



שם תלמידה: תמר מאור

ת.ז.: 327798377

מקום לימודים: סמינר "בית יעקב רכסים".

שם המנחה: המורה איילה עמור

תאריך הגשה: יוני 2025

## תוכן עניינים:

1.....	תוכן עניינים:
3.....	מבוא:
3.....	הצעת פרויקט
3.....	שם הפרויקט:
3.....	תיאור הפרויקט:
3.....	רכיבי חומרה:
3.....	שלבי הפרויקט:
3.....	קהל יעד:
4.....	רקע לפרויקט:
4.....	מטרת המערכת:
4.....	סקירה ספרותית
4.....	המצב כיום:
4.....	ניתוח חלופות מערכת:
4.....	ספריות
7.....	בקרים
7.....	בקר ESP32
8.....	מנוע סרוו
10.....	חיישן מגע
11.....	חיישן מים
12.....	חיישן אולטראסוני
13.....	מצלמת WIFI
14.....	מסך LCD
15.....	תיאור החלופה הנבחרת
15.....	מטרות ויעדים
16.....	אתגרים
16.....	תיעוד
16.....	תרשים חיבורים
17.....	חיבורים ואופן הפעולה
17.....	תיעוד קטעי קוד
26.....	חיבורים
26.....	תיאור הארכיטקטורה
26.....	תיאור פרוטוקולי תקשורת
27.....	תרשים זרימה
28.....	מדריך למשתמש

- 28..... סיכום ומסקנות
- 28..... בבלוגרפיה
- 28..... תודות

## מבוא:

### הצעת פרויקט

שם הפרויקט: פח אלקטרוני.

### תיאור הפרויקט:

פיתוח פח אשפה חכם, הפח נפתח לבד כשאדם מתקרב אליו, ללא צורך בפתיחה פיזית של אדם.

הוא מזהה מתי הוא התכולה מלאה בפח ושולח התראה מתאימה

אם יש רטיבות בתחתית הפח, הוא מפעיל ייבוש עצמי באופן אוטומטי, לשמירה על ניקיון וריח נעים.

המכסה נפתח בצורה מסודרת רק כאשר מזוהה תנועה קרובה. בנוסף, משולבת בפח מצלמה שבודקת מה מנסים לזרוק. אם מזוהה בקבוק – הפח נשאר סגור, כדי לעודד השלכה למחזורית.

המטרה של הפרויקט היא להפוך את השימוש בפח ליותר נוח, נקי, ומתחשב בסביבה – בעזרת טכנולוגיה פשוטה וזמינה.

### רכיבי חומרה:

- חיישן מגע
- מנועי סרוו
- חיישן מים
- חיישן מרחק אולטראסוני
- מסך OLED
- Esp32
- מצלמת WIFI
- חוטים
- בית בטריות

### שלבי הפרויקט:

- חיבור חיישן אולטראסוני, מגע, מנועי סרוו, מים והפעלת מצלמה.
- כתיבת קוד להפעלת החיישנים.
- סנכרון הנתונים המתקבלים בצורה אסינכרונית.
- תצוגה במסך
- חיבור הפח לבקר ESP.

### קהל יעד:

הפח מיועד לשימוש ביתי, לאנשים שאוהבים ניקיון גם בזמן בישול- פתיחת הפח ללא מגע יד אדם וייבוש אוטומטי, וכן לאנשים שמעודדים סביבה ירוקה- למניעת זריקת בקבוקים לפח האשפה.

## רקע לפרויקט:

אני אוהבת טכנולוגיה ודברים חכמים לבית, ורציתי לבנות פח שיעזור לשמור על ניקיון וסדר – בלי לגעת בפח כשהיד מלוכלכת.

בנוסף, חשוב לי לשמור על הסביבה, לכן הוספתי זיהוי של בקבוקים – כדי שהפח לא ייפתח כשזורקים משהו שצריך למחזר. כך נוצר הפח החכם – נוח, נקי, לעידוד סביבה ירוקה.

## מטרת המערכת:

מטרת המערכת היא לפתח פח חכם שנפתח אוטומטי בזיהוי קרבת אדם, וכן מפעיל ייבוש אוטומטי במקרה של רטיבות תחת השקית, הוא בודק האם הפח מלא ומציג הודעה מתאימה למשתמש באחוזים של מילוי רמת הפח.

וכן זיהוי הכנסת בקבוק לפח ומניעת פתיחת המכסה.

## סקירה ספרותית

### המצב כיום:

כיום קיימים פחים אלקטרוניים חכמים שנפתחים אוטומטית כאשר מתקרבים אליהם. סוגים אלו של פחים משמשים בעיקר במקומות ציבוריים או אזורים עתירי תנועה, כדי לשמור על היגיינה ולאפשר זריקה נוחה של פסולת – מבלי לגעת בפח עצמו.

בנוסף, יש כיום פחים חכמים עם חיישנים שמזהים מתי הפח מלא, ושולחים התרעה לניקוי או לריקון. הפחים האלו מסייעים בניהול יעיל יותר של האשפה, במיוחד במקומות עם תנועה מרובה של אנשים.

### ניתוח חלופות מערכתי:

בתחילת הדרך רציתי מאוד לממש מנגנון של קשירת שקית אוטומטית.

תכננתי להשתמש במנועי סרוו ולהצמיד וו לכל אחד מהם, כך שיתפסו את חורי הידית של השקית וייצרו קשר ע"י תנועה מצטלבת.

עם זאת, הבנתי במהרה שהפתרון הזה דורש שימוש בשתי זרועות נפרדות, מה שמוביל למבנה גדול, מורכב ומסורבל – לא מתאים לפח ביתי סטנדרטי, אלא יותר למוסדות או לחצרות. לכן עברתי לכיוון נוסף אחר ובחרתי לזהות בקבוקי מיחזור.

חיברתי מנועי סרוו לפתיחת הפח ולייבוש הרטיבות והתקדמתי לחפש ספריות לתהליכונים.

חיפשתי ספריות של תהליכונים הקיימות, שיכולות לעזור לי ולאט לאט, התחלתי ליצור מערכת הפועלת בצורה אסינכרונית תודות לתהליכונים הפועלים בה.

כדי להתריע האם תכולת הפח מלאה, בחרתי להציג את ההודעה במסך OLED, ובצורה כזו להציג האם הפח מלא.

## ספריות

ESP32Servo - ספרייה לשליטה במנועי סרוו

ספריית ESP32Servo היא ספרייה סטנדרטית שמפשטת את משימת השליטה במנועי סרוו. מנועי סרוו משמשים בדרך כלל להשגת שליטה זוויתית מדויקת באובייקטים שונים, מה שהופך אותם לאידיאליים עבור יישומים כמו רובוטיקה, אוטומציה והתקנים הנשלטים מרחוק.

הפונקציה write() משמשת להגדרת הזווית של מנוע הסרוו למיקום ספציפי בין 0 ל-180 מעלות. פונקציה זו מאפשרת לכם לשלוט במיקום מנוע הסרוו על ידי ציון הזווית הרצויה. הפונקציה read() מאפשרת לכם לקרוא את הזווית הנוכחית של מנוע הסרוו. זה יכול להיות שימושי למטרות ניטור או משוב בפרויקטים.

הפונקציה attached() מחזירה ערך בוליאני המציין אם מנוע סרוו מחובר כעת לפין במיקרו-בקר.

הפונקציה Microseconds(microseconds): פונקציה זו מספקת שליטה מדויקת יותר על מנוע הסרוו בכך שהיא מאפשרת להגדיר את רוחב הפולס במיקרו-שניות. זה יכול להיות מועיל לכוונון עדין של מיקום מנוע הסרוו.

## esp32-camera

ה-ESP32-CAM הוא כרטיס פיתוח המשלב את בקר המיקרו ESP32 עם מודול מצלמה מובנה (לרוב מסוג OV2640), ויכולות קישוריות Wi-Fi ו-Bluetooth.

שימושים:

ספריית ה-ESP32-CAM מפשטת את תהליך שילוב המצלמה בפרויקטים. היא מאפשרת למפתחים לקבל תמונות או וידאו בזמן אמת ולהעבירם דרך רשת ה-Wi-Fi. הספרייה מטפלת בפעולות הטכניות המורכבות הקשורות להפעלת המצלמה, כגון הגדרת הפנים, ניהול זיכרון ה-PSRAM וטיפול בפורמטי תמונות. כך, מפתחים יכולים להתמקד בלוגיקת היישום של הפרויקט עצמו, במקום בפרטי המימוש הנמוכים של המצלמה.

פעולות עיקריות

הספרייה מציעה מספר פונקציות מרכזיות לשליטה במצלמה:

`esp_camera_init()`: פונקציה זו מאתחלת את המצלמה ומגדירה את הפרמטרים שלה. לדוגמה, היא מאפשרת לקבוע את רזולוציית התמונה וסוג הדחיסה (לדוגמה, JPEG).

`esp_camera_fb_get()`: פונקציה זו משמשת לשליפת תמונה (frame buffer) מהמצלמה. ניתן להשתמש בה לקבלת תמונה בודדת או כחלק מזרם וידאו. הפונקציה מחזירה מצביע למבנה נתונים המכיל את נתוני התמונה.

`esp_camera_fb_return(fb)`: לאחר עיבוד התמונה, פונקציה זו משחררת את הזיכרון שהוקצה עבורה. פעולה זו חיונית לניהול זיכרון יעיל ולשמירה על יציבות המערכת לאורך זמן.

`esp_camera_deinit()`: כאשר המצלמה אינה נחוצה יותר, פונקציה זו מבצעת ניתוק וניקוי מסודרים של משאבי המצלמה. זה משחרר את המשאבים לשימושים אחרים במערכת.

## ספריית LiquidCrystal\_I2C:

הספרייה LiquidCrystal\_I2C מפשטת את השליטה במסכי LCD (כמו מסכי x216 או x420) על ידי שימוש בתקשורת I2C. זה מאפשר לחבר את המסך למיקרו-בקר (כמו ארדואינו) באמצעות ארבעה חוטים בלבד, במקום שימוש בהרבה חוטים.

פעולות עיקריות:

```
;;LiquidCrystal_I2C lcd()
```

יוצר את אובייקט ה-LCD ומגדיר את כתובת ה-I2C שלו (לרוב x270 או x3F0), מספר העמודות ומספר השורות של המסך.

```
;;lcd.init()
```

מפעיל את המסך.

```
;;lcd.backlight(); / lcd.noBacklight
```

מדליק או מכבה את תאורת הרקע של המסך.

```
;;lcd.print()
```

מציג טקסט על המסך מהמיקום הנוכחי של הסמן.

```
;;lcd.setCursor()
```

מזיז את הסמן למיקום ספציפי על המסך (עמודה ושורה, החל מ-0).

```
;;lcd.clear()
```

מנקה את כל התוכן מהמסך ומחזיר את הסמן למיקום ההתחלתי (0,0).

```
;;lcd.home()
```

מחזיר את הסמן למיקום ההתחלתי (0,0) מבלי למחוק את תוכן המסך.

```
;;lcd.noDisplay(); / lcd.display
```

מכבה או מדליקה את תצוגת הטקסט, אך משאירה את תאורת הרקע ללא שינוי.

```
;;lcd.noCursor(); / lcd.cursor
```

מכבה או מדליקה את הסמן המהבהב.

```
;;lcd.scrollDisplayLeft(); / lcd.scrollDisplayRight
```

מגלגלת את כל תוכן המסך שמאלה או ימינה.

```
;;lcd.leftToRight(); / lcd.rightToLeft
```

מגדירה את כיוון כתיבת הטקסט משמאל לימין או מימין לשמאל.

## WiFi.h

ספרייה זו מאפשרת לבקר ה-ESP32 להתחבר לרשת אלחוטית (Wi-Fi).

ניתן להשתמש בה גם במצב שרת וגם במצב לקוח. הספרייה כוללת פונקציות כמו WiFi.begin() לצורך התחברות לרשת, WiFi.status() לבדוק את סטטוס החיבור, ו-WiFi.localIP() כדי לקבל את כתובת ה-IP שהוקצתה לבקר. הספרייה תומכת בפרוטוקולי אבטחה כמו WPA/WPA2 וכן בהגדרת IP סטטי או DNS.

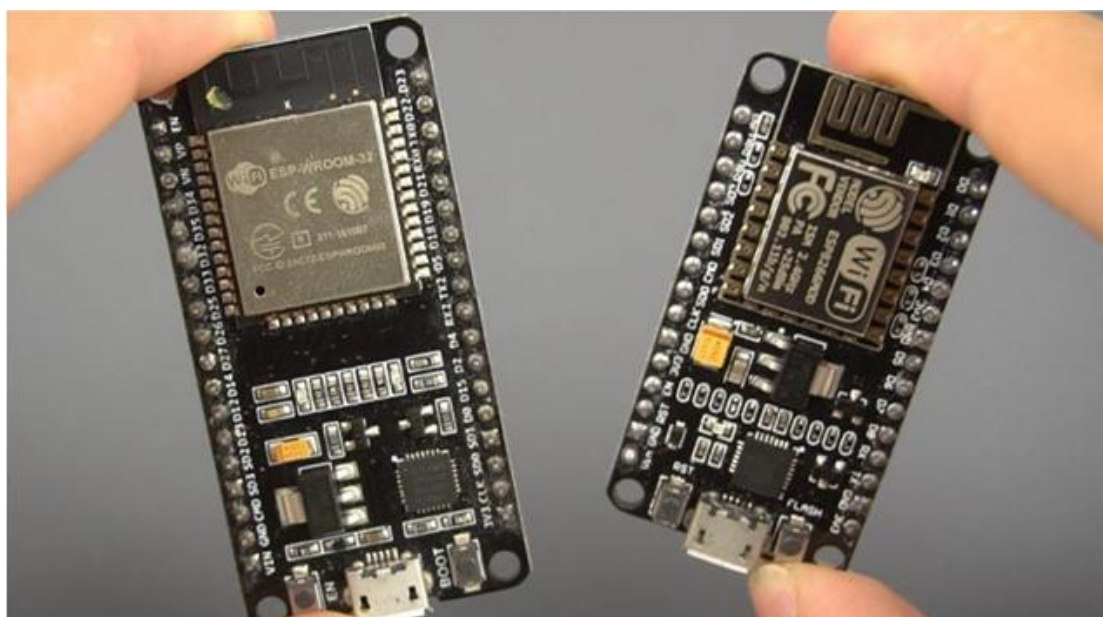
### ניהול הנתונים בצורה אסינכרונית

ה-ESP32 מגיע עם תמיכה מובנית ב-FreeRTOS (מערכת הפעלה בזמן אמת).

זה מאפשר ניהול תהליכים בפח החכם.

## בקרים

### בקר ESP32



ה-ESP32 הוא בקר מתקדם ופופולרי ממשפחת הבקרים של חברת Espressif. הוא משמש במגוון רחב של יישומים בזכות ביצועיו הגבוהים ותכונותיו הרבות.

תכונותיו המרכזיות של ה-ESP32:

מעבד כפול ליבה:

ה-ESP32 כולל שתי ליבות מעבד Xtensa® 32-bit LX6 עם תדר שעון של עד 240 MHz. זה מאפשר ביצועים גבוהים ותמיכה במשימות מקביליות.

קישוריות:

Wi-Fi: תמיכה בתקן 802.11 b/g/n, המאפשר קישוריות אלחוטית עם טווח רחב.

Bluetooth: תמיכה ב-Bluetooth 4.2 (כולל Classic ו-BLE), לחיבור למגוון מכשירים אחרים.

זיכרון:

SRAM: בנפח של עד 520 KB.

זיכרון חיצוני: קיימת אפשרות לשימוש בזיכרון SPI Flash חיצוני.



כניסות/יציאות רב-תכליתיות (GPIO):

ה-ESP32 כולל עד 34 פני GPIO. ניתן לתכנת אותם ולהשתמש בהם לחיבור מגוון רחב של חיישנים ורכיבים חיצוניים.

ממשקים נוספים:

UART, SPI, I2C, I2S, CAN, ADC, DAC, PWM.

יכולות ניהול צריכת חשמל:

כולל מצבי חיסכון באנרגיה (למשל, Deep Sleep, Light Sleep ועוד), המאפשרים יישומים חסכוניים באנרגיה.

הסבר רכיבים:

## מנוע סרוו



מנוע סרוו הוא סוג של מנוע שיכול להסתובב בדיוק רב. מנוע זה מורכב ממעגל בקרה המספק משוב על המיקום הנוכחי של ציר המנוע. משוב זה מאפשר למנוע הסרוו להסתובב בדיוק רב. אם אתה רוצה לסובב אובייקט בכמה זוויות או מרחק מסוים, אז אתה משתמש במנוע סרוו. זהו לא מנוע פשוט המורכב ממנוע שפועל דרך מנגנון סרוו, אלא מופעל על ידי אספקת כוח.

הבקר (ארדואינו או ESP32, ואפילו בקר סרוו פשוט) שולח אותות למנוע הסרוו, שקובעים לאיזה מיקום עליו להסתובב.

הבקר מקבל (Pulse Width Modulation) PWM – שזהו אות אפנון רוחב פעימה, כלומר הוא שולח פולסים של מתח. ניתן לשנות את רוחב הפולס הזה, בדומה לאדם שלוחץ על מתג כדי להדליק ולכבות את האור. ככל שהמתג נלחץ לזמן ארוך יותר, כך פולס החשמל יהיה ארוך יותר.

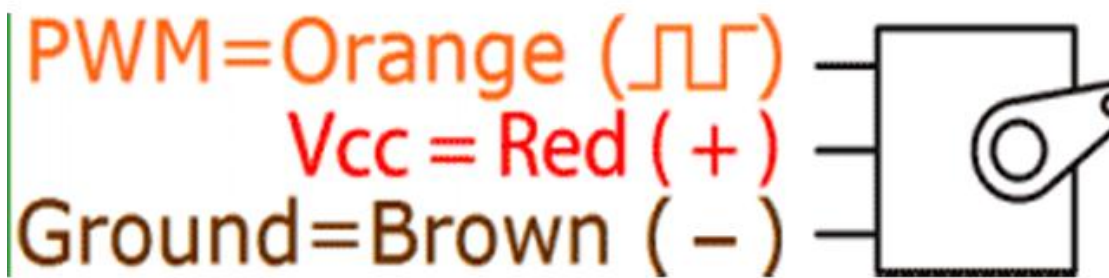
פולסים אלו נשלחים כל 20 מילישניות (כלומר, 50 פולסים לשנייה).

זוהי מערכת בלולאה סגורה שבה היא משתמשת במערכת משוב חיובי כדי לשלוט בתנועה ובמיקום הסופי של הפריט. כאן ההתקן נשלט על ידי אות משוב שנוצר על ידי השוואת אות קלט הייחוס ואות הפלט.

כאן, אות קלט הייחוס מושווה לאות המוצא של הייחוס ואות שלישי מסופק על ידי מערכת המשוב. אות שלישי זה פועל כאות קלט לשליטה בהתקן. אות זה קיים כל עוד קיים הבדל בין אות קלט הייחוס לאות מוצא הייחוס או אות המשוב שנוצר. אז המשימה היא פלט של המערכת בערך הרצוי בנוכחות רעש.

המטרה העיקרית של מנגנון סרוו היא לשמור על דיוק ולמנוע רעשים.

חיבורים:



מספר חוט	צבע חוט	תיאור
1	חום	חוט הארקה מחובר להארקה של המערכת
2	אדום	מפעיל את המנוע בדרך כלל אדום בשימוש V5+
3	כתום	אות PWM* ניתן דרך חוט זה כדי להניע את המנוע

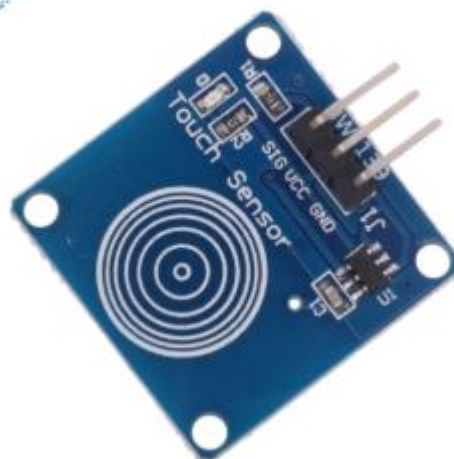
PWM (Pulse Width Modulation): שליטה על כוח חשמלי  
 PWM משמש לשליטה בכוח החשמלי בתוך סליל המנוע. עוצמת היציאה נשלטת על ידי הפעלה או כיבוי חוזרים ונשנים של המתח הקבוע, עם תקופה קבועה של הפולסים. בדרך כלל, מנוע סרוו מקבל שני סוגי קלט ומייצר פלט אחד: קלט:

PWM (Pulse Width Modulation): זהו סוג הקלט הנפוץ ביותר לשליטה במנועי סרוו. רוחב הפולס: רוחב הפולס קובע לאיזה מיקום מנוע הסרוו צריך לנוע. באופן כללי, רוחב הפולס משתנה בין 1 מ"ש ל-2 מ"ש, עם תדר של סביב 50 הרץ. אספקת כוח:

מנועי סרוו דורשים אספקת כוח (בדרך כלל מתח DC) כדי לפעול. אספקת הכוח מפעילה את הרכיבים הפנימיים של המנוע ומספקת את המומנט הנדרש להזזת גלגל המנוע, אשר בתורו מזיז את החפץ המחובר למנוע לזווית הרצויה.

## חיישן מגע

DIY-VICTOR



חיישן מגע קיבולי הוא רכיב המזהה מגע אנושי ללא צורך בלחיצה פיזית. הוא דומה למסכי מגע בטלפונים.

ממה הוא מורכב?

החיישן מורכב מלוח מעגל מודפס (PCB) קטן, שבמרכזו ממוקם משטח מגע ייעודי. משטח זה, לרוב מעוצב בצורת ספירלה או עיגולים קונצנטריים, משמש למעשה כאלקטרודה. הליבה של החיישן היא שבב ייעודי (כמו ה-TTP223 או דגמים דומים) האחראי על זיהוי השינויים החשמליים הנובעים ממגע. בנוסף לכך, החיישן כולל מספר פני חיבור סטנדרטיים: VCC (אספקת מתח), GND (הארקה), ו-SIG (פין האות שמספק את הפלט לבקר). לעיתים קרובות, קיימים על הלוח גם רכיבים אלקטרוניים פסיביים מינימליים (כמו נגדים או קבלים) שתפקידם לייצב את פעולת השבב ולשפר את רגישות ודיוק הזיהוי.

איך הוא עובד?

עקרון הפעולה של חיישן מגע קיבולי מבוסס על מדידת קיבוליות חשמלית. קיבוליות היא היכולת של עצם או מעגל לאגור מטען חשמלי.

מדידה בסיסית (מצב רגיל): כאשר אין מגע, החיישן מודד באופן רציף את הקיבוליות הבסיסית של משטח המגע שלו ושל סביבתו הקרובה. ערך זה משמש כנקודת ייחוס.

שינוי קיבוליות (מגע): כאשר אצבע אנושית (או כל אובייקט מוליך אחר, כמו חפץ מתכתי) מתקרבת או נוגעת במשטח החיישן, היא משנה את השדה החשמלי סביב החיישן. גוף האדם, בהיותו מוליך חשמלי, מוסיף קיבוליות למערכת הכוללת. שינוי זה גורם לעלייה בקיבוליות הנמדדת.

זיהוי סף: שבב ה-TTP223B מתוכנת לזהות שינוי משמעותי בקיבוליות. ברגע שהקיבוליות הנמדדת עולה מעל סף מוגדר מראש, השבב מזהה זאת כ"מגע" או "לחיצה".

הפקת אות יציאה: בתגובה לזיהוי המגע, החיישן משנה את מצבו של פין ה-SIG שלו. לדוגמה, במצב רגיל (ללא מגע) הפין עשוי להיות במתח חשמלי נמוך (LOW), וברגע של מגע הוא יעבור למתח גבוה (HIGH). המיקרו-בקר (כמו ארדואינו או ESP32) שאליו מחובר פין ה-SIG קורא את שינוי המתח הזה ומפעיל פעולה מתאימה בקוד (לדוגמה, הדלקת נורה, הפעלת מנוע, או שליחת נתונים).

למה הוא משמש?

- החלפת כפתורים מכניים: השימוש העיקרי הוא כתחליף לכפתורים לחיצים מסורתיים. זה מפחית בלאי מכני, משפר עמידות לאבק ולחות, ומאפשר עיצוב מוצרים אלגנטיים יותר.
- בקרת תאורה: הדלקה, כיבוי או עמעום של גופי תאורה באמצעות נגיעה קלה.
- מכשירי חשמל ביתיים: הפעלת מכשירים כמו מיקרוגלים, תנורים, או מדיחי כלים באמצעות ממשקי מגע.
- פרויקטים אינטראקטיביים: שילוב במשחקים, צעצועים, אמנות אינטראקטיבית, או מתקני תצוגה שבהם נדרשת תגובה למגע.

## חיישן מים



איך הוא עובד?

חיישן מפלס מים הוא רכיב אלקטרוני חשוב שמטרתו **לזהות מים** ולמדוד את **גובהם**. הוא פתרון פשוט ויעיל לפרויקטים שונים, מבית חכם ועד לתעשייה.

**ממה הוא מורכב?**

החיישן עשוי מ**לוח אלקטרוני (PCB)** על הלוח יש **פסים מתכתיים חשופים** (בדרך כלל נחושת) שהם למעשה **אלקטרודות**. חלק מהפסים מחוברים לחשמל) כמו (VCC וחלקם מחוברים להארקה (GND) או לפין מיוחד לבקר) ארדואינו (ESP32/לעיתים יש גם רכיבים קטנים נוספים על הלוח, כמו נגדים, שעוזרים לייצב את המדידה.

עיקרון הפעולה של חיישן מפלס מים מתבסס על מוליכות חשמלית של מים. כאשר מים (או נוזל אחר בעל מוליכות חשמלית מסוימת) באים במגע עם הפסים המוליכים של החיישן, הם יוצרים מעגל חשמלי בין הפסים.

התהליך מתרחש באופן הבא:

מגע עם מים: ככל שיותר פסים על גבי החיישן טבולים במים, כך נוצרים יותר נתיבים חשמליים דרך המים בין האלקטרודות.

שינוי התנגדות: יצירת נתיבים חשמליים אלו משנה את ההתנגדות החשמלית הכוללת בין פסי המדידה. ככל שיותר פסים מכוסים במים, כך ההתנגדות הכוללת יורדת.

קריאת מתח: שינוי זה בהתנגדות מתורגם לשינוי במתח החשמלי הנקרא על ידי פין הכניסה של המיקרו-בקר (לדוגמה, פין אנלוגי בארדואינו או ESP32). המיקרו-בקר מקבל ערך מתח שונה בהתאם לכמות המים שנוגעת בחיישן.

ערך נמוך (או אפס): אין מים או כמות מים קטנה מאוד.

ערך גבוה: כמות מים גדולה יותר.

זיהוי וקביעת מפלס: המיקרו-בקר מפרש את ערך המתח המתקבל. על ידי כיול מתאים, ניתן לקבוע האם יש נוכחות מים כלל (זיהוי של "רטוב/יבש") או למדוד את גובה המפלס באופן יחסי (לדוגמה, 0% מים, 25% מים, 50% מים וכו').

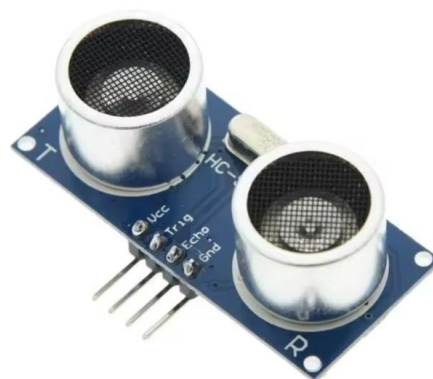
למה משתמשים בו?

חיישני מפלס מים שימושיים במגוון רחב של יישומים:

מערכות השקיה אוטומטיות: זיהוי מתי אדמה יבשה ודורשת השקיה, או ניטור מפלס המים במיכלי השקיה.

התרעה מפני הצפות ונזילות: התקנת החיישן במקומות רגישים כמו מרתפים, חדרי שירות או תחת כיורים יכולה לספק התראה מוקדמת על נזילות או הצפות.

## חיישן אולטראסוני

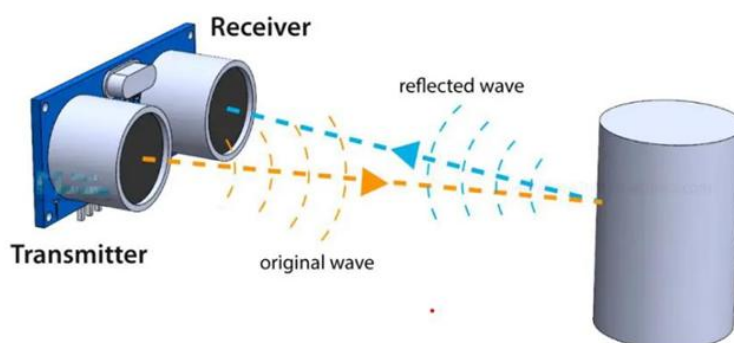


חיישן המרחק האולטרה-סוני הוא מכשיר המשתמש בגלי קול בתדרים גבוהים (מעל לטווח הנשמע של בני אדם, 20 קילו-הרץ) כדי למדוד מרחקים ולזהות עצמים. הוא פועל על בסיס עקרון ההד, בדומה לאופן שבו עטלפים ודולפינים מנווטים ומזהים עצמים בסביבתם.

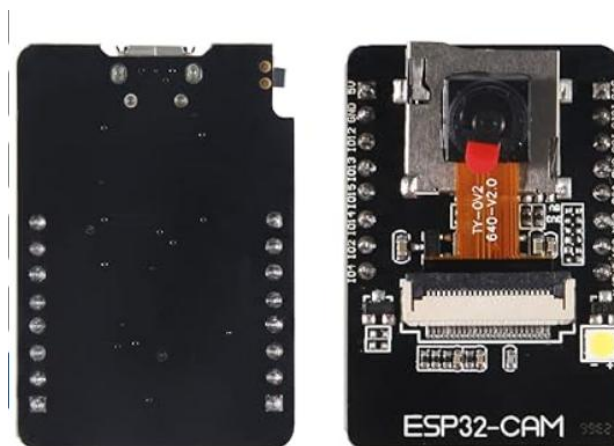
משדר: שולח 8 מחזורים של גל קול בתדר 40 קילו-הרץ לחלל.

קולט: מקבל את ההד שחוזר מהאובייקט שבו נתקל גל הקול שנשלח.

ניתן למדוד את המרחק בין העצם לאות שנשלח, לפי הזמן שלקח להד לחזור.



## מצלמת WIFI



מודול ה-ESP32-CAM הוא מודול קטן וחסכוני באנרגיה המשלב מיקרו-בקר ESP32 עם מצלמה. הוא פופולרי במיוחד עבור פרויקטים של אינטרנט של הדברים (IoT) הדורשים יכולות וידאו או תמונה.

ממה היא מורכבת?

ה-ESP32-CAM מורכבת בעיקר מהרכיבים הבאים:

שבב ESP32: זהו המוח של המודול. ה-ESP32 הוא מיקרו-בקר חזק עם יכולות Wi-Fi ו-Bluetooth מובנות. הוא אחראי על עיבוד הנתונים מהמצלמה, תקשורת אלחוטית, וביצוע קוד התוכנה שאתה טוען עליו.

מודול מצלמה (לרוב OV2640 או OV7670): זוהי המצלמה עצמה, שאחראית על לכידת תמונות או וידאו. המצלמה המובנית היא בדרך כלל בעלת רזולוציה בינונית, המתאימה למגוון רחב של יישומים.

שקע כרטיס זיכרון MicroSD: מאפשר שמירה של תמונות, סרטונים או נתונים אחרים על כרטיס זיכרון חיצוני.

נורת LED: נורת פלאש קטנה שיכולה לשמש לתאורה במצבי תאורה חלשה.

אנטנת Wi-Fi: חלק מהמודולים כוללים אנטנה מובנית, ואחרים מאפשרים חיבור אנטנה חיצונית לטווח קליטה טוב יותר.

פיני GPIO: אלו הם פינים ניתנים לתכנות המאפשרים חיבור לרכיבים חיצוניים אחרים כמו חיישנים, מפעילים (actuators) ועוד.

מחבר Micro USB (בחלק מהגרסאות): משמש לאספקת חשמל ולתכנות המודול.

איך היא עובדת?

אספקת חשמל ותכנות: המודול מקבל מתח כלל דרך פיני החשמל או מחבר Micro USB. ניתן לתכנת את שבב ה-ESP32 באמצעות סביבות פיתוח כמו Arduino IDE, MicroPython, או ESP-IDF. הקוד שאתה טוען קובע את התנהגות המודול.

לכידת תמונה/וידאו: שבב ה-ESP32 מפעיל את המצלמה. המצלמה לוכדת את התמונה או הווידאו ומעבירה את הנתונים לשבב ה-ESP32.

עיבוד נתונים: ה-ESP32 מעבד את נתוני התמונה/וידאו. הוא יכול לבצע דחיסה, לשנות רזולוציה, או לבצע ניתוח כלשהו על הנתונים.

שמירה/שידור:

שמירה על כרטיס SD: ה-ESP32 יכול לשמור את התמונות/וידאו על כרטיס ה-MicroSD המוכנס למודול.

שידור אלחוטי: באמצעות יכולות ה-Wi-Fi המובנות, ה-ESP32 יכול לשדר את נתוני התמונה/וידאו לרשת מקומית, לשרת בענן, או למכשיר אחר (כמו טלפון סלולרי או מחשב). השידור יכול להיות בזרם וידאו (streaming) או כתמונות בודדות.

אינטראקציה עם רכיבים חיצוניים: אם חוברו חיישנים או מפעילים לפיני ה-GPIO, ה-ESP32 יכול לקבל מהם קלט או לשלוט בהם, בהתאם לקוד שכתבת. לדוגמה, הוא יכול להדליק נורת LED כשהוא מזהה תנועה.

השימושים בה:

מצלמת אבטחה ביתית, זיהוי פנים, ומעקב וידאו.

## מסך LCD



ממה הוא מורכב?

מסך LCD 16x2:

פנל תצוגה: זהו החלק המלבני הכחול שמציג את הטקסט. הוא מורכב מקריסטלים נוזליים שמגיבים למתח חשמלי כדי ליצור את התווים.

בקר LCD פנימי: שבב קטן המוטמע מאחורי הפנל, שאחראי על קבלת נתונים ופקודות ותרגומם לתצוגה על המסך.

תאורה אחורית (Backlight): נורות LED הממוקמות מאחורי הפנל ומאירות אותו כדי לאפשר ראות גם בתאורה חלשה.

פיני חיבור (16 פינים בחלק העליון): פינים אלו משמשים לחיבור המסך, אך במקרה זה הם מתחברים למודול המתאם I2C ולא ישירות למיקרו-בקר.

מודול מתאם I2C (הכרטיס השחור הקטן):

שבב בקר I/O (כמו PCF8574): זהו המוח של המתאם. תפקידו לקבל נתונים מפרוטוקול I2C (שדורש רק 2 פניי נתונים) ולהמיר אותם לאותות מקביליים רבים יותר שה-LCD דורש.

נגד משתנה (בורג הכיוון הכחול): משמש לכוון עדין של הניגודיות (Contrast) של הטקסט על המסך, כדי להבטיח קריאות אופטימלית.

פיני חיבור I2C (4 פינים בצד ימין): אלו הפינים שאליהם מחברים את המודול למיקרו-בקר: GND: הארקה.

VCC: מתח הזנה (לרוב V5).

SDA (Serial Data Line): קו הנתונים של פרוטוקול I2C.

SCL (Serial Clock Line): קו השעון של פרוטוקול I2C.

מתג/נגד עבור תאורה אחורית LED (בצד שמאל): רכיב קטן שמאפשר שליטה על תאורת ה-LED האחורית של ה-LCD.

איך הוא עובד?

השילוב של שני הרכיבים מאפשר תפעול פשוט ויעיל:

חיבור פיזי: מסך ה-LCD מחובר באמצעות 16 פיני למודול מתאם ה-I2C. לאחר מכן, מודול ה-I2C מתחבר למיקרו-בקר (לדוגמה, ארדואינו) באמצעות 4 חוטים בלבד: שניים למתח (VCC ו-GND) ושניים לתקשורת I2C (SDA ו-SCL).

תקשורת מהמיקרו-בקר למתאם I2C: המיקרו-בקר שולח פקודות ונתונים למודול מתאם ה-I2C באופן טורי דרך קווי ה-SDA וה-SCL (פרוטוקול I2C).

המרה על ידי מתאם I2C: שבב ה-PCF8574 במודול ה-I2C מקבל את הנתונים הטוריים, וממיר אותם במהירות לאותות מקביליים מתאימים (בפורמט שה-LCD מבין).

הצגת הטקסט על ה-LCD: בקר ה-LCD הפנימי שבמסך מקבל את האותות המקביליים ממתאם ה-I2C, ומפעיל את הקריסטלים הנוזליים ואת התאורה האחורית כדי להציג את התווים והטקסט הרצויים על המסך.

## תיאור החלופה הנבחרת

הפח יכול ליישם מרחק אולטראסוני לזיהוי אדם ממרחק מטר, בזיהוי מרחק נמוך, מנוע הסרוו יופעל לפתיחת מכסה הפח במקביל חיישן המים עובד ומחכה לזהות לחות בתחתית הפח-ובמקרה כזה, מנוע הסרוו שמחובר אליו מטלית קטנה יופעל עד לייבוש תחתית הפח, ובנוסף חיישן מרחק אולטראסוני תחת מכסה הפח ימדוד את המרחק לזיהוי תכולת הפח, ובמקרה שהפח מלא ידפיס הודעה במסך ה-LCD באחוזי מילוי.

במקרה של זריקת עצמים וחפצים לפח, המצלמה תזהה לפני פתיחת המכסה אם העצם לזריקה הוא בקבוק, אם כן הפח לא יפתח ותישלח הודעה למסך ה-LCD.

## מטרות ויעדים

- חיישן מרחק עובד.
- חיישן מנוע עובד.
- חיישן מים עובד.



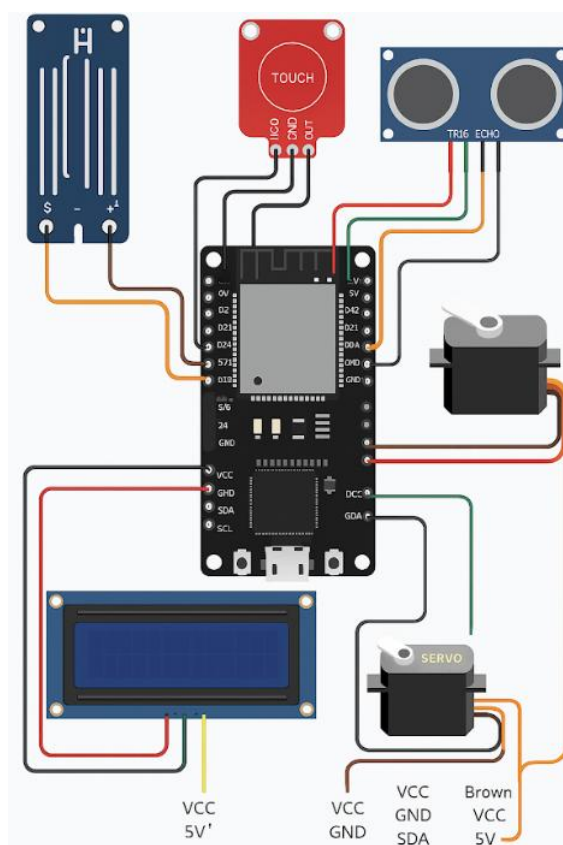
- חיישן מגע עובד.
- זיהוי בקבוקים על ידי המצלמה.
- הפעלת נתונים בצורה מסונכרנת וטובה.
- שימוש נכון בתצוגת המסך.
- תצוגת הפרויקט אסתטית וללא חוטים גלויים.

## אתגרים

- לחבר ולהפעיל כל חיישן.
- חיבור המצלמה וזיהוי בקבוקים על ידי מודל מאומן חיצונית לארדואינו.
- הפעלת הנתונים בצורה מקבילית.
- ליצור את החומרה בצורה יפה לעין.
- למקם כל חיישן במקום הנכון כדי שיפתח מכסה הפח בצורה נכונה, ויפעל ייבוש אוטומטי.

## תיעוד

### תרשים חיבורים



בתרשים צריכים להיות גם חיישן מים ומגע, פשוט לא היה לי איור שלהם...ולכן הם ככה.

## חיבורים ואופן הפעולה

חיישן מגע	בקר
VCC	3V
GND	GND
OUT	D13-D26

חיישן מים	בקר
S	34D
-	GND
+	3V3

חיישן אולטראסוני	בקר
VCC	V5
TRIG	22D
ECHO	21D
GND	GND

מסך LCD	בקר
VCC	V5
GND	GND
SDA	GPIO21
SCL	GPIO22

מנוע סררו	צבע חוט	מספר חוט
GND	חום	1
VCC	אדום	2
DATA	כתום	3

## תיעוד קטעי קוד

קוד לסגירת ופתיחת המכסה בהתאם למרחק המזוהה.

```
#include <ESP32Servo.h> // ספרייה תואמת ל-ESP32

#define SERVO_PIN 32
#define TRIG_PIN 5
#define ECHO_PIN 18

Servo myServo;

void setup() {
  Serial.begin(115200);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  myServo.attach(SERVO_PIN);
  myServo.write(0);
}

float readDistanceCM() {
  // שליחת פולס קצר ל-TRIG
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  // מדידת הזמן של חזרת האות
  long duration = pulseIn(ECHO_PIN, HIGH, 30000); // 30 מ"ס
  if (duration == 0) return -1; // לא נקלט אות

  float distance = duration * 0.034 / 2;
  return distance;
}

void loop() {
  float distance = readDistanceCM();
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance > 0 && distance < 10) {
    Serial.println("Distance < 10cm: Moving servo 1/4 turn forward and returning.");
    myServo.write(90); // סיבוב
    delay(2000); // השהייה להיות פתוח
    myServo.write(0); // סגירה
    delay(2000); // דיליי לסיום
  } else {
    Serial.println("Distance >= 10cm or no echo: Servo remains off.");
    myServo.write(0);
    delay(2000); // דיליי קטן לפני שזה מתחיל לבדוק מחדש
  }
}
```

קוד להפעלת מנוע הסרוו של מטלית הייבוש בהתאם לזיהוי רמת המים.

```
#include <ESP32Servo.h>
// פין של מנוע הסרוו
#define WATER_PIN 34

Servo myServo;

void setup() {
  Serial.begin(115200);
  myServo.attach(SERVO_PIN);
  myServo.write(90); // לעצור

  pinMode(WATER_PIN, INPUT); // קלט מחיישן מים
}

void loop() {
  int sensorValue = analogRead(WATER_PIN);

  Serial.print("Water sensor value: ");
  Serial.println(sensorValue);

  if (sensorValue > 1000) { // סף זיהוי רטיבות
    Serial.println("Detected water! Spinning servo...");
    myServo.write(0); // סיובב רגיל סיבוב בכיוון אחד
  } else {
    Serial.println("Dry. Stopping servo.");
    myServo.write(90);
  }

  delay(500);
}
```

קוד להפעלת המצלמה:

```

define CAMERA_MODEL_AI_THINKER // Has PSRAM#
define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM#
define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM#
** Espressif Internal Boards ** //
define CAMERA_MODEL_ESP32_CAM_BOARD//
define CAMERA_MODEL_ESP32S2_CAM_BOARD//
define CAMERA_MODEL_ESP32S3_CAM_LCD//
define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM#
define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM#
#include "camera_pins.h#

===== //
Enter your WiFi credentials //
===== //
;const char *ssid = "Kita-2
;const char *password = "Xnhbrfxfho

;()void startCameraServer
;void setupLedFlash(int pin)

} ()void setup
;Serial.begin(115200)
;Serial.setDebugOutput(true)
;()Serial.println

```

```

;camera_config_t config
;config.ledc_channel = LEDC_CHANNEL_0
;config.ledc_timer = LEDC_TIMER_0
;config.pin_d0 = Y2_GPIO_NUM
;config.pin_d1 = Y3_GPIO_NUM
;config.pin_d2 = Y4_GPIO_NUM
;config.pin_d3 = Y5_GPIO_NUM
;config.pin_d4 = Y6_GPIO_NUM
;config.pin_d5 = Y7_GPIO_NUM
;config.pin_d6 = Y8_GPIO_NUM
;config.pin_d7 = Y9_GPIO_NUM
;config.pin_xclk = XCLK_GPIO_NUM
;config.pin_pclk = PCLK_GPIO_NUM
;config.pin_vsync = VSYNC_GPIO_NUM
;config.pin_href = HREF_GPIO_NUM
;config.pin_sccb_sda = SIOD_GPIO_NUM
;config.pin_sccb_scl = SIOC_GPIO_NUM
;config.pin_pwdn = PWDN_GPIO_NUM
;config.pin_reset = RESET_GPIO_NUM
;config.xclk_freq_hz = 20000000
;config.frame_size = FRAMESIZE_UXGA
config.pixel_format = PIXFORMAT_JPEG; // for streaming
config.pixel_format = PIXFORMAT_RGB565; // for face detection/recognition//
;config.grab_mode = CAMERA_GRAB_WHEN_EMPTY
;config.fb_location = CAMERA_FB_IN_PSRAM
;config.jpeg_quality = 12
;config.fb_count = 1
    
```

```

if PSRAM IC present, init with UXGA resolution and higher JPEG quality //
for larger pre-allocated frame buffer //
} if (config.pixel_format == PIXFORMAT_JPEG)
{ if (psramFound())
;config.jpeg_quality = 10
;config.fb_count = 2
;config.grab_mode = CAMERA_GRAB_LATEST
} else {
limit the frame size when PSRAM is not available //
;config.frame_size = FRAMESIZE_SVGA
;config.fb_location = CAMERA_FB_IN_DRAM
}
} else {
Best option for face detection/recognition //
;config.frame_size = FRAMESIZE_240X240
if CONFIG_IDF_TARGET_ESP32S3#
;config.fb_count = 2
endif#
{

if defined(CAMERA_MODEL_ESP_EYE)#
;pinMode(13, INPUT_PULLUP)
;pinMode(14, INPUT_PULLUP)
endif#
    
```

```

camera init //
;esp_err_t err = esp_camera_init(&config)
} if (err != ESP_OK)
;Serial.printf("Camera init failed with error 0x%x", err)
;return
{

;()sensor_t *s = esp_camera_sensor_get
initial sensors are flipped vertically and colors are a bit saturated //
} if (s->id.PID == OV3660_PID)
s->set_vflip(s, 1); // flip it back
s->set_brightness(s, 1); // up the brightness just a bit
s->set_saturation(s, -2); // lower the saturation
{
drop down frame size for higher initial frame rate //
} if (config.pixel_format == PIXFORMAT_JPEG)
;s->set_framesize(s, FRAMESIZE_QVGA)
{

if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)#
;s->set_vflip(s, 1)
;s->set_hmirror(s, 1)
endif#

if defined(CAMERA_MODEL_ESP32S3_EYE)#
;s->set_vflip(s, 1)
endif#

```

```

Setup LED FLash if LED pin is defined in camera_pins.h //
if defined(LED_GPIO_NUM)#
;setupLedFlash(LED_GPIO_NUM)
endif#

;WiFi.begin(ssid, password)
;WiFi.setSleep(false)

;Serial.print("WiFi connecting")
} while (WiFi.status() != WL_CONNECTED)
;delay(500)
;("." )Serial.print
{
;("")Serial.println
;Serial.println("WiFi connected")

;()startCameraServer

;Serial.print("Camera Ready! Use 'http://")
;Serial.print(WiFi.localIP())
;Serial.println("' to connect")
{

} ()void loop
Do nothing. Everything is done in another task by the web server //
;delay(10000)
{

```



קוד להדפסת תכולת הפח על המסך:

```
#include <LiquidCrystal_I2C.h>

const int lcdColumns = 16;
const int lcdRows    = 2;
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
const int trigPin = 9;
const int echoPin = 10;

// גובה הפח
const float BIN_HEIGHT_CM = 20.0;

void setup() {
  lcd.init();
  lcd.backlight();
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  lcd.setCursor(0, 0);
  lcd.print("מדידת פח זבל");
  delay(1500);
  lcd.clear();
}

float readDistanceCM() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);

  // חישוב בסנטימטר
  float distanceCm = duration * 0.0343 / 2.0;
  return distanceCm;
}
```

```
void loop() {
    float distance = readDistanceCM();

    if (distance < 0) distance = 0;
    if (distance > BIN_HEIGHT_CM) distance = BIN_HEIGHT_CM;

    float percentFull = (BIN_HEIGHT_CM - distance) * 10.0 / 2.0;
    if (percentFull < 0) percentFull = 0;
    if (percentFull > 100) percentFull = 100;

    // הצגה על המסך
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("תכולה: ");
    lcd.print((int)percentFull);
    lcd.print("%");

    if (percentFull > 85) {
        lcd.setCursor(0, 1);
        lcd.print("!נא לפנות את הפח");
    }

    delay(1000);
}
```

במצב זה הקודים עדיין לא גמורים, אני צריכה לעשות אותם כתהליכונים, ולשלוח כל תמונה למודל שיחזיר האם הפריט לזריקה, הוא בקבוק או לא.

## חיבורים

השתמשתי בחיבור Wi-Fi כדי לאפשר למצלמה לשלוח תמונות של הפריטים המושלכים לפח. תמונות אלו נשלחות לשרת חיצוני (או לשירות ענן), שם מותקן ומורץ מודל למידת מכונה. מודל זה מנתח את התמונה שהתקבלה ומזהה האם הפריט הוא בקבוק או לא.

## תיאור הארכיטקטורה

קובץ BIN.ino שאחראי על הפעלת הפח, קליטת הנתונים בצורה מקבילית בתהליכונים וקליטה מהמצלמה. וכן תיקייה ובה המודל- שרת שבודק אם התמונה היא בקבוק.

## תיאור פרוטוקולי תקשורת

### פרוטוקול WIFI

Wi-Fi (Wireless Fidelity) - תקשורת אלחוטית

תיאור: Wi-Fi היא טכנולוגיה לתקשורת אלחוטית המבוססת על תקני IEEE 802.11. היא מאפשרת למכשירים להתחבר לרשתות מקומיות (LAN) ללא צורך בכבלים פיזיים. שבב ה-ESP32, המשמש כבקר הראשי במערכת זו, ידוע ביכולות ה-Wi-Fi המובנות שלו.

מאפיינים עיקריים:

תקנים: פועל על פי מגוון תקנים (כמו 802.11 b, g, n) המגדירים את מהירות התקשורת, טווח ואבטחה.

קישוריות לרשת: מאפשר למכשירים לתקשר עם נתב (ראוטר) אלחוטי, ודרכו לגשת למכשירים אחרים באותה רשת מקומית.

פרוטוקולי רשת: מעל שכבת ה-Wi-Fi, ניתן להשתמש בפרוטוקולי רשת כגון TCP/IP ו-HTTP להעברת נתונים.

טווח: הטווח משתנה בהתאם לתנאי הסביבה, עוצמת המשדר, והתקן בשימוש, אך בדרך כלל נע בין עשרות למאות מטרים בשטח פתוח.

אבטחה: תומך בפרוטוקולי אבטחה שונים (כמו WPA2, WPA3) להצפנת התקשורת והגנה מפני גישה לא מורשית.

### פרוטוקול SPI - תקשורת טורית סינכרונית

ממשק היקפי טורי, הוא ממשק תקשורת המשמש לשליחת נתונים במהירות גבוהה בין מספר התקנים. התקנים אלה מאורגנים בתצורת מאסטר-סלייב, שבה למאסטר יש שליטה, והסלייבים מקבלים הוראות מהמאסטר.

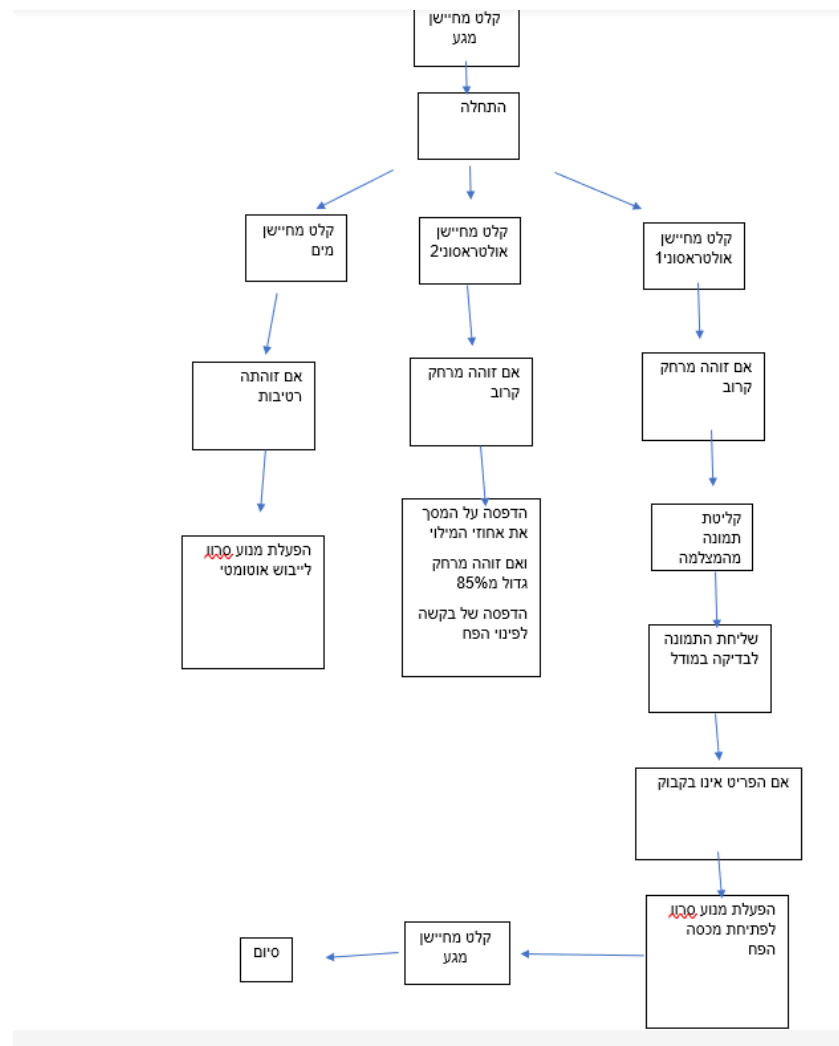
היישום הנפוץ ביותר של SPI הוא בתצורה שבה התקן בודד הוא המאסטר, ושאר ההתקנים הם סלייבים. SPI הוא פרוטוקול תקשורת סינכרוני, המיועד להעברת נתונים בקצבים גבוהים

ולקבלה מיידית. הוא מתאים לתקשורת ברמת הלוח למרחקים קצרים. הוא משתמש בארבעה חוטים (\*\*SS, MISO, MOSI, SCK\*\*) להחלפת נתונים בין ההתקנים.

ממשק התקשורת \*\*SPI\*\* הוא יתרון כאשר יש צורך לתקשר בין מספר התקנים. הוא מציע קצב העברת נתונים גבוה יותר מרוב סוגי ממשקי התקשורת האחרים, ומאפשר שליחה וקבלה של נתונים בזמן. עם זאת, \*\*SPI\*\* דורש יותר חוטים או קווי אות מאשר סוגים אחרים של תקשורת. יתרה מכך, אין פרוטוקול הודעות סטנדרטי לתקשורת באמצעות \*\*SPI\*\*, כלומר לכל התקן יכולה להיות סכמה משלו לעיצוב נתונים והודעות. **פרוטוקול PWM (Pulse Width Modulation)**

PWM היא שיטה לשליטה בעוצמת האספקה של רכיבים אלקטרוניים על ידי שינוי יחס הזמן שאות דיגיטלי מופעל בפרק זמן מסוים לזמן ההפסקה שלו. במילים אחרות PWM, משנה את רוחב הפולס של אות דיגיטלי בתדירות קבועה, ובכך שולט במתח הממוצע המוזן לרכיב.

## תרשים זרימה



## מדריך למשתמש

על הפח לעמוד במיקום האסטרטגי בבית- שיהיה זמין לזריקת אשפה, לרב עדיף למקם במטבח.

על מנת לזרוק חפץ לפח יש להתקרב ובמקרה והעצם לזריקה אינו עונה על בקבוקי מיחזור, מכסה הפח יפתח לזריקת הזבל.

אין להסתיר את המצלמה לאורך זמן, על מנת שהפח יוכל להיפתח.

יש לוודא שהבטריות מוטענות.

## סיכום ומסקנות

הפרויקט היווה עבורי אתגר משולב יחד עם חוויה גדולה, היו זמנים שלא ממש ידעתי איך מתקדמים, אבל לאט לאט הפרויקט התקדם וממש התרגשתי לראות כל חיישן שעובד.

למדתי מאוד מהחיבור החיישנים וההיכרות איתם, ושמחתי לראות שיש פיתוחים שגם אני יכולה לפתח ולראות דברים עובדים.

## בבלוגרפיה

<https://www.arikporat.com/arduino1>

[https://kits4.me/kit1\\_projects](https://kits4.me/kit1_projects)

<https://wokwi.com>

<https://www.elec4u.co.il>

[https://www.arikporat.com/wp-content/uploads/2024/01/esp32-with-temperature-sensor-LM35.pdf?&~nfopt\(fileDistorted=43894763093847244&uploadEmbeddedImages=1\)](https://www.arikporat.com/wp-content/uploads/2024/01/esp32-with-temperature-sensor-LM35.pdf?&~nfopt(fileDistorted=43894763093847244&uploadEmbeddedImages=1))

<https://www.mrcoral.co.il>

<https://pro.easyeda.com>

<https://see-sys.co.il>

<https://internet-israel.com/%D7%9E%D7%93%D7%A8%D7%99%D7%9B%D7%99%D7%9D/%D7%9E%D7%99%D7%A7%D7%A8%D7%95-%D7%91%D7%A7%D7%A8%D7%99%D7%9D/%D7%A7%D7%9C%D7%98-%D7%90%D7%A0%D7%9C%D7%95%D7%92%D7%99-%D7%95%D7%93%D7%99%D7%92%D7%99%D7%98%D7%9C%D7%99-%D7%A2%D7%9D-esp32>

<https://randomnerdtutorials.com/getting-started-with-esp32>

## תודות

ראשית כל מודה לבורא עולם שהביאני עד הלום, והיה איתי ממש לאורך כל הדרך גם בזמנים הקשים וגם בזמנים השמחים יותר.

תודה רבה למורה אלה- רכזת המגמה על אחריות והתחשבות בלתי נגמרת, פשוט אין מילים לתודה ולהערכה שמגיע לך!

תודה גדולה למורה בתיה על לימוד מהנה במשך כל השנה, תוך כדי אחריות על רמה גבוהה.

ותודה ענקית למורה איילה עמור על עידוד, עזרה והכלה גדולה... עזרה גם במקרים שלא עבד כלום, השקעה והרצון הענק הזה שנצליח בגדול, ללא התפשרות מקצועית, והכל בכיף ובמאור פנים.