



אוניברסיטת בן-גוריון בנגב  
Ben-Gurion University of Negev

# פרויקט גמר

קורס: מבנה מחשבים ספרתיים

361-1-4191

**Light source and object proximity  
detector system**

מערכת לגילוי מקורות אור  
וניטור אובייקטים במרחב

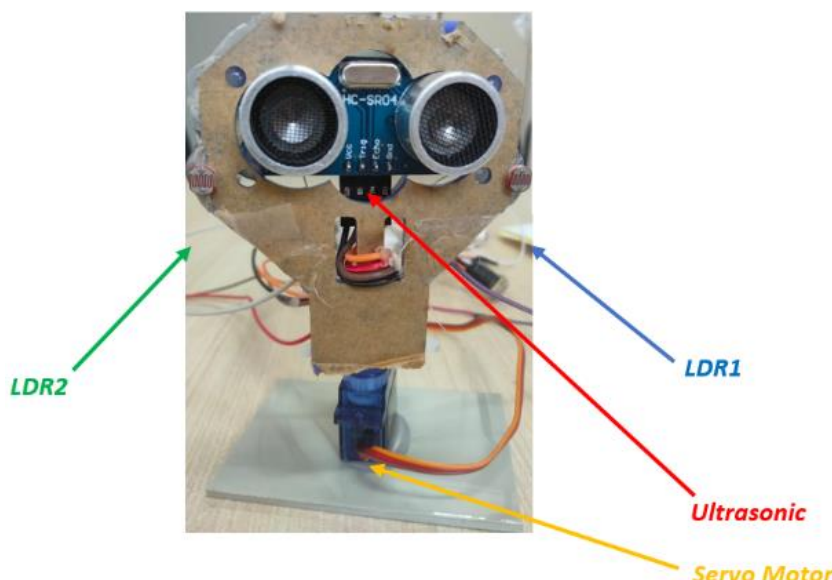
06/07/2025

## Table of Contents

|                   |   |    |
|-------------------|---|----|
| <a href="#">A</a> | מטרת הפרויקט:   | 3  |
| <a href="#">B</a> | תיאור משימת הפרויקט:  | 4  |
|                   | • ממשק משתמש בצד ה-PC:  | 4  |
| 1.                | Objects Detector System : (משקל 30%)                              | 5  |
| 2.                | Telemeter : (משקל 10%)  | 5  |
| 3.                | Light Sources Detector System : (משקל 30%)                        | 5  |
| 4.                | Light Sources and Objects Detector System : (סעיף בונס במשקל 20%) | 6  |
| 5.                | File Mode : (משקל 30%)  | 7  |
| <a href="#">C</a> | הסברים טכניים – חיישן ומנוע סרבו:                                 | 9  |
| 1.                | חיישן מרחק Ultrasonic (רכיב HC-SR04):                             | 9  |
| 2.                | מנוע Servo Motor:   | 11 |
| 3.                | ביצוע סריקה ע"י מנוע ה-Servo:                                     | 11 |
| 4.                | חיישן LDR כגלאי מרחק של מקור אור:                                 | 12 |
| 5.                | חיבורי חומרה והקצאת רגלי הבקר (ערכת פיתוח אישית):                 | 12 |
| <a href="#">D</a> | דו"ח מכין: (משקל 10%)   | 13 |
| <a href="#">E</a> | מבנה הציון בפרויקט:   | 13 |

## A. מטרת הפרויקט:

- i. תכנון ומימוש מערכת מבוססת MCU לגילוי מקורות אור וניטור אובייקטים במרחב באמצעות מד מרחק אולטראסוניק, חיישני אור LDR ומנוע Servo. סריקת המרחב תבוצע בגזרה של 180 מעלות באמצעות תנועת מנוע Servo, מדידת מרחק באמצעות מד המרחק בעל טווח מדידה בין  $2\text{cm} \div 450\text{cm}$ . בקרת התנועה הזוויתית של מנוע Servo תהא מבוססת PWM.
- ii. במסגרת הפרויקט יפותח קוד בשפת C++/C למימוש מערכת Embedded מבוססת גרעין הפעלה FSM וכתובת המערכת במתודולוגיה של שכבות אבסטרקציה למימוש מערכת Embedded מרובת חיישנים תחת משטר Hard Real Time. מחשב PC ישמש לצורך ממשק GUI ( [PySimpleGUI](#), [Tkinter](#), etc ) למשתמש ולתצוגה לכל פעולה המוגדרת במערכת ודורשת תצוגה וממשק למשתמש. ה-MCU יחובר למחשב ה-PC באמצעות תקשורת טורית אסינכרונית בסטנדרט RS-232.
- iii. ממשק למשתמש בצד ה-PC יאפשר קביעת פרמטרים, שליחת קבצים ופקודות High-level ל-MCU. הממשק בצד ה-PC יכתב בשפה עילית (לבחירתכם: Python, Matlab, C++, JAVA או שימוש במעטפת C#) ויתמוך במימוש של תקשורת טורית בין הבקר ל-PC.
- iv. הממשק יאפשר העברת קבצים (שליחה וקבלה) בין צד מחשב ולבין צד בקר.
- v. הגדרת הפרויקט המתוארת בקובץ זה מכילה דרישות של מערכת Embedded בלבד, שלב ראשון בתהליך הפיתוח של המערכת לאחר הבנת הדרישות הוא שלב אפיון המערכת. בשלב זה יש צורך לחקור ולהבין את המגבלות ההנדסיות של המערכת, לכתוב גרף מפורט של גרעין ההפעלה FSM של המערכת, ולכתוב תיאור של האלגוריתמים השונים הנדרשים למימוש במערכת. בשלב חשוב זה נדרש לבצע תכנון מדויק ולפתור את כל השגיאות הלוגיות הקיימות בתכנון (במקביל לשלב זה ניתן לכתוב דרייברים בשכבה המתאימה בנפרד לכל מודל חומרה בפרויקט). רק לאחר מכן ניתן להתחיל בכתובת קוד המערכת ומימוש האלגוריתמים השונים. **זכרו, הגדרת הפרויקט משאירה שטח "אפור" בו כל זוג צריך להביא את עצמו לידי ביטוי ויצירתיות במימוש אופטימאלי מבחינת סיבוכיות מקום, סיבוכיות זמן, צריכת הספק לצורך מימוש מערכת משובצת מחשב רובסטי (מערכת הפועלת בצורה יציבה ואינה "נתקעת") בזמן תגובה RT תחת מגבלות הנדסיות של המערכת.**
- vi. להלן המחשה וויזואלית בלבד של ציוד קצה חומרתי למימוש מערכת רדאר מבוססת MCU לניטור וגילוי אובייקטים ומקורות אור במרחב באמצעות מד מרחק אולטראסוניק, חיישני LDR, ומנוע Servo.



## B. תיאור משימת הפרויקט:

- פיתוח מערכת לגילוי מקורות אור וניטור אובייקטים במרחב באמצעות מד מרחק אולטראסוניק, שני חיישני אור LDR ומנוע Servo לצורך סריקת המרחב.
  - ארכיטקטורת התוכנה של המערכת נדרשת להיות מבוססת פרדיגמת תכנות גרעין הפעלה מסוג **FSM** **Simple** המבצעת קטע קוד השייך לאחד ממצבי המערכת בהינתן בקשה של פסיקת RX שמגיעה מה PC לבקר דרך ערוץ התקשורת למודול פריפריאלי UART כאשר שליחה וקבלת מידע בין הבקר ל- PC מבוססת תקשורת טורית אלחוטית Bluetooth. **קוד המערכת נדרש להיות מחולק לשכבות אבסטרקציה כך שיהיה נייד (portable) בקלות בין משפחות בקר MSP430 ע"י החלפת שכבת ה- BSP בלבד.**
  - טרם שלב כתיבת הקוד בשלב התכנון נדרש לשרטט גרף של דיאגרמות FSM מפורטות, אחת של ארכיטקטורת התוכנה של המערכת בצד MCU והשנייה של חלק התמיכה בתקשורת באפליקציה בצד מחשב ולצרפן לדו"ח מכין. גרף של דיאגרמת FSM בצד MCU המצבים אלו הצמתים והקשתות אלו המעברים ממצב למצב בגין בקשות פסיקת RX (המסווגות לקליטת מידע מסוג Command ומסוג Data, כפי הנלמד בניסוי מעבדה 4).
  - אסור לבצע השהייה ע"י שימוש ב poling למעט עבור debounce ברוטינת שירות של בקשות פסיקה בגין לחצנים.
  - בתחילת פעולת המערכת (בלחיצה על כפתור RESET), הבקר נמצא במצב שינה.
- הערה: כפתור RESET מותר לשימוש אך ורק לאתחול המערכת בלבד**
- רמת הדיוק, זמן תגובת המערכת והביצוע בהתאם לדרישות מהווה חלק חשוב בהערכת הפרויקט.
  - מקוריות העבודה היא חלק חשוב בביצוע הפרויקט, במקרה של העתקה, הפרויקטים של שני הצדדים ייפסלו.
  - עקב מגבלה של גודל ה RAM עליכם להשתמש בתבונה בזיכרון ה FLASH (ראו חומר עזר במודל).
  - נדרש לעבוד בסביבת פיתוח **CCS IDE** מבוססת **Eclipse**.
  - ממשק משתמש בצד ה- PC :
- בצד מחשב נדרשת תמיכה בממשק למשתמש על גבי מסך ה- PC הכולל יכולת הצגה וויזואלית דינאמית על גבי מסך ה- PC. עליכם לכתוב את המעטפת והממשק (GUI) בצד ה- PC בכל שפה שתבחרו, C++ , Python, Matlab, JAVA או שימוש במעטפת C# - מומלץ. במעטפת זו תצטרכו לתמוך בתקשורת טורית אסינכרונית של המחשב עם הבקר מבוססת סטנדרט RS-232 לצורך העברת תווים וקבצים (העברת הקובץ תו אחר תו) בין המחשב לבקר וההיפך. מצב ברירת המחדל הוא:
- 9600 BPS , 8-bits , 1 Start , 1 Stop , (none) No parity**
- עבור פעולות עם מספרים לא שלמים נדרש לעבוד עם ייצוג נקודה קבועה בשימוש Q-Format (ולא בנקודה צפה בשימוש טיפוסים float, double) כפי שנלמד בניסוי מעבדה 2 (ובקורס מבוא למחשבים).
  - נדרשת תמיכה בגילוי שגיאות בשכבה הפיזית בתוך הפריים הנקלט (גודל הפריים הנקלט יכול להיות קבוע או משתנה כתלות בתכנון הנדסי של המערכת) המבוסס על מנגנון checksum 256 ובקשה לשליחה נוספת של הפריים (במידה ולא התקבל Acknowledge בזמן Timeout מוגדר מראש או שהתקבל Non-Acknowledge) במקרה שהתגלתה שגיאה בפריים הנקלט.
  - המידע הנקלט ברגל RX ועבור המידע המשודר ברגל TX נכתוב לתוך circular buffer מבוסס מנגנון FIFO.
  - הדגש בתכנון הוא של זמן תגובת המערכת עם רמת דיוק ביצוע של תוכן קובץ ה scripts ואמינות תוכן המידע הנשלח/מתקבל דרך ערוץ התקשורת.

**1. Objects Detector System : (משקל 30%)**

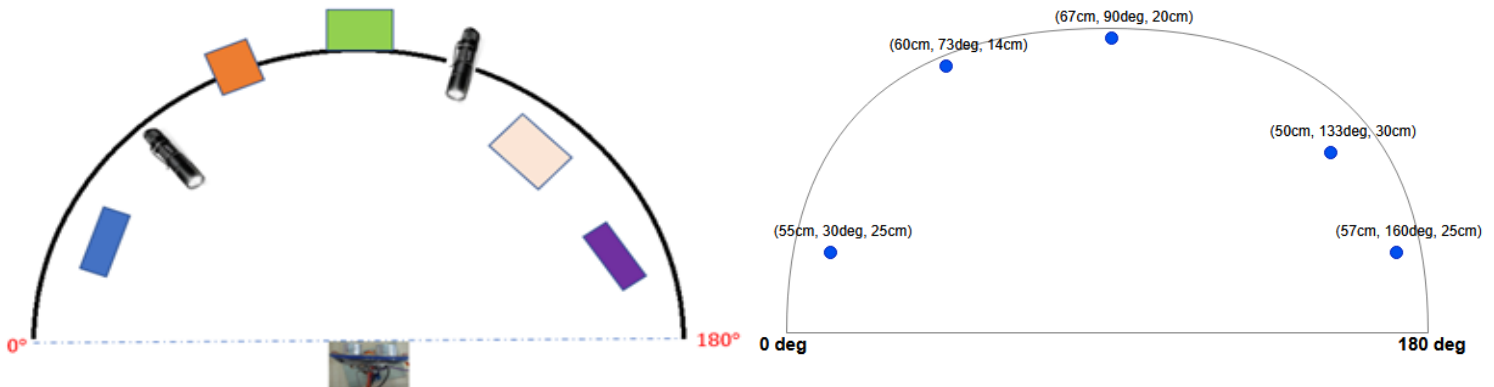
מימוש מערכת Objects Detector System לניטור אובייקטים (באופן דינאמי) במרחב בביצוע סריקה אחת בלבד בהיקף סריקה של 180 מעלות וברמת דיוק אופטימאלית.

**הסבר:**

חיישן המרחק סורק את המרחב ב-180 מעלות (מזווית 0 ל-180 מעלות) סביב נקודת המרכז של ידית מנוע Servo במרחק מיסוך בטווח מקסימאלי של עד ארבע מטר (מרחק שממנו והלאה אנו מתייחסים לערך הנמדד כמחוץ לתחום ואינו נלקח בחשבון) המוגדר מראש דרך הממשק למשתמש. לאחר סריקה בודדת מידע של גילוי האובייקטים מוצג על מסך המחשב כאשר עבור כל אובייקט מצוינים שלושה פרמטרים:  $\rho$  מרחק האובייקט מנקודת המרכז, זווית  $\phi$  של מיקום האובייקט בייחס לזווית אפס,  $l$  רוחב האובייקט  $(\rho, \phi, l)$ .

**הערה:** איסוף מידע עבור סריקה של המרחב מורכב מ-180 דגימות (דגימה לכל זווית), כאשר כל דגימה מורכבת ממספר ערכים (ערך input capture ברוחב 16-bit השקול למרחק הנמדד מחיישן המרחק, ערך זווית ברוחב 8-bit, סה"כ 3-Byte). בשלב התכנון יש צורך להחליט איזה חלק של עיבוד הנתונים יתבצע בצד הבקר ואיזה חלק יתבצע בצד המחשב, נדרש לשלוח את המידע בערוץ התקשורת בצורה דחוסה כך שמספר הבתים הנשלח מינימאלי (אסור לשלוח תרגום ASCII של המידע).

**שאלה תיאורטית:** כתבו ביטוי וחשבו את הנצילות של המידע הנשלח לכל דגימה בהתאם לאופן המימוש במערכת



**2. Telemeter : (משקל 10%)**

נדרש למקם את מנוע Servo בזווית נתונה לבחירה דרך ממשק למשתמש ולהציג על מסך ה-PC את המרחק הנמדד מחיישן המרחק בזמן אמת באופן דינאמי ברזולוציה של cm (ללא רישום היסטוריית מדידות).

**3. Light Sources Detector System : (משקל 30%)**

מימוש מערכת Light Sources Detector System לניטור מקורות אור (באופן דינאמי) במרחב בטווח של עד חצי מטר בביצוע סריקה אחת בלבד בהיקף סריקה של 180 וברמת דיוק אופטימאלית.

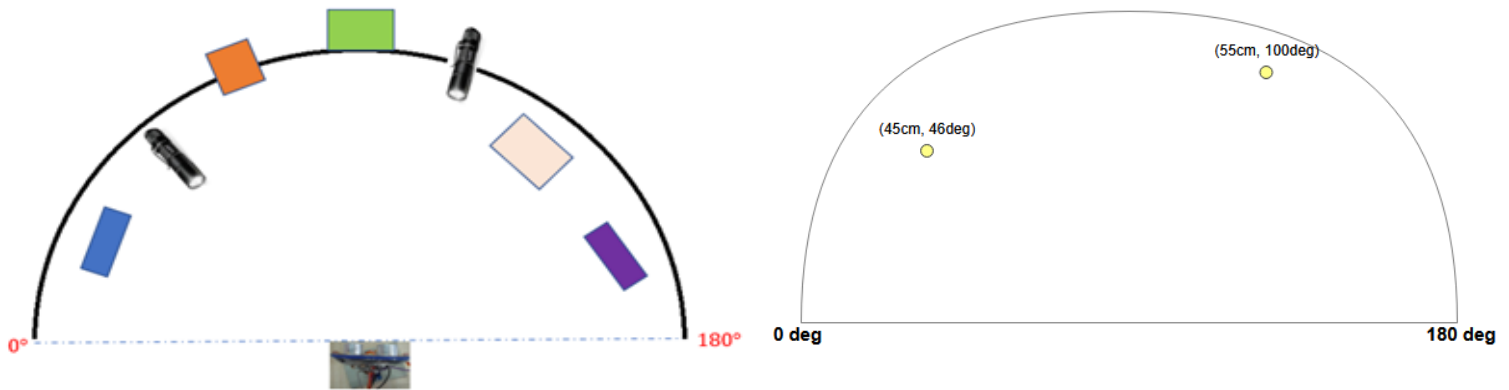
**הסבר:**

בעזרת שני חיישני אור מסוג LDR הממוקמים מצדי חיישן המרחק ניתן לבצע גילוי של מקורות אור מבחינת מיקום ומרחק (בטווח של עד חצי מטר) בתוך טווח סריקה של 180 מעלות סביב נקודת המרכז

## של ידית מנוע Servo ולהציג את תוצאות המיקום והמרחק של מקורות האור על מסך ה-PC דרך הממשק למשתמש.

**הערה:** איסוף מידע עבור סריקה של המרחב מורכב מ-180 דגימות (דגימה לכל זווית), כאשר כל דגימה מורכבת ממספר ערכים (ערך ADC10 ברוחב 10-bit השקול למרחק הנמדד מחיישן המרחק, ערך זווית ברוחב 8-bit, סה"כ 3-Byte). בשלב התכנון יש צורך להחליט איזה חלק של עיבוד הנתונים יתבצע בצד הבקר ואיזה חלק יתבצע בצד המחשב, נדרש לשלוח את המידע בערוץ התקשורת בצורה דחוסה כך שמספר הבתים הנשלח מינימאלי (אסור לשלוח תרגום ASCII של המידע).

שאלה תיאורטית: כתבו ביטוי וחשבו את הנצילות של המידע הנשלח לכל דגימה בהתאם לאופן המימוש במערכת

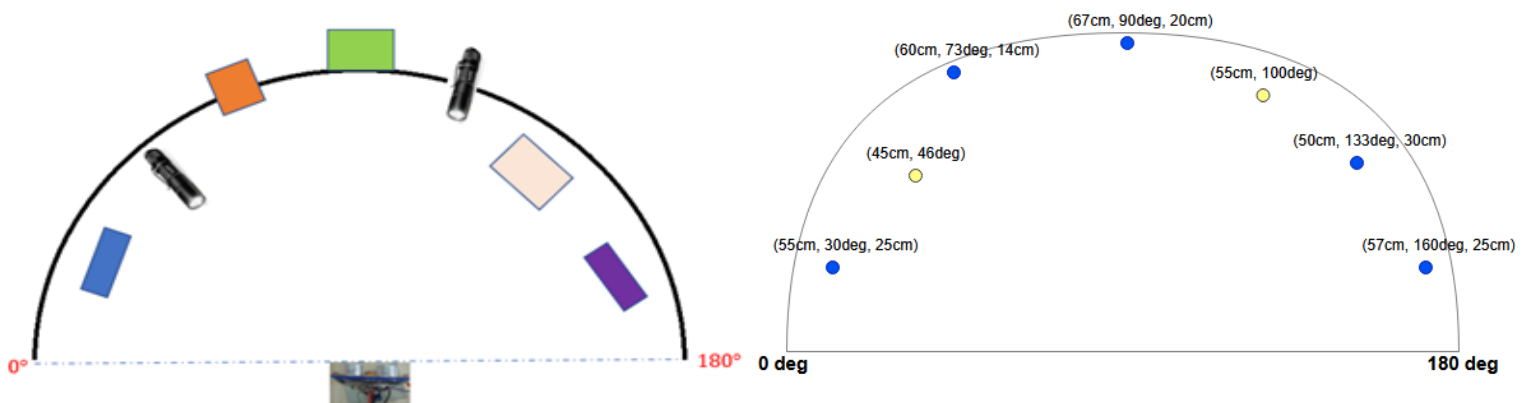


### 4. Light Sources and Objects Detector System : (סעיף בונס במשקל 20%)

מימוש מערכת Light Sources and Objects Detector System, מערכת זו מבצעת שילוב של הדרישות המפורטות בסעיפים 1,3 בתפריט, לניטור אובייקטים ומקורות אור יחד (באופן דינאמי) במרחב בביצוע סריקה אחת בלבד בהיקף סריקה של 180 מעלות וברמת דיוק אופטימאלית.

**הערה:** איסוף מידע עבור סריקה של המרחב מורכב מ-180 דגימות (דגימה לכל זווית), כאשר כל דגימה מורכבת ממספר ערכים (ערך input capture ברוחב 16-bit השקול למרחק הנמדד מחיישן המרחק, ערך ADC10 ברוחב 10-bit השקול למרחק הנמדד מחיישן המרחק, ערך זווית ברוחב 8-bit, סה"כ 5-Byte). בשלב התכנון יש צורך להחליט איזה חלק של עיבוד הנתונים יתבצע בצד הבקר ואיזה חלק יתבצע בצד המחשב, נדרש לשלוח את המידע בערוץ התקשורת בצורה דחוסה כך שמספר הבתים הנשלח מינימאלי (אסור לשלוח תרגום ASCII של המידע).

שאלה תיאורטית: כתבו ביטוי וחשבו את הנצילות של המידע הנשלח לכל דגימה בהתאם לאופן המימוש במערכת



### הקדמה:

לצורך תמיכה בקבצים במערכת של אחסון עד 10 קבצים (בגודל משתנה) במרחב כתובות של 2kB על פני ארבעת הסגמנטים האחרונים בזיכרון Flash באזור ה- code memory (Seg0-Seg3). כתיבה לארבעת הסגמנטים האחרונים כדי לא לדרוס את קוד המערכת (נדרש לוודא באלו סגמנטים מאוכלס קוד המערכת),

גודל סגמנט באזור זה (המכיל 32)  $\frac{16kB}{512B} = 32$ ,  $\frac{code\ memory\ size}{segment\ size}$  , 32 סגמנטים) הוא 512B.

המערכת נדרשת לתמוך בשני סוגי קבצים הבאים:

- **קובץ סקריפט (Script file):** תוכן הקובץ מכיל תרגום של קידוד בינארי של קובץ סקריפט
- **קובץ טקסט (Text file):** תוכן הקובץ מכיל טקסט המורכב מרצף פיזי של תווים (עם תו מסיים EOF).

### ניהול מערכת קבצים:

לצורך ניהול מערכת הקבצים נדרש להגדיר מבנה נתונים מסוג struct המשמש לניהול מערכת קבצים ומכיל לכל

קובץ מצביע, שם קובץ, גודל הקובץ, סוג הקובץ וכו'. שמירת מידע זה נדרש בזיכרון FLASH באזור

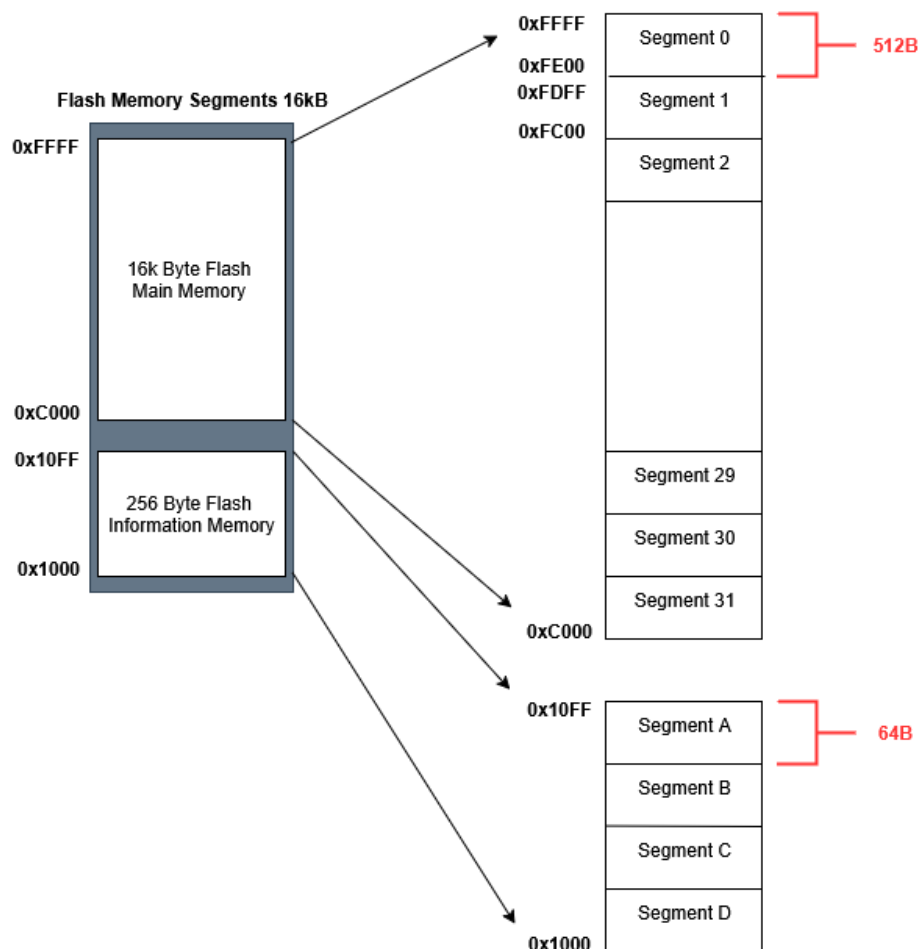
information memory כמופיע בקוד לדוגמה בו כל סגמנט (מתוך שלושה סגמנטים B-D, כאשר סגמנט A

**אסור בשימוש**) בגודל 64B. זאת כדי שגם כאשר המערכת מופעלת מחדש המידע לניהול הקבצים נשמר (**מידע**

הקשור לכיול חומרת קצה, במקרה זה 10 ערכי כיול לכל אחד מחיישני ה- LDR נדרש לאחסון באזור זה

בזיכרון ה- Flash, נדרש להוסיף בממשק למשתמש אפשרות לבחירת ביצוע הכיול כמתואר בסעיף C4

ומשקל תמיכה בכיול 3% מסעיף התפריט (File Mode).



בצד הבקר נדרש לתמוך ביכולת שליחה וקבלה של עד עשרה קבצים באופן בלתי תלוי בבחירת פעולה על הקובץ (יכולת של קריאת קובץ על גבי מסך ה-LCD עם יכולת גלילה בשימוש לחצן או הפעלת של קובץ script דרך ממשק למשתמש בצד PC). שליחת קובץ לבקר נעשית בלחיצת כפתור מתאים דרך הממשק למשתמש ולאחר מכן שליחת קבלת הקובץ מהמחשב לצד הבקר. קובץ text נשלח לבקר באופן טורי תו אחר תו (ללא קידוד) כאשר שליחת קובץ script מבוצעת ע"י שליחת הקידוד של הקובץ תו אחר תו. לאחר קבלת הקובץ בצד הבקר, תישלח הודעת Acknowledge למסך ה-PC.

### הגדרת קובץ Script:

קובץ script הינו קובץ מקודד של פקודות High Level המוגדרות מראש (לפי הגדרת Script ISA). בשימוש קובץ script ניתן להפעיל את חלקי המערכת באופן אוטומטי. להלן הגדרת Script ISA:

| OPC<br>(first Byte) | Instruction | Operand<br>(next Bytes) | Explanation  |
|---------------------|-------------|-------------------------|--|
| 0x01                | inc_lcd     | x                       | Increment from zero to <b>x</b> with a delay <b>d</b> onto the LCD   |
| 0x02                | dec_lcd     | x                       | Count down from <b>x</b> to zero with a delay <b>d</b> onto the LCD  |
| 0x03                | rra_lcd     | x                       | Rotate <b>right</b> onto LCD from pixel index 0 to pixel index 31 a single char <b>x</b> (ASCII value) with delay <b>d</b>                               |
| 0x04                | set_delay   | d                       | Set the delay <b>d</b> value ( <b>units of 10ms</b> ), where its default value is 50   |
| 0x05                | clear_lcd   |                         | Clear LCD  |
| 0x06                | servo_deg   | p                       | Point the Ultrasonic sensor to degree <b>p</b> and show the degree and distance ( <u>dynamically</u> ) on the PC screen                                  |
| 0x07                | servo_scan  | l,r                     | Scan area between left <b>l</b> angle to right <b>r</b> angle ( <u>once</u> ) and show the degree and distance ( <u>dynamically</u> ) onto the PC screen |
| 0x08                | sleep       |                         | Set the MCU into sleep mode  |

```
inc_lcd 10
set_delay 30
dec_lcd 20
clear_lcd
servo_deg 35
inc_lcd 20
servo_scan 20,60
sleep
```

Script file as a text file

*Python based  
translator script*



```
010A
041E
0214
05
0623
0114
07143C
08
```

Encoded Script file

בבחירת תפריט קבצים בממשק למשתמש צד מחשב, לצורך תמיכה בקריאת קובץ בצד בקר נשתמש בלחצן PB0 לדפדוף שמות הקבצים על גבי מסך ה-LCD ובלחצן PB1 לבחירת קובץ ששמו מופיע בשורה הראשונה. לאחר בחירת הקובץ תוכנו יוצג על גבי מסך ה-LCD, נשתמש בלחצן PB0 לצורך גלילה, 32 תווים בכל פעם ובלחצן PB1 חזרה להצגת רשימת הקבצים. בצורה זו ניתן לבצע קריאה מהקבצים הנמצאים בזיכרון הבקר.



## C. הסברים טכניים – חיישן ומנוע סרבו:

### 1. חיישן מרחק Ultrasonic (רכיב HC-SR04):

בהוצאת פולס דרך הבקר ברוחב של **לפחות** 10μsec המהווה טריגר דרך רגל **Trigger** של החיישן (מרווח מינימאלי בין טריגר לטריגר הוא 60msec, כלומר תדר עבודה מקסימאלי של 16.7Hz), **בסיום** הפולס חיישן המרחק "יורה" גל קול (Sound wave) באורך שמונה מחזורים בתדר 40kHz לכיוון האובייקט וקולט את ההחזרים המגיעים ממנו. מעגל חשמלי הנמצא בחיישן ממיר את החזרי גל הקול וממיר אותו לפולס היוצא מרגל **Echo**, באורך הזמן שעבר מרגע שידור גל הקול ועד לקבלת ההחזרים מהאובייקט הנמצא מול החיישן. הפולס היוצא מרגל **Echo** של החיישן נכנס לרגל הבקר בעל יכולת פסיקה (דפי המידע והמפרט נמצאים **באתר הקורס ב-Moodle**). טווח המדידה המעשי הוא  $2cm \div 450cm$ .

#### סרטון הסבר: Using of Distance sensor Ultrasonic

מדידת המרחק תתבצע בעזרת הנוסחה הבאה (מימוש בעזרת **Input Capture בלבד**):

$$Range[cm] \cong Echo\_high\_level\_time[\mu sec] \cdot \frac{34,000 \left[ \frac{cm}{sec} \right]}{2}$$

when:  $c = 34,000 \left[ \frac{cm}{sec} \right]$  is the speed of sound

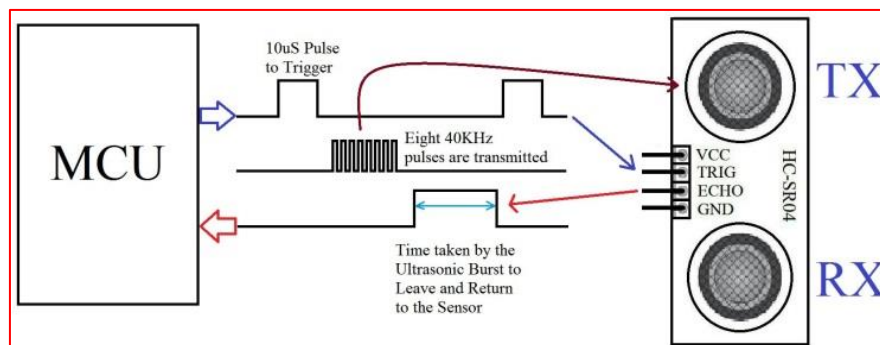
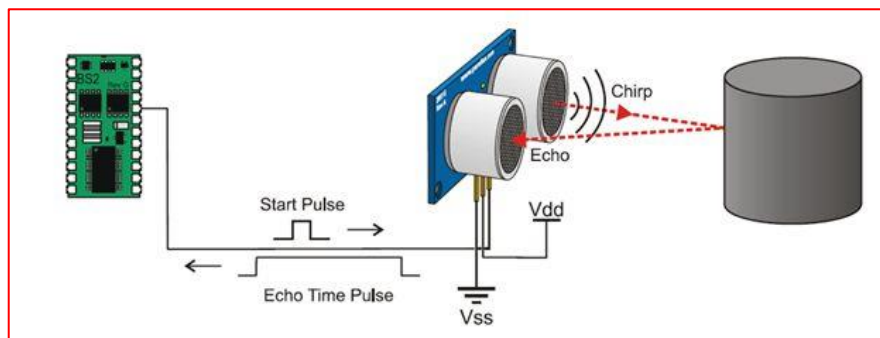
לצורך דיוק מרבי, נציין שמהירות הקול באוויר תלויה בטמפרטורה, לצורך ציוד נצטרך להתחשב בכך לפי הנוסחה הבאה ([Speed of sound](#)):

$$speed\ of\ sound\ c \left[ \frac{m}{sec} \right] = 331.3 + 0.606 \times Temperature\_in\_Celsius$$

עבור טמפרטורת החדר 25 מעלות צלזיוס (הנחת העבודה בפרויקט),  $c = 346.45 \left[ \frac{m}{sec} \right] = 34,645 \left[ \frac{cm}{sec} \right]$ ,

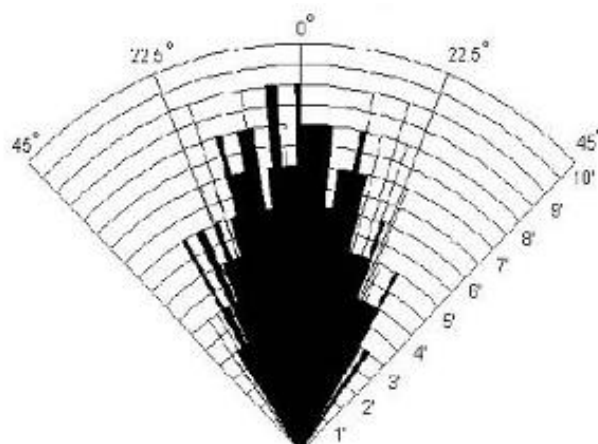
**הערה:** התחשבות בטמפרטורה מאפשרת לשפר את הדיוק במדידת המרחק. לצורך מדידת טמפרטורה

**משמש חיישן טמפרטורה** (בפרויקט זה לא נדרש לבצע מדידת טמפרטורה, הנחת טמפרטורת חדר 25°C).



| Label Name         | Description                    | MSP430 Pin                  |
|--------------------|--------------------------------|-----------------------------|
| UltraSonic_Echo    | אות ECHO המוחזר מחיישן המרחק   | Timer Input Capture         |
| UltraSonic_Trigger | אות Trigger הנכנס לחיישן המרחק | PWM output / Digital Output |

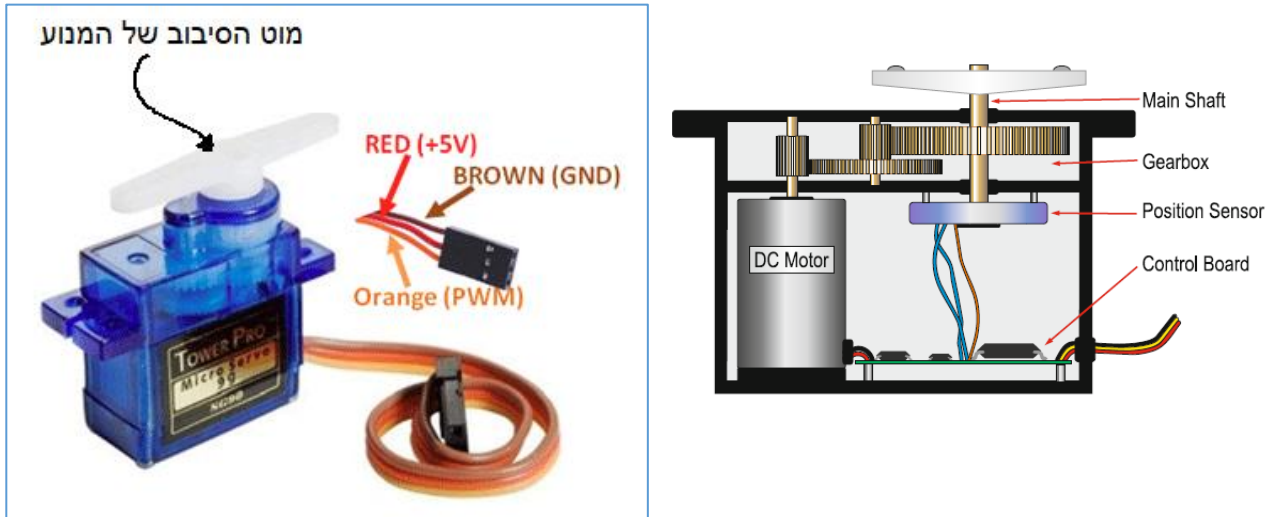
| HC-SR04                      |  | HY-SRF05   |
|------------------------------|--|--|
| Working Voltage              | 5 VDC  | 5 VDC  |
| Static current               | < 2mA  | <2 mA  |
| Output signal:               | Electric frequency signal, high level 5V, low level 0V | Electric frequency signal, high level 5V, low level 0V |
| Sensor angle                 | < 15 degrees   | < 15 degrees   |
| Detection distance (claimed) | 2cm-450cm  | 2cm-450cm  |
| precision                    | ~3 mm  | ~2 mm  |
| Input trigger signal         | 10us TTL impulse                                       | 10us TTL impulse                                       |
| Echo signal                  | output TTL PWL signal                                  | output TTL PWL signal                                  |
| Pins                         | 1. VCC<br>2. trig(T)<br>3. echo(R)<br>4. GND           | 1. VCC<br>2. trig(T)<br>3. echo(R)<br>4. OUT<br>5. GND |



*Practical test of performance,  
Best in 30 degree angle*

## 2. מנוע Servo Motor:

מנוע Servo ניתן להפעילו כך שמוט הסיבוב שלו, יסתובב בתחום זוויות בין  $0^\circ \div 180^\circ$  מעלות. המנוע מהווה עומס, המחובר לכרטיס ממשק המתווך בין הבקר לעומס. מצד אחד נחבר לכרטיס הממשק את ה-MCU המעביר מידע, מצד שני נחבר את המנוע המהווה עומס וצורך הספק גבוה. כרטיס המכיל את הדרייבר החומרתי מחובר למתח הפעלה של 5V ברגל המיועדת לכך (ראה תצלום הבא).



מוצא PWM המחובר למנוע מאפשר שליטה על מיקום זוויתי של מוט הסיבוב של המנוע (מיקום בזווית בין  $0^\circ \div 180^\circ$ ) בעזרת מוצא PWM מהבקר ערך ה-Duty Cycle של אות ה-PWM קובע את מיקום הזרוע. בהגדרות הבאות יצוין ה-PWM המינימאלי עבור מיקום זוויתי של  $0^\circ$  ו-PWM מקסימאלי עבור מיקום זוויתי של  $180^\circ$ . טווחי ביניים של PWM ייתן זוויות בתחום  $0^\circ \div 180^\circ$ .

### Servo Motors (Nominal Values)

$f_{max} = 40Hz \rightarrow T_{min} = 25msec$   
זווית של **0 מעלות**  $T_{on} = 0.6msec$   
זווית של **180 מעלות**  $T_{on} = 2.5msec$

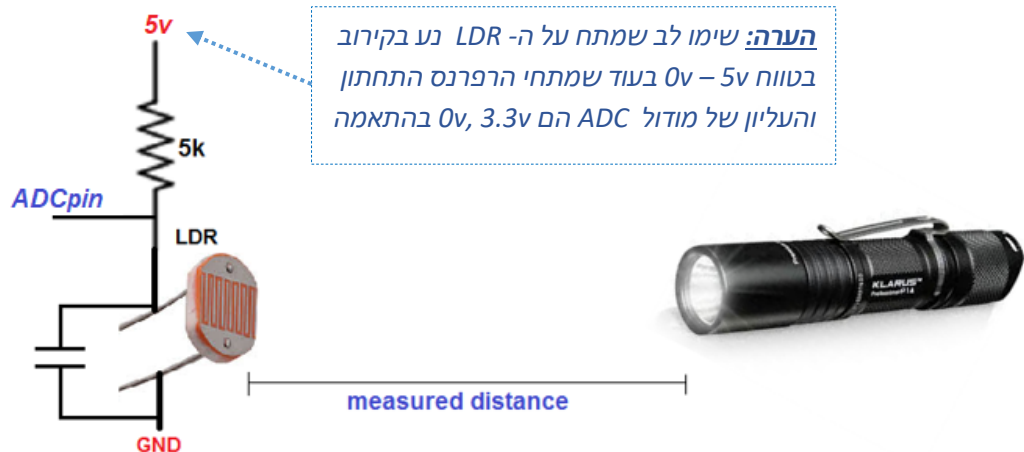
למציאת ערכים מעשיים חובה לבצע כיול חומרתי בשימוש מחולל אותות למציאת ערכי  $T_{on}$  ב  $0, 180$  מעלות

## 3. ביצוע סריקה ע"י מנוע ה-Servo:

- ✓ במצב של  $0^\circ$  - מוט הסיבוב של המנוע מכוונים למצב התחלתי קבוע וידוע מראש.
- ✓ כדי להגיע לזוויות ספציפיות (בשונה מסריקה) אפשר להכתיב PWM עם Duty Cycle מתאים **ללא** צורך בהגעה לזווית בעזרת צעדים קטנים.
- ✓ לצורך סריקה תקינה של מנועי ה-Servo יש לשנות זוויות בתחום של  $0^\circ \div 180^\circ$  ע"י שינוי הרגיסטרים השולטים על ה-Duty Cycle, בצעדים של בערך  $\pm 9$  עם השהייה של 4msec בין צעד לצעד (באופן כללי, הדבר תלוי לכמה צעדים תצטרכו לחלק את  $180^\circ$  של טווח סיבוב המנוע). הסיבה לכך, בשימוש רב של תנועות חדות המנוע עלול להשתגע ולצאת משליטה. כמובן שאין לחצות את גבולות הרגיסטרים השולטים על ה-Duty Cycle עבור גבולות של  $0^\circ \div 180^\circ$ , אחרת המנוע יתחיל לרעוד.
- ✓ כדי להימנע מרעשים, בנקודות זמן של איסוף דגימות מהחיישנים נדרש לכבות את אות PWM המזין את מנוע ה-Servo, במצב זה מוט הסיבוב של המנוע ננעל על הזווית בה הייתה טרם כיבוי האות.

#### 4. חיישן LDR כגלאי מרחק של מקור אור:

בעזרת הבקר ושימוש במודול ADC במחבר למחלק מתח בין נגד LDR ונגד של  $4.7k\Omega$ , נוכל לדגום מתח משתנה בתחום בין  $v_{min}$  ל-  $v_{max}$  הנופל על נגד LDR כתלות בעוצמת הארה עליו.



המרחק הנמדד יהיה בטווח 0cm - 50cm, לצורך מדידת מרחק של מקור האור נדרש להכין מבנה נתונים של המרחק הנמדד בצורת מערך חד-מימדי בגודל  $50 \times 2$ . אינדקס המערך  $d=0 \div 49$  מתאר את המרחק כאשר תוכן התא מתאר את ערך המתח הנמדד על ה-LDR.

בצורה זו קבלנו 50 דגימות, דגימת מתח אחת לכל ס"מ. את המדידות נבצע בכל 5cm (סה"כ 10 מדידות) ואת שאר הערכים נחשב בעזרת ההנחה שבין 2 מדידות עוקבות ההתנהגות ליניארית.

לצורך כיול בהגעה לסביבת עבודה חדשה של המערכת (ביחס לתאורת הסביבה) יש צורך בביצוע 10 מדידות לאורך טווח של 50cm, ניתן להכין פס נייר באורך מתאים ולסמן מידות בכל 5cm. את פס הנייר נניח על השולחן בקו ישר מנגד ה-LDR. את הפנס (יש לכם כזה במכשיר הנייד) נניח לאורך הסימונים על הסרגל ובכל סימון נלחץ על לחצן לבחירתכם לביצוע דגימת מתח במוצא LDR ולשמירת הערכים המתאימים במערך המיועד לכך. בלחצן נשתמש להתחלה ולסיום תהליך המדידות. בבחירת תפריט כיול חיישני ה-LDR נדרש להשתמש בלחצן PB0 לצורך ביצוע מדידות כיול.

#### 5. חיבורי חומרה והקצאת רגלי הבקר (ערכת פיתוח אישית):

- קישור המכיל סרטון וקובץ בינארי לצריבה ל MCU לבדיקת תקינות מנוע Servo וחיישן אולטראסוניק.  
[Video+binary.d43 - Servo Ultrasonic](#)
- רגליים P1.1, P1.2 – אינן בשימוש, תפוסות לתקשורת טורית מול המחשב (ראה הגדרת ניסוי LAB4).
- מנוע Servo – חיבור לרגל מוצא Timer לבחירתכם (PWM).
- חיישן מרחק – חיבור שתי רגליים (Echo, Trig) למוצא וכניסה של Timer לבחירתכם.
- שני חיישני LDR מחוברים לשתי רגליים של הבקר בעלות כניסה אנלוגית לבחירתכם.
- מסך LCD נדרש לחבר את D7-D4 קווי מידע (אופן עבודה של ה-LCD בארבעה קווי מידע) + שלושת קווי הבקרה של ה-LCD לבחירתכם
- חיבור שני לחצנים PB0, PB1: לצורך כיול חיישני ה-LDR לאחר בחירת תכונה זו בממשק למשתמש בצד ה-PC ובנוסף לצורך תמיכה בקריאת תוכן קבצים בצד הבקר.
- סה"כ חיבור 16 רגליים של הבקר לממשק החומרה הנדרש בפרויקט.

## C. דו"ח מכין: (משקל 10%)

1. כתיבת דו"ח מכין פרויקט מסכם, לפי הוראות לכתיבה ועריכת דו"ח מכין הנמצא במודל.
2. צורת הגשה:
  - הגשת דוח מכין תיעשה ע"י העלאה למודל של תיקיית zip מהצורה **id1\_id2.zip** (כאשר  $id1 < id2$ ), רק הסטודנט עם הת"ז id1 מעלה את הקבצים למודל.
  - התיקייה תכיל את שלושת הפרטים הבאים בלבד:
    - ✓ קובץ **pre\_finalx.pdf** – מכיל תשובות לחלק תיאורטי דו"ח מכין
    - ✓ תיקייה בשם **CCS** - מכילה שתי תיקיות, אחת של **קובצי source** (קבצים עם סיומת \*.c) והשנייה של **קובצי header** (קבצים עם סיומת \*.h).
    - ✓ תיקייה בשם **PC\_side** - המכילה קובצי מקור של אפליקציית צד מחשב + קובץ ReadMe המתאר בקצרה מה תפקיד כל קובץ מקור במימוש האפליקציה.

## D. מבנה הציון בפרויקט:

1. משקל הפרויקט הוא 70% מהציון הסופי - חלק ביצוע (תמיכה בכל סעיפי התפריט) משקלו 90% וחלק תיעוד הפרויקט 10%
2. הציון יינתן על-פי הערכה המבוססת על קריטריונים של עמידה ודיוק בדרישות הפרויקט, בקיאות בקוד + אלגוריתם + תיאוריה, דו"ח מכין והגנה על הפרויקט. המשמעות, כל סטודנט בנפרד יידרש לגלות הבנה מעמיקה במרכיבי הפרויקט (תיאוריה, חומרה, תוכנה ואלגוריתמיקה וכו').
3. כל קבוצה תגיש דו"ח תיעוד הפרויקט לפי קובץ "הוראות לכתיבת תיעוד הפרויקט" המופיע במודל.

בהצלחה!