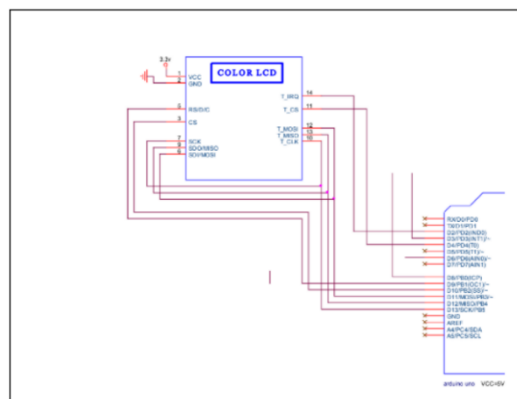


משטח מצופה התנגדות מכוסה בזכוכית דקה (בפי שמתואר באיור), בזמן נגיעה במסך למעשה מתבצעת חלוקה של המסך לשני נגדים R1, R2 בצירה-Y. על מנת ליצור חלוקה גם בציר ה-X הכניסו משטח נוסף כך שמתבצעת חלוקה לשני נגדים גם בציר זה.

החלוקה לשני נגדים בכל נקודת מגע מאפשרת חלוקת מתח, המתח תמיד יהיה יחסי לנגיעה. באמצעות ממיר ADC ברזולוציה של 12 סיביות דו ערוצי (x,y) הנמצא על המסך אנו ממירים את המתח למידע דיגיטלי הנשלח החוצה למעבד בתקשורת טורית SPI.

כעת נותר רק להתאים את התוצאות המתקבלות לכמות הפיקסלים על המסך.

#### שרטוט חשמלי



הרגליים המסומנות ב-T הן האחראיות לתקשורת הטורית, 4 מהן בפרוטוקול SPI, ההדק T\_IRQ יורד ל-'0' בזמן נגיעה על המסך. למעשה באמצעות תקשורת זאת אנו קוראים את ערכיה ממיר לצירים X,Y מהמסך.

## קוד לדוגמא

```
#include <MD_MSSEQ7.h>

// hardware pin definitions - change to suit circuit
const uint8_t DATA_PIN = A5;
const uint8_t RESET_PIN = 6;
const uint8_t STROBE_PIN = 7;

// frequency reading the IC data
const uint32_t READ_DELAY = 50;

MD_MSSEQ7 MSSEQ7(RESET_PIN, STROBE_PIN, DATA_PIN, READ_DELAY);

void setup(void)
{
    MSSEQ7.begin();

    Serial.begin(57600);
    Serial.println("[MD_MSG_SEQ7_Serial]");
}

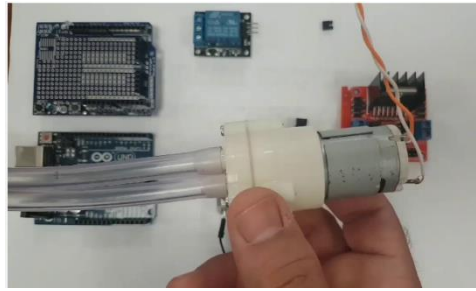
void loop(void)
{
    if (MSSEQ7.read())
    {
        // Serial output
        for (uint8_t i=0; i<MD_MSSEQ7::MAX_BAND; i++)
        {
            Serial.print(MSSEQ7.get(i));
            Serial.print('\t');
        }
        Serial.println();
    }
}
```

דוגמאות לפונקציות מהספרייה TFT שהשתמשתי בהן לתכניות שלי:

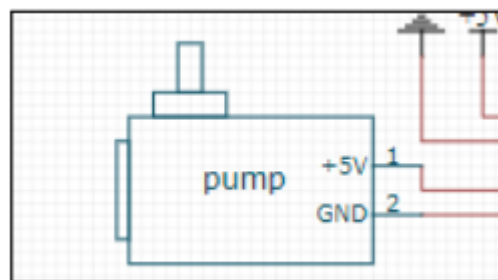
שם הפונקציה	תאור	קלט/פלט
begin	אתחול התצוגה	אין
println	הדפסה הודעה על המסך + ירידת שורה	קלט – מיקום ההודעה X,Y, מחרוזת, גודל הפונט, צבע ההודעה וצבע הרקע.
touched	זיהוי לחיצה על המסך	מחזירה ערך האם נלחץ או לא
setCursor	בחירת מיקום על המסך	מקבלת ערכי X ו Y למיקום על המסך

#### 4.4.משאבת מים:

##### תמונה



##### שרטוט חשמלי

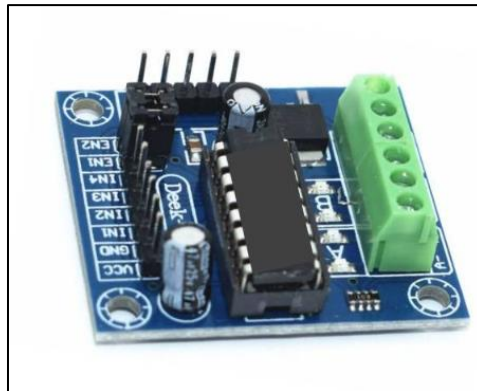


##### קוד לדוגמא

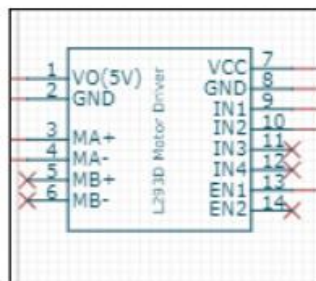
```
void setup() {  
  Serial.begin(9600);  
  pinMode(7, OUTPUT); // Set pin 7 to output  
}  
  
void loop() {  
  digitalWrite(7, HIGH); // Set pin 7 as "high" or turn it on  
  Serial.println("Should be on"); // prints that the pump should be on  
  delay(10000); // 10 second delay  
  digitalWrite(7, LOW); // Set pin 7 as "low" or turn it off  
  Serial.println("Should be off"); // prints that the pump should be off  
  delay(10000); // another 10 second delay  
}
```

## 4.5. דוחף זרם:

### תמונה



### שרטוט חשמלי



### קוד לדוגמא

```
const int pumpPin = A1;

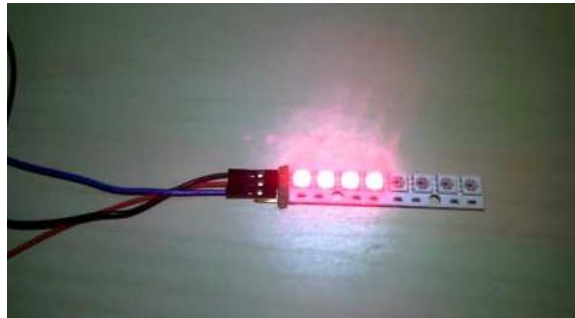
void setup() {
  // הגדרת הפין של המשאבה כפלט
  pinMode(pumpPin, OUTPUT);
}

void loop() {
  // הפעלת המשאבה
  digitalWrite(pumpPin, HIGH);
  delay(10000); // הפעלת המשאבה למשך 5 שניות

  // כיבוי המשאבה
  digitalWrite(pumpPin, LOW);
  delay(5000); // המתנה של דקה אחת לפני הפעלת המשאבה שוב
}
```

## 4.6 בר לדים:

### תמונה



### קוד לדוגמא

```
#include <Adafruit_NeoPixel.h>

#define PIN 3 // input pin Neopixel is attached to

#define NUMPIXELS 8 // number of neopixels in strip

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

int delayval = 100; // timing delay in milliseconds

int redColor = 0;
int greenColor = 0;
int blueColor = 0;
int yellowColor = 0;

void setup() {
  // Initialize the NeoPixel library.
  pixels.begin();
}

void loop() {
  setColor();

  for (int i=0; i < NUMPIXELS; i++) {
    // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
    pixels.setPixelColor(i, pixels.Color(redColor, greenColor, blueColor));

    // This sends the updated pixel color to the hardware.
    pixels.show();

    // Delay for a period of time (in milliseconds).
    delay(delayval);
  }
}

// setColor()
// picks random values to set for RGB
void setColor(){
  redColor = random(0, 255);
  greenColor = random(0,255);
  blueColor = random(0, 255);
  yellowColor = random(0, 255);
}
```

#### **תכונות עיקריות:**

כל לד יכול להפיק כל צבע בטווח ה-RGB (אדום, ירוק, כחול) באמצעות שליטה על עוצמת כל אחד משלושת הצבעים הבסיסיים.  
אמצעות שילוב ערכי ה-RGB ניתן להפיק מגוון רחב של צבעים.

#### **אופן הפעולה:**

רצועת הלדים WS2812B-8 מורכבת מ-8 לדים שניתן לשלוט עליהם באופן דיגיטלי באמצעות פרוטוקול תקשורת ייחודי. כל לד ברצועה מכיל בקר פנימי שמאפשר שליטה בצבע ובעוצמת התאורה של הלד באמצעות אות דיגיטלי.

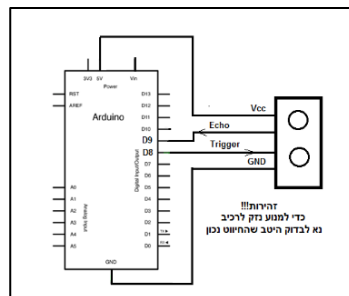


## 4.7. חיישן מרחק:

### תמונה



### שרטוט חשמלי



### קוד לדוגמא

```
const int trigPin = A5;
const int echoPin = A4;

float duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration*0.0343)/2;
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
}
```

### עקרון פעולת החיישן

החיישן מודד מרחק על ידי שידור גלי קול לעבר אובייקט מרוחק, המתנה להד המוחזר מהאובייקט, מדידת הזמן שעבר מרגע השידור ועד קליטת ההד וחישוב המרחק לפי משוואת התנועה במהירות קבועה.

החיישן משדר גלי קול בתדר אולטראסוני, שזה אומר תדר שהוא מעבר ליכולת השמיעה של אוזן אדם. גלי הקול מתקדמים באוויר ואם יש בדרכם עצם מסוים, הגלים מוחזרים חזרה בהתאם לזווית הפגיעה בעצם (אם הפגיעה היא בניצב לעצם אז הגלים יוחזרו באותו כיוון שהגיעו ממנו, אבל אם זווית הפגיעה איננה בניצב הגלים מוחזרים בזווית שווה לזווית הפגיעה אבל בכיוון הרחק מהחיישן).

## 5. תיעוד הפרויקט

### 5.1. תהליך ביצוע הפרויקט:

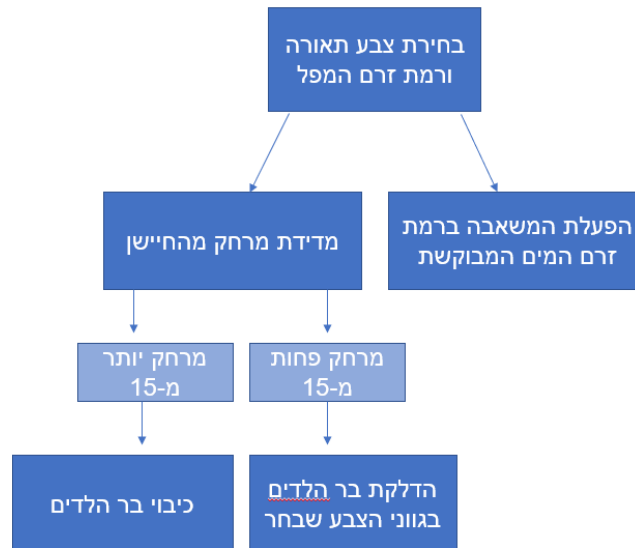
הפעלת כל רכיב בנפרד: משאבת מים, בר לדים, חיישן מרחק.  
אח"כ עבדתי לשנות את המשאבה כך שתעבוד עם הדוחף זרם.  
ואז חיברתי בין הקוד של חיישן המרחק לבר לדים וזה עבד מהמם.  
שלב הבא – הוספתי את המחלקה color וכתבתי את האלגוריתם לסינכרון בחירת המשתמש עם צבעי הלדים.  
ואז הוספתי גם את הקוד של המשאבה שתפעל במקביל והזרם בה יהיה לפי בחירת המשתמש.

### 5.2. בעיות בתהליך הפרויקט:

היו חיישנים שלקח לי הרבה זמן להפעיל אותם.  
חלק מהם כבר עבדו ופתאום לפעמים הפסיקו לעבוד בלי שום סיבה הנראית לעין.  
אתגר גדול היה לי עם כתיבת האלגוריתם של גוני הלדים לפי בחירת המשתמש.

## 6. תוכנה

### 6.1. תרשים זרימה:



### 6.2. פירוט פונקציות:

שם הפונקציה	תפקידה	קלט / פלט
setColor	להגדיל מספר עבור שלושת צבעי הלדים	אין
screenColor	פונקציה שמציגה את המסך של בחירת צבע הלדים	אין
screenMain	פונקציה שמציגה את המסך הראשי	אין
screenWater	פונקציה שמציגה את המסך של בחירת רמת זרם המים	אין

## 6.3 קוד התכנית:

```
#include <Adafruit_NeoPixel.h>
#define PIN 3 // input pin Neopixel is attached to
#define NUMPIXELS 8 // number of neopixels in strip
#include "TFT9341Touch.h"
tft9341touch LcdTouch (10, 9, 7, 2); //cs, dc ,tcs, tirq

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

const int IN_A0 = A2; // analog input
const int IN_D0 = 5; // digital input
int value_A0;
bool value_D0;
int colorLight=10;
int ButtonColor=0;
int delayval = 100; // timing delay in milliseconds
int blueColor=0;
int greenColor=0;
int redColor=0;
const int trigPin = A5;
const int echoPin = A4;
float duration, distance;

struct Color {
    int colorNumber; // מספר הצבע
    uint16_t colorName; // שם הצבע
    bool IsStormy; // האם סופרי
    int rArr[3];
    int gArr[3];
    int bArr[3];
};

Color colors[] = {
    {10, GREEN, false, {0,40,100}, {255,255,255}, {0,40,100}},
    {11, YELLOW, false, {225,255,255}, {220,255,255}, {0,15,90}},
    {12, ORANGE, true, {255,255,255}, {100,150,195}, {0,75,155}},
    {13, RED, true, {255,255,255}, {0,0,100}, {0,40,100}},
    {14, WHITE, false, {255,255,255}, {255,255,255}, {255,255,255}},
    {15, MAGENTA, false, {255,205,240}, {0,100,210}, {40,255,255}},
    {16, BLUE, true, {0,70,120}, {0,70,120}, {255,255,255}},
    {17, PURPLE, true, {100,150,190}, {0,40,130}, {205,255,255}}
};
```

```

void setup() {

  Serial.begin(9600);
  pinMode (IN_A0, INPUT);
  pinMode (IN_D0, INPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  pixels.begin();

  LcdTouch.begin();
  LcdTouch.clearButton();
  LcdTouch.setRotation(0);
  LcdTouch.setTextSize(2);
  LcdTouch.setCursor(40, 40);
  LcdTouch.set(3780, 372, 489, 3811);

  screenMain();
}

void loop() {
  value_A0 = analogRead(IN_A0); // reads the analog input from the IR distance sensor
  value_D0 = digitalRead(IN_D0); // reads the digital input from the IR distance sensor
  Serial.print(" Analogue = ");
  Serial.print(value_A0);
  Serial.print("\t Digital =");
  Serial.println(value_D0);
  delay(100);

  pixels.setPixelColor(0, pixels.Color(0, 0, 0)); // שיהיה כבוי בהתחלה//
  pixels.show();
  uint16_t x, y;
  String str;

  if (LcdTouch.touched())
  {
    LcdTouch.readTouch();
    x = LcdTouch.xTouch;
    y = LcdTouch.yTouch;

    ButtonColor = LcdTouch.ButtonTouch(x, y);
    Serial.println(ButtonColor);

    switch (ButtonColor) {
      case 10:
        colorLight=0;
        break;
      case 11:
        colorLight=1;
        break;
      case 12:
        colorLight=2;
        break;
      case 13:
        colorLight=3;
        break;
      case 14:
        colorLight=4;
        break;
      case 15:
        colorLight=5;
        break;
      case 16:
        colorLight=6;
        break;
      case 17:
        colorLight=7;
        break;
      default:
        colorLight=10;
        break;
    }

    screenMain();
  }
  else if (ButtonNum == 2) "מים" על הכפתור מים//
  {
    screen2();
    screenMain();
  }
}

```

```

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

duration = pulseIn(echoPin, HIGH);
distance = (duration*.0343)/2;
Serial.print("Distance: ");
Serial.println(distance);

if (distance < 15 )
{
  if(colorLight==10)
  {
    setColor();
    for (int i=0; i < NUMPIXELS ; i++)
    {
      pixels.setPixelColor(i, pixels.Color(redColor,greenColor,blueColor));
      pixels.show();
      delay(delayval);
    }
  }
  else
  {
    for(int j=0;j<3;j++)
    {
      redColor = colors[colorLight].rArr[j];
      greenColor = colors[colorLight].gArr[j];
      blueColor = colors[colorLight].bArr[j];
      for (int i=0; i < NUMPIXELS ; i++)
      {
        pixels.setPixelColor(i, pixels.Color(redColor,greenColor,blueColor));
        Serial.println(redColor);
        Serial.println(greenColor);
        Serial.println(blueColor);
        pixels.show();
        delay(delayval);
      }
    }
  }
}
else
{
  // מכתבת את הבר לדים אם אין מישהו במרחק הקרוב
  for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(0, 0, 0));
  }
  pixels.show();
}
delay(100);
}

void screenMain() {
  LcdTouch.fillScreen (BLACK);
  LcdTouch.printheb(25, 40, "מערכת השקיה אוטומטית", 2, WHITE);
  LcdTouch.drawButton(1, 15, 90, 290, 40, 10, RED, WHITE, "הרואת", 2); // NumButton, x, y, width, height, z, Color, textcolor, label, textsize);
  LcdTouch.drawButton(2, 15, 140, 290, 40, 10, RED, WHITE, "סימ", 2);
}

void screen1() {
  LcdTouch.fillScreen (BLACK);
  LcdTouch.setTextColor(WHITE);
  LcdTouch.setTextSize (1);
  LcdTouch.setCursor (45, 30);
  LcdTouch.setTextSize (3);
  LcdTouch.print ("הרואת יעבד רחוב");
  LcdTouch.drawButton(10, 15, 90, 60, 60, 10, GREEN, WHITE, "", 2);
  LcdTouch.drawButton(11, 90, 90, 60, 60, 10, YELLOW, WHITE, "", 2);
  LcdTouch.drawButton(12, 165, 90, 60, 60, 10, ORANGE, WHITE, "", 2);
  LcdTouch.drawButton(13, 240, 90, 60, 60, 10, RED, WHITE, "", 2);
  LcdTouch.drawButton(14, 15, 165, 60, 60, 10, WHITE, WHITE, "", 2);
  LcdTouch.drawButton(15, 90, 165, 60, 60, 10, MAGENTA, WHITE, "", 2);
  LcdTouch.drawButton(16, 165, 165, 60, 60, 10, BLUE, WHITE, "", 2);
  LcdTouch.drawButton(17, 240, 165, 60, 60, 10, PURPLE, WHITE, "", 2);
  while (!LcdTouch.touched());
}

```

```

void screen2() {
  LcdTouch.fillScreen (BLACK);
  LcdTouch.setTextColor(WHITE);
  LcdTouch.setCursor (30, 30);
  LcdTouch.setTextSize (2);
  LcdTouch.print ("הייקשחל סימ תומכ רחוב");
  LcdTouch.drawButton(20, 40, 80, 240, 40, 15, RED, WHITE, "הלודג תומכ", 2);
  LcdTouch.drawButton(21, 40, 130, 240, 40, 15, RED, WHITE, "תינוניב תומכ", 2);
  LcdTouch.drawButton(22, 40, 180, 240, 40, 15, RED, WHITE, "הנטק תומכ", 2);

  while (!LcdTouch.touched());
}

void setColor(){
  redColor = random(0, 255);
  greenColor = random(0,255);
  blueColor = random(0, 255);
}

```

---

## 7. סיכום ומסקנות

מאד נהניתי מהעבודה על הפרויקט ואני מרגישה שזה נתן לי המון ידע והבנה בחומרה. הדהים אותי לראות איך התוכנה והחומרה עובדים יפה כל כך יחד. כל רכיב שעבד גרם לי לאושר וסיפוק רב. ההתקדמות הייתה צעד אחר צעד כשכל אחד גורם להנאה מרובה ויותר התעמקות בפרטים. חבל שלא היה לי זמן ויכולת להתעסק עם עוד חיישנים ולהכיר עוד קודים ורכיבים. הפרויקט העלה לי את הביטחון העצמי ונתן לי תחושה של הצלחה. המקצוע בכלל היה מאד נחמד ומגוון, משהו אחר ושובר שיגרה.

## 8. פיתוחים עתידיים

נוסיף לצמחייה בצדדים גם השקיה אוטומטית ע"י החיישן לחות אדמה.

## 9. ביבלוגרפיה

- \* מידע על החיישנים לקחתי מהאתר של Arduino [./https://www.arduino.cc](https://www.arduino.cc).
- \* השתמשתי בחומרים מהספר של מכללת אורט גבעת רם – מגמת רובוטיקה.
- \* שרטוט חשמלי יצרתי באתר: <https://easyeda.com/iw>.