



## **דו"ח תיעוד פרויקט מסכם לקורס** **"מבנה מחשבים ספרתיים" 361-1-4191**

מערכת לגילוי מקורות אור וניטור אובייקטים במרחב

חברי הקבוצה: עילי ברוך  
גיא מיוסט

שם המדריך האחראי: חנן ריבוא

תאריך הגשה: 01.09.2025



## תוכן עניינים

- הגדרת ומטרת הפרויקט
- תיאור הפרויקט
- תיאור ביצועי החומרה והתוכנה
- תיאור המצבי העבודה השונים :  
(File mode, Telemeter, Light sources detector, objects detector system)
- מכונת מצבים צד MCU
- תיאור חיבורי חומרת קצה
- מסקנות והצעות לשיפורים



## הגדרת ומטרת הפרויקט

בפרויקט הגמר בקורס מבנה מחשבים ספרתיים נאחד את הידע שרכשנו במהלך הסמסטר בהרצאות ובמעבדות לידי תכנון ומימוש מערכת מבוססת MCU לגילוי מקורות אור וניטור אובייקטים במרחב באמצעות מד מרחק UltraSonic, חיישני אור LDR ומנוע Servo. במסגרת הפרויקט פיתחנו קוד בשפת C למימוש מערכת Embedded מבוססת גרעין הפעלה FSM וכתבנו את המערכת במתודולוגיה של שכבות אבסטרקציה למימוש מערכת Embedded מרובת חיישנים. מימוש צד משתמש, בצד PC הוא על ידי שימוש בממשק GUI כך שה-MCU מחובר ל-PC באמצעות תקשורת טורית אסינכרונית בסטנדרט RS-232.

קוד המערכת פותח בשפת C ויממש מכונת מצבים מבוססת פסיקות לתפעול הרכיבים ושליחה וקבלה בערוץ התקשורת. מכונת המצבים תפורט בהמשך בהרחבה.

קוד צד המחשב יפותח בשפת python ויצגי ממשק משתמש (gui) ממנו המשתמש יוכל לתפעל כל פעולה המוגדרת במערכת ודורשת תצוגה וממשק למשתמש. כמו כן, הממשק יאשר העברת קבצים הכוללים פקודות High-level מקודדות למימוש בצד הבקר ויבדקו את חלקי המערכת. הקבצים בצד הבקר ישמרו בזיכרון flash ובחירת הקוד להרצה יעשה דרך ממשק gui.

## תיאור הפרויקט

המערכת מחולקת ל-4 פונקציות עיקריות:

1. ניתור אובייקטים במרחב תוך ביצוע סריקה אחת בלבד בהיקף של 180 מעלות ורמת דיוק אופטימלית. בסוף הסריקה הוצגו אובייקטים על גבי המסך עם פרמטרים של מרחק וזווית מנקודת הסריקה וגם רוחב אובייקט.
2. מיקום חיישן האולטרה סוניק בזווית שנבחרה על ידי המשתמש בממשק ה-GUI והצגה על גבי מסך ה-PC את המרחק הנמדד מחיישן המרחק בזמן אמת באופן דינאמי.
3. ניתור מקורות אור במרחב בטווח של עד חצי מטר בביצוע סריקה בהיקף של 180 מעלות ורמת דיוק אופטימלית.
4. שליחה מ-PC של עד עשרה קבצים עם גודל משתנה ושמירתם בזיכרון flash. הקבצים הם מסוג text או script. קבצי text יוצגו על גבי מסך LCD של MCU וקבצי script הם קבצים שיכילו פקודות High-level.



## תיאור ביצועי החומרה והתוכנה

### זיהוי אובייקטים במרחב בעזרת חיישן ultrasonic

במשימה זו השתמשנו בחיישן ultrasonic ובמנוע servo לצורך זיהוי אובייקטים במרחב ממרחק 2 ס"מ ועד 450 ס"מ. בצד PC, דרך GUI המשתמש יוזם סריקה. מנוע הסרבו עובד על אות PWM. ערך ה duty cycle קובע את מיקום הזרוע. אצלנו לאחר כיוול זוויות קצה, קיבלנו ערך מינימום 450ms, עבור זווית 0 מעלות, וערך מקסימום 2175ms, עבור זווית 180 מעלות.

על גבי זרוע מנוע הסרבו קיים חיישן ultrasonic שמקבל דרך רגל trigger פולס ברוחב של לפחות 10usec עם דיליי של לפחות 60msec כלומר תדר עבודה של מקסימום 16.7Hz. בסיום הפולס חיישן המרחק שולח גל קול באורך שמונה מחזורים בתדר 40kHz לכיוון האובייקט וקולט את ההחזרים המגיעים ממנו. המעגל החשמלי הנמצא בחיישן ממיר את החזרי גל הקול לפולס היוצא מרגל Echo. הפולס שיוצא הוא אורך הזמן שעבר מרגע שידור גל הקול ועד לקבלת החזרים מהאובייקט הנמצא מול החיישן. הפולס היוצא מרגל Echo נכנסת לרגל P2.4 עם קונפיגורציה של input capture של טיימר A1. דרך input capture אנחנו מודדים את הרוחב הפולס שמגיע דרך רגל Echo.

צד PC שולח את ערך threshold שנקבע על ידי המשתמש. בכל זווית אנו דוגמים 3 פעמים ולוקחים את הערך החציוני. ערך זה אנחנו שולחים דרך UART בתקשורת טורית לצד המחשב. המחשב בכל פעם מקבל מידע בגודל 2 bytes דרך UART ומקדם את הזווית במעלה נוספת. כך אנחנו חוסכים בהפסק ומגדילים יעילות (נשלחים שני בתים במקום שלושה). כל דגימה שמגיעה נשמרת בטבלה. האלגוריתם תוכנן לקטלג דגימות עם זוויות עוקבות ומרחק דומה לכדי אובייקט אחד. כאשר האלגוריתם כבר לא מזהה את הדגמות כשייכות לאותו אובייקט, האלגוריתם מחשב את זווית האובייקט כממוצע הזוויות שנמדדו. כלומר לוקחים את ממוצע כל הזוויות בהן הופיע האובייקט.

$$\varphi = \text{round}\left(\frac{\sum_i \text{deg}_i}{N}\right)$$

חישוב המרחק מתבצע על ידי אנליזה של אותו רצף דגימות. לוקחים את כל המרחקים בהם נמצא האובייקט ומחשבים ממוצע בין כל המרחקים.

$$\rho = \text{round}\left(\frac{\sum_i \text{distance}_i}{N}\right)$$

בפועל המרחק נגזר מזמן ההד באולטרסוניק כיוון שאנחנו יודעים שגל הקול יצא מ-TX התפשט עד האובייקט וחזר. עד שגל ההחזר הגיע ל-RX הוא עבר 2dists ולכן:

$$\text{dists} = \frac{vt}{2}$$

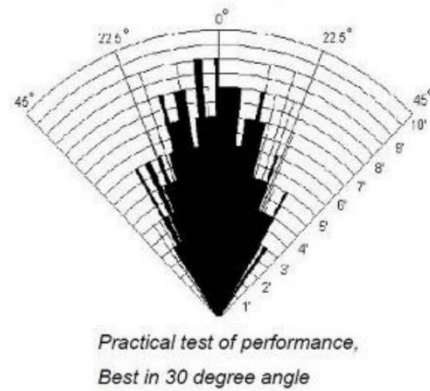
חישוב אורך האובייקט מתבצע עבור אותו רצף דגימות תוך שימוש בחישוב לפי משפט הקוסינוסים:

$$l = \sqrt{r_1^2 + r_2^2 - 2\cos(\Delta\theta)}$$

כאשר  $r_1, r_2$  הם מרחקי קצוות הרצף,  $\Delta\theta$  זה הזווית מפתח בין שני קצוות הרצף.



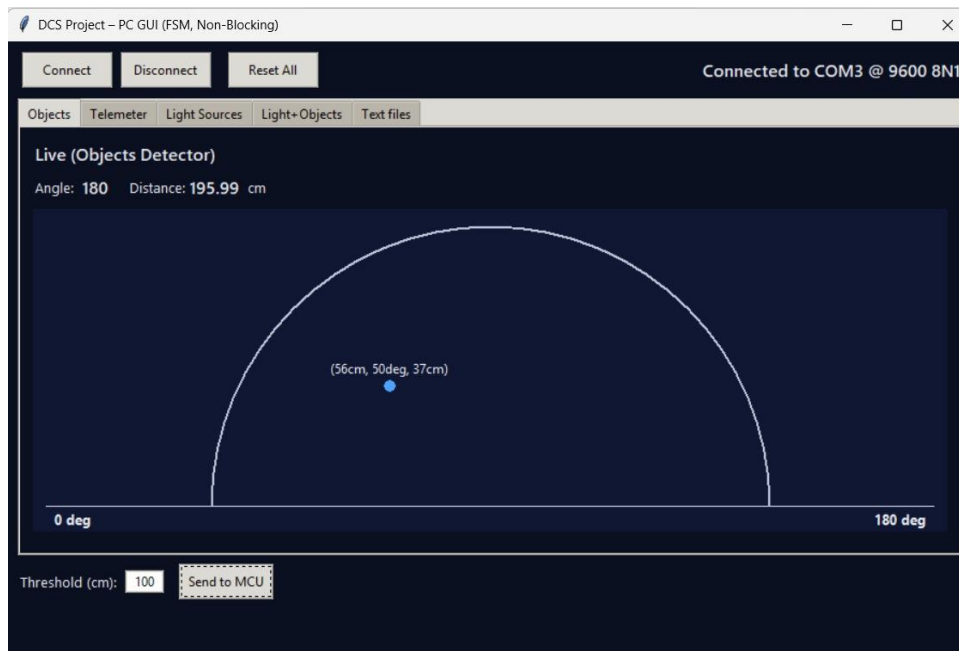
לאחר מכן אנחנו עושים תיקון בשל התנהגות מפתח הסאונד המוחזר כמו בתמונה :



לכן קיבלנו :

$$l = \sqrt{r_1^2 + r_2^2 - 2\cos(\Delta\theta)} - \tan(15) * (r_1 + r_2)$$

עבור כל אובייקט שנמצא רחוק יותר מהthreshold צד הPC לא שומר ערכים ולכן הדבר חוסך בסיבוכיות מקום. בסוף הסריקה מופיע על מסך הPC מפת הראדר ועליה מצויינים ערכי המרחק, הזווית ורוחב האובייקט.





## Telemeter – מדידת מרחק האובייקט בזווית קבועה באופן דינאמי

במצב זה המשתמש מגדיר זווית סריקה סטטית. חיישן האולטרסוניק עובד באותה דרך המוסברת בהרחבה במצב Object Detector. רגל Echon נדגמת בעזרת טיימר A1 ב input capture. כל דגימה נשלחת ל-PC דרך UART וזאת מוצגת על גבי המסך GUI של המשתמש.



## כיוול גלאי אור

כיוול גלאי אור הוא שלב חשוב ביותר לפני גילוי מקורות האור. גלאי האור משמש כנגד משתנה ולכן כל שינוי של אור במרחב החיישן יכול לשנות את התנגדות החיישן. הכיוול מתבצע בעזרת מטר על מנת להעלות את הדיוק. על המשתמש להכנס בממשק GUI למצב Light Source Detector וללחוץ על כפתור LED Calibration. המשתמש צריך לדגום 10 דגימות מ 5 ס"מ ועד 50 ס"מ בעזרת לחיצה על כפתור PB0. בסיום תהליך הדגימה, הבקר שומר את הדגימות בסגמנט D על גבי ה-Flash בעזרת struct חדש שהגדרנו. בצד הבקר הגדרנו struct חדש בשם LDRFile המכיל את המשתנים הבאים:

struct LDRFile	
ldr1[10]	מערך שמכיל את דגימות ldr1
ldr2[10]	מערך שמכיל את דגימות ldr2
name[10]	שם
size	גודל המידע המוכל בstruct
type[10]	סוג המידע

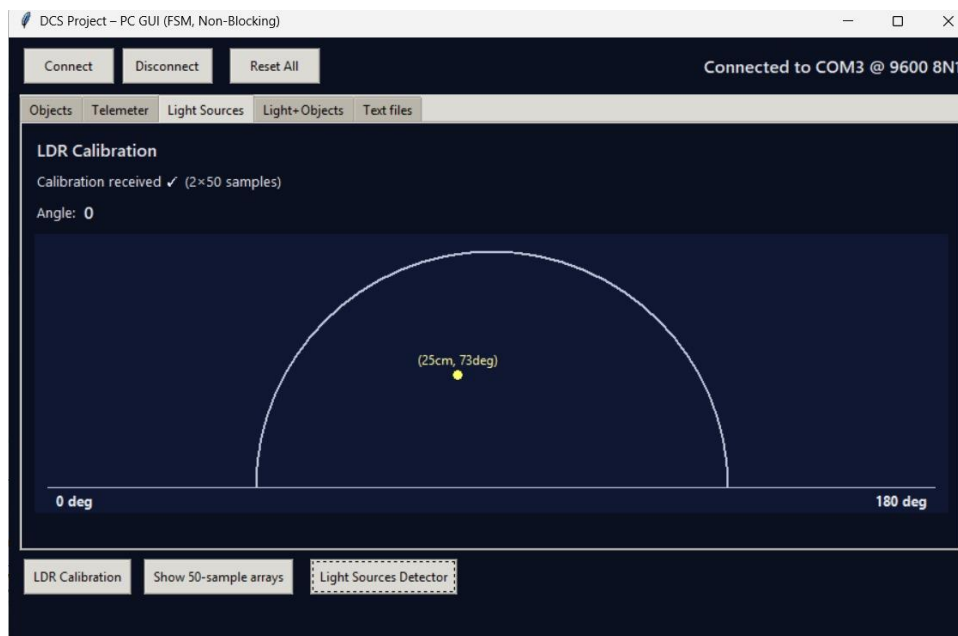


לאחר שמירת ה struct בסגמנט D, נשלחות הדגימות דרך UART לצד PC. צד PC מקבל את הדגימות ומייצר Lookup Table על ידי אינטרפולציה לינארית ומציג למשתמש את הגרף.



## זיהוי מקורות אור במרחב

בשלב זה על ידי לחיצה על כפתור Light Source Detector מתחילה סריקה של 180 מעלות. שני חיישני LDR נדגמים בעזרת ADC10 כל זווית 3 דגימות. בצד MCU מתבצע חישוב חציון עבור ערכי כל חיישן ולאחר מכן מחשבים ממוצע בין שני החציונים. הערך הממוצע נשלח לצד PC דרך UART שאוגר את כל הנתונים בטבלה. זווית הגלאי אינה נשלחת MCU, ובכך חוסכים בהספק המערכת, אלא צד PC מקדם עבור כל דגימה שנשלחת (2 בתים בלבד) את הזווית במעלה אחת. צד PC מקטלג כל רצף של דגימות ברוחב גדול 4 מעלות עם ערך דומה למקור אור פוטנציאלי. צד PC מחשב את הנקודות המינימום בדגימות שקיבל. התוכנית מחשבת לפי LUT את המרחק מהאובייקט. לבסוף מוצג על גבי מסך GUI מקורות האור שהתגלו.





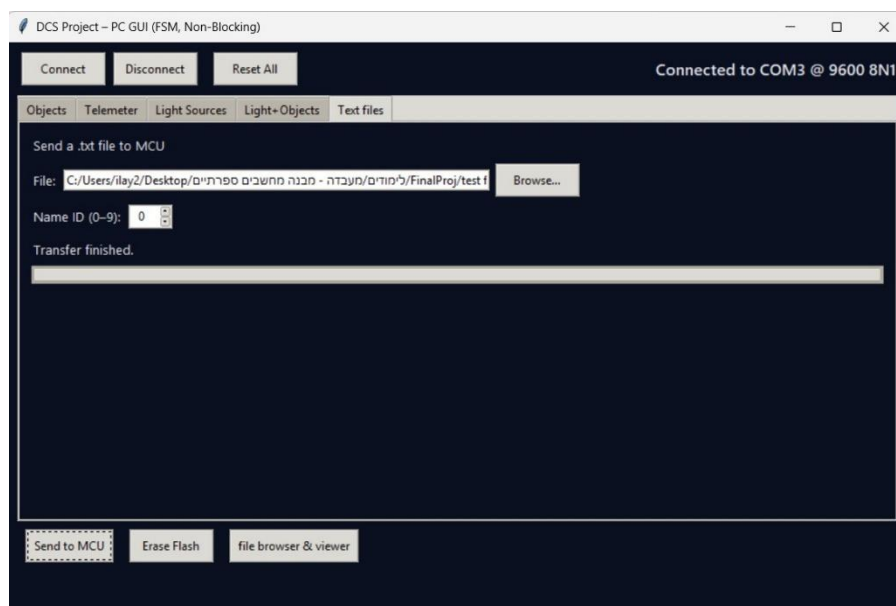
## טעינת טקסט והצגתו על גבי מסך LCD

במצב 4 המשתמש יוכל לטעון עד 10 קבצי text לא דחוסים, ולהציג אותם על גבי מסך LCD בצד MCU. סך קבצי הטקסט והscript הוא 10. בממשק GUI אפשר לבחור קובץ מהPC ולטעון אותו לMCU דרך UART. בצד הבקר הגדרנו struct חדש בשם FileEntry המכיל את המשתנים הבאים:

struct FileEntry	
name	שם הקובץ הוא מספר מ0 עד 9
type	סוג הקובץ – 0 קובץ טקסט, 1 קובץ סקריפט
Size_bytes	גודל המידע ביחידות של bytes
Start_addr	כתובת התחלתית בה שמור המידע בזכרון flash

MCU מקבל תחילה את חלק ה header של הקובץ. ה header של הקובץ מכיל שם, סוג וגודל הקובץ ביחידות של bytes. שם הקובץ הוא מספר מ0 עד 9, זאת על מנת לחסוך בזמן ובמקום, בשילוב עם ידיעה של הסוג הקובץ כלומר type נוכל לשחזר את השם המלא של הקובץ. לדוגמה – type=0; name=0; נוכל לשחזר לשם text0.txt. גודל הקובץ חשוב מאוד לתהליכים רבים בניהול הזיכרון. לאחר קבלת הheader, הMCU שולח את האות 'A', שהיא אומרת שהMCU קיבל את הheader ללא שגיאות וכי יש מספיק מקום בזיכרון על מנת לקבל את המידע ולאכסנו. במידה ונשלח האות 'F' זה אומר כי זיכרון flash מלא וכי אין אפשרות לקבל את הקובץ, כאשר נשלח האות 'E' זה אומר כי קיימת שגיאה.

בעת קבלת האות 'A' הבקר יכול לשמור את הstruct שמכיל מידע על הקובץ text. השמירה של כל הstructs היא בסגמנט C, בעוד שהמידע עצמו נשמר בסגמנטים 4 עד 1. התוכנית מגיעה עד לכדי מספר סגמנטים מכתובת הdata החדש, לכן הכתיבה לא דורסת ערכים קיימים. הPC מתחיל להעביר byte אחרי byte אל הMCU, כאשר כל byte שמתקבל נכתב ישירות לתוך הflash. כאשר כל המידע עובד בהצלחה הMCU שולח לצד הPC את האות 'K' ובצד הGUI מופיע All Data Trasmited. עד כה היה שלב הטעינה ושמירה בflash.





כאשר אנחנו לא שומרים את קובץ הטקסט אנחנו יכולים להשתמש בPB0 על מנת לדפדף בכל קבצי הטקסט הקיימים במערכת. שימוש בPB1 מאפשר לנו לקרוא קובץ ספציפי. בכל דף ניתן להציג עד 32 אותיות וזאת בגלל מגבלה של גודל הLCD. יציאה חזרה לתפריט תהיה על ידי לחיצה נוספת על PB1.

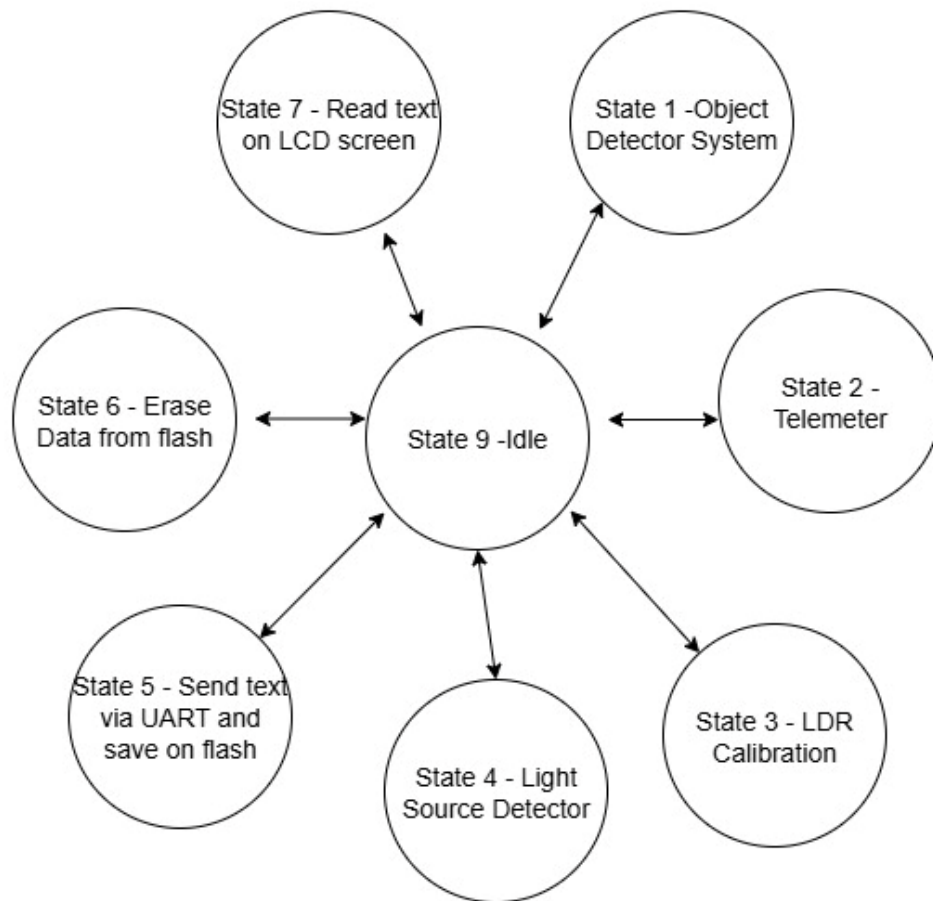
## הרצת קוד על הבקר

בתכנון מצב זה המשתמש יוכל לטעון עד 10 קבצים מהמחשב, ולהריץ אותם על הבקר. סך קבצי הטקסט והscript הוא 10. לחיצה על כפתור submit שולחת כל קובץ שנבחר לבקר על ידי תקשורת טורית ונשמרת בזיכרון הFLASH של הבקר. לאחר מכן המשתמש יוכל לבחור להריץ את אחד הסקריפטים שטען.

בצד הבקר הגדרנו struct בשם FileEntry, אותו struct שהשתמשנו עבור הtextn. המשתנים מתעדכנים בעת הטעינה של הסקריפטים וכתיבתם לזיכרון הflash, ומשמשים את הבקר בעת קריאה מהזיכרון והרצה של הסקריפט.



## FSM - מכונת מצבים צד MCU



Every state can move to the other and not have to go through state 9



## פירוט מצבים לפי פונקציות עיקריות

### זיהוי אובייקטים במרחב בעזרת חיישן ultrasonic

בחירת פונקציה ראשונה מממשק המשתמש ← הגעה למצב state1 ← תחילת תנועת מנוע Servo ותחילת דגימה של חיישן ה-Ultrasonic על ידי Input capture mode. בסיום המצב הצגה של מפת אובייקטים שזוהו על גבי המפה. מצב זה הוא מצב אטומי וניתן לעבור לכל מצב רק בסיום הסריקה.

### Telemeter

בחירת פונקציה שנייה מממשק המשתמש ← הגעה למצב state2 ← דגימת חיישן Ultrasonic והצגה של המרחק האובייקט באופן דינמי על מסך GUI. ניתן לעבור ממצב זה לכל מצב אחר בכל עת.

### כיוול חיישני LDR

בחירת פונקציה שלישית מממשק המשתמש ← הגעה למצב state3. לחיצה על הכפתור PB0 תדגום בעזרת ADC10 את ערכי חיישני ה-LDR. נדגום מ-5 ס"מ ועד 50 ס"מ 10 דגימות. בסיום הכיוול יופיע גרף האינטרפולציה של ערכי החיישנים. מצב זה אינו אטומי וניתן לעבור ממצב זה לכל מצב אחר בכל עת.

### Light Source Detector

בחירת פונקציה רביעית מממשק המשתמש ← הגעה למצב state4. במצב זה נבצע סריקה של חיישני ה-LDR, אשר ידגמו בכל זווית 3 פעמים כל אחד בסיום המצב יופע מפת זיהוי מקורות אור בממשק המשתמש. מצב זה הוא מצב אטומי וניתן לעבור לכל מצב רק בסיום הסריקה.

### שמירת קובץ text בזיכרון Flash

בחירת פונקציה חמישית מממשק המשתמש ← הגעה למצב state5. נפתח מסך המאפשר לטעון עד 10 קבצים מהמחשב. לחיצה על כפתור submit שולחת את קבצי הקוד לבקר ומתחילה את תהליך השמירה בflash. צד בקר מנהל את הזיכרון בעזרת data struct שהגדרנו שנשמרים בסגמנט C בעוד שהמידע עצמו נשמר בסגמנטים 1-4 (טווח כתובות: 0xFE00 – 0xF600). קובץ גדול מ-2KB אינו נשלח ונעצר בצד PC. ראשית ה-PC שולח את שם הקובץ, סוג הקובץ וגודל הקובץ. במידה ואין מקום מספיק כדי לשמור את הקובץ נשלחת הודעת שגיאה. במידה ויש מקום, צד בקר שולח 'A' ל-PC ואז ה-PC מתחיל להזרים את הקובץ עצמו. בסיום השמירה נקבל ack.

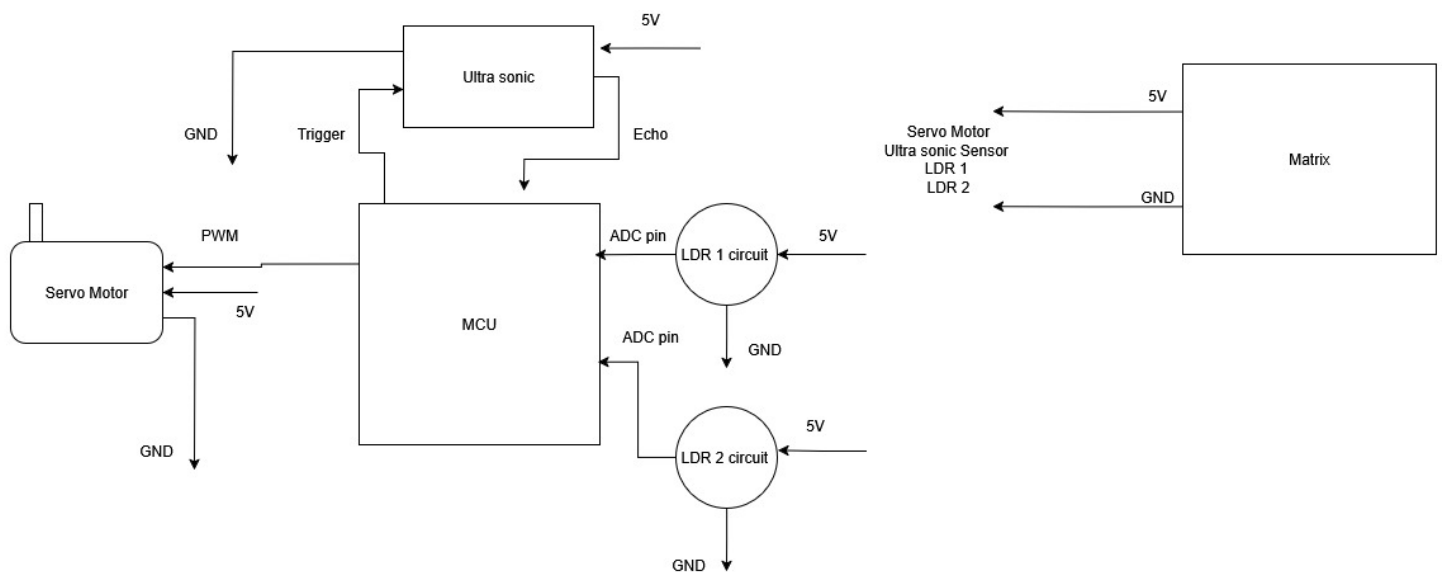
### מחיקת קובץ text בזיכרון Flash

בחירת פונקציה שישית מממשק המשתמש ← הגעה למצב state6. ניתן ללחוץ על כפתור Erase Flash על מנת למחוק את כל ה-Flash. בסוף התהליך נקבל ack.

## קריאת קבצי text על גבי מסך LCD

בחירת פונקציה שבעית מממשק המשתמש ← הגעה למצב state7. בעת לחיצה על כפתור PB0 ניתן לדפדף על שמות הקבצים הקיימים בזכרון. לחיצה על PB1 קריאת text על גבי מסך LCD ששמו מופיע בשורה הראשונה. לחיצה נוספת על PB0 דפדוף נוסף על מנת להמשיך לקרוא את הtext. ולחיצה על PB1 חזרה לתפריט שמות הקבצים.

## תיאור חיבורי חומרת קצה





## מסקנות והצעות לשיפורים

- חשבנו הרבה על תכנון המערכת ועל זמן העבודה למול יעילות הקוד והחומרה. למדנו המון על תכנון מערכות ובניית מערכת סגורה עם ממשק למשתמש.
- חיישני LDR מתנהגים שונה בין חיישן לחיישן, חיישנים שונים בעלי רגישות שונה. אנו מציעים לתת את דגם החיישן הספציפי על מנת לאפשר לנו למצוא את הdata sheets של היצרן ולבדוק האם אנחנו מקבלים תוצאות דומות לתכנון היצרן.
- בהינתן יותר זמן היינו רוצים לדייק יותר את המערכת על מנת להפוך אותה לטובה הרבה יותר ואיכותית יותר.

