



תיאור מבנה התוכנה הראשית והתנהגות הרובוט

ראשית כדאי לציין כי לבקר איתו עבדנו, HCS12DP256 מבית מומורולה, שפת אסמבלי שונה משפות אסמבלי מסורתיות וקיימים בה אספקטים של תכנות גבוה. החל בפקודות include לקבצים חיצוניים, פקודות if, else, סט פקודות מתמטיות המאפשר עבודה עם מספרים של עד כ- 32 ביט, סט פקודות של לוגיקה עמומה (fuzzy logic) ועוד. ההחלטה לתכנת באסמבלי באה מכך שרצינו תוכנה יעילה המסוגלת בזמן אמת להתמודד עם כמות גבוהה של נתונים חיצוניים.

נעת ננסה לתאר את מבנה התוכנה הראשית ומה היא מבצעת ובהמשך נרחיב על כל נקודה בתוכנה זו, על תכנון האלגוריתמים המבוצעים בה ועוד.

התוכנה מחולקת לשני חלקים עיקריים :

• תוכנות רקע ו Service Routines

משימות מתוזמנות אשר דואגות לפעולתו השוטפת והתקינה של הרובוט. תוכנות אלו רצות בפסיקות (Interrupts) אשר מתוזמנים או על ידי זמנים קבועים, או על ידי אירועים חיצוניים. תוכנות הרקע תופסות חלק קטן מזמן העיבוד וזאת על מנת שלא יפריעו למשימה המרכזית, אך בו זמנית יבצעו את תפקידן כהלכה על מנת שהרובוט יוכל לתפקד. תוכנות הרקע:

○ Real Time Interrupt – פסיקת זמן אמת משולבת עם פעולת Multi Tasking

פסיקה זו רצה כל $426.66_{[\mu s]}$ (תדירות – לפי השעון הראשי של $24_{[MHz]}$ מחולק

-1

10240). כל המשימות המתוזמנות מופעלות על ידי פסיקת הזמן אמת וכל זמן פסיקה הוא זמן היחידה הבסיסי. את כל המשימות המתוזמנות נתאר לפי מספר זמני היחידה הבסיסיים:

- מיצוע קריאות חיישני מרחק – נקבע על פי תדירות הפעולה של חיישני המרחק ($30.3_{[Hz]}$). המיצוע מופעל כל 78 יחידות זמן בסיסיות.
- בקרת מהירות מנועי DC : PID (ראה הסבר בפרק בקרת מנועים) מופעלת כל 20 יחידות זמן בסיסיות.
- בקרת הפרשי דרכים למנועי ה DC : PID (ראה הסבר בפרק בקרת מנועים). מופעלת כל 3 יחידות זמן בסיסיות.
- Auto Shut Down – כיבוי עצמי של הרובוט מבוצע לאחר חצי שעה של עבודה רצופה (ללא כיבוי). זאת על מנת שהסוללות לא יתרוקנו לריק.(על הפונקציה של כיבוי עצמי ראה הסבר בפרק על מייצב המתח ל-5v). פועל פעם יחידה לאחר 2,818,005 יחידות זמן בסיסיות.
- I^2C comms handling – שליטה חיצונית על תקשורת I^2C . על קו תקשורת זה קיים חיישן יחיד (חיישן מרחק אולטרה סוני). לאחר שליחת פקודת קריאה על גבי ה-BUS, החיישן נדגם כל כ- $29.86_{[ms]}$. כך נוצר מצב בו שליחת פקודת קריאה לאחר "זמן צינון", מונעת cross talking בין גלים באוויר שנשלחו בקריאה קודמת והגיעו לחיישן בזמן מאוחר יותר. בקשת קבלת קריאה מן החיישן מופעלת כל 70 יחידות זמן בסיסיות.

בעלי אתר הרובוטיקה הישראלי לא ישאו באחריות כלשהי לכל נזק, כספי או אחר שייגרם במישורין או בעקיפין משימוש במידע המצוי באתר זה

© כל הזכויות שמורות לאסף פוניס, גיא יונה ואלי קולברג
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



○ פסיקת תקשורת RS232 למצלמות :

בעת קבלת מידע מן המצלמות נקראת פסיקה זו. תפקידה היא לקבל את סך הנתונים מן המצלמה ולעבד אותם מפורמט ASCII למספרים בינאריים, לקבלת המידע הסמטיסי של אותה מצלמה. בסוף הפסיקה, נשלח אות למעגל ניתוב המידע הסיריאלי, להחליף את קו המידע מהמצלמה שממנה נשלח המידע, אל המצלמה השניה (ראה פרק מעגל ניתוב מידע סיריאלי)

○ פסיקת תקשורת RS232 ממחשב חיצוני : ממעק משתמש

פסיקה זו נראית כאשר הרובוט מחובר בכבל סיריאלי למחשב חיצוני ונלחץ מקש כלשהו במקלדת. רומינה זו משמשת גם כ- Debugger, שדרכו ניתן לשלוח בזמן אמת נתונים מהרובוט אל המחשב החיצוני. כמו כן הפונקציה יכולה לשלוח חיצונית על תנועת הרובוט. להלן הפונקציות של ממעק המשתמש:

Key	פונקציה
A	הפעלת מנוע ימין קדימה, מנוע שמאל אחורה - סיבוב הרובוט שמאלה
D	הפעלת מנוע שמאל קדימה, מנוע ימין אחורה - סיבוב הרובוט ימינה
W	הפעלת שני המנועים קדימה – רובוט נוסע קדימה
X	הפעלת שני המנועים אחורה – רובוט נוסע אחורה
S	עצירת מנועים
Q	העלה מהירות רצויה למנוע שמאל
E	העלה מהירות רצויה למנוע ימין
Z	הורד מהירות רצויה למנוע שמאל
C	הורד מהירות רצויה למנוע ימין
1	עבור לברירת מצב של הדפסת ערכי קווי A/D – לאחר הקשת "1", הקשת מספר בין 0 ל 9 – ידפיס את קו ה-A/D המתאים על גבי המסך ובנוסף את המרחק המדוד על ידי החיישן האולטרה סוני
P	הדפס על המסך את הערכים של סמטיסימיקת הצבעים הנראים על ידי 2 המצלמות
O	הדפס על המסך את הערכים של מיקום הגוף הרצוי הנראה על ידי 2 המצלמות
I	החלף (toggle) את הקו המחליף בין המצלמות (ראה מעגל ניתוב סיריאלי)
/	עוצר את תוכנת הרובוט SWI – Software Interrupt
M	ולאחר מכן הספרה 1 או 2 – הפעל/הפסק מאוורר

כמו כן על מנת שלא יהיה הבדל בין אותיות ראשיות ומשניות, יש צורך להמיר, אך זוהי תוספת קבועה לכל האותיות, בשל היותן בפורמט ASCII:

```
LDAA SCODRL          מען לצובר A את ערך ה- ASCII של המקש שהתקבל
converting all "small" letters to "big"
CMPA #'a'           השווה לערך ה- ASCII של a
BLO NEXT           אם נמוך, עבור הלאה
CMPA #'z'           אם לא נמוך, השווה לערך ה- ASCII של z
BHI NEXT           אם גבוה, עבור הלאה
ADDA #'A'           אם לא גבוה, הוסף לצובר A בדיוק את הפרש
SUBA #'a'           שבין אות גדולה (capital) לאותה אות קטנה.
```

ממעק המשתמש יכול לפעול כל עוד ישנו קשר בין המחשב החיצוני והמחשב ובעזרתו ניתן לקבל בזמן אמת נתונים מן הרובוט.

בעלי אתר הרובוטיקה הישראלי לא ישאו באחריות כלשהי לכל נזק, כספי או אחר שייגרם במישורין או בעקיפין משימוש במידע המצוי באתר זה



○ **פסיקת צוברי הפולסים:**

בבקר הראשי של הרובוט קיימים 2 סופרי פולסים המחוברים ישירות למקודדים האופטיים המחוברים למנועי ה- DC (משוב להירות המנוע). כאשר ישנו overflow בצובר (נספרו 65535 פולסים), נקראת פסיקה זו.

○ **פסיקת תקשורת I²C**

פסיקה זו נקראת כאשר בייט שלם הועבר בהצלחה (או לא בהצלחה) על גבי ה-BUS. בהתאם לנסיבות, באם היה כשל בתקשורת, הקו נסגר ומאותחל מחדש. העברת נתונים עוברת דרך Buffer ומצביע פנימי לנתונים שעליהם להישלח. באם הבייט הועבר בהצלחה, נשלח/מתקבל הבייט הנוסף הבא.

בעלי אתר הרובוטיקה הישראלי לא ישאו באחריות כלשהי לכל נזק, כספי או אחר שייגרם במישורין או בעקיפין משימוש במידע המצוי באתר זה

© כל הזכויות שמורות לאסף פוניס, גיא יונה ואלי קולברג
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



● תוכנה ראשית

תוכנה זו תופסת את רוב זמן המעבד והיא למעשה מקבלת משתנים מתוכנות הרקע ומחיישנים חיצוניים ומחליטה על התנהגות הרובוט בהתאם לכך. מטרתה היחידה היא המטרה העליונה של הרובוט – זיהוי האובייקט הרצוי, בהתאם לפרמטרי הסטטיסטיקה המתקבלים מתוכנות הרקע. על מנת לשמור על יעילות גבוהה, תוכנה זו פושטת עד כמה שהייה ניתן. האלגוריתם הכללי הוא שהרובוט מסתובב עד מציאת בועה בצבע ובגודל המתאים ללהבה, באחת מן המצלמות. בגלל תדירות נמוכה יחסית של תמונות ושל התקשורת (תקשורת סיריאלית) הרובוט אינו מסתובב באופן חלק, אלא מסתובב זווית של כ- 72 מעלות, עוצר, ואז ישנו זמן חשיפה למצלמות על מנת שמסנן ה- white balance יסתגל לעוצמה במצב החדש, ואז נלקחת הדגימה.

באופן זה, מסתובב הרובוט ומחפש את אותו אובייקט (להבה) עד שמוצא אותה. כעת, תוך כדי בדיקה חוזרת ונשנית כי לא נעלמה אותה בועה מן המצלמה הזו, מחפש הרובוט עם המצלמה השנייה את אותו אובייקט. מכיוון שלהבה נראית כבועה, היא בעלת סימטריה סיבובית ועל כן צורתה תראה זהה גם במצלמה השנייה. כמו כן, 2 המצלמות באותו הגובה ולכן יראו את הבועה באותו הגובה.

כאשר יראו 2 המצלמות את הבועה, יחושב המרחק אל הנר ע"י אלגוריתם הראיה הסטיריאוסקופית (ראה פרק הראיה הסטיריאוסקופית). כעת יתחיל לנוע הרובוט אל הנר תוך כדי לקיחת דגימות מן המצלמות, עד אשר יגיע אל הלהבה. את המרחק הרצוי מקבל הרובוט לפי המרחק שלו מן הלהבה, המתקבל מאלגוריתם הראיה הסטיריאוסקופית. בכל נקודה בזמן ניתן לדעת את המרחק שעבר הרובוט בעזרת משוב מן המקודדים האופטיים המורכבים על מנועי הרובוט. באם מאבד הרובוט את האובייקט באחת מן המצלמות, יעצור ויחפש שוב את האובייקט באותה מצלמה, תוך כדי בדיקה חוזרת ונשנית, כי האובייקט לא נעלם גם מהמצלמה השנייה. כאשר מגיע הרובוט למרחק קצר מן הלהבה – יתכוונן הרובוט כלפיה, יפעיל את מנוע המאוורר וכיבה את הלהבה.

כמו כן בכל דגימה חדשה של כל אחת מן המצלמות, בהנחה כי האובייקט עוד בתמונה, ישנו מיצוע על קוארדינטות האובייקט ומדידת ה- variance של המדידה האחרונה ביחד עם שלושת המדידות האחרונות. כמו כן, הערכים בין המצלמות משווים על מנת לראות שהמצלמות מסתכלות על אותו האובייקט.

שילוב 2 מצלמות מקל על מציאת הבועה (הלהבה) שכן, בשימוש במצלמה יחידה, בהתאם לכל מרחק ומרחק מתקבל רדיוס בועה שונה, אך בשילוב 2 המצלמות, ניתן להשוות בין גדלי הבועות המתקבלות. שתייהן באותו סדר גודל ובמיקומים (לאורך הציר האנכי) דומים.

אחת הבעיות המרכזיות בראיה הסטיריאוסקופית הינה תפיסת אותה נקודה של האובייקט על ידי 2 המצלמות וזהו הגורם העיקרי לשגיאות.

התוכנה הראשית מקבלת את תוצאות חיישני המרחק, לאחר שעברו מיצוע ב- Real Time Interrupt, ובודקת כל חיישן כזה על מנת לראות והרובוט אינו קרוב מידי לחפץ חיצוני. במקרה כזה, הרובוט מתרחק מהחפץ הזה וחוזר על חיפוש הלהבה.

המשימה דורשת שליטה ובקרה על חלקים רבים, עיבוד מידע חיצוני המתקבל בזמן אמת ותיאום מלא בין כל תת-המשימות. בפרקים הבאים נפרק את הבעיה הכוללת לתת-בעיות ונסביר כיצד פתרנו אותם, את האלגוריתמים אשר שימשו בתוכנה, את האלקטרוניקה אותה תכנונו והרכבנו, את המערכת המכאנית הכוללת והחשוב מכל – כיצד מבוצעת האינטגרציה ביניהם.

בעלי אתר הרובוטיקה הישראלי לא ישאו באחריות כלשהי לכלל נזק, כספי או אחר שייגרם במישורין או בעקיפין משימוש במידע המצוי באתר זה

© כל הזכויות שמורות לאסף פוניס, גיא יונה ואלי קולברג
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



● **הממשק בין התוכנה הראשית, הפסיקות וה- service routines**

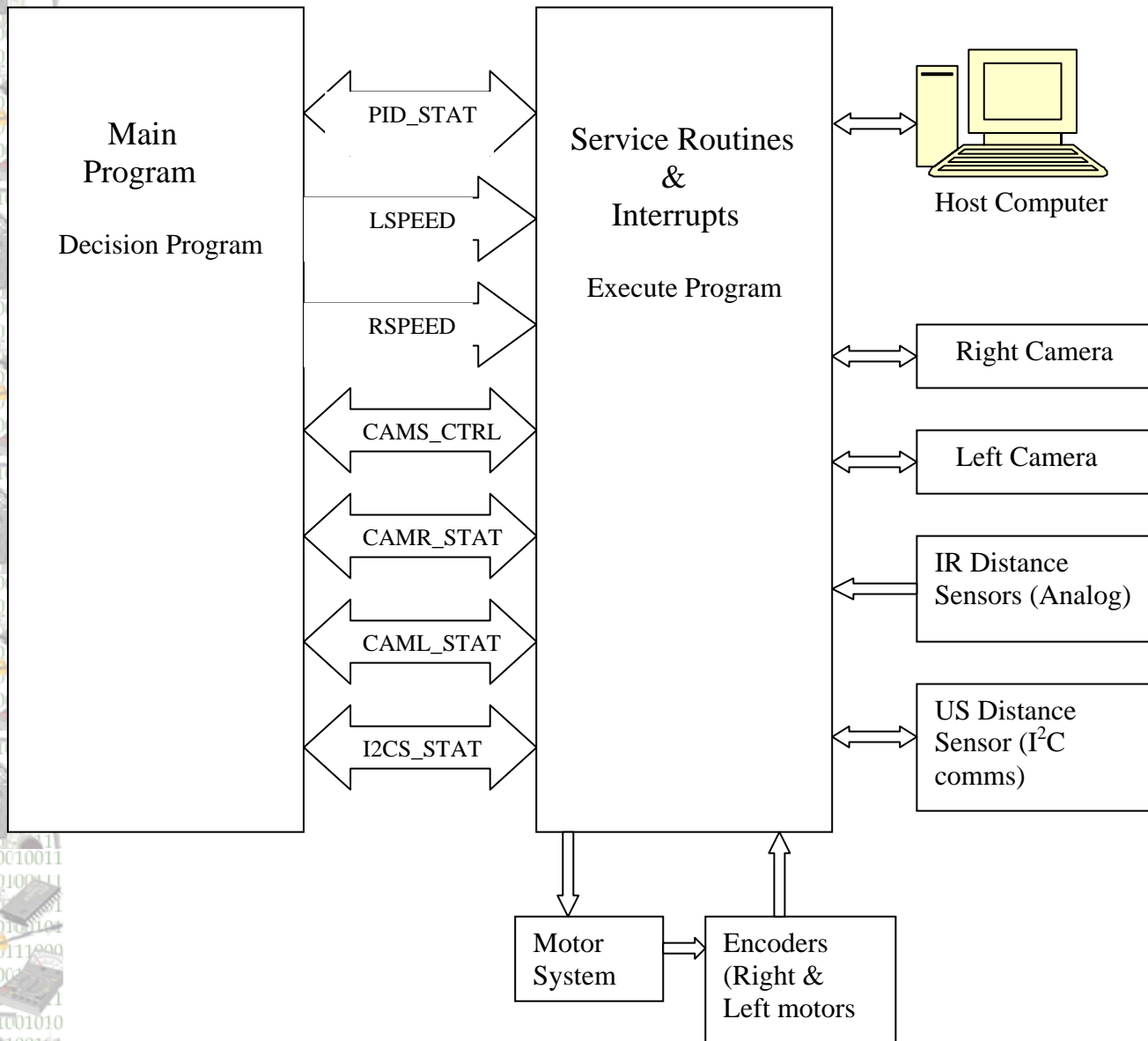
על מנת לאפשר תקשורת פנימית בין התוכנה הראשית לבין הפסיקות וה- service routines, סט של חוצצים פנימיים משמש להעברת המצב (סטטוס) בכל זמן נתון, של כל תת מערכת. למערכת הבקרת מהירות – pid_stat (דרכו ניתן לדעת האם יש לבצע בקרה בכלל, האם רק לאחד המנועים, לשניהם ועוד). בנוסף, המהירויות הרצויות לכל מנוע נקבעות בתוכנה הראשית על ידי האוגרים RSPEED ו-LSPEED. למצלמות – cams_ctrl (מורה ל- service routines כיצד להשתמש במצלמות – רק באחת מהן, בשתייהן, למתג ועוד), camr_stat ו- caml_stat (המורים על האם התקבלו קריאות חדשות מאז שנודגמו ועוד, לכל מצלמה בנפרד), למערכת התקשורת לחיישן המרחק האולטרה סוני- I²C – i2cs_stat.

ההחלטות המרכזיות נעשות בתוכנה הראשית (Decision Program) ובהתאם לכך נקבעים, דרך הממשק, מתקבלים משתנים אלו לתוכנות הרקע (Execute Program). לתוכנה הראשית אין מגע ישיר עם העולם החיצוני וכל המידע היוצא והנכנס ממנה ואליה, מגיע דרך תוכנות הרקע.

להלן דיאגרמת בלוקים של מערכת המידע (חץ דו כיווני מצביע על תקשורת דו כיוונית, חץ חד כיווני מצביע על תקשורת חד כיוונית):

בעלי אתר הרובוטיקה הישראלי לא ישאו באחריות כלשהי לכל נזק, כספי או אחר שייגרם במישורין או בעקיפין משימוש במידע המצוי באתר זה

© כל הזכויות שמורות לאסף פוניס, גיא יונה ואלי קולברג
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



בעלי אתר הרובוטיקה הישראלי לא ישאו באחריות כלשהי לכל נזק, כספי או אחר שייגרם במישרין או בעקיפין משימוש במידע המצוי באתר זה

© כל הזכויות שמורות לאסף פוניס, גיא יונה ואלי קולברג
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר