



דו"ח תיעוד פרויקט מסכם לקורס

"מבנה מחשבים ספרטטיים" 361-1-4191

מערכת לגילוי מקורות אור וניטור אובייקטים למרחב



חברי הקבוצה :
יעלי ברוך
גיא מיויסט

שם המדריך האחראי : חנן ריבוא

תאריך הגשה : 01.09.2025



תוכן עניינים

- הגדרת ומטרת הפרויקט
- תיאור הפרויקט
- תיאור ביצועי החומרה והתוכנה
- תיאור המרכיבים העיקריים :
(File mode, Telemeter, Light sources detector, objects detector system)
- מכונת מצלבים צד MCU
- תיאור חיבורו כומרת קצה
- מסקנות והצעות לשיפורים



הגדרת ומטרת הפרויקט

בפרויקט הגמר בקורס מבנה מחשבים ספראטיים נאחד את הידע שרכשנו במהלך הסמסטר בהרצאות ובמעבדות לידי תכנון וIMPLEMENTATION מערכות מבוססות MCU לגילוי מקורות אור וניתור אובייקטים למרחב C באמצעות מודול מרחוק **UltraSonic**, חיישני אור LDR ומנוע Servo. במסגרת הפרויקט פיתחנו קוד בשפת C למימוש מערכת **Embedded** מבוססת גרעין הפעלה FSM וכתבנו את המערכת בתודולוגיה של שכבות ABSTRACTIZING למימוש מערכת Embedded מרובת חיישנים. שימוש צד משתמש, בצד PC הוא על ידי שימוש במסך GUI כזשהMCU מחובר ל-PC באמצעות תקשורת טורית אסינכרונית בסטנדרט RS-232.

קוד המערכת פותח בשפת C וימש מכונת מצבים מבוססת פסיקות לתפעול הרכיבים ושליחה וקבלת עזרץ התקשורת. מכונת המצבים תפורט בהמשך בהרחבה.

קוד צד המחשב יפותח בשפת **python** ויציג ממשק משתמש (**gui**) ממנו המשמש בלבד לתפעול כל פעולה המוגדרת במערכת ודורשת תצוגה ומשחק למשתמש. כמו כן, המשחק יאשר העברת קבצים הכוללים פקודות High-level מוקדמות למימוש בצד הבקר ויבדקו את חלקי המערכת. הקבצים בצד הבקר ישמרו בזיכרון **flash** ובבחירה הקוד להרצתה יעשה דרך ממשק **gui**.

תיאור הפרויקט

המערכת מחולקת ל-4 פונקציות עיקריות:

1. ניתור אובייקטים למרחב תוך ביצוע סריקה אחת בלבד בהיקף של 180 מעלות ורמת דיווק אופטימלית. בסוף הסריקה הוצגו אובייקטים על גבי המסך עם פרמטרים של מרחוק וזווית מוקדות הסריקה וגם רוחב אובייקט.
2. מיקום חיישן האולטרה סאוניק בזווית שנבחרה על ידי המשתמש בממשק GUI והציג על גבי מסך PC את המרחק הנמדד מחיישן המרחק בזמן אמת באופן דינامي.
3. ניתור מקורות אור למרחב בטוח של עד חצי מטר ביצוע סריקה בהיקף 180 מעלות ורמת דיווק אופטימלית.
4. שליחה מסך PC של עד עשרה קבצים עם גודל משתנה ושמירותם 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 למרחב בעזרת חיישןasonic

במשימה זו השתמשנו בחישןasonic servo ובמנוע servo לצורך זיהוי אובייקטים למרחב מרחק 2 ס"מ ועד 450 ס"מ. בצד PC, דרך GUI המשמש יוזם סריקה. מנוע הסרבו עובד על אות PWM. ערך ה duty cycle קובע את מיקום הזרוע. אצלנו לאחר כיוול זווית כזה, קיבלנו ערך מינימום 450ms , עברו זווית 0 מעלות , וערך מקסימום 2175ms, עברו זווית 180 מעלות.

על גבי זרוע מנוע הסרבו קיים חיישןasonic שמקבל דרך trigger פולס ברוחב של לפחות 10usec עם דילוי של לפחות 60usec לולר תדר עבורה של מקסימום 16.7Hz. בסיום הפולס חיישן sonic שלח גל קול באורך שווה מחזורי בתדר 40kHz לכיוון האובייקט וקולט את החזרים המגיעים ממנו. המגל החשמלי הנמצא בחישן 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 החישן. הפולס היוצא מרגל Echo נכנסת לרגל P2.4 עם קונפיגורציה של input capture של טימר 1. דרך Echo נקלט מודדים את הרוחב הפולס שגיע דרך רגל Echo.

צד PC שלח את ערך threshold שנקבע על ידי המשתמש. בכל זווית אנו דוגמים 3 פעמים ולוקחים את הערך החיצוני. ערך זה אנחנו שלחimos דרך UART בתקשורת טורית לצד המחשב. המחשב בכל פעם מקבל מידע בגודל bytes 2 דרך UART ומקדם את הזווית במעלה נוספת. כך אנחנו חוסכים בהפסק וגדילים יעילות (שלחים שני בתים במקום שלושה). כל דגימה שמייה נשמרת בטבלה. האלגוריתם תוכנן לקטול דגימות עם זווית עוקבות ומרחק דומה לכדי אובייקט אחד. כאשר האלגוריתם כבר לא מזזה את הדגימות כשייכות לאותו אובייקט, האלגוריתם מחשב את זווית האובייקט כממוצע הזווית שנמדדזו. ככלMORE לוקחים את ממוצע כל הזווית ביחס להיפע האובייקט.

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

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

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

בפועל המרחק נוצר מזמן ההד באולטרסיניק כיוון שאנו יודעים שהקול יצא מTA התפשט עד האובייקט וחזור. עד שהקהל הגיע לRX הוא עבר 2dists וכאן :

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

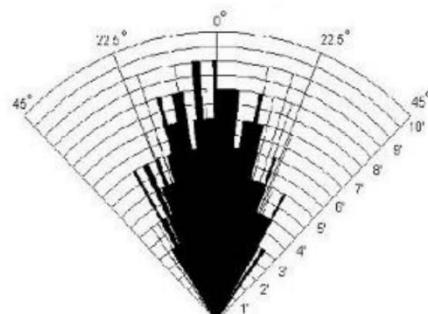
чисוב אורך האובייקט מתבצע עבור אותו רצף דגימות תוך שימוש בחישוב לפי משפט הקוסינוסים :

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

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



לאחר מכון אנחנו עושים תיקון בשל התנוגות מפתח הסאונד המוחזר כמו בתמונה :

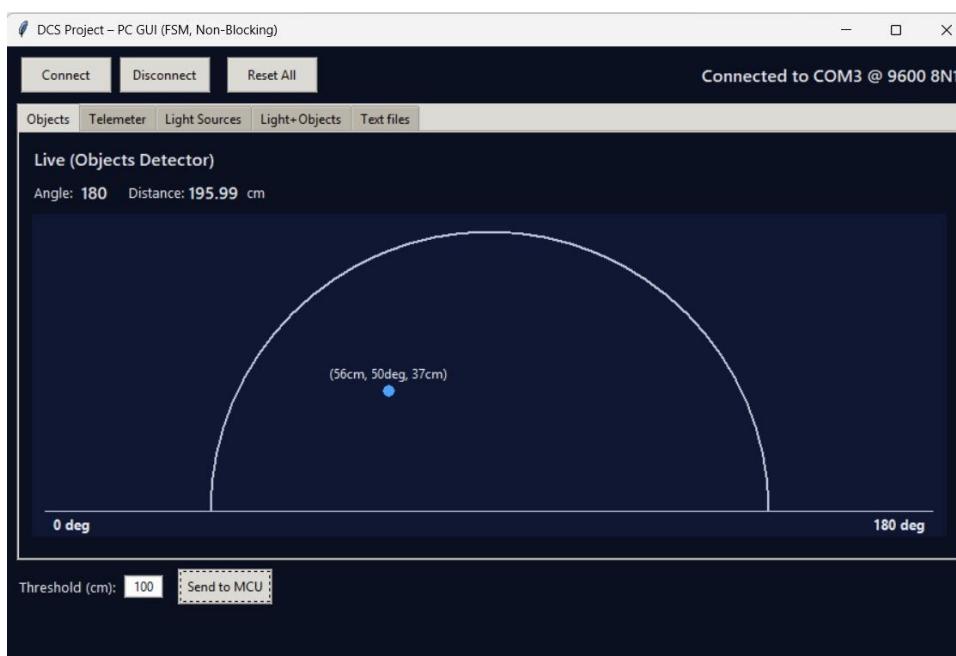


*Practical test of performance,
Best in 30 degree angle*

לכן קיבלנו :

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

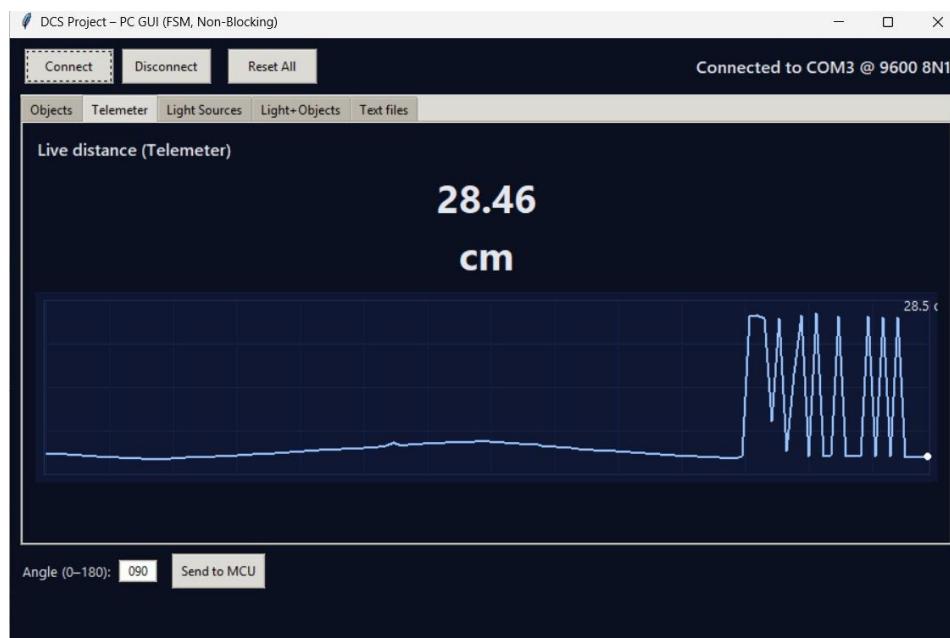
עבור כל אובייקט שנמצא רחוק יותר מהreshold צד PC לא שומר ערכיהם ולכן הדבר חוסך בסיבוכיות מקום. בסוף הבדיקה מופיע על מסך PC מפת הרادر ועליה מצוינים ערכי המרחק, הזוויות ורוחב האובייקט.





מדידת מרחק האובייקט בזווית קבועה באמצעות Telemeter

במצב זה המשמש מגדר זווית סריקה סטטית. חישון האולטרסוניים עובד באותה דרך המוסברת בהרחבה במצב Object Detector. רgel ה-Echo נדגמת בעזרת טימר A1 ב input capture. כל דגימה נשלחת ל-PC דרך UART וזאת מוצגת על גבי המסך GUI של המשתמש.



כיוול גלאי אור

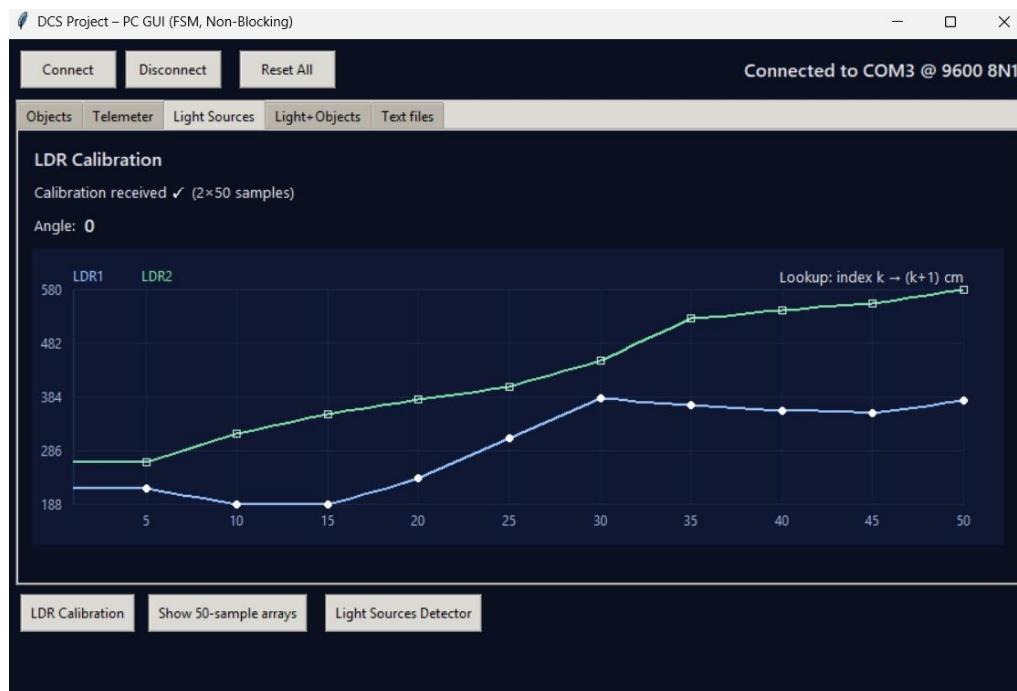
כיוול גלאי אור הוא שלב חשוב ביותר לפני גילוי מקורות האור. גלאי האור משמש כנגד לשנתנה ולכך כל שינוי של אור במרחב החישון יכול לשנות את התנגדות החישון. הכיוול מתבצע בעזרת מטר על מנת להעלות את הדיקוק. על המשמש להכנס במכשיר GUI למצב Light Source Detector למדוקן LED וללחוץ על כפתור Calibration. המשמש צרך לדגום 10 דגימות מ5 ס"מ ועד 50 ס"מ בעזרת לחיצה על כפתור PB0. בסיום תהליך הדגימה, הAKER שומר את הדגימות בסגמנט D על גבי Flash memory החדש שהגדנו.

בצד הAKER החדש struct חדש בשם LDRFile המכיל את המשנים הבאים :

struct LDRFile	
ldr1[10]	מערך שמכיל את דגימות 1
ldr2[10]	מערך שמכיל את דגימות 2
name[10]	שם
size	גודל המידע המוכל ב-
type[10]	סוג המידע

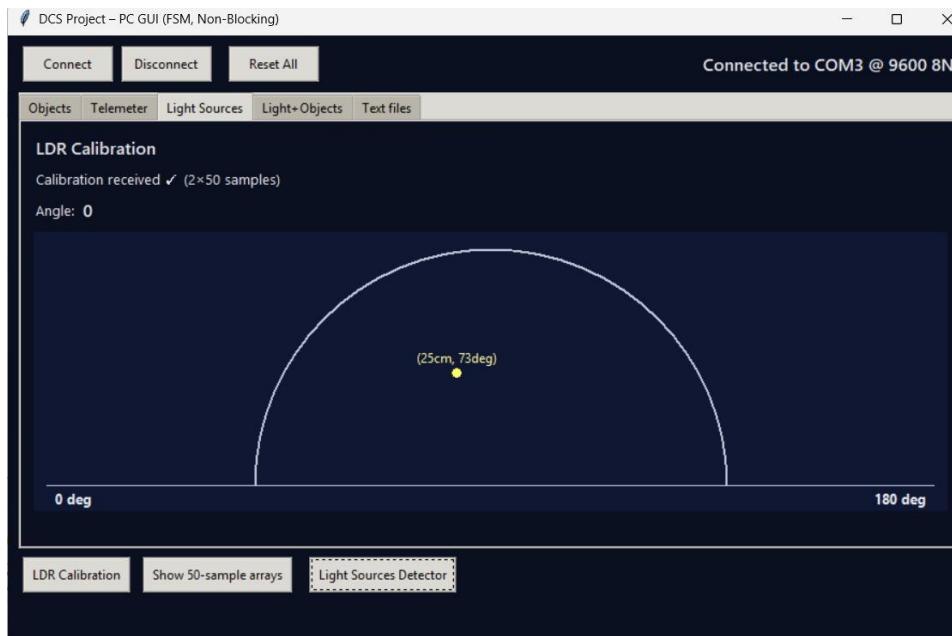


לאחר שמירת ה `struct` בסגמנט D, נשלחות הדגימות דרך UART לצד PC. הצד PC מקבל את הדגימות ומיציר Lookup Table על ידי אינטראפלציה לינארית ומציג למשתמש את הנגרף.



זיהוי מקורות אור במרחב

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





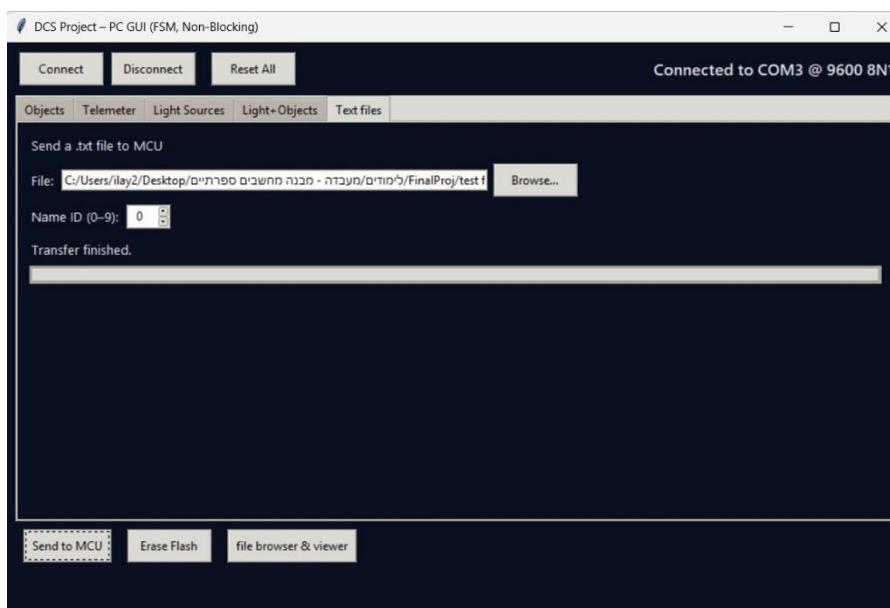
טיעינת טקסט והציגו על גבי מסך LCD

במצב 4 המשמש יכול לטעון עד 10 קבצי text לא דחוסים, ולהציג אותם על גבי מסך LCD בצד MCU. סך קבצי הטקסט והקובץ הוא 10. במכשיר GUI אפשר לבחור קובץ מהPC ולטעון אותו לMCU דרך UART. בצד הבקר הגדכנו חדש בשם struct המכיל את המשתנים הבאים:

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

MCU מקבל תחילה את חלק header של הקובץ. header כולל שם, סוג וגודל הקובץ ביחידות של bytes. שם הקובץ הוא מספר מ-0 עד 9, זאת על מנת לחסוך בזמן ובמקום, בשילוב עם ידיעה של הסוג הקובץ יכול לשחזר את השם המלא של הקובץ. לדוגמה – name=0; type=0 לשחזר לשם text0.txt. גודל הקובץ חשוב מאוד לתהליכים רבים בניהול הזיכרון. לאחר קבלת header, MCU שולח את האות 'A', שהיא אומרת שהMCU קיבל את header 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 לשמר את struct שמכיל מידע על הקובץ text. השמירה של כל structs היא בסגמנט C, בעוד שהמידע עצמו נשמר בסגמנטים 4 עד 1. התוכנית מגיעה עד לכדי מספר סגמנטים מכתבת dataן החדש, אך הכתיבה לא דורשת ערכים קיימים. PC מתחילה להעביר byte אחריו byte אל MCU, כאשר כל byte שמתאפשר נכתב ישירות לתוכן flash. כאשר כל המידע עובד בהצלחה MCU שולח lcd את האות 'K' ובצד GUI מופיע All Data Trasmited. עד כה היה שלב הטיעינה ושמירה בflash.





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

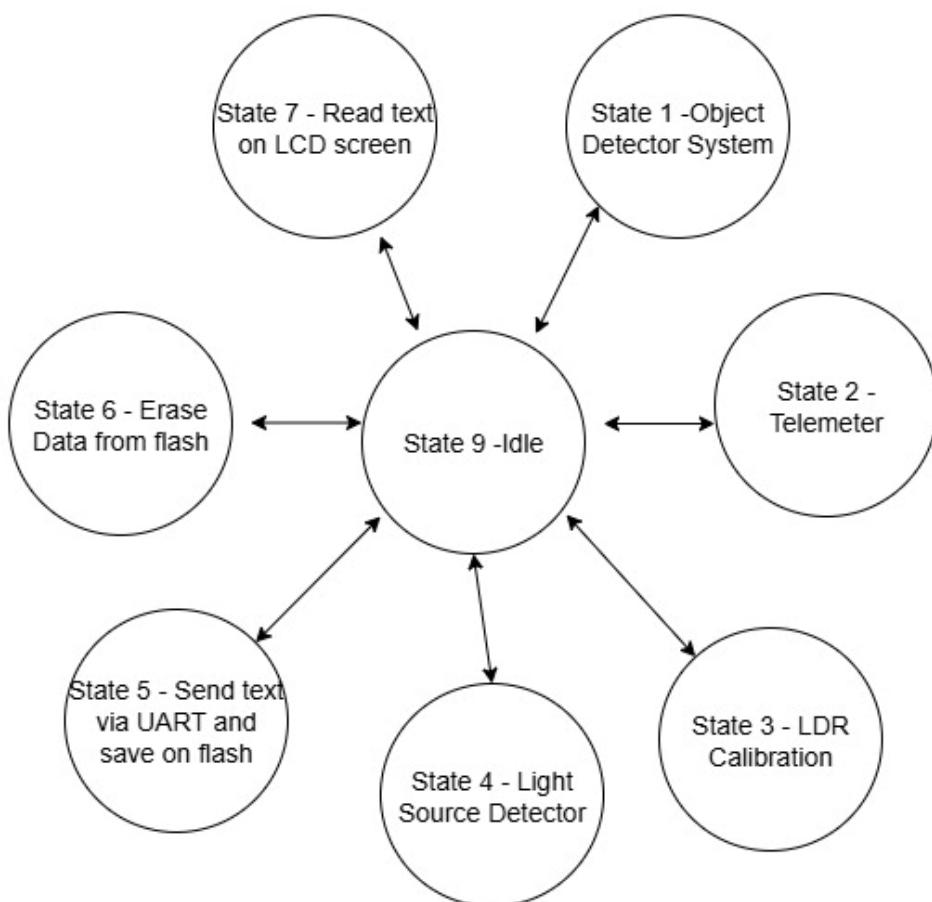
הרצה קוד על הבקר

בתכnuן מצב זה המשמש 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 FLASH של הבקר. לאחר מכן המשמש יוכל לבחור להריץ את אחד הסקריפטים שטוען.

בצד הבקר הגדרנו struct בשם `struct FileEntry`, אותו struct שהשתמשו עבור הtext.
המשתנים מתעדכנים בעת הטעינה של הסקריפטים וכתייתם לזכרון הflash, ומשמשים את הבקר בעת קרייה מהזיכרון והרצה של הסקריפט.



MCU - מכונת מצבים צד FSM



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



פירוט מצלבים לפי פונקציות עיקריות

זיהוי אובייקטים למרחב בעזרת חיישן Ultrasonic

בחירת פונקציה ראשונה ממושך המשמש ← הגעה למצב state1 ← תחילת תנועת מנוע Servo ותחילת דגימה של חיישן Ultrasonic על ידי Input capture mode. בסיום המצב הצגה של מפת אובייקטים שזוהה על ידי המפה. מצב זה הוא מצב אוטומי וניתן לעبور לכל מצב רק בסיום הסריקה.

Telemeter

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

Cיול חיישני LDR

בחירת פונקציה שלישית ממושך המשמש ← הגעה למצב state3. לחיצה על הכפתור PB0 תדגם בעזרת ADC10 את ערכי חיישני LDR. נדgos מ-5 ס"מ ועד 50 ס"מ 10 דגימות. בסיום הقياس יופיע גוף האינטראפלציה של ערכי החישנים. מצב זה אינו אוטומי וניתן לעبور ממצב זה לכל מצב אחר בכל עת.

Light Source Detector

בחירת פונקציה רביעית ממושך המשמש ← הגעה למצב state4. במצב זה נבצע סריקה של חיישני LDR, אשר ידגוו בכל זווית 3 פעמים כל אחד בסיום המצב יופע מפה זיהוי מקורות אור ממושך המשמש. מצב זה הוא מצב אוטומי וניתן לעبور לכל מצב רק בסיום הסריקה.

שמירת קובץ text בזיכרון Flash

בחירת פונקציה חמישית ממושך המשמש ← הגעה למצב state5. נפתח מסך המאפשר לטען עד 10 קבצים מהמחשב. לחיצה על כפתור submit שולחת את קבצי הקוד לבקר ומתיילה את תהליך השמירה בflash. צד בקר מנהל את הזיכרון בעזרת data struct שהגדנו שנשמרם בסגמנט C בעוד שהמידע עצמו נשמר בסגמנטים 4-1 (טוווח כתובות : 0xF600 – 0xFE00). קובץ גדול מ-2KB אינו נשלח ונוצר בצד PC. ראשית PC שולח את שם הקובץ, סוג הקובץ וגודלו הקובץ. במידה ואין מקום מספיק כדי לשמר את הקובץ נשלחת הודעה שגיאה. במידה יש מקום, צד בקר שולח 'A' ל-PC ואז PC מתחילה להזיר את הקובץ עצמו. בסיום השמירה נקלט .ack.

מחיקת קובץ text בזיכרון Flash

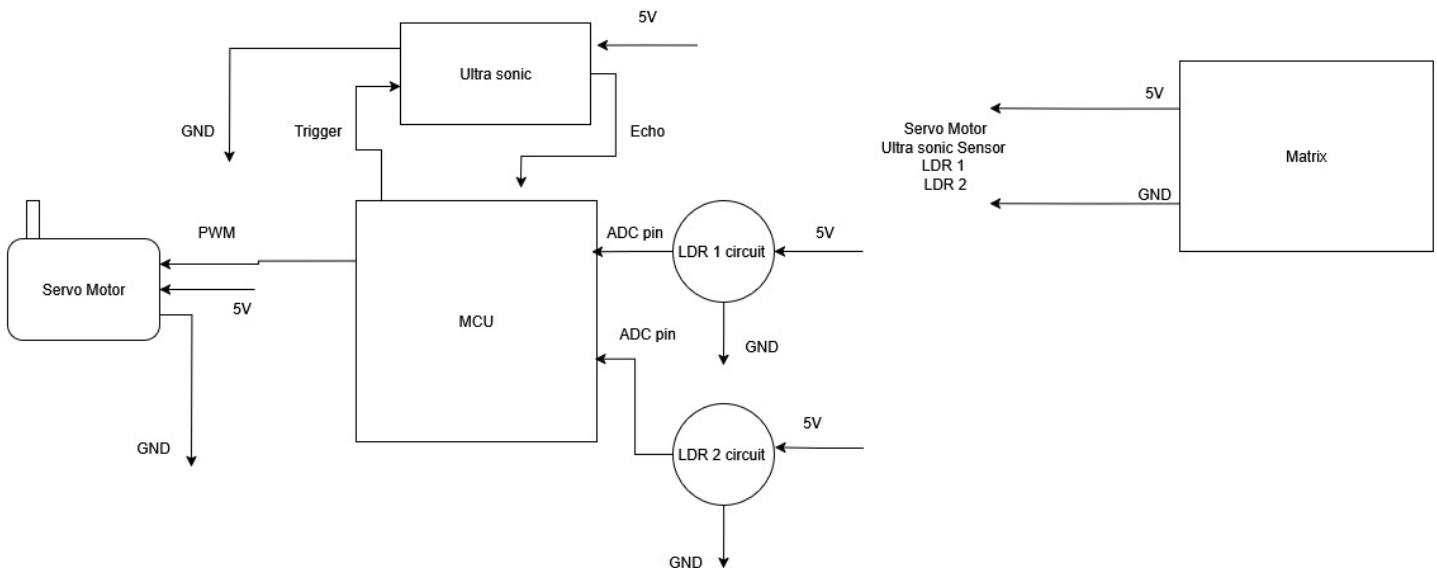
בחירת פונקציה ששית ממושך המשמש ← הגעה למצב state6. ניתן ללחוץ על כפתור Erase Flash על מנת למחוק את כל Flash. בסוף התהליך נקלט .ack.



קריאה קבצי text על גבי מסך LCD

בחירת פונקציה שבująca ממסך המשמש \leftarrow הגעה למצב 7.state. בעת לחיצה על כפתור 0 PB0 ניתן לדפס על שמות הקבצים הקיימים בזיכרון. לחיצה על 1 PB1 קריית text על גבי מסך LCD שמו מופיע בשורה הראשונה. לחיצה נוספת על 0 PB0 דפדף נוסף על מנת להמשיך לקרוא את text. ולחיצה על 1 PB1 חוזרת לתפריט שמות הקבצים.

תיאור חיבור חומרת קצה





مسקנות והצעות לשיפורים

- חשבנו הרבה על תכנון המערכת ועל זמן העבודה למול יעילות הקוד והחומרה. למדנו המונע על תכנון מערכות ובנויות מערכת סגורת עם משקל למשתמש.
- חישני LDR מתנהגים שונה בין חישון לחישון, חישנים שונים בעלי רגישות שונה. אנו מציעים לתת את דגם החישון הספציפי על מנת לאפשר לנו למצוא את data sheets של היצרן ולבזוק האם אנחנו מקבלים תוצאות דומות לתכנון היצרן.
- בהינתן יותר זמן היינו רוצים לבדוק יותר את המערכת על מנת להפוך אותה לטובה הרבה יותר ואיכותית יותר.

