

## תוכן עניינים

2.....	1. <b>הצהרת הסטודנטית</b>
3.....	2. <b>מבוא</b>
4.....	3. <b>תקציר הפרויקט</b>
5.....	4. <b>תרשים מלכינים</b>
6.....	5. <b>הסבר פועלות המערכת</b>
7.....	6. <b>שרוטוט צפוני</b>
8.....	7. <b>הברker UNO ARDUINO</b>
10.....	8. <b>מסך TFT – מסך LCD צבעוני גרפי</b>
13.....	8.1 <b>8.1 מסך LCD צבעוני גרפי</b>
17.....	9. <b>אסטרטגיה לבחירת הרכיבים וনימוקים</b>
19.....	10. <b>חישון לחות אדמה</b>
22.....	11. <b>חישון אור</b>
26.....	12. <b>משאבת מים</b>
30.....	13. <b>תרשים זרימה</b>
31.....	14. <b>מסכים</b>
33.....	15. <b>תוכנית</b>
33.....	15.1 <b>קוד</b>
40.....	15.2 <b>הסבר פונקציות</b>
41.....	16. <b>תקשורות</b>
43.....	17. <b>תקלות</b>
44.....	18. <b>תיעוד הפרויקט</b>
47.....	19. <b>סיכום</b>
48.....	20. <b>מקורות</b>
49.....	21. <b>נספחים</b>

## 1. הצהרת הסטודנטית

סטודנט: תרצה כהן      ת.ז: 326132370

אני הח"מ, מצהירה בזאת כי פרויקט הגמר וספר הפרויקט המצח"ב נעשו על ידי בלבד.  
פרויקט הגמר נעשה על סמך הנושאים שלמדתי בבית הספר ובאופן עצמאי.  
פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתה של המנהה האישית.  
מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצויים בראשימת המקורות המצויים  
בספר הפרויקט.  
אני מודעת לאחריות שהנני מקבלת על עצמי על ידי חתימתה על הצהרה זו שכל הנאמר בה  
אמת ורק אמת.

### אישור המנהה האישית

הריני מאשרת שהפרויקט בוצע בהנחייתך, קראתني את ספר הפרויקט ומצאת כי הוא מוכן  
לצורך הגשת הסטודנט להגנה על פרויקט גמר.

שם המנהה: בת ציון שלביץ      חתימה: \_\_\_\_\_ תאריך: \_\_\_\_\_

## 2. מבוא

במסגרת לימודי בмагמת הנדסת תוכנה בקורס מערכות משובצות מחשב נדרש לבנות פרויקט גמר.

מהות הפרויקט היא לשלב את החומר הנלמד יחד עם נושאים ותובנות חדשים שיילמדן במהלך ביצוע הפרויקט באופן עצמאי.

אני ביצעתו פרויקט לפיתוח מערכת גינה חכמה שמטרתה לחולל מהפכה בפרקטיות הגינון המסורתיות על ידי שילוב טכנולוגיות מתקדמות.

מערכת זו תשמש בחיישנים ואוטומציה כדי לנטר ולשמור על רמות לחות אופטימליות בקרקע, תוך התחשבות בזמן ההשקייה (יום/לילה) ולהבטיח גינות ירוקות בריאות ושותפות.

המטרה העיקרית של פרויקט זה היא לספק פתרון יעיל ובר קיימת עבור חובבי גינון, המאפשר להם לשמור ללא מאמץ על מראה יפה, תוך חיסכון במים ושמירה על בריאות הצמחים.

### 3. תקציר הפרויקט

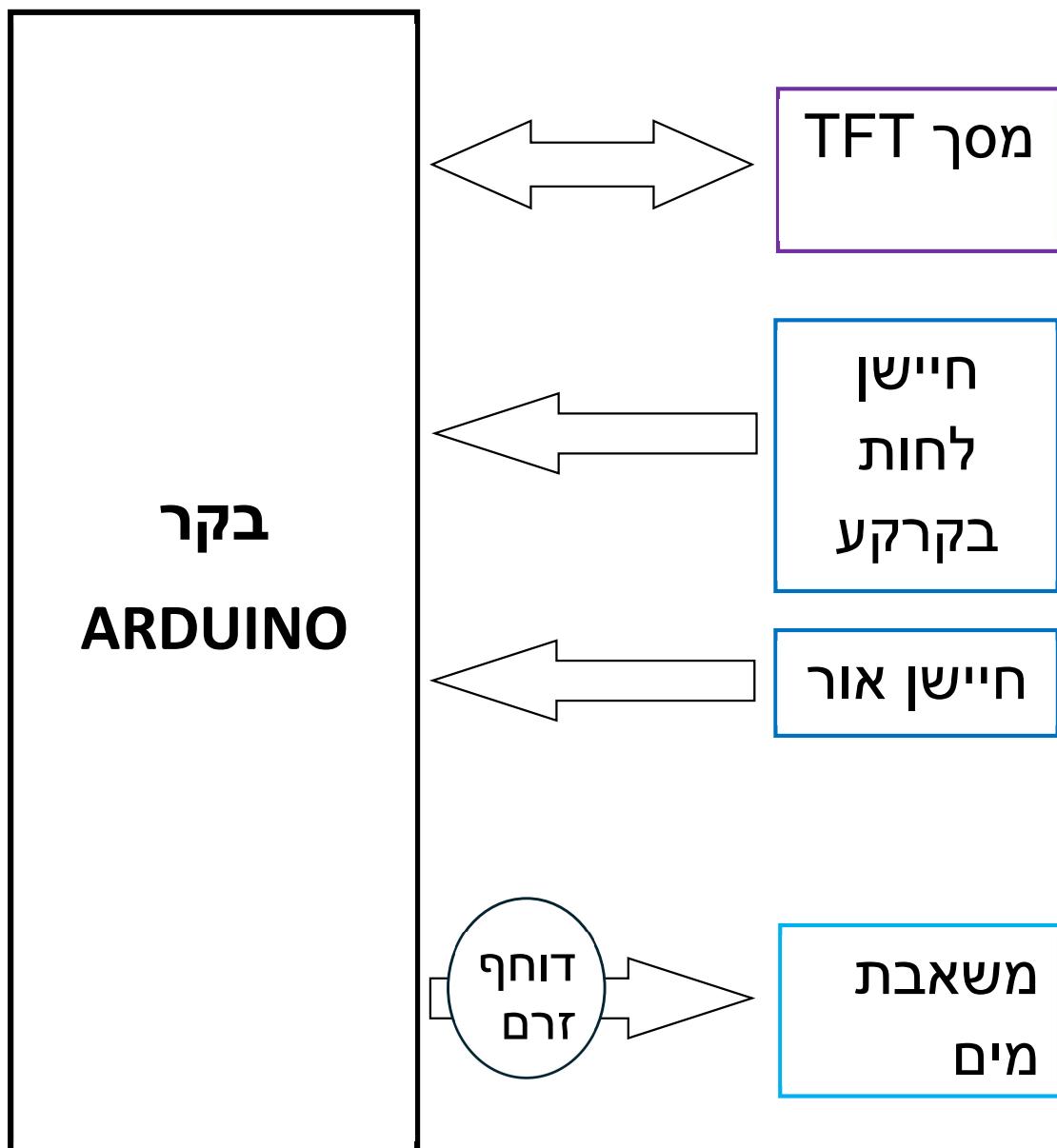
המשתמש מכניס למערכת את מידת הלחות הרצiosa אודוט הצמח שעליו מותקנת המערכת וכן את הזמן המועדף להשקיה – יום/לילה, מעתה ואילך המערכת תנתר את מידת הלחות של הצמח ותשקה במידת הצורך בזמן שנבחר.

השיקול שהיה לי בבחירה הנושא, לטפח גינה ביתית זה מושאלת לב של רבים רק שהוא דורש הרבה שימוש לב, טיפול ותחזוקה הפרויקט עונה על הצריכים האלה ומלבד ההתקנה אין למשתמש דאגות על התחזוקה של הצמח ונשאר לו רק להנוט מגינה יפה, פרחוניות ושובעת.

אפשרויות לפיתוח עתידי:

- התחשבות בכמות מים רצiosa,
- שליטה מרוחק על המערכת ועוד.

4. תרשימים מלכניים

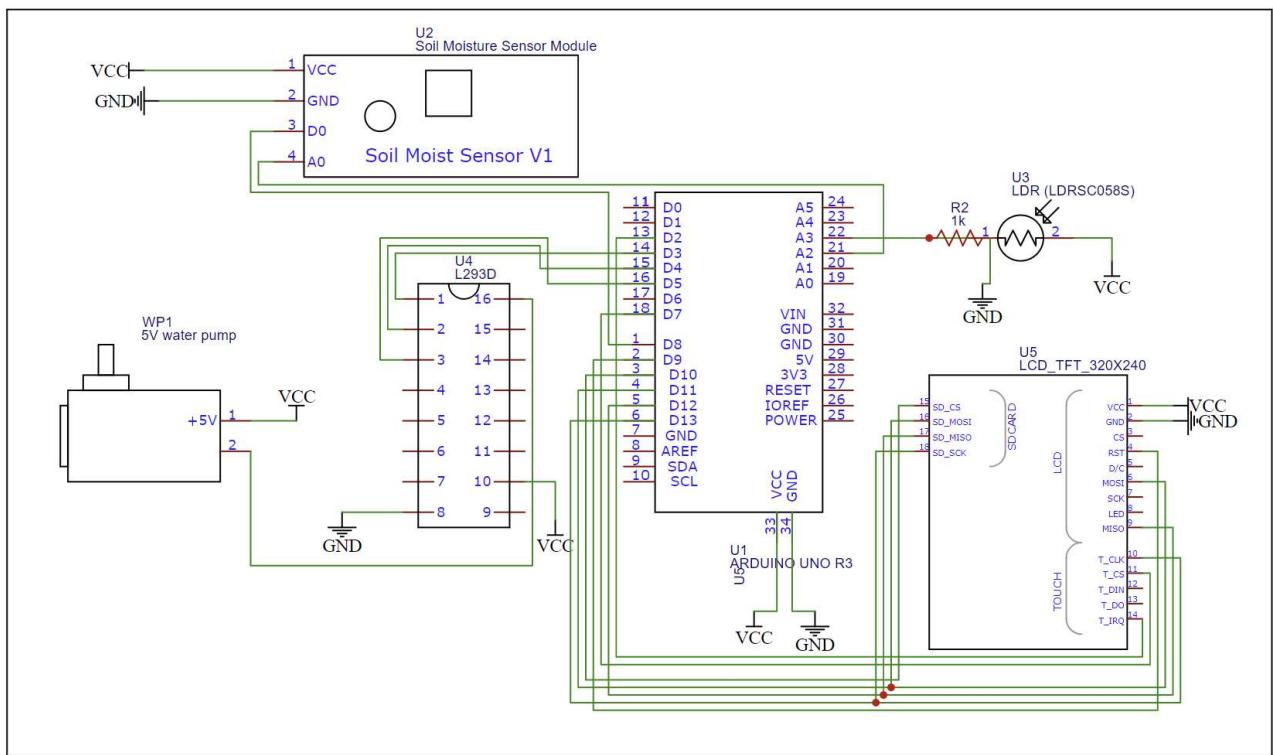


## 5. הסבר פועלות המערכת

הGINA החכמה משלבת מספר רכיבים מרכזיים המסייעים בפעולתה:

- **בקב** - אליו מחוברים כל חלקו הרכיבים והציג הגרפי. התוכנה למשה צרובה בבקב ומחברת בין כל חלקו הפרוייקט.
- **מסך TFT** - תצוגה גרפית צבעונית  $240 \times 320$  פיקסלים, מסך למשתמש דרכו יוכל להציג את המערכת ולעקב אחריה במצב של הGINA.
- **חישן לחות בקרקע** - מודד את רמת הלחות בקרקע ו מעביר את הנתונים לבקר.
- **חישן אור** - חישן המודד את עוצמת האור בסביבתו, על ידו נבצע את ההשקייה בזמן הרצוי.
- **מערכת השקיה** – כוללת משאבה שמייצרת את הזרימה ממאגר המים שמופעלים ע"י הבקר בהתאם לצורכי השקיה.

## 6. שירטוט חשמלי

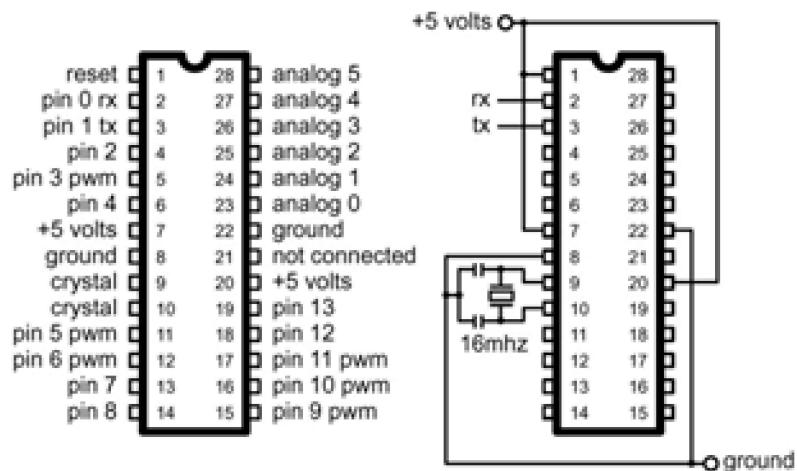
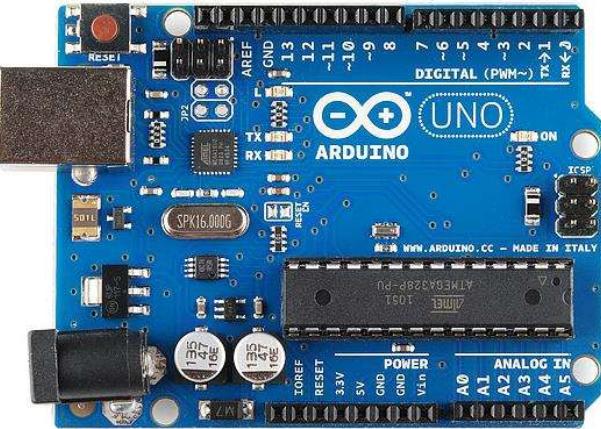


## 7. הAKER ARDUINO UNO

### ЛОЧ АРДУОИНО



בקר הארדואינו מבוסס על המיקרו-בקר Atmega328p



המאפיינים העיקריים של הרכיב Atmega328p הם:

- תדר שעון 16MHz.
- מתח עבודה 5v (אספקת מתח לכרטיס 7v-12v).
- זרם בהדק O/I עד .40mA.
- זיכרון תוכנית (flash) בגודל 32k.
- זיכרון נתונים (ram) בגודל 2k.
- 14 כניסות ויציאות דיגיטליות.
- 6 יציאות PWM.
- 6 כניסות אנלוגיות ברזולוציה של 10 סיביות.
- תקשורת טורית (spi, i2c , rs232).
- 2 פסיקות חייזנויות.

## 8. מסך TFT – מסך Touch



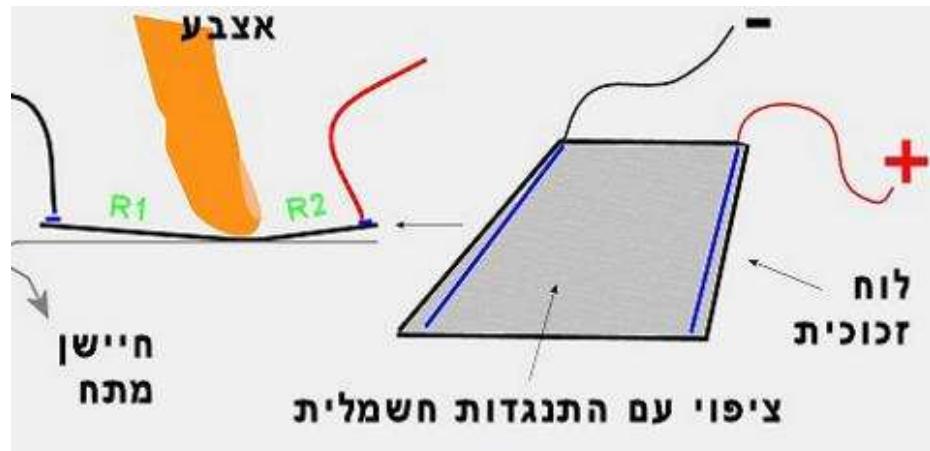
### עיקרונות הפעולה

משטח מצופה בתנגדות מכוסה בזכוכית דקה (כפי שמתואר באIOR), בזמן נגייע במסך למשה מתבצעת חלוקה של המסך לשני נגדים  $R_1$ ,  $R_2$  בציר  $Z$ . על מנת ליצור חלוקה גם בציר  $X$ , הכניסו משטח נוסף כרך שמתבצעת חלוקה לשני נגדים גם בציר זה.

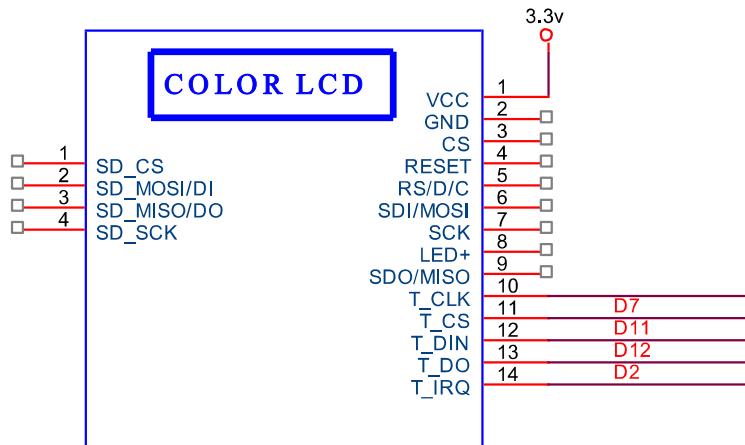
חלוקת לשני נגדים בכל נקודה מגע מאפשרת חלוקת מתח, המתח תמיד יהיה ייחודי לנגיעה.

באמצעות ממיר ADC ברזולציה של 12 סיביות דו ערכות ( $X, Y$ ) הנמצא על המסך אנו מmirים את המתח למדוד דיגיטלי הנשלח למעבד בתקשורת טורית SPI.

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



### חיבור חשמלי לבקר



הרגלים המסומנות ב- 'T' הן האחראיות לתקשורת הطارית , 4 מהן בפרוטוקול SPI . IRQ\_T יורד ל- '0' בזמן נגיעה על המסר .

למעשה באמצעות תקשורת זאת אנו קוראים את ערכי המmir לצירים Y,X מהמסר .

#### פונקציות לקרואת המיקום :

ציר X : `x = LcdTouch.xTouch;`

ציר Y : `y = LcdTouch.yTouch;`

הפונקציות מוחזירות את מיקום הלחיצה .

דוגמא לקטע תוכנית הממתין לחיצת מקש וקריאה ערכי Y,X ובדיקה האם הלחיצה בוצעה באזור מוגדר .

```

unit16_t x, y;

Void setup()
{
    LcdTouch.begin();
    LcdTouch.clearButton();
    LcdTouch.set(3780, 372, 489, 3811);
    LcdTouch.fillScreen(0x7F82BB);
}

Void loop()
{
    if (LcdTouch.touched())

```

```
{  
    LcdTouch.readTouch();  
    x = LcdTouch.xTouch;  
    y = LcdTouch.yTouch;  
    int ButtonNum = LcdTouch.ButtonTouch(x, y);  
    if (ButtonNum == 1)  
    {  
        screen1();  
    }  
    else if(ButtonNum == 2)  
    {  
        Screen2();  
    }  
}
```

## 8.1 LCD צבעוני גרפי

LCD גרפי צבעוני, ברזולוציה של 240x320 פיקסלים RGB, כל פיקסל מקבל ערך של 16bit או 18bit המרכיבים את הצבע.

שליחת המידע היא באמצעות תקשורת טורית SPI המורכב מ-4 הנקודות: MOSI, MISO, SCK, CS ( ניתן לחבר 3 מערכות : צג, מסך מגע, כרטיס זיכרון ).

ועוד שני רגלי בקרה: הדק RS לקביעת נתון או בקרה ו-RESET לאיפוס.

### תפקיד ה-Pins:

-Reset - מתחילה את מעבד התצוגה.

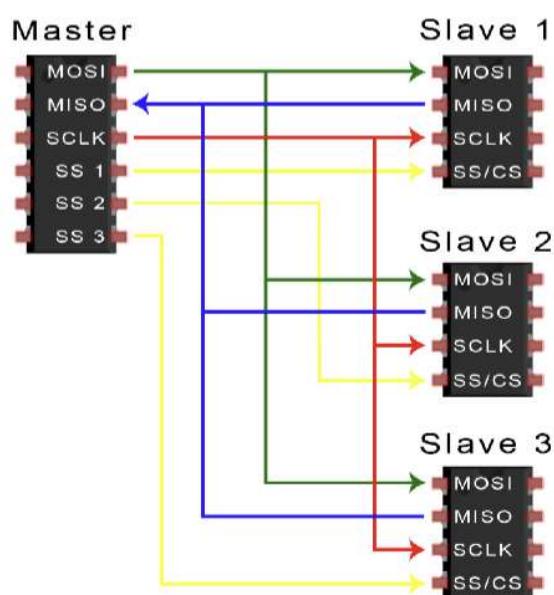
-AD/RS - הדק הקובע אם נשלח מילת בקרה (כתובת פיקסל) RS=0 או קובע אם נשלח מידע של הצבע RS=1.

-MOSI/SDI - קו כניסה נתונים טורי (מהמעבד למאסטר).

-MISO/SDO - קו יציאה נתונים טורי (מהמאסטר למעבד).

-SCK - קו שעון ששולח המעבד החיצוני (התקשורת סינכרונית).

-SS - הדק אפשר תקשורת. מודע לבחור עבור עמו תבצע התקשרות. כאשר מתחילה תקשורת, ההדק יורד ל-0.



תקשרות SPI היא סינכרונית Duplex full, יש בה מסטר אחד בלבד, אין צורך לאשר קבלת נתונים.

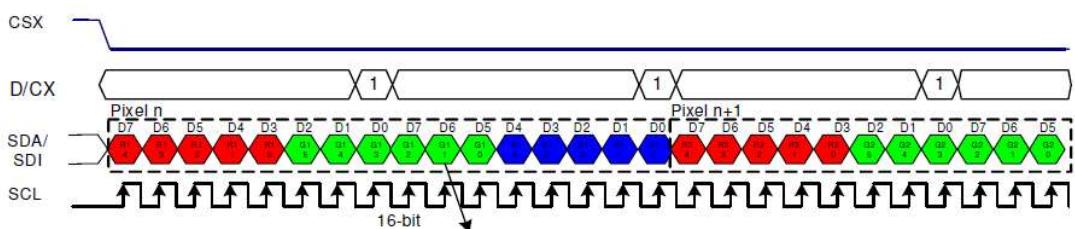
אופן ביצוע התקשרות: השעון שולח פולסים כאשר בכל פולס נתונים נשלחים מהMASTER לעבד דרך ה- MOSI ומחזרו לMASTER דרך ה- MISO. כאשר התקשרות מסתיימת, המASTER מורה לשעון שיפסיק לשולח פולסים והוא SS מתנתק מהרכיב התורן.

### **קביעת צבע לכל פיקסל**

ניתן לשולח מידע על צבע הפיקסל לפי bit 16bit ואז שולחים מידע של שני בתים או לפי bit 18bit ואז שולחים שלושה בתים וכל בית מכיל מידע על צבע יroke, אדום או כחול.

### **מוד של 16bit**

בתוך המילה המשודרת של 16Bit יש את כל הצבעים לפי התיאור הבא:



5 ביטים- מרכיב של צבע אדום.

6 ביטים- מרכיב של צבע יroke.

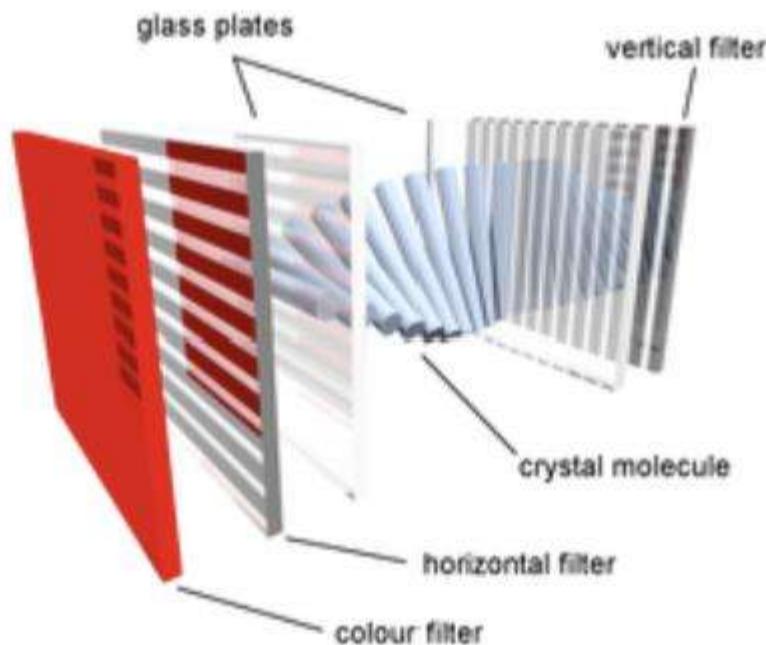
5 ביטים- מרכיב של צבע כחול.

בעזרת שילוב של כל הצבעים ניתן לקבל A56 צבעים, כדי להגיאו לכל פיקסל יש צורך כתוב קודם את הכתובת ולאחר מכן את המידע.

לפי התרשים רואים שבהתחלת התקשרות  $S=0$  ולאחר מכן משדרים 16 ביטים בצורה טוירית.

כל עליית שעון מועבר ביט אחד לפי הפרוטוקול.

כדי לקבוע אם שידור הנתונים יהיה כתובות או מידע נשנה את המצב הלוגי של RS.

תת פיקסל של צג LCD צבעוני:

להלן פונקציות המסך שהשתמשתי בהן בפרויקט:

פרמטרים שהפונקציה מקבלת או מוחזירה	תפקידה	שם הפונקציה
---	אתחול התצוגה	LcdTouch.begin()
מחזירה ערך מספרי של קואורדינטת ה- <i>x</i>	קורא את קואורדינטת ה- <i>x</i> של המגע	LcdTouch.xTouch()
מחזירה ערך מספרי של קואורדינטת ה- <i>y</i>	קורא את קואורדינטת ה- <i>y</i> של המגע	LcdTouch.yTouch()
מחזירה ערך בוליאני true/false	בודק אם מסך המגע נלחץ	LcdTouch.touched()

מחזירה ערכים מספריים קואורדינטות ה-x וה-y	קוראת את קואורדינטות המגע מסך המגע	LcdTouch.readTouch()
מקבלת color – צבע מסך לפי 16 סיביות או שם באותיות גדולות של הצבע.	צובעת את כל המסך	LcdTouch.fillScreen ()
מקבלת את מיקום החליצה y, x ומחזירה ערך מסווג של הפקד שנילחץ	בודקת את מספר הפקד שנילחץ לפי המיקום (x, y)	LcdTouch.ButtonTouch
מקבלת [פרמטרים של מיקום, גודל, צבע וטקסט הפקד	צייר פקד במסך עם טקסט	LcdTouch.drawButton()
מיקום ההודעה y , מהירות, גודל הפונט, צבע ההודעה, צבע הרקע הפונט, צבע ורקע ההודעה	כתיבת הודעה על המסך עם קביעת המיקום, גודל הפונט, צבע ורקע ההודעה	LcdTouch.print()
מקבלת color- צבע הטקסט לפי 16 סיביות או שם באותיות גדולות של הצבע	קובע את צבע הטקסט	LcdTouch.setTextColor()
מקבלת x - y- קואורדינטות המיקום של הסמן	קובע את מיקום הסמן על המסך	LcdTouch.setCursor()

## 9. אסטרטגיה לבחירת הרכיבים וニימוקים

### YL-69 Soil Moisture Detection Sensor Module - חיישן לחות אדמה

#### חלופות אפשריות:

1. Capacitive Soil Moisture Sensor: חיישן לחות קיבולי, אינו מושפע מהתחמצנות או קורוזיה, אך עלותו גבוהה יותר.
2. Tensiometer: טנסiomטר לממדית מתח הקרקע, מספק נתונים מדויקים יותר אך דרוש תחזוקה שוטפת ומורכב יותר לשימוש.
3. Gravimetric Method: שיטת מדידה גרבימטרית, מדויקת מאוד אך דרושת ציוד מיוחד ותחזוקה רבה.

#### אסטרטגיה לבחירה:

- ✓ פשוטות ופעולת קל: החישן פשוט להפעלה ומספק מדידות מדויקות של לחות הקרקע.
- ✓ עלות נמוכה: מחירו של החישן נמוך ומתאים לתקציב הפרויקט.
- ✓ קלות אינטגרציה: קל לשלב את החישן במערכת ולהחבר אותו לבקר.

### חיישן אור - 5mm LDR

#### חלופות אפשריות:

1. Photo Diode: דיודה פוטו, מספקת מדידה מדויקת יותר אך מורכבת יותר לשימוש ולהטמעה.
2. Phototransistor: טרנזיסטור פוטו, רגיש יותר לשינויים בעוצמת האור, אך דרוש מעגל חשמלי מורכב יותר.
3. Light Sensor Module (TSL2561): מודול חיישן אור דיגיטלי, מספק נתונים מדויקים על עוצמת האור אך דרוש תכונות מתקדמות יותר.

#### אסטרטגיה לבחירה:

- ✓ פשוטות ואמינות: (LDR) Light Dependent Resistor הוא חיישן פשוט ואמני למדידת עוצמת האור.
- ✓ זמינות ווניגשות: החישן זמין בקלות בשוק ומחירו נמוך.
- ✓ קלות אינטגרציה: קל לשלב את ה-LDR במערכת החשמלית והאלקטרונית של הפרויקט.

## מshaבה - DC Motor Pump Water Pump VERTICAL TYPE

### פתרונות אפשרויות:

1. Peristaltic Pump: משאבה פריסטלקטי, מתאימה למדידת כמות מדויקות של נוזלים, אך עלותה גבוהה יותר וモרכבות השימוש מעטה.
2. Submersible Pump: משאבה טבולה, מתאימה להעברת מים ממכיל מים גדול, אך פחותה מתאימה להתקנה במקומות קטנים.
3. Diaphragm Pump: משאבת דיאפרגמה, מתאימה לשימושים עם לחצים גבוהים יותר, אך מורכבת יותר לשימוש ותחזוקה.

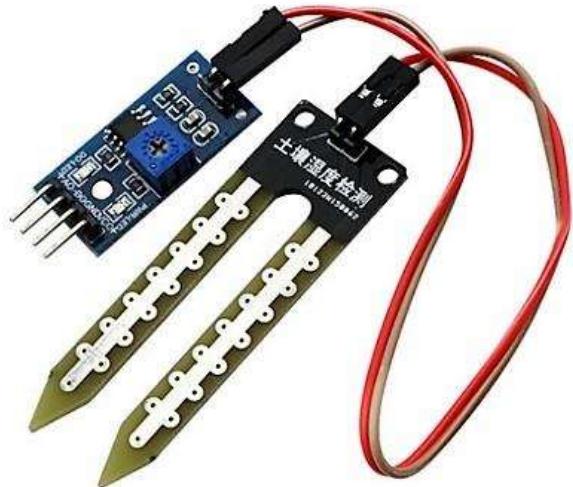
### אסטרטגייה לבחירה:

- ✓ אמינות וbijouxim: המשאבה נבחרה בזכות האמינות והיכולת שלה לספק את כמות המים הנדרשת לחץ עבודה מסוים.
- ✓ קלות אינטגרציה: המשאבה קלה לחבר ולשליטה באמצעות הבקר המרכזי.
- ✓ עלות-תועלות: המחיר של המשאבה בגין לביצועים והתמורה שהיא מספקת היה מתאים לתקציב הפרויקט.

הסיבה לבחירת הרכיבים הללו הייתה למצוא פתרונות אמינים, פשוטים לשימוש וointegracyja, עם עלות מתאימה לתקציב הפרויקט. הבחירה בחלופות נעשתה בהתחשב בצריכים הספציפיים של הפרויקט, השמירה על פשוטות וסקולול העלות-תועלות.

## 10. חיישן לחות אדמה

### YL-69 Soil Moisture Detection Sensor Module -

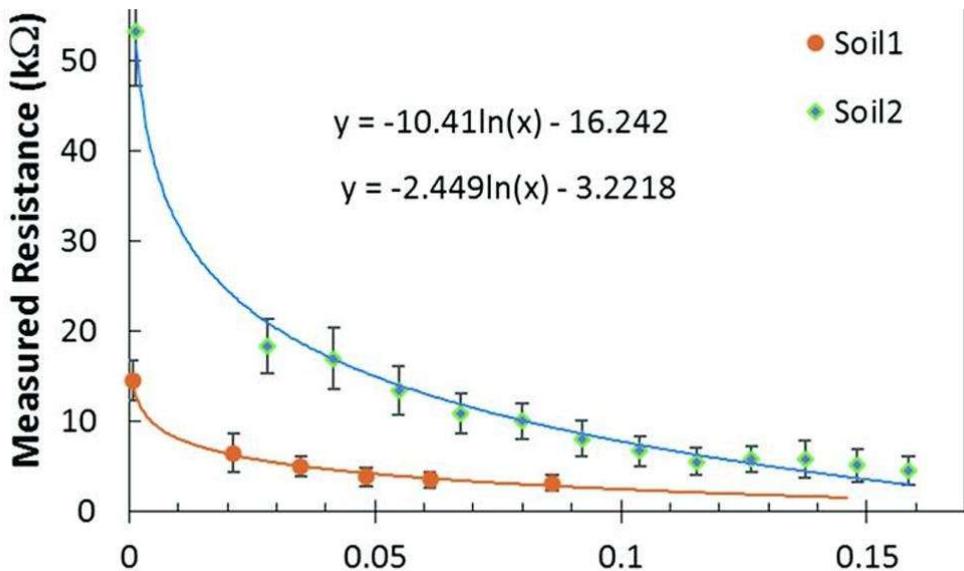


**מטרת הרכיב:** למדוד את לחות הקרקע כולם למדוד האם הקרקע יבשה, רטובה ומה שביניהם.

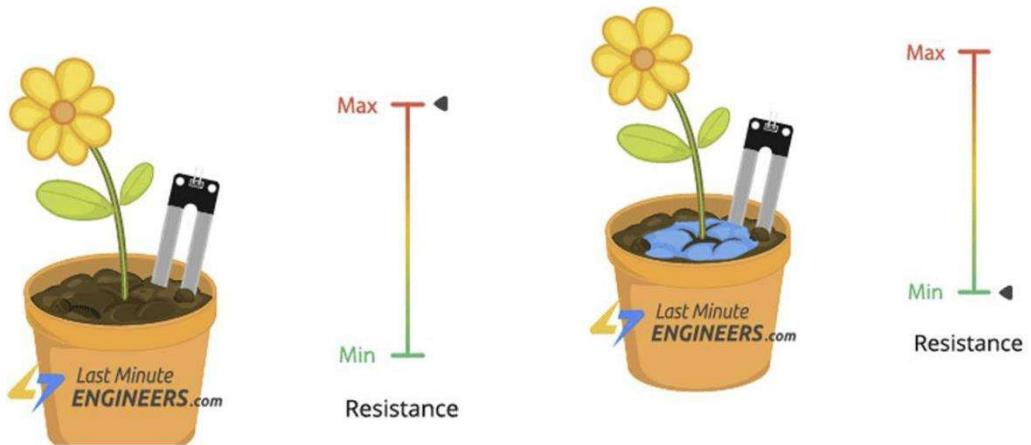
**אופן פעולה הרכיב:** הרכיב הוא נגד משתנה. נגד משתנה הוא רכיב חשמלי שתפקידו להתנגד למעבר זרם חשמלי דרכו. ערך ההתנגדות שלו נמדד באוהם כמו בנגד קבוע, אך ייחודה של הנגד המשתנה היא בכך שההתנגדותו אינה קבועה. חיישן לחות הקרקע הוא נגד משתנה שההתנגדותו משתנה בהתאם לשינוי בלחות היחסית סביבו. החישון מוחזיר ערכים אנלוגיים. חיישן לחות קרקע מורכב משני רכיבים מרכזיים הראשונים הוא חלק הבדיקה, חלק הבדיקה בניי שני מוליכים (אלקטטרודות) שנראים כמו שניים שננעוצות באדמה, האלקטרודות מSHOWOT את ההתנגדות בהתאם לחות באדמה ומעבירות את ההתנגדות לרכיב השני שהוא המודול, המודול מוציא מתח בהתאם להתנגדות ובכך הוא מאפשר לארדואינו להציג לנו נתונים שנותנים לקרוא.

אופן העבודה של חיישן לחות הקרקע די פשוטה. החלק הבודק בצורה מצלג עם שני מוליכים חסופים (מיין אלקטטרודות), החישון פועל כנגד משתנה (כמו פוטנציאומטר) שההתנגדות שלו משתנה בהתאם לכמות המים בקרקע. התנגדות החישון היא ביחס הפוך ללחות הקרקע: ככל שייותר מים בקרקע פירוש טובה יותר (כי מים הם מוליך מעולה) ויבילו להתנגדות נמוכה יותר. פחות מים בקרקע פירוש מוליכות ירודה ויבילו להתנגדות גבוהה יותר.

בגרף הבא ניתן לראות איך ההתנגדות של החישון לחות קרקע יורדת ככל שיש יותר מים בקרקע.



להלן הממחשה של פועלות החישון: כשייש מים ההתנגדות של החישון יורדת ולהיפך.



לתשומת לב, אם המתח נשאר קבוע אנחנו לא יכולים לקרוא נתונים מהארדוינו שטראה לנו ערכים בתווור 1023-0 שהמשמעות שלהם היא מתח, כאשר V5 זה 1023 ו- V0 זה 0. לשם כך יש לנו את המודול, כאשר משתמשים באરדוינו הוא מספק V5 החישון לחות קרקע מקבל את כל המתח והערך שהוא ייתן יהיה באותו ערך של 1023 (ערך אנלוגי ל- V5) גם אם התנגדות שלו תשתנה בהתאם לכמות המים בקרקע, כל המתח עדין נפל עליו כי הוא הרצק היחיד. כאשר מוסיפים למעגל את המודול מתבצעת חילוקה על הרצקה, בהתאם ליחס בין התנגדותיהם. משום שההתנגדות של החישון משתנה לפי כמות המים בקרקע היחס בין התנגדות שלו לזו של המודול שההתנגדות שלו קבועה תשתנה ויפול עליו מתח

שונה בהתאם למידת ההתנגדות של החישון (יותר ההתנגדות = יותר מתח). בעוד, הארדואין יכול למדוד את השינוי במתח על פני הרכיב ונוכל לקבל מדידה של השינוי בכמות המים בקרקע (לחחות האדמה).

כאשר האלקטרודות נמצאות באוויר נראה במסך הסריאלי 1023 וכאשר הם נמצאים במים (בלי אדמה) נראה במסך הסריאלי בין 200 ל-250. קטע תוכנית לקרואת הטמפרטורה והצגתה על הציג.

```
Void setup()
{
    Serial.begin(9600);
    pinMode(6,INPUT);
}

Void loop ()
{
    Serial.println(analogRead(A3));
    Serial.println(digitalRead(6));
    delay(1500);
}
```

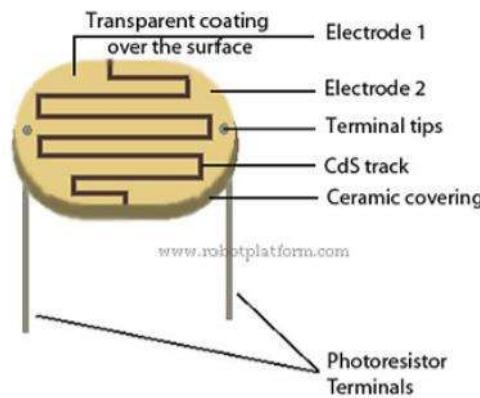
## 11. חיישן אור

**LDR 5mm -**

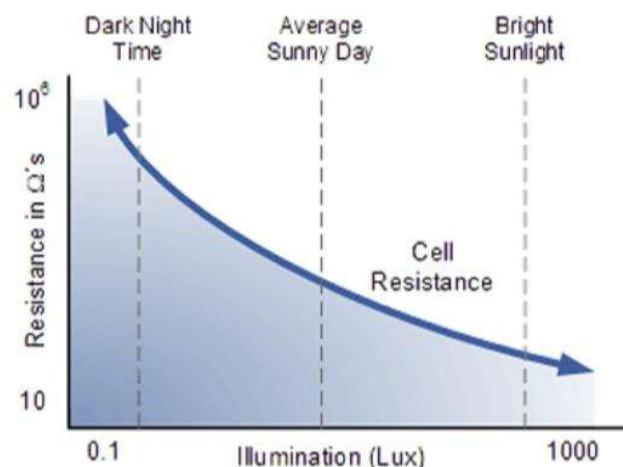
**מטרת הרכיב:** למדוד את עוצמת האור הנמצאת בסביבתו ולפי זה נחליט אם יומ/לילה. שמו באנגלית מסגיר את אופן פועלתו - נגד תלוי אור, כלומר נגד שימושה את התנגדותו בתלות באור אליו הוא נחשף.

**אופן פועלות הרכיב:** חיישן אור הוא נגד משתנה שתפקידו להתנגד למעבר זרם חשמלי דרכו. התנגדותו אינה קבועה ומשתנה בהתאם לפרמטר פיזיקלי בסביבה. הפרמטר הפיזיקלי זה עוצמת האור במקורה של החישין זהה. חיישן האור מודד את שטף האור ובכך הוא משנה את התנגדותו בהתאם לשטף האור אליו הוא חשוף.

החיישון מורכב מדסקית עגולה העשויה חומר קרמי. הדסקית מחולקת לשני אזורים המופרדים ביניהם על ידי חומר הנקרא קדמים סולפיד CdS. חומר זה אחראי להתנגדות החשמלית של החישין ותפקידו הוא בעצם לבודד, מכיוון שאם לא יהיה בידוד בין שני חלקים הדסקית לא תוכל למדודו את ההתנגדות. קדמים סולפיד CdS הוא חומר שימושים נפוצים בעיקר למטרת בידוד. משני חצאי הקרמיים יוצאים גם שתי קטבים מוליכים המחברים את המערכת.

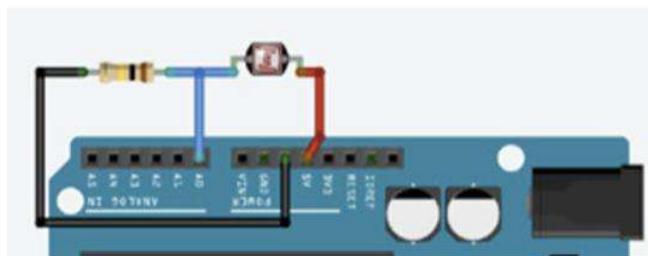
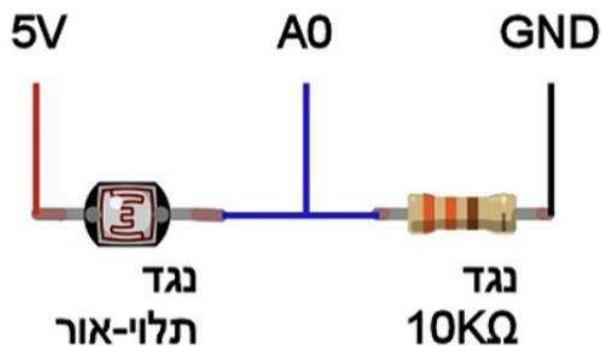
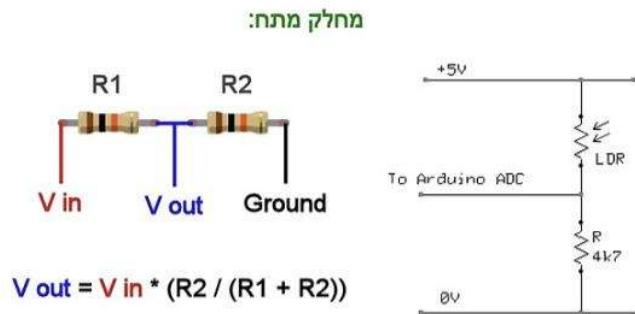


החישון משנה את התנגדותו בהתאם לעוצמת האור אליה הוא נחשף. כאשר החישון בחושך מוחלט איז התנגדותו מאד גבוהה, ואילו כאשר הוא מואר התנגדותו יורדת באופן מהיר. עוצמת התאורה נמדדת ביחידות מידת הנקראות Lux.



#### אוף המדידה של החישון:

החישון מודד תופעה פיזיקלית שמתורגמת למתח חשמלי אותו מודד הבקר בחיבור הפינים. החישון מקבל זרם קבוע מהבקר דרך חיבור ל- $-5V$ . מחברים את החישון לנגד נוסף בטור בגודל של 1000 אומם כדי ליצור חלק מתח. בגלל שינוי ההתנגדות של החישון, חלוקת המתח בין הנגד והחישון משתנה בהתאם. הבקר מודד את מפל המתח על הנגד ומירגם את ערך המתח לערך מספרי בין 0 ל-1023 כמו לכל החיבורים האנלוגיים.



לכל רכיב קולט מגדים שני משתנים. אחד לאחסון מספר הפין אליו מחברים את הנורה, ומשתנה לאחסון הערך הנקלט מהחישוב.

תוכנית דוגמא שגורמת להדלקת נורת לד כאשר חושך בחוץ:

```
#define ldrPin A1;
Int ldrValue = 0;

#define ledPin = 10;

Void setup() {
    pinMode(ldrPin, INPUT);
    pinMode(ledPin, OUTPUT);
}
```

```
Void loop() {  
    ldrValue = analogRead(ldrPin);  
    if (ldrValue < 400) {  
        digitalWrite(ledPin, HIGH);  
    }  
    else {  
        digitalWrite(ledPin, LOW);  
    }  
}
```

## 12. משאבת מים

### DC Motor Pump Water Pump VERTICAL TYPE -



**מטרת הרכיב:** ליצור את זרימת המים לצמחים.

#### מבנה המשאבה:

- מנוע DC: המנוע הוא החלק המרכזי של המשאבה. הוא פועל על מתח DC ( direct current - זרם ישיר ), בדרך כלל בטווח של 12 – 3V. המנוע הוא קטן, חסכוני באנרגיה ומספק את הכוח הדרוש לשיכוב האימפלר.
- Impeller: בתוך המשאבה ישנו אימפלר קטן המורכב מלהבים. האימפלר מחובר למנוע ומסתובב במהירות גבוהה כדי ליצור כוח מרכזיוגלי שמאפשר לשאוב את המים.
- Pump Housing: בית המשאבה הוא החלק החיצוני שמכיל את המנוע והאימפלר, הוא עשוי פלסטיק קשיח ועמיד למים ולחומרים כימיים.
- צינורות כניסה ויציאה<sup>1</sup> המשאבה מצויה בפיזות כניסה ויציאה המאפשרות חיבור לצינורות שונים ניתן לשאוב ולפלווט מים.

**אופן פעולה הרכיב:** משאבת מים DC מסוג VERTICAL TYPE פועלת באמצעות מנוע

DC (Direct Current) שמניע את המשאבה ויוצר תנוצה של מים. להלן הסבר על אופן

הפעולה של המשאבה:

1. הפעלה חשמלית: המשאבה מקבלת מתח DC ממוקור כוח (סוללה, שנאי וכו').

2. סיבוב המנווע: המנווע החשמלי מתחילה להסתובב ומוסובב את גלגל המשאבה שמחובר לציר המנווע.
3. יצירת לחץ: סיבוב גלגל המשאבה יוצר ואקום לצד הכניסה של המשאבה ולחץ מצד הייציאה.
4. שאיבת מים: הוואקום שנוצר בכניסת המים שואב מים לתוך המשאבה דרך כניסה המים.
5. הוצאה מים: המים שנכנסים למשאבה נלחצים החוצה דרך יצאת המים בזכות הלחץ שנוצר על ידי סיבוב גלגל המשאבה.
6. תנועה רציפה: כל עוד המנווע ממשיך להסתובב, המשאבה ממשיכה לשאוב ולהוציא מים בצורה רציפה.

חיבור מנווע DC לפינים של בקר ארדואינו יכול לקבל עד MH40 של זרם. זרם זה בדרך כלל מספיק לרוב רכיבי הקלה והחישונים אך אינו מספיק למრבית המנוועים שצריכים זרם גבוה יותר להתחיל לנועו וגם כדי לספק את תומנת הכוח שלהם.

בנוסף, אם נחבר צורך של זרם גבוה לבקר ארדואינו נשבר את יתר פעולות הבקרה שמתבצעות מכיוון שהצרוך ינסה למשוך את כמות הזרם שהוא צריך וזה יבוא על חשבון הרכיבים האחרים שמחוברים לארדואינו. במצב זה החישונים לא יתפקדו באמינות ויכולים לגרום לבקר ארדואינו לקבל החלטות שגויות שיגרמו נזק למערכת ולסביבה.

וכן, חיבור צורך למשוך זרם גבוה מהפינים של הבקר יכול לגרום לשיריפת רכיבי הבקר או את כלו מכיוון שהחישוט והרכיבים המורכבים על בקר ארדואינו מתוכננים לשאת בזרם נמוך.

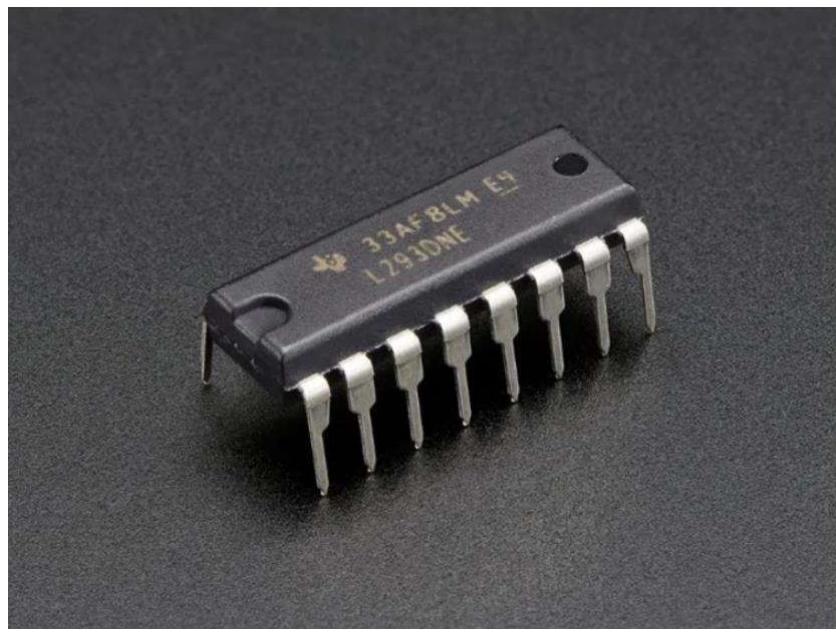
לאור הסיבות הללו יש צורך ברכיב נוסף שמסוגל להזרים דרכו זרם גבוה המגיע מקור מתח חיצוני. כמו כן הרכיב מקבל את הפיקוד מבקר ארדואינו ומאפשר שליטה בכיוון הסיבוב ובמהירות הסיבוב. רכיב זה נקרא **דוחף זרם**.

### דוחף זרם L293D:

יש כל מיני סוגי של דוחפי זרם והם בעיקר נבדלים ב: סוג המנווע אותו הם יודעים לבקר, הזרם המקורי אותו הם מסוגלים להעביר, כמות המנוועים שנייתן לחבר.

דוחף הזרם הבסיסי אתו מתחילה נקרא L293D מתוצרת חברת Texas Instruments.

לרכיב יש 16 רגליים והוא אפשר לבקר שני מנוועי DC או מנווע אחד. המתח המקורי למנווע שהוא מבקר עומד על 36 V והזרם המקורי עומד על 600mA.



#### **חיבור דוחף זרם D293D לבקר ארדואינו:**

הchipורים ממושפרים מ-1 עד 16. אם נחזיק את הרכיב מולנו כאשר המגרעת העגולה נמצאת בחלק העליון, המספור מתחילה משמאלי למגרעת לעלונה וממשיך נגד כיוון השעון. חיבורים מס' 1 ו- 9 נקראים Enable מאחר שהם מאפשרים העברת מתח/זרם להפעלת המנוע. אם נחבר אותם ל-GND המנוע יעבוד ללא תנועה. אם נחבר אותם ל-5 ווולט המנוע יסתובב במהירות מקסימלית. אם נחבר אותם לפין דיגיטלי PWM וניתן ערך בינוני המנוע יסתובב במהירות מתאימה באופן ייחודי למתח אותו נתנו.

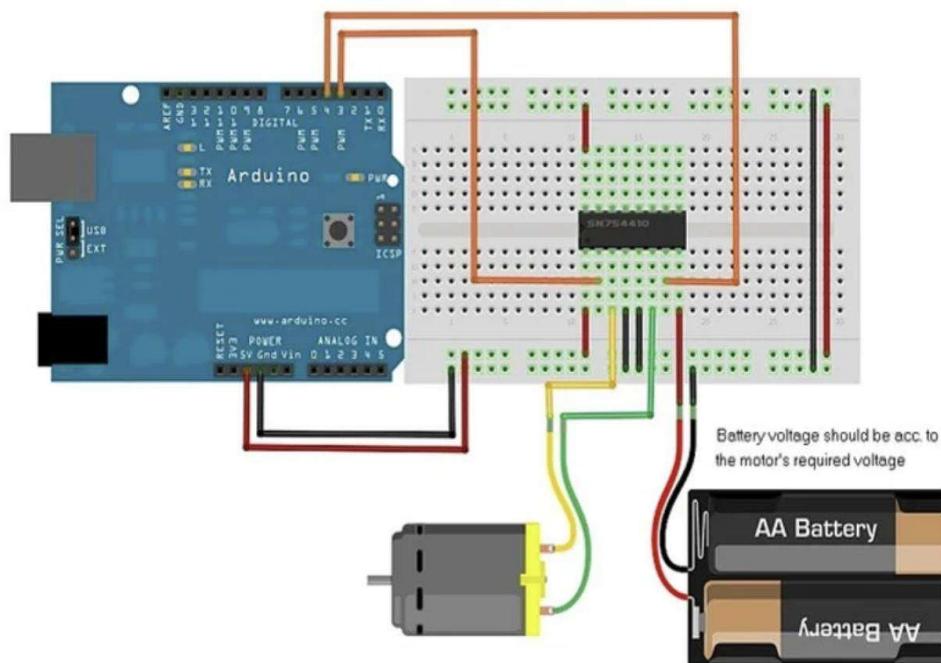
chipורים 3 ו-6 מיועדים לחבר שני החוטים המגיעים מהמנוע M1. באופן דומה chipורים 11 ו-14 מאפשרים לחבר מנוע DC שני M2.

chipורים 4,5,12,13 מתחברים ל-GND. חיברים לחבר את כולם ל-GND. בדוחפי זרם אחרים דאגו לחבר את כולם באופן מבנה.

chipורים 2 ו-7 מיועדים לקבל פיקוד מהארדואינו משני פינים דיגיטליים על הבקר לשיליטה במנוע M1. באופן דומה chipורים 10 ו-15 מקבלים פיקוד מהארדואינו לשיליטה על מנוע M2.

chipור מס' 16 מקבל 5 ווולט קבוע והוא מספק את המתח הנדרש לפעולת דוחף הזרם. chipור מס' 8 מתחבר לפולו של מקור המתח החיצוני שיפעיל את המנוע. ניתן לחבר עד 36 ווולט. המינוס של מקור המתח חיב להיות מחובר ל-GND של הארדואינו.

בתמונה שלפניכם צורת חיבור של מנוע אחד:



במקרה שלם המנוע מוגנה בתוך במכשיר.

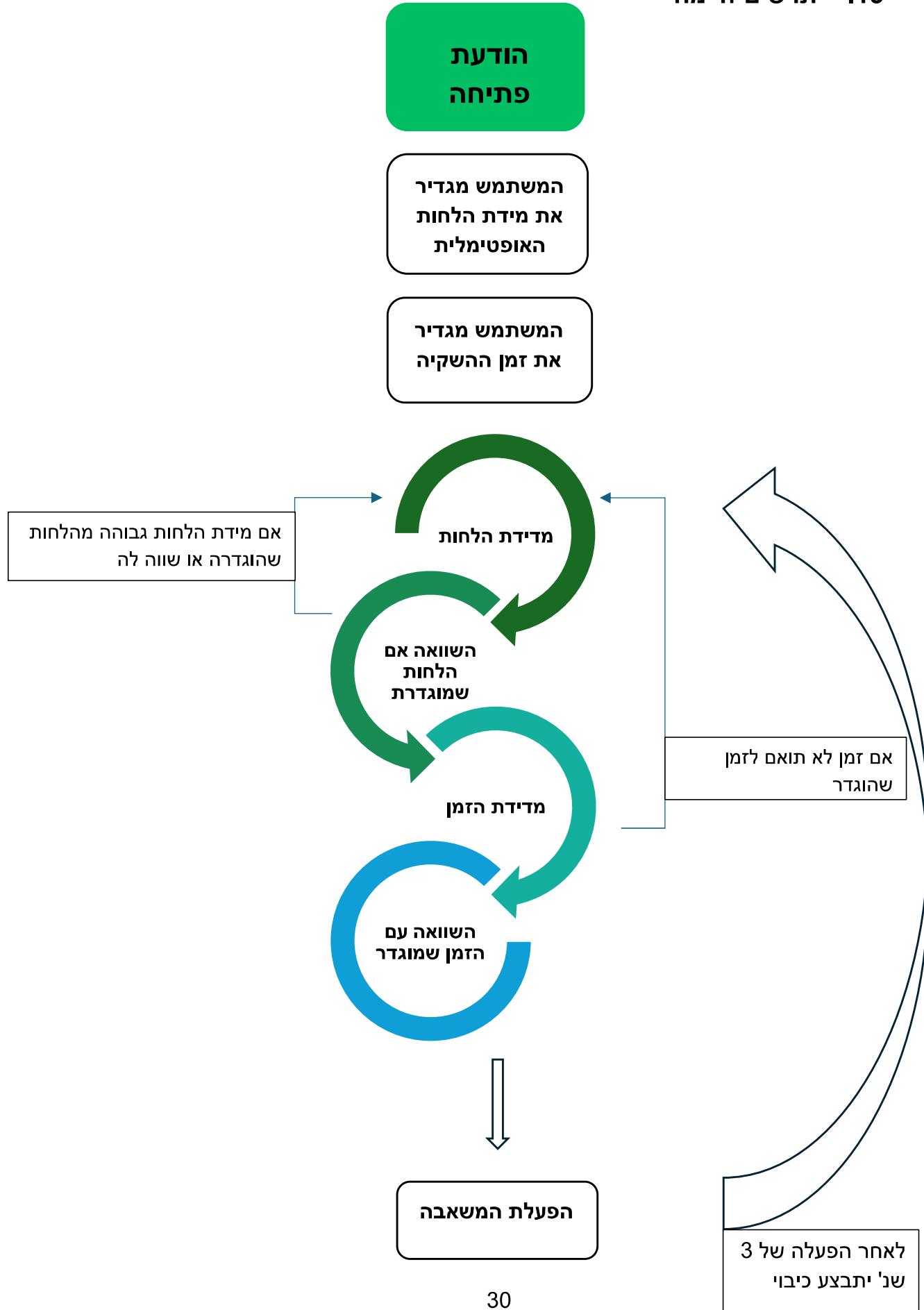
קטע תכנית להפעלת המשאבה:

```
const int pin = 9;

void setup ()
{
    pinMode (pin, OUTPUT);
}

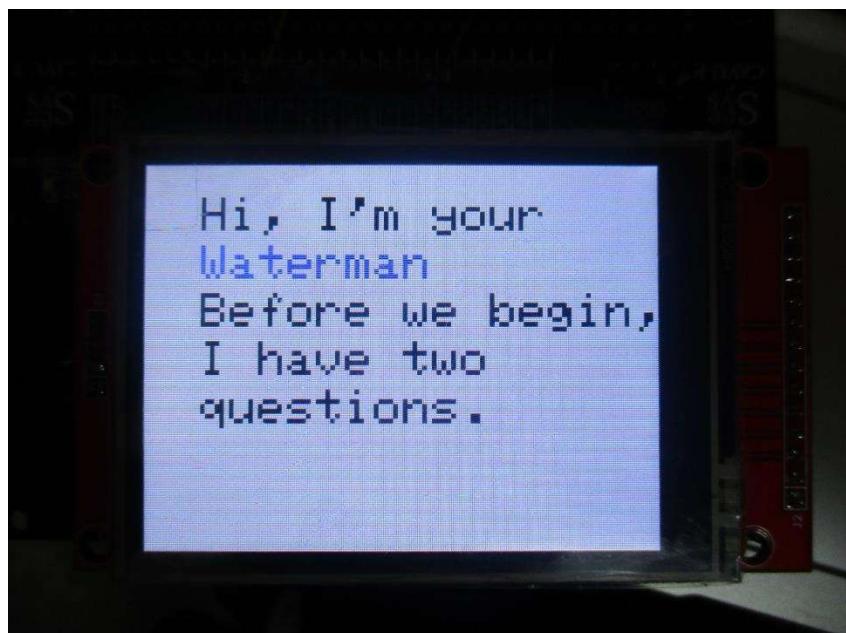
void loop()
{
    digitalWrite(pin, HIGH);
    delay(600000);
    digitalWrite(pin, LOW);
    delay(2000);
}
```

## 13. תרשימים זרימה

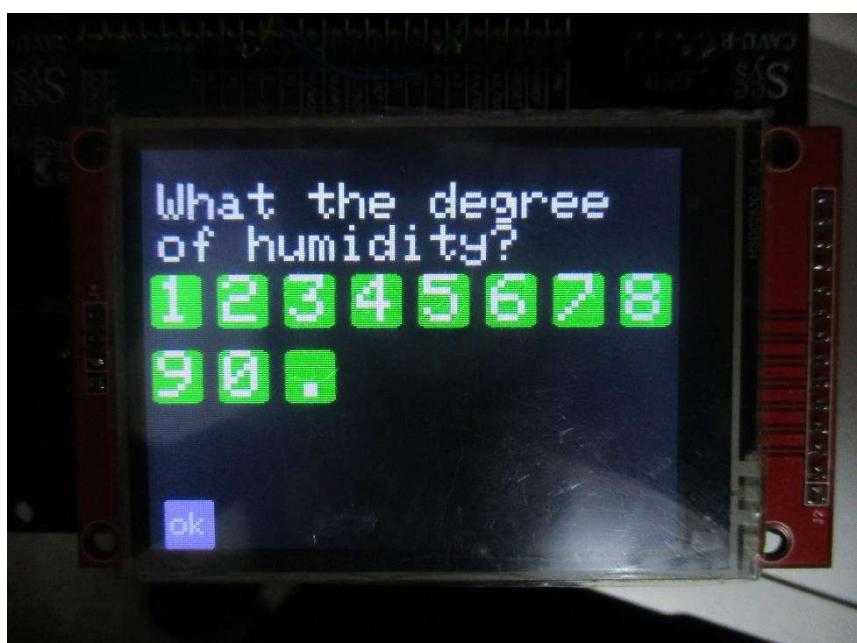


#### 14. מוכים

הודעת פתיחה למשתמש -



הגדרת מידת הלחות האופטימלית של הצמח -



הגדרת זמן השקיה -



המשך אחר המערכת, מצב: השקיה או לא, מידת הלחות של הצמח -



## 15. תוכנית

### 15.1 קוד

"בוא ספריית המסך TFT:

```
#include "TFT9341Touch.h"
```

אתחול המסר:

```
tft9341touch LcdTouch(10, 9, 7, 2);
```

הגדרת צבעים בהקסא:

```
#define BLACK 0x0000
#define WHITE 0xFFFF
#define GREEN #00BF63
#define BLUE 0x001F
#define RED 0xF800
```

הגדרת ה-חו"ק של הרכיבים:

```
#define ldrPin A3
#define hygrometerPin A2
#define pumpPin 3
```

הגדרת משתנים גלובליים שיכילו את הגדרות המשמש ש לרמת הלחות וזמן השקיה:

```
char hygrometer[20] = "";
int watering_time;
```

הגדרת משתנים גלובליים שישמשו בהמשך התוכנית:

```
uint16_t x, y;
int ButtonNum;
int cursor=0;
char string[20];
int indexString=0;
```

אתחול המרכיבת:

```
void setup() {
    Serial.begin(9600);

    LcdTouch.begin();
    LcdTouch.clearButton();
    LcdTouch.set(3780, 372, 489, 3811);
    LcdTouch.fillScreen(WHITE);

    pinMode(hygrometerPin, INPUT);
    pinMode(ldrPin, INPUT);
    pinMode(pumpPin, OUTPUT);

    screenMain();
}
```

הציג הودעת פתיחה למשתמש:

```
void screenMain()
{
    LcdTouch.print(35, 20, "Hi, I'm your", 3 , BLACK);
    LcdTouch.print(35, 50, "Waterman", 3 , BLUE);
    LcdTouch.print(35, 80, "Before we begin,", 3 , BLACK);
    LcdTouch.print(35, 110, "I have two", 3 , BLACK);
    LcdTouch.print(35, 140, "questions.", 3 , BLACK);
    while (!LcdTouch.touched());
    screenInitHygrometer();
}
```

בקשה מהמשתמש להכניס את מידת הלוחות האופטימלית לצמח שעליו מותקנת המערכת  
ע"י מקלדת ספרות, כל ספרה שעלייה יקיש השימוש תופיע במסך ע"י סמן, לסיום המשתמש  
לחץ על OK :

```

void screenInitHygrometer(){
    LcdTouch.fillScreen (BLACK);
    LcdTouch.print(10, 20, "What the degree", 3 , WHITE);
    LcdTouch.print(10, 45, "of humidity?", 3 , WHITE);
    LcdTouch.clearButton();
    LcdTouch.drawButton(1, 5, 75, 30, 30, 2, GREEN, WHITE, "1", 4);
    LcdTouch.drawButton(2, 45, 75, 30, 30, 2, GREEN, WHITE, "2", 4);
    LcdTouch.drawButton(3, 85, 75, 30, 30, 2, GREEN, WHITE, "3", 4);
    LcdTouch.drawButton(4, 125, 75, 30, 30, 2, GREEN, WHITE, "4", 4);
    LcdTouch.drawButton(5, 165, 75, 30, 30, 2, GREEN, WHITE, "5", 4);
    LcdTouch.drawButton(6, 205, 75, 30, 30, 2, GREEN, WHITE, "6", 4);
    LcdTouch.drawButton(7, 245, 75, 30, 30, 2, GREEN, WHITE, "7", 4);
    LcdTouch.drawButton(8, 285, 75, 30, 30, 2, GREEN, WHITE, "8", 4);
    LcdTouch.drawButton(9, 5, 120, 30, 30, 2, GREEN, WHITE, "9", 4);
    LcdTouch.drawButton(10, 45, 120, 30, 30, 2, GREEN, WHITE, "0", 4);
    LcdTouch.drawButton(11, 85, 120, 30, 30, 2, GREEN, WHITE, ".", 4);
    LcdTouch.drawButton(0, 10, 210, 30, 30, 2, 0x7F82BB, WHITE, "ok", 2);
    LcdTouch.setTextColor(WHITE,BLACK);

    int i=0;
    cursor=10;
    ButtonNum = 999;

    while(ButtonNum!=0){
        if (LcdTouch.touched()){
            LcdTouch.readTouch();
            x = LcdTouch.xTouch;
            y = LcdTouch.yTouch;
            ButtonNum = LcdTouch.ButtonTouch(x, y);
            keyboard(ButtonNum);
            while(string[i]!=NULL){
                LcdTouch.print(String(string[i++]));
                LcdTouch.setCursor(cursor,170);
            }
        }
    }
    LcdTouch.setTextColor(WHITE,RED);
    LcdTouch.setCursor (cursor,170);
    InitWateringTime();
}

```

פונקציה שמקבלת את מספר הceptor שנילחץ ומציגה אותו במסך ע"י סמן:

```

void keyboard(int ButtonNum)
{
    switch(ButtonNum)
    {
        case 1: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '1';
                  delay(500);
                  break;
        case 2: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '2';
                  delay(500);
                  break;
        case 3: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '3';
                  delay(500);
                  break;
        case 4: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '4';
                  delay(500);
                  break;
        case 5: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '5';
                  delay(500);
                  break;
        case 6: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '6';
                  delay(500);
                  break;

        case 7: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '7';
                  delay(500);
                  break;
        case 8: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '8';
                  delay(500);
                  break;
        case 9: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '9';
                  delay(500);
                  break;
        case 10: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '0';
                  delay(500);
                  break;
        case 11: LcdTouch.setCursor (cursor+=15,170);
                  string[indexString++]= '.';
                  delay(500);
                  break;
    }
    while(!LcdTouch.touched());
}

```

פונקציה שמצויגה בקשה מהמשתמש להגדיר את זמן השקיה המועדף :NIGHT/DAY

```
void InitWateringTime(){
    LcdTouch.fillScreen (BLACK);
    LcdTouch.print(40, 20, "What is the", 3 , WHITE);
    LcdTouch.print(40, 70, "Watering time?", 3 , WHITE);
    LcdTouch.clearButton();
    LcdTouch.drawButton(1, 45, 120, 60, 30, 2, WHITE, YELLOW, "DAY", 4);
    LcdTouch.drawButton(2, 200, 120, 80, 30, 2, WHITE, BLACK, "NIGHT", 4);

    int i=0;
    ButtonNum = 999;

    while(!LcdTouch.touched()){
        LcdTouch.readTouch();
        x = LcdTouch.xTouch;
        y = LcdTouch.yTouch;
        ButtonNum = LcdTouch.ButtonTouch(x, y);
        if(ButtonNum == 1){
            watering_time = 1;
            screenOK();
        }
        if (ButtonNum == 2){
            watering_time = 0;
            screenOK();
        }
    }
}
```

פונקציה שמצויגה מסך שמאשר את הגדרת הנתונים:

```
void screenOK(){
    LcdTouch.fillScreen (WHITE);
    LcdTouch.print(120, 70, "OK", 10 , GREEN);
    while (!LcdTouch.touched());
    InitDisplay();
}
```

אתחול התצוגה:

```
void InitDisplay(){
    LcdTouch.begin();
    LcdTouch.fillScreen(WHITE);
    loop();
}
```

הלוואה הראשית, מנטרת את החישנים, בודקת האם רמת הלחות מצריכה השקיה, במידה וכן מאשר עם הזמן וקוראת להפעלת ההשקיה:

```
void loop() {
    int humidity = analogRead(hygrometerPin);
    int ldr = analogRead(ldrPin);
    char state[9] = "";

    if(humidity<(int(hygrometer))){
        if (ldr<20 && watering_time==0 || ldr>50 && watering_time==0){
            playWatering();
            state = "Watering"
        }
    }

    LcdTouch.clearButton();

    LcdTouch.print(30, 20, "State:", 3 , BLACK);
    LcdTouch.print(state);
    LcdTouch.print(30, 50, "hygrometer: ", 3 , BLACK);
    LcdTouch.print(humidity);
    LcdTouch.drawButton(999, 150, 200, 30, 30, 2, WHITE, RED, "EXSIT", 5);

    if (LcdTouch.touched()){
        LcdTouch.readTouch();
        x = LcdTouch.xTouch;
        y = LcdTouch.yTouch;
        ButtonNum = LcdTouch.ButtonTouch(x, y);
        if (ButtonNum == 999){
            screenMain();
        }
    }
    delay(5000);
    InitDisplay();
}
```

פונקציה שמבצעת את המשאבה, ממתינה 3 שניות וקוראת לכיבוי המשאבה:

```
void playWatering(){
    digitalWrite(pumpPin, HIGH);
    delay(3000);
    stopWatering();
}
```

פונקציה שמכבה את המשאבה:

```
void stopWatering(){
| digitalWrite(pumpPin, LOW);
}
```

## 15.2 הסבר פונקציות

שם הפונקציה	תפקידה	ערכים שהוא מקבלת או מחזירה
screenMain()	הציג הودעת פתיחה	---
screenInitHygrometer()	הגדרת רמת הלחות האופטימלית	---
keyboard(int ButtonNum)	מציגה את הערך של הכפתור שנלחץ	מקבלת את מספר הכפתור
InitWateringTime()	הגדרת זמן השקיה מועדף	---
screenOK()	הציג אישור של הגדרת הנתונים	---
InitDisplay()	אתחול התצוגה	---
playWatering()	הפעלת המשאבה	---
stopWatering()	כיבוי המשאבה	---

## 16. תקשורת

מלבד התקשרות של הצג הגרפי אותה הסברנו לעיל, שאר הרכיבים שלנו מתקשרים בתקשורת אנלוגית וdigיטלית פשוטה ולא בתקשורת מורכבת כמו I2C, PWM, או SPI. להלן פירוט סוגי התקשרות עבור כל רכיב:

### משאבה - DC Motor Pump Water Pump VERTICAL TYPE

סוג תקשורת:

- **PWM (Pulse Width Modulation)**: אם יש צורך לשלוט על מהירות המשאבה, ניתן להשתמש באותות PWM מהבקר.
- **דיגיטלית פשוטה**: אם מפעילים או מכבים את המשאבה ללא שליטה על מהירות, ניתן להשתמש באות דיגיטלי פשוט (LOW/HIGH) דרך מסר או טרנזיסטור.

חיבור לבקר:

- **מסר (Relay)**: מפעיל או מכבה את המשאבה על ידי אות דיגיטלי.
- **דריבר מנוע (Motor Driver)**: ניתן להשתמש באות PWM מהבקר כדי לשלוט על מהירות המשאבה.

### טישן אור - LDR 5mm

סוג תקשורת:

- **אנלוגית**: ה-LDR משנה את ההתנגדות שלו בהתאם לעוצמת האור, זה מתורגם למתח שנקרא על ידי הבקר בכניסה האנלוגית.

חיבור לבקר:

- **כניסה אנלוגית**: LDR מחובר למחולק מתח והמתוך שנוצר נקרא על ידי כניסה אנלוגית של הבקר.

### טישן לחות אדמה - YL-69 Soil Moisture Detection Sensor Module

סוג תקשורת:

- **אנלוגית**: מספק אותן אנלוגיות המתאר את רמת הלחות.
- **דיגיטלית**: מספק אותן דיגיטליicas אשר הלחות עובר סף מסוים (בעזרת מודול 38-YL).

חיבור לבקר:

- **כניסה אנלוגית:** לחיבור האות האנלוגי.

- **כניסה דיגיטלי:** לחיבור האות הדיגיטלי.

#### **סיכום סוגי התקשרות**

- **תקשורת אנלוגית:** ה-LDR והחישון 69-YL מספקים אותן אנות אנלוגיים לקריאה על ידי הבקר.

- **תקשורת דיגיטלית פשוטה:** ה-69-YL מספק גם אותן דיגיטלי לחיבור לבקר, ומהשאבה יכולה להיות מחוברת דרך מסר להפעלה/כיבוי פשוטים.

## 17. תקלות

כמובן שחלק מהתהליכי זה תקלות,

אחד מהן הייתה עם המשאבה שגם לאחר חיבור תקין לבקר היא לא עבדה, הסתבר שהיא זקוקה לספק כח.

המתוח בבר לבנו מוגבל ל- 5V, אך היה צריך בספק כוח הוא יחידה אלקטרונית שתפקידה לספק למעגלים אלקטרוניים את המתוח והזרם החשמלי הדרושים להפעלתם.

יש מגוון רחב של רכיבים היכולים לשמש לאספקת כוח לארדוינו, אני השתמשתי בספק V12.

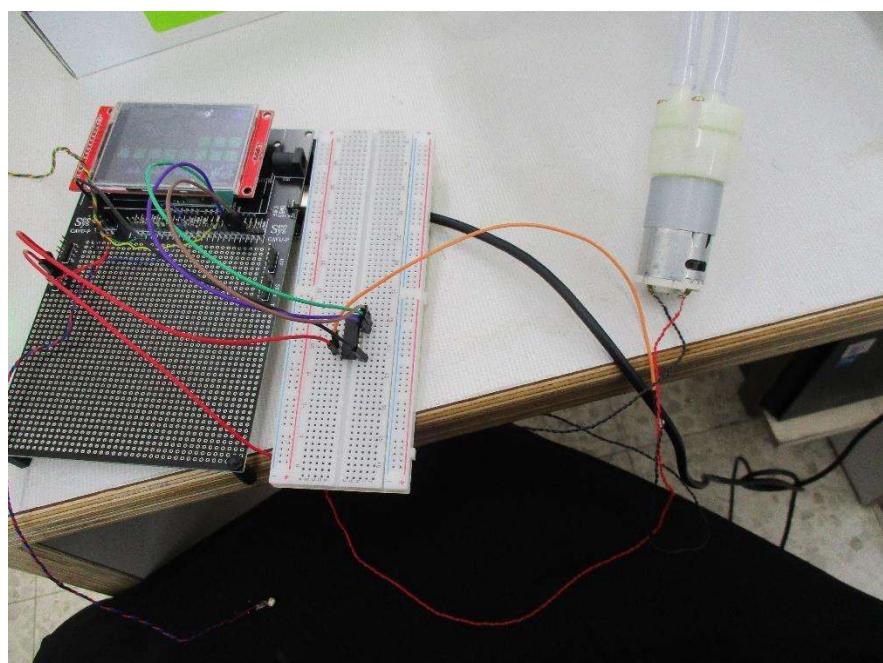
ספק V12 - ספקים אלו מתחררים לשקע בקייר וממירים את הזרם חילופין לזרם ישיר, בנוסף הם מפחיתים את המתוח והזרם ובכך מאפשרים אספקת כוח, בדרך יהיה להם חיבור עגול המתאים לארדוינו, לחילופין ניתן גם לחבר אותו לחו"ל. יתרון: מספק כוח קבוע יוכל לספק גם 12V לרכיבים. חיסרון: מחובר לשקע.



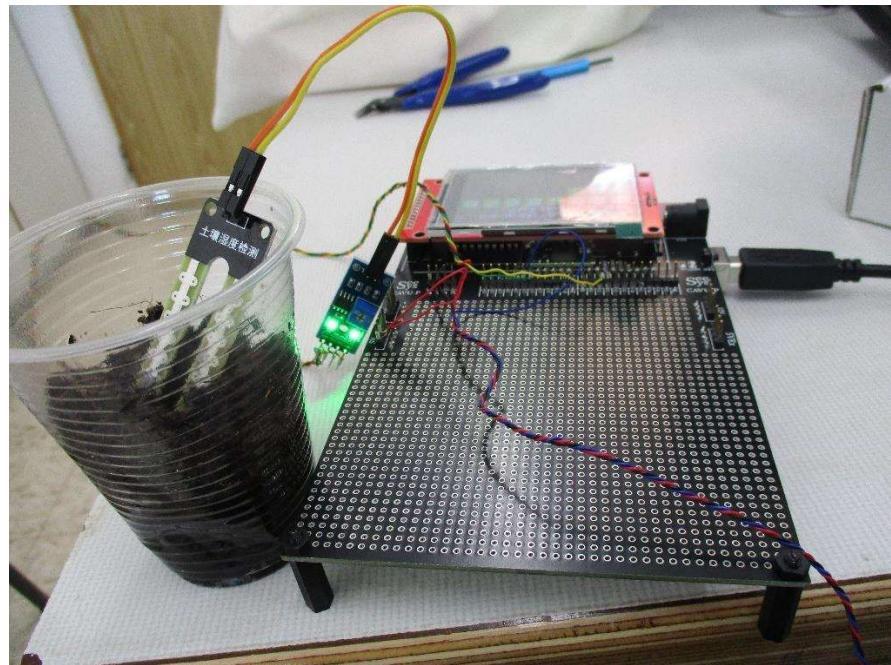
**ספק V12**

## 18. תייעוד הפרויקט

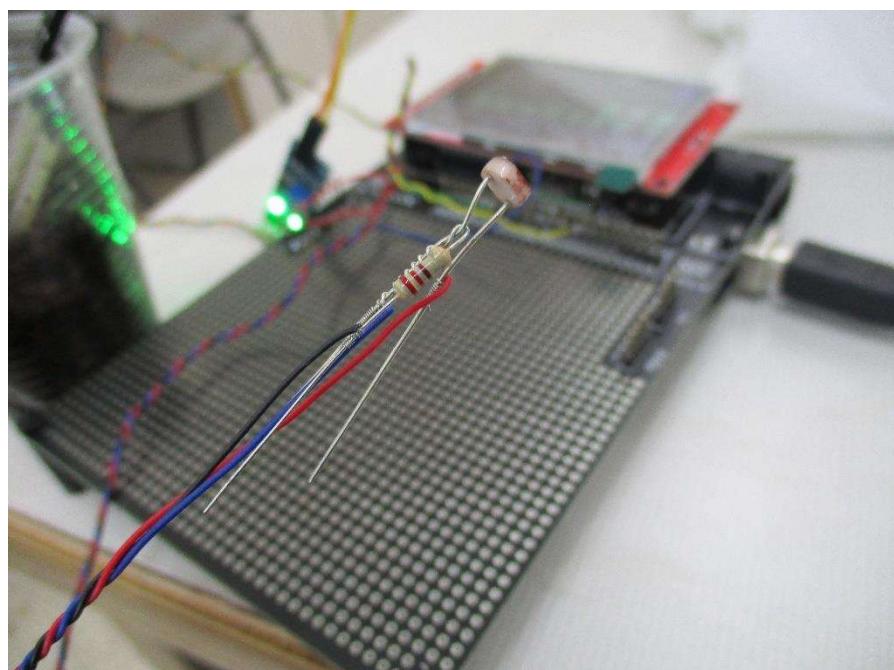
- בדיקת המשאבה באמצעות חיבור למטריצה



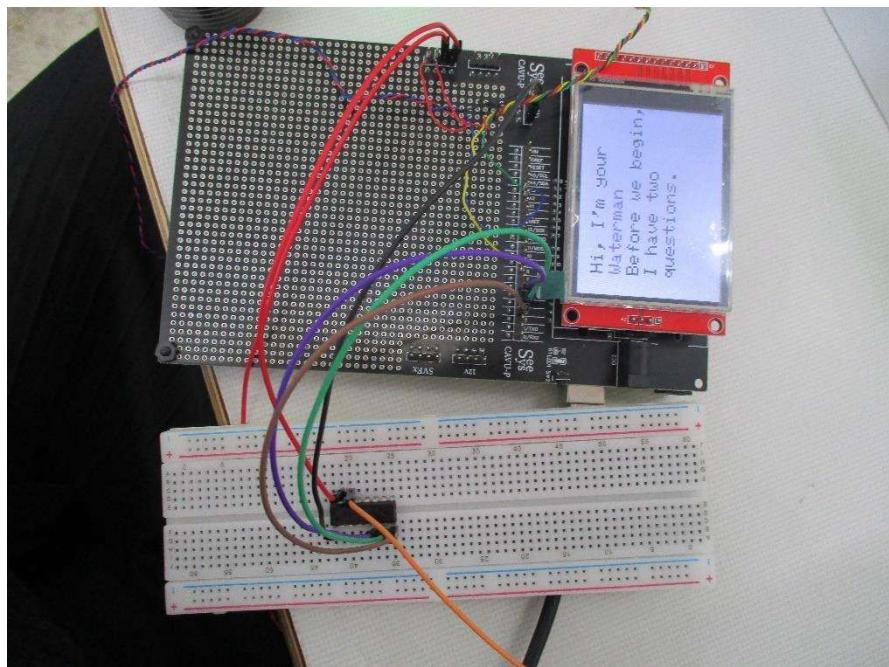
- בדיקת חיישן הלחות



- בדיקת חיישן האור



- לוח הארדואינו נעזר במטריצה במהלך העבודה



## 19. סיכום

במהלך הפרויקט למדתי רבות על שינויים שונים, על הבקרים הקיימים בשוק ועל אופן התכונות שלהם. התמודדתי עם אתגרים כמו יכול החישנים והבטחת אמינות המערכת למשך זמן.

**נקודות לשיפור:**

- אמינות המערכת: למצוא פתרונות לגיבוי במידה והמערכת נכשלה, כמו למשל חישון נוספת או אזעקה במקרה של תקלה.
  - חישוב כמות המים להשקיה לפי רמת הלחות.
- ישן מספר דרכים להרחב ו לשפר את המערכת:
1. אפליקציה לניטור ושליטה מרוחק: פיתוח אפליקציה שתאפשר למשתמש לעקוב אחר מצב הגינה בזמן אמת ולהתאים את הגדרות ההשקיה לפי הצורך.
  2. שילוב עם תחזית מגiaoור: חיבור המערכת לשירות תחזית מגgeoor על מנת למנוע השקיה במקרה של גשם צפוי.
  3. אנליזת נתונים: איסוף וניתוח נתונים למשך זמן על מנת לשפר את הדיווק והיעילות של המערכת.

הפרויקט הצלח להראות את הפוטנציאלי הרב של מערכות משובצות מחשב בניהול גיננות חכומות. עם הפיתוחים העתידיים והרחבות נוספות, ניתן לשפר את המערכת ולהפוך אותה לכלי שימושי בחקלאות ובתחזוקת גינות פרטיות וציבוריות.

## 20. מקורות

- האתר של אבי חיון וויאל כהן

<http://www.2all.co.il/Web/Sites/hayun/PAGE6.asp>

- אתר Arduino

[/https://www.arduino.cc](https://www.arduino.cc)

- ספר Arduino חלקיים א' ב', שי מלול

- הסבר על חישון הלחות

<https://www.handasiya.co.il/soilsensor>

- הסבר על חישון האור

<https://gabbyshimoni.wixsite.com/arduino-programming/ldr>

- օբק כת

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