The three fundamental Rules of Robotics:

One: a robot may not injure a human being, or, through inaction, allow a human being to come to harm.

Two: a robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

Three: a robot must protect its own existence as long as such protection does not conflict with the First and Second Laws.

- Isaac Asimov

תוכן העניינים

מבוא............ 3
חלק א' – חומרה............ 4
פרק א' – פרוט חלקי הרובוט............ 5
סעיף א' – חיישנים............ 5
סעיף ב' – כרטיסי אלקטרוניקה............ 17
סעיף ג' – בטריות (מצברים) 23
סעיף ד' – גלגלי תנועה גלגלי עזר............ 25
סעיף ה' – מנועים............ 28
פרק ב' – שיטות הנעת הרובוט............ 29
פרק ג' – מבנה הרובוט............ 35
פרק ד' – מבנה הרובוט "קרפלעך"............ 37
חלק ב' – התוכנה............ 41
פרק ה' – מבנה התוכנה............ 42
פרק ו' – פסיקות בתוכנה............ 46
פרק ח' – הניווט............ 53
פרק ט' – תפעול חיישנים בתוכנה............ 55
פרק י' – הכיבוי............ 65
פרק יא' – תיקונים............ 69
פרק יב' – תפעול מנועים בתוכנה............ 74
נספח א' – בסיסי ספירה, אריתמטיקה בינארית ושערים לוגיים............ 75
נספח ב' – משוואות תנועה וחיישנים............ 88
נספח ג' – המכניקה של תנועת הרובוט............ 93
נספח ד' – פניות............ 97
נספח ה' – בעיות............ 102
נספח ו' – מימוש משוואות באסמבלי............ 112
נספח ח' – תיקונים יחסיים (PID) 117
נספח ט' – חוקי תחרות הרובונר העולמית 2002............ 120
נספח י' – תוכנת הרובוט "קרפלעך"............ 135
התוכנה הראשית............ 135
Tikunim.asm............ 140
Equates.asm............ 145
Fire.asm............ 150
HelpFunc.asm............ 157
Inits.asm............ 158
Ir_Ovf.asm............ 161
Motors.asm............ 163
Sensors.asm............ 168
Print.asm............ 173
ביבליוגרפיה............ 174

מבוא

רובוט – מילה זו מגלמת מאחוריה דמיון ומדע בדיוני. האדם הממוצע רואה בעיני רוחו אוטומט, אולי דומה בצורתו לבן-אנוש, העושה פעולות מורכבות היוצרות רושם שהן נעשות על ידי בן-אדם חי ובר דעת. האמת שונה במעט, הרובוטים פלשו לחיינו מבלי שחשנו בהם. הם מקיפים אותנו מכל עבר. אנו פוגשים אותם יום יום שעה שעה בבית בעבודה ובמקומות הבילוי. הם עושים במקומנו עבודות קשות ובתנאים קשים הם עובדים ומיצרים ללא לאות ללא הפסקה ללא חופשה וללא דרישות מיוחדות. האם חשבנו לעצור ולחפש את הרובוט? האם חשבנו שפקיד הבנק הוחלף על ידי הרובוט – הכספומט. פועלי היצור בתעשיית הרכב הוחלפו מזמן על ידי רובוטים המייצרים את החלקים השונים, אלה הם מערכות ה-CNC ואחרים יודעים להרכיב את החלקים זה לזה. האם המשיבון האוטומטי שמנתב אותנו אל קווי הטלפון הרצויים אינו רובוט? ומה היא מכונת הכריכים והשתייה האוטומטיים? האם זה אינו רובוט?

מבט כה פשוט סביבנו, מלמד עד כמה הוא פלש לחיינו, ואנו קיבלנו אותו כמובן מאליו. עוד רבה הדרך עד לאותו יום שבו ידמה הרובוט לאדם בצורתו החיצונית. בדיבורו ואופן מחשבתו. אך כנראה שיום זה יגיע. כל דמיון היום ייהפך למציאות בעתיד, כדאי לקרוא את ספריו ה"דמיוניים" של ז'ול וורן. האם הם דמיוניים היום?

אנו שנולדנו לתקופה מלאת אתגרים בתחום המחשבים, החלל והרובוטיקה. נלהבנו לצלול לעולם כה קסום וללמוד מעט על הטמון בו. בדיעבד אנו שמחים על האפשרות שניתנה לנו להכיר ולהתנסות בטכנולוגיות כה מתקדמות.

הגענו להישגים מרשימים הודות לבית הספר שתמך בפרויקט. לחברינו בצוות וכמובן לחשוב מכולם מורה ומנחה הקבוצה מר קולברג אלי.

קבוצת הרובוט "קרפלעך"
רמת-גן
יוני 2002

חלק א' – חומרה

מהי חומרה?

חומרת הרובוט היא למעשה כל החלקים הפיזיים המרכיבים אותו, כלומר הכרטיסים האלקטרוניים השונים, החיישנים, המנועים, הגלגלים, החומר ממנו מורכב בסיס הרובוט וכדומה. החומרה מהווה חלק בלתי נפרד מהרובוט מכיוון שבלעדיה לא ניתן יהיה ליצור רובוט שפועל.

בפרקים הבאים נסביר על מבנה רובוט אידיאלי, חיישנים, שיטות הנעה של רובוטים, החומרה ברובוט "קרפלעך" ועוד.

פרק א' – פירוט חלקי הרובוט

סעיף א' - חיישנים

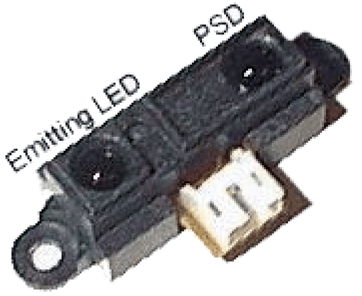
חיישני מרחק

למען ביצוע המשימה העומדת בפני הרובוט, הוא חייב לנווט בהצלחה במסדרונות ובחדרי הזירה, ולנתח את פעולותיו הבאות בהתחשב בנתונים של מרחקו מהקירות מצדדיו, מלפניו וממיקומו הכללי במבוך ובחדריו.

לשם כך, התקנו ברובוט חיישני מרחק ספורים, אשר יתפקדו בשבילו כעיניים, ובעזרתם יתמצא יצליח במשימתו.

לביצוע המשימות הדרושות ניתן היה להשתמש בחיישני קירבה, אשר מחזירים אות רק ברגע שהם מגיעים למרחק מסוים מעצם כלשהו, אך עדיפים עליהם חיישני המרחק, המפיקים אות רציף המשתנה בהתאם למרחקם מעצמים.

חיישן מרחק אינפרא-אדום GP2D12

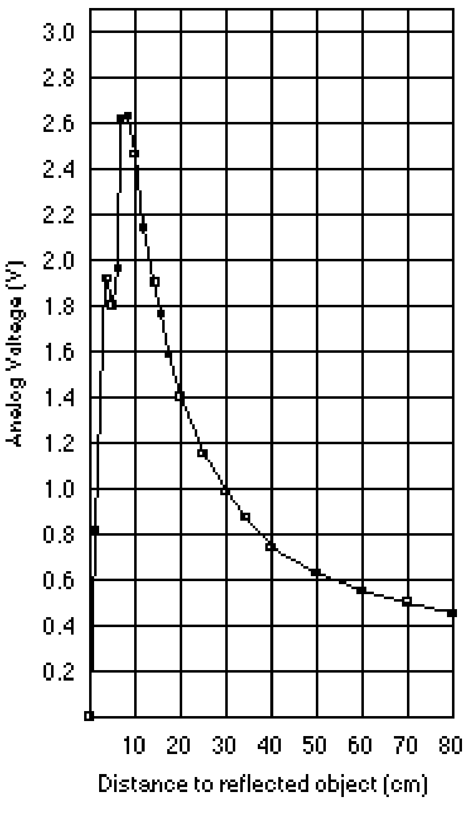


חיישן אינפרא אדום GP2D12

כיצד פועל חיישן מרחק אינפרא אדום?

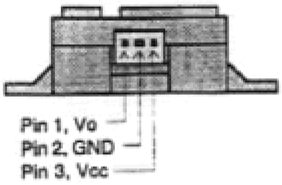
חיישן מרחק זה שולח אות אינפרא אדום מהמשדר (Emitting LED), הנקלט בחלק הקולט של החיישן. האות הנקלט מגיע בעוצמה שהולכת ונחלשת ככל שהמרחק גדל.
לאחר קליטת האות מספק החיישן מתח בהתאם לעוצמת האור שנקלט (יפורט בהמשך). הזמן הנדרש לחישוב המרחק ועדכון המתח המתקבל מהחיישן הוא 32ms.
החיישן מחזיר אות אנלוגי (0-5 וולט) המציין את המרחק בינו לבין העצם מולו הוא נמצא. טווח הגילוי של חיישן זה הוא 8 – 60 ס"מ.

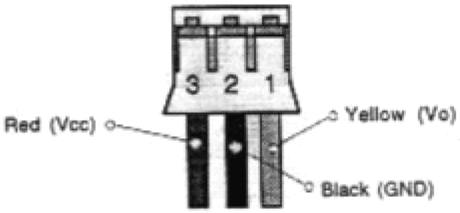
הערך האנלוגי, שהחיישן מחזיר בכל 32 ms, מועבר לממיר A/D (ממיר אנלוגי לדיגיטאלי), שממיר את האות לערכים מספריים איתם יוכל לעבוד המעבד. הפלט המתקבל מהחיישן אינו ליניארי ומתואר בגרף:



<u>חיבורי חיישן האינפרא אדום</u>

לחיישן ישנם שלושה הדקים:





הדק מס' 1 - חיבור המידע בו החיישן מחזיר מתח בין 0 ל 2.6 וולט. מידע זה מועבר דרך כרטיס הInterface למעבד שבו מתבצעת המרת המידע האנלוגי למידע דיגיטלי.

הדק מס' 2 - חיבור ה Ground.

הדק מס' 3 - חיבור המתח (VCC) – נע בין 4.5 ל 5.5 וולט.

יתרונותיו העיקריים של חיישן זה הם מימדיו הקטנים ומחירו הזול ונוחות השימוש שלו.

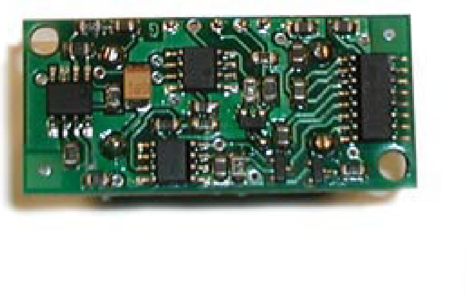
חסרונות חיישן האינפרא אדום הם –

- טווח הפעולה שלו – כאמור, החיישן מזהה עצמים רק במרחקים של 810 עד 60 ס"מ. כלומר, מרחקו של כל עצם הנמצא במרחק של פחות משמונה ס"מ או יותר משישים ס"מ לא יחושב כהלכה. בעיה זו לא הייתה נוראה לו לא היה החיישן מספק קריאות בטווחים אלה, אך במקום זאת הוא מספק קריאות לא נכונות שעלולות לגרום לתוכנה לבצע פעולות מוטעות.
- החיישן האינפרא אדום רגיש למדי, ויש לשמור על העדשות שלו נקיות מאבק ו/או כל חומר אחר, כי כל לכלוך ישבש את קריאותיו.
- מקורות אור חיצוניים בתחום האינפרא אדום (השמש, פלאשים של מצלמות, מצלמות וידאו וכדומה) המכוונים ישירות לעדשה הקולטת של החיישן ישבשו אף הם את קריאותיו.

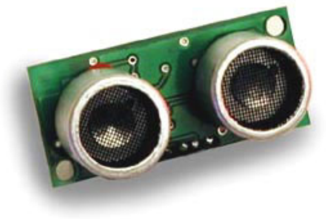
<u>טיפים לשמירה על החיישנים:</u>

- יש לשמור על העדשות נקיות. כל לכלוך עליהן ינמיך את ביצועי הקולטים.
- החיישנים משדרים באורך גל של 780 ננומטר עד 920 ננומטר, לכן אם מכסים את החיישן במכסה מגן, יש לוודא שהמגן שקוף לאורכי גל אלו.
- יש להימנע מחשיפה של החיישן לאור "חזק" כמו אור שמש, נורות טונגסטן, זרקורים וכדומה.
- כדי להפחית את טעויות המדידה כאשר מודדים עצם בתנועה, יש להציב את החיישן כאשר החלק הארוך של החיישן מקביל לכיוון התנועה.
- מומלץ לחבר קבל של 10 מיקרו פרד בין חיבור מס' 2 ל 3 כדי לייצב את המתח הנכנס לחיישן.
- טמפרטורת הפעולה של החיישנים היא מנוס 10 מעלות עד 60 מעלות צלזיוס.

חיישן מרחק אולטרא-סוני SRF04



צילום החיישן מאחור



חיישן אולטרא-סוני SRF04

כיצד פועל חיישן מרחק אולטרא-סוני?

החיישן משדר גל קול, אשר נמצא מעבר לטווח השמיעה של אדם, ונע, מן הסתם, במהירות הקול (300 מטרים לשנייה בקירוב). הגל נע בצורת חרוט שקודקודו בחיישן, ומוחזר אל החיישן אחרי פגיעה בעצם. לאחר שידור הגל, החיישן "מחכה" לקבלת החזר הגל (הד). ברגע קבלת ההד החיישן מסוגל לחשב את מרחקו מן העצם ע"פ משך הזמן שלקח להד לחזור.

הדקי החיישן האולטרא-סוני

לחיישן זה ארבעה הדקים בהם אנו משתמשים -

הדק מס' 1 הוא חיבור המתח.

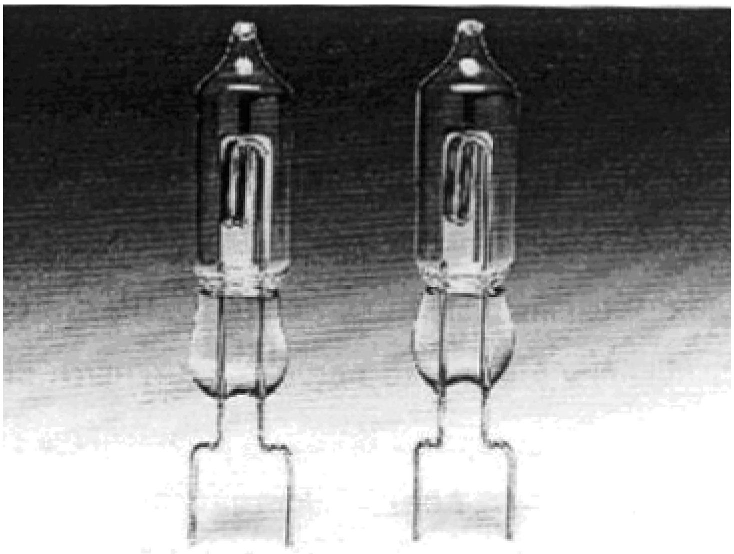
הדק מס' 2 הוא חיבור ה Ground

הדקים מס' 3 ו-4 משמשים לחיישן לאות לשליחת גל ואות שמסמן את שליחת הגל וחזרתו.

חיישנים למציאת נר

על הרובוט לזהות את הנר בחדרי המבוך, ולכן הוא זקוק לחיישנים אשר מסוגלים לזהות את הנר בצורה של חום או בצורה של קרינה אולטרא-סגולה. לכן כל רובוט שמשתתף בתחרות זקוק לחיישני זיהוי נר.

חיישן קרינה אולטרא סגולה Hamamatsu UVtron R2868

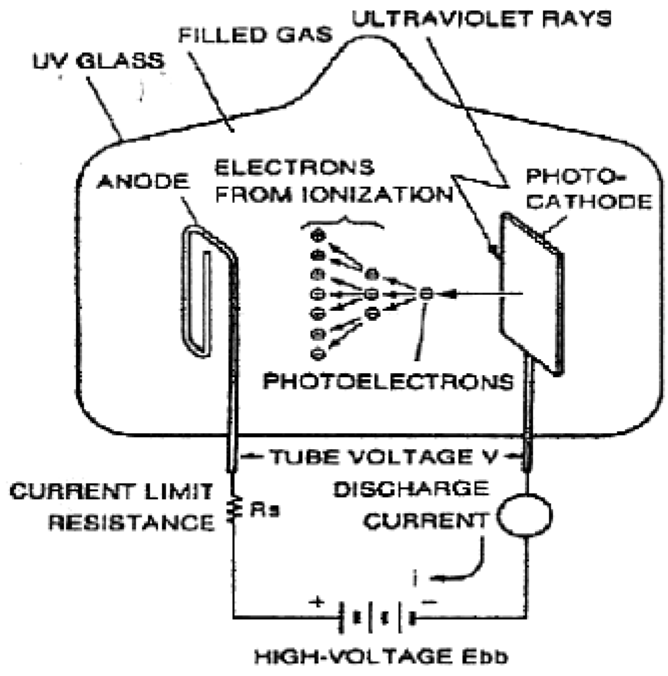


תמונת החיישן R2868

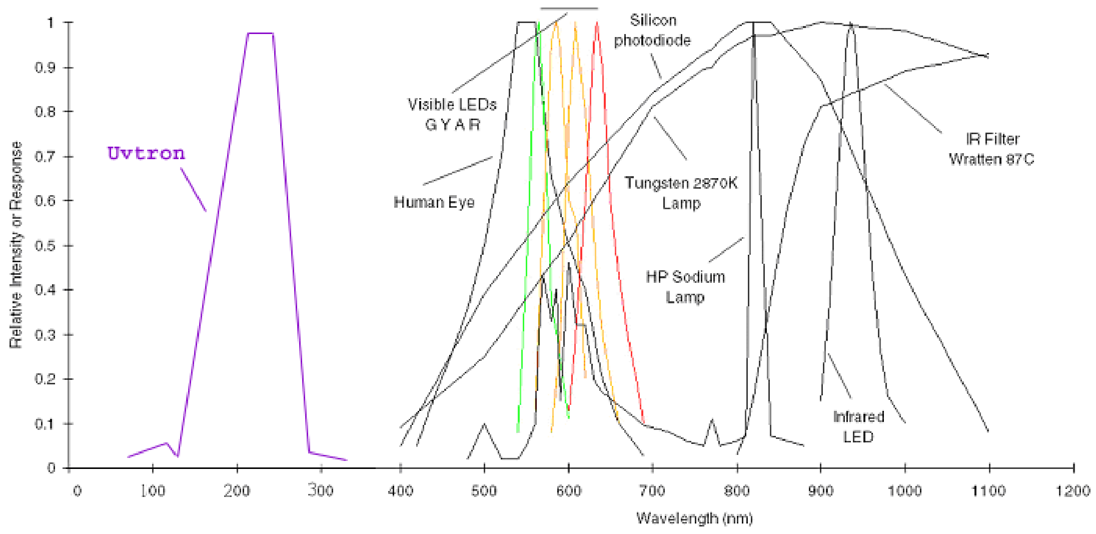
חיישן הUvtron משמש את הרובוט לבדיקה האם הנר נמצא בחדר מסוים של המבוך או לא. חיישן זה רגיש מאוד וחוסך לרובוט כניסה לחדר על מנת לבדוק אם הנר נמצא בו ברוב המקרים, וחיפוש דקדקני אחר הנר. בפרק זמן קצר מאוד הרובוט יכול לדעת אם הנר נמצא בקרבתו או לא.
יש לציין שהחיישן יכול לזהות רק נר דולק בחדר, ולא את מיקומו, עוצמתו או מרחקו מהרובוט.
החיישן מזהה אורכי גל של קרינה אולטרא סגולה הנפלטת מלהבת הנר (או כל להבה אחרת), באורך גל של 185-260nm וברגע זיהוי קרינה באורך גל שכזה הוא שולח אותות חשמליים בתדירות נמוכה שאומרים לתוכנה שלהבה נמצאת בקרבת מקום.

החיישן עצמו מורכב משפופרת זכוכית שקופה, הממולאת בגז דליל. בתוך השפופרת יש שני לוחות מתכת, המשמשים כאנודה וקתודה. תחת תנאים רגילים, כשאין להבה בסביבה, בתוך כדור הזכוכית קיים נתק.
כאשר קרינה אולטרא סגולה חודרת לשפופרת, פוטונים של הקרינה האולטרא סגולה פוגעים בקתודה ומשחררים אלקטרונים לגז הנמצא בשפופרת. האטומים בגז מקבלים פרץ אנרגיה מהאלקטרונים הללו, ופולטים גם הם אור באורך גל אולטרא סגול. האור הנ"ל גורם לעוד אלקטרונים להיפלט מהקתודה והתופעה חוזרת על עצמה. במהלך תהליך זה הגז הנמצא בשפופרת מתמלא באלקטרונים חופשיים, והופך למעשה למוליך.
כך, רואים שחיישן זה הוא מעין מפסק זרם, המספק שיאים של מתח כתגובה לחשיפה לקרינה אולטרא סגולה.

להלן שירטוט של מבנה השפופרת:



החיישן יזהה באופן אופטימאלי קרינת UV של להבה שמקורה בבערה, וזה בדיוק סוג התפקוד הדרוש לנו. בעירה של גז פולטת קרינה באורכי גל של עד 180 nm, ובעוד שהקרינה האולטרא סגולה המגיעה מהשמש יכולה להיות מזוהה על ידי החיישן, רובה נספגת על ידי האטמוספרה ולא מגיעה לקרקע.



תגובת החיישן כפונקציה של אורך הגל

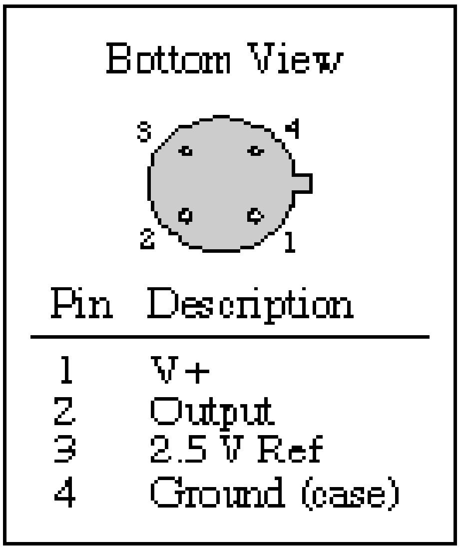
החיישן רגיש מאוד, ומסוגל לגלות קרינה אולטרא סגולה הנפלטת מלהבה קטנה במרחק של יותר מחמישה מטרים. רגישות זו, על אף יתרונה באמינות מציאת הנר, בעלת חיסרון גדול, שכן במידה ואדם מחוץ למבוך מדליק להבה בזמן ריצת הרובוט, הסיכויים גדולים שהרובוט יזהה את הלהבה ויסיק מכך שנר נמצא בקרבתו.

בנוסף, יש לשים לב לכך שהחיישן עצמו עדין מאוד, ויש לטפל בו בזהירות מקסימאלית. הוא עשוי מזכוכית, ואם היא תישבר ישתחרר הגז הכלוא בה ותיפגע פעולתו של החיישן.

<u>חיישן פירו-אלקטרי Eltec-442</u>

חיישן הפירו רגיש לאור, ומיועד לזיהוי תנועה של בעלי חיים או בני אדם, או לזיהוי מקורות חום, כגון נרות במקרה שלנו.
על החיישן מורכבת עדשה עשויה פוליאתילן. חומר זה נבחר מכיוון שהוא בולע את האור פרט לאורכי הגל 800-1400nm, וזהו טווח אורכי הגל אותו אנו רוצים לזהות. החיישן רגיש לכל סוגי הקרינה האופטית בין אולטרא סגול לגלי אינפרא אדום. העדשה המורכבת על החיישן מגבילה את האנרגיה המגיעה אליו לחלון קטן מאוד של בערך 700-1600nm. בני אדם, למשל, פולטים קרניים בגלי אורך מקסימאליים של מעט מעל 900nm – גלי אורך אותם החיישן יזהה באופן אופטימאלי, זאת מכיוון שהחיישן נועד לגלות בני-אדם וחיות ולא להבת נר.
החיישן פועל בטמפרטורות של (-40) - 70 מעלות צלזיוס.

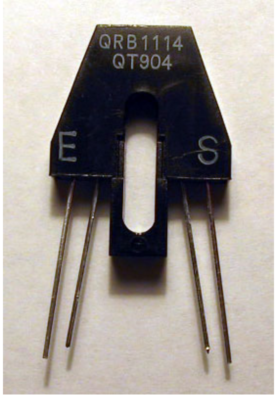
להלן פירוט ההדקים של החיישן:



הרכבת עדשת הפרנזל על החיישן הייתה כריכה בהכנת קונוס העדשה. עם החיישן, צורף גם נייר קרטון עליו שרטוטים להרכבת העדשה. היה צורך בגזירה והדבקה עדינה של כל המרכיבים על מנת להגיע לקונוס הרצוי.

הקונוס הורכב מגוף הקונוס, ומשלוש טבעות – טבעת מרכזית, חיצונית ופנימית. הטבעות מקיפות את העדשה, ומחברות אותה לקונוס באופן בו היא לא תזוז כלל, כדי שקריאות החיישן יהיו יציבות.

<u>חיישן פס לבן</u>



החיישן QRB1114

חיישן הפס הלבן, כשמו כן הוא, מגלה את הצבע הלבן. הוא לא מזהה רק פסים לבנים, כמובן, אלא כל משטח לבן (או בהיר צבע) אליו הוא מכוון.

החיישן שולח קרן אור. קרן האור פוגעת במשטח שנמצא מול החיישן (אנחנו הנחנו את החיישן מילימטרים ספורים מהרצפה, כי מטרתו ברובוט שלנו הייתה לסמן לרובוט כאשר הוא נמצא בכניסה לחדר, או מול נר), ולפי אחוז האור המוחזר אל החיישן מהמשטח, נקבע צבע המשטח. משטחים לבנים, או בהירים מאוד, יחזירו אחוז גבוה יותר של האור אל החיישן.

האור המוחזר מהמשטח אל החיישן נקלט על גבי לוחיות מתכת קטנות, אשר מקבלות כתוצאה מכך מתח חלש, המוגבר על ידי מעגל על החיישן, ומעגל אחר שולח לבקר אותות דיגיטאליים בהתאם לקריאת הלוחיות.

<u>חיישן המיקרופון</u>

חיישן המיקרופון על גבי הכרטיס

חיישן המיקרופון מזהה תדרים בין 0-8Khz ומחזיר מתח בהתאם. החזר החיישן הוא לינארי (ב1Khz – 1 וולט, ב2Khz- 2 וולט וכדומה) כאשר מעל 8Khz החיישן יחזיר תמיד 8 וולט.
השימוש בחיישן המיקרופון פשוט ביותר, ולכן הוחלט להשתמש בו ברובוט על מנת לקבל את ניקוד הבונוס שמתקבל אם הרובוט מתחיל את הריצה שלו על ידי צפצוף, ולא על ידי הפעלה ידנית.

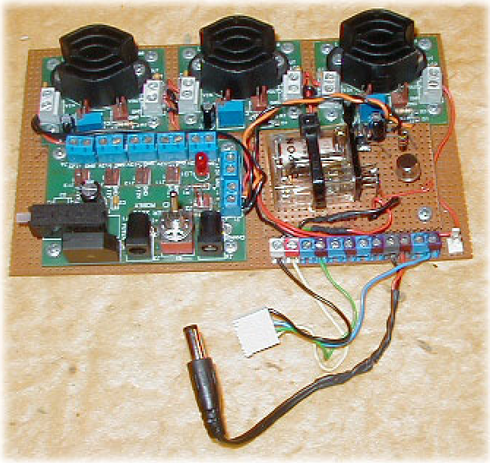
סעיף ב' – כרטיסי אלקטרוניקה

כרטיס החשמל

חלקי הרובוט דורשים מתחים שונים. לכן בנינו כרטיס חשמל שתפקידו לחלק את המתחים השונים בין הכרטיסים והחיישנים. כרטיס החשמל בנוי מכרטיס הגנה וחלוקת מתחים ראשונית, שלושה מייצבים, וכרטיס השהיית ספק.

כרטיס ההגנה וחלוקת המתחים הראשונים הינו כרטיס אשר מקבל 16 וולט ומוציא מתח אחד של 16 וולט וחמש יציאות של מתח 14.4 וולט. תפקיד נוסף של הכרטיס הוא הגנה. על הכרטיס ישנו גשר דיודות אשר מונע חזרה של זרם ונתיך אשר מונע לזרם גדול לעבור דרך הכרטיס.

כרטיס המייצב הוא רכיב חשמלי אשר מקבל בכניסה מתח מסוים וביציאה מוציא מתח נמוך יותר על ידי המרת ההפרש לחום שמתפתח על הרכיב הממיר. בשל עובדה זו התקנו "אמבטיות חום" שתפקידן הוא למשוך את החום מרכיב המייצב ולמנוע התחממות יתר שלו ועקב כך הריסתו. באמצעות המייצב אנו יכולים ליצור מתחים שונים לשימוש הכרטיסים והחיישנים שעל הרובוט.



כרטיס החשמל

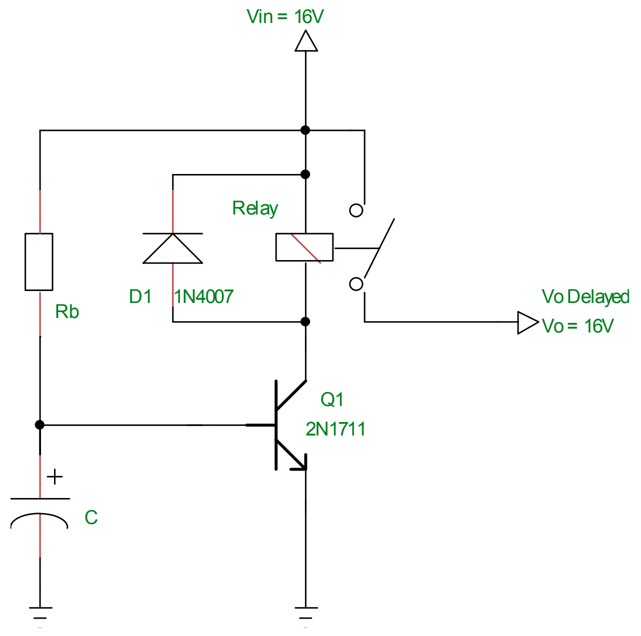
כרטיס השהיית ספק 16 וולט

יש לספק לכרטיסי המנועים מתח 16 וולט אך ורק לאחר אספקת מתח ה5 וולט לכרטיס זה. ההשהיה הדרושה בין אספקת המתחים היא לפחות שנייה אחת, אחרת הכרטיס נכנס לרוויה וזורם זרם קצר במנועים.
בהפעלות הראשוניות של הרובוט סיפקנו את מתח ה16 וולט בעזרת מתג נפרד אשר הופעל ידנית לאחר אספקת מתח ה5 וולט. בגמר בניית הרובוט נוצר הצורך לפשט את פעולת אספקת המתחים. אי-לכך החלטנו לבנות מעגל השהייה אשר יספק אוטומטית מתח זה לאחר הופעת מתח ה5 וולט.
להגדלת אמינות מעגל ההשהיה החלטנו לבנות אותו סביב טרנזיסטור ולא מעגל מוכלל (אינטגרלי) כגון ה555 כי הטרנזיסטור חסין יותר לרעשים ולתנודות מתח.

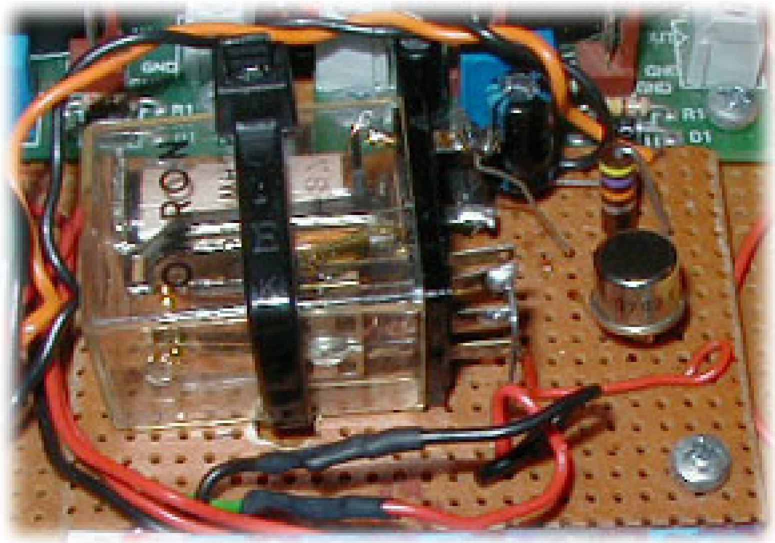
המעגל בנוי סביב טרנזיסטור 2N1711 (ראה איור). טרנזיסטור זה מסוגל לעמוד בזרמי קולקטור של עד 600mA. מסיבה זו נבחר סליל הממסר שמחובר להדק זה לזרם תפיסה נמוך יותר. הטרנזיסטור נכנס להולכה ולרוויה כאשר מתח בסיס-אמיטר הוא כ0.7 וולט. מתח זה מתקבל ממעגל השהייה Rb,C. המתח על פני הקבל נקבע מתוך הנוסחא:

$$V_c = V_0 \cdot e^{-\frac{t}{R\cdot C}}$$

בעת ניתוק סליל הממסר נוצר על פני הסליל מתח מושרה גבוה (לפי חוק לנץ). מתח זה שקוטביותו הפוכה למתח הטבעי של הקולקטור יכול לגרום לנזק בלתי הפיך בטרנזיסטור. אי לכך, חוברה דיודה D1 במקביל לסליל הממסר. הדיודה נכנסת להולכה אך ורק בהיווצר תופעת המעבר של המתח על הסליל בעת הניתוק. בשאר הזמן הדיודה נמצאת נגד כיוון ההולכת, אי לכך היא בקיטעון ואינה משפיעה.
בעת שהממסר מופעל נסגר מגעו אשר מעביר את מתח ה16 וולט לכרטיס המנועים.



שרטוט מעגל ההשהיה



מעגל ההשהיה שבנינו

<u>תכנון המעגל:</u>

- התנגדות סליל הממסר 600 אום.
- כאשר הטרנזיסטור רווי המתח על הטרנזיסטור 0.2 וולט ועל הממסר 15.8 וולט.
- זרם הקולקטור (הממסר): $Ic = \frac{15.8}{600} = 26mA$
- לפי דף הנתונים של הטרנזיסטור, יש לו $\beta = 100$ (למעשה, מקדם ההגברה משתנה בין 100 ל-300. בחרנו את הערך 100 כדי לתכנן את המעגל למקרה הגרוע ביותר).
- לכן: $Ib = \frac{Ic}{\beta} = \frac{26 \cdot 10^{-3}}{100} = 26\mu A$
- מפל המתח על נגד הבסיס ברוויה יהיה $V = 16 - 0.7 = 15.3V$
- מכאן התנגדותו של נגד הבסיס: $Rb = \frac{15.3}{26 \cdot 10^{-6}} = 588K\Omega$
- נגדים סטנדרטים שקיימים בשוק הקרובים לערך זה הם $560K\Omega$ ו- $620K\Omega$. כדי להבטיח רוויה של הטרנזיסטור נבחר את הערך הקטן יותר.
- ערכם של הקבל והנגד יקבעו את זמן ההשהיה של המעגל. המעגל יכנס לרוויה כאשר המתח על הקבל יגיע לכ- 0.7 וולט.
- נוסחת הטעינה של הקבל: $Vc = Vo \cdot e^{-\frac{t}{R \cdot C}}$

מתוך ניסיונות שעשינו במעגל קיים צורך של <u>לפחות</u> שנייה אחת בין הפעלת הספקים לבין הופעת מתח ה-16 וולט על הרובוט. לכן כל קבל שערכו גדול מהערך שיחושב יתאים למטרתנו.
נציב בנוסחא הקודמת:

$$0.7 = 15.3 \cdot e^{-\frac{1}{560 \cdot 10^3 \cdot C}}$$

נחלק ב-15.3:

$$0.045 = e^{-\frac{1}{560 \cdot 10^3 \cdot C}}$$

נוציא Ln משני האגפים:

$$-3.084 = -\frac{1}{560 \cdot 10^3 \cdot C}$$

נחלץ את C ונקבל:

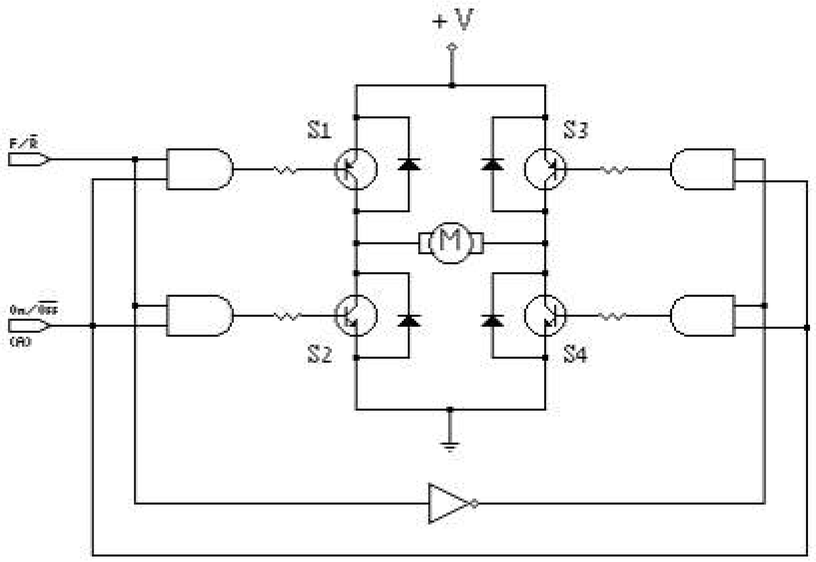
$$\boxed{C = 0.579\mu F}$$

החישובים הנ"ל בוצעו בהנחה שלא קיימים זרמי העמסה של הקבל על ידי צומת בסיס-אמיטר. מסיבה זו וכדי להימצא בזמן טעינה של לפחות 1 שנייה, בחרנו בקבל שערכו פי 10 מהערך המחושב. קבל בעל ערך מעשי שקיים בשוק הוא 6.4 מיקרו פרד (למעשה, לא מצאנו קבל כזה בשוק והשתמשנו בקבל של 10 מיקרו פרד).

<u>כרטיס מנועים</u>

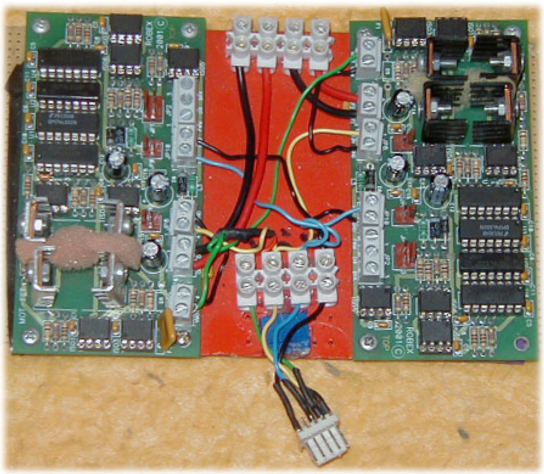
כרטיס המנועים משמש לבקרה על פעולת המנועים. הוא מאפשר קביעה של שלושה ערכים למנועים: כיוון, עוצמה ובלימה. ערכי הכיוון והבלימה נקבעים על-פי שתי כניסות בינאריות, לערך הכיוון 1 מסמל נסיעה קדימה ו-0 אחורה. לערך הבלימה 1 מסמל בלימה ו-0 מסמל מצב רגיל. הבלימה מפעילה כוח מגנטי חזק על המנועים כנגד כיוון הסיבוב.

שליטת הכיוון וההפעלה נעשית באמצעות מעגל המכונה H-bridge. מעגל זה פועל באמצעות ארבעה מפסקים ההופכים את כיוון הזרם בהתאם לערך הכיוון.



סרטוט מעגל H-bridge

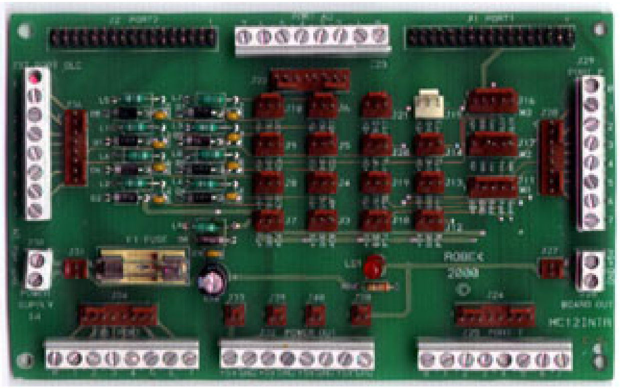
עוצמת המנוע נקבעת לפי ה-PWM שמגיע מהמעבד, כרטיס המנועים ממיר ערך זה למתח שיוצא למנועים.



כרטיסי המנועים על הרובוט

<u>כרטיס ה- Interface</u>

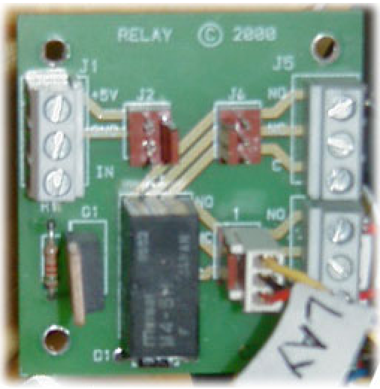
כרטיס הInterface משמש לחיבור נוח של החיישנים והמנועים למעבד, וחלוקת המתחים לחיישנים.
על המעבד ישנם שני פורטים המכונים MCU1 ו- MCU2 מהם המעבד מוציא ומקבל את המידע מהרובוט. חיבור החיישנים והמנועים ישירות לפורטים אלה יהיה מסובך ולא נוח, ובנוסף ייצור בלאגן על הרובוט.
כרטיס הInterface מחובר למעבד באמצעות שני כבלי מידע סטנדרטיים. החיישנים מחוברים לכרטיס הInterface באמצעות מחבר של שלוש חוטים – מתח, Ground ומידע.
בנוסף מספק כרטיס הInterface הגנה למעבד מהמתח שנכנס מהכניסות האנלוגיות, ומייצב את המתח שנכנס לחיישנים.



כרטיס הInterface המצוי על הרובוט

<u>כרטיס Relay</u>

כרטיס ה-Relay משמש להפעלה של המאורר. הכרטיס מכיל ממסר אשר מחבר בין שתי הכניסות המצויות על הכרטיס כאשר הוא מקבל מתח של 5 וולט. בצורה זו ניתן לחבר בין המצבר אשר מפעיל את המאוורר למאוורר עצמו וכך ניתן להפעיל את המאוורר.



כרטיס הRelay המצוי ברובוט

<u>סעיף ג' - בטריות (מצברים)</u>

ברובוט השתמשנו בשני מצברים, אחת למאוורר ואחת לשאר הרובוט. הבטריות שבחרנו להשתמש בהם היו ניקל מטאל-הידריד (Ni-MH), אותם בחרנו בגלל היתרונות שלהם יחסית לסוגי הבטריות האחרות (ראה סעיף – בחירת בטרייה).
המצבר שסיפק חשמל לרובוט הורכב מתריסר בטריות Ni-MH שחוברו בטור. כל תא סיפק 1.2 וולט ו-1800 מיליאמפר-שעה. כך שסך הכול המצבר סיפק 14.4 וולט ו-1800 מיליאמפר-שעה. בפועל המצבר הגיע עד 17 וולט שירדו ל-15.8 וולט כאשר המצבר התרוקן. מצבר זה הספיק ל10-12 ריצות בממוצע.
המצבר שסיפק חשמל למאוורר הורכב מחמש בטריות Ni-MH שחוברו בטור. כל בטרייה סיפקה 1.2 וולט ו-1800 מיליאמפר-שעה. כך שסך הכול המצבר סיפק 6 וולט ו-1800 מיליאמפר-שעה. בפועל המצבר הגיע עד 7 וולט שירדו ל-6 וולט כאשר המצבר התרוקן והספיק ל-50 כיבויים בממוצע.

<u>בחירת בטרייה</u>

בעת בחירת הבטרייה בה משתמשים יש מספר גורמים בהם צריך להתחשב:

- צפיפות אנרגיה והספק: צפיפות האנרגיה נמדדת לפי היחס בין האנרגיה למשקל ולנפח. אנרגיה גבוהה חשובה לנו מכיוון שאנו רוצים שהרובוט יוכל לבצע ריצות רבות בין הטענות. צפיפות הספק נמדדת לפי היחס בין ההספק לנפח ולמשקל. ההספק קובע כמה מהר יכולות הבטריות לפרוק את עצמן, ובעצם את הזרם המכסימלי שהן יכולות לספק. לכן חשובות לנו בטריות בעלות הספק גבוה כדי שנוכל לנצל את מלוא הכוח של המנועים, ושבעת הפעלת המנועים לא יכבה הרובוט מכיוון שלא נשאר מספיק זרם להפעיל את המעבד. ככל שלבטרייה צפיפות אנרגיה והספק משקלית גבוהה יותר כך היא תספק יותר אנרגיה והספק לפחות משקל. ככל שלבטרייה צפיפות אנרגיה והספק נפחית גבוהה יותר כך היא תספק יותר אנרגיה לפחות נפח. מאחר ובבניית רובוט אנו רוצים לבנותו קטן וקל ככל האפשר, אנו נרצה להשתמש בבטריות בעלות צפיפות אנרגיה והספק גבוהות.
- מספר מחזורים: מספר המחזורים קובע את מספר הפעמים שניתן לפרוק ולהטעין את הבטרייה לפני שתשחק וביצועיה ירדו. ערך זה חשוב לנו מכיוון שאנו מנסים את הרובוט פעמים רבות ואנו לא רוצים שכאשר נגיע לתחרות נגלה שהבטרייה מספיקה לפחות זמן, או מספקת פחות זרם או מתח.
- אפקט הזיכרון: בבטריות רבות קיים אפקט הזיכרון. אפקט זה גורם לכך שאם מתחילים לטעון בטרייה כאשר אינה ריקה לחלוטין הופכת נקודת ההתחלה של ההטענה למצב האפס החדש של הבטרייה כך שבעצם הבטרייה מאבדת מיליאמפר-שעה. מאחר ואנו רוצים לדאוג שהבטרייה תיתן תמיד מיליאמפר-שעה מכסימלי אנו מעוניינים בבטרייה ללא אפקט זיכרון.
- התנגדות פנימית: התנגדות פנימית משפיעה על ההספק המכסימלי שהבטרייה מסוגלת לספק, וזאת בגלל הנוסחא $V=\varepsilon-i\cdot r$. (V מייצג את מתח ההדקים ו ε את הכוח האלקטרו מניע של הבטרייה)
- מחיר.
- עקומת פריקה שטוחה: עקומת פריקה מייצגת ירידה במתח של הבטרייה לאורך זמן, כאשר הזרם קבוע. כאשר עקומת הפריקה שטוחה המשמעות היא שהבטרייה מספקת מתח קבוע עד שהיא מתרוקנת לחלוטין. המשמעות היא גם שהבטרייה תספק זרם קבוע כנגד התנגדות קבועה

עד שתתרוקן. גורם זה משני ברובוט מכיוון שהמתחים יוצאים ממיצבים ולכן אינם תלויים בזרם הנכנס.

- קיבול - הקיבול של סוללה מראה כמה זרם היא יכולה לספק במשך שעה – בטרייה שקיבולה אחד אמפר-שעה יכולה לספק זרם של אחד אמפר במשך שעה בטרם תיגמר. החישוב לא תמיד נכון לגבי זרמים שונים, ותלוי גם בתנאי השימוש ובהתנגדות הפנימית של התא.
- התפרקות עצמית – כל בטרייה נוטה לשחרר חלק מהמטען שבה גם ללא חיבור למעגל חשמלי. איבוד המטען לאורך זמן תלוי בסוג הבטרייה ובתנאי הסביבה (טמפרטורה, לחות וכדומה), מסיבה זו מומלץ לשמור סוללות במקרר.

סוג תא	עופרת Pb	ניקל מטאל-הידריד NiMH	ניקל קדמיום NiCa	ליתיום יון Li-ION
מתח - V	1.2*	1.2	1.2	3.6
אנודה \ קתודה	Pb \ PbO	H_2 (metals) \ NiOOH	Cd \ Ni NiO	
צפיפות אנרגיה משקלית – Wh\Kg	37	53	44	130
צפיפות אנרגיה נפחית – Wh\l	120	170	120	300
זרם מקסימלי - A	2	2	5***	1.5
קיבול - Ah	1	1.5	1	3
התנגדות פנימית - mΩ	100	30	10	200
התפרקות עצמית - אחוז לחודש		30	20	10
מחזורים		500	1500	500
זכרון		אין		אין
מחיר	10	20	25	80

*כל המצברים פרט למצבר העופרת מורכבים ממספר תאים המחוברים בטור כדי ליצור מתח גבוה יותר. התאים יכולים לבוא בגדלים שונים לפי תקנים של בטריות (A, AA, AF וכדומה). מצבר העופרת יכול להופיע בצורות ובגדלים שונים. נתוני הטבלה מתייחסים לתא בגודל AA, ולבטריית עופרת המספק מתח דומה.

** חלק מנתוני הטבלה עלולים להיות לא עדכניים עקב קצב ההתקדמות בטכנולוגית התאים, בגלל השוק הצומח של הטלפונים הסלולאריים והמחשבים הניידים.

*** תא ניקל קדמיום מסוגל לספק זרמים גבוהים מאוד בגלל המבנה השכבתי המיוחד של התא.

סעיף ד' - גלגלי תנועה וגלגלי עזר

גלגלי תנועה

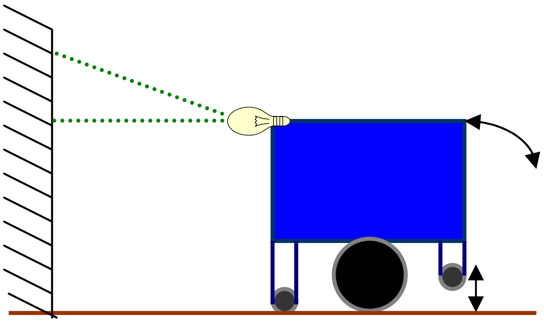
ברובוט "קרפלעך" שני גלגלי תנועה, המניעים את הרובוט קדימה (הנעה דיפרנציאלית אסימטרית).
קוטר גלגל התנועה הוא 8 ס"מ ורוחבו 0.5 ס"מ. הגלגלים מתחברים למנוע באמצעות בורג שנסגר על ציר הגלגל. כדי ליצור מקדם חיכוך גבוה ככל האפשר עם הרצפה. בהיקף הגלגל יש מעין תעלה שבה העברנו צינור גומי (ראה טבלת חיכוך בסוף סעיף זה).

הגלגל בו השתמשנו

גלגלי עזר

כדי לשמור על יציבות הרובוט נדרשים גלגלי עזר. גלגלי העזר צריכים לענות על שתי דרישות:

- לשמור על הרובוט יציב: בזמן נסיעה אסור לרובוט לשנות זווית ביחס לרצפה מכיוון ששינוי זה יביא גם לשינוי בקריאת החיישנים.



- לאפשר עליה וירידה של הרובוט ממכשול (במקרה ומבצעים בונוס זה). הגלגלים צריכים לאפשר לרובוט לעלות על המכשול ולא להיתקע עליו (מצב שיכול להתרחש בהנעה דיפרנציאלית מרכזית כמו של קרפלעך (ראה פרק ב' - שיטות הנעה)). בנוסף הגלגלים צריכים לדאוג שבעת הירידה מהמכשול ישמור הרובוט על יציבות ולא יטה לצד מסוים, דבר שישבש את קריאות חיישני הצד ויגרום לתיקונים בתנועה לפעול הפוך, ולגרום לכך שהרובוט יפגע בקיר.

משקלול של שתי סיבות אלה, ובעקבות ניסויי של שיטות שונות הגענו למערך הבא של גלגלי עזר:
שני גלגלי עזר קדמיים: בקדמת הרובוט ישנם בכל צד שני גלגלי עזר זהים, זאת כדי שבעת הירידה מהמכשול הרובוט לא יינטה לצד, דבר שקרה כאשר ניסינו לשים גלגל עזר יחיד במרכז, וזאת בעקבות האסימטריה במקום הגלגלים. גלגלי העזר הקדמיים הורכבו מטפלון משויף לצורת עיגול, שהורכב על קפיץ שמיקמנו בתוך צינור. הקפיץ נועד לאפשר עלייה של הגלגל בעת עלייה על מכשול בכדי למנוע היתקעות על הגלגל ועצירה של הרובוט בעלייה על המכשול (אם המכשול לא הוצמד היטב לרצפה). הטפלון המשויף שימש אותנו במקום גלגלי מיסב מכיוון שבדיקות הראו שהוא מחליק ביותר קלות (ראה טבלת חיכוך בסוף סעיף זה), וגם לגלגלי מיסב היה חיסרון שגם בזווית קטנה המיסב לא מסתובב, והרובוט נוסע על המתכת.

הגלגלים הקדמיים של הרובוט "קרפלעך"

גלגל עזר אחורי: גלגל העזר האחורי הורכב ממספר ספוגי גומי שהודבקו יחדיו ועליהם הרוכב גלגל מיסב. גלגל העזר האחורי לא נגע ברצפה בעת נסיעה, אלה הורכב כך ש"ירחף" כחצי ס"מ מעל הרצפה, זאת כדי לאפשר עליה בטוחה על המכשול.

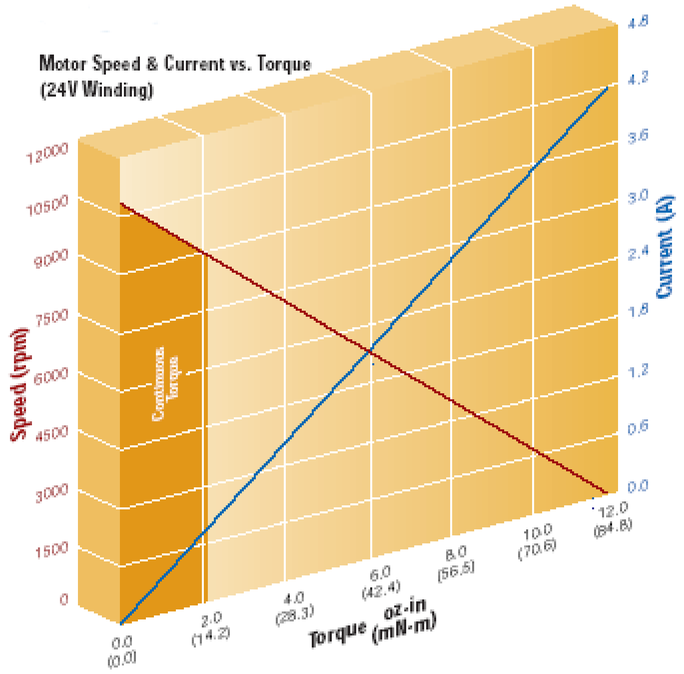
הגלגל האחורי של הרובוט "קרפלעך"

<u>טבלת מקדמי חיכוך</u>

חומר	על משטח	מקדם חיכוך סטטי	מקדם חיכוך קינטי
פלדה	אלומיניום	0.61	0.47
צמיג	כביש (יבש)	1.0	0.8
קרח	קרח	0.1	0.04
גומי	עץ	0.95	0.76
טפלון	פלדה	0.04	0.04

<u>סעיף ה' - מנועים</u>

המנועים מספקים את הכוח שמניע את הרובוט בדירה. אנו השתמשנו במנועים של חברת Pittman מספר סידורי GM874S015.
בגרף אפשר לראות את היחס בין הזרם שנכנס למנוע לכוח המופעל ע"י ציר המנוע (Torque), והיחס בין כוח זה למהירות המנוע (סיבובים לשניה). יותר על משמעות הכוח והמהירות שהמנוע מספק בנספח ג' – מכניקה של תנועות הרובוט.



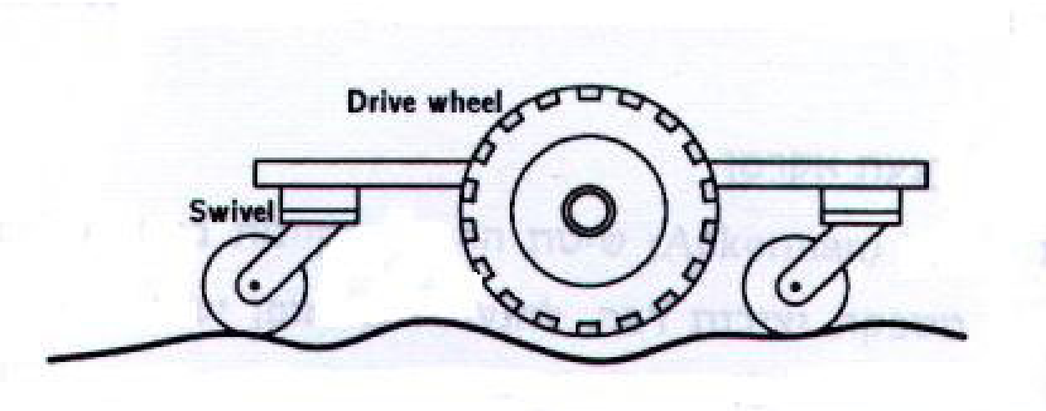
המנועים בהם השתמשנו ברובוט

פרק ב' – שיטות הנעת הרובוט

אחת הדילמות בהן נתקלנו בשלבים הראשונים של בניית הרובוט הייתה בנוגע לשיטת ההנעה.
אופציות רבות עמדו בפנינו, שכן רבים הם סוגי ההנעה האפשריים ברובוט.
להלן פירוט קצר על שיטות ההנעה השונות עליהן חשבנו.

<u>הנעה דיפרנציאלית סימטרית</u>

הנעה דיפרנציאלית היא הפשוטה ביותר למימוש, הן לצוות התוכנה והן לצוות החומרה. משמעות ההנעה הדיפרנציאלית הסימטרית היא הנעה של שני גלגלים המורכבים על ציר משותף (בניגוד להנעה הדיפרנציאלית האסימטרית, בה שני הגלגלים אינם על אותו הציר), וכל גלגל מונע בנפרד על ידי מנוע.
ההנעה הדיפרנציאלית הסימטרית מאפשרת לרובוט לנוע בחופשיות ישר, לעשות פניות בקשת ופניות במקום. רדיוס הסיבוב של פניות הרובוט תלוי בסוג ההנעה הדיפרנציאלית הסימטרית – הנעה אחורית, מרכזית או קדמית (פירוט יבוא בהמשך).
כשמחליטים על בניית רובוט ובו הנעה דיפרנציאלית סינכרונית יש לקחת בחשבון בעיות של איזון שיוצרו, אלא אם נרכיב לרובוט גלגל שלישי, או יותר מגלגל אחד נוסף לשניים המניעים. בלי גלגלי העזר הנ"ל הרובוט עלול להתהפך בפניות, בהאצה או בתאוטה.
את הגלגלים הנוספים ניתן להציב בבסיס הרובוט בצורות שונות – משולש, מעוין או אף צורה בעלת יותר צלעות, ותכנון ההצבה שלהם תלוי בשאלות של מרכז כובד הרובוט, מיקום המנועים וכדומה.
יש לציין שברובוט בעל הנעה דיפרנציאלית סימטרית ובעל שני גלגלים נוספים ויותר, עלול להיווצר מצב בו מכשולים בדרכו של הרובוט מביאים אותו למצב בו הוא תלוי אך ורק על הגלגלים הנוספים, בלי שאף אחד מהגלגלים המניעים נוגע ברצפה, ונתקע במצב זה. ניתן להימנע ממצב זה, אם הגלגלים הנוספים שאנו מציבים בבסיס הרובוט אינם קשיחים, או בעלי מנגנון כלשהו המאפשר להם להתכופף לכיוון הרצוי כך שהרובוט לא יישאר תלוי באוויר עליהם.

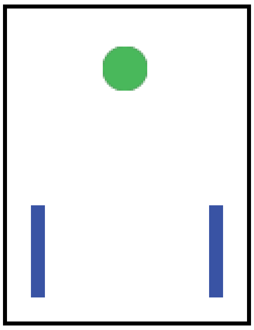


בעיה נוספת העלולה להיווצר ברובוט בעל הנעה דיפרנציאלית סינכרונית היא שאנו חייבים לוודא שהרובוט נוסע ישר. כל הבדל קטן במהירות בין המנועים, או הבדל בקוטר הגלגלים, או אף הבדל בין המשטחים עליהם נוסע כל גלגל, עלול לגרום לרובוט לא לנסוע בקו ישר, ולגרום להתנגשותו בקיר. לכן, דרושים תיקוני כיוון בלתי פוסקים במהלך נסיעת הרובוט, לשם ווידוי שאינו מתקרב יתר על המידה לקירות המבוך.

כאמור, ישנן 3 תת-שיטות להנעה דיפרנציאלית:

<u>הנעה אחורית</u>

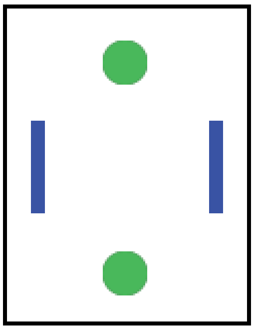
הכוונה בהנעה אחורית היא להציב את ציר הגלגלים המניעים בחלק האחורי של בסיס הרובוט.
לשיטה זה חסרונות ספורים – היא מקשה על הרובוט בעלייה על שיפועים, היא מקשה על תכנון הפניות, משום שהיא יוצרת רדיוס סיבוב גדול יחסית בפניות של הרובוט, ויש סיכון שקדמת הרובוט תיתקע באחד הקירות במהלך הפניות, ומשום שהפניות נעשות על ידי שינוי במהירות הגלגלים המניעים של הרובוט, והם נמצאים באחורי הרובוט, כל שינוי קטן יגרור שינוי גדול יחסית בכיוון התנועה שלו, ותיקון השגיאות במהלך נסיעה בקו ישר (אותו תיקון שהכרחי, כאמור, לכל רובוט בעל מנגנון הנעה דיפרנציאלי), עלול להיות גס מדי וקשה לתכנון.
יתרונות השיטה הם שהסיבובים אותם עושה הרובוט יהיו חלקים בחלקו האחורי, ולכן הסיכוי שייפגע קטן יותר, וכן יש שיאמרו שתיקון השגיאות במהלך נסיעה יהיה קל יותר מכיוון שהשינוי הקל ביותר במהירויות המנועים מספיק כדי לשנות את כיוון הרובוט באופן משמעותי.
באיור ניתן לראות את הגלגלים המניעים (כחול) ואת הגלגל הפסיבי (ירוק).



הנעה אחורית

<u>הנעה מרכזית</u>

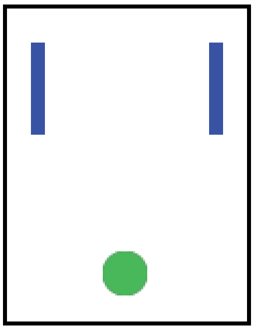
בהנעה מרכזית, הגלגלים המניעים של הרובוט ימוקמו במרכז בסיסו.
חסרונה המרכזי של שיטת הנעה זו נעוץ בעובדה שרק בה עלול להיווצר המצב בו הרובוט "תקוע" באוויר, על גלגליו הנוספים, ואף אחד מגלגליו המניעים אינו נוגע ברצפה.
לשיטה זו גם יתרון גדול – מיקום הגלגלים המניעים במרכז בסיסו של הרובוט נותן לרובוט את היכולת לבצע סיבובים במקום, עם רדיוס סיבוב קטן מאוד. כך ניתן לעצב את תנועתו של הרובוט במהלך פניות כך שיבצע פניות יעילות ויציבות יותר. יתרון זה מקל על ביצוע בונוס הארביטררי ועל בונוס הרהיטים.



הנעה מרכזית

<u>הנעה קדמית</u>

חסרונה המרכזי של שיטת הנעה זו הוא רדיוס הסיבוב הרחב שלה. אם רובוט בעל שיטת הנעה דיפרנציאלית קדמית גדול מידי, חלקו האחורי עלול להיתקע בקיר במהלך פניה.
עם מיקום הגלגלים בחלקו הקדמי של הרובוט, באה גם הבעיה שדרוש שינוי גדול יותר במהירות המנועים כדי לתקן את כיוונו, בניגוד לבעיה ההפוכה בהנעה אחורית.
יתרונותיה של שיטה זו הם עלייה קלה יותר על מכשולים, שכן הגלגלים המניעים ממוקמים בקדמת הרובוט והם הראשונים העולים על המכשול ועוברים אותו, ותכנון קל יותר של כניסה לפניות ויציאה מהן.



הנעה קדמית

<u>הנעה דיפרנציאלית אסימטרית</u>

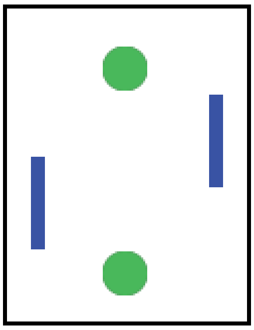
ההנעה הדיפרנציאלית האסימטרית עובדת בדומה לאחותה, ההנעה הדיפרנציאלית הסימטרית, אך בהבדל משמעותי אחד – שני הגלגלים המניעים של הרובוט הבנוי בשיטה זו לא ממוקמים על ציר אחד. פרט להבדל זה אין הבדל בין סוגי ההנעות.

בניגוד למקובל לחשוב, שיטה זו לא יוצרת שום בעיה בתכנון הפניות של הרובוט, והדרך בה הוא פונה זהה לזו של ההנעה הסינכרונית. המקום היחידי בו נתקלנו בהבדל היה בעת ביצוע פניות במקום שדרשו כוח רב יותר מהמנועים.

כשמציבים את המנועים בצורה סימטרית, זה לצד זה, המנועים המורכבים אחד מול השני תופסים מקום רב וגורמים לרובוט להיות רחב יתר על המידה. רובוט בעל הנעה אסימטרית יכול להיות צר עד פי 2 מרובוט אחר, מה שמקל עליו לפנות ולנווט במבוך מבלי חשש לפגיעה בקירות.

גם בשיטת הנעה זו, אנו נדרשים להרכיב גלגלים נוספים לשם ייצוב הרובוט, וגם בה עלול להיווצר מצב בו הרובוט תלוי באוויר.

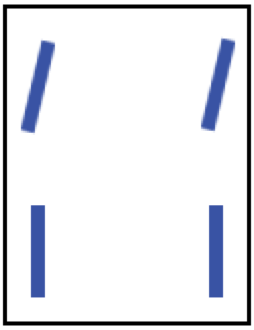
בבניית הרובוט קרפלעך בחרנו בשיטת הנעה יוצאת דופן זו, בשל יתרונותיה בגודל הרובוט וחסרונותיה המועטים.



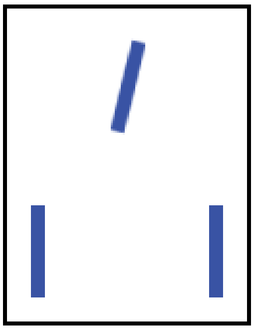
הנעה דיפרנציאלית אסימטרית

הנעת תלת אופן / הנעת מכונית

הרעיון העומד מאחורי הנעת התלת אופן, ומאחורי ההנעה דמוית הנעת המכונית (הקרויה הנעת אקרמן), הוא שני גלגלים מניעים במחוברים למנועים בנפרד, או למנוע אחד, וגלגל אחד להיגוי (במקרה של הנעת תלת אופן) או שני גלגלי היגוי (במקרה של הנעת מכונית).
שתי שיטות ההנעה מספקות יציבות גדולה לרובוט (הנעת אקרמן מספקת יציבות מוחלטת, שכן בהנעה זו הרובוט מורכב על 4 גלגלים היוצרים מלבן).
הנעת התלת אופן פשוטה יותר מהנעת אקרמן לבנייה, משום שבהנעת אקרמן יש לחבר בין שני גלגלי ההיגוי באופן בו הם יזוזו באותה הזווית בדיוק תמיד.
יתרונן של שיטות הנעה שלה הוא שאין צורך לשנות את מהירויות המנועים כדי לבצע פניות או תיקונים, והגלגלים המכוונים עושים את כל העבודה. בנוסף, הצורך בתיקונים במהלך נסיעה בקו ישר קטן יותר, וכל הדרוש על מנת לנסוע בקו ישר הוא לוודא שגלגל/י הכיוון פונים קדימה.
חסרונה העיקרי של שיטה זו הוא רדיוס סיבוב גדול מאוד, המקשה על ביצוע פניות במבוך מבלי להיתקל בקירות.



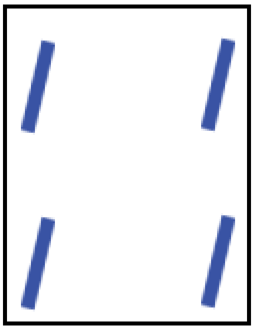
הנעת אקרמן



הנעת תלת אופן

<u>הנעה סינכרונית</u>

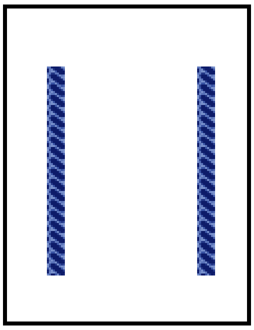
בהנעה סינכרונית הרובוט בנוי על 3 או 4 גלגלים, אשר משמשים אותו הן להנעה והן להיגוי. שלושת, או ארבעת הגלגלים מסתובבים תמיד באותה הזווית וכך גוף הרובוט נשאר תמיד פונה לכיוון מסוים.
אם בוחרים להשתמש בשיטה זו יש למצב את החיישנים על הרובוט בצורה חכמה, משום שגופו של הרובוט לא מסוגל לפנות והוא תמיד פונה לכיוון אחד ויחיד. לכן יש למקם חיישני מרחק בכל צלע של הרובוט, ונוצר הצורך להרכיב את חיישן הPyro על סרבו.



הנעה סינכרונית

<u>הנעת זחל</u>

בהנעת זחל, לרובוט יהיו שני מנועים, המסובבים בעזרת גלגלי שיניים הממוקמים באחורי הרובוט שני זחלים, או רצועות כלשהן שיהיו במגע עם הרצפה. יהיו גם גלגלים פאסיביים בקדמת הרובוט שתפקידם להחזיק את הרצועות מתוחות.
יתרונה של הנעת זחל הוא היכולת לעלות על כל מכשול ולעבור אותו, ורדיוס סיבוב קטן יחסית, משום שהפניות יתבצעו בעזרת שינוי המהירות של המנועים בצדדים הרצויים.
חסרונה העיקרי ש שיטת הנעה זו הוא שהיא מסובכת לבניה מבחינה טכנית, ובנוסף יש לתחזק אותה באופן קבוע, מכיוון שחשוב לשמור על הרצועה מתוחה ונקייה.



הנעת זחל

פרק ג' - מבנה הרובוט

בעת בניית הרובוט ישנם גורמים רבים בהם אלינו להתחשב, חלקם מתנגשים ומניעים זה את זה וחלקם ידרשו שינוי מחולט של כל שאר הרובוט. מאחר ואת הרובוט מתחילים ממצב של חוסר ידע מוחלט, פעמים רבות החלטות שנראו ההגיונית ביותר בהתחלה מתגלות כשגויות לחלוטין בהמשך, לעתים ימצא לבעיות אלה פתרון פשוט, אך לעתים תכופות יותר לא יהיה מנוס משינוי הרובוט לחלוטין ואף בנייתו מחדש. ואכן כמעט כל קבוצת רובונר, בנתה לפחות פעם אחת את הרובוט מחדש.
הגורמים אליהם צריכים להתייחס בעת בניית הרובוט:

- אסטרטגיה: האסטרטגיה היא הגורם החשוב ביותר בעת בניית הרובוט, האסטרטגיה מורכבת מגורמים רבים: איזה בונוסים עושים, איזה מסלול ניווט מועדף, איך נרצה לבצע פניות ועוד אלפי גורמים שונים. בקרפלעך האסטרטגיה השפיעה כמעט על כל פן בבניה: רצינו לחסוך זמן בניווט באמצעות סריקת נר מחוץ לחדרים ולכן הרכבנו שני חיישני UVtron, רצינו לבצע רהיטים ולכן בחרנו בשיטת הנעה דיפרנציאלית מרכזית כדי שנוכל לבצע פניות במקום, וישנן עוד דוגמאות אין-ספור.
- גמישות: מי שחושב שבעת בניית הרובוט הראשון האסטרטגיה עליה מחליטים היא סופית לא עשה פרויקט מימיו. ברגע שהרובוט גמור ומתחילים להריץ עליו את התוכנה מתגלות כל חסרונות האסטרטגיה שנבחרה בתחילה – ובעקבות זאת משתנה האסטרטגיה, ואתה כמובן צריך להשתנות גם הרובוט. בנוסף מתגלות גם טעויות התכנון שגם להן נדרש פתרון ע"י שינוי הרובוט. מסיבות אלה נדרש שלרובוט תהיה גמישות מבנית רבה ככל האפשר – שטח פנוי להוסיף כרטיסים וחיישנים, אפשרות להחליף גלגלי עזר, אפשרות להוסיף או להוריד בטריות מהרובוט. את קרפלעך ניסינו לבנות קטן ככל האפשר – מה שמנע את האפשרות לגמישות מבנית גבוהה, ואכן חסרון זה פגע בנו לפני התחרות הארצית כאשר גילינו שאין באפשרותנו להוסיף בטרייה למאוורר במבנה הצפוף (פירוט נוסף בנספח ה' – בעיות).
- נגישות: חשוב מאוד שתהיה נגישות גבוהה לכל חלקי הרובוט זאת מכיוון שחלקים יתקלקלו ויהיו חלקים שנרצה להחליף.

מבנה הרובוט מורכב ממספר גורמים המשפיעים ומגבילים זה את זה:

- צורות המבנה: צורת המבנה בדרך כלל מופיעה באחת משתי צורות: עיגול או מרובע. היתרון בצורה עגולה הוא שאין אפשרות שהרובוט יתקע בקיר או בפינה, הצורה העגולה תגרום לכך שהרובוט פשוט "יחליק" לאורך הקיר ולא יתקע לחלוטין. לצורה העגולה גם יתרון כאשר מבצעים פניות במקום (דורש שיטת הנעה מתאימה – ראה פרק ב' – שיטות הנעה), הצורה העגולה תבטיח שכאשר הרובוט מבצע פניות במקום הוא לא יפגע בשום דבר. יתרון הצורה המרובעת הוא בכך שאין בזבוז של מקום – רוב החלקים שנרכיב על הרובוט יופיעו בצורה מרובעת (כרטיסי אלקטרוניקה, בטריות), ולכן בצורה העגולה יישארו שטחים ריקים. חסרון זה מהווה גם יתרון מאחר והוא מעלה את גמישות מבנה הרובוט.
- גדלים: לרובוט ישנם שלושה גדלים עיקריים: רוחב, אורך וגובה.

ככל שרוחב הרובוט קטן יותר, כך תהיה עבודתו של צוות התוכנה קלה יותר. רובוט צר אומר שהתיקונים בתנועה יכולים להיות פחות מדויקים, שאפשר לעשות פניות רחבות יותר ולא מדויקות ושהכניסה לחדר תהיה קלה יותר. יתרון זה נובע מכך שאם נקבע מרחק בטיחות של 5

ס"מ מקירות המבוך, רובוט שרוחבו 18 ס"מ יוכל לסטות עד 9 ס"מ מהאמצע לכל צד בעוד שרובוט שרוחבו 24 ס"מ יוכל לסטות רק 6 ס"מ לכל צד.
אורך הרובוט משפיע בעיקר על ביצוע של פניות צרות ופניות במקום, רובוט שאורכו גדול יחסית לרוחבו עלול לפגוע בקירות וברהיטים בעת ביצוע פניות.
גובה הרובוט, למרות הדעה הרווחת אינו קובע את נטייתו של רובוט להתהפך – תכונה זו נקבעת מגובהו של מרכז המסה בלבד. המקרה היחידי בו משפיע גובה הרובוט הוא כאשר ממקמים את החיישנים גבוה, ואז בעת עליה על מכשול עלולים החיישנים הקדמיים לראות מעל לקיר הזירה, מה שעלול להוביל לביצוע מאוחר של פניה.

- שיטת הנעה: מפורט בפרק ב' - שיטות הנעה.
- מערך כרטיסי אלקטרוניקה: את כרטיסי האלקטרוניקה מומלץ לשים כך שתהיה אפשרות להוציא ולהחליף אותם בקלות במקרה של תקלה. בנוסף ישנם כרטיסים שמומלץ מאוד שתהיה אליהם גישה נוחה, כמו כרטיס הInterface, שדרכו עוברים רוב חיבורי הרובוט,
- מערך חיישנים: מערך החיישנים על הרובוט תלוי בעיקר באסטרטגיה, אך צריכים לדאוג שלא יהיו גורמים שיפריעו לקריאת החיישנים כמו חוטים או קרבה למנועים (הפרעות מגנטיות).

<u>עצות על מיקום חיישנים</u>

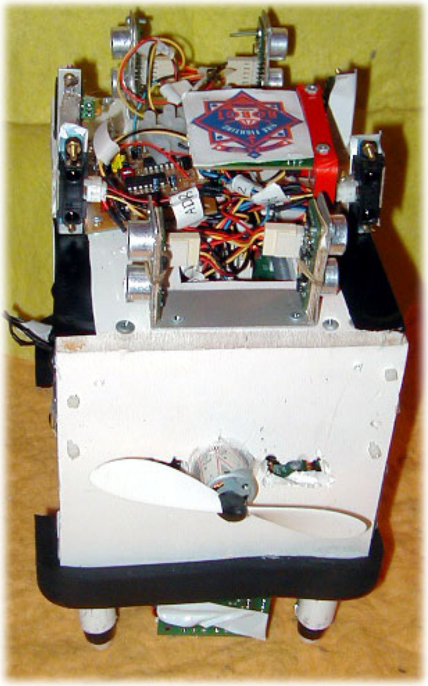
בחרנו להשתמש בשישה חיישני מרחק ברובוט אותו בנינו.
שני חיישנים אינפרא אדומים המכוונים קדימה ושני חיישנים אולטרא-סוניים בכל צד של הרובוט. מכיוון שטווח המדידה של חיישני האינפרא אדום מתחיל מעשרה ס"מ, לא הצבנו אותם בקדמת הרובוט, אלא במרכזו, כשהם פונים קדימה, כדי שיוכלו לספק קריאות מהימנות גם כשהרובוט קרוב מאוד לעצמים.
בזמן הרכבת חיישני מרחק יש לקחת בחשבון גורמים ספורים -

- הם צריכים להיות ממוקמים במקום בו יש סיכוי מינימאלי שחוטים וגורמים אחרים יעמדו מולם בזמן ריצת הרובוט.
- מכיוון שחיישני מרחק נוטים להיתקל בתקלות לעתים תכופות, יש להציב אותם במקומות ברובוט בהם יהיה נוח להוציא אותם, ולהכניס אותם מחדש בקלות.
- קריאות החיישנים אופטימאליות כאשר הם מאונכים למשטח עליו הם מורכבים ולא צמודים אליו במאוזן, משום שכאשר הם צמודים אל המשטח, חלק מאלומות האור שהם שולחים (או גלי הקול במקרה של החיישנים האולטרא-סוניים) נחסם על ידי המשטח.
- כשמציבים שני חיישנים הפונים לאותו הכיוון, כמו שמומלץ לעשות באם רוצים להשתמש בהם לתיקונים כשהרובוט נמצא מול קיר או במקביל לקיר, יש לוודא שמציבים אותם אחד ליד השני, על אותו הציר, ובלי זווית ביניהם. שניהם חייבים להיות באותה הזווית בדיוק ביחס למשטח עליו הם מוצבים וביחס לעצם העומד מולם לשם קבלת קריאה אופטימאלית.
- מומלץ לוודא שחיבור ה Crimping המחובר אל החיישן מחובר היטב, משום שלאחר זמן מה של שימוש, ומספר היתקלויות בקירות (דבר שצפוי שיקרה במספר לא מבוטל של ריצות של הרובוט בתחילת דרכו), החיבור עלול להתרופף ולשבש את האות אותו מחזיר החיישן. ברובוט שלנו השתמשנו בדבק חם כדי לוודא שהחיבור לא יתרופף.

פרק ד' - מבנה הרובוט "קרפלעך"

<u>צורה וגודל</u>

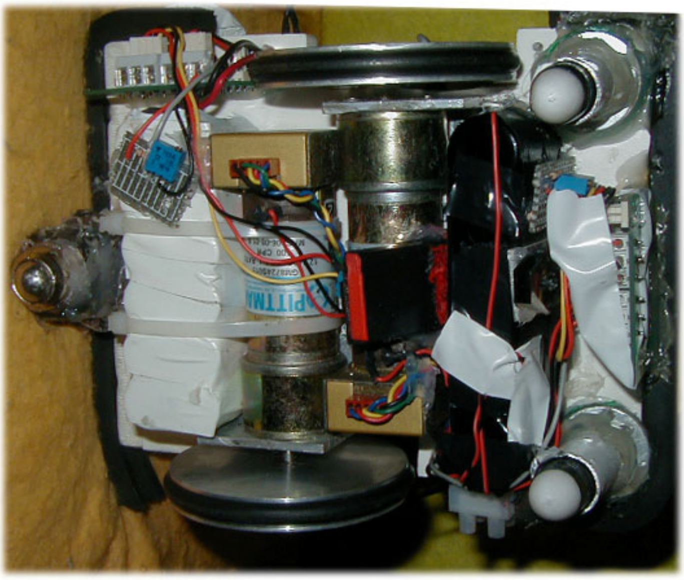
צורתו של קרפלעך היא תיבה: רוחבו 15 ס"מ, אורכו 19.5 ס"מ וגובהו 26 ס"מ. אנו בחרנו בצורה זו ובמידות אלה מכיוון שרצינו לבנות רובוט קטן ככל האפשר (יתרונות וחסרונות בפרק ג' – מבנה הרובוט). בחינה מדוקדקת של הרובוט תראה שאין כמעט ס"מ אחד מבוזבז על קרפלעך.



הרובוט "קרעפלך" – מבט עילי מלפנים

<u>שיטת הנעה:</u>

אנו בחרנו בשיטת הנעה דיפרנציאלית אסימטרית מרכזית, זאת מכיוון שרצינו את יתרונות ההנעה הדיפרנציאלית המרכזית אך רצינו גם לצמצם את רוחבו של הרובוט ככל האפשר. (הרחבה בפרק ב' – שיטות הנעה). פירוט על מערך גלגלי העזר בפרק א' – פירוט חלקי הרובוט.

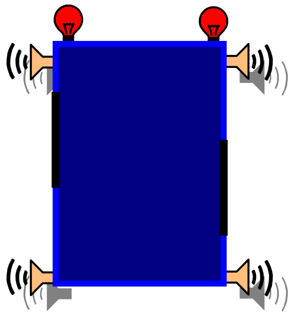


הרובוט "קרפלעך" – מבט מעין הנמלה

<u>מערך חיישנים:</u>

<u>חיישני מרחק:</u>

לרובוט קרפלעך שישה חיישני מרחק, שני חיישני אינפרה אדום שפונים קדימה ועוד שני זוגות של חיישנים אולטרה סוניים שפונים לצדיים.
החיישנים הקדמיים מודעים לרובוט מתי לפנות במסדרון ומסייעים לעקיבה לאורך קיר בחדרים.
חיישני הצד מספקים את המידע לצורך ביצוע תיקונים בנסיעה ותיקונים במקום, וגם מסייעים לרובוט להתמצא בדירה.



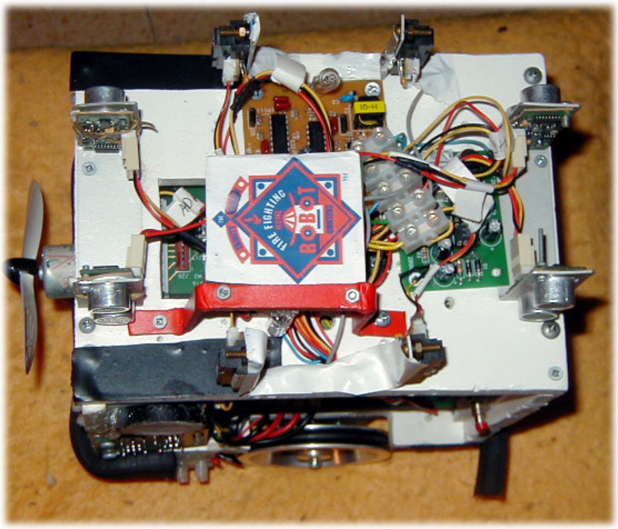
איור אילוסטרציה של הרובוט "קרפלעך"

<u>חיישנים לגילוי נר:</u>

לקרפלעך שני חיישני UVtron הממוקמים בשני צדדיו, שני חיישנים פירואלקטרים שממוקמים בשני צדדיו ושני חיישני פס-לבן הממוקמים בתחתית הרובוט אחד מאחורי השני.

חיישני ה- UVtron מזהים האם בחדר יש נר או לא, אנו מיקמנו שניים כדי שנוכל לזהות נר ללא כניסה לחדר וכדי לחסוך זמן בסריקות.
חיישנים פירואלקטרים מזהים נר כאשר הם חולפים על פניו, בדרך כלל ממקמים חישן זה בקדמת הרובוט ומבצעים סריקה בחדר אחר הנר. אנו בחרנו אסטרטגיה שונה לחיפוש נר בחדר, הרובוט נוסע לאורך הקירות עד שהוא מגיע לנר, או עד שהוא רואה נר בחיישן הפירואלקטרי הצדדי שלו ופונה אל הנר.

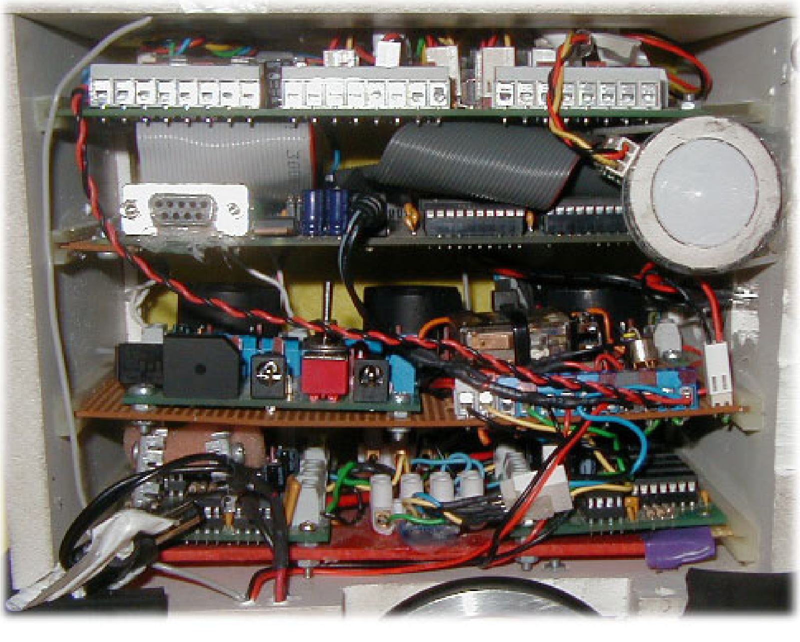
חיישני פס-לבן מודעים לרובוט כאשר הוא נכנס לחדר וכאשר הוא מגיע לנר, אנו הצבנו שניים כדי שיהיה גיבוי למקרה שחישן אחד יפספס.



הרובוט "קרפלעך" – מבט מעין הציפור

<u>מערך כרטיסי אלקטרוניקה:</u>

אנו מקימנו את רוב כרטיסי האלקטרוניקה במגרות, זאת כדי שתהיה גישה נוחה אליהם ונוכל לשלוף אותם בקלות. הגישה לכרטיס הInterface מתאפשרת באמצעות "הגג הנפתח" (חלון שחתכנו בגג הרובוט). את כרטיסי האלקטרוניקה של מיקמנו במגירות הברגנו במקומות שונים על הרובוט.



מבט פרופיל על "קרפלעך"

חלק ב' – התוכנה

תוכנת הרובוט נכתבה בשפת סף בשם "אסמבלי", שהיא השפה הקרובה ביותר לשפת המעבד איתו עבדנו, ה68HC12B32.

את התוכנה כתבנו בממשק (IDE) שהכינה החברה שיצרה את המעבד, Axiom. בעזרת ממשק זה היה ניתן בקלות לכתוב את התוכנה, להדרה ולהריצה.

יש לציין שאת כל הידע התכנותי רכשנו משני ספרים בהוצאת מוטורולה שמלווים את המעבד: ספר הפקודות (הספר הסגול) – Reference Manual, וספר המידע על המעבד (הספר הלבן) – Advance Information.

בעמודים הבאים נסביר כיצד תפעלנו את חלקי הרובוט כגון מנועים וחיישנים וכיצד התגברנו על בעיות בהן נתקלנו במהלך תכנות הרובוט. חשוב לציין שההסברים מפורטים ומיועדים לאנשים בעלי ידע בסיסי באסמבלי אשר תכנתו רובוטים בעבר או לאנשים אשר מעוניינים לבנות רובוט בעל יחידת עיבוד מרכזית של מוטורולה ממשפחת הHC12.

פרק ה' - מבנה התוכנה

התוכנה מורכבת ממספר חלקים עיקריים, אותם פיצלנו לקבצים נפרדים על מנת שיהיה לנו יותר קל לערוך ולטפל בחלקים השונים של התוכנה. הקבצים השונים אוחדו לתוכנה המרכזית באמצעות הפקודה Include אשר מצרפת קובץ אסמבלי (.asm) לתוכנה בה הפקודה נרשמת. הפקודה היא למעשה פקודת Pre-Compiler – כלומר היא מתבצעת לפני ההידור המלא של התוכנה. המהדר מהדר את התוכנה כך שהקבצים מופעים במקום בו קוראת להם פקודת ה-Include.

לדוגמא:

#Include c:\ax12\includes\equates.asm

כך ניתן לצרף את הקובץ equates לתוכנה הראשית.
יתרונות של הפקודה Include: ניתן לחלק את התוכנה לחלקים רבים, מה שמקל על כתיבתה.
חסרונות של הפקודה Include: כאשר משנים דבר בקובץ אשר תוכנות רבות מבצעות עליו את פקודת הIncludeה, תוכניות שלמות עלולות להשתבש עקב השינוי. לכן יש לשים לב לשינויים אותם מבצעים בקבצים אלו.

חלקי התוכנה שלנו הם כדלהלן (על כל חלק יפורט בהרחבה בהמשך):

- Equates – החלק בו מוגדרות הכתובות בזיכרון בתור שמות אנגליים, וכן גם קבועים (Consts).
- Inits – קטע איתחולים בו מאפסים משתנים גלובליים, אתחול מנועים, חיישנים ופסיקות (Interrupts).
- Tikunim – חלק זה מבצע את התיקונים במהלך הניווט, בו כלולים תיקונים במקום ותיקונים תוך כדי נסיעה.
- PID – בקטע זה מצוי חלק התוכנה שמבצע בקרת מהירות בשיטת PID, עליה מפורט בנספח ז'. בעקבות התנגשות של חלקים שונים בתוכנה, לא הרצנו את חלק זה בתחרות.
- Sensors – בקובץ זה ישנן פונקציות אשר אחראיות על תפעול חיישנים, כגון חיישנים אולטרא-סוניים, חיישני מרחק אינפרא אדומים (IR), חיישן Uvtron אשר רגיש לאור אולטרא סגול, חיישן פס לבן וחיישן הPyro שאחראי על מציאת הנר כאשר הרובוט נמצא בתוך החדר.
- Fire – חלק זה אחראי על מציאת וכיבוי הנר לאחר זיהוי החדר בו ממוקם הנר בניווט.
- Motors – יחידה זו אחראית על תפעול המנועים, בה ניתן למצוא פונקציות כמו סיבוב תוך כדי נסיעה לפי Encoders, סיבוב במקום, עצירות חזקות וחלשות, פרימת חוט המתח וכדומה.
- HelpFunc (Help Functions) – קטע זה מכיל פונקציות בסיסיות שכתבנו על מנת להקל את העבודה בשאר התוכנה. בקטע זה פונקציות כמו ערך מוחלט, חיסור X,Y, ממוצעים שונים, לולאות השהייה וכדומה.
- Interrupts – בחלק זה ניתן למצוא את הפונקציות שמתבצעות כאשר ישנה פסיקה כגון חיפוש קיר קדמי, חיפוש נר ופס לבן וכניסה לחדר לאחר זיהוי נר על ידי חיישן הUvTron.
- Print – בחלק זה ישנן פקודות הקשורות להדפסת תוכן אוגרים של המעבד על המסך, חלק זה לא נכלל בתוכנה הסופית מכיוון שהוא משמש בעיקר לניפוי שגיאות (Debugging).

יחד עם זאת כתבנו תוכנה עיקרית אשר מכילה את חלק הניווט ואת החלקים שצוינו לעיל. חלק הניווט אחראי על תנועת הרובוט בין החדרים וחיפוש אחר החדר בו ממוקם הנר.

פירוט חלקי התוכנה:

הEquates:

חלק הEquates משמש אותנו להגדרת מקומות בזיכרון בתור שמות שניתן לזכור. לדוגמא - ניתן לקרוא לכתובת $100 בשם Motor1 וכך יותר קל לכתוב תוכנית אשר משתמשת במקום זה בזיכרון. כך, אם נרצה להעביר את תוכן האוגר A לתוך מקום זה בזיכרון נצטרך לכתוב את הפקודה:

STAA Motor1

וזאת במקום הפקודה:

STAA $100

דבר זה אולי נראה שולי אך במעבד זה ישנן בקירוב 500 כתובות אליהן אנו רוצים לגשת והגישה אליהן תהיה קלה יותר אם נקרא להן בשמות שאנו מכירים או שמות שמוכרים על ידי הספרים שמסופקים על ידי מוטורולה.

הגדרת המקום בזיכרון מתבצעת בצורה הבאה:

<Name>: EQU $<Address Number>

כאשר Name הוא השם שרוצים לתת לכתובת הזיכרון ו - Address Number מייצג את כתובת הזיכרון.

בחלק הEquates ניתן גם להגדיר קבועים.
הגדרת הקבועים מתבצעת בשיטה דומה לשיטה הקודמת:

<Name>: EQU #<Value>

כאשר Name הוא שם הקבוע וValue הוא ערכו. ערך יכול להיות דצימלי, בינארי או הקסאדצימאלי, כאשר על מנת לקבל ערך הקסאדצימאלי יש להוסיף $ לאחר הסולמית ועל מנת לציין ערך בינארי יש להוסיף % לאחר הסולמית.

<u>הInits:</u>

כשמו כן הוא, חלק זה תפקידו לבצע איתחולים (Initializations) במעבד למספר תחומים:

- איתחול A2D (Analog To Digital) – באיתחול זה אנו מגדירים אילו יציאות רכיב ה a2d צריך להמיר.
- איתחול PWM (Pulse Width Modulation) – הגדרה של הDuty Period ושל ה Duty Cycle.
- איתחול פסיקות – הגדרת הפורטים והמצבים שבהם יתבצעו פסיקות, והגדרת הפונקציות אליהן יש להגיע כאשר מתבצעת פסיקה.
- איתחול השעון (טיימר) של המעבד – אם מסיבה כלשהי השעון מופסק, יש להפעילו על מנת שיהיה אפשר לבצע פסיקות כל זמן קצוב.
- איתחול IC (Input Capture) – איתחול היציאות הדיגיטליות שנמצאות על המעבד. באיתחול זה ניתן להגדיר אילו יציאות יוגדרו בתור קלט ואילו בתור פלט. בנוסף נגדיר גם מה אנו מחפשים כקלט מהיציאה הדיגיטלית (גל עולה, גל יורד, אף אחד או שניהם).
- איתחול PDLC (Port Data Link Communications) – באיתחול זה מוגדר אילו יציאות ייחשבו כיציאות פלט ואילו יציאות יוגדרו כיציאות קלט.

<u>הTikunim:</u>

ביחידה זו כלולים תיקונים במקום ותיקונים תוך כדי נסיעה. התיקונים במקום כוללים תיקונים לפי חיישנים אולטרא-סוניים ימניים ושמאליים (בהתאם לקיר לפיו אנו רוצים לתקן). התיקונים במקום כוללים תיקונים לפני אולטרא-סוניים ימניים ושמאליים וגם תיקונים לפני חיישני אינפרא-אדום קדמיים ואחוריים.

<u>הSensors:</u>

ביחידת החיישנים יש מספר פונקציות שמקלות על העבודה בתוכנה. היחידה מחולקת לחמישה חלקים:

חיישנים אולטרא-סוניים – ניתן להריץ ארבע פונקציות שקוראות מכל חיישן בנפרד. התוצאה מאוחסנת בזיכרון בשם us_read.

חיישנים אינפרא-אדומים – בחלק זה קיימת הפונקציה Read_ATD שבאמצעותה אנו נותנים הוראה לרכיב הATD להמיר אות דיגיטלי לאנלוגי. חלק זה גם כולל ממוצעי חיישנים קדמיים ואחוריים.

חיישן פס לבן – בחלק זה ישנה הפונקציה Pas_Lav שתפקידה להחזיר אמת או שקר בנוגע להימצאות פס לבן.

חיישן UvTron – קטע זה בודק את שני חיישני הUvTron להימצאות נר בחדר.

חיישן הPyro – קטע זה בודק האם הPyro ראה את הנר או לא והוא מחזיר אמת או שקר בהתאמה.

הMotors:

חלק זה מחולק לשלושה חלקים עיקריים: חלק התאוצה, חלק הפנייה וחלק הבלימה.
חלק התאוצה – חלק זה מורכב ממספר פונקציות כגון Engage, Q_Engage ו Accelerate שתפקידן הוא להתחיל נסיעה ולהתגבר על הבעיה שהרובוט מתחיל לנסוע לא ישר. הפונקציה Accelerate מיועדת להאיץ לאט, כדי שהרובוט לא יחליק בתחילת הנסיעה.
חלק הפנייה – בחלק זה יש פונקציות כמו Trn_r_f, Trn_r_l שתפקידן לפנות ימינה או שמאלה לפי הEncoders. הפונקציה מקבלת מראש משתנה שהוא מספר הEncoders הרצוי.
חלק הבלימה – קטע זה אחראי על בלימת הרובוט כאשר ישנן מספר צורות בלימה: בלימה חזקה, בלימה בינונית ובלימה חלשה.

הHelpFunc:

יחידה זו כוללת פונקציות שימושיות שעוזרות לנו לכל אורך התוכנה. בתוך יחידה זו ישנן פונקציות כמו Dec(X,Y) שמפחיתה את X מ Y, פונקציות שמבצעות Absolute (ערך מוחלט) על אוגרים, לולאות השהייה שמקבלות פרמטרים חיצוניים על מנת לעכב תהליכים, ממוצעים של 8 ו 16 ביט ועוד.

הFire:

קטע הכיבוי מחולק לארבעה חלקים: חלק הפתיחה, תפעול חיישני זיהוי נר, חלק העקיבה וחלק הכיבוי עליהם יפורט בהמשך.

פרק ו' - Interrupts – פסיקות בתוכנה

תיאור כללי

הפסיקה היא קטע תוכנית אשר רץ רק כאשר אירועים מוגדרים מראש קורים. הייחודיות של הפסיקה היא בכך שהמעבד מפסיק את כל הפעולות האחרות שהוא מבצע וניגש להריץ את קטע התוכנית המוגדר כפסיקה. לאחר סיום הקטע המעבד חוזר לנקודה בה הפסיק.
הפסיקות במעבד 68hc12 מתחלקות לשני חלקים עיקריים. האחד, פסיקה הרצה כל פרק זמן קבוע (8.192 מילי שניות), והשני, פסיקה הרצה בהתאם לקלט או פלט מסוים המתרחשים במעבד והוגדרו מראש על ידי המתכנת.

לפני שרוצים להגדיר פסיקות בתוכנה, מומלץ להכיר את האוגרים הבאים:

TMSK1 – מסכה שבאמצעותה ניתן להגדיר איזה כניסות מPortT יוגדרו כInterruptible.
TMSK2 – ניתן להעביר 1 לביט השמאלי ביותר של אוגר זה ואז תתבצע פסיקה כל גלישה של השעון הפנימי של המעבד, כלומר כל 8 מיקרו שניות.
TFLG2 – זהו דגל שהביט השמאלי שלו "עולה" כאשר מתקיימת גלישה בשעון המעבד.

יחד עם זאת, קיים דגל במעבד המסומן באות "I" שכאשר הא מוגדר כ-1 אין פסיקות וכאשר הוא מוגדר כ-0 הפסיקות מופעלות. שינויי המצב בדגל זה מבוצע על ידי שתי הפקודות והן CLI וSEI.

CLI – Clear Interrupt Mask Bit – פקודה זו מאפסת את דגל הפסיקה כלומר מפעילה את כל הפסיקות.

SEI – Set Interrupt Mask Bit – פקודה זו מעלה את דגל הפסיקה כלומר מפסיקה את כל הפסיקות.

כדי שהמעבד ידע איזה קטע תוכנית צריך להריץ יש לסמן את קטע זה בתווית ובסופו לאפס את הגדל המתאים לפסיקה שבוצעה (במקרה וזוהי פסיקה PortT יש לאפס את TFLG1 במקום המתאים ואם זוהי פסיקת גלישה יש לאפס את TFLG2 במקום המתאים) ולהוסיף את הפקודה RTI (Return From Interrupt) כדי שהמעבד ידע שפעולת הפסיקה הסתיימה.
למעבד יש כתובות מוגדרות מראש בזיכרון בהן יש לשים את כתובת הקטע אליו אנו רוצים שהמעבד יקפוץ ברגע הפסיקה (וקטור הפסיקה).

Vector Address	Interrupt Source	CCR Mask	Local Enable		HPRIO Value to Elevate to Highest I Bit
			Register	Bit(s)	
$FFFE, $FFFF	Reset	None	None	None	—
$FFFC, $FFFD	COP clock monitor fail reset	None	COPCTL	CME, FCME	—
$FFFA, $FFFB	COP failure reset	None	None	COP rate selected	—
$FFF8, $FFF9	Unimplemented instruction trap	None	None	None	—
$FFF6, $FFF7	SWI	None	None	None	—
$FFF4, $FFF5	XIRQ	X bit	None	None	—
$FFF2, $FFF3	IRQ	I bit	INTCR	IRQEN	$F2
$FFF0, $FFF1	Real-time interrupt	I bit	RTICTL	RTIE	$F0
$FFEE, $FFEF	Timer channel 0	I bit	TMSK1	C0I	$EE
$FFEC, $FFED	Timer channel 1	I bit	TMSK1	C1I	$EC
$FFEA, $FFEB	Timer channel 2	I bit	TMSK1	C2I	$EA
$FFE8, $FFE9	Timer channel 3	I bit	TMSK1	C3I	$E8
$FFE6, $FFE7	Timer channel 4	I bit	TMSK1	C4I	$E6
$FFE4, $FFE5	Timer channel 5	I bit	TMSK1	C5I	$E4
$FFE2, $FFE3	Timer channel 6	I bit	TMSK1	C6I	$E2
$FFE0, $FFE1	Timer channel 7	I bit	TMSK1	C7I	$E0
$FFDE, $FFDF	Timer overflow	I bit	TMSK2	TOI	$DE
$FFDC, $FFDD	Pulse accumulator overflow	I bit	PACTL	PAOVI	$DC
$FFDA, $FFDB	Pulse accumulator input edge	I bit	PACTL	PAI	$DA
$FFD8, $FFD9	SPI serial transfer complete	I bit	SP0CR1	SPIE	$D8
$FFD6, $FFD7	SCI 0	I bit	SC0CR2	TIE, TCIE, RIE, ILIE	$D6
$FFD4, $FFD5	Reserved	I bit	—	—	$D4
$FFD2, $FFD3	ATD	I bit	ATDCTL2	ASCIE	$D2
$FFD0, $FFD1	BDLC	I bit	BCR1	IE	$D0
$FF80–$FFC1	Reserved (not implemented)	I bit	—	—	$80–$C0
$FFC2–$FFC9	Reserved (implemented)	I bit	—	—	$C2–$C8
$FFCA, $FFCB	Pulse accumulator B overflow	I bit	PBCTL	PBOVI	$CA
$FFCC, $FFCD	Modulus down counter underflow	I bit	MCCTL	MCZI	$CC
$FFCE, $FFCF	Reserved (implemented)	I bit	—	—	$CE

טבלת וקטורי הפסיקות של המעבד 68HC12B32

כעת נדגים שימוש בפסיקה בה נרצה להגדיל מונה כל קליטת גל עולה בPortT0 (PT0). לכן נריץ את קטע הקוד המתחיל בכתובית INT_DEMO:

```
INT_INIT:
       CLR  MONE                 ; Clear the counter
       MOVB #$01,TCTL4           ; Set the input capture parameters
       MOVB #$01,TFLG1           ; Clear the flag
       LDX  #INT_DEMO            ; Load the label address into x
       STX  CH0_INT              ; Store the memory address in the
                                 ; specified address
       MOVB #$01,TMSK1           ; Set the interrupt mask to PT0
       CLI                       ; Enable interrupts
       …
       …

INT_DEMO:
       INC   MONE                ;  Increase the counter by 1
       MOVB  #$01,TFLG1          ;  Clear the appropriate flag
       RTI                       ;  Return from the interrupt
```

<u>פסיקות בתוכנה</u>

בקטע זה נתאר כיצד השתמשנו בפסיקות בתוכנה. קודם כל נעבור על המקומות בהם השתמשנו בפסיקות:
ניווט – בדיקת קיר קדמי.
ניווט – בדיקת חיישן Uvtron למציאת נר.
כיבוי נר – בדיקת פס לבן וחיישן Pyro.

קטע בדיקת הקיר הקדמי מוגדר כבר בתחילת התוכנית בקובץ Inits :

```
INT_INIT:
           MOVB #$00,TMSK1     ; Clear interrupts for PortT
           MOVB #$80,TMSK2     ; Set overflow interrupt
           LDX  #UV_INT        ; Load Uvtron interrupt mem. location
           STX  INT_CH2        ; Put memory location in channel 2
           LDX  #UV_INT        ; Load Uvtron interrupt mem. location
           STX  INT_CH3        ; Put memory location in channel 3
           LDX  #T_OVF         ; Load timer overflow subroutine adr.
           STX  INT_OVF        ; Put timer ovf. Subroutine in debug12
                               ;                              vector
           MOVB #$80,TFLG2     ; Clear overflow flag
           SEI                 ; Stop all interrupts
           RTS
```

כפי שניתן לראות, אנו מאפשרים פסיקת גלישה ובנוסף גם מגדירים את פסיקות הUvtron, אך לא מפעילים אותן עדיין. יחד עם זאת אנו מפסיקים את כל הפסיקות ומאפשרים אותן רק בתחילת הניווט, זאת כדי למנוע בעיות מיותרות.

ועכשיו נראה את הפונקציה T_OVF שרצה בכל גלישת שעון מעבד:

```
T_OVF:
            PSHD                    ; Push registers and accumulators
            PSHX
            PSHY
            PSHC

SIUM:
            LDX   #$28              ; Times to check IR Sensors
CHK_WALL:                           ; Check for front wall
            JSR    READ_ATD         ; Read from left forward IR
            LDAB   ADR0H
            CMPB   WALL             ; Compare to "near wall" value
            BMI    SIOM
            JSR    READ_ATD         ; Read from right forward IR
            LDAA   ADR1H
            CMPA   WALL             ; Compare to "near wall" value
            BMI    SIOM
            DEX
            BNE    CHK_WALL

NEARWALL:
            MOVB   #$FF,PLZ_TURN  ; Set the "Wall is near" flag
SIOM:
            MOVB   #$80,TFLG2     ; Clear the overflow flag
            PULC                   ; Pull registers and accumulators
            PULY
            PULX
            PULD
            RTI                    ; Return from interrupt
```

נסביר בקצרה על הפרוצדורה שלהלן. הפרוצדורה מקבלת משתנה X שבמקרה זה ערכו הוא 28, והוא קובע את מספר הפעמים שבהן חיישני האינפרא אדום יבדקו את הקיר (בבסיס 16). לאחר מכן אנו עוברים לקריאות חיישנים כאשר קוראים כל פעם מחיישן נפרד. כפי שניתן לראות החיישנים מחוברים לכניסת הa/d במקומות Adr0 ו- Adr1. על מנת שנזהה קיר נצטרך לעבור על כל חיישן 28 פעמים ולבדוק שהערך שהוא נותן באמת מראה שהקיר קרוב יותר או שווה לערך שנתנו בתוך תוכנת הניווט (זאת מכיוון שזמן התעדכנות החיישן הוא 33 מילי שניות ורצינו לוודא שנקבל לפחות 2 קריאות שונות). במידה והדבר קרה, נעביר לדגל (משתנה בזיכרון) שנקרא בשם PLZ_TURN (Please Turn) סימן לכך שצריך לפנות ובתוכנת הניווט נבדוק כל הזמן את דגל זה על מנת לוודא שאנו רחוקים מהקיר וניתן להמשיך בתוכנית הראשית ואם אנו קרובים לקיר נפנה.

חשוב! – יש לשים דגש על הדחיפות והמשיכות מהמחסנית בתחילת ובסוף כל פסיקה שרצה! אנו עושים זאת מכיוון שהפסיקה לא דוחפת ומושכת את האוגרים A,B,X,Y ואת הCCR (Condition Code Register) אוטומטית בכל פסיקה! דבר זה אומר שאם לדוגמה אנו מריצים לולאה שבודקת את האוגר X ובאותו הזמן רצה פסיקה שמשנה את האוגר X, התוכנית הראשית תיתקע או תיתן תוצאות שגויות מכיוון שתוכן האוגר X ישתנה!

נמשיך לפונקצית חיישן הUvtron. הכנו שתי פונקציות, UV_OV ו-UV_OFF שתפקידן הוא להפעיל את שני חיישני הUvtron ולכבותם בהתאם (הפעלה – אפשור פסיקת UvTron). בנוסף, גילינו לאחר

מכן שאנו זקוקים גם לעוד שתי פונקציות: UV_ON_L ו-UV_ON_R שתפקידן הוא להדליק את החישן השמאלי או הימני בלבד בהתאמה. פונקצית הכיבוי של החיישנים זהה כמובן.

כעת נסתכל על פונקציות ההדלקה והכיבוי:

```
UV_ON:
          MOVB   #$0C,TFLG1          ; Clear UV Reads from TFLG1
          LDAA   TMSK1               ; Enable UV interrupts
          ORAA   #$0C                ; Enable UV interrupts
          STAA   TMSK1               ; Enable UV interrupts
          RTS
```

```
UV_OFF:
          MOVB   #$0C,TFLG1
          LDAA   TMSK1
          ANDA   #$F3
          STAA   TMSK1
          RTS
```

חיישני הUvtron מחוברים לכניסות pt2 וpt3 על PortT באמצעות כרטיס הInterface.
בשתי הפונקציות אנו בהתחלה מאפסים את קריאות הUV במידה והיו כאלו לפני שהחלטנו להגדיר פסיקה לחיישן. בפונקציה הראשונה אנו מבצעים פעולת Or של תוכן הזיכרון TMSK1 עם הערך 0C בבסיס 16, וזאת במקום להעביר את הערך ישירות לתא הזיכרון. ביצוע דבר זה ארוך יותר אך הוא בטוח ומונע בעיות ואף מקל על העבודה בתוכנה, מכיוון שכך אפשר בטעות להפעיל/לכבות פסיקות אחרות.
על מנת להראות את היתרון ניקח דוגמא שרירותית של המצוי בתוך האוגר TMSK1:

TMSK1	0	1	1	0	0	0	0	0

כעת נבצע עליו פעולת Or עם הערך $0C:

$0C	0	0	0	0	1	1	0	0
TMSK1+$0C	0	1	1	0	1	1	0	0

יש לשים לב שהפקודה <u>לא שינתה את הערך הקיים בtmsk1</u>, וזאת בניגוד לפקודה movb הרגילה (Move Bytes) אשר מוחקת את מה שהיה בtmsk1, והרי כאשר אנו מכניסים ערך חדש לtmsk1 אנו לא יודעים מה היה שם קודם.

בפונקציה השנייה אנו מבצעים פעולת And של תוכן הזיכרון שבTMSK1 עם הערך F3 בבסיס 16.
נדגים גם את ביצוע פעולה זו:

TMSK1	0	1	1	0	1	1	0	0

כעת נבצע עליו פעולת And עם הערך $F3:

$F3	1	1	1	1	0	0	1	1
TMSK1*$F3	0	1	1	0	0	0	0	0

עוד מידע ניתן לקבל בפרק על שערים לוגיים.

נמשיך ונראה את הפונקציה שרצה כאשר חיישן הUvtron קולט נר:

```
UV_INIT:
        MOVB   #$FF,UV_FLAG
        MOVB   #$0C,TFLG1
        RTI
```

בפונקציה זו אנו מאפסים או הדגל של קריאת הנר ומכניסים לתוך דגל שיצרנו (Uv_Flag) ערך שמציג זיהוי נר. לאחר מכן אנו חוזרים לתוכנית הראשית. בתוכנית אנו נבדוק את דגל זה כדי לדעת אם יש נר בחדר.

נעבור כעת לפסיקה הרצה בתוך חדר כאשר מחפשים את הנר. הפסיקה כוללת חיפוש פס לבן וPyro בהתאמה:

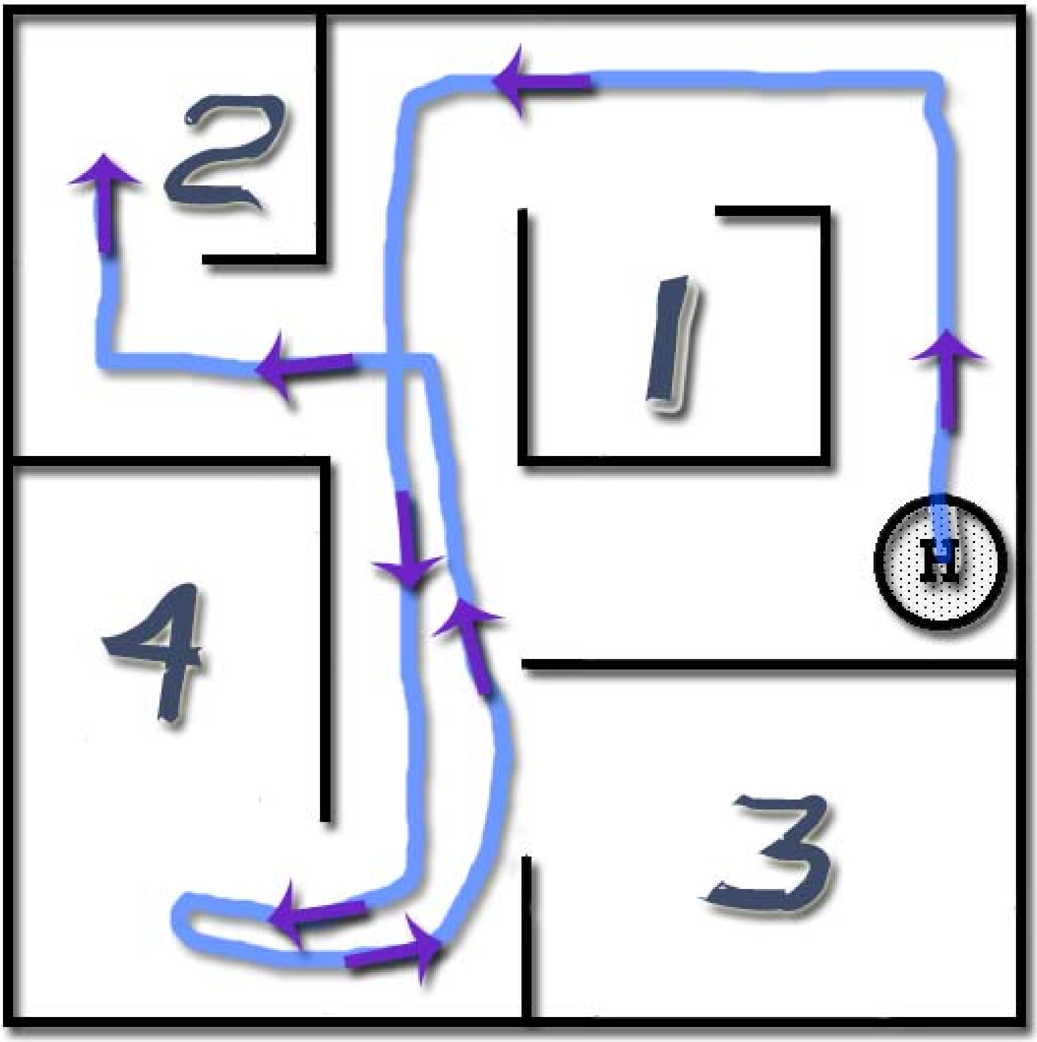
```
AKOV_INT:
        PSHX                        ; Push registers and accumulators
        PSHD
        PSHC
        CLR    PAS_FLAG             ; Clear pas lavan flag
        LDAA   PDLC                 ; Read from pas lavan
        ORAA   #$DF                 ; Read from pas lavan
        INCA                        ; Read from pas lavan
        LBNE   TURN_OFF_I           ; If found, jump to fan function
        LDAA   PDLC                 ; Read from pas lavan
        ORAA   #$EF                 ; Read from pas lavan
        INCA                        ; Read from pas lavan
        LBNE   TURN_OFF_I           ; If found, jump to fan function
        TST    SML_FLG              ; Flag to prevent pyro chk on
                                    ; small rooms
        LBNE    PYRO_CHK            ; Go to Pyro check

END_INT_F:
        PULC                        ; Pull registers and accumulators
        PULD
        PULX
        MOVB   #$80,TFLG2           ; Clear overflow interrupt
        RTI                         ; Return from interrupt
```

כצפוי, הפונקציה דוחפת ומושכת דברים מהמחסנית (הסבר מצוי קודם בפרק זה), בודקת פס לבן וחיישן Pyro שאת הפונקציה שלו נראה בהמשך (הפונקציה רצה רק בחדרים גדולים מאחר וגילינו שאין טעם להריץ אותה בחדרים קטנים). במידה והתגלה פס לבן באחד משני החיישנים הרובוט עובר למצב כיבוי והמאוורר מופעל.

פרק ח' - הניווט – Navigation

חלק הניווט בתוכנה הוא למעשה החלק העיקרי בתוכנה, בו אנו "מסבירים" לרובוט איך לנוע בתוך המבוך. קטע זה בעצם מאחד את שאר חלקי התוכנה, גם בצורה מעשית (ביצוע הפקודה Include), וגם על ידי שימוש בפונקציות הקיימות בשאר החלקים.

לפני שנתחיל בהסבר מקיף על הניווט, נסביר את תנועת הרובוט במבוך:



תנועת הרובוט במבוך

יש לשים מספר דגשים על דרך הניווט של הרובוט במבוך:

- הרובוט מסוגל לגלות את הנר בחדר מספר 1 על ידי מעבר על פני הכניסה ללא צורך להיכנס אליו. עובדה זו אושרה בניסויים רבים על ידי בדיקת מצבים קיצוניים ביותר של הנר והרהיט. עקב כך כאשר מתגלה נר בחדר הראשון הרובוט נע אחורנית ונכנס לחדר.
- כאשר הרובוט מגיע לחדר מספר 4 הוא מבצע שתי פניות ימינה על מנת לצאת ממנו, ותוך כדי הפניות בודק האם הנר דולק בחדר זה, וזאת על מנת להגיע לאחוזי גילוי מכסימליים.
- הרובוט לא נכנס גם לחדר מספר 3 מכיוון שגם בחדר זה אפשר לגלות את הנר ללא כניסה (כמו חדר 1), אך כדי לעשות זאת צריך להתקרב אל פתח החדר ולבצע סריקת Uvtron מחוץ לחדר.

- כאשר הרובוט מגיע לחדר האחרון – חדר מספר 2 הוא מחכה בו עד שהוא מזהה נר. הרעיון המקורי היה חזרה הביתה ובדיקה חוזרת וחוזרת במידה והרובוט לא נכנס למצב "כיבוי", אך מפאת קוצר זמן לא הספקנו לבצע זאת.

נעבור כעת על תוכנת הניווט בחלקים:

```
PRE_RUN:
             JSR  READ_ATD ; Reads from analog sensors
             LDAA ADR3H    ; Load microphone read to A
             CMPA #$C0     ; Compare A with $c0
             BLO  PRE_RUN  ; If not higher, go back to PRE_RUN
```

קטע זה מחכה עד לשמיעת גלאי עשן – או במקרה שלנו צפצפה בתדר 3-4Khz. באמצעות כך קיבלנו בונוס של 0.95 מהתוצאה הסופית של ריצת הרובוט.
התוכנה קוראת את הערך שכרטיס המיקרופון נותן ומשווה אותו לערך C0. במידה והערך הוא גבוה מהערך C0 התוכנה ממשיכה קדימה לתחילת הריצה. (יש לשים לב שכרטיס המיקרופון מחובר לממיר האנלוגי לדיגיטלי בכתובת adr3.

```
INT_RUN:
             CLR  ROOM            ; Clear room counter
             CLR  PLZ_TURN        ; Clear turns flag
             MOVB #$20,WALL       ; Distance from wall for turn
             MOVB #$90,TURNWEAK   ; Duty for wheel for turn
             MOVB #$9900,E_TURN   ; Number of encoders for turn
             JSR  X_L             ; Accelerate
             CLI                  ; Enable Interrupts
```

שלב האיתחולים ותחילת הריצה: בשלב זה אנו מאפסים את מונה החדר, לפיו הרובוט יודע כיצד להתנהג כאשר זיהה נר בכל אחד מהחדרים, ומאפסים גם את הדגל של בקשת הפנייה עליה נפרט בהמשך. בנוסף, אנו עורכים הגדרות לפנייה הראשונה: הגדרת מרחק מהקיר, עוצמת סיבוב של הגלגל פנימי וגודל הסיבוב (לפי חורי Encoder). לאחר מכן אנו מאיצים בצורה הדרגתית כדי למנוע את החלקה ומפעילים את הפסיקות, שהן למעשה מודיעות לנו מתי צריך לבצע פנייה (עוד מידע בקטע הפסיקות של התוכנה).

כעת נעבור בזריזות על הניווט עצמו.
הניווט מחולק לחלקים (Sections), כאשר כל חלק מהווה קטע מסוים בזירה אותו אנו עוברים.

- חלק 1: נסיעה עד לקיר ראשון
- חלק 2-4: נסיעה עד לקיר שני כולל בדיקת נר בחדר 1.
- חלק 5-8: נסיעה במסדרון הראשי עד לפנייה כולל החלפות תיקונים (לפי ימין ולפי שמאל)
- חלק 8ב'-9: שתי פניות של 90 מעלות ויציאה מחדר 4 (סריקת נר בחדר 4).
- חלק 9-12 – פנייה למסדרון הראשי, תזוזה לכיוון חדר 3 וחזרה למסדרון.
- חלק 13-19 – נסיעה עד לחדר 2 כולל פנייה באמצע המסדרון הראשי.

פרק ט' - תפעול חיישנים בתוכנה

בפרק זה נסביר כיצד תפעלנו את החיישנים השונים באמצעות התוכנה.
החיישנים עליהם נסביר הם:

- חיישני אינפרא-אדום
- חיישנים אולטרא-סוניים
- חיישני פס לבן
- חיישני UvTron
- חיישני Pyro Electric
- חיישן מיקרופון
- חיישן מקודד (Encoder)

<u>חיישני אינפרא-אדום:</u>

החיישן GPD12 נותן אות אנלוגי מ0-5v בהתאם למרחק שהוא "רואה". האות האנלוגי עובר עיבוד על ידי רכיב הAnalog to Digital והוא מגיע לאחד הכניסות Adr0-7. על מנת לקרוא את האות הדיגיטלי המתקבל עלינו לבצע את הפעולות הבאות:
אתחול הממיר האנלוגי לדיגיטלי: מתבצע בחלק האיתחולים ורץ רק פעם אחת בתחילת התוכנה.
קריאה והמרה מכל החיישנים – ניתן גם להמיר רק ארבעה חיישנים בכל פעם.
העברת המידע לאוגר ו/או מקום בזיכרון.

שלב האיתחולים הוא כדלהלן:

```
AD_INIT:    MOVB #$80,ATDCTL2       ; A/D Power On
            LDAA #$C8               ; Delay
AD_DELAY:   DECA                    ; Delay
            BNE  AD_DELAY           ; Delay
            MOVB #$00,ATDCTL3
            MOVB #$01,ATDCTL4
            RTS
```

בתחילה אנו מעבירים 1 לביט השמאלי ביותר של האוגר ATDCTL2, כלומר מאפשרים את פעולת הממיר. במידה והיה בביט זה 0, הממיר לא היה מקבל מתח מהמעבד, דבר שימושי לצריכת חשמל כאשר אין שימוש בממיר. במידה ומועבר 0 לביט זה בתהליך המרה, הוא מופסק מיידית.
לאחר שהגדרנו את הממיר כ"מופעל", עלינו לחכות זמן קצר עד שהממיר "מתאושש" ולכן ישנה לולאת השהייה של כ-100 מיקרו שניות.
לאחר מכן אנו מאפסים את תוכן האוגר ATDCTL3, שמשמש לניפוי וניטור שגיאות. כאשר הוא מאופס הממיר ממשיך לעבוד כרגיל.
האוגר ATDCTL4 אחראי על שעון הממיר ועל מספר הביטים להמרה. בתוכנית זו אנו מבקשים המרה של 8 ביט.

את שלב הקריאות הכנסנו לפונקציה הנקראת READ_ATD אליה אנו קופצים בכל פעם שאנו רוצים לבצע המרה:

```
READ_ATD:  MOVB   #$50,ATDCTL5            ; Define A/D convert method
AD_PROC:   BRCLR ATDSTAT $80 AD_PROC      ; Wait till convert is done
           RTS                            ; Return from subroutine
```

האוגר ATDCTL5 מגדיר את שיטת ההמרה ואת ערוצי ההמרה. כל כתיבה לאוגר זה אומרת לממיר הa/d להתחיל להמיר לפי התנאים שכתבנו באוגר זה. כאשר אנו מעבירים את הערך $50 אנו אומרים לממיר שאנו רוצים להמיר את כל שמונה הכתובות, שאנו רוצים המרה אחת בלבד ושההמרה תתבצע מערוץ אנלוגי מספר X לכתובת בזיכרון AdrX (0 ל-0, 1 ל-1 וכדומה).
לאחר מכן אנו מחכים עד שההמרה תושלם. הסימן לכך שההמרה הסתיימה היא שהביט השמאלי באוגר ATDSTAT מוגדר כ1 ולכן אנו משתמשים בפקודה Branch If Clear ששמה "מסכה" על כל הביטים אשר מוגדרים בפקודה כ0 ובודקת אם הביטים שמוגדרים כ1 מוגדרים כ0. במידה וכן הפקודה תקפוץ לתווית המוגדרת כAD_PROC עד אשר יהיה 1 בביט השמאלי ביותר באוגר ATDSTAT.

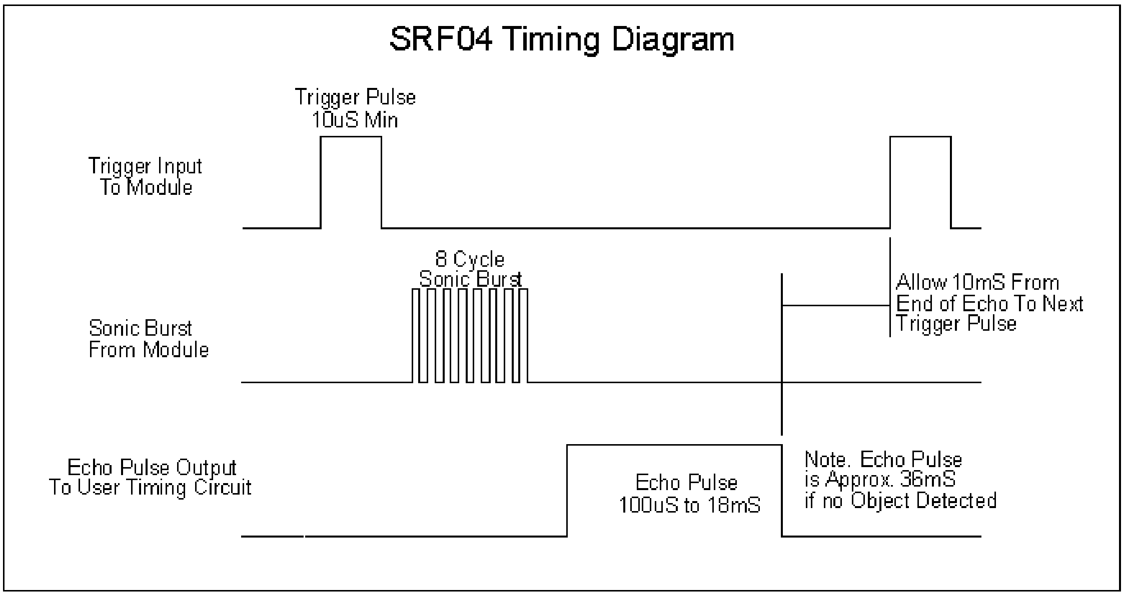
החלק האחרון של קריאת התוצאות מתבצע בתוכנה עצמה. ניקח דוגמא מהפסיקה בה בדקנו האם יש קיר הקרוב אלינו:

```
...
...
CHK_WALL:                          ; Check for front wall
            JSR     READ_ATD       ; Read from left forward IR
            LDAB    ADR0H
            CMPB    WALL           ; Compare to "near wall" value
            BMI     SIOM
...
...
```

אנו קופצים לפונקציה הקודמת, ולאחר מכן מעלים את המידע לאקומולאטור מהערוץ הרצוי – במקרה זה האקומולאטור B. הערוצים נעים בין ADR0-ADR7. יש לציין שההמרה יכולה להיות של 8 ביט או 10 ביט ומכיוון שאנו משתמשים רק ב8 אנו תמיד מעלים רק את הHigh Byte (ADR0H) של ההמרה. צריך לציין שאנו לא משתמשים בהמרה של 10 ביט מכיוון שאנו לא צריכים דיוק עד רמה כזו ומכיוון שחיישני האינפרא אדום גם כך נותנים קריאות לא יציבות הנעות בתחום של ±2/3 מהקריאה האמיתית.
לאחר שהעברנו את המידע לאקומולאטור ניתן לבצע פעולות חישוב כמו השוואה לנתון מסוים ותגובה בהתאם (במקרה זה, בדיקת מרחק מהקיר הקדמי).

<u>חיישנים אולטרא סוניים</u>

החיישן SRF04 בודק את המרחק מהקיר באמצעות גל קול בטווח האולטרא-סוני. החיישן מודד למעשה את המרחק שלוקח לגל לפגוע בקיר ולחזור. החיישן מתופעל באופן הבא:



דיאגרמת תזמון של החיישן SRF04

כפי שניתן לראות, על מנת לגרום לחיישן לשלוח קריאה יש צורך:

- להעביר לרגל הTrigger גל שאורכו הוא 10 מיקרו שניות.
- לחכות עד אשר תתקבל קריאה ברגל הEcho ולמדוד את זמן הקריאה. זמן זה הוא למעשה המרחק.
- לחכות בין קריאה לקריאה לפחות 10 מילי שניות אם הקריאות רצופות.

חיבורי החיישן: החיישן מחובר לPortT אך על מנת לייצר גל בTrigger חיברנו רגל אחת מכל חיישן לPortDLC שכאשר מגדירים אותו כ-1 הוא נותן 5v ביציאה. את פורט זה ניתן להגדיר גם כפורט כניסה וגם כפורט יציאה (יש 7 כניסות לפורט זה, אך לא ניתן להשתמש בpdlc0, Pdlc1 וPdlc7)

לפני שנתחיל, יש צורך להגדיר את הPortDLC כפורט יציאה ברגליים בהן אנו משתמשים:

```
PDLC_INI:
        MOVB  #$CF,DDRDLC
        RTS
```

DDRDLC הוא האוגר שאחראי על הגדרת PortDLC כפורט של כניסה או יציאה. כעת הגדרנו את הפורט בצורה כזו:

	7	6	5	4	3	2	1	0
PDLC	N/A	OUT	IN	IN	OUT	OUT	N/A	N/A

החיישנים מחוברים לרגליים מספר 2 ו3 שמוגדרות כOutput.

התוכנה שבאמצעותה אנו משתמשים כדי לקרוא מהחיישנים נראית כך:

```
US0:     MOVB #$40,USUPW     ; Rising edge TCTL3
         MOVB #$80,USFLG     ; Flag for TFLG1
         MOVB #$80,USDNW     ; Falling edge for TCTL3
         MOVB #$08,US_INIT   ; PDLC leg to send wave
         JMP  USENG          ; Jump to main function

US1:     MOVB #$10,USUPW
         MOVB #$40,USFLG
         MOVB #$20,USDNW
         MOVB #$04,US_INIT
         JMP  USENG

US2:     MOVB #$04,USUPW
         MOVB #$20,USFLG
         MOVB #$08,USDNW
         MOVB #$08,US_INIT
         JMP  USENG

US3:     MOVB #$01,USUPW
         MOVB #$10,USFLG
         MOVB #$02,USDNW
         MOVB #$04,US_INIT
         JMP  USENG
```

בתחילה, בנינו ממשק פשוט על מנת שיהיה לנו קל לקרוא מכל ארבעת החיישנים. הכנו פונקציות בשם US0-US3 שלמעשה מבצעות הגדרות של מיקום החיישן בPortT ואת מיקום רגל הTrigger בPortDLC. לאחר מכן הפונקציה קופצת לקטע המרכזי בתוכנה – USENG:

```
USENG:
         PSHC                 ; Push registers and accumulators
         PSHD
         PSHX
         SEI                  ; Stop Interrupts
PREREAD: LDX  #$0400          ; 10 mili seconds delay
PREDAL:  LDAA #$20
PRERD:   DECA
         BNE  PRERD
         DEX
         BNE  PREDAL

         LDAA PDLC            ; Set PDLC leg = +5v
         ORAA US_INIT
         STAA PDLC
         LDAA #$20            ; 10 micro seconds delay
USSHEV:  DECA
         BNE  USSHEV
         LDAA PDLC
         LDAB US_INIT
         COMB
         STAB US_INIT
         ANDA US_INIT
         STAA PDLC            ; Set PDLC leg = 0v
```

בתחילת קטע זה אנו מבצעים לולאת השהייה של 10 מילי שניות בשביל ההשהיה בין קריאה לקריאה. היה ניתן גם לשים את ההשהיה בסוף הקריאה.

לאחר מכן אנו מגדירים את הרגל הרצויה של PortDLC כ-1. החיישנים מחוברים לרגל מספר 2 או רגל מספר 3 ומבצעים השהייה של כ10 מיקרו שניות ואז מפסיקים את המתח ברגל זו.

```
USREAD:
         LDAA TCTL3           ; Set TCTL3 value for rising edge
         ORAA USUPW           ; Set TCTL3 value for rising edge
         STAA TCTL3           ; Set TCTL3 value for rising edge
         MOVB USFLG,TFLG1     ; Clear the appropriate flag
SHUV:    LDAA TFLG1           ; Check if rising edge has come
         ANDA USFLG           ; Check if rising edge has come
         BEQ  SHUV            ; Jump back to check
         LDAA TCTL3           ; Set TCTL3 for falling edge
         EORA USUPW           ; Set TCTL3 for falling edge
         ORAA USDNW           ; Set TCTL3 for falling edge
         STAA TCTL3           ; Set TCTL3 for falling edge
         MOVB USFLG,TFLG1     ; Clear the appropriate flag
         LDX  #$0000          ; Clear accumulator X

MONE:    INX                  ; Increase X
         LDAA TFLG1           ; Check for falling edge
         ANDA USFLG           ; Check for falling edge
         BEQ  MONE            ; Jump back to check
         STX  US_READ         ; Put X in US_READ
         PULX                 ; Pull registers and accumulators
         PULD
         PULC
         RTS                  ; Return from subroutine
```

אחרי הפעלת הTrigger החיישן אמור לתת קריאה ברגל הEcho ולכן אנו מגדירים את ה Input Capture כדי ש"יתפוס" את הגל בעלייתו ואנו מחכים עד שדבר זה קורה. לאחר מכן אנו מתחילים לספור מונה שהוא למעשה האוגר X ואנו מחכים בלולאה אינסופית בה מעלים את המונה לגל יורד שמראא על סיום הקריאה. לאחר סיום הקריאה אנו מכניסים אותה לתא בזיכרון בשם US_READ בו נשתמש בתוכנה כדי לקרוא מהחיישן.

בחלק זה של התוכנה יש שני דברים עליהם נרחיב:

בדיקת TFLG1 לגילוי הגל העולה

שינוי TCTL3 מגל עולה ליורד

בדיקת האוגר TFLG1 היא פשוטה: כל הזמן מבצעים פעולת AND כאשר הביט הרצוי מסומן כ1. לדוגמה: במקרה זה נרצה לבדוק האם הביט השמאלי ביותר מוגדר כ1 ולכן נבצע פעולת AND עם הערך $80.

TFLG1	0	0	0	0	1	1	0	0
TFLG1+$80	0	0	0	0	0	0	0	0

כאן הביט עדיין לא עלה, לכן התוצאה היא 0.

TFLG1	1	0	0	0	1	1	0	0
TFG11*$80	1	0	0	0	0	0	0	0

ברגע שהביט עלה, התוצאה משתנה מהערך 0 ואז יודעים שהערך בtflg1 השתנה.

<u>הערה:</u> ניתן לבצע זאת גם עם הפקודה BRCLR (Branch If Clear) אך במקרה זה הערך שמבצעים איתו מסכה נמצע בזיכרון באחר מהאוגרים ולכן אי אפשר לבצע זאת עם הפקודה הנ"ל.

הפעולה השנייה עליה נרחיב היא פעולת השינוי מגל עולה ליורד. בפעולה זו אנו מבצעים פעולת Xor (עוד בנספח א') של TCTL3 עם הערך שקודם עשינו לו OR איתו – US0UPW על מנת לבטל את השינוי שביצענו קודם במהירות ולאחר מכן אנו מבצעים פעולת OR עם ערך הגל היורד על מנת להחיל את בקשת תפיסת הגל היורד.

<u>הערה:</u> למעשה לא היה צורך להגדיר בתוכנה את הגל היורד, היה ניתן פשוט להעלות את ערך הגל העולה לאקומולאטור מסוים ועליו לבצע את הפעולהASRA שמזיזה את כל הביטים מקום אחד ימינה.

US0UPW	1	0	0	0	0	0	0	0
US0DNW	0	1	0	0	0	0	0	0
TCTL3	1	0	0	0	0	0	0	1
TCTL3 ⊕ $80	0	0	0	0	0	0	0	1
TCTL3+$40	0	1	0	0	0	0	0	1

והגענו לתוצאה הרצויה.

כעת נראה דוגמא לקריאה מהחיישן בתוכנה הראשית:

```
...
...
        JSR   US0              ; Read from sensor number 0
        LDX   US_READ          ; Load US_READ to accumulator X
        CPX   #$200            ; Compare to "near right wall" value
        BMI   NEAR_RW          ; Jump to the appropriate function
...
...
```

חיישני פס לבן

חיישני הפס הלבן נותנים אות דיגיטלי. את החיישנים ניתן לחבר בשתי דרכים:
חיבור לPortT – דורש מקום באותו הפורט שלא היה לנו.
חיבור לPortDLC – חיבור לרגל אחת באותו הפורט.

את החיישנים חיברנו לPortDLC לרגליים מספר 4 ו-5, והגדרנו אותן כרגלי Input בתוכנה (ראה זאת בסעיף על החיישנים האולטרא-סוניים).

הפס הלבן נותן +5v כאשר הוא <u>לא</u> רואה פס לבן ו0v כאשר הוא רואה פס לבן (לוגיקה הפוכה).

התוכנה לבדיקת פס לבן המחובר לרגל מס' 5 נראית כך:

```
PAS_LAV:
          PSHD                       ; Save content of D
          CLR    PAS_FLAG            ; Clear flag
          LDAA   PDLC                ; Load content of Pdlc to A
          ORAA   #$DF                ; A=A+$DF
          INCA                       ; A=A+$1
          BNE    PAS_1               ; If not equal to zero = there is pas
PAS_0:    MOVB   #$00,PAS_FLAG       ; No line
          PULD                       ; Pull D from before
          RTS                        ; Return from subroutine
PAS_1:    MOVB   #$FF,PAS_FLAG       ; There is a line
          PULD                       ; Pull D from before
          RTS                        ; Return from subroutine
```

אנו משתמשים בדגל שיצרנו המכונה PAS_FLAG כדי לבדוק האם יש פס לבן או לאו. במידה ויש, הדגל יכיל $FF ובמידה ואין הוא יכיל $00. את הדגל הנ"ל אנו בודקים בתוכנה הראשית לאחר קפיצה לפונקציה זו.
כעת נראה את לוגיקת בדיקת הפס הלבן. יש לזכור שהחיישן נותן +5v כאשר הוא <u>לא</u> רואה פס לבן לכן אנו רוצים שPortDLC יראה 0 ברגל הרצויה כאשר אנו מעל פס לבן ו1 כאשר הוא לא רואה.

דוגמא לPortDLC כאשר החיישן לא מעל פס לבן:

$DF	1	1	0	1	1	1	1	1
PDLC	0	0	1	0	1	0	1	0
PDLC+$DF	1	1	1	1	1	1	1	1
PDLC+$DF+$1	0	0	0	0	0	0	0	0

PDLC שווה ל-0 ולכן לא גילינו פס לבן (יש לשים לב שאומנם יש 1 בCarry, אך דבר זה לא משנה לפקודת הBNE – Branch If Not Equal To Zero).

דוגמא לPortDLC כאשר החיישן מעל פס לבן:

$DF	1	1	0	1	1	1	1	1
PDLC	0	0	0	0	1	0	1	0
PDLC+$DF	1	1	0	1	1	1	1	1
PDLC+$DF+$1	1	1	1	0	0	0	0	0

PDLC אינו שווה ל-0 ולכן גילינו פס לבן.

באותה צורה ניתן לבדוק גם את חיישן הפס הלבן השני שמחובר לרגל מספר 4.

דוגמא לתפעול חיישן הפס הלבן בתוכנה:

```
...
...
SECT12:
            JSR     PAS_LAV           ; Check for white line
            TST     PAS_FLAG          ; Check the white line flag
            BEQ     SECT12            ; Loop until there is white line
...
...
```

בקטע זה השייך לניווט אנו ממשיכים לנסוע עד אשר אנו רואים פס לבן בחיישן האחורי.

<u>חיישני UvTron</u>

החיישנים מחזירים אות דיגיטלי ומחוברים לPortT. עוד פירוט על חיישנים אלו בפרק על פסיקות.

<u>חיישני Pyro</u>

חיישנים אלו מחזירים אות אנלוגי ומחוברים לפורט האנלוגי.
על מנת לקרוא מהחיישנים הללו השתמשנו בממיר הAnalog to Digital בצורה הבאה:

```
...
...
            JSR     READ_ATD      ; Read from ATD
            LDAA    ADR6H         ; Load right Pyro read to A
            CMPA    #$10          ; Check for fire
            BLO     CNDL_R        ; Check for fire
            CMPA    #$E8          ; Check for fire
            BHI     CNDL_R        ; Check for fire
            LDAA    ADR7H         ; Load left Pyro read to A
            CMPA    #$10          ; Check for fire
            BLO     CNDL_L        ; Check for fire
            CMPA    #$B0          ; Check for fire
            BHI     CNDL_L        ; Check for fire
...
...
```

בקטע תוכנית זו קראנו מכל הפורטים האנלוגיים ובחרנו להתייחס רק לערוצי הפיירו. הבדיקה נערכת פעמיים לכל חיישן – לטווח מקסימאלי ומינימאלי. לדוגמה נראה שהחיישן הימני "יראה" שריפה רק כאשר הוא נותן ערך מתחת ל10$ ומעל לE8$. ההסבר לכך נעוץ בעובדה שהפיירו בודק שינויים בחום ולכן הוא נותן ערכים נמוכים כאשר הוא עובד מימין לשמאל לחום וערכים גבוהים כאשר הוא עובר משמאל לימין. הערך הרגיל שהפיירו נותן כאשר הוא לא רואה שינויים בחום הוא בערך 80$.

<u>חיישן המיקרופון</u>

חיישן זה מחזיר אות אנלוגי ואת תפעולו ניתן לראות בפרק הניווט (Pre_Run).

<u>חיישן המקודד (Encoder)</u>

חיישן זה מחזיר אות דיגיטלי והוא מצוי בתוך המנוע. חיישן זה סופר את המרחק שהגלגל נע כאשר ישנם 9750 חורים לכל סיבוב מלא של הגלגל. בצורה זו ניתן למדוד מרחקים מדויקים שאותם הרובוט עובר. המקודדים בתוכנה שלנו התחברו לכניסות 6 ו-7 ולכן התוכנה שלנו נראית כך:

```
L_ENCO:
            PSHA
            LDAA   #ENC_PUL1
            STAA   ENC_PULS
            LDAA   #ENC_FLG1
            STAA   ENC_FLAG
            PULA
            RTS

R_ENCO:
            PSHA
            LDAA   #ENC_PUL0
            STAA   ENC_PULS
            LDAA   #ENC_FLG0
            STAA   ENC_FLAG
            PULA
            RTS

PLS_CNT:
            MOVB   TCTL4,TCTLPLCE
            MOVB   ENC_PULS,TCTL4
            MOVB   ENC_FLAG,TFLG1
C_P_CNT:    LDAA   TFLG1
            ANDA   ENC_FLAG
            BEQ    C_P_CNT
            MOVB   ENC_FLAG,TFLG1
            DEX
            BNE    C_P_CNT
            MOVB   TCTLPLCE,TCTL4
            RTS
```

לפני הקפיצה לתווית PLS_CNT אנו מעלים ערך לאוגר X שהוא למעשה מספר החורים שאותם אנו רוצים לספור (בבסיס 10 או 16) וקופצים לL_Enco או לR_Enco בהתאמה לפי המקודד הרצוי. פרוצדורה זו מגדירה את המקודד שלפיו אנו רוצים לספור - הימני או השמאלי. לאחר מכן אני עוברים לתווית PLS_CNT שהיא מחכה כל פעם לחור ומחסירה מהאוגר X אחד עד לאיפוס האוגר – סיום ספירת החורים.

פרק י' – הכיבוי

קטע הכיבוי מחולק לארבעה חלקים:

- חלק הפתיחה – כניסה לחדר וכדומה.
- תפעול חיישני זיהוי נר.
- חלק העקיבה לפי קיר ועקיפת רהיט.
- חלק הכיבוי – הפעלת המאוורר ו"וידוי הריגה".

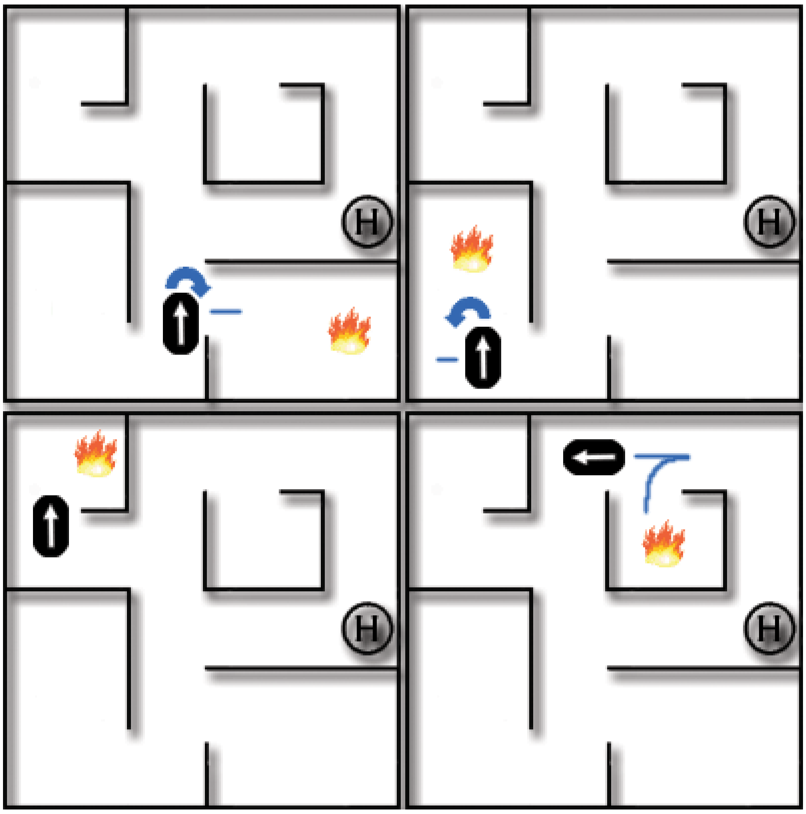
<u>חלק הפתיחה</u>

כאשר התוכנה מזהה נר באחד מן החדרים היא קופצת לפרוצדורה בשם Pre_Akov. פרוצדורה זו תפקידה של פרוצדורה זו הוא לבצע כניסה לשלושת החדרים הראשונים ולמקם את הרובוט בצורה נוחה להתחלת עקיבה אחר קיר.
במהלך תוכנת הניווט אנו מעבירים לדגל בשם R00M את מספר החדר בו אנו נמצאים. כאשר אנו קופצים לפונקצית הכיבוי אנו בודקים את הדגל וכך ניתן לדעת באיזה חדר אנו נמצאים ומה הדרוש לעשות על מנת להיכנס לחדר ולהגיע למקום הרצוי לפני תחילת העקיבה.

```
CHK_ROOM:  LDAA    R00M
           BEQ     ROOM0
           CMPA    #$1
           BEQ     ROOM1
           CMPA    #$2
           LBEQ    ROOM2
           CMPA    #$3
           LBEQ    ROOM3
           RTS
```

בנוסף, הכנו פונקציות בשם ROOM1, ROOM2 ו ROOM3 שתפקידן להיכנס לתוך החדרים. בחדר מספר 4 אין צורך לבצע שום כניסה מכיוון שהרובוט מצוי כבר במצב אידיאלי.

כניסת הרובוט לחדרים מתוארת באיור הבא כאשר הקו הכחול מסמל תנועה של הרובוט והחיצים מסמנים סיבוב של תשעים מעלות בכיוון החץ.



תנועת הרובוט בעת זיהוי נר

<u>תפעול חיישני זיהוי נר</u>

בחלק זה אנו מתפעלים שני סוגי חיישנים:

- חיישני Pyro
- חיישני פס לבן

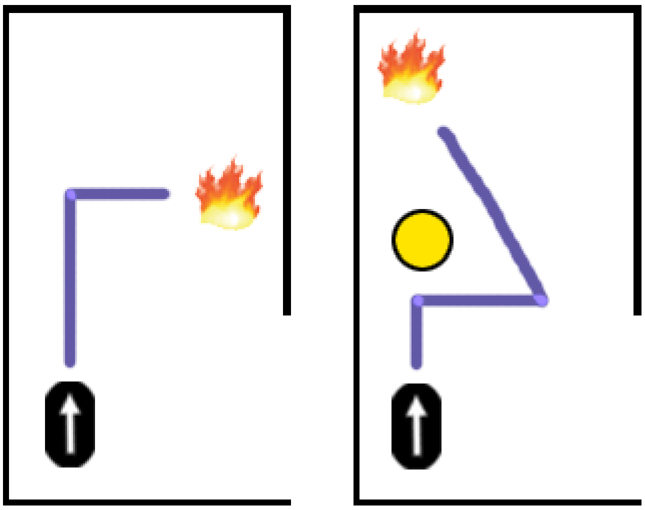
תפעול חיישני הPyro מתואר כבר בפרק תפעול החיישנים. אך נעבור עליו הפעם בשילוב עם תוכנת הכיבוי.

```
PYRO_CHK:
            CLR     PYRO_FLG
            JSR     READ_ATD
            LDAA    ADR6H
            CMPA    #$10
            BLO     CNDL_R
            CMPA    #$E8
            BHI     CNDL_R
            LDAA    ADR7H
            CMPA    #$10
            BLO     CNDL_L
            CMPA    #$B0
            BHI     CNDL_L
            JSR     SRCH_PAS
            JMP     END_INT_F
CNDL_L:
            MOVB    #$43,PYRO_FLG
            BRA     GO_CNDL
```

```
CNDL_R:
            MOVB    #$13,PYRO_FLG
GO_CNDL:
            JSR     L_BRAKE
            MOVB    #$03,PWEN
            LDAA    PYRO_FLG
            STAA    PORTPP
            MOVB    #$B8,PWDTY0
            MOVB    #$B8,PWDTY1
            CMPA    #$20
            BMI     PYRO_R
PYRO_L:     JSR     L_ENCO
            LDX     #4200
            MOVB    #$FF,TURN_FLG
            BRA     PYRO_CONT
PYRO_R:     JSR     R_ENCO
            CLR     TURN_FLG
            LDX     #4000
PYRO_CONT:
            JSR     SRCH_PAS
            JSR     PLS_CNT
            JSR     L_BRAKE
            MOVB    #$03,PWEN
            MOVB    #$03,PORTPP
            MOVB    #$B8,PWDTY0
            MOVB    #$B8,PWDTY1
...
...
```

כפי שניתן לראות, בחלק זה אנו בודקים האם חיישן הPyro זיהה חום לפי טווח (עוד בפרק על תפעול חיישנים). במידה והחיישן ראה אנו מעבירים לדגל בשם Pyro_Flg את הערך שצריך להעביר לPortPP לפנייה (ימינה או שמאלה) ולאוגר X את מספר הEncoders שאנו רוצים לספור בפנייה ולאחר מכן מבצעים את הפנייה וחוזרים לנסיעה בקו ישר שאמורה להיות נסיעה בקו ישר אל הנר.

דוגמא לכך מופיעה באיור הבא (מומלץ להצטייד בידע מחלק העקיבה בתוכנה), כאשר הרובוט נמצא בחדר מספר 3. דוגמא אחת ללא רהיט ודוגמא אחת עם רהיט. הקו הכחול מסמל את תנועת הרובוט בחדר עד להגעה לנר.



תנועת הרובוט באמצעות חיישן הPyro

<u>חלק העקיבה</u>

חלק זה מתואר בפרק יא' – תיקונים.

<u>חלק הכיבוי</u>

התוכנה מגיעה לחלק זה כאשר הרובוט מזהה פס לבן במהלך העקיבה בחדר. פס לבן זה מציג שתי אפשרויות:

- פס לבן של יציאה מחדר.
- פס לבן שנמצא מול הנר.

בדיקת האפשרות נעשית על ידי בדיקת הקיר הקדמי לפי חיישני המרחק. במידה והמרחק גדול מ30 סנטימטרים אנו יודעים שמדובר בפס לבן של חדר מכיון שהנר מצוי תמיד בפינת החדר.

```
TURN_OFF:
            SEI                     ; Cancel Interrupts
            CLR    PAS_INT
            LDX    #$08             ; Times to check IR's
WALL_FAN:                           ; Check for front wall
            JSR     READ_ATD        ; Read From IR Sensors
            LDAB    ADR0H           ; Load left front IR read
            CMPB    #FAN_WALL       ; Compare it with wall value
            BLO     FALSE_LINE      ; If it is a wall value, go to the
                                    ; false line procedure
            JSR     READ_ATD        ; Read From IR Sensors
            LDAA    ADR1H           ; Load right front IR read
            CMPA    #FAN_WALL       ; Compare it with wall value
            BLO     FALSE_LINE      ; If it is a wall value, go to the
                                    ; false line procedure
            DEX                     ; Decrease X
            BNE     WALL_FAN        ; If X<>0, go to wall_fan
            LBRA    TURN_OK         ; Start extinguish procedure
```

במידה והפס הלבן הוא פס הנמצא בכל כניסה לחדר, אנו מבצעים סיבוב של 180 מעלות וחוזרים לחדר בפנייה לכיוון המקום בו יש אפשרות לפנייה (תמיד יש קיר באחד הצדדים), וזאת כדי למנוע יציאה נוספת כזו.

אם הפס הלבן מזוהה עם האפשרות השנייה, אנו מפעילים את המאוורר למספר שניות. במידה ואנו ממשיכים לזהות בחיישן הUvTron נר אנו נבצע סיבוב של 720 מעלות כאשר המאוורר פועל, ונמשיך לבדוק את החיישן הנ"ל. במידה והחיישן עדיין מזהה נר, נמשיך מההתחלה עד שנזהה שהנר כובה.

פרק יא' - תיקונים

תיקונים בתנועה

התיקונים בתנועה מתחלקים לשני חלקים: תיקונים לאמצע ותיקוני זווית.
התיקונים למרכז דואגים לכך שהרובוט ייסע במרכז המסדרון, לנסיעה במרכז המסדרון יש חשיבות בכך שאם הרובוט מתקרב לקירות יש סיכון שיפגע בהם. הבעייתיות בביצוע תיקונים למרכז היא שכאשר הרובוט נמצא בזווית המרחק שנמדד בחיישנים יותר גבוה, ולכן אם הרובוט קרוב לקיר אבל בזווית הקריאה בחישן תראה שהרובוט באמצע המסדרון, לכן אנו חייבים להתייחס לזווית המופיעה לפי ההפרש בין המרחקים שנמדדים בחישן האחורי והקדמי. אנו לא יכולים פשוט להשוות קריאות מרחק משני צידי הרובוט מכיוון שבמקרים רבים בדירה אין לנו קיר משני הצדדים.
מטרת תיקוני זווית היא לדואג שהרובוט ייסע במקביל לקירות – בעצם לדאוג שכאשר הרובוט נוסע באמצע המסדרון- הוא לא יסטה ויתקרב לאחד הקירות. תיקוני זווית גם חשובים לפניות מכיוון שאם הרובוט מגיע לפניה בזווית הוא עלול לבצע אותה בצורה לא נכונה ולפגוע בקיר.

שיטות תיקונים

תיקונים פשוטים:

תיקונים פשוטים הם בעצם הרמה הפשוטה ביותר של תיקונים, הרעיון שעומד מאחוריהם הוא פשוט ביותר, טעות מתוקנת בצורה קבועה בלי קשר לגודל הטעות. המשמעות היא שלא משנה מה גודל הטעות התיקון תמיד יהיה זהה. תיקונים אלה הם הפשוטים ביותר מכיוון שאין שום צורך לקיים יחס בין הטעות לתיקון, פשוט ברגע שמזהים טעות קופצים לתיקון הקבוע.
אפשר לבצע תיקונים פשוטים רק לזווית, בשיטה זו הרובוט לא ייסע במרכז המסדרון אבל הוא ייסע ישר ולכן יש סיכוי טוב שלא יעל על קירות. בנוסף יש לזכור שכשאר הרובוט מבצע פניות במקום הוא יסיים אותם באמצע המסדרון, דבר שיפצה על כך שאין תיקונים למרכז. כמובן שבשיטת תיקונים זו אין אופציה לבצע פניות רחבות או אפילו לנסוע במהירות גבוהה.
לצורך ביצוע של תיקונים למרכז, תהפוך התוכנה למסובכת בהרבה מכיוון שיידרש חישוב המרחק האמיתי בין הרובוט לקיר לפי הזווית. אבל ישנה אופציה פשוטה לשלב תיקונים למרכז עם תיקוני זווית, מאחר וכשהרובוט נוסע ישר מראים החיישנים על המרחק האמיתי לקיר, נבצע תיקונים למרכז רק כאשר הרובוט נוסע במקביל לקירות. מה שיקרה בפועל זה שהרובוט יתקן את עצמו להיות מקביל לקיר ואז יפנה כדי להגיע למרכז, בעקבות הפניה הרובוט שוב יהיה בזווית והיא תתוקן ע"י תיקוני הזווית.
אם נתן רק ערך אחד לתיקונים הרובוט בעצם יבצע את הפעולה המתוארת פעמים רבות עד שיגיע למרכז.

אנו תכנתנו תיקוני זווית פשוטים על הרובוט "במפר" שעליו התאמנו בתחילה ועל הרובוט "שכטר" אותו בנינו לפני קרפלעך.

<u>תיקונים בתנועה לפי תיקונים יחסיים: (פירוט על שיטת תיקונים יחסיים בנספח ח' – תיקונים יחסיים)</u>

אפשר לבצע תיקונים יחסיים גם על מרחק הרובוט מאמצע המסדרון (תיקונים למרכז) וגם על הזווית בין הרובוט לקיר. כאשר מבצעים תיקונים יחסיים על מרחק הרובוט מאמצע המסדרון כדי מאוד להוסיף גם את תיקון השיפוע (D), מכיוון שהוא ימנע תיקון יתר, וגם יגרום לביצוע של תיקוני זווית (מאחר וקצב שינוי המרחק מהאמצע יחסי לזווית של הרובוט כלפי האמצע). לתיקוני זווית אין שום צורך בתיקון השיפוע מכיוון שכאשר טעות הזווית תהיה אפס (הרובוט יהיה ישר) אז מהירות הגלגלים תהיה שווה והרובוט ימשיך ישר – אין אפשרות שייוצר תיקון יתר.

<u>תיקונים לפי נוסחאות:</u>

תיקונים לפי נוסחאות יתבצעו לפי שלוש נוסחאות ראשיות (פיתוחם בנספח ב' משוואות תנועה והיישנים):

$$\delta = \frac{D}{R}$$

$$\Delta Y = \sin\delta \cdot R$$

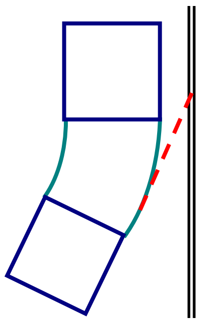
$$\Delta X = R(1-\cos\delta)$$

$$R = \frac{W \cdot V_1}{V_{1-}V_2}$$

ניתן לראות שבאמצעות משוואות אלה נוכל לבצע תיקונים מדויקים במעגל סגור. תיקון במעגל סגור הוא תיקון בו נמדדת הטעות רק פעם אחת ומתבצע תיקון מתאים ללא בדיקה שוטפת. לעומת תיקון במעגל פתוח בו הטעות נבדקת כל הזמן והתיקון משתנה בהתאם (תיקוני יחס, תיקונים פשוטים).

כמו שראינו התיקונים בנסיעה נועדו לפצות על שני סוגי טעויות, זווית ומרחק מאמצע המסדרון. השימוש בנוסחאות יכול להתבצע בשיטות רבות. הדרך הפשוטה ביותר היא לבצע תיקוני זווית ולאחר מכן תיקונים למרכז, אך שילוב של שתי הנוסחאות במקביל יביא לתיקונים מהירים יותר. אפשר גם להחליט לבצע תיקונים עם מהיריות קבועות כאשר הזמן לסיום התיקון משתנה או להפך, הזמן קבוע ובהתאם אליו לשנות את המהירויות.

תיקון זווית באמצעות נוסחאות:



D_1, V_1 – מייצגים את הגלגל החיצוני.
D_2, V_2 – מייצגים את הגלגל הפנימי.

בתיקון הזווית נציב את נוסחאות הרדיוס בנוסחת הזווית ותתקבל התוצאה הבאה:

$$\delta = \frac{D_2 \cdot V_{1-}V_2}{W \cdot V_2}$$

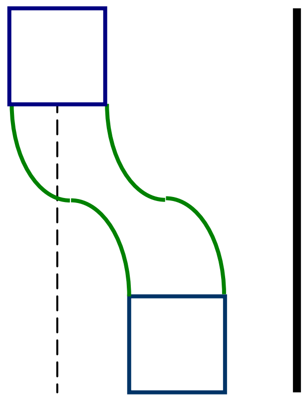
נציב במקום המהירות בגלגל החיצוני את המהירות הרגילה בה נוסע הרובוט מכיוון שאנו רוצים שהתיקונים לא יורידו ממהירות הרובוט.

$$\delta = \frac{D_2 \cdot V_{REG-}V_2}{W \cdot V_2}$$

בנוסחא זו אפשר להציב או את V_2 כקבוע או D_2 כקבוע, תלוי אם נרצה שהתיקון יתבצע לאורך מרחק קבוע, או במהירות קבועה.

<u>תיקונים למרכז:</u>

תיקונים למרכז לפי נוסחאות יתבצעו בעזרת הנוסחא ל- ΔX. נשתמש פעמים בנוסחא פעם אחת ליצרית זווית לכיוון האמצע ופעם שניה בכיוון ההפוך כדי להגיע לאמצע ללא זווית סטיה.



$$\Delta X = R(1 - \cos\delta)$$

$$\Delta X = R(1-\cos(\frac{D}{R}))$$

$$\Delta X = \frac{W \cdot V_1}{V_1 - V_2}(1-\cos(\frac{D_1}{\frac{W \cdot V_1}{V_1 - V_2}})$$

כמו בתיקון הזווית נציב במקום המהירות הגבוהה את מהירות הנסיעה, ונבחר האם מהירות התיקון תהיה קבועה או המרחק שבו יעשה התיקון.

$$\Delta X = \frac{W \cdot V_{REG}}{V_{REG} - V_2}(1-\cos(\frac{D_1}{\frac{W \cdot V_{REG}}{V_{REG} - V_2}})$$

<u>התיקונים שביצענו במסדרונות:</u>

התיקונים שביצענו לבסוף היו שילוב של תיקוני זווית ותיקונים למרכז. הרעיון היה שמאחר ואם הרובוט בזווית אז קריאות המרחק שיתנו החיישנים לא יהיו אמיתיות והתיקונים למרכז עלולים לפעול הפוך, נבצע תיקונים למרכז רק כאשר הזווית בין הרובוט לקירות קטנה.
בהתחלה התוכנה הייתה דוגמת את שני חיישני הצד, מכניסה את הקריאות ואת ההפרש ביניהם (ערך מוחלט) לזיכרון. אם ההפרש היה גבוה מארבע, התוכנה הייתה עוברת לתיקוני זווית, אם ההפרש היה קטן יותר התוכנה הייתה עוברת לתיקונים למרכז.
תיקוני הזווית התבצעו בעזרת שיטת התיקון היחסי (PID), הסתפקנו בביצוע P (תיקון יחס לטעות בלבד) – זאת מאחר ותיקוני הזווית עבדו ללא דופי ולא נדרש להם תיקון נוסף. את המקדם לא חישבנו אלה בנינו טבלה (ראה נספח ו' – מימוש משוואות באסמבלי), של ערכי ההפרש לעומת הערך שצריך להכניס לתיקון. הכיוון אליו צריך לתקן הוכנס לדגל מיוחד בזמן ביצוע פעולת הערך המוחלט על ההפרש, תיקון הזווית בדק ערך זה אוטומטית ולפיו תקן את מהירותו של גלגל ימין או שמאל.
התיקונים למרכז, היו תיקונים פשוטים יותר, לא עשינו להם תיקון יחס מלא אלה רק שתי רמות: אם הרובוט קרוב מאוד לקיר אז הוא יבצע תיקון חזק, אם הרובוט קרוב יותר לאמצע אז הוא ייתן תיקון עדין יותר. אם התוכנה עברה לתיקונים למרכז, אך אין צורך גם בתיקונים למרכז (הרובוט ישר ונוסע באמצע המסדר), נבדק שוב ההפרש בין החיישנים ואם הוא גדול מאחד חוזרת התוכנה לביצוע תיקון זווית. הסיבה לתיקון הזווית הקטנה, היא כדי לשמור על הרובוט בקו נסיעה ישר לחלוטין כאשר הוא נמצא במרכז המסדרון.

<u>התיקונים שביצענו בחדר (עקיבה לאורך קיר):</u>

העקיבה לאורך קיר שביצענו בחדר, מומשה באמצעות שיטת התיקונים הפשוטים. ישנו רק ערך אחד לתיקונים, שמופעלים כאשר הרובוט בזווית לקיר הוא כאשר הוא רחוק או קרוב מדי לקיר.
התחכום בעקיבה היה שהרובוט היה צריך לבחור לבד אחר איזה קיר לעקוב, בחירה זו נעשתה ע"י בדיקת איזה קיר קרוב יותר, ושינוי דגל בהתאם, שאמר לרובוט אחרי איזה קיר לעקוב.

<u>תיקונים במקום:</u>

תיקונים במקום מתבצעים כאשר הרובוט עומד במקום, מטרת תיקונים אלה היא לכוון את הרובוט כך שיקביל לקיר מסוים. התיקונים אותם אנו ביצענו היו פשוטים ביותר – אם יש זווית הרובוט מבצע סיבוב במקום עד שההפרש בין החיישנים שווה לאפס, אז הוא בולם את הסיבוב וחוזר לתוכנה הראשית.

פרק יב' - תפעול מנועים בתוכנה

הרובוט "קרעפלך" נע באמצעות שני מנועים אשר נמצאים בשני הצדדים של הרובוט. מנועים אלו מופעלים על ידי PWM שאת הDuty Cycle שלהם אנו קובעים בתוכנה.

חשוב לעבור על מספר אוגרים הנוגעים לתפעול המנועים:

- DDRP – אוגר זה מגדיר כל ביט ב-PORTP ככניסה או כיציאה, כאשר 1 מסמן יציאה ו-0 מסמן כניסה.
- PORTPP – באמצעות פורט זה אנו שולטים על היציאות של פורט זה ובאמצעותו ניתן לתפעל את המנועים.
- PWCLK – באמצעות אוגר זה ניתן לחבר שני ערוצי PWM של 8 ביט לערוץ אחד של 16 ביט. בנוסף, באוגר זה מגדירים את הPrescaler של שעון A ושעון B.
- PWPOL – באוגר זה מגדירים באיזה שעון אנו רוצים להשתמש לכל ערוץ PWM (שעון B או שעון S1 או S0) ובאיזה לוגיקה (חיובית או שלילית) אנו רוצים להשתמש לכל ערוץ.
- PWPERx – מגדיר את זמן הPeriod לערוץ PWM מספר x.
- PWDTYx – מגדיר את זמן הDuty לערוץ PWM מספר x.
- PWEN – מפעיל את הPWM, באוגר זה רק ארבעת הביטים הימניים מוגדרים והם אחד לכל ערוץ PWM.

כעת נעבור על אתחול הPWM המופיע אצלנו בתוכנה בחלק האיתחולים:

```
PWM_INI:
        MOVB #$FF,DDRP
        MOVB #$00,PORTPP
        MOVB #$08,PWCLK
        MOVB #$0F,PWPOL
        MOVB #$FF,PWPER0
        MOVB #PWDREG,PWDTY0
        MOVB #$FF,PWPER1
        MOVB #PWDREG,PWDTY1
        MOVB #$00,PWEN
        RTS
```

בתחילה אנו מגדירים את PortP כפורט יציאה. לאחר מכן אנו מבטלים את נסיעת המנועים כאשר אנו מאפסים את PortPP. לאחר מכן אנו מגדירים Prescaler של 1/16 של השעון ומגדירים את ה Duty Cycle שהוא למעשה PWDREG/$FF כאשר Pwdreg מוגדר בקובץ הEquates בתור $E6. לבסוף, אנו מגדירים את הPWM כמבוטל (Disabled) וזאת מכיוון שאנו רק בתהליך אתחול ואנו לא רוצים שהמנועים יתחילו לנסוע.

נספח א' – בסיסי ספירה, אריתמטיקה בינארית, שערים לוגיים

בסיסי ספירה

שיטת הספירה לה הורגלנו, קרויה שיטת הספירה העשרונית. בשיטה זה ישנן עשר ספרות בלבד (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0). מיקום הספרה במספר קובע את ערך הספרה. לדוגמא: במספר 239 ערך הספרה 3 הוא 30, ואילו במספר 2390, ערך הספרה 3 הוא 300.

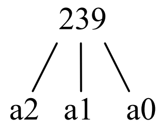
עד עתה קיבלנו כמובן מאליו את שיטת הספירה העשרונית ונמנענו מלהשתמש בשיטות אחרות. עם התפתחות האלקטרוניקה והמחשבים, נזקקנו לשיטת ספירה שונה מהשיטה העשרונית, שיטה שאפשר יהיה לייצגה עד ידי פעולות מיתוג חשמליות, וכידוע – למתג רק שתי אפשרויות – סגור או פתוח.
שיטה כזו המיוצגת על ידי שתי ספרות – 1 ו- 0 נקראת השיטה הבינארית.

ייצוג מספרים בבסיסים שונים

ערך מספר המיוצג בשיטת הספירה העשרונית תלוי במיקום הספרות. לדוגמא ערכו של המספר 239 יתקבל על ידי:

$$N=2*100+3*10+9*1$$

נסמן את הספרות בהתאם למיקומן במספר באופן הבא:



ניתן לרשום באופן כללי את ערכו של מספר:

$$N=a_0*10^0+a_1*10^1+a_2*10^2+\ldots+a_n10^n$$

המספר 10 מכונה בסיס הספירה.

בצורה כזו ניתן לבטא מספר בבסיסים שונים. מספר הספרות הדרוש לספירה בבסיס כלשהו n, יהיה n ספרות.

לדוגמא:
בסיס 10 – דרושות 10 ספרות: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
בסיס 8 – דרושות 8 ספרות: 0, 1, 2, 3, 4, 5, 6, 7
בסיס 2 – דרושות 2 ספרות: 0 ו-1

מעבר בסיסים

מעבר מבסיס כלשהו לבסיס 10

על מנת לבצע את המעבר, נשתמש בנוסחא הכללית:

$$N_b=a_0*b^0+a_1*b^1+a_2*b^2+\ldots+a_nb^n=N_{10}$$

כאשר a_0 הוא הספרה הימנית ביותר a_n, הוא הספרה השמאלית ביותר וb הוא הבסיס המקורי של המספר השונה מ10.
לדוגמא: על מנת להמיר את המספר 673_8 לבסיס 10:

$$673_8=3*8^0+7*8^1+6*8^2=443_{10}$$

מעבר מבסיס 10 לבסיס כלשהו

על מנת לעבור מבסיס 10 לבסיס שונה, יש לבצע את הפעולות הבאות:
לחלק את המספר בבסיס היעד.
לרשום את השארית בצד ואת התוצאה המעוגלת כלפי מטה בהמשך.
להמשיך לחלק עד שהמספר קטן ממספר הבסיס.
לרשום את המספר האחרון עם השארית שרשמנו.
המספר הוא השארית (השארית הראשונה היא הספרה הימנית ביותר והאחרונה היא הספרה השמאלית ביותר).

להדגמה ניקח את המספר 443_{10} מהסעיף הקודם ונמיר אותו לבסיס 8.

השארית	המספר
3	443
7	55
6	6

התהליך היה כדלקמן:
- חילקנו את המספר 443_{10} ב8, התקבלה התוצאה 55.375.
- רשמנו למטה את המספר 55, ולאחר מכן הכפלנו את 55 ב-8 לקבלת השארית (8*55-443=3)
- לאחר מכן המשכנו וחילקנו את 55 ב-8, וקיבלנו 6.875. רשמנו למטה 6 וגילינו שהשארית היא 7 (8*6-55).
- מכיוון שהמספר 6 קטן מהבסיס 8 רשמנו אותו בשארית וסיימנו.
- לכן המספר המבוקש הוא 673_8.

<u>בסיס 16 (Hexadecimal)</u>

ברוב המעבדים הקיימים בשוק כיום משתמשים בעיקר ב3 בסיסי ספירה: בסיס 2 (בינארי), בסיס 10 (דצימלי) ובסיס 16 (הקסאדצימאלי).
השימוש בבסיס 16 הינו רחב כל כך מכיוון שכל ספרה בבסיס 16 מייצגת ארבע ספרות בינאריות ולכן הוא משמש בסיס ספירה נוח יותר לייצוג בינאריים. לכן שתי ספרות הקסאדצימאליות מייצגות 8 ביט, ארבע מייצגות 16 ביט וכך הלאה.
הספרות המשמשות את בסיס הספירה 16 הן:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

דוגמא:

$$F6D2_{16}$$

$$1111\ 0110\ 1101\ 0010_2$$

כפי שניתן לראות, כל ספרה הקסאדצימאלית מיוצגת על ידי ארבע ספרות בינאריות (ארבעה ביטים).

<u>שיטת הגודל והסימן ומשלים-2 של מספר בינארי</u>

אנו רגילים לסמן מספר עשרוני שלילי באמצעות סימן מינוס משמאל למספר. אולם, לצורך שימוש בבסיס בינארי במחשב, אנו צריכים למצוא שיטה כדי לבטא את הסימן (חיובי או שלילי) באמצעות הסמלים 0 ו-1, כי אלה הסימנים היחידים שמחשב מזהה. נהוג לסמן מספר בינארי חיובי על ידי הוספת הספרה 0 משמאל למספר, ומספר בינארי שלילי על ידי הוספת הספרה 1 משמאל למספר.
כדי לבנות את הייצוג הבינארי של המספרים השלמים **החיוביים והשליליים** ,שערכם המוחלט בין 0 ל-7, נזדקק ל-4 סיביות: שלוש מהן יבטאו את ערכם המוחלט של המספרים(החיוביים והשליליים) שבין 0 ל-7 והרביעית – השמאלית ביותר- תבטא את הסימן.
בטבלה רשומים 15 מספרים בינאריים, חיוביים ושליליים, שנבנו בשיטה שתיארנו. המספר 0 נחשב לחיובי.

סיבית הסימן	סיבית הסימן		
		0000	0
1001	-1	0001	1
1010	-2	0010	2
1011	-3	0011	3
1100	-4	0100	4
1101	-5	0101	5
1110	-6	0110	6
1111	-7	0111	7

בין שני מספרים, הרשומים באותה שורה בטבלה, קיים הבדל רק בסיבית הסימן. שאר הסיביות, המייצגות את הערך המוחלט של המספרים, זהות.
שיטה זו להצגת מספרים שליליים נקראת שיטת **הגודל והסימן** .כפי שנראה מיד, שיטה זו אינה יעילה לביצוע פעולות חשבון.

1100+1=1101

מתקבל הצירוף 1101 המסמל את 5-, אולם התוצאה אמורה להיות: 3-=1+4-
לפיכך צריך להפוך את סדר הצירופים בעמודה הימנית כך שהאחרון יהיה לראשון, זה שלפני האחרון יהיה שני וכדומה – ולכן אנו משתמשים בשיטת ה<u>משלים ל-2</u>.

סיבית הסימן			סיבית הסימן
0	0000		
1	0001	-1	1111
2	0010	-2	1110
3	0011	-3	1101
4	0100	-4	1100
5	0101	-5	1011
6	0110	-6	1010
7	0111	-7	1001

נבדוק כעת ‎-3+1=-2‎ :

```
  1101
+    1
  ----
  1110
```

ואמנם, התוצאה שקבלנו עתה היא נכונה.
מהתבוננות בטבלה ניווכח שקיים בה כלל: מחובר אחד גדל ביחידה משורה לשורה והמחובר השני קטן ביחידה משורה לשורה. לכן שווים סכומי הצירופים של 3+(-3), 2+(-2), 1+(-1) וכדומה.
כל אחד משני הצירופים בשורה נקרא <u>משלים-2</u> של השני. מספר בינארי אחד באורך n סיביות, נקרא משלים-2 של מספר בינארי שני (באותו אורך) כאשר סכומם שווה ל-0 ויוצר גלישה. גם שיטת ייצוג המספרים השליליים המתוארת בטבלה נקראת <u>שיטת משלים- 2</u>.

<u>משלים-1 ומשלים-2 של מספר בינארי</u>

הייצוג העשרוני בשיטת משלים ל-2	הייצוג העשרוני בשיטת משלים ל-1			מספר עשרוני
			0000	0
0	-1	1111	0001	1
-1	-2	1110	0010	2
-2	-3	1101	0011	3
-3	-4	1100	0100	4
-4	-5	1011	0101	5
-5	-6	1010	0110	6
-6	-7	1001	0111	7
-7	-8	1000		

מהו הקשר בין צירופי הסיביות הנמצאים משני עברי כל חץ?

כל צירוף מהווה היפוך הספרות של הצירוף הנמצא מצידו השני של החץ, לכן סכום שני הצירופים הוא 1111.
כל זוג מספרים, הקשורים ביניהם על ידי חיצים, נקראים <u>משלימי-1</u>. מספר בינארי נקרא <u>משלים-1</u> של מספר בינארי אחר אם סכומם מורכב כולו מסיביות 1, כלומר, אם סיביותיהם המתאימות הפוכות.

ברוב המחשבים מיוצגים המספרים השליליים בשיטת <u>משלים-2</u>. גם היפוך הסימן של מספר נעשה ברוב המחשבים בשיטה זו.

<u>נתבונן במספרים:</u>

0011	+3
1101	-3

כל אחד ממספרים אלה מהווה <u>משלים-2</u> של השני כי:

```
   0011   +3
+  1101   -3
  -----------
  10000    0
```

כפי שאנו רואים המשלים ל-2 הופך מספר חיובי לשלילי ולהפך כאשר 0 נשאר 0. חסרונה של שיטה זו הוא שהמספר השלילי הגבוה ביותר לא משתנה (בדוגמא 8- נשאר 8-).
לעומת זאת, כל אחד מן המספרים הבאים:

0011
1100

נקרא משלים-1 של השני. בשיטה זו 0 הופך ל1, 2 הופך ל3-, וכן הלאה מכיוון ש0 נחשב למספר החיובי הראשון.

פעולות ושערים לוגיים

שער הוא מתקן הממשש קשר בין משתנים לוגיים. השערים יכולים להיות מסוגים שונים, מכאניים או אלקטרוניים. בשערים אלקטרוניים מייצגים את הערך הלוגי "1" על ידי קיום זרם או מתח ואת הערך הלוגי "0" על ידי העדר מתח או זרם.
בפרק זה נעבור על השערים והפעולות OR, AND, XOR ו-NOT.

הפעולה OR

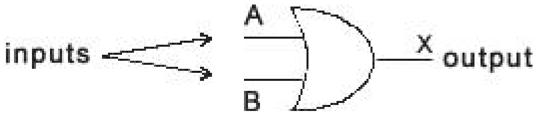
הפעולה OR ("או") מבצעת פעולה בוליאנית אשר מוגדרת על ידי שני משתנים או יותר והמקבלת ערך "1" כאשר לפחות אחד מהמשתנים מקבל את הערך "1".

הסמל המשמש לתיאור פעולת ה-OR הוא +.

כעת נציג את טבלת האמת של פעולה זו. טבלת האמת הינה טבלה אשר מציגה את כל הערכים האפשריים אשר יכולים להגדיר את הפונקציה (Input) ואת תוצאת הפונקציה לגביהן (Output).

X1	X2	$f(X1, X2) = X1 + X2$
0	0	0
0	1	1
1	0	1
1	1	1

שער הOR מסומל בצורה הבאה:



שימושים לפעולה OR: הפיכת ביטים רצויים ל-1 ללא שינוי של הביטים האחרים. לדוגמא:

$46	0	1	0	0	0	1	1	0
$F0	1	1	1	1	0	0	0	0
$46+$F0	1	1	1	1	0	1	1	0

בצורה זו הפכנו רק את הביטים הרצויים (ביטים 5-8) לביטים אשר יוגדרו כ-"1" והתעלמנו משאר הביטים.

הפעולה AND

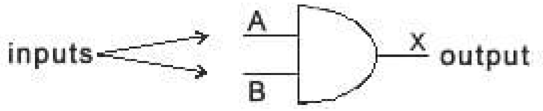
הפעולה AND ("וגם") מבצעת פעולה בוליאנית אשר מוגדרת על ידי שני משתנים או יותר והמקבלת ערך "1" אך ורק כאשר כל המשתנים מקבלים את הערך "1".

הסמל המשמש לתיאור פעולת ה-AND הוא *.

כעת נציג את טבלת האמת של פעולה זו:

X1	X2	$f(X1,X2)=X1*X2$
0	0	0
0	1	0
1	0	0
1	1	1

שער הAND מסומל בצורה הבאה:



שימושים לפעולה AND: הפיכת ביטים רצויים ל-0 ללא שינוי של הביטים האחרים. לדוגמא:

$46	0	1	0	0	0	1	1	0
$F0	1	1	1	1	0	0	0	0
$46*$F0	0	1	0	0	0	0	0	0

בצורה זו הפכנו רק את הביטים הרצויים (ביטים 1-4) לביטים אשר יוגדרו כ-"0" והתעלמנו משאר הביטים.

<u>הפעולה NOT</u>

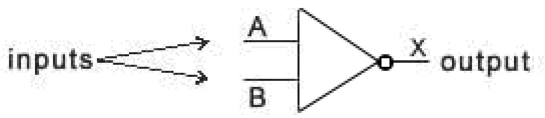
הפעולה NOT ("לא") מבצעת פעולה בוליאנית אשר מוגדרת על ידי משתנה אחד בלבד. הפעולה הופכת את ערך המשתנה. במוצא יתקבל "1" כאשר בכניסה ישנו "0" ולהפך.

הסמל המשמש לתיאור פעולת ה-NOT הוא ¯ ("גג" מעל המשתנה).

כעת נציג את טבלת האמת של פעולה זו:

X	$f(x)=\overline{X}$
0	1
1	0

שער הNOT מסומל בצורה הבאה:



שימושים לפעולה NOT: ביצוע משלים ל-1. לדוגמא:

\$46	0	1	0	0	0	1	1	0
\$46^	1	0	1	1	1	0	0	1

בצורה זו הפכנו את המספר 70 בבסיס 10 ל71-.

<u>הפעולה XOR (Exclusive Or)</u>

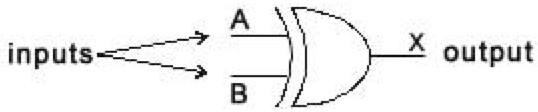
הפעולה XOR מבצעת פעולה בוליאנית אשר מוגדרת על ידי שני משתנים והמקבלת ערך "1" אך ורק כאשר אחד משני המשתנים מקבל את הערך "1".

הסמל המשמש לתיאור פעולת ה-XOR הוא $\oplus$.

כעת נציג את טבלת האמת של פעולה זו:

X1	X2	$f(X1,X2)=X1\oplus X2$
0	0	0
0	1	1
1	0	1
1	1	0

שער הXOR מסומל בצורה הבאה:



שימושים לפעולה XOR: הפיכת ביטים רצויים מ-0 ל-1 ולהפך ללא שינוי של הביטים האחרים. לדוגמא:

\$46	0	1	0	0	0	1	1	0
\$F0	1	1	1	1	0	0	0	0
\$46$\oplus$\$F0	1	0	1	1	0	1	1	0

בצורה זו הפכנו רק את הביטים הרצויים (ביטים 5-8) לביטים אשר ישתנו מ-"1" ל-"0" ולהיפך והתעלמנו משאר הביטים.

חישוב בינארי

כאשר אנו מבצעים חישובים בתוכנה אנחנו לא יכולים פשוט להשתמש בפקודת חילוק או חיבור, אנו צריכים לזכור שאנחנו מתעסקים עם מספרים המבוטאים ע"י שמונה או שש-עשרה ביט שכל ביט הוא ספרה בינארית.

כדי להראות את כל הגורמים בהם צריך להתחשב בעת חישוב בינארי נשתמש בדוגמא של ממוצע.

כאשר אנו מבצעים ממוצע חשבוני אין הבדל בין:

$$Avg = \frac{A+B+C+D}{4} = \frac{A+B}{4} + \frac{C+D}{4} = \frac{A}{4} + \frac{B}{4} + \frac{C}{4} + \frac{D}{4}$$

אך כאשר מדובר בחישוב בינארי המצב שונה. כאשר אנו מחשבים בתוכנה אנו צריכים תמיד לחשוב על ארבעה גורמים - מספרים מסומנים, גלישה (איבוד הביט החשוב ביותר), אי-דיוק (איבוד הביטים הנמוכים) ויעילות.

- גלישה - כאשר אנו מחברים 190 ל- 170 התוצאה היא 260, אך כאן מסתתרת הבעיה: אם אנו משתמשים באוגר של 8 ביט (עד 255) התוצאה "תגלוש" מחוץ לאוגר ובעצם התוצאה תהיה 104.

$$\begin{array}{r} 10101010 \\ +\ \underline{10111110} \\ 101101000 \end{array}$$

$$01101000_2 = 104_{10}$$

```
LDAA    #170            ; Load 17 to A
ADDA    #190            ; Add 190 to A
```

מצב זה יתרחש אם נעשה את הממוצע בצורת:

$$Avg = \frac{A+B+C+D}{4}$$

נניח שהמספרים להם אנו עושים ממוצע מיוצגים ב-8 ביט. אם ננסה לחבר את המספרים בצורה רגילה כמעט תמיד תיווצר גלישה (אלא אם כן מדובר במספרים מאוד נמוכים). ישנן שתי דרכים לטפל בבעיה זו: שימוש באוגר D (אוגר של 16 ביט) או שימוש בשיטה אחרת לביצוע הממוצע.

שימוש באוגר D מתבצע בצורה הבאה:

```
CLRA                    ; Clear A
LDAB    NUM1            ; Load first number to B
ADDB    NUM2            ; Add second number to B
ADCA    #0              ; Add carry to A
ADDB    NUM3            ; Add third number to B
ADCA    #0              ; Add carry to A
ADDB    NUM4            ; Add forth number to B
ADCA    #0              ; Add carry to A
LSRD                    ; Divide D by two
LSRD                    ; Divide D by two
```

שיטה זו אינה מתאימה למספרים מסומנים. שיטה לחישוב הממוצע שתמנע גלישה היא:

$$Avg = \frac{A}{4} + \frac{B}{4} + \frac{C}{4} + \frac{D}{4}$$

בשיטה זו כל מספר מחולק בארבע לפני ביצוע פעולת הסכום ולכן אין סיכוי לגלישה. בשיטה זו גם נוכל לחבר מספרים המיוצגים ב-16 ביט ולהשתמש במספרים מסומנים.

לשיטה זו שני חסרונות: יעילות נמוכה יותר ואי-דיוק.

- יעילות - חילוק כל מספר לחוד דורש זמן מעבד גבוה יותר ומאריך את זמן הביצוע של הממוצע. בדוגמא הקודמת זמן ההרצה של התוכנית הוא 2.25 מיקרו שניות, זמן ההרצה של התוכנית הנוכחית הוא 6 מיקרו שניות.
- אי-דיוק - בכל פעם שאנו מחלקים מספר אנחנו מאבדים את השארית (0-3). אם מבצעים קודם חילוק של כל מספר לפני החיבור הטעות מצטברת (0-12=…+0-3+0-3).

```
LSR     NUM1     ; Divide the first number by two
LSR     NUM1     ; Divide the first number by two
LSR     NUM2     ; Divide the second number by two
LSR     NUM2     ; Divide the second number by two
LSR     NUM3     ; Divide the third number by two
LSR     NUM3     ; Divide the third number by two
LSR     NUM4     ; Divide the forth number by two
LSR     NUM4     ; Divide the forth number by two
LDAA    NUM1     ; Load the first number to A
ADDA    NUM2     ; Add the second number to A
ADDA    NUM3     ; Add the third number to A
ADDA    NUM4     ; Add the forth number to A
```

אם אנו עושים ממוצע למספרים מסומנים נשתמש בפקודה ASR במקום בפקודה LSR.

כמובן שהבחירה באיזה סוג של ממוצע נשתמש תלויה בתנאים בהם אנו עושים את הממוצע. לדוגמא, בתוכנה של קרפלעך ביצענו ממוצע לחישני האינפרה אדום, מכיוון שידענו שתוצאות חישנים אלה לא עוברות את הערך 128 יכולנו לחבר שתי תוצאות ללא חשש שתיווצר גלישה. חישוב הממוצע התבצע בתוך פסיקה ולכן רצינו להשיג יעילות מכסימלית ללא הוספה של אי-דיוקים (אין טעם לממוצע של קריאות חישן אם הוא לא הופך אותן ליותר מדויקות). מסיבות אלה התוכנה משתמשת בנוסחה:

$$Avg = \frac{\frac{A+B}{2} + \frac{C+D}{2}}{2}$$

התוכנה מבצעת ממוצע של שני החישנים הקדמיים בו זמנית כדי לחסוך בזמן עיבוד. בנוסף כדאי היה להשתמש בתוך הפסיקה בהמרה של ארבע פורטים אנלוגיים ולא של כל השמונה כדי לחסוך זמן. ההבדל בין המרה של ארבע לשמונה פורטים הוא 28 מילישניות. בנוסף, אפשר לעשות גם המרה של חישנים יחידים. התוכנה גם מנצלת את המחסנית ושיטת המיעון Auto pre/post decrement/increment indexed addressing. שיטת מיעון זו שימושית מאוד לביצוע חישובים בינאריים מכיוון שהיא חוסכת פקודות וזמן עיבוד.

תוכנת ממוצע החיישנים נראית כך:

```
    JSR         READ_ATD     ; Read Analog sensors
    LDAB        ADR0H        ; Load right sensor read to B
    LDAA        ADR1H        ; Load left sensor read to A
    JSR         READ_ATD     ; Read Analog sensors
    ADDB        ADR0H        ; Add right sensor read to B
    ADDA        ADR1H        ; Add left sensor read to A
    LSRA                     ; Divide A by two
    LSRB                     ; Divide B by two
    PSHB                     ; Push B into stack
    PSHA                     ; Push A into stack
    JSR         READ_ATD     ; Read Analog sensors
    LDAB        ADR0H        ; Load right sensor read to B
    LDAA        ADR1H        ; Load left sensor read to A
    JSR         READ_ATD     ; Read Analog sensors
    ADDB        ADR0H        ; Add right sensor read to B
    ADDA        ADR1H        ; Add left sensor read to A
    LSRA                     ; Divide A by two
    LSRB                     ; Divide B by two
    ADDA        1,SP+        ; Add the average of the two first left
                              ; reads to A
    ADDB        1,SP+        ; Add the average of the two first right
                             ; reads to B
    LSRA                     ; Divide A by two
    LSRB                     ; Divide B by two
```

חישוב בינארי מעניין נוסף שביצענו על קרפלעך הוא ביצוע המקדם ל-P בבקרת המהירות של הרובוט. המקדם ב-PID הוא מצורה: a/b. ישנן שתי דרכים לבצע כפולה במקדם: לחלק ב-B ולכפול ב-A או הגעה למקדם בשלבים.

כאשר אנו מחלקים ב-B וכופלים ב-A אנו מאבדים ביטים רבים בשארית, אנו מחויבים להשתמש בשש-עשרה ביט ופעולת החילוק לבדה דורשת 1.5 מיקרושניות לביצוע.

כאשר אנו מגיעים למקדם בשלבים אנו דואגים שלא יהיה איבוד של ביטים ואנו משתמשים בפקודות פשוטות החוסכות זמן מעבד.

בתוכנה זו המספר אותו אנו רוצים לכפול המקדם נמצא ב-A והמקדם הוא 51/64.

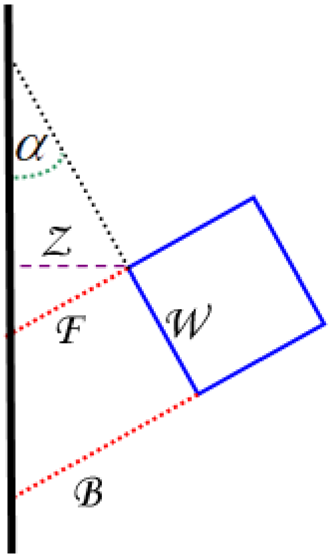
זמן הביצוע של כל התוכנה, שמגיעה למקדם מדויק מאוד, הוא 2.125 מיקרו שניות.

```
    TAB                  ; Transfer value in A to B - Value in A=S
    LSRB                 ; Divide B by two
    LSRB                 ; Divide B by two
    LSRB                 ; Divide B by two    -  B=S/8
    SBA                  ; Subtract B from A -  A=7/8*S
    LSRB                 ; Divide B by two    -  B=S/16
    SBA                  ; Subtract B from A -  A=13/16*S
    LSRB                 ; Divide B by two
    LSRB                 ; Divide B by two    -  B=S/64
    SBA                  ; Subtract B from A -  A=51/64*S
```

נספח ב' – משוואות תנועה וחיישנים

משוואות חיישנים

חישוב הזווית בין הרובוט לקיר (לפי חישני צד)-



F - המרחק שהתקבל בחישן הצדדי הקדמי.
B - המרחק שהתקבל בחישן הצדדי האחורי.
D - המרחק בין החישנים.
α - הזווית בין כיוון נסיעת הרובוט לקיר.
Z – המרחק האמיתי מהקיר.
לפי בנית העזר ניתן לראות ש:

$$\tan(\alpha) = \frac{B-F}{D}$$

$$\alpha \quad \alpha = \arctan(\frac{B-F}{D})$$

$$Z = \cos(\alpha)\cdot F$$

דרכים למימוש משוואה זו בתוכנה – נספח ו' – מימוש משוואות באסמבלר.

יישור תוצאות חיישנים למשוואה ליניארית וליחידות סטנדרטיות:

כאשר אנו משתמשים בחישנים אנו מעדיפים שהתוצאות יתקבלו בצורה ליניארית (משוואה ליניארית מקיימת את הצורה $Y = A\cdot X + B$), וביחידות שאנו רגילים להשתמש בהם: ס"מ (Encoders, חיישני מרחק), Khz (מיקרופון) וכיוצא בזאת.

תהליך הפיכת תוצאות חישן ליחידת מידה:

הערה: הדוגמאות בנושא זה יהיו על המרת תוצאות חישן אינפרה אדום לס"מ.

- השלב הראשון בהפיכת תוצאות החישן ליחידות מידה היא דגימת החישן. דגימת החישן מתבצעת ע"י יצירת טבלה – הסקאלה לפיה אנו רוצים לעבוד (8-60 ס"מ), לעומת תוצאות החישן במרחקים השונים אותה נבדוק. בדוגמא הנוכחית 8-60 ס"מ יהיו הסקאלה לפיה נבדוק תוצאות – אין טעם לקחת דגימות מחוץ לטווח היציב של החישן – תוצאות אלה לא יציבות, חסרות משמעות ויסבכו את המשוואה הסופית ללא צורך. יש לזכור לבצע את המדידות באותם תנאים שבהם ירוץ הרובוט. שינויי תאורה חיצונים, או מתח שונה לחיישן ישנו את התוצאות. בנוסף, כדאי לבדוק מספר תוצאות על אותו מרחק כדי לבדוק את יציבות החישן.

- לאחר שמילאנו את הטבלה בתוצאות החישן לסקאלות היחידות אליה אנו רוצים להמיר, מכניסים את הטבלה לתוכנת גיליון אלקטרוני כדוגמת Excel, ויוצרים גרף פיזור XY. בגרף זה היחידות יהיו ציר ה-X (הציר הלא-תלוי), והתוצאות יהיו על ציר Y (הציר התלוי), כל תוצאה שמדדנו תופיע כנקודה על הגרף. במקרים שבהם משוואת החישן היא ליניארית יופיעו הנקודות כקו ישר. במקרה זה התוכנה יכולה למצוא את המשוואה בשבילנו – מבקשים מהתוכנה להציג קו מגמה ואת משוואת הקו – זוהי המשוואה שממירה בין תוצאות החישן (X) ליחידות מידה (Y).
- במקרה המדובר של חישני האינפרא אדום התוצאה אינה ליניארית, ולכן צריך לגלות מהי הפונקציה המקשרת בין X ל-Y. הדרך לעשות זאת היא ניסוי וטעייה לגבי האופציות השונות של קו המגמה בגיליון האלקטרוני.

אנו לא ידענו על אופציה זו בExcel ולכן השתמשנו בדרך שונה למציאת הפונקציה: יצרנו טבלה חדשה שבה מכניסים את התוצאות של הלוגריתם הטבעי (Ln) של ערכי הטבלה המקורית. לטבלה זו עושים גרף כמו לטבלה הראשונה. אם יוצא שהנקודות יוצרות קו ישר אז שיפוע קו המגמה (ערך ה-A במשוואה הליניארית) הוא החזקה של X במשוואה שאנו מנסים למצוא. כך נדע שהמשוואה היא:

$$Y=AX^n+B$$

בעזרת שתי השיטות אפשר להגיע לכך שמשוואת ההמרה בין תוצאות חישן האינפרה אדום לס"מ היא:

$$Y \approx \frac{1400}{X}$$

את משוואה זו אפשר לממש בקלות בתוכנה:

```
LDAB    ADR0H
CLRA
LDX     #1400
XGDX
IDIV
XGDX
```

۞ דרך נוספת לממש משוואה זו היא באמצעות טבלה (ראה נספח ו' – מימוש משוואות באסמבלר).

יש לזכור שיהיו הבדלים קלים בין חישני אינפרה שונים ולכן יהיו גם הבדלים בנוסחאות שלהם – ניתן לעשות נוסחא שונה לכל אחד או לעשות נוסחא מקורבת לכולם.

משוואות תנועה

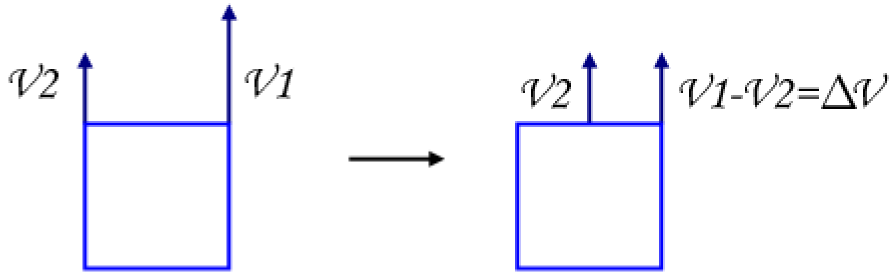
תנועת הרובוט במקרה של אי-שוויון במהירות הגלגלים

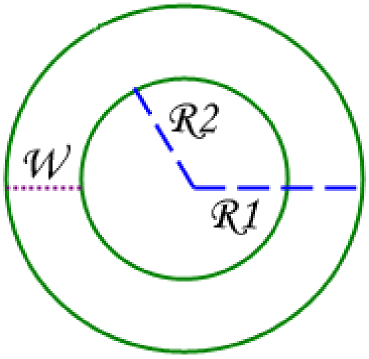
המשמעות של גלגל אחד שנע מהר יותר מהגלגל השני היא שהוא עושה יותר דרך באותו הזמן.

$$V_1 > V_2$$
$$S = V \cdot T$$
$$S_1 > S_2$$

המשמעות של תוצאה זו היא שהרובוט נע במעגל.

אפשר להגיע לאותה תוצאה דרך מכאניקה פשוטה – כל הרובוט נע קדימה במהירות מסוימת – אך הצד שלו נע במהירות גבוהה יותר כך שהרובוט משנה זווית בצורה קבועה – נע במעגל.





פיתוח היחס בן המהירות לרדיוס

W – המרחק בין הגלגלים.
R_1 - רדיוס הסיבוב של הגלגל המהיר.
R_2 - רדיוס הסיבוב של הגלגל האיטי.
V_1- מהירות הסיבוב של הגלגל המהיר.
V_2- מהירות הסיבוב של הגלגל האיטי.

S=V×T
כאשר הרובוט יעשה סיבוב שלם הוא יחזור לאותו המצב (אותו הזמן לשני הגלגלים).

$$2\pi R_1 = T \cdot V_1$$
$$2\pi R_2 = T \cdot V_2$$

צמצום והעברת אגפים

$$\frac{R_1}{R_2} = \frac{V_1}{V_2}$$

נובע מהסרטוט

$$R_1 = R_2 + W$$

הצבה

$$\frac{R_2 + W}{R_2} = \frac{V_1}{V_2}$$

העברת אגפים

$$R_2 + W = R_2\left(\frac{V_1}{V_2}\right)$$

חיסור R_2

$$W = R_2\left(\frac{V_1}{V_2}\right) - R_2$$

הוצאת גורם משותף

$$W = R_2\left(\frac{V_1 - V_2}{V_2}\right)$$

חלוקה ב- $\left(\frac{V_1 - V_2}{V_2}\right)$

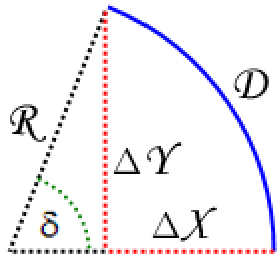
$$R_2 = \frac{W}{\frac{V_{1-}V_2}{V_2}}$$

העלאת V_2 למונה

$$\boxed{R_2 = \frac{W \cdot V_2}{V_{1-}V_2}}$$

$$\boxed{R_1 = \frac{W \cdot V_1}{V_{1-}V_2}}$$

הגענו למשוואות הקבועות את רדיוס הסיבוב של הגלגלים לפי המהירויות שלהם.
δ - זווית השינוי בכיוון תנועת הרובוט (ברדיאנים).



ΔY- העתק הרובוט בציר Y.
ΔX- העתק הרובוט בציר X.
D- אורך הקשת.
R- רדיוס הסיבוב

חישוב זווית השינוי בכיוון הרובוט:

$$\boxed{\delta = \frac{D}{R}}$$

משמעות: זווית הסיבוב (ברדיאנים) שווה למרחק הנמדד בEncoder חלקי רדיוס הסיבוב.

<u>חישוב העתק הרובוט:</u>

$$\boxed{\Delta Y = \sin\delta \cdot R_1}$$

$$\Delta X = R_1 - \sqrt{R_1^{\,2} - (\sin\delta \cdot R_1)^2}$$

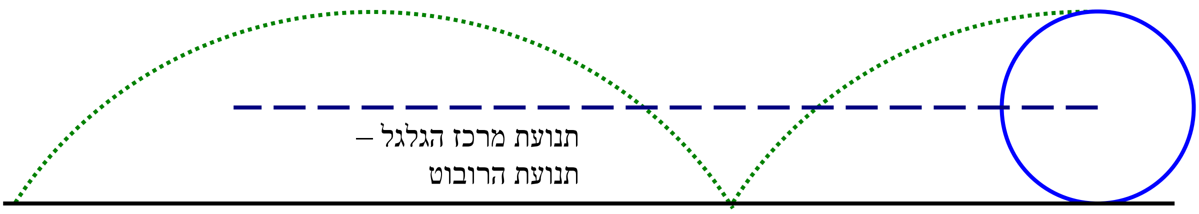
$$\Delta X = R_1 - R_1\sqrt{1 - \sin^2\delta}$$

$$\boxed{\Delta X = R_1(1 - \cos\delta)}$$

נספח ג' – המכניקה של תנועת הרובוט

<u>תנועה ללא החלקה</u>

כאשר הרובוט נע במבוך תנועתו הרצויה היא תנועה ללא החלקה.

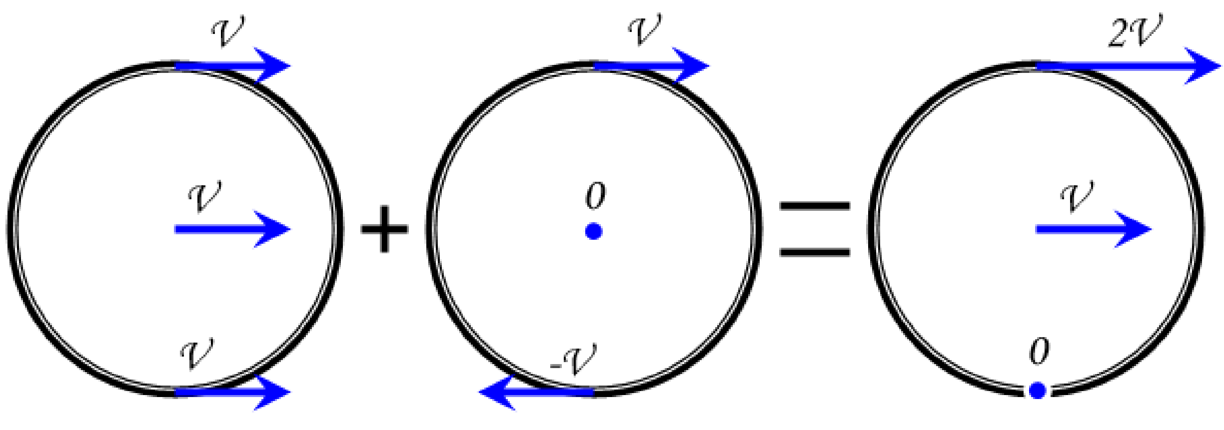


כאשר הרובוט נע ללא החלקה בזמן סיבוב אחד יתקדם הרובוט בדיוק את היקף הגלגל – כך שהזמן שייקח לנקודה לחזור למקומה (לעשות סיבוב שלם) שווה לזמן שייקח לרובוט להתקדם את היקף הגלגל. נכונותו של משפט היא המאפשרת לנו למדוד את המרחק שעבר הרובוט באמצעות הEncoder.

$$\frac{2\pi r}{V_{\text{Center of wheel}}} = T$$

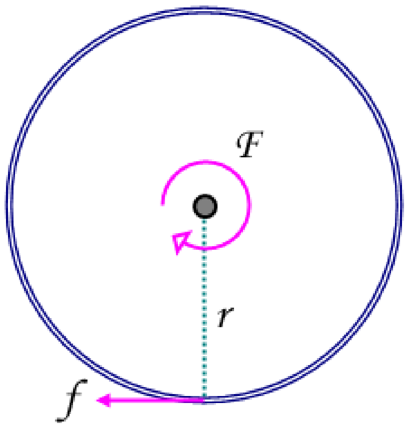
$$\frac{2\pi r}{V_{\text{Point on the outside circle}}} = T$$

$$V_{\text{Center Wheel}} = V_{\text{Point On Wheel}}$$



אפשר לראות שלנקודת המגע אין מהירות, נובע מכך שהגדרנו מצב ללא החלקה.
מתוצאה זו נוכל להסיק על כוח החיכוך שפועל על הגלגל (קינטי לעומת סטטי).

הכוחות הפועלים בתנועה:



הכוח שמניע את הרובוט הוא הכוח שמפעילים המנועים. זהו כוח סיבובי המועבר לגלגל, דרך ציר המנוע לגלגל. כוח זה מתורגם לכוח קווי הפועל בהיקף הגלגל, ככל שרדיוס הגלגל גדול יותר כך יקטן הכוח שפועל בהיקף.

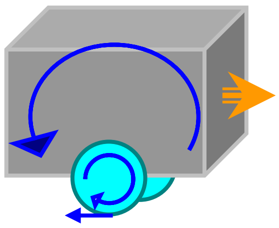
F - הכוח שמפעיל ציר המנוע.
f- הכוח שמפעיל היקף הגלגל.
r - רדיוס הגלגל (המרחק בין ציר המנוע להיקף הגלגל).

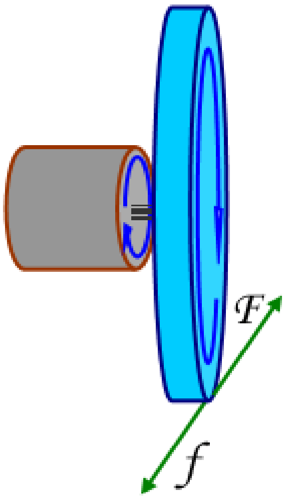
$$F = \mathrm{f} \cdot r$$

$$\mathrm{f} = \frac{F}{r}$$

אפשר לראות שמתקיים יחס הפוך בין רדיוס הגלגל לכוח הפועל בהיקף.

יש לזכור שלכל פועלה יש תגובה וכוח סיבובי פועל חזרה על המנוע, ועל הרובוט דרך חיבור המנוע לבסיס – לכן כאשר מאיצים את הרובוט קדימה - הרובוט עצמו נדחף אחורה מסביב לציר המנוע.





הכוח הקווי שפועל בהיקף הגלגל יוצר חיכוך עם המשטח עליו מונח הרובוט, הגלגל "דוחף" את המשטח אחורה וכך יוצר תנועה קדימה. הכוח המכסימלי שיכול הגלגל להפעיל על הרצפה הוא החיכוך הסטטי המכסימלי כי עם הכוח יהיה גבוה יותר הגלגל יחליק ויפעל נגדו רק חיכוך קינטי – לחיכוך זה יש כוח קבוע שנמוך מהחיכוך הסטטי המכסימלי. במצב זה לרובוט אין שליטה: אין שליטה של המנועים כי הכוח קבוע ולא קשור לכוח המופעל. לא ניתן למדוד Encoders כי סיבובי הגלגל אינם קשורים להתקדמות הרובוט.
על מנת לצאת ממצב החלקה יש לגרום שהמהירות בין נקודת המגע למשטח תהיה אפס כלומר המנוע צריך להפעיל כוח בכיוון ההפוך לכוח שגרם להתחלת ההחלקה (הכוח שעבר את החיכוך הסטטי המכסימלי). הבעיה היא שאין כמעט דרך לגלות החלקה פרט להשוואה בין התקדמות לפי Encoders להתקדמות המעשית (לפי חישוני מרחק), וגילוי הפרש ביניהם.

בתנועת הרובוט יש שני גורמים חשובים אשר מגבילים אותנו: התאוצה המכסימלית והמהירות המכסימלית. התאוצה המכסימלית קובעת כמה חזק נוכל להאט ולהאיץ וכמה מהר ומדויק נוכל לפנות. המהירות המכסימלית קובעת כמה מהר נוכל לסיים את הניווט בדירה.

תאוצה מכסימלית:

התאוצה המכסימלית קשורה לכוח המכסימלי אותו יכולים להפעיל הגלגלים על המשטח, כוח זה נובע משני גורמים: כוח החיכוך המכסימלי שפועל בין המשטח לרובוט (אם נעבור אותו תיווצר החלקה – מצב לא רצוי. בנוסף גם החיכוך הקינטי נמוך בד"כ מהחיכוך הסטטי המכסימלי), והכוח המכסימלי אותו יכולים להפעיל הגלגלים על המשטח.
(הנוסחאות לפי הסרטוט בעמוד הקודם)

החיכוך הסטטי המכסימלי שווה לכוח על המשטח כפול מקדם החיכוך הסטטי של המשטח.

$$f_{\max} = N \cdot \mu_s$$

הכוח הפועל על המשטח שווה למסה כפול מקדם כוח המשיכה.

$$N = m \cdot g$$

מהצבה במשוואה הראשונה נקבל:

$$f = m \cdot g \cdot \mu_s$$

הכוח שהרובוט מפעיל על המשטח שווה לכוח שהמשטח מפעיל על הרובוט – כוח זה מאיץ את הרובוט.

$$f = F$$

הצבה:

$$F = m \cdot g \cdot \mu$$

לפי החוק השני של ניוטון

$$a = \frac{F}{m}$$

הצבה:

$$a = \frac{m \cdot g \cdot \mu}{m}$$

מהתוצאה אפשר לראות שמבחינת כוח החיכוך המסה אינה משפיעה על התאוצה המכסימלית של הרובוט – רק מקדם החיכוך בין המשטח לגלגלים.

$$a = g \cdot \mu$$

הגורם השני שקובע את התאוצה המכסימלית הוא הכוח שבו "דוחף" הגלגל את הרצפה, כוח זה תלוי בשני גורמים: הכוח המכסימלי שיכול המנוע לספק, וקוטר הגלגל (ראה איור קודם). הכוח המכסימלי שיכול המנוע לספק תלוי במנוע עצמו ובמתח ובזרם שמגיעים אליו.
כאשר אין החלקה תאוצת הרובוט נובעת מהמשוואה הבאה:

$$a = \frac{F_{Engine}}{m_{Robot} R_{Wheel}}$$

התאוצה המכסימלית (כשאין החלקה) נובעת מאותה משוואה:

$$a = \frac{F_{Engine\ Max}}{m_{Robot} R_{Wheel}}$$

<u>מהירות מכסימלית</u>

המהירות המכסימלית נובעת משני גורמים: המהירות המכסימלית בה יכול המנוע להסתובב וקוטר הגלגל.
מהירות הרובוט נובעת מהמשוואה הבאה:

$$V = W_{\text{Motor}} \cdot 2\pi R$$

W- המהירות הזוויתית של המנוע (מספר סיבובים לשנייה שמבצע המנוע).
2πR - היקף הגלגל.

$$V_{\text{Max}} = W_{\text{Motor Max}} \cdot 2\pi R_{\text{Max}}$$

אם כן ישנם מספר גורמים בהם צריך להתחשב בהם בעת בניית הרובוט בקשר לתנועה:

- מקדם חיכוך: מקדם החיכוך צריך להיות גבוה ככל האפשר. ככל שמקדם החיכוך גבוה יותר כך יהיה לנו תווך תאוצות גבוה יותר ויקטן הסיכוי להחלקה (מצב לא רצוי).
- מסה: ככל שהמסה תהיה גבוהה יותר התאוצה תהיה קטנה יותר, אך האחיזה על המשטח תהיה גבוהה יותר, כך שאם המנועים מסוגלים לתת כוח גבוה מספיק כדי להגיע עד לחיכוך הסטטי המכסימלי בעצם למסה אין משמעות מבחינת תאוצה.
- קוטר הגלגלים: קוטר הגלגלים משפיע גם על הכוח אותו מפעילים המנועים על הרצפה וגם על המהירות. לכן צריך להתאים את קוטר הגלגלים לסוג המנוע בו משתמשים.
- מנועים: המנועים צריכים לספק את הכוח והמהירות בהם נרצה להשתמש (בהתאמה אם הגלגלים).

נספח ד' – פניות

לא משנה באיזה מסלול ניווט בוחרים הרובוט יצטרך לבצע פניות במסדרון. לפניות חשיבות גבוהה במסלול הניווט וזאת מכיוון שפניות איטיות יכולות להכפיל את זמן הניווט, ופניות לא יציבות עלולות להשאיר את הרובוט במצב בו התיקונים בנסיעה לא יוכלו להחזיר אותו למרכז והרובוט יפגע בקיר.
ישנן שני סוגים עיקריים של פניות, פניות במקום ופניות בתנועה (פניות רחבות).

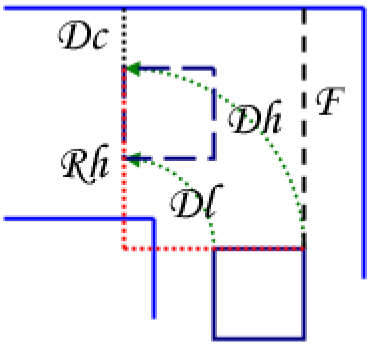
<u>פניות רחבות</u>

פניות רחבות הן פניות המתרחשות ללא עצירה מוחלטת של תנועת הרובוט, אלה דורשות האטה רק של גלגל אחד לצורך יצירת פניה.

הפיתוחים לכל הנוסחאות המופיעות בחלק זה נמצאים בנספח ב', "משוואות תנועה וחיישנים".

את רדיוס הסיבוב נקבע לפי המרחק מהקיר שמולנו.

D_c – המרחק מהקיר הקדמי בו צריך להימצא הרובוט בסוף הפנייה (קו שחור מקווקו דק)
F – המרחק בין הרובוט לקיר הקדמי (קו שחור מקווקו עבה)
V_h – מהירות גבוהה (גלגל חיצוני)
V_l – מהירות נמוכה (גלגל פנימי)
D_h- הדרך שעובר הגלגל החיצוני (הקשת הגדולה) (קו ירוק מקווקו דק)
D_l- הדרך שעובר הגלגל הפנימי (הקשת הקטנה) (קו ירוק מקווקו דק)
W- המרחק בין הגלגלים
R_h – רדיוס הסיבוב של הגלגל החיצוני (קו אדום מקווקו דק)
R_l – רדיוס הסיבוב של הגלגל הפנימי



אנו רוצים שבסוף הפניה הרובוט ימצא באמצע המסדרון

$$D_c = \frac{46 - W}{2}$$

לפי הסרטוט

$$R_H = F - D_c$$

הצבה

$$R_H = F - \frac{46 - W}{2}$$

משוואת הרדיוס לפי המהירויות (פיתוח בנספח ב')

$$R_H = \frac{W \cdot V_H}{V_{H-}V_L}$$

הצבה

$$\frac{W \cdot V_H}{V_{H-}V_L} = F - \frac{46 - W}{2}$$

מאחר ואנו רוצים לבצע פניה מהר ככל האפשר נציב במהירות הגבוהה את המהירות בה אנו נוסעים.

$$V_H = V_{reg}$$

$$\boxed{V_L = V_{reg}(1 - \frac{W}{F - \frac{46-W}{2}})}$$

הגענו למשוואה למהירות שניתן לגלגל הפנימי לפי המרחק מהקיר הקדמי, W ו- V_{reg} קבועים.
יש לשים שמשוואה זו מתאימה גם לפניות צרות מאוד (המהירות תצא אפס או שלילית).

כעת נפתח את המשוואה למרחק שעובר הגלגל לפי המרחק מהקיר (הפיתוח לפי הסרטוט בתחילת הנספח).

הזווית ברדיאנים שווה לקשת חלקי הרדיוס

$$\delta = \frac{D_H}{R_H}$$

$$D_H = \delta \cdot R_H$$

לפי הסרטוט

$$R_L = R_H - W$$

הצבה

$$D_L = \delta(R_H - W)$$

הצבה

לפי הסרטוט

$$R_H = D_F - \frac{46 - W}{2}$$

הצבה

$$D_H = \delta(D_F - \frac{46 - W}{2})$$

$$D_L = \delta((D_F - \frac{46 - W}{2}) - W)$$

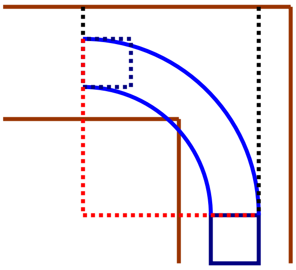
כל הפניות בדירה הן פניות של $90°$, לכן נציב בנוסחא $\delta=\pi/2$:

$D_H = \frac{\pi}{2}(D_F - \frac{46 - W}{2})$
$D_L = \frac{\pi}{2}((D_F - \frac{46 - W}{2}) - W)$

הגענו למשוואה של המרחק אותו נמדוד בEncoders כדי לדעת מתי סיימנו פנייה של $90°$.
אפשר לראות שככל שנתחיל לפנות רחוק יותר מהקיר כך נצטרך להוריד פחות ממהירות הגלגל. כך נאבד פחות ממהירות הרובוט, ונוריד את הזמן שלוקח לרובוט לסיים ניווט שלם.

ישנם גם יתרונות לבצע פניה צרה יותר:

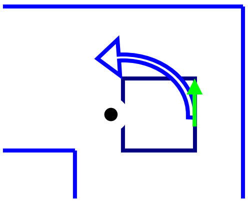
- ככל שמבצעים פניה רחבה יותר הרובוט מתקרב יותר לפינה ואם מתחילים ממרחק גבוה מדי הוא יפגע בה.

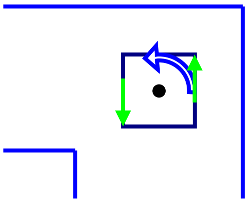


- חישני האינפרה אדום מדויקים יותר ככל שהמרחק קצר יותר, כך שאם נבחר להתחיל לפנות במרחק גבוה, לא בטוח שתמיד נתחיל את הפניה מאותו מקום.

<u>פניות במקום</u>

פניות במקום הן הפניות הבטוחות ביותר, ואפשר גם לבצע אותן כך שאיבוד הזמן לעומת פניות רחבות יהיה מינימאלי. פניות במקום, לפי שמן, מבוצעות במקום, הרובוט בולם בהגיעו לפניה, מסתובב תשעים מעלות במקום, בולם ומאיץ חזרה בכיוון החדש, לאחר הבלימה השנייה. פניה במקום אפשר לבצע כשהציר עובר במרכז הרובוט (רק כאשר הגלגלים במרכז) כשגלגל אחד מפעיל כוח בכיוון הפוך לגלגל השני. או שאפשר לבצע פניה כשהציר הוא אחד הגלגלים - פשוט בולמים את הגלגל.





יש לזכור שכל עוד הרובוט פונה במרחק הנכון מהקיר הקדמי, הוא יסיים את הפניה באמצע המסדרון שאליו הוא פנה, לא משנה אם היה קרוב לקיר לפני הפניה. לעומת זאת אם הרובוט מגיע לפניה בזווית אז לאחר ביצוע פניה של $90°$ מעלות יישאר הרובוט בזווית ביחס לקיר. אם התיקונים בתנועה לא מדויקים מספיק והרובוט מגיע לפניות אם זווית יש צורך לבצע תיקון במקום לאחר הפניה.

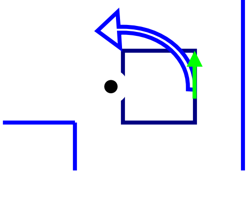
<u>פניות שונות:</u>

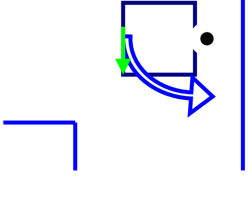
ישנן שיטות נוספות לבצע פניות, רובן איטיות ומסובכות יותר אך לחלק יש יתרון כאשר בפניה מוצב מכשול.

פניות XY: הרעיון מאחורי פניות XY הוא שבמקום לבצע פניה, הרובוט יכול לנוע על שני צירים מאונכים X ו- Y, וכאשר רוצים לבצע פניה פשוט נעים בציר השני.

שיטה זו ניתן לבצע בדרכים שונות: הורדת רגל שמסובבת את כל הרובוט לכיוון החדש (סוג של פניה במקום), סיבוב הגלגלים לכיוון החדש (הנעה סינכרונית).

פניות לאחור: פניות לאחור מתבצעות כמו פניות בלימה במקום פרט להבדל אחד, במקום לבלום את הגלגל הפנימי ונסיעה קדימה עם הגלגל החיצוני, בולמים את הגלגל החיצוני ונוסעים אחורה עם הפנימי.





היתרון של שיטה זו הוא שאם יש בפניה מכשול הרובוט קודם יעלה על המכשול ואז יפנה, וגם הפעלת הכוח תהיה עם הירידה של המכשול ולא נגד – פחות סיכוי להחלקה ויותר דיוק בביצוע הפניה.

נספח ה' - בעיות

בעת עבודתנו על הרובוט נתקלנו במספר בעיות שפגעו ועכבו את עבודתנו. בעיות נבעו מגורמים שונים, בעיות שנבעו מהסביבה בה עבדנו, בעיות שנבעו משגיאות שעשינו בעצמנו ומחוסר תכנון, ובעיות שנבעו מהציוד בו השתמשנו.

<u>בעיות מנהלתיות:</u>

<u>עיכוב העבודה:</u>

בעיה מאוד גדולה שפגעה בעבודה שלנו הייתה, שלמרות שאנחנו היינו מוכנים להתחיל לעבוד עוד לפני תחילת השנה, החלקים לבניית הרובוט הגיעו רק כשלושה חודשים לאחר תחילת השנה. עיכוב זה דחה מאוד את תחילת העבודה על קרפלעך וצמצם מאוד את הזמן שהיה לנו כדי לבנות ולתכנת אותו.
כדי לא לבזבז לגמרי את הזמן עד להגעת החלקים, פירקנו רובוט ישן (לידי מקבת) והשתמשנו בחלקיו לבניית רובוט חדש המכונה "שכטר". בניה זו נתנה לנו ניסיון בבניה ובתכנון רובוט, לאחר סיום הבניה התחלנו לתכנת על שכטר, ואף הגענו לרמה בה שכטר ביצע ניווט פרמיטבי (תיקונים בתנועה, ובחירת פניות). בזמן בניית שכטר ואף לפני התאמן צוות התוכנה בתכנות הרובוט במפר של שחר מנדלוביץ'.

<u>בעיית המחשבים הניידים:</u>

מחשב נייד הוא מצרך חובה לכל קבוצת רובונר בתחרות: הוא מאפשר לתכנת את הרובוט ליד הזירות בתחרות, מעבר מהיר מזירה לזירה, וכמובן שכל להעבירו ממקום למקום הרבה יותר מאשר מחשב-שולחני. לקבוצות הרובונר מאהל-שם לא היו מחשבים ניידים ולכן לתחרות הארצית בבאר-שבע נאלצנו לסחוב מחשבים-שולחניים. המחשב האישי שעליו עבדה הקבוצה שלנו התקלקל בהעברה לבאר-שבע, ולא הצלחנו להדליק אותו. במזל הצליחו אנשי הקבוצה לגלות ולתקן את הבעיה שחזרה והופיעה כמעט בכל הדלקה של המחשב לאחר מכן. בנוסף מכיוון שעבדנו על מחשבים-שולחניים שדורשים מקור-מתח לא היינו יכולים לעבור מזירת התחרות לזירה שהבאנו איתנו כאשר נוצר תור על הזירה.
גם לארה"ב נאלצנו לסחוב את המחשבים השולחנים, אבל למחשב של קבוצתנו לא טרחנו לסחוב מסך. פעולה זו הוכחה כחכמה כי ברגע הגיענו לבית-הכנסת בו עבדנו לפני התחרות השיג לנו המנחה אלי קולברג מסך דרכו יכולנו לעבוד בימים ספורים אלו. זאת ועוד, לפני הטיסה ארגנו כך שנוכל להפעיל את הרובוט ממחשב-נייד השייך לאחד מחברי הקבוצה. הבעיה בחיבור הרובוט למחשב זה הייתה שלמחשב לא הייתה יציאת RS-232 הנדרשת כדי לחבר את הרובוט למחשב. כדי להתגבר על הבעיה נאלצנו לקנות מעביר USB to serial , אך למרות מחירו הגבוה של המעביר הוא גרם בעיות רבות מכיוון שלא היה ניתן להגדירו בצורה טובה תחת מערכת ההפעלה הקיימת על המחשב המדובר. הבעיה נפתרה בשלהי יום האימונים לפני התחרות בארה"ב כאשר הצלחנו למצוא תוכנה שמתגברת על הבעיות בהגדרת המעביר וכך יכולנו להעביר תוכנה ללא ניתוקים אקראיים מהרובוט, והיתקעות תוכנת ההורדה כל חמש דקות.

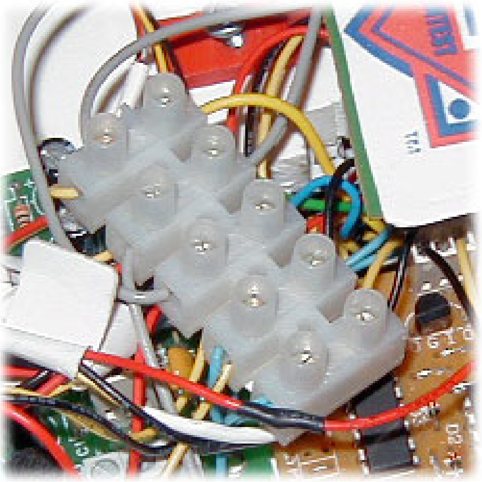
<u>בעיות תכנון:</u>

<u>מאוורר:</u>

מתוך הנחה שהמאוורר הוא חלק טריוויאלי שאפשר להכניס כמעט בכל מקום, דחינו את התקנתו עד לשבועיים לפני התחרות. בעת התקנתו התגלו שתי בעיות: הבטרייה וזרימת האוויר.
הבטרייה שיועדה למאוורר הייתה כבדה מדי וגרמה לשינויים בתנועת הרובוט וגובהה גרם לכך שלא הייתה אפשרות להכניס אותה מתחת לבסיס ובמרכז במקום שלא תגרום לשינויים. בזבזנו זמן יקר בניסיון להתאים מחדש את הניווט, לחבר את המאוורר לבטריות של 9 וולט ובהמצאת שיטות למנוע מהמאוורר "לסחוב" את כל זרם החשמל משאר הרובוט, עד שהחלטנו לרוץ בתחרות עם ספק ולהשתמש בבטרייה של הרובוט כדי להפעיל את המאוורר לבד. בעיה זו גזלה מאיתנו ימים רבים ממש לפני התחרות וגרמה לכך שלא נספיק לסיים את תוכנת הכיבוי לפני התחרות, ובעצם הייתה גורם מקדים לבעיה שקרתה לנו בתחרות.
לאחר התחרות הארצית, כאשר הזמן לתחרות העולמית היה גדול, פנינו ליבואן בטריות והזמנו בטריית ניקל-מטאל הידריד. בטרייה זו נכנסה בקלות מתחת לרובוט, והיו לה יתרונות רבים נוספים על בטרית העופרת בה תכננו להשתמש. (יתרונות בטריית ניקל מטאל הידריד על עופרת בפרק "פירוט חלקי הרובוט").
זרימת האוויר: מיקום המאוורר בתוך הקיר הקדמי של הרובוט גרם לכך של התאפשרה זרימת אויר מאחורי המאוורר, ובעצם לא היה למאוורר אויר לדחוף. במקום לדחוס אויר קדימה דחף המאוורר אויר לצדדים, ובעצם לא הייתה שום אופציה שבמצב זה יצליח לכבות את הנר. הפתרון לבעיה היה ברור וביצענו אותו מיד עם גילוי הבעיה כדי לא לבזבז זמן יקר. קדחנו מספר חורים גדולים בקיר בו הוכנס המאוורר, והרחקנו את המאוורר מהקיר. פעולות אלה יצרו אפשרות לזרימת אויר מאחורי המאוורר, ואפשרו לרובוט לכבות נר ממרחק שעולה על חצי מטר.

<u>מיקום כרטיס הInterface:</u>

בתכנון הראשוני של קרפלעך תכננו לשים את כל החישנים ואת כרטיס הInterface על גג הרובוט. אך לאחר בניית הרובוט התברר שבגלל גודל המחזיקים של חיישני האולטרה סוני לא נשאר מקום לכרטיס הInterface.
בתור פתרון זמני הורדנו את כרטיס הInterface לשכב על צד הרובוט. פתרון זה היה פתרון זמני מכיוון שכרטיס הInterface היה חשוף והיה יכול להיתקע בקיר ממקומו ובנוסף הוא חסם את הגישה לכרטיסים האחרים שהיו בתוך הרובוט. לאחר כשבועיים של עבודה עם כרטיס הInterface במיקום זה הציע אילן מהקבוצה המקבילה לסייע לנו לבנות רובוט חדש. בעזרתו סיימנו לבנות את הרובוט בפחות מארבע ימים כאשר את כרטיס הInterface מיקמנו מתחת ל"גג" בו יצרנו "חלון" שאפשר חיבור של החישנים לכרטיס הInterface. אך גם במיקום החדש התגלו בעיות, למרות שהחלון אפשר גישה להכנסת החיישנים והמנועיים הוא לא אפשר גישה ל-PortDLC. כדי לפתור בעיה זו התקנו הארכה לפורט זה באמצעות מהדק חשמל.



מהדק החשמל בו השתמשנו

<u>בעיות ציוד:</u>

<u>כרטיסי המנועים:</u>

כרטיסי המנועים החדשים בהם השתמשנו דרשו שלושה מתחים: 16V מהבטרייה, 12V מהמיצב, ו-5V מהמיצב להפעלת המעגלים הלוגיים. תכנון זה של הכרטיסים יצר שתי בעיות:

- ריבוי חוטים: כל כרטיס קיבל שישה חוטים מכרטיס החשמל (כולל שלושה חיבורים לGND). כדי לצמצם כמות זו של חוטים יצרנו כרטיס חיבור מיוחד בו חלקו הכרטיסים את המתחים ויציאה GND אחת. שיטה זו צמצמה את כמות החוטים שעוברים מכרטיס החשמל לכרטיסי המנועים משנים-עשר לארבע בלבד. בנוסף, שיטה זו גם אפשרה ניתוק מהיר של הכרטיסים, וכך במקרה של תקלה היינו יכולים להוציא ולהכניס את הכרטיסים תוך מספר שניות.
- סדר המתחים: כרטיסי המנועים לא רק שדרשו שלושה מתחים, אלה גם דרשו הפרש מופע בין מתח ה5v למתח ה16v. לכן, אם מתח ה16v היה מחובר קודם, הזרם היה חוזר לתוך המייצב, והיה נוצר מצב של קצר – הרובוט היה מושך את כל הזרם מהספק והמיצבים היו מתחממים מאוד.

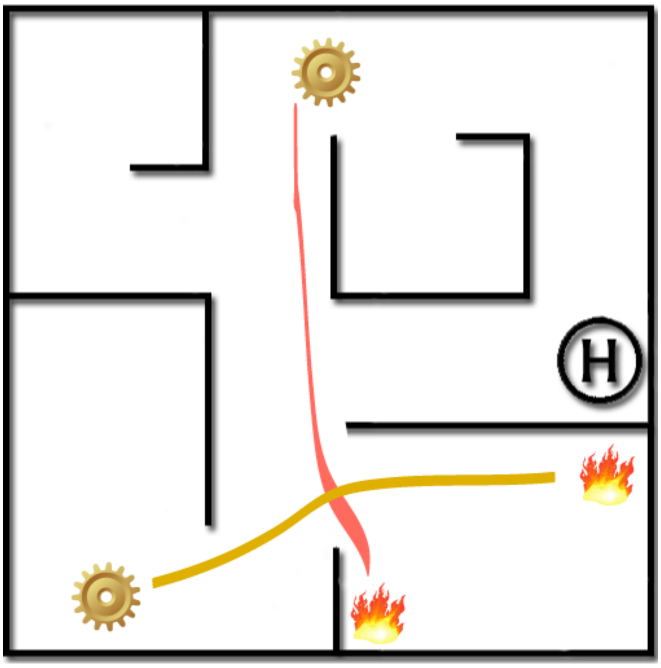
על מנת להתגבר על בעיית המתחים, התקנו בתחילה מתג נוסף לחיבור של השש-עשרה וולט, כאשר היה צורך להקפיד על סדר ההדלקה. הבעיה שנוצרה הייתה שלפעמים היה נוצר נתק רגעי בחיבור המתח של כניסת ה5v והיה נוצר קצר. במשך שבועות פשוט עבדנו תוך תשומות לב מכסימלית לספק על מנת לבדוק שהוא לא עובר למצב של Current Limit – כלומר מצב של קצר, עד שהחלטנו להתמודד עם הבעיה. פירקנו את כרטיס החשמל ובו עברנו על כל ההלחמות על מנת למצוא את החיבור הרופף ובנוסף הלחמנו מחדש חוטים שנראו חשודים. אך כל זאת ללא תועלת, הרובוט המשיך ליצור קצרים בצורה אקראית. לבסוף בדרך של מזל גילנו שהקצרים נוצרים כאשר המחבר שחיבר את החשמל מהספק לרובוט זז לעיתים כאשר הרובוט נע. לאחר זיהוי הבעיה החלפנו את המחבר והבעיה נפתרה.
במהלך שבוע ההכנה לתחרות העולמית חזרה הבעיה, ושוב עברנו על כל החוטים ועברנו על כל ההלחמות וההברגות של הרובוט – עד שבעזרת מזל ראינו שהספק מגיע לCurrent Limit גם כאשר אין רובוט בקצה המחבר – ושוב החלפת המחבר פתרה את הבעיה.

כדי לחסוך את ההפעלה בשני מתגים, וכך לחסוך זמן וקצרים שנגרמים מטעויות הפעלה, החלפנו את כרטיס החשמל בכרטיס חדש עליו התקנו מעבכב-מתח. מעבכ המתח דאג שמתח ה-16v יחובר תמיד לאחר מתח ה-5v. גם אם היה נוצר מצב בו התנתק מתח ה-5v לשנייה ונוצר קצר, מעבכ המתח היה מנתק את מתח ה-16v ומחבר אותו מחדש, כך שלא הייתה סכנה שהרובוט יכנס למצב קבוע של קצר.

<u>חישנים:</u>

<u>חישן Uvtron:</u>

חישן הUvtron תפקד כראוי וזיהה את הנר ממצבים שונים ומשונים, אפילו כאשר הרובוט היה מצוי מחוץ לחדרים מסוימים, דבר שחסך מאיתנו את הכניסה אליהם לשווא.
הבעיה היחידה עם חיישן זה הייתה שהוא היה רגיש מדי לתאורה אולטרא-סגולה ונוצר מצב שכאשר בדקנו אם יש נר בחדר מספר 1 ודלק נר בנקודה ספציפית בחדר מספר 4 החישן זיהה נר והרובוט התחיל כיבוי בחדר מספר 1. אותו הדבר קרה כאשר הרובוט בדק את חדר מספר 4 והיה נר במקום ספציפי בחדר מספר 3.



באיור זה ניתן לראות את שני המצבים, כאשר את הרובוט מסמל גלגל שיניים ואת הנר מסמלת אש. הקווים בצבעים השונים מסמנים את מסלול האור האולטרא סגול אל חיישן הUvtron אשר נמצא על הרובוט.

את הבעיה פתרנו בצורה פרטית לכל חלק. בחלק הראשון פתרנו את הבעיה בכך שהפסקנו את הקריאה מהחישן לפני שהוא מגיע לקטע בו הוא קולט את הנר מחדר מספר 4.
בחדר מספר 4 במקום לקרוא משני החישנים במשך הכניסה לחדר קראנו רק מהשמאלי (החישן הימיני הוא זה שקלט את הנר מחדר מספר 3). כדי שהחישן השמאלי יקלוט את כל החדר בצורה בטוחה החלפנו את הסיבוב במקום של 180 מעלות שביצענו אחרי הסריקה כדי לצאת מהחדר. הרובוט עושה סיבוב של

110 מעלות ימינה על מנת שהחישן יראה את כל החדר, סיבוב של 20 מעלות חזרה למקרה שרהיט חסם את החזר האור האולטרא-סגול, הפסקת הקריאה בחישן וסיבוב נוסף של תשעים מעלות כדי לסיים 180 מעלות.

בעיה נוספת שיש לחישן האולטרה סגול היא שהנורה הקולטת מחוברת לכרטיס באמצעות שתי רגלי מתכת דקות שנשברות בקלות. כדי למנוע את שבירתן הכנו חתיכת PVC שמתלבשת על הנורה ומונעת הזזה וכיפוף של הרגלים. מתקן זה נעלם תוך ימים ספורים ומחוסר זמן לא הוחלף באחר, ולכן כאשר הרובוט נחת בארה"ב גילנו שאחת מהרגלים נשברו. ניסיונות חוזרים ונשנים להלחים מחדש את הרגל נכשלו, ולבסוף החלפנו את הנורה השבורה בחדשה.

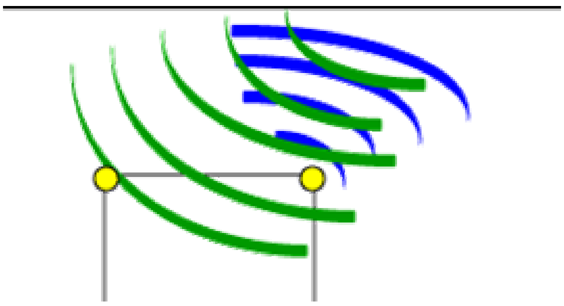
<u>חישני אינפרה-אדום:</u>

הבעייתיות בשימוש בחישני האינפרה נובעת מדרך הפעולה שלהם. חיישני האינפרה אדום מחזירים מתחים בין אפס לחמש וולט כדי לייצג את המרחק. החיישן בעצם יוצר התנגדות לחמש וולט שהוא מקבל וכך משנה את הערך המוחזר. הבעיה בשיטת פעולה זו היא בכך שאם במקום 5V נכנס 4.9V הערך של כל התוצאות ירד. בעיה זו הייתה מאוד חמורה אצלנו מכיוון שכרטיס הInterface בו השתמשנו לא שמר על מתח יציאה של חמש וולט. כאשר חוברו אליו חיישנים שונים, עלה או ירד המתח מחישן לחישן, כך נוצר מצב שבו כל פעם שהוספנו או הורדנו חישן השתנו תוצאת החיישנים.
בעיה זו נפתרה כאשר סיימנו לחלוטין את בניית הרובוט והחלטנו על מערך חישנים סופי. אך עדיין הייתה לנו בעיה – מדי פעם היו נופלות תוצאות חיישני האינפרה אדום וגורמות לרובוט לפנות כאשר הוא היה עדיין באמצע המסדרון. בעיה זו פתרנו גם בתוכנה וגם בחומרה. בתוכנה בנינו תוכנית הדורשת 40 אישורים מכל חישן קדמי בטרם ביצוע פניה. בחומרה עשינו בדיקות לברר ממה נגרמות הקראיות השגויות וגילנו שלפעמים נופל המתח לחישן, בעקבות ניסויים לשחק אם חלק שונים ברובוט בזמן הבדיקה הגענו למסקנה שמקור הבעיה הוא בכרטיס המנועים. הוצאת הכרטיס ובדיקה מדוקדקת שלו הובילה לתגלית שחיבור ה-GND על אחד מכרטיסי המנועים היה מוברג על ציפוי החוט ולא על החוט עצמו, מה שגרם לכך שבעת תזוזה החוט היה יכול להתנתק לשברירי שנייה, מה שגרם לשינוי המתח שיצא מהמייצב של החמש וולט. תיקון בעיה זו לא פתר את הבעיה לחלוטין, לכן החלפנו את חיבור הInterface, שנעשה באמצעות Cramping לחיבור בעזרת ברגים. לאחר בדיקת והחלפת החיבורים ווידוי התוצאות בתוכנה, לא פנה הרובוט באמצע מסדרון, פרט למקרים חד-פעמים לא מוסברים.

<u>חישנים אולטרה-סוניים:</u>

החישנים האולטרה סוניים שירתו אותנו נאמנה: הם נתנו תוצאות קבועות ולא משתנות מיום ליום, ולא התקלקלו (על קרפלעך נמצאים ארבעת החישנים המקוריים). אך כדי להגיע למצב זה היינו צריכים לגלות בעיה רצינית בתפעול החישנים.
כשהתחלנו להשתמש בחיישנים גילנו שהתוצאות מדויקות מאוד פרט לכך שאחת לעשר קריאות בערך מתקבלת תוצאה שגבוהה מהתוצאה הנורמאלית פי שתיים. לכן, יישמנו שיטות שונות לבטל את הקריאה הגבוהה בתוכנה: ממוצע של קריאות, הוצאת חציון והוצאת המספר הנפוץ ביותר. שיטות אלו עזרו לביטול הקריאה הקיצונית אך עיכבו מאוד את ביצוע התוכנה וזאת מכיוון שקריאה של חישן אולטרה-סוני לוקחת זמן רב ולכן ביצוע פעולות סטטיסטיות שדורשות ארבע קריאות ואף יותר, אינן מתאימות לביצוע תיקונים בנסיעה שאמורים להתרחש בתדירות גבוהה ובמהירות.

בעקבות הכישלון לסידור הבעיה בתוכנה, היינו חייבים לגלות מה גורם לבעיה כדי לפתור אותה בחומרה. ביצענו ניסויים רבים כדי לראות מה משפיע על הקריאות, אבל לא גילינו שם גורם קבוע, ומכיוון שהקריאה החריגה הופיע לעתים נדירות היה קשה לראות מה משפיע עליה.
לאחר ביצוע ניסויים אין ספור הועלתה השערה בקשר למקור הבעיה: כאשר קוראים מחישן אולטרה סוני החיישן שולח גל אולטרא-סוני. על מנת לחסוך מקום על המעבד ההוראה לבקשת קריאה הייתה מגיעה יחדיו לכל החישנים האולטרה סוניים מאותה נקודה, כך שכאשר קראנו מחישן אולטרה-סוני אחד, פלטו שלושת האחרים גם כן גל ופשוט לא התייחסנו אליהם.
ההעשרה שהועלתה הייתה שחישן אחד, מסיבות שעד עכשיו לא ברורות, קולט את ההד מפולס שיוצר חישן אחר במקום את ההד שלו, וזאת כפי שניתן לראות באיור:



עיגול צהוב – חיישן אולטרא-סוני.
גל כחול – הגל היוצא מהחיישן הימני אחורי.
גל ירוק – הגל החוזר מהחיישן המגיע גם לחיישן הימני קדמי.

בעקבות השערה על מקור הבעיה ביצענו ניסוי: חסמנו אחד מחישני הצד בעת קריאת החישן השני ובדקנו את הקריאות. לאחר קריאה מהחישן כמה עשרות פעמים ובדיקה שהפעם הבעיה נפתרה, השתכנענו שהשערה נכונה.
הפתרון לבעיה היה פשוט: עשינו יציאה אחת לחישנים הקדמיים ויציאה אחת לחישנים האחוריים, לאחר פעולה זו לא היו לנו יותר בעיות עם חיישנים אלו.
בעיה נוספת שהתגלתה בקשר לחישנים האולטרה סוניים הייתה כאשר ניסינו למקם אותם קדימה (כלפי כיוון תנועת הרובוט) כגיבוי לחישני האינפרה-אדום. בדקנו במצב מנוחה מהי קריאת החישן במרחק בו אנו רוצים לפנות, והכנסנו ערך זה לתוכנה. אך כאשר הרובוט הגיע לפניה הוא פנה במרחקים רנדומאליים. לבעיה זו לא מצאנו פתרון, פשוט השתמשנו בחישני האינפרה-אדום כחישנים הפונים קדימה.
הבעיה ככל הנראה נגרמת מהחזרות של הפינה, ובשינוים במיקום הרובוט בין שליחת האות לקבלתו.

<u>הבעיות בתחרות הארצית:</u>

במקום לתאר את כל התקלות שהתרחשו בתחרות הארצית נתאר את תהליך העבודה שביצענו מרגע הגיענו לבאר-שבע עד לתחרות עצמה.
ברגע שהגענו לבאר-שבע, התחלנו את פריקת המחשבים והציוד שהבאנו ואת הרכבת הזירה. בזמן זה גם הרכבנו וכיוונו את המיצב שהמיר את המתח מהבטרייה לשימוש המאוורר (מ16v ל6v). הבעיה הראשונה שנתקלנו בה הייתה שהמחשב שהבאנו סירב להידלק (יותר על בעיה זו בחלק בעיות סביבה).

לאחר שפתרנו בעיה זו התחלנו להריץ את חלק כיבוי הנר של תוכנת הרובוט, עבדו כשעה על שיפוצים אחרונים לתוכנה וכיוון מדויק של חישני הPyro. בזמן ההרצה גילינו שחישן Pyro אחד קורא פחות טוב מהשני ולכן שינינו את הערכים שלו. לאחר שראינו שכיבוי הנר עובד טוב, עברנו לעבוד על זירות התחרות כדי להתאים את הניווט לזירות התחרות. עבדנו כשעתיים כדי להתאים את הניווט לזירות החדשות, במהלך זמן זה ראינו גם שהרובוט שלנו מהיר בהרבה מהרובוטים האחרים ועושה תוצאות טובות מאוד ביחס לרובוטים האחרים. באותו הזמן נוצרה הבעיה הראשונה: הספק הונח על הרצפה מכיוון שלא היו שולחנות ליד זירות התחרות. החוט שחיבר את הרובוט לספק עבר על הכפתור שקובע את המתח שיוצא מהספק. עקב כך המתח היוצא מהספק השתנה מ16v ל22v. תוך מספר שניות ראינו את השינוי וכיבינו מיד את הספק. כאשר ניסינו להריץ שוב את הרובוט לבדיקה גילינו שמנוע אחד לא פועל, הנחנו שנשרף כרטיס מנועים, ובדיקת מתחים פשוטה הראתה שההשערה נכונה. פנינו מהר להחלפת הכרטיס השרוף באחר. מזלנו היה שמספר ימים לפני התחרות החליפה הקבוצה השנייה מאהל-שם את כרטיסי המנועים שלהם בדגם ישן, ולכן היו להם שני חלפים בדוקים. ברגע שסיימנו את החלפת הכרטיס השרוף חזרנו להריץ את הרובוט וגילינו את הבעיה השנייה: מחבר ה-RS-232 הנמצא על הרובוט התנתק מכרטיס המעבד ולכן לא היינו יכולים להעביר את התוכנה אל הרובוט. מאחר והחיבור היה רופף עוד קודם הבאנו חלפים לחלק זה, אך הבעיה עדיין הייתה גדולה מכיוון שההלחמה של חלק זה מורכבת מאוד ועדיין היה צורך להוציא לגמרי את החלק השבור ללא אמצעים מקצועיים. לבסוף, הצלחנו לשכנע מורה לאלקטרוניקה מבי"ס היובל, אורי דים, לבצע את ההלחמה המסובכת בשבילנו. לאחר כחצי שעה של עבודה הוא סיים להחליף את המחבר וביצע עבודה מושלמת.

לאחר סיום הניווט ותיקון בעיות החומרה בהן נתקלנו, איחדנו את הניווט עם כיבוי הנר, בתקווה לראות את הרובוט מכבה נר בזירת התחרות (זאת לאחר שבדקנו את הכיבוי והניווט לחוד). אף במקום לבצע את הכיבוי כמו בבוקר, היה הרובוט נכנס לחדר, ומתחיל את העקיבה אבל במקום לבצע את תוכנית הכיבוי בשלמותה הרובוט היה נכנס למצב קבוע של סיבוב במקום נגד כיוון השעון. לאחר שהרובוט חזר על אותה בעיה פעמים מספר, הרצנו מחדש את חלק הכיבוי לבדו – אך הבעיה הופיעה בכל ריצה.

כמובן שמייד התחלנו לבדוק כל סיבה אפשרית לתקלה, מאחר והרצנו את אותה תוכנה מהבוקר (לפחות כך חשבנו) בדקנו בתחילה גורמים אפשריים לתקלה בחומרה: אינקודר מקולקל, חיבורים רופפים על הInterface, בעיות מתחים וקריאות חיישנים. אך הכול עבד בצורה תקינה ופנינו לבדוק את התוכנה, ברגע זה גילנו שחלק הכיבוי – אותו התאמנו בבוקר אותו יום לא נשמר ובמקומו מופיעה גרסא ישנה יותר של התוכנה. מקרה זה קרה לנו פעם אחת לפני התחרות ופעם נוספת אחרי התחרות – אין לנו מושג למה או כיצד העלו המחשב או העורך גרסא ישנה של התוכנה. מקרה זה ייאש אותנו מאוד, גם לא נשמרה התוכנה העדכנית ביותר, וגם הופיעה כל פעם הבעיה – שלא הייתה קיימת גם בגרסאות הישנות של התוכנה.

עד שלוש בלילה באותו יום ניסינו כל אפשרות לתקן את הבעיה, כתבנו את התוכנה מחדש פעמיים, עברנו כמעט על כל התוכנה של הרובוט, ואפילו ניסינו לשבת עם אלי קולברג, המנחה שלנו, ושחר מנדלוביץ', תלמיד עתודה שהשתתף בתחרות בשנת 2001, ולבנות יחד איתם את התוכנה. אך תמיד הופיעה שוב הבעיה. בשלוש הלכנו למלון לישון וחזרנו רעננים בשמונה בבוקר (לפני שינה קצרה זו היינו ערים במשך יותר משלושים שעות רצוף). אך למרות שהיינו במצב טוב יותר לפתור את הבעיה מבערב הקודם, החלטנו לוותר בגלל הייאוש והאכזבה מהבעיה הלא צפויה. האכזבה הייתה גבוהה מכיוון שמבין כל הרובוטים ביום האימונים "קרפלעך" היה הרובוט המהיר ביותר וגם ביצע את בונוס הרהיטים (תוצאותיו של "קרפלעך" לא עוברות 10 נקודות, הרובוט שהגיע למקום הראשון עשה זאת בתוצאה של 14). החלטנו להריץ את הרובוט – לפחות יראו שהוא מגיע לחדר מהר מכל הרובוטים – למרות שאינו מכבה

את הנר. ואכן בריצה הראשונה הגיע "קרפלעך" לחדר השלישי, התחיל לנסוע לכיוון הנר וכמובן הסתובב במקום במרחק חצי מטר מהנר, אם הרובוט היה משלים כיבוי זה הוא היה מקבל תוצאה של 3 נקודות.
לאחר צפייה בריצה זו התבהר לאחד מחברי הצוות מקור הבעיה, אך לא היה מספיק זמן בין הריצות כדי לסדר אותה, ושלושת הריצות של קרפלעך נגמרו ללא כיבוי.
הבעיה נבעה ממספר גומרים שונים שהתחברו ביחד:
כיוון חיישן הפירואלקטרי שנעשה ביום הראשון, בוצע כאשר עדשת החיישן הפירואלקטרי הייתה בזווית, ולכן ניתנו ערכים גדולים מדי, בשלב מאוחר יותר יושרה עדשת החיישן ולכן הוא נתן קריאה של נר בצורה רנדומאלית גם כאשר לא היה נר מול החיישן – מצב זה הכניס את הרובוט למצב הסיבוב האין סופי.
הטעות שגרמה לרובוט להיתקע בסיבוב אינסופי הייתה ידוע לצוות שבוע קודם לכן, הם תכננו לתקן אותה, אך מלחץ הזמן נשכח הדבר ולא סודר על ליום התחרות.
כדי למנוע טעיות כאלה בעתיד יש לדאוג לדברים הבאים:

1. לתקן כל בעיה ברגע שהיא מתגלית – לא לדחות דברים לעתיד.
2. לדאוג לייצב את מערכת החיישנים, כך שהם לא יוכלו לזוז והם יתנו תוצאות זהות בכל מצב.
3. לישון לפני התחרות – יום האימונים הוא יום עמוס מאוד, בו מבצעים כיוונים אחרונים ותיקונים במערכת הרובוט, לכן חשוב להיות מרוכזים ובשיא הכושר ליום זה, לא מטושטשים ועייפים.
4. לגמור את הרובוט בזמן ולא ברגע האחרון (קל להגיד מאוד קשה לעשות).

התחרות העולמית

לקראת התחרות העולמית שיפרנו את תוכנתו של קרפלעך כדי למנוע את כל הצרות שקרו בתחרות הארצית. בנינו בקרת מהירות בשיטת PID (בה לא השתמשנו בסוף בגלל התנגשות תוכנה), שכללנו ושיפרנו את תוכנת התיקונים בנסיעה, כתבנו מחדש את תוכנית הכיבוי נר – ייעלנו ושיפרנו אותה וגם הוספנו פונקציות בטיחות.

יומיים לפני הטיסה לארה"ב "קרפלעך" כיבה את הנר במצבים קיצוניים מאוד, באמינות של 90%. הבדיקות נערכו בתנאים קיצוניים מאוד – הרהיט חוסם לגמרי את הנר, נמצא מול הכניסה, הנר במקומות שאסורים לפי התחרות, אפילו הגענו למצב בו בדקנו את הרובוט עם רהיט וקלסר עמוד בחדר – והוא כיבה את הנר בהצלחה.

בארה"ב עבדנו כל יום במרתף בית הכנסת היהודי.

למרבה הפתעתנו הרובוט לא נפגע בטיסה לארה"ב כלל. התקלה היחידה שקרתה היא שנשברה אחת מרגלי חיישן ה-UVtron. ניסינו להלחים אותה בחזרה, אך היא נשברה שוב ושוב, בסופו של דבר נכנענו והחלפנו את נורת ה-UVtron השבורה בחדשה.

לאחר מכן התחלנו להריץ את הרובוט, ופרט למספר כיוונים של תוכנת הכיבוי והניווט, הרובוט עבד כמו בארץ וכיבה את הנר במצבים קיצוניים (בארה"ב בדקנו אותו עם שני רהיטים באותו החדר).

אך ביום השני התגלתה בעיה בכרטיס המעכב, הוא סגר את המעגל מיידית ללא עיכוב המתח. הוצאנו אותו והחלפנו אותו במתג, כך שיכולנו להמשיך לעבוד בזמן שאחד מחברי הקבוצה בודק את הכרטיס. אך לאחר הוצאת הכרטיס התגלו בעיות חשמל ברובוט, ובליית ברירה הפסקנו להריץ אותו ופירקנו את כל חלקיו כדי למצוא את החוט הסורר, במהלך הפירוק חיברנו מחדש כל חוט, עברנו על הלחמות חשודות, ובעצם הרכבנו מחדש את הרובוט. לבסוף גילינו שהבעיה לא הייתה ברובוט אלה במחבר שחיבר את הספק לכרטיס החשמל. לא התרגזנו ממקרה זה, כי עכשיו היינו יכולים להיות בטוחים שלא יקרו לנו תקלות מכניות ברובוט, מאחר ועברנו על כל האלקטרוניקה שבו.

בעת סיום הרכבת הרובוט מחדש, והלחמת מחבר חדש לחוט המתח, הצלחנו גם למצוא ולתקן את הבעיה בכרטיס המעכב. לאחר, החלפת הדיודה, הטרנזיסטור והקבל (כל החלקים פרט לממסר והנגד), ובדיקת כל רכיב במעגל עם המנחה, גילינו שאחד מפסי הנחושת על לוח המעגל נקרע, ולכן המעגל החשמלי בכרטיס נפגם ומנע את פעולתו.

לאחר תיקון שתי בעיות אלה שבנו להריץ את הרובוט, ניצלנו את הימים לפני התחרות כדי להוסיף פונקציות בטיחות, ולבצע תיקונים נוספים בתוכנת הרובוט.

כמו בתחרות הארצית גם בתחרות בטרינטי ניתן יום אימונים לפני התחרות עצמה, ביום זה הרצנו את הרובוט על זירת התחרות פעמיים רבות. לקרפלעך הייתה בעיה עם זירת התחרות, רצפת הזירה הייתה עשויה עץ גבשושי שגרם לחיכוך גבוה בהרבה מאשר בזירה עליה התאמנו. חיכוך גבוה זה גרם לכך ש"קרפלעך" היה נתקע בעת ביצוע סיבובים ותיקונים במקום. בעיה זו הקשתה עלינו יותר מעל שאר הקבוצות בגלל שיטת ההנעה בה בחרנו, הנעה דיפרנציאלית אסימטרית. בעת ביצוע סיבוב במקום בשיטה זו מפעילים המנועים כוח גדול זה על זה, ופחות כוח פועל כדי לסובב את הרובוט. בגלל החיכוך הגבוה בזירות התחרות לא היה מספיק כוח למנועים לבצע את הסיבובים במקום. בעיה זו פתרנו בקלות, פשוט ע"י הוספת כוח למנועים בעת ביצוע סיבוב ותיקון במקום. לאחר תיקון הבעיה בילינו את שאר היום בהרצות, כדי לבדוק כל מצב, ובשכלול תנועת הרובוט כאשר הוא מכבה את הנר, כדי להבטיח כיבוי של הנר (מיותר לגמרי מאחר בדרך כלל כיבה "קרפלעך" את הנר בשנייה הראשונה להפעלת המאוורר – אבל לא רצינו לאפשר שום טעות).

ביום האימונים גם גילנו שהמכשולים בהם משתמשים בתחרות היו נמוכים בהרבה מהמכשולים עליהם התאמנו, ולכן החלטנו לנסות לבצע את בונוס המכשולים, החלפת הגלגל האחורי בגלגל גמיש יותר, אפשרה עליה קלה על המכשולים, והגלגלים הקדמיים שתוכננו בדיוק לצורך זה, הפכו את ירידת קרפלעך מהמכשול לחלקה ביותר. ואכן בכל ריצות האימונים הרצנו את הרובוט עם מכשולים בזירה ללא תקלות.
מהמצב ביום האימונים ראינו שאין רובוט המהווה תחרות ל"קרפלעך", גם מבחינת תוצאות (מהירות ובונוסים) וגם מבחינת אמינות, פרט לרובוט "Silent Bob" מתיכון "היובל" בהרצליה. לקראת הצהרים – כאשר קבוצות רבות אחרות עדיין התאמנו על הזירות אנו ארזנו את "קרפלעך" וחזרנו לבתי המארחים.
ביום התחרות לא אפשרו להריץ על הזירות, דבר שהיינו צריכים מאוד, מכיוון שבבוקר התחרות צבעו השופטים מחדש את ריצפת הזירות, וכך שינו שוב את מקדם החיכוך. לא ידענו האם להחליש את עוצמת הסיבובים במקום או להשאיר אותם כמות שהם, והחלטנו שהכי טוב יהיה ללכת אם התוכנית שעבדה יום קודם.

מהלך הריצות היה כדקלמן:

ריצה ראשונה: קרפלעך הגיע לחדר הנר, והתחיל לחפש את הנר בחדר, כאשר פנה קרפלעך אל הנר היינו בטוחים שנראה כיבוי בזמן שיא, אך במקום להגיע לפס-הלבן ולכבות את הנר, פנה במפתיע קרפלעך מהנר ויצא מן החדר. עד היום אין לנו הסבר לתקרית המצערת – ככל הנראה קריאת שווא של החיישן הפירואלקטרי.

ריצה שניה: קרפלעך נכנס לחדר הראשון, בו מוקם הנר ופנה לנר. אך כאשר הגיע קרפלעך לפס-הלבן במקום להפעיל את המאוורר, פעלה תוכנת ביטחון שנועה למנוע יציאה מהחדר, וגרמה לכך שקרפלעך נגע בנר, בטרם הפעיל את המאוורר לכבות את הנר, מה שהוסיף 50 נקודות להרצה.

ריצה שלישית: הנר שוב מוקם בחדר הראשון, אך כאשר פנה קרפלעך להיכנס לחדר, קרה הדבר אשר חששנו ממנו, והתיקונים במקום, במקום ליישר את קרפלעך לעומת הקיר, גרמו לו לקבל זווית חדה ולהתקע בכניסה לחדר.
בעיות אלה לא יכולנו למנוע, אלא באמצעות פונקציות גיבוי נוספות, וריצות בבוקר התחרות על הזירות התקניות.

נספח ו' – מימוש משוואות באסמבלי

האסמבלי כולל בתוכו את הפעולות האריתמטיות הבסיסיות ביותר. הפקודות המתמטיות היחידות שיש בו הן: חיבור, חיסור, כפל וחילוק. אך לפעמים מתעורר הצורך לבצע פונקציה מורכבת יותר – כפל בשבר, משוואה טריגונומטרית, שורש, וכדומה.

כאשר אנו באים לממש משוואות אלה אנו צריכים להתייחס לשני גורמים חשובים:

- הערכים של הקלט (הערך המוכנס) והפלט (התוצאה). כל המספרים באסמבלי מבוטאים באמצעות שמונה או שש-עשרה ספרות בינאריות כך שטווח המספרים האפשרי הוא: 0 עד 255 או 128- עד 127 לשמונה ביט ו- 0 עד 65535 או 32768- עד 32767 לשש-עשרה ביט. עקב כך לא נוכל להכניס ערכים שאינם בתחום וגם התוצאה לא תוכל לעבור תחום זה. באסמבלי גם אין ייצוג מובנה לשברים.

  הפתרון לבעיה הוא הצבת ערכי האמת האפשריים לעומת הערכים אותם אנו יכולים לבטא (הערכים המייצגים). למעשה אנו צריכים למצוא מקדם אשר ימיר את ערכי האמת לערכים המייצגים. את המקדם אנו מוצאים על ידי חלוקת כמות ערכי האמת בכמות הערכים המייצגים. למעשה, אנו צריכים "לכווץ" או "למתוח" את הערכים כך שיוכלו בשלמות על ידי הערכים המייצגים. את המקדם שלפיו "נמתח" או "נכווץ" את ערכי האמת אנו מוצאים באמצעות חלוקת כמות ערכי האמת בכמות הערכים שאנו יכולים לבטא.

<u>לדוגמא:</u> אם אנו רוצים לבטא את הערכים 32768- עד 32767 ע"י אוגר של שמונה ביט (128- עד 127), אנו צריכים ל"כווץ" את ערכי האמת שיתאימו לערכים האפשריים. מאחר ויש לנו 65536 ערכים אפשריים ו-256 מקומות בלבד, המקדם יהיה:

$$K = \frac{256}{65536} = 1/256$$

את המשמעות של הפעולה שביצענו נציג בטבלה הבאה:

ערך באוגר	הערכים המיוצגים
0	0-255
1	256-511
2	512-767
...	...
255	65281-65536

<u>דוגמא נוספת:</u> אנו רוצים לבטא את כל השברים העשרוניים עד סיפרה אחת אחרי הנקודה ((-1) (-0.9) – 0.8, 0.9, 1) ע"י אוגר של שמונה ביט. המקדם יהיה: $K=\frac{255}{21}$

את המשמעות של הפעולה שביצענו נציג בטבלה הבאה:

הערך המיוצגים	ערכים באוגר
-1	(-128)-(-115)
-0.9	(-114)-(-102)
…	…
0.9	100-113
1	114-127

- זמן ביצוע: כאשר אנו מעבדים משוואה מורכבת זמן העיבוד עד הגעה לפיתרון עלול לקחת זמן רב, ויש תוכנות שלא ניתן להשהות את ריצתן לזמן ארוך. כדוגמת תיקונים בנסיעה: כאשר הרובוט מתקן תוך כדי נסיעה הוא צריך לדגום בתדירות גבוהה את מצבו. אם נערוך חישוב שייקח זמן רב בתוך הדגימות, הרובוט לא ידגום בתדירות הרצויה ולא יהיה זמן עיבוד לחלקים אחרים של התוכנה.

ישנן שיטות שונות לבצע המרה של משוואות מורכבות לתוכנה, הן נבדלות זו מזו בתכונות רבות ולכן צריכים לדעת להתאים את השיטה למשוואה, וליישום בו רוצים להכניס את המשוואה.

<u>קירובים:</u>

קירובים לנוסחאות הם ללא ספק השיטה הנוחה והמהירה ביותר להפוך נוסחא מורכבת לנוסחא פשוטה שאפשר להכניס בקלות לתוכנה. הרעיון מאחורי קירוב הוא למצוא נוסחא פשוטה שנותנת תוצאות דומות לנוסחא המקורית (ראה גם את טור טיילור בנספח זה). ישנן נוסחאות קירוב שמתאימות רק לטווח ערכים מסוים בנוסחא, אפשר להשתמש בהן כאשר מגיעים הערכים המתאימים. כך אפשר לחסוך זמן ביצוע גם אם לא מחליפים לגמרי את הנוסחא המקורים בנוסחת הקירוב.

דוגמא לנוסחת קירוב היא:

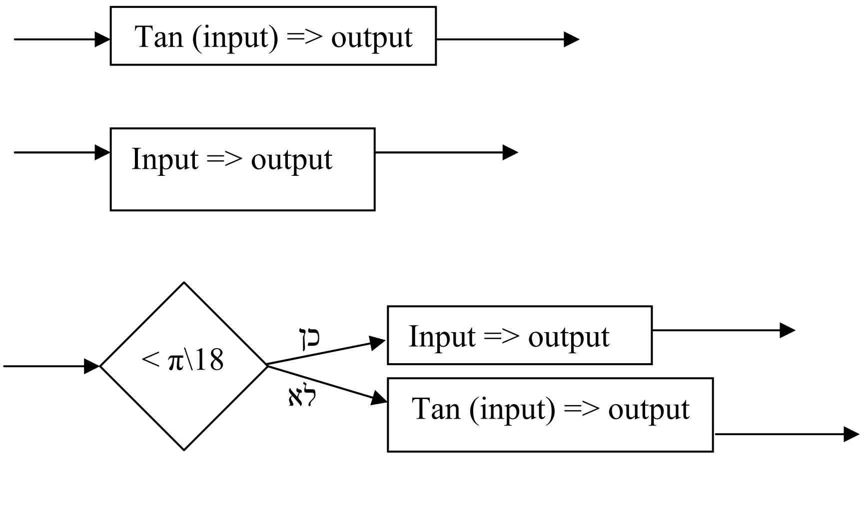
$$\tan X = X$$

נבדוק את אמינות הנוסחא בטבלה:

זווית ברדיאנים	$\tan X$	X	אחוז השגיאה
0	0.0000	0.0000	0.0%
π/36	0.0875	0.0873	0.2%
π/18	0.1763	0.1745	1.0%
π/9	0.3640	0.3690	4.1%
π/6	0.5773	0.5236	9.3%
π/4	1.0000	0.7854	21.5%
π/3	1.7320	1.0471	39.5%

אפשר לראות מהטבלה שלקירוב יש אמינות גבוהה עד π\18 (10°), לערכים גבוהים יותר השגיאה נהפכת למשמעותית ועבור ערכים מעל π\6 (30°) נוסחת הקירוב חסרת ערך.

אין ספק שנוסחת קירוב כמו $\tan X = X$ יכולה להקל מאוד על עבודת המתכנת: בעוד שביצוע פונקצית Tan בשיטה אחרת ידרוש מחשבה, זמן עבודה וזמן ביצוע, העברה פשוטה של הפלט לקלט ללא ביצוע כל פעולה היא הדבר הפשוט והמהיר ביותר.
ברור שנוכל להשתמש בשיטה רק עד π\18, כי בערכים גבוהים יותר אי-הדיוק נהפך למשמעותי. אבל גם אם אנו יוצרים תוכנה בה הקלט האפשרי גבוה מ-π\18 עדיין נוכל להשתמש בנוסחת הקירוב כל עוד הערכים נמוכים וכך לחסוך זמן עיבוד על אחוזים מסוימים של הקלט.



המקרה הראשון נותן דיוק גבוה אך לוקח זמן מעבד רב.
המקרה השני נותן דיוק בזויות קטנות בלבד אך אינו דורש זמן מעבד רב.
המקרה השלישי משלב את שני המקרים הקודמים.
<u>טור טיילור:</u>

טור טיילור היא שיטה מתמטית להפיכת משוואת שונות לפולינומים. מאחר ואין לנו פקודת לביצוע פעולות מתמטיות מיוחדות, אבל אנו יכולים לבצע פעולות של הכפלה וחילוק, טור טיילור יספק לנו משוואה שנוכל לבצע בתוכנית.

$$f(x)=f(a)+f\ '(a)(x-a)+\frac{\left[f\,''(a)(x-a)\right]^2}{2!}+\frac{\left[f\,'''(a)(x-a)\right]^3}{3!}+...+\frac{\left[f^{n}\,'(a)(x-a)\right]^n}{n!}$$

דוגמא: ניקח את המשוואה $\cos(x)$, a=0:
נכניס אותה לטור טיילור:

$$\begin{aligned} f(x) &= \cos(x) \\ f\,'(x) &= -\sin(x) \\ f\,''(x) &= -\cos(x) \\ f\,'''(x) &= \sin(x) \\ f\,''''(x) &= \cos(x) \end{aligned}$$

$$\cos(x)=1-0-\frac{x^2}{2}+0+\frac{x^4}{24}-0-\frac{x^6}{6!}+0+\frac{x^8}{8!}...$$

$$\cos(x)=1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{6!}...$$

כמובן שלא נכניס את כל הטור לתוכנה. אפשר לראות שכאשר ה-x נמוך בחלק הראשון של המשוואה הוא המשמעותי, כאשר ה-x גבוה החלקים שקודם היו משניים משפיעים מאוד על התוצאה.
לכן לפי הצורך (הקלט למשוואה) נשתמש בחלקים שונים של המשוואה – בעצם נשתמש בטור טיילור כדי למצוא משוואת קירוב.
לדוגמא: כאשר ניקח רק את שני האברים הראשונים בטור:

$$\cos X = 1-\frac{X^2}{2}$$

נבדוק את אמינות הנוסחא בטבלה:

מעלות (רדיאנים)	$\cos X$	$1-\frac{X^2}{2}$	טעות באחוזים
0	1	1.000	0.0%
π/90	0.9962	0.9962	0.0%
π/18	0.9848	0.9848	0.0%
π/9	0.9397	0.9391	0.1%
π/6	0.8660	0.8629	0.4%
π/4	0.7071	0.6916	2.2%
π/3	0.5000	0.4516	9.6%

אפשר לראות שעד π/6 (30°) אין כמעט משמעות לכך שויתרנו על יתר אברי הטור. בצורה זו הגענו לנוסחת קירוב יעילה ומדויקת שפשוטה ליישום באסמבלי.
אם נרצה להשתמש במשוואה שתהיה מדויקת גם לקלט גבוה יותר נצטרך לכלול בנוסחה יותר איברים מהטור.

<u>טבלה:</u>

כאשר אנו מבצעים פונקציה, אנו מתאימים Y לכל X ובעצם מתאימים קלט אפשרי לכל פלט שמתאים לו. מסיבה זו אנו יכולים לבטא פונקציות בגרפים וטבלאות.
הרעיון מאחורי טבלה הוא פשוט: מול כל קלט מופיע הפלט שלו, כך שבמקום לבצע פונקציות מספיק לחפש בטבלה את ערך הקלט ולהוציא את ערך הפלט המתאים לו.
מעבד ה-HC12 תומך בשיטות מעיון המאפשרות שימוש בטבלה בצורה שונה מחיפוש ערך בטבלה, שיטה זו פשוטה לשימוש ויעילה יותר מבחינת זמן מעבד. הרעיון מאחורי השיטה הוא שבמקום לעשות טבלה של ערכי X לעומת ערכי Y ואז חיפוש ערך מסוים בטור של X והוצאת ערך Y המקביל לו, אנו עושים טבלה של ערכי Y בלבד כאשר X (הקלט) אומר כמה שורות צריך לרדת מראש הטבלה כדי להגיע לערך המתאים.

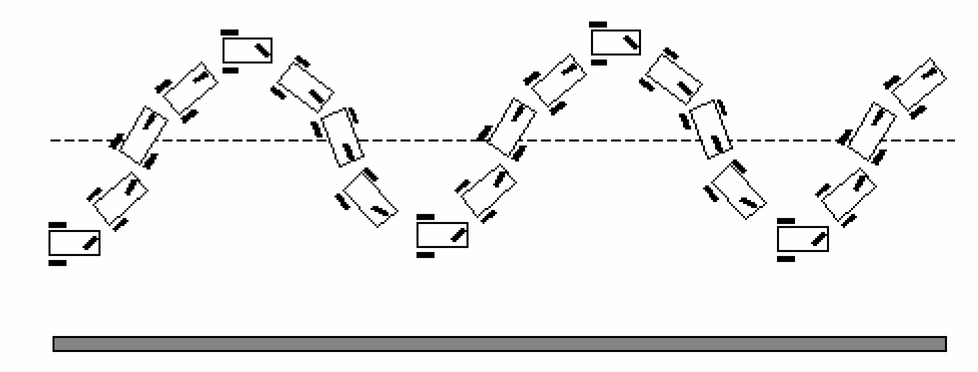
בפרק יא' - תיקונים מופיעה דוגמא לשימוש בטבלאות.

נספח ח' - תיקונים יחסיים (PID)

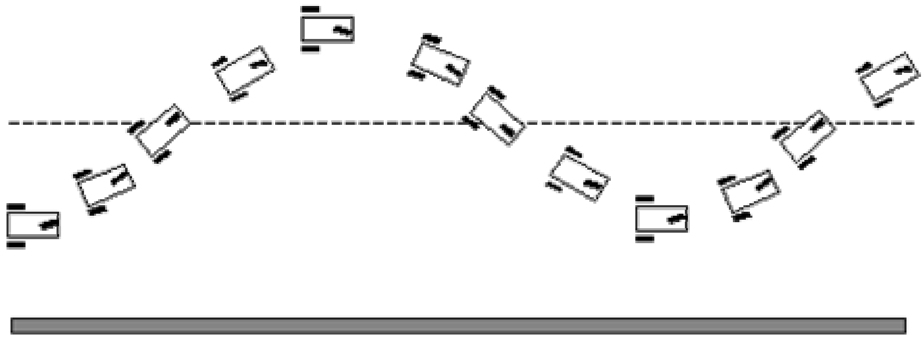
PID (proportional, integral, divertive – יחס, אינטגרל, שיפוע) - היא שיטת הבקרה הטובה והנפוצה ביותר, ומשמשת לבקרת מהירות ומיקום של גופים נעים. השיטה נובעת מנוסחאות מתמטיות מורכבות, אבל הרעיון מאחוריה נובע מעקרונות פשוטים של בקרה.
את הרעיון מאחורי שיטת בקרה זו נסביר באמצעות דוגמא של מכונית שמטרתה לנסוע על קו ישר.

<u>יחס:</u>

החלק הראשון בשיטת הבקרה הוא ה-P – התיקון היחסי. מטרת התיקון היחסי היא להפוך את השגיאה לאפס.
הרעיון מאחורי חלק זה הוא הפשוט ביותר – ככל שהטעות גדולה יותר כך גדול יותר גם התיקון. מתקיים יחס ישר בין התיקון לטעות – היחס הזה הוא מקדם הטעות (K_p).
לפי הדוגמא, המכונית נמצאת בתחילה במרחק 12 מטר מימין לקו. במצב זה נסובב את ההגה 20 מעלות שמאלה, מה שיגרום לשינוי כיוון המכונית ותחילת תנועתה לעבר הקו, ככל שהמכונית תתקרב לקו כך יקטן התיקון היחסי: כאשר המכונית תהיה 6 מטר מימין לקו נסובב את ההגה 10 מעלות שמאלה בשלוש מטר נסובב 5 מעלות וכאשר נהיה על הקו נחזיק את ההגה ישר.
התוצאה של תיקון זה תהיה שהמכונית תעבור את הקו ותתחיל לבצע בדיוק את אותו התהליך לצד השני:



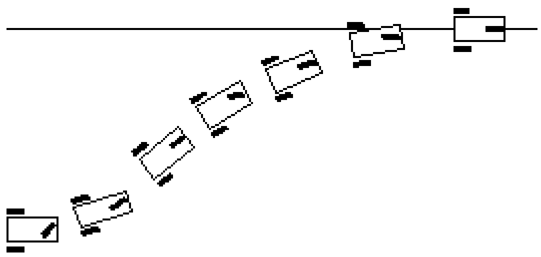
שינוי המקדם של התיקון היחסי לא יביא לביטול תיקון היתר הקבוע אלא רק ישנה את המהירות בה תחזור המכונית לקו:



הדרך היחידה לתקן את תיקון היתר היא על ידי הוספת ערך השיפוע לתיקון היחס.

<u>שיפוע:</u>

השיפוע מתאר את קצב שינוי הסטייה. מטרת תיקון השיפוע היא להוריד את קצב שינוי הסטייה לאפס. גם תיקון זה הוא יחסי לקצב השינוי בטעות ולא לטעות עצמה.
נשתמש בדוגמא של מכונית הנוסעת על קו: השיפוע יתאר את שינוי המרחק של המכונית מהקו ובעצם את המהירות בה חוזרת המכונית אל הקו. אם מהירות המכונית קבוע – יתאר קצב השינוי את הזווית בין כיוון תנועת המכונית לקו.
תיקון השיפוע יביא את המכונית לנסוע במקביל לקו – אך ללא התייחסות למרחק בין המכונית לקו.



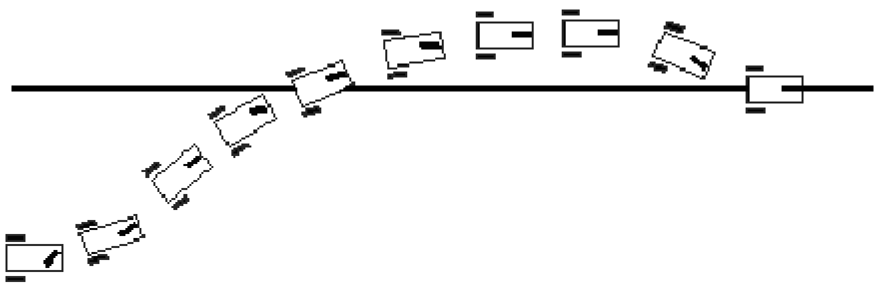
<u>שילוב תיקוני יחס ושיפוע:</u>

שילוב של תיקוני יחס ושיפוע יגרום לתוצאה הבאה:

מצב ההתחלה של המכונית הוא 12 מטר מימין לקו. התיקון היחסי יגרום לסיבוב ההגה 20 מעלות ימינה, המכונית תתחיל לסטות לכיוון הקו. כאשר תתקרב המכונית לקו, וזוויתה תהיה גבוהה, יאזנו התיקון היחסי והתיקון הזוויתי זה את זה (מצב שלוש). לאחר מכן ימשכו התיקונים עד לנסיעת המכונית על הקו.
יש לזכור שבביצוע תיקונים יחסיים יש חשיבות רבה לגודל הקבועים וליחס ביניהם: קבועים קטנים מדי עלולים להפוך את התיקון למזערי וכך יעבור זמן רב עד להשלמת התיקון, קבועים גדולים מדי יגרמו לכך שתמיד יבוצע תיקון חזק מדי ותנועת המכונית תהיה קיצונית. יחס גבוה בין מקדם התיקון היחסי למקדם תיקון השיפוע יגרמו לכך שימשיכו להיווצר תיקוני יתר, בעוד שיחס קטן יגרום לכך שיידרש זמן רב עד שתגיע המכונית לקו.

<u>סכום:</u>

תיקוני הסכום נועדו לסייע בתיקון שגיאה קבועה המפריעה לתיקון הרגיל. תיקוני הסכום מפצים על סטייה במערכת התיקונים הרגילה, מטרתם לאפס את סכום הטעיות.
במקרה הדוגמא יבוא לביטוי תיקון הסכום אם מערכת ההיגוי של המכונית אינה מכוונת: קיימת סטייה בהגה – או שחסר אויר באחד הגלגלים. ללא תיקוני סכום תגרום הסטייה במערכת ההיגוי לכך שהמכונית תיסע במקביל לקו, אף במרחק מסוים ממנו – זאת כתוצאה מכך שבמרחק זה תיקוני היחס יהיו שווים לסטייה במערכת ההיגוי. תיקוני הסכום יוציאו את המכונית ממצב זה מכיוון שכאשר תנוע המכונית לאורך זמן במרחק מהקו, יצטבר סכום הטעויות – ותיקון הסכום יניע את המכונית חזרה לקו.



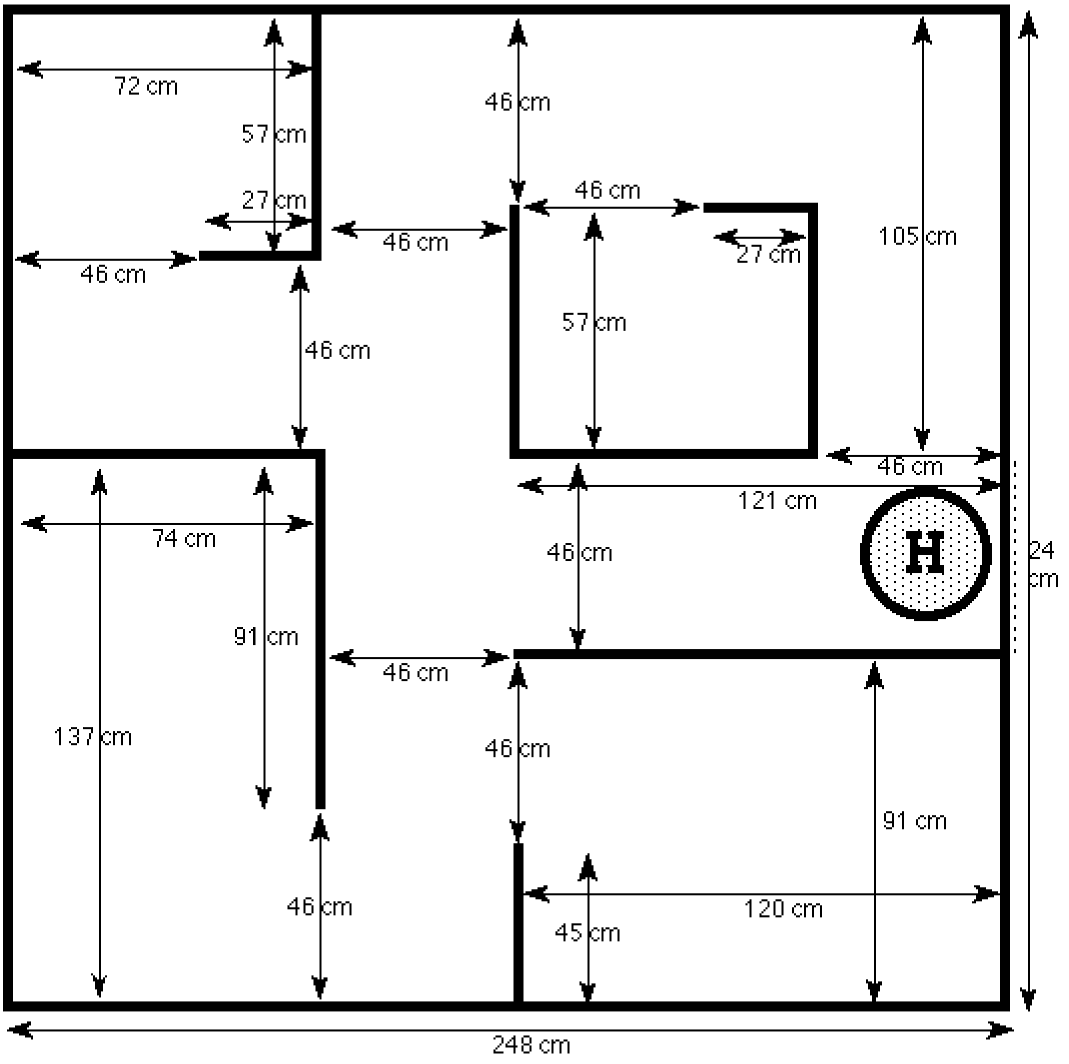
נספח ט' – חוקי תחרות הרובונר העולמית – 2002

<u>מטרת התחרות.</u>

המטרה המסוימת של התחרות הזו היא לבנות רובוט הנשלט באמצעות מחשב שיכול לנוע בתוך מבוך הבנוי כדירה, למצוא נר דולק ואז לכבותו בזמן הקצר ביותר, בהתאם למספר גורמי פעולה (מופיע בהליך ניקוד בהמשך). מהלך זה מכוון להדמיית פעולה בעולם האמיתי של רובוט המבצע פונקציה של הגנת בית אמיתי משריפה. הנר מייצג שריפה שהתחילה בבית ושהרובוט חייב לגלותה ואז לכבותה. למרות האמור לעיל, המטרה האמיתית של התחרות היא לקדם את הידע והטכנולוגיה של רובוטיקה במתחרים עצמם ובעולם בכלל.

<u>מבנה הזירה ותכונותיה.</u>

מבנה הזירה:



בדומה לעולם האמיתי שבו יש תמיד מידה מסוימת של אי ודאות בכל מידע, מידות רצפת הזירה הן מקורבות בלבד. המידות האמיתיות יכולות להשתנות עד לשני סנטימטרים מהערכים הנתונים. ברוכים הבאים להפעלת רובוט בעולם האמיתי.
הקירות של המבנה יעשו מעץ וגובהם יהיה 33 ס"מ. הקירות יצבעו בצבע לבן. רצפת הזירה תהיה ישרה, מעץ וצבועה בצבע לטקס שחור. כל תפר או חיבור ייסתם ויצבע באותו צבע שחור. חיבורי חלקי הרצפה יתכן ולא יהיו שטוחים לחלוטין. יש לוודא שהרובוט שלך יכול להסתדר עם אי רציפויות של עד 3 מ"מ אם צריך.

כל המסדרונות והכניסות לחדרים יהיו ברוחב 46 ס"מ. אין דלת בכניסות לחדרים, רק פתח ברוחב 46 ס"מ. יהיה פס לבן ברוחב 2.5 ס"מ לרוחב כניסה של כל חדר בכדי לסמן את הכניסה לכל חדר.

רצפת הזירה תהיה שחורה אולם חלק מהרובוטים עשויים להשתמש בקצף או חומרים אחרים בכדי לכבות את להבת הנר. יעשו מאמצים לנקות את הזירה אחרי ריצת כל רובוט, אולם אין בטחון שהרצפה תישאר שחורה באופן אחיד לכל אורך התחרות. הרצפה גם עשויה להכיל גם נקודות אדומות או כחולות קטנות (בקוטר 3 מ"מ) בכדי לסמן מיקומים אפשריים לנרות או רהיטים. אם הרובוט לא פועל במצב NDR (ראה סעיף 16), הרצפה תהיה ישרה ללא שיפועים או מדרגות.

הרובוט יכול להתחיל במיקום עיגול הבית. עיגול הבית האמיתי יהיה בצבע לבן ללא האות H המסומנת בשרטוט הזירה. קוטר העיגול 30 ס"מ, והוא יוצב במרכז המסדרון שרחבו 46 ס"מ, כך שיישארו שמונה ס"מ של מרווח בין קצה העיגול לקירות. לכן מרכז העיגול יהיה במרחק של 23 ס"מ מכל צד של הקירות הצדדיים. יש לשים לב לכך שבשרטוט הזירה יש פתח בקיר החיצוני של הזירה מאחורי עיגול הבית. חלק זה של הקיר ניתן להסיר בכדי לאפשר למתחרים גישה קלה יותר לרובוטים שלהם בזמן הצבתם בזירה. אולם ניתן להציב את הקיר במקומו אם זה עוזר לפעולת הרובוט. ניתן להשתמש במתקני הצבה של הרובוטים אם זה עוזר להם בתחילה בזמן ההצבה, ליישר את עצמם במעגל הבית. הרובוט חייב להתחיל בתוך מעגל הבית, אולם מרגע שהתחיל, הוא יכול לנוע לכל כיוון רצוי ויכול לנוע אופקית או אנכית בשרטוט הזירה המופיע בנספח א'.

חלק מהפינות של הזירה יתכן ויהיו בהן חלקי פלסטיק שחורים המשמשים לחיבור חלקי הזירה, ובולטים בערך 2 ס"מ לתוך הזירה.

<u>אור סביבה</u>

כל התאורה באולם תעשה ע"י אותן נורות אדי נתרן בלחץ גבוה, ששימשו בהצלחה את התחרויות הקודמות (למעט תחרות 2001). כמו בעבר, התאורה ישירות מעל אזור הזירות תעומעם קלות.

למתחרים יינתן זמן ביום התחרות לבצע קריאות של רמת אור הסביבה, אם צריך, בכדי לכייל את הרובוטים שלהם. ברגע שתופעל מערכת התאורה למצב קבוע ביום שבת בבוקר בשטח התחרות, לא יהיה ניתן לשנות את התאורה כך שתתאים למתחרה או מתחרים באופן אישי. חלק מהאתגר של התחרות הוא ליצור רובוט שיכול לפעול במצבים של העולם האמיתי, וזה כולל תנאי תאורה שאינם עקביים, צללים, סנוור, וכו'.

פעולת הרובוט

מהרגע שהרובוט הותנע, הוא חייב להיות בעל מנגנון בקרה עצמית ללא שום התערבות אנושית, כלומר, ההתקנים חייבים להיות התקנים מבוקרים מחשב ולא מבוקרים ידנית.

הרובוט יכול להיתקע בקירות או לנגוע בקירות כאשר הוא נוסע בזירה, אבל אסור לו לסמן או לפגוע בקירות בעשותו כך. בכל מקרה יוטלו עונשים (ניקוד פחות טוב) עבור נגיעה בקיר (ראה סעיף 18 על עונשים). הרובוט לא יכול להשאיר דבר כל שהוא מאחור כאשר הוא מנווט בזירה. הוא אינו יכול לעשות סימנים כלשהם על רצפת הזירה שיעזרו לו לנווט כשהוא נוסע בזירה. רובוט שבאופן מכוון, לדעת השופטים, יפגע בזירת התחרות (כולל הקירות), יפסל. זה אינו כולל סימונים מקריים או שריטות שנעשו תוך כדי תנועת הרובוט.
הרובוט חייב, לפי דעת השופטים הרשמיים, למצוא את הנר לפני שהוא מנסה לכבות אותו. הרובוט אינו יכול רק להציף את מבנה הזירה בדו תחמוצת הפחמן (CO_2) ולכבות את הנר במקרה.

כיבוי הנר

לרובוט אסור להשתמש בשיטות הרסניות או מסוכנות בכדי לכבות את הנר. הוא יכול להשתמש חומרים כמו מים, אויר, CO_2, וכו'., אולם אסור השימוש בשיטה או בחומר שהם מסוכנים או שיסבו נזק לזירה. לדוגמא, הרובוט אינו יכול לפוצץ גליל נפץ ולכבות את הנר כתוצאה מההדף. הרובוט אינו יכול להכות בנר או להפוך אותו בכדי לכבותו.

מותר לכבות את הנר באמצעות משב אויר שנשלח לעברו. למרות שזו אינה שיטה מעשית כל כך לכיבוי שריפה בעולם האמיתי, היא מותרת בתחרות זו היות ומטרת התחרות היא קידום הרובוטיקה ולא בהכרח קידום טכניקות לכיבוי שריפות.

אסור לפגוע או להפיל הנר כל עוד הוא דולק. אם הרובוט פוגע או מפיל את הנר במקרה, אחרי שהוא כובה, ריצת הרובוט עדיין תחשב, אך זה יגרור ענישה (ראה סעיף 18 לגבי ענישה). הנר יורכב על בסיס עץ כך שלא יוכל ליפול בקלות מזרם מים או אוויר.

כל לכלוך שעושה רובוט (מים, קצף, וכו') במאמציו לכבות את הנר, ינוקה על ידי השופטים כמיטב יכולתם, בין הריצות.

חידושים אחרונים בתחום כיבוי נר אפשרו לרובוטים בתחרויות קודמות לכבות את הנר ממרחק של 1 מטר ממנו. יכולת זו לכבות את הנר ממרחק גדול יחסית, נמצאת בניגוד למטרת התחרות של בנית רובוטים אינטליגנטים שיכולים למצוא ולכוון ללהבה. בכדי להחזיר את התחרות חזרה למסלולה המקורי, הרובוט חייב להגיע למרחק של 30 ס"מ או פחות מהנר, לפני שהוא מנסה לכבות את הלהבה. יהיה מעגל (או קשת, אם הקיר נמצא בדרך) בצבע לבן, ברדיוס של 30 ס"מ, על הרצפה מסביב לנר. חלק כלשהו של הרובוט חייב להיות מעל מעגל (או קשת) זה, לפני שהוא מכבה את הנר. הרובוט עדיין יכול לירות סילון של CO_2, אולם חלק כלשהו של הרובוט חייב להיות בתוך 30 ס"מ מהנר לפני שהוא מבצע זאת. מעגל הנר נעשה מ לוח דק בעובי 0.5 מ"מ, והנר יוצב במרכז המעגל.

גודל הרובוט

הגודל המקסימלי של הרובוט יהיה 31 ס"מ X 31 ס"מ X 31 ס"מ. הרובוט לא יוכל להסתכל מעבר לקירות של המבנה ואסור שיאריך את עצמו אף פעם מעבר ל- 31 ס"מ במידה כלשהי שלו. כל הרובוטים ימדדו בקפידה. אל תיתן לרובוט שלך להיפסל בגלל שהוא מעט חורג ממגבלות אילו.

אם לרובוט חיישנים שחשים חפצים או קירות, חייבים לקחת אותם בחשבון כחלק של ממדי הרובוט ואינם יכולים לעבור 31 ס"מ מצד לצד.

אם המתחרים רוצים להוסיף דגל, כובע או פריטים שהם דקורטיביים טהורים ללא תפקוד כלשהו, לרובוט, הם יכולים לעשות זאת כל עוד לפריט אין באופן מוחלט שום השפעה על פעולת הרובוט.

משקל הרובוט

אין מגבלות על משקל הרובוט.

חומרי בניית הרובוט

אין מגבלות על סוגי החומרים המשמשים לבניית הרובוט.

הנר

הנר הדלוק אמור לייצג שריפה קטנה בבית שהרובוט מנסה למצוא ולכבות. תחתית להבת הנר תהיה בין 15 ל- 20 ס"מ מהרצפה. גובה זה כולל את גובה בסיס העץ. תחתית להבת הנר תתחיל בגובה 20 ס"מ מעל לרצפה, ובמהלך התחרות כשחלב הנר נשרף, ותחתית להבת הנר תגיע לגובה של 15 ס"מ מעל הרצפה, הנר יוחלף בנר חדש. הלהבה תראה מהצד ולא תוסתר על ידי חלב שלא נמס כמו שקורה בנרות עבים. הגובה והגודל המדויק של הלהבה אינם ידועים ומשתנים ויקבעו על ידי התנאים המסוימים של הנר וסביבתו. כל עוד הנר נמצא במסגרת המפרט שצוין למעלה, הרובוט נדרש למצוא את הנר בלי קשר לגודל הלהבה באותו רגע.

הנר יוצב באופן אקראי באחד מחדרי הזירה. לנר יש סיכוי שווה להיות בכל אחד מארבעת החדרים בכל אחת משלוש הריצות שיש לרובוט. תקוותינו שהנר יוצב בחדרים שונים בכל ניסיון ריצה של הרובוט, ובכך לבחון בצורה הטובה ביותר את פעולת הרובוט, אולם קיימת אפשרות שהנר יוצב באותו חדר פעמיים. אם קורה שהנר מוצב באותו חדר בריצות הראשונה והשנייה, אנו נדאג שהוא לא יהיה באותו חדר בריצה השלישית והאחרונה. וכך עבור כל רובוט הנר יהיה בלפחות שני חדרים ואולי שלושה במהלך שלושת ניסיונותיו.

הנר לא יוצב במסדרון, אבל יתכן ויוצב בחדר קרוב לכניסה. בכל מקרה, קדמת הרובוט תוכל לנוע לפחות 33 ס"מ לתוך החדר לפני שתתקל בנר. מעגל הנר לא יגע בקו הכניסה לחדר. יתכן ומיקומי הנר ששימשו בתחרות השנה האחרונה לא יהיו זהים לאילו שישמשו בתחרות השנה.

המתחרים לא יכולים למדוד או לנגוע בנר לפני שמשתמשים בו. השופטים יכולים להשתמש בכל נר חוקי עבור כל ניסיון ריצה ויכולים גם להחליף נרות בין ריצות.

הנר יוצב על בסיס עץ עגול (קוטרו 7 ס"מ וגובהו 3 ס"מ) הצבוע בצבע צהוב. בסיס זה משמש לשמירת הנר מנפילה בקלות. יהיה אפשר להפיל את הנר באמצעות התנגשות בו (מה שלא תרצה לעשות – ראה עונשים סעיף 18), אולם הנר לא ייפול אם פוגע בו זרם אויר או מים בלבד.

<u>חיישנים</u>

אין מגבלה על סוג החיישנים שניתן להשתמש בהם כל עוד הם אינם מפרים תקנות או כללים אחרים.

המתחרים אינם מורשים להציב סימונים, קרניים, מקורות אור או מחזירי אור על הקירות או על הרצפה בכדי לעזור בניווט הרובוט.

בוני הרובוטים צריכים להיות מודעים לכך שמצלמות סטילס ומצלמות וידאו מודרניות משדרים אור אינפרא אדום כחלק ממערכות המיקוד האוטומטיות שלהן. אור הסביבה באולם התחרות יכול גם הוא להיות מקור לאור IR, נראה, ו- UV. אם הרובוט משתמש בחיישני אור בכדי למצוא את הנר או לגלות קירות, או רהיטים, על בוני הרובוט לנקוט צעדים בכדי למנוע ממקורות אור אלו מלהפריע לפעולת הרובוט.

<u>חשמל</u>

הדרישות המקסימליות עבור כל מערכת הזקוקה לחיבור חשמל, יהיו 20 אמפר במתח חילופין של 120VAC. אם הרובוט או המחשב שלך זקוקים לחשמל – הבא כבלים מאריכים ארוכים מאורקים עם שקעים לחיבור מתח.

<u>כבלים</u>

אם הרובוט מחובר למערכת מחשב חיצוני עבור הוראות ו/או עבור מתח, יש לדאוג שהכבל ארוך מספיק כך שהרובוט יגיע לכל השטחים בזירה. הכבל יכול להיגרר מאחורי הרובוט כשהוא נוסע בתוך המבנה או שיוחזק מעל הקירות על ידי אחד מנושאי התפקידים הרשמיים שאינו נוגע בדבר. אם המתחרים רוצים להחזיק את הכבל הם יכולים לעשות זאת, אבל אם במהלך ניסיון הריצה, לדעת השופטים, הם השתמשו בכבל בכדי לעזור לרובוט, ניסיון הריצה יסתיים ללא ניקוד.

<u>סדר הריצה</u>

הרובוטים יסומנו עם מספרים בכדי לקבוע את הסדר שבו הם יתחרו בתחרות. כל רובוט יבצע ריצה בזירה לפי הסדר כפי שהוא סומן. אחרי שכל הרובוטים סיימו את ריצתם הראשונה, אזי כל התהליך יתחיל שוב עבור הריצה השנייה. במילים אחרות, הרובוטים יתחרו זה אחר זה, וכאשר כולם סיימו עם ניסיונם הראשון, כל התהליך יחזור על עצמו עבור הניסיונות השני ולבסוף השלישי.

למתחרים יהיה זמן בין ריצותיהם לבצע התאמות, שיפורים או תיקונים לרובוטים שלהם, אולם מהרגע שהרובוט שלפניהם סיים את ריצתו, תהיה להם דקה אחת להביא את הרובוט שלהם לזירה ולהתחיל את ריצתו. יהיה שעון מיוחד בכל זירה שהשופטים יפעילו כאשר הם קוראים למתחרים הבאים להיות מוכנים. הרובוט חייב להתחיל את ריצתו לפני שהשעון מגיע לדקה אחת. רובוט שלא יהיה מוכן לריצה אחרי דקה אחת יפסיד את הזדמנותו ויפסל בניסיון זה. הוא יוכל עדיין להתחרות בריצות האחרות שנשארו. ברגע שסדר הריצות נקבע, הוא לא יהיה ניתן לשינוי. אם אינך מוכן, הפסדת את תורך. הזמן בין הסבבים אינו מוגדר ומושפע ממשך הזמן שלקח למתחרים האחרים להשלים את ריצותיהם.

למרות זאת, אם מתחרים רוצים להריץ את הרובוט שלהם ברציפות מבלי לחכות בין הריצות, הם יכולים לעשות זאת. אנו מעודדים זאת למעשה היות וזה מאפשר לקצר את זמן התחרות.

ברגע שהרובוט מוכן, יקבע המיקום של הנר ושל רהיט כלשהו, אם צריך, והנר והרהיט יושמו במקומות המתאימים.

המתחרים יראו לשופט איך להפעיל את הרובוט ואז השופט ילחץ על הכפתור הנחוץ להתחלת הפעלת הרובוט.

<u>מגבלות זמן</u>

בכדי להשיג את מטרת התחרות של בניית רובוט המסוגל למצוא ולכבות שריפה בבית, חשוב מאוד למצוא את השריפה בתוך פרק זמן סביר. מגבלת הזמן המקסימלית למציאת הנר על ידי הרובוט היא 5 דקות. אחרי חמש דקות ניסיון הריצה יופסק. הזמן המקסימלי שבמסגרתו על הרובוט לחזור למעגל הבית באופן הנסיעה חזרה יהיה 2 דקות. אם הרובוט תקוע בלולאה ומבצע את אותן תנועות חמש פעמים ברציפות, בנתיב מסוים, ניסיון ריצה זה יופסק. בכל זמן שהוא שהרובוט לא זז במשך 30 שניות, ניסיון הריצה יופסק. כל ניסיון ריצה שיופסק יחשב כחסר ניקוד עבור הרובוט הזה. עצירת ניסיון ריצה מאחת הסיבות שפורטו כאן לא ישפיע על אף אחד משני ניסיונות הריצה האחרים של הרובוט.

<u>ניקוד</u>

הרובוט עם הניקוד הסופי (FS) הנמוך ביותר הוא המנצח. הניקוד הסופי מחושב מתוך מספר גורמים שונים, המוסברים בהמשך. תהליך הניקוד אינו באמת מסובך כפי שזה נראה בתחילה. הוא מיועד לעשות את התחרות מציאותית והוגנת ככל האפשר. אנו מצטערים אם זה מזכיר לך את חוקי מס הכנסה.

<u>אופני פעולה</u>

עבור כל ניסיון ריצה, ככל שניקוד הפעולה (OS) נמוך יותר, כך טוב יותר. השיטה הפשוטה ביותר לריצה של הרובוט היא באופן פעולה סטנדרטי. קיימים 6 גורמי אופני פעולה (OM) שונים, שהמתחרים יכולים לבחור עבור הרובוט שלהם, או בנפרד או צירוף שלהם בכדי להפחית את ניקוד

הפעולה עבור ניסיון ריצה זה. אופני פעולה אילו הם: ללא כבלים של מתח או נתונים, הפעלה על ידי צפצוף, נסיעה חזרה, רהיטים, ללא הסקה מחושבת (חישובי מיקום), ונקודת התחלה שרירותית.

פעולה סטנדרטית

באופן זה הרובוט מחובר להתקן חיצוני המספק לו חשמל או פקודות. הגורם המחליט בקביעת האופן הוא האם יש כבל המחובר לרובוט. גורם האופן לריצה באופן סטנדרטי הוא 1.0 (MF=1.0).

ללא כבל

באופן זה הרובוט נשלט או על ידי מחשב PC חיצוני באמצעות קשר RF או שהוא משתמש במחשב פנימי עצמאי, ובנוסף לרובוט מקור הספק פנימי עצמאי. בכל מקרה לא יהיה שום כבל מחובר לרובוט. גורם האופן עבור ריצה באופן ללא כבל הוא 0.9 (MF=0.9).

הפעלה על ידי צפצוף

במקום להפעיל את הרובוט באופן ידני על ידי לחיצה על מתג ההתחלה או על מקש ה- Enter במקלדת, הרובוט מפעיל את עצמו כאשר הוא מגלה אות צפצוף כמו זה שבאזעקה שמפעיל גלאי עשן בין 3.0 KHz ו- 4.0 KHz. זו היא התדירות שבה משתמשים בדרך כלל בגלאי עשן והיא נוצרת על ידי התקן פירו אלקטרי שניתן לרכוש במקומות רבים. מהרגע שהרובוט הופעל הוא אינו יכול להתחיל לנוע עד שמופעל אות הצפצוף. אם הרובוט מתחיל לנוע לפני שהופעל אות הצפצוף, למשל בגלל שהוא גילה בטעות את רעש הסביבה באולם, הניסיון הריצה יחשב עדיין, אבל הרובוט לא יקבל קרדיט עבור הפעלה על ידי צפצוף. אם הרובוט לא מתחיל לנוע בתגובה לאות הצפצוף, אזי ניסיון הריצה יכול להתחיל מהתחלה (שעוני העצר יאופסו) ללא קנס והרובוט יופעל ידנית. מכל מקום אם הרובוט לא מתחיל לנוע בתגובה לאות הצפצוף לא תהיה לו הזדמנות שניה (כלומר לחיצה נוספת על מתג הצפצוף) בכדי לרוץ באופן הפעלה על ידי צפצוף עבור ניסיון ריצה זה. התקן אות הצפצוף יכול להיות מוחזק בכל מרחק מהרובוט שהמתחרים רוצים והצפצוף יכול להמשך עד 10 שניות. הזמן עבור ניסיון הריצה יתחיל כאשר נוצר אות הצפצוף ולא כאשר הרובוט מתחיל לנוע בפועל בתגובה לאות. יהיה התקן אות צפצוף רשמי בתחרות, אולם המתחרים יכולים להביא ולהשתמש בהתקני אות צפצוף שלהם עצמם שפועלים בתוך טווח התדירויות המתאים, אם הם רוצים בכך. יהיה 5% הפחתה בניקוד עבור רובוט הפועל באופן זה. גורם אופן הפעולה עבור ריצה עם הפעלה על ידי צפצוף הוא 0.95 (OM=0.95).

נסיעה חזרה

אחרי כיבוי להבת הנר, הרובוט חוזר לעיגול הבית. הוא לא חייב לחזור על מסלולו בחזרו לעיגול הבית או אפילו לקחת את המסלול היעיל ביותר, הוא רק חייב לחזור חזרה, אבל מבלי שיכנס לחדרים האחרים שנמצאים בדרך. במילים אחרות, הוא חייב לכבות את הנר, הוא חייב לצאת מהחדר הזה ולחזור למעגל הבית מבלי להיכנס לחדר כלשהו מחדרים האחרים.

הרובוט יחשב כחוזר למעגל הבית אם חלק כלשהו של הרובוט נמצא בתוך מעגל הבית בקוטר 30 ס"מ. הרובוט לא צריך להיות באותו מצב שבו הוא היה כאשר התחיל את התחרות, הוא רק חייב שחלק מסוים מגופו יהיה בתוך מעגל הבית.

אם הרובוט נכנס לריצה באופן נסיעה חזרה ומוצא ומכבה את הנר, אולם הוא אינו חוזר למעגל הבית, הרובוט לא ייפסל. במקום זאת אנו נחזיר את הרובוט בחזרה לאופן פעולה סטנדרטית והוא יקבל רק את ניקוד ההפעלה ללא הפחתת גורם אופן נסיעה חזרה.

ניקוד הזמן הממשי (AT) יכלול רק את הזמן שלוקח לרובוט למצוא ולכבות את הנר. הוא לא יכלול את הזמן שלוקח לרובוט לנסיעה חזרה לעיגול הבית. פעולה באופן זה תניב הפחתה של 20% בניקוד. גורם אופן הפעולה עבור ריצה באופן נסיעה חזרה הוא 0.8 (OM=0.8).

<u>ללא חישובי מיקום</u>

רובוטים רבים משתמשים בצורה כלשהי של הסקת מיקום על ידי חישוב (dead-reckoning) בכדי לנוע בתוך הזירה. כלומר, פעם שהרובוטים מוצבים ומכוונים בתחילת הזירה, הם סופרים את המרחק שעברו והזווית שבה הסתובבו ומוסיפים אותם למיקום הקודם שלהם בכדי לקבל את המיקום והכיוון החדשים שלהם. בעוד שזו דרך לגיטימית של תנועה בתוך הזירה בתחרות זו, היא אינה מעשית או שימושית בעולם האמיתי. לכן בכדי לעודד רובוטים להשתמש בשיטות מתוחכמות יותר לקביעת מיקומם בזירה, אנו מעניקים בונוס הפחתת ניקוד לרובוטים שלא משתמשים בשיטת הסקת מיקום מחושבת.

המפתח לשימוש בשיטת הסקה מחושבת היא ידיעה מראש של המרחק לחדרים השונים בזירה. אם תחליט להריץ את הרובוט שלך באופן ללא חישובי מיקום, אנו נציב משטח משופע אחד או יותר במסדרונות הזירה, שתהיה להם השפעה על שינוי המרחק לחדרים.

משטח משופע יוצב באחד מהמסדרונות. רחבו יהיה 46 ס"מ, כך שימלא לחלוטין את המסדרון. יהיו לו צדדים משופעים בצורה עדינה. בגלל הצדדים המשופעים, המרחק הכולל שהרובוט עובר בנסיעתו כשהוא עובר דרך הקטע המשופע יהיה גדול יותר מאשר אם הוא היה עובר רק על רצפה ישרה, ובכך מוגבלת יעילות הסקה המיקום המחושבת. השיפוע יכול להיות כזה שגלגל בצד אחד של הרובוט עשוי לנסוע מרחק גדול יותר מאשר הגלגל שבקצה השני. היות והרובוט לא ידע בדיוק היכן יוצב מסלול גדול יותר זה, יהיה עליו להשתמש בשיטות אחרות זולת הסקה מחושבת בכדי לקבוע את מיקומו וכיוונו בתוך הזירה.

יכולים להיות יותר מאשר משטח משופע אחד במהלך ניסיון הריצה. המקטעים המשופעים יוצבו אך ורק במסדרונות ולא בתוך חדרים. משטח משופע לא יוצב בכניסות לחדרים ולא יחסום כניסות לחדרים. המספר והמיקום של המקטעים המשופעים ישתנה מניסיון ריצה אחד למשנהו אם אופן זה נבחר. המשטחים המשופעים יישארו במקומם במהלך קטע הנסיעה חזרה של ניסיון הריצה. הגובה המרבי של המקטעים המשופעים יהיה קטן

מ- 5 ס"מ. המקטעים המשופעים יהיו מחודדים ויהיה חיבור חלק ככל האפשר עם הרצפה השטוחה. **לא** יהיו מדרגות או נפילות חדות. המידות המדויקות של המקטעים המשופעים לא יהיו ידועות לרובוט לפני תחילת התחרות. השיפוע המרבי של המקטע המשופע יהיה
13 מעלות. המקטע המשופע יהיה צבוע בצבע שחור בדיוק כמו הרצפה. הפעלה מוצלחת באופן זה תקנה 40% הפחתה בניקוד. גורם אופן הפעולה בריצה באופן ללא חישובי מיקום הוא 0.6 (OM=0.6).

<u>רהיטים</u>

באופן זה תהיה יחידה אחת של ריהוט בכל חדר. הרהיטים יוצבו באופן אקראי בחדרים. הרובוט יכול לנגוע ברהיט, אולם הוא לא יכול לדחוף אותו מדרכו. הרהיט יעשה מגליל מתכת בקוטר 11.5 ס"מ, צבוע בצבע צהוב. הגלילים הם בגובה 30 ס"מ ושוקלים 2.3 ק"ג.

האפשרות שיתכן ורהיט יחסום את ראיית הרובוט את הנר או שיתכן והרובוט יאלץ לעקוף את הרהיט בכדי להגיע לנר, היא מה שעושה את אופן הרהיטים למעניין ולאתגר של העולם האמיתי. יתכן ועל הרובוט להביט ממיקומים שונים בחדר בכדי לראות אם הרהיט באמת חוסם את ראייתו את הנר. אם הנר באמת נמצא מאחורי רהיט, הרובוט יצטרך להחליט מה היא הדרך הטובה ביותר לעקוף את הרהיט בכדי להגיע לנר. פעולה מוצלחת באופן זה תניב הפחתה של 50% בניקוד. גורם אופן הפעולה עבור ריצה באופן רהיטים הוא 0.5 (OM=0.5).

<u>נקודת התחלה שרירותית</u>

רק בכדי להפוך את התחרות ליותר מציאותית ומאתגרת, יהיה אופן אחד נוסף אפשרי. אם המתחרים מחליטים לפעול באופן נקודת התחלה שרירותית, השופטים יניחו באופן שרירותי את הרובוט באחד מהחדרים שלא מכיל את הנר והרובוט יתחיל את ניסיון ריצתו מחדר זה. הרובוט יכול להיות מוצב על ידי השופטים בכל כיוון בתוך החדר. אם הרובוט מופעל גם באופן רהיטים במהלך ניסיון הריצה, הרהיטים לא יחסמו לחלוטין את הרובוט מיציאה מהחדר, אולם יתכן ועל הרובוט יהיה לעקוף את הרהיט בכדי לצאת. אם אחרי כיבוי הנר, הרובוט יכול לחזור לחדר שממנו הוא התחיל, אזי הוא השלים בהצלחה את אופן הנסיעה חזרה ויקבל את הבונוס הנוסף של גורם אופן הנסיעה חזרה (0.8). באופן נקודת התחלה שרירותית, סיום הניסיון בעיגול הבית **לא** יחשב לסיום מוצלח של אופן נסיעה חזרה. הרובוט חייב לחזור לחדר שממנו הוא התחיל בכדי לקבל את גורם אופן נסיעה חזרה, אבל הוא **לא** חייב להיות באותו מיקום התחלתי מדויק בתוך החדר. באופן נקודת התחלה שרירותית, גורם החדר (ראה סעיף 19) ייקח בחשבון את החדרים שהרובוט מחפש בהם כשהוא מנסה לגלות את הנר. היות והנר **לא** יהיה בחדר המוצא, יהיו רק 3 חדרים אפשריים אחרים שנשאר לבדוק. לכן גורם החדר יכול להיות 1.0, 0.85 או 0.5. הפעלה מוצלחת באופן נקודת התחלה שרירותית זה, יניב הפחתה של 20% מהניקוד. גורם אופן הפעולה עבור ריצה באופן נקודת התחלה שרירותית הוא 0.8 (OM=0.8).

<u>הפעלה ידנית של הרובוט</u>

אם הרובוט לא רץ באופן הפעלה על ידי צפצוף, הוא חייב להתחיל את פעולתו באופן ידני, כלומר באמצעות לחיצה של נושא משרה רשמי של התחרות על כפתור ה"התחלה".

<u>מחשב חיצוני</u>

אם הרובוט משתמש בחיבור של כבל למחשב חיצוני, אזי המקש היחיד שאפשר ללחוץ עליו בכדי להתחיל את פעולת הרובוט הוא מקש ה- ENTER או RETURN שעל מקלדת המחשב. נושא משרה רשמי של התחרות ילחץ על המקש. כל תוכנית הנחוצה להרצה יש להטעין לרובוט לפני שהוא מוצב בזירה. פעם שהרובוט מונח במקומו והנר מוצב למקומו שלו, ניתן ללחוץ אך ורק על מקש ה-ENTER או RETURN בכדי להתחיל את פעולת הרובוט. אם מכל סיבה שהיא הרובוט אינו מתחיל, ניסיון הריצה יסתיים.

<u>מחשב על הרובוט</u>

אם הרובוט משתמש במחשב פנימי, אזי יכול להיות כפתור אחד ורק אחד שניתן ללחוץ עליו בכדי שהרובוט יתחיל את פעולתו. כפתור זה חייב להיות ממוקם במקום שקל לראות אותו ולהגיע אליו על הרובוט והוא חייב להיות מסומן ב- "START". כל תוכנית נחוצה יש להטעין לרובוט לפני שהוא מוצב בזירה. לאחר שזה נעשה, ניתן ללחוץ אך ורק על כפתור ה"התחלה" בכדי להתחיל בפועל את פעולת הרובוט. אם מכל סיבה שהיא הרובוט אינו מתחיל, ניסיון הריצה יסתיים.

<u>עונשים</u>

מטרת התחרות היא להיות מציאותית ככל האפשר כמו בעולם האמיתי. לכן קיימות שתי פעולות שבעוד שהן אינן בלתי חוקיות, הן אינן מה שיחשב לשיטת פעולה טובה בעולם האמיתי. לכן קיימות נקודות ענישה (PP) שנוספות לזמן הממשי (AT) של כל רובוט שעושה אותן. תוכל עדיין לעשות דברים אילו, אולם תקבל נקודות שיתווספו לניקוד הזמן שלך אם תעשה אותם. אל תיתן לנקודות עונשין אילו להפחיד אותך יותר מדי. נקודות עונשין אילו הן בדרך כלל מחיר קטן שיש לשלם עבור רובוט שבעצם מצליח להשלים את המשימה.

<u>נגיעה בקיר</u>

כל רובוט שנוגע בקיר עם חלק כלשהו של גופו או עם חיישניו, בין אם במכוון ובין אם בשוגג, יוספו לו 5 נקודות לניקוד הזמן הממשי שלו בכל פעם שהוא נוגע בקיר. כל רובוט שמחליק לאורך קיר, תתווסף לו נקודה אחת לניקוד הזמן שלו עבור כל 2 ס"מ של קיר שהוא נוגע בו כשהוא מחליק בצמוד אליו. רובוט יכול עדיין לנגוע בקיר בכדי לכוון את עצמו, אבל הוא יוענש בעבור שעשה כך. (PP=5 לפגיעה ו- PP=1 לכל 2 ס"מ של החלקה).

אין שום עונשים שנלקחים בחשבון עבור פגיעה בקירות בזמן הנסיעה חזרה לעיגול הבית אחרי כיבוי הנר.

<u>נגיעה בנר</u>

כל רובוט שנוגע בנר או בבסיס שלו עם חלק כלשהו מגופו או חיישניו, בין אם במכוון ובין אם בשוגג, בזמן שהנר דלוק, יתווספו לו 50 נקודות לניקוד הזמן הממשי שלו. אם הנגיעה תיקרה אחרי שהנר כובה, לא יהיו נקודות עונשין. נגיעה זו מתייחסת רק לחלק של גוף הרובוט ולא כוללת מים, אויר או חומר אחר שהרובוט יכול להשתמש בו לכיבוי הנר. (PP=50).

<u>גורם חדר</u>

בכדי לעשות את התחרות מציאותית ולעודד יצירת רובוטים חכמים, הוספנו במתכוון אי ודאות לתוך התחרות. הרובוט לא יודע באיזה מארבעת החדרים נמצא הנר. לפעמים לרובוט יש מזל והנר נמצא בחדר הראשון שבו הוא מחפש ולפעמים הרובוט הוא חסר מזל והנר נמצא בחדר הרביעי לפי סדר החיפוש. חוסר ההוגנות שבכך הוא שמציאת הנר בחדר הרביעי שבו הרובוט מחפש היא קשה יותר ולוקחת זמן רב יותר מאשר מציאתו בחדר הראשון שבו הוא מחפש. בכדי להפחית את ההשפעה של ה"מזל" ולתת בונוס מסוים לרובוטים המתוחכמים יותר שיכולים לחפש בהרבה חדרים בהצלחה, יהיה גורם חדר מעורב בניקוד שיוכפל בניקוד הזמן בכדי לקבל את ניקוד הפעולה. ככל שלרובוט יש יותר חדרים לחפש בהם, לפני שהוא מוצא את הנר, כך קטן יותר גורם החדר וכך טוב יותר ניקוד הפעולה.

אם הנר נמצא בחדר הראשון שבו הרובוט מחפש, גורם החדר יהיה 1.0
אם הנר נמצא בחדר השני שבו הרובוט מחפש, גורם החדר יהיה 0.85
אם הנר נמצא בחדר השלישי שבו הרובוט מחפש, גורם החדר יהיה 0.50
אם הנר נמצא בחדר הרביעי שבו הרובוט מחפש, גורם החדר יהיה 0.35

אין זה משנה מהו הסדר שבו הרובוט מחפש בחדרים. הדבר היחיד שמשנה הוא בכמה חדרים חיפש הרובוט לפני שהוא מצא את הנר.

לרובוטים מסוימים יש חיישנים רגישים מאוד שיכולים להגיד להם אם הנר נמצא בחדר רק על ידי מעבר ליד פתח החדר כשהרובוט חולף על פניו. הרובוט לא חייב להיכנס לחדר בכדי שיחשב שחיפש בו. כל רובוט שעובר ליד פתח של חדר שלא עבר על פניו לפני כן, ייחשב ככזה שחיפש בחדר זה. אם הרובוט חיפש כבר בחדר ואז עבר שוב ליד פתחו בדרכו לחדר אחר, חדר זה לא ייחשב פעמיים.

<u>גורם מהימנות</u>

היות וחשוב שכל רובוט לכיבוי שריפות יהיה אמין כמו גם מהיר, כל רובוט שמכבה את הנר בכל שלושת ניסיונותיו, יקבל הפחתה של 10% מתוצאתו הסופית (הסכום של שני ניקודי ההפעלה הטובים ביותר שלו). למשל, לרובוט א' ניקודי פעולה של 25 ו- 24, אבל נכשל לכבות את הנר בניסיונו השלישי. לרובוט ב' יש ניקודי פעולה של 28, 24 ו- 47. מחברים את שני ניקודי הפעולה של רובוט א' בכדי לקבל תוצאה סופית של ניקוד 49 לרובוט א'. מחברים את שתי תוצאות ניקוד הפעולה הטובות ביותר של רובוט ב' ומקבלים 52, אולם היות והוא כיבה את הנר בכל שלושת ניסיונותיו, הוא מקבל הפחתה של 10% מהסיכום הכללי של ניקוד הפעולה שלו. לכן ניקודו הסופי

יהיה 46.8 ורובוט ב' ינצח את רובוט א'. עבור כיבוי הנר בכל שלושת הניסיונות גורם מהימנות (RF) = 0.9, אחרת, RF=1.0.

תהליך הניקוד

יש להכפיל את אופני הפעולה יחדיו בכדי לקבל גורם אופן (MF).
(ללא כבל=0.9, צפצוף =0.95, חזרה=0.8, ללא הסקת מיקום מחושבת=0.6, רהיטים=0.5, נקודת התחלה שרירותית=0.8). אם הרובוט לא מבצע אופני פעולה כלשהם והוא רץ בפעולה סטנדרטית, אזי MF=1.

יש לרשום את הזמן בפועל (AT) בשניות הנדרש לכיבוי הנר.

ג. יש לחבר את נקודות העונשין (PP) ביחד.
פגיעה בקיר =5 נקודות לכל פגיעה.
החלקה לאורך הקיר =נקודה אחת לכל 2 ס"מ.
נגיעה בנר או בבסיסו בעוד הנר דולק = 50 נקודות.

יש לרשום את גורם החדר (RF).
חדר ראשון = 1.0, חדר שני = 0.85, חדר שלישי= 0.50, חדר רביעי = 0.35.

יש לחבר את הזמן בפועל לנקודות העונשין בכדי לקבל את ניקוד הזמן (TS).

$$TS=AT+PP$$

יש להכפיל את ניקוד הזמן, גורם החדר וגורם האופן ביחד בכדי לקבל את ניקוד הפעולה (OS) לניסיון זה.

$$OS=TS \times RF \times MF$$

אחרי שלושה ניסיונות, יש לחבר ביחד את שתי תוצאות ניקוד הפעולה הנמוכות ביותר בכדי לקבל את סכום שני הניסיונות הטובים ביותר (SB2T).

אם לרובוט היו שני ניסיונות מוצלחים (הנר כובה), גורם המהימנות (RF) יהיה 1.0. אם לרובוט היו 3 ניסיונות מוצלחים, גורם המהימנות יהיה 0.9.

יש להכפיל את סכום שני הניסיונות הטובים ביותר בגורם המהימנות בכדי לקבל את התוצאה הסופית (FS).

$$FS=SB2T \times RF$$

הרובוט עם התוצאה הסופית הנמוכה ביותר הוא המנצח.

דוגמאות ניקוד

ניסיון ראשון: אם הרובוט מבצע את ניסיונו הראשון באופנים הסטנדרטי, צפצוף ונסיעה חזרה, לוקח לו דקה אחת ו23 שניות לכבות את הנר בחדר השני תוך שהוא פוגע בקיר 3 פעמים, ניקוד הפעולה שלו עבור ניסיון זה יהיה:

הכפלת אופני הפעולה יחד בכדי לקבל את גורם האופן (MF) (סטנדרטי=1.0, צפצוף=0.95, וחזרה=0.8).

MF=STD x SND x RTN=1.0 x 0.95 x 0.8 = 0.76

רישום הזמן בפועל (AT) בשניות שנדרש לכיבוי הנר.

AT=83

הוספת נקודות העונשין (PP) ביחד (פגיעה בקיר = 5 נקודות / פגיעה)

PP=15

רישום גורם החדר (RF) (חדר שני = 0.85)

RF=0.85

הוספת הזמן בפועל לנקודות העונשין בכדי לקבל את ניקוד הזמן (TS).

TS = AT + PP = 83 + 15 = 98

הכפלת ניקוד הזמן, גורם החדר וגורם האופן ביחד בכדי לקבל את ניקוד הפעולה (OS).

OS = TS x RF x MF = 98 x 0.85 x 0.76 = 63.31

ניסיון שני: אם הרובוט רץ בניסיונו השני באופנים הסטנדרטי, צפצוף, נסיעה חזרה, וללא הסקת מיקום מחושבת, לוקח לו דקה אחת ו- 41 שניות לכבות את הנר בחדר הרביעי תוך שהוא פוגע בנר במקרה, ניקוד הפעולה שלו עבור ניסיון זה יהיה:

הכפלת אופני הפעולה יחד בכדי לקבל את גורם האופן (MF) (סטנדרטי=1.0, צפצוף=0.95, חזרה=0.8, וללא הסקה מחושבת=0.6).

MF=STD x SND x RTN x NDR=1.0 x 0.95 x 0.8 x 0.6 = 0.456

רישום הזמן בפועל (AT) בשניות שנדרש לכיבוי הנר.

AT=101

הוספת נקודות העונשין (PP) ביחד (פגיעה בנר = 50 נקודות)

PP=50

רישום גורם החדר (RF) (חדר רביעי = 0.35)

RF=0.35

הוספת הזמן בפועל לנקודות העונשין בכדי לקבל את ניקוד הזמן (TS).

TS = AT + PP = 101 + 50 = 151

הכפלת ניקוד הזמן, גורם החדר וגורם האופן ביחד בכדי לקבל את ניקוד הפעולה (OS).

OS = TS x RF x MF = 151 x 0.35 x 0.456 = 24.10

ניסיון שלישי: בניסיון השלישי הרובוט ניסה לרוץ באופנים צפצוף, נסיעה חזרה, ורהיטים. הוא כיבה את הנר בחדר הראשון בדקה ועשר שניות, אבל הוא לא חזר לעיגול הבית.

הכפלת אופני הפעולה יחד בכדי לקבל את גורם האופן (MF). הרובוט לא חזר לעיגול הבית ולכן הוא מפסיד את הפחתת אופן הנסיעה חזרה. (סטנדרטי = 1.0, צפצוף = 0.95, ורהיטים = 0.5).
MF=STD x SND x FRN = 1.0 x 0.95 x 0.5 = 0.475

רישום הזמן בפועל (AT) בשניות שנדרש לכיבוי הנר.

AT=70

הוספת נקודות העונשין (PP) ביחד.

PP=0

רישום גורם החדר (RF) (חדר ראשון = 1.0)

RF=1.0

הוספת הזמן בפועל לנקודות העונשין בכדי לקבל את ניקוד הזמן (TS).

$$TS = AT + PP = 70 + 0 = 70$$

הכפלת ניקוד הזמן, גורם החדר וגורם האופן ביחד בכדי לקבל את ניקוד הפעולה (OS).

$$OS = TS \times RF \times MF = 70 \times 1.0 \times 0.475 = 33.25$$

חישובים סופיים: כעת הרובוט סיים שלושה ניסיונות.

אחרי שלושה ניסיונות יש להוסיף את שני ניקודי הפעולה הנמוכים ביותר בכדי לקבל את סכום שני הניסיונות הטובים ביותר (SB2T).

$$SB2T = 24.10 + 33.25 = 57.35$$

היות והרובוט השלים שלושה ניסיונות מוצלחים (שבכולם כובה הנר), גורם המהימנות (RF) הוא 0.9.

$$RF=0.9$$

יש להכפיל את סכום שני הניסיונות הטובים ביותר בגורם המהימנות בכדי לקבל את התוצאה הסופית (FS).

$$FS = SB2T \times RF = 57.35 \times 0.9 = 51.62$$

הניקוד הסופי של הרובוט יהיה 51.62.

תהליך ניקוד זה תוכנן לנסות לאפשר כניסות רובוטים בכל הרמות השונות של תחכום להתחרות באותה תחרות. כל אחד יכול לנצח.

לכל רובוט יש 3 הזדמנויות למצוא ולכבות את הנר. הרובוט יכול לפעול באופנים שונים במהלך כל ניסיון. הסכום של ניקוד שני ניסיונותיו הטובים ביותר מתוך שלושה ניסיונות ישמש לקביעת המנצח.

נספח י' – תוכנת הרובוט "קרפלעך"

התוכנה המופיעה בנספח זה הינה התוכנה בגרסתה האחרונה אותה הרצנו בתחרות העולמית בארצות הברית. חלק מהתוכנה כבר מופיע בפרקים שונים בתוך העבודה אך כאן היא מופיעה בשלמותה.

התוכנה הראשית

```
#include c:\ax12\include\EQUATES.asm
#include c:\ax12\include\INITS.asm
#include c:\ax12\include\HELPFUNC.asm
#include c:\ax12\include\SENSORS.asm
#include c:\ax12\include\motors.asm
#include c:\ax12\include\IR_OVF.asm
#include c:\ax12\include\Tikunim.asm
#include c:\ax12\include\FIRE.asm
;#include c:\ax12\include\PRINT.asm

PROGRAM:
INT_RUN:
             CLR   ROOM
             CLR   PLZ_TURN
             MOVB  #$21,WALL          ;2A
             MOVB  #$90,TURNWEAK
             MOVW  #9000,E_TURN       ;6
             JSR   X_L
             CLI
SECT1:
             JSR   TKNBYRGT
             TST   PLZ_TURN
             BEQ   SECT1
             JSR   M_BRAKE
             MOVB  #$03,PWEN
             JSR   TURN_LEFT
             JSR   L_BRAKE
             JSR   Q_ENGAGE
             JSR   UV_ON

SECT2:
             JSR   TKNBYRGT
             JSR   US1
             LDX   US_READ
             CPX   #$400
             BPL   SECT2

SECT3:
             JSR   TKNBYRGT
             JSR   US1
             LDX   US_READ
             CPX   #$400
             BMI   SECT3
             MOVB  #$1A,WALL
             CLR   PLZ_TURN
SECT3_1:
             JSR   TKNBYRGT
             TST   PLZ_TURN
             BEQ   SECT3_1
             JSR   UV_OFF
             CLR   PLZ_TURN
```

```
                MOVB #$1F,WALL
                MOVB #$90,TURNWEAK   ;A0
                MOVW #9100,E_TURN     ;8500

SECT4:
                JSR  TKNBYRGT
                TST  PLZ_TURN
                BEQ  SECT4
                JSR  CHECK_UV
                INC  ROOM
                JSR  TURN_LEFT
                JSR  L_BRAKE
        ;        JSR  MOVBYTES
                JSR  Q_ENGAGE
SECT5:
                JSR   TKNBYRGT
                JSR   US0
                LDX   US_READ
                CPX   #$400
                BHI   SECT5
                MOVB  #PWDREG,PWDTY0
                MOVB  #PWDREG,PWDTY1

SECT6:
                JSR   TKNBYLFT
                JSR   US2
                LDX   US_READ
                CPX   #$400
                BMI   SECT6
SECT7:
                MOVB  #PWDREG,PWDTY0
                MOVB  #PWDREG,PWDTY1
                JSR   US2
                LDX   US_READ
                CPX   #$400
                BPL   SECT7

                CLR     PLZ_TURN
                MOVB    #$20,WALL
                MOVB    #$90,TURNWEAK
                MOVW    #11000,E_TURN
SECT8:
                JSR     TKNBYRGT
                TST     PLZ_TURN
                BEQ     SECT8
                JSR     TURN_RIGHT         ; Turn Into Room 4
                JSR     L_BRAKE
OTR_STRT:       MOVB    #$03,PWEN
                JSR     MAKOM_L
                JSR     Q_ENGAGE
SECT8_1:        JSR     PAS_LAV         ; Check For Pas Lavan
                TST     PAS_FLAG
                BEQ     SECT8_1
                LDX     #1200
                JSR     L_ENCO
                JSR     PLS_CNT
                JSR     L_BRAKE
                MOVB    #$03,PWEN
                JSR     MAKOM_L
                JSR     UV_ON_L
SECT8B:
              MOVB  #$03,PWEN
```

```
            MOVB   #$13,PORTPP
            MOVB   #$98,PWDTY1
            MOVB   #$98,PWDTY0
            LDX    #6700             ;  Turn 90 Degrees
            JSR    L_ENCO
            JSR    PLS_CNT
            JSR    BRK_TRNB
            JSR    L_BRAKE

            MOVB   #$03,PWEN
            MOVB   #$43,PORTPP
            MOVB   #$98,PWDTY1
            MOVB   #$98,PWDTY0
            LDX    #700
            JSR    L_ENCO
            JSR    PLS_CNT
            JSR    BRK_TRNB
            JSR    L_BRAKE


            JSR    UV_OFF
            JSR    CHECK_UV
            INC    R00M
            MOVB   #$03,PWEN
            MOVB   #$13,PORTPP
            MOVB   #$C8,PWDTY1
            MOVB   #$C8,PWDTY0
            LDX    #3000      ;6      ;  Turn 90 Degrees
            JSR    L_ENCO
            JSR    PLS_CNT
            JSR    BRK_TRNB
            JSR    L_BRAKE
            MOVB   #$03,PWEN
            JSR    MAKOM_R
            CLR    PLZ_TURN             ;  Clear Value
            MOVB   #$24,WALL            ;  Enter Values For Wall

SECT9:
            JSR     Q_ENGAGE
SECT9_1:    JSR     TKNBYRGT
            TST     PLZ_TURN
            BEQ     SECT9_1             ;  Taken By Right Till Turn
            JSR     L_BRAKE
            MOVW    #8750,E_TURN
            JSR     TURN_LEFT_B
            JSR     L_BRAKE             ;  Do Turn  ;M
            MOVB    #$03,PWEN
            JSR     MAKOM_R             ;  Taken Bamakom By Right
            JSR     L_BRAKE
            JSR     Q_ENGAGE
            MOVB    #$B0,PWDTY0
            MOVB    #$B0,PWDTY1
SECTA:
            JSR     US3
            LDX     US_READ
            CPX     #$400
            BLO     SECTA
            JSR     UV_ON
            MOVB    #$A0,PWDTY0
            MOVB    #$A0,PWDTY1
SECTA2:
            JSR     US2
```

```
            LDX     US_READ
            CPX     #$400
            BLO     SECTA2
            MOVW    #1500,E_TURN
            JSR     TURN_RIGHT
            JSR     L_BRAKE
            MOVB    #$03,PWEN
            JSR     MAKOM_L
            MOVB    #$03,PWEN
            MOVB    #$43,PORTPP
            MOVB    #PWDREG,PWDTY1
            MOVB    #PWDREG,PWDTY0
            LDX     #1200
            JSR     L_ENCO
            JSR     PLS_CNT
            JSR     BRK_TRNB
            JSR     L_BRAKE
            JSR     UV_OFF
            JSR     CHECK_UV
            INC     ROOM
            JSR     Q_ENGAGE

;=================================
;===========TO ROOM 2!!!!========
;=================================
SECTE:
            JSR     TKNBYLFT            ; Taken By Left to room 2
            JSR     US1
            LDX     US_READ
            CPX     #$450
            BMI     SECTE
            JSR     L_BRAKE
            MOVB    #$03,PWEN
            JSR     MAKOM_R            ;  Taken Bamakom Before Turn
            MOVB    #$03,PWEN
            JSR     MAKOM_R            ;  Taken Bamakom Before Turn
            MOVB    #$03,PWEN
            MOVW    #11000,E_TURN
            JSR     TURN_LEFT_B       ;  Turn To Corridor 2
            JSR     L_BRAKE
            JSR     Q_ENGAGE
;             JSR     MOVBYTES

SECTF:      JSR     US0               ; Wait Till In Corridor
            LDX     US_READ
            CPX     #$360
            BHI     SECTF
            CLR     PLZ_TURN
            MOVB    #$21,WALL
SECT11:
            JSR     TKNBYLFT
            TST     PLZ_TURN
            BEQ     SECT11
            JSR     L_BRAKE
            MOVW    #9250,E_TURN      ;8150
            JSR     TURN_RIGHT_B
            JSR     L_BRAKE
            MOVB    #$03,PWEN
            JSR     MAKOM_L
            JSR     Q_ENGAGE
SECT12:
            JSR     PAS_LAV
```

```
            TST     PAS_FLAG
            BEQ     SECT12
            JSR     UV_ON
            JSR     L_BRAKE
SECT13:
            JSR     CHECK_UV
            BRA     SECT13
```

Tikunim.asm

```
TKNBYRGT:
             PSHD
             PSHX
             JSR     US2
             LDD     US_READ
             JSR     D_INTO_A
             STAA    BACK_DIS
             JSR     US3
             LDD     US_READ
             JSR     D_INTO_A
             STAA    FRNT_DIS
             LDAB    BACK_DIS
             SBA
             JSR     ABS_A
             STAA    DIFFER
             CMPA    #$4
             BPL     TKN_ANGL

TKN_MID:
             MOVB    #$58,PWD4TKN
             LDAA    FRNT_DIS
             LDAB    BACK_DIS
             ABA
             LSRA
             CMPA    #$16
             BLO     TKN_LFT
             MOVB    #$6A,PWD4TKN
             CMPA    #$20
             BLO     TKN_LFT
             MOVB    #$58,PWD4TKN
             CMPA    #$31
             BHI     TKN_RGT
             MOVB    #$74,PWD4TKN
             CMPA    #$27
             BHI     TKN_RGT
             LDAA    DIFFER
             CMPA    #$1
             BHI     TKN_ANGL


TKN_ANGL:
             LDX     #DUTY4TKN
             LDAB    A,X
             STAB    PWD4TKN

CHS_TKN:     TST     WR_2_TKN
             BEQ     TKN_LFT
TKN_RGT:
             MOVB    PWD4TKN,PWDTY0
             MOVB    #PWDREG,PWDTY1
             BRA     END_TKN
TKN_LFT:
             MOVB    PWD4TKN,PWDTY1
             MOVB    #PWDREG,PWDTY0
             BRA     END_TKN

NO_TKN:
           MOVB   #PWDREG,PWDTY0
           MOVB   #PWDREG,PWDTY1
```

```
END_TKN:
           PULX
           PULD
           RTS


TKNBYLFT:
             PSHD
             PSHX
             JSR       US0
             LDD       US_READ
             JSR       D_INTO_A
             STAA      BACK_DIS
             JSR       US1
             LDD       US_READ
             JSR       D_INTO_A
             STAA      FRNT_DIS
             LDAB      BACK_DIS
             SBA
             COMA
             JSR       ABS_A
             STAA      DIFFER
             CMPA      #$4
             LBMI      TKN_MID2
             LBRA      TKN_ANGL

TKN_MID2:
             MOVB      #$70,PWD4TKN
             LDAA      FRNT_DIS
             LDAB      BACK_DIS
             ABA
             LSRA
             CMPA      #$1A
             LBLO      TKN_RGT
             MOVB      #$70,PWD4TKN
             CMPA      #$25
             LBHI      TKN_LFT
             MOVB      #$A0,PWD4TKN
             LDAA      DIFFER
             CMPA      #$1
             LBPL      TKN_ANGL
             LBRA      NO_TKN


;==================================
;            TAKEN BAMAKOM
;==================================

MAKOM_L:
             CLR      MAKM_MONE
             MOVB     #$FF,MAKOM_C
MAKOM_L2:
             INC      MAKM_MONE
             LDAA     MAKM_MONE
             CMPA     #MAKOM_OVF
             LBEQ     M_RESCUE
             JSR      US0
             LDD      US_READ
             JSR      D_INTO_A
             STAA     BACK_DIS
             JSR      US1
             LDD      US_READ
```

```
            JSR     D_INTO_A
            STAA    FRNT_DIS
            LDAB    BACK_DIS
            SBA
            JSR     ABS_A
            STAA    DIFFER
            CMPA    #$2
            BMI     DO_NOT
            TST     WR_2_TKN
            BNE     DO_LEFT
            BRA     DO_RIGHT

MAKOM_R:
            CLR     MAKM_MONE
            MOVB    #$00,MAKOM_C
MAKOM_R2:
            INC     MAKM_MONE
            LDAA    MAKM_MONE
            CMPA    #MAKOM_OVF
            BEQ     M_RESCUE
            JSR     US2
            LDD     US_READ
            JSR     D_INTO_A
            STAA    BACK_DIS
            JSR     US3
            LDD     US_READ
            JSR     D_INTO_A
            STAA    FRNT_DIS
            LDAB    BACK_DIS
            SBA
            JSR     ABS_A
            STAA    DIFFER
            CMPA    #$2
            BMI     DO_NOT
            TST     WR_2_TKN
            BNE     DO_RIGHT
            BRA     DO_LEFT

DO_RIGHT:
          MOVB   #$88,PWDTY1
          MOVB   #$88,PWDTY0
          MOVB   #$13,PORTPP
          TST    MAKOM_C
          BEQ    MAKOM_R2
          LBRA   MAKOM_L2

DO_LEFT:
          MOVB   #$88,PWDTY1
          MOVB   #$88,PWDTY0
          MOVB   #$43,PORTPP
          TST    MAKOM_C
          BEQ    MAKOM_R2
          LBRA   MAKOM_L2

DO_NOT:
          JSR    L_BRAKE
          MOVB   #$00,PWDTY1
          MOVB   #$00,PWDTY0
          MOVB   #$00,PORTPP
          MOVB   #$00,PWEN
          RTS
```

```
M_RESCUE:
          MOVB   #$03,PORTPP
          MOVB   #$B8,PWDTY1
          MOVB   #$B8,PWDTY0
          JSR    L_ENCO
          LDX    #200
          JSR    PLS_CNT
          TST    MAKOM_C
          LBEQ   MAKOM_R
          LBRA   MAKOM_L

;==================================
;             TAKEN BACK
;==================================

TKN_BCK:
          JSR    READ_ATD
          LDAB   ADR4H
          LDAA   ADR5H
          SBA
          BLO    TKN_ST_R
          TSTA
          BHI    TKN_ST_L
          MOVB   #$03,PORTPP
          MOVB   #$A0,PWDTY0
          MOVB   #$A0,PWDTY1
          RTS
TKN_ST_L:
          MOVB   #$A0,PWDTY1
          MOVB   #$A0,PWDTY0
          MOVB   #$13,PORTPP
          BRA    TKN_BCK
TKN_ST_R:
          MOVB   #$A0,PWDTY0
          MOVB   #$A0,PWDTY1
          MOVB   #$43,PORTPP
          BRA    TKN_BCK



DUTY4TKN:
         DB   200
         DB   200
         DB   180
         DB   176
         DB   170
         DB   163
         DB   156
         DB   148
         DB   140
         DB   130
         DB   120
         DB   110
         DB   110
         DB   110
         DB   110
         DB   110
         DB   110
```

Equates.asm

```
RAMSTRT:        EQU     $0800   ; start of internal ram
MONRAM:         EQU     $0A00   ; monitor ram start
RAMSIZE:        EQU     $0400   ; Size of internal ram
REGBS:          EQU     $0000   ; start of registers
ROMBS:          EQU     $FD80   ; start of initialization rom
DSTREE:         EQU     $1000   ; start of eeprom
DENDEE:         EQU     $1FFF   ; end of eeprom
STACK:          EQU     RAMSTRT+RAMSIZE ; top of stack
                LDS     $7000

MONSTRT:        EQU     $2000   ; start of this program if running
under debugger
PRGSTRT:        EQU     $8000   ; start of this program if running
stand-alone

IBUFSIZ:        EQU     35      ; input buffer size
EOT:            EQU     $04     ; end of text/table character

;*******************************************************************
; define I/O registers...
;*******************************************************************
PORTA:  EQU     REGBS+0         ;port A = Address lines A8 - A15
PORTB:  EQU     REGBS+1         ;port B = Address lines A0 - A7
DDRA:   EQU     REGBS+2         ;port A direction register
DDRB:   EQU     REGBS+3         ;port A direction register
PORTC:  EQU     REGBS+4         ;port C = Data 7-15 wide,Data 0-
7narrow
PORTD:  EQU     REGBS+5         ;port D = Data 0-7 wide
DDRC:   EQU     REGBS+6         ;port C direction register
DDRD:   EQU     REGBS+7         ;port D direction register
PORTE:  EQU     REGBS+8         ;port E = mode,IRQandcontrolsignals
DDRE:   EQU     REGBS+9         ;port E direction register
PEAR:   EQU     REGBS+$A        ;port E assignments
MODE:   EQU     REGBS+$B        ;Mode register
PUCR:   EQU     REGBS+$C        ;port pull-up control register
RDRIV:  EQU     REGBS+$D        ;port reduced drive control register
INITRM: EQU     REGBS+$10       ;Ram location register
INITRG: EQU     REGBS+$11       ;Register location register
INITEE: EQU     REGBS+$12       ;EEprom location register
MISC:   EQU     REGBS+$13       ;Miscellaneous Mapping control
RTICTL: EQU     REGBS+$14       ;Real time clock control
RTIFLG: EQU     REGBS+$15       ;Real time clock flag
COPCTL: EQU     REGBS+$16       ;Clock operating properly control
COPRST: EQU     REGBS+$17       ;COP reset register

INTCR:  EQU     REGBS+$1E       ;interrupt control register
HPRIO:  EQU     REGBS+$1F       ;high priority reg
KWIED:  EQU     REGBS+$20       ;Key wake-up port D enable
KWIFD:  EQU     REGBS+$21       ;Key wake-up port D flags
PORTH:  EQU     REGBS+$24       ;port H = keypad port
DDRH:   EQU     REGBS+$25       ;port H direction register
KWIEH:  EQU     REGBS+$26       ;Key wake-up port H enable
KWIFH:  EQU     REGBS+$27       ;Key wake-up port H flags
PORTJ:  EQU     REGBS+$28       ;port J = Keypad / Serial ctl lines
DDRJ:   EQU     REGBS+$29       ;port J direction register
KWIEJ:  EQU     REGBS+$2A       ;Key wake-up port J enable
KWIFJ:  EQU     REGBS+$2B       ;Key wake-up port J flags
KPOLJ:  EQU     REGBS+$2C       ;port J wake-up polarity
PUPSJ:  EQU     REGBS+$2D       ;port J pull-up/down select
```

```
PULEJ:   EQU      REGBS+$2E        ;port J Pull-up/down enable
PORTF:   EQU      REGBS+$30        ;port F = Chip selects
PORTG:   EQU      REGBS+$31        ;port G = Address lines A16 - A21
DDRF:    EQU      REGBS+$32        ;port F direction register
DDRG:    EQU      REGBS+$33        ;port G direction register
DPAGE:   EQU      REGBS+$34        ;CSD chip select page register
PPAGE:   EQU      REGBS+$35        ;CSP chip select page register
EPAGE:   EQU      REGBS+$36        ;CS3/Epage chip select page register
WINDEF:  EQU      REGBS+$37        ;memory page window enable register
MXAR:    EQU      REGBS+$38        ;memory expansion enable A16-A21
CSCTL0:  EQU      REGBS+$3C        ;chip select control register
CSCTL1:  EQU      REGBS+$3D        ;chip select control register
CSSTR0:  EQU      REGBS+$3E        ;chip select stretch register
CSSTR1:  EQU      REGBS+$3F        ;chip select stretch register

LDV:     EQU      REGBS+$40        ;PLL loop divider value hi
;LDV     EQU      REGBS+$41        PLL loop divider value lo
RDV:     EQU      REGBS+$42        ;PLL reference divider value hi
;RDV     EQU      REGBS+$43        PLL reference divider value lo
CLKCTL:  EQU      REGBS+$47        ;System clock control
PWCLK:    EQU     REGBS+$40
PWPOL:    EQU     REGBS+$41
PWEN:     EQU     REGBS+$42
PWSCAL0:  EQU     REGBS+$44

PWPER0:   EQU     REGBS+$4C
PWDTY0:   EQU     REGBS+$50
PWPER1:   EQU     REGBS+$4D
PWDTY1:   EQU     REGBS+$51
PORTPP:   EQU     REGBS+$56
DDRP:     EQU     REGBS+$57

ATDCTL0:  EQU     REGBS+$60        ;ADC control 0 (reserved)
ATDCTL1:  EQU     REGBS+$61        ;ADC control 1 (reserved)
ATDCTL2:  EQU     REGBS+$62        ;ADC control 2
ATDCTL3:  EQU     REGBS+$63        ;ADC control 3
ATDCTL4:  EQU     REGBS+$64        ;ADC control 4
ATDCTL5:  EQU     REGBS+$65        ;ADC control 5
ATDSTAT:  EQU     REGBS+$66        ;ADC status register hi
;ATDSTAT  EQU     REGBS+$67        ADC status register lo
ATDTEST:  EQU     REGBS+$68        ;ADC test (reserved)
;ATDTEST  EQU     REGBS+$69
PORTAD:   EQU     REGBS+$6F        ;port ADC = input only
ADR0H:   EQU      REGBS+$70        ;ADC result 0 register
ADR0L:   EQU      REGBS+$71        ;ADC result 0 register LOW
ADR1H:   EQU      REGBS+$72        ;ADC result 1 register
ADR2H:   EQU      REGBS+$74        ;ADC result 2 register
ADR3H:   EQU      REGBS+$76        ;ADC result 3 register
ADR4H:   EQU      REGBS+$78        ;ADC result 4 register
ADR5H:   EQU      REGBS+$7A        ;ADC result 5 register
ADR6H:   EQU      REGBS+$7C        ;ADC result 6 register
ADR7H:   EQU      REGBS+$7E        ;ADC result 7 register
TIOS:    EQU      REGBS+$80        ;timer input/output select
CFORC:   EQU      REGBS+$81        ;timer compare force
OC7M:    EQU      REGBS+$82        ;timer output compare 7 mask
OC7D:    EQU      REGBS+$83        ;timer output compare 7 data
TCNT:    EQU      REGBS+$84        ;timer counter register hi
;TCNT    EQU      REGBS+$85        timer counter register lo
TSCR:    EQU      REGBS+$86        ;timer system control register
TQCR:    EQU      REGBS+$87        ;reserved
TCTL1:   EQU      REGBS+$88        ;timer control register 1
TCTL2:   EQU      REGBS+$89        ;timer control register 2
```

```
TCTL3:  EQU     REGBS+$8A       ;timer control register 3
TCTL4:  EQU     REGBS+$8B       ;timer control register 4
TMSK1:  EQU     REGBS+$8C       ;timer interrupt mask 1
TMSK2:  EQU     REGBS+$8D       ;timer interrupt mask 2
TFLG1:  EQU     REGBS+$8E       ;timer flags 1
TFLG2:  EQU     REGBS+$8F       ;timer flags 2
TC0:    EQU     REGBS+$90       ;timer capture/compare register 0
;TC0    EQU     REGBS+$91
TC1:    EQU     REGBS+$92       ;timer capture/compare register 1
;TC1    EQU     REGBS+$93
TC2:    EQU     REGBS+$94       ;timer capture/compare register 2
;TC2    EQU     REGBS+$95
TC3:    EQU     REGBS+$96       ;timer capture/compare register 3
;TC3    EQU     REGBS+$97
TC4:    EQU     REGBS+$98       ;timer capture/compare register 4
;TC4    EQU     REGBS+$99
TC5:    EQU     REGBS+$9A       ;timer capture/compare register 5
;TC5    EQU     REGBS+$9B
TC6:    EQU     REGBS+$9C       ;timer capture/compare register 6
;TC6    EQU     REGBS+$9D
TC7:    EQU     REGBS+$9E       ;timer capture/compare register 7
;TC7    EQU     REGBS+$9F
PACTL:  EQU     REGBS+$A0       ;pulse accumulator controls
PAFLG:  EQU     REGBS+$A1       ;pulse accumulator flags
PACNT:  EQU     REGBS+$A2       ;pulse accumulator counter
;PACNT  EQU     REGBS+$A3
TIMTST: EQU     REGBS+$AD       ;timer test register
PORTT:  EQU     REGBS+$AE       ;port T = Timer port
DDRT:   EQU     REGBS+$AF       ;port T direction register
SC0BDH: EQU     REGBS+$C0       ;sci 0 baud reg hi byte
SC0BDL: EQU     REGBS+$C1       ;sci 0 baud reg lo byte
SC0CR1: EQU     REGBS+$C2       ;sci 0 control1 reg
SC0CR2: EQU     REGBS+$C3       ;sci 0 control2 reg
SC0SR1: EQU     REGBS+$C4       ;sci 0 status reg 1
SC0SR2: EQU     REGBS+$C5       ;sci 0 status reg 2
SC0DRH: EQU     REGBS+$C6       ;sci 0 data reg hi
SC0DRL: EQU     REGBS+$C7       ;sci 0 data reg lo
SC1BDH: EQU     REGBS+$C8       ;sci 1 baud reg hi byte
SC1BDL: EQU     REGBS+$C9       ;sci 1 baud reg lo byte
SC1CR1: EQU     REGBS+$CA       ;sci 1 control1 reg
SC1CR2: EQU     REGBS+$CB       ;sci 1 control2 reg
SC1SR1: EQU     REGBS+$CC       ;sci 1 status reg 1
SC1SR2: EQU     REGBS+$CD       ;sci 1 status reg 2
SC1DRH: EQU     REGBS+$CE       ;sci 1 data reg hi
SC1DRL: EQU     REGBS+$CF       ;sci 1 data reg lo
SP0CR1: EQU     REGBS+$D0       ;spi 0 control1 reg
SP0CR2: EQU     REGBS+$D1       ;spi 0 control2 reg
SP0BR:  EQU     REGBS+$D2       ;spi 0 baud reg
SP0SR:  EQU     REGBS+$D3       ;spi 0 status reg hi
SP0DR:  EQU     REGBS+$D5       ;spi 0 data reg
PORTS:  EQU     REGBS+$D6       ;port S = Serial port
DDRS:   EQU     REGBS+$D7       ;port S direction register
EEMCR:  EQU     REGBS+$F0       ;EEprom mode control
EEPROT: EQU     REGBS+$F1       ;EEprom  block protect reg
EETST:  EQU     REGBS+$F2       ;EEprom test register
EEPROG: EQU     REGBS+$F3       ;EEprom program reg
PDLC:   EQU     REGBS+$FE       ;PortDLC
DDRDLC: EQU     REGBS+$FF       ;Port DLC Data Direction Control
INT_CH0:  EQU  $0B2E
INT_CH1:  EQU  $0B2C
INT_CH2:  EQU  $0B2A
INT_CH3:  EQU  $0B28
```

```
INT_OVF:  EQU  $0B1E
;______________________________________________
;______________________________________________
;______________________________________________
;______________________________________________
;****************
;* memory table  *
;****************
USFLG:     EQU      DSTREE+$00
USDNW:     EQU      DSTREE+$01
USUPW:     EQU      DSTREE+$02
US_READ:   EQU      DSTREE+$03    ;A SAVED PLACE 4 SENS READ
;US_READ:  EQU      DSTREE+$04    ;16BIT
DSIGN:     EQU      DSTREE+$05    ;FOR ABS THE SIGN THAT D HAD
NTSTRT:    EQU      DSTREE+$06    ;FOR TIKUN GOINT STRAIGHT OR NOT
TEMP16B2:  EQU      DSTREE+$07    ;FOR ONE TIME USE
;TEMP16B2:  EQU      DSTREE+$08
ENC_MONE:  EQU      DSTREE+$09    ;Encoder Mone
TEMP16B1:  EQU      DSTREE+$0A    ;FOR ONE TIME USE
;TEMP16B1:  EQU      DSTREE+$0B
USAVG:     EQU      DSTREE+$0C
;USAVG:     EQU      DSTREE+$0D
TKN_DUTY:  EQU      DSTREE+$0E
ENC_PULS:  EQU      DSTREE+$0F
ENC_FLAG:  EQU      DSTREE+$10
TEMP8B1:   EQU      DSTREE+$11
TEMP8B2:   EQU      DSTREE+$12
DSIGN2:    EQU      DSTREE+$13      ;ANOTHER SIGN FOR D FOR TIKKUNIM
DIF_TIK:   EQU      DSTREE+$14
;DIF_TIK:   EQU      DSTREE+$15
WLL_TIK:   EQU      DSTREE+$16
;WLL_TIK:   EQU      DSTREE+$17
US_INIT:    EQU      DSTREE+$18
AVG1:       EQU      DSTREE+$19
AVG2:       EQU      DSTREE+$1A
AVG3:       EQU      DSTREE+$1B
AVG4:       EQU      DSTREE+$1C
AVG5:       EQU      DSTREE+$1D
AVG6:       EQU      DSTREE+$1E
AVG7:       EQU      DSTREE+$1F
AVG8:       EQU      DSTREE+$20
AVG_SUM1:   EQU      DSTREE+$21
AVG_SUM2:   EQU      DSTREE+$22
L_MONE:     EQU      DSTREE+$23
L_MONE2:     EQU      DSTREE+$24
R_MONE:     EQU      DSTREE+$25
R_MONE2:     EQU      DSTREE+$26
WHAT_US:    EQU      DSTREE+$27
;WHAT_US:    EQU      DSTREE+$28
OVF_MONE:   EQU      DSTREE+$29
E_TURN:   EQU      DSTREE+$2A
;E_TURN:   EQU      DSTREE+$2B
;WHATCTL was here
SECTION:    EQU      DSTREE+$2C
R_SPEED:    EQU      DSTREE+$2D
L_SPEED:    EQU      DSTREE+$2E
W_2_TRN:    EQU      DSTREE+$2F
IR_AVG:     EQU      DSTREE+$30
SECTIONS:   EQU      DSTREE+$31
BCK_FWD:    EQU      DSTREE+$32
LFT_DUTY:   EQU      DSTREE+$33
;LFT_DUTY:    EQU      DSTREE+$34
```

```
RGT_DUTY:  EQU     DSTREE+$35
;RGT_DUTY:  EQU     DSTREE+$36
BF_4_TKN:  EQU     DSTREE+$37
MM_4_TK1:  EQU     DSTREE+$38
;MM_4_TK1:  EQU     DSTREE+$39
MM_4_TK2:  EQU     DSTREE+$3A
;MM_4_TK2:  EQU     DSTREE+$3B
MM_4_WTK:  EQU     DSTREE+$3C
MM_4_DT:   EQU     DSTREE+$3D
;MM_4_DT:   EQU     DSTREE+$3E
DID_TURN:  EQU     DSTREE+$3F
SECT_J:    EQU     DSTREE+$40
PAS_FLAG:  EQU     DSTREE+$41
PYRO_FLG:  EQU     DSTREE+$42
BACK_DIS:  EQU     DSTREE+$43
FRNT_DIS:  EQU     DSTREE+$44
DIFFER:    EQU     DSTREE+$45
WR_2_TKN:  EQU     DSTREE+$46
PWD4TKN:   EQU     DSTREE+$47
UV_FLAG:   EQU     DSTREE+$48
WALL:      EQU     DSTREE+$49
TURNWEAK:  EQU     DSTREE+$4A
ERR_NOWL:  EQU     DSTREE+$4B
ERR_NOWR:  EQU     DSTREE+$4C
ERR_LSTL:  EQU     DSTREE+$4D
ERR_LSTR:  EQU     DSTREE+$4E
D4PID_R:   EQU     DSTREE+$4F
D4PID_L:   EQU     DSTREE+$50
MAKOM_C:   EQU     DSTREE+$51
ERR_SUML:  EQU     DSTREE+$52
ERR_SUMR:  EQU     DSTREE+$53
SPID4TKN:  EQU     DSTREE+$54
PLZ_TURN:  EQU     DSTREE+$55
I4PID_R:   EQU     DSTREE+$56
I4PID_L:   EQU     DSTREE+$57
TKN_FLG:   EQU     DSTREE+$58
TCTLPLCE:  EQU     DSTREE+$59
CNDL_OFF:  EQU     DSTREE+$5A
AKOV_SIDE: EQU     DSTREE+$5B
R00M:      EQU     DSTREE+$5C
MAKM_MONE: EQU     DSTREE+$5D
WHO_SENT:  EQU     DSTREE+$5E
;*****************
;*  const table  *
;*****************
DUTY_MAX: EQU  $F6
DUTY_MIN: EQU  $40
MAX_SPID: EQU  $88
STRT_TRN: EQU  $600
PWDREG:   EQU  $E6       ;ROBOT DO 20ד PER SEC
PWDTRNW:  EQU  $09
PWDTRNS:  EQU  $E6
ENCTURN:  EQU  12800     ;THE ENC READ FOR A TURN   (BIG R)
ENCTURNS: EQU  3000     ;THE ENC READ FOR A TURN   (BIG R)
ENC180T:  EQU  19500
;ENC_PUL0: EQU  $02
ENC_PUL0: EQU  $F2
ENC_FLG0: EQU  $01
;ENC_PUL1  EQU  $08
ENC_PUL1  EQU  $F8
ENC_FLG1: EQU  $02
US_AVG_R: EQU  $03       ; Number of reads to memuza in US
```

```
TRAKLINM: EQU  $210      ; Middle of misdaron
TEN_CM:   EQU  $5000
TRN_1_RM: EQU  $9C0      ;DISTANE FROM WHICH TO TURN TO FIRST ROOM
WALL_AKOV: EQU  $32    ; Wall For Akiva    ; 30
PWD_AKOV:  EQU  $B8    ; Duty For Akiva
REG_SPID: EQU  135
MAKOM_OVF: EQU  $20   ; Numbers of tikum bamakom to do rescue
```

Fire.asm

```
TURN_OFF:
            JSR     L_BRAKE
            LDAA    PDLC
            ORAA    #$40
            STAA    PDLC
            LDX     #$FFFF
            LDY     #$15
            JSR     DLOOP
            JSR     L_BRAKE
;TURN_OFF2: MOVB    #$03,PWEN
;           MOVB    #$A0,PWDTY0
;           MOVB    #$A0,PWDTY1
;           MOVB    #$43,PORTPP
;           LDX     #$8000      ;  =\
;           JSR     L_ENCO      ;  =->  CHNG TO UV OFF
;           JSR     PLS_CNT     ;  =/
TURN_OFF3: LDAA    PDLC
            ANDA    #$BF
            STAA    PDLC
            JSR     M_BRAKE
          ; JSR     RTRN_OM
            SWI

;=========================================
;                   PYRO
;=========================================
PYRO_CHK:
            CLR     PYRO_FLG
            JSR     READ_ATD
            LDAA    ADR6H
            CMPA    #$18
            BLO     CNDL_R
            CMPA    #$E0
            BHI     CNDL_R
            LDAA    ADR7H
            CMPA    #$25
            BLO     CNDL_L
            CMPA    #$C0
            BHI     CNDL_L
            TST     WHO_SENT
            BEQ     WAIT2CND
            JMP     END_INT_F
CNDL_L:
            MOVB    #$43,PYRO_FLG
            BRA     GO_CNDL
CNDL_R:
            MOVB    #$13,PYRO_FLG
GO_CNDL:
            JSR     L_BRAKE
            MOVB    #$03,PWEN
            LDAA    PYRO_FLG
            STAA    PORTPP
            MOVB    #$A0,PWDTY0
            MOVB    #$A0,PWDTY1
            CMPA    #$20
            BMI     PYRO_L
PYRO_L:     JSR     L_ENCO
            LDX     #3000
            BRA     PYRO_CONT
```

```
PYRO_R:     JSR     R_ENCO
            LDX     #7000
PYRO_CONT:
            JSR     PLS_CNT
            JSR     L_BRAKE
            MOVB    #$03,PWEN
            MOVB    #$03,PORTPP
            MOVB    #$B0,PWDTY0
            MOVB    #$B0,PWDTY1

WAIT2CND:
            CLR     WHO_SENT
            JSR     SRCH_PAS
     ;       JSR     WC4WAIT
            BRA     PYRO_CHK
           ; ADD ANTI FURNITURE SUBROUTINE HERE
            BRA     WAIT2CND

WC4WAIT:                ;WALL CHECK 4 WAIT2CND
             JSR      READ_ATD
             LDAA     ADR0H
             CMPA     #$4A
             BPL      HALT
             LDAA     ADR1H
             CMPA     #$4A
             BPL      HALT
             RTS
HALT:        JSR      L_BRAKE
             MOVB     #$03,PWEN
             MOVB     #$53,PORTPP
             MOVB     #$C0,PWDTY0
             MOVB     #$C0,PWDTY1
             JSR      L_ENCO
             LDX      #3000
             JSR      PLS_CNT
             JSR      L_BRAKE
             MOVB     #$03,PWEN
             MOVB     #$03,PORTPP
             MOVB     #$90,PWDTY0
             MOVB     #$90,PWDTY1
             INS
             INS
             JMP      END_INT_F

SRCH_PAS:
    ;        PSHX
            JSR     PAS_LAV
            TST     PAS_FLAG
            LBNE    TURN_OFF
            JSR     PAS_LAV2
            TST     PAS_FLAG
            LBNE    TURN_OFF
    ;        PULX
            RTS

;=========================================
;                 EXTINGUISHES
;=========================================
PRE_AKOV:
               JSR      CHK_ROOM
CNT_PRE_A:
               LDX      #AKOV_INT
```

```
                STX     INT_OVF
                MOVB    #$03,PWEN
                JSR     L_BRAKE
                MOVB    #$03,PWEN
                JSR     MKM_LR
                JSR     Q_ENGAGE
AKOV:
                JSR     CHECKWALL   ; CHECK FOR TURNS
                JSR     CHK_SIDE    ; Side Cases for us for akiva
                TST     DSIGN
                BNE     AKOV_R
AKOV_L:
                MOVB    #$00,AKOV_SIDE
                JSR     US1
                LDX     US_READ
                CPX     #$120
                LBMI    KAROV
                CPX     #$200
                LBPL    RACHOK
                JSR     US0
                LDX     US_READ
                JSR     US1
                LDY     US_READ
                JSR     DECXYD
                JSR     ABSD
                CPD     #$30
                LBMI    DONT_TKN
                TST     DSIGN
                LBEQ    TAKN_L
                LBRA    TAKN_R

AKOV_R:
                MOVB    #$FF,AKOV_SIDE
                JSR     US3
                LDX     US_READ
                CPX     #$120
                LBMI    RACHOK
                CPX     #$200
                LBPL    KAROV
                JSR     US2
                LDX     US_READ
                JSR     US3
                LDY     US_READ
                JSR     DECXYD
                JSR     ABSD
                CPD     #$30
                LBMI    DONT_TKN
                TST     DSIGN
                LBEQ    TAKN_R
                LBRA    TAKN_L

DONT_TKN:
                MOVB  #$03,PORTPP
                MOVB  #PWD_AKOV,PWDTY0
                MOVB  #PWD_AKOV,PWDTY1
                LBRA  AKOV
TAKN_L:
                MOVB  #$80,PWDTY1
                MOVB  #PWD_AKOV,PWDTY0
                LBRA  AKOV
TAKN_R:
                MOVB  #$80,PWDTY0
```

```
              MOVB   #PWD_AKOV,PWDTY1
              LBRA   AKOV

KAROV:
              MOVB   #$80,PWDTY0
              MOVB   #PWD_AKOV,PWDTY1
              LBRA    AKOV
RACHOK:
              MOVB   #$80,PWDTY1
              MOVB   #PWD_AKOV,PWDTY0
              LBRA   AKOV

CHECKWALL:
           JSR    READ_ATD
           LDAA   ADR0H
           CMPA   #WALL_AKOV
           BPL    SEE_WALL
           LDAA   ADR1H
           CMPA   #WALL_AKOV
           BPL    SEE_WALL
           RTS

SEE_WALL:
            JSR    L_BRAKE
            JSR    CHK_SIDE
            TST    DSIGN        ; CHANGE LATER (right and then
left...)
            BNE    T_LFT_RM

T_RGT_RM:
           MOVB   #$03,PWEN
           MOVB   #$13,PORTPP
           MOVB   #$A0,PWDTY0
           MOVB   #$A0,PWDTY1
           JSR    L_ENCO
           LDX    #6000
           BRA    END_TRNR

T_LFT_RM:
           MOVB   #$03,PWEN
           MOVB   #$43,PORTPP
           MOVB   #$A0,PWDTY1   ; #PWDTRNS,PWDTY0
           MOVB   #$A0,PWDTY0
           JSR    R_ENCO
           LDX    #5000

END_TRNR:         ;ENCODETS COUNTER WITH WHITE LINE BUILDIN
           MOVB    TCTL4,TCTLPLCE
           MOVB    ENC_PULS,TCTL4
           MOVB    ENC_FLAG,TFLG1
C_P_CNTR:  LDAA    TFLG1
           ANDA    ENC_FLAG
           BEQ     C_P_CNTR
           MOVB    ENC_FLAG,TFLG1
           DEX
           LBNE    C_P_CNTR
           LDAA    PORTPP     ; Brake
           EORA    #$A0       ; Brake
           STAA    PORTPP     ; Brake
           LDX     #$9FFF     ; Brake
           JSR     LOOP       ; Brake
           MOVB    #$03,PORTPP
```

```
            MOVB    #PWD_AKOV,PWDTY0
            MOVB    #PWD_AKOV,PWDTY1
            MOVB    TCTLPLCE,TCTL4
MKM_LR:     TST     AKOV_SIDE
            BNE     TKN_MKM_R
TKN_MKM_L:  JSR     MAKOM_L          ; Check for left or right tikun
makom
            BRA     END_TRNR2
TKN_MKM_R:  JSR     MAKOM_R
END_TRNR2:  MOVB    #$03,PWEN
            RTS

CHK_SIDE:    JSR    US2
             LDY    US_READ
             JSR    US0
             LDX    US_READ
             JSR    DECXYD
             JSR    ABSD
             RTS


CHK_WLL_BCK:                          ;CECHK WALL 4 GOING BACK
               JSR      US2
               CPX      #$170
               BLO      CLR_WLL
               MOVB     #$D0,PWDTY1
               RTS
CLR_WLL:       MOVB     #$60,PWDTY1
               RTS


CHK_ROOM:      TST      R0OM
               BEQ      ROOM1
               LDAA     R0OM
               CMPA     #$1
               LBEQ     ROOM2
               CMPA     #$2
               LBEQ     ROOM3
               RTS

ROOM1:         MOVB     #$03,PWEN
               JSR      L_BRAKE
               MOVB     #$03,PWEN
               JSR      MAKOM_R
               MOVB     #$03,PWEN
               MOVB     #$53,PORTPP
               MOVB     #$D0,PWDTY0
               MOVB     #$D0,PWDTY1
DOOP:          JSR      CHK_WLL_BCK
               JSR      US1
               LDX      US_READ
               CPX      #$400
               BPL      DOOP
DOOP2:         JSR      CHK_WLL_BCK
               JSR      US0
               LDX      US_READ
               CPX      #$400
               BPL      DOOP2

               JSR       L_BRAKE
               MOVB     #$03,PWEN
               JSR      MAKOM_R
```

```
                JSR     Q_ENGAGE

DOOP2B:         JSR     US1
                LDX     US_READ
                CPX     #$400
                BMI     DOOP2B
                LDX     #1450
                JSR     L_ENCO
                JSR     PLS_CNT
                JSR     L_BRAKE
                MOVB    #$03,PWEN
                JSR     MAKOM_R
                MOVB    #$03,PWEN
                MOVW    #9420,E_TURN
                JSR     TURN_LEFT_B
                JSR     L_BRAKE                   ;  Do Turn  ;M
                JSR     MOVBYTES
D00P3:
                JSR    PAS_LAV
                TST    PAS_FLAG
                BEQ    D00P3
                LDX    #$200
                JSR    PLS_CNT
                RTS

ROOM3:
             MOVB   #$03,PWEN
             MOVB   #$13,PORTPP
             MOVB   #$B8,PWDTY1
             MOVB   #$B8,PWDTY0
             LDX    #4000               ;  Turn 180 Degrees
             JSR    L_ENCO
             JSR    PLS_CNT
             JSR    BRK_TRNB
             JSR    L_BRAKE
             MOVB   #$03,PWEN
             MOVB   #$03,PORTPP
             MOVB   #$B8,PWDTY1
             MOVB   #$B8,PWDTY0
ROOM3_1:     JSR    PAS_LAV             ; Check For Pas Lavan
             TST    PAS_FLAG
             BEQ    ROOM3_1
             LDX    #$300
             JSR    PLS_CNT
             RTS

ROOM2:
             MOVB   #$03,PWEN
             MOVB   #$43,PORTPP
             MOVB   #$A0,PWDTY1
             MOVB   #$A0,PWDTY0
             LDX    #4000               ;  Turn 90 Degrees
             JSR    L_ENCO
             JSR    PLS_CNT
             JSR    BRK_TRNB
             JSR    L_BRAKE
             MOVB   #$03,PWEN
             RTS
```

<u>Helpfunc.asm</u>

```
D_INTO_A:
          LSLD          ;    - |
          LSLD          ;      |=\  PUT 16 BIT D INTO
          LSLD          ;      |=/      8 BIT A
          LSLD          ;    - |
          RTS


DECXYD: STY  TEMP16B1
        STX  TEMP16B2
        LDD  TEMP16B2
        SUBD TEMP16B1
        RTS

ABSA:
         TSTA
         BPL  ENDABSA
         NEGA
ENDABSA: RTS

ABSD:
         MOVB #$FF,DSIGN
         CPD  #$0
         BPL  ENDABSD
         MOVB #$00,DSIGN
         STD  TEMP16B1
         LDD  #$0000
         SUBD TEMP16B1
ENDABSD: RTS

LOOP:   DEX
        BNE LOOP
        RTS

DLOOP:  STX TEMP16B1
ILOOP:  LDX TEMP16B1
        JSR LOOP
        DEY
        BNE ILOOP
        RTS

ABS_A:
          MOVB  #$FF,WR_2_TKN
          TSTA
          BPL   END_ABSA
          NEGA
          CLR   WR_2_TKN
END_ABSA: RTS

PRIMA:
          MOVB  #$03,PORTPP
          MOVB  #$13,PORTPP
          MOVB  #$B0,PWDTY1
          MOVB  #$B0,PWDTY0
          SWI
```

<u>Inits.asm</u>

```
;**********************
; Program starts here *
;**********************

;**************
;*    RAM     *
;**************
        ORG   RAMSTRT

INBUFF    RMB   IBUFSIZ  ;  input buffer, defined but not used
ENDBUFF   EQU   *
COUNT     RMB   1      ;# characters read, also unused

;***********************************************************

;       ORG     PRGSTRT
        ORG     MONSTRT
;       LDS     #STACK   ; initialize the stack pointer (don't do this
if under monitor)
START:
        JSR     COP_RESET       ; Disable watch-dog timer
        JSR     AD_INIT
        JSR     TIMER_INI
        JSR     PWM_INI
        JSR     PDLC_INI
        JSR     IC_INIT
        JSR     INT_INIT
        JSR     PROP_OFF
        JSR     UV_INIT
        JSR     PROGRAM
        SWI
        RTS
;**********************
; End of main loop
;***********************

;
;    Start Of Initializations
;


COP_RESET:
         MOVB    #$00,COPCTL    ; Turn OFF Cop timer.
         RTS

PDLC_INI:
          MOVB #$DF,DDRDLC
          RTS

IC_INIT: MOVB #$00,TIOS
         MOVB #$80,TSCR
         MOVB #$05,TCTL4
;         LDAA TCTL4 ; Pas Lavan Interrupt Init
;         ORAA #$10 ; Pas Lavan Interrupt Init
;         STAA TCTL4 ; Pas Lavan Interrupt Init
         MOVB #$FF,TFLG1
         RTS

PWM_INI:
```

```
        MOVB #$FF,DDRP
        MOVB #$00,PORTPP
        MOVB #$08,PWCLK         ;20 KHz period
        MOVB #$0F,PWPOL
        MOVB #$FF,PWPER0       ; Period - 20 KHz and 5 KHz.
        MOVB #PWDREG,PWDTY0       ; Duty - for 20 KHz and 5 KHz.
        MOVB #$FF,PWPER1       ; Period - Period - 20 KHz and 5 KHz.
        MOVB #PWDREG,PWDTY1       ; Duty - Half 20 KHz and 5 KHz.
        MOVB #$00,PWEN
        RTS

AD_INID:   LDAA #$C8
AD_DELAY:  DECA
           BNE AD_DELAY
           RTS

AD_INIT:   MOVB #$80,ATDCTL2
           JSR AD_INID
           MOVB #$00,ATDCTL3
           MOVB #$01,ATDCTL4
           RTS

TIMER_INI:  MOVB #$80,TSCR
            RTS

INT_INIT:
            MOVB #$00,TMSK1
;            MOVB #$0C,TMSK1
            MOVB #$80,TMSK2
;            LDX  #R_ENC
;            STX  INT_CH0
;            LDX  #L_ENC
;            STX  INT_CH1
            LDX  #UV_IS_GAY
            STX  INT_CH2
            LDX  #UV_IS_GAY
            STX  INT_CH3
            LDX  #T_OVF
            STX  INT_OVF
;            CLR  R_MONE
;            CLR  R_MONE2
;            CLR  L_MONE
;            CLR  L_MONE2
;            CLR  OVF_MONE
;            CLR  ERR_LSTL
;            CLR  ERR_LSTR
;            CLR  ERR_SUML
;            CLR  ERR_SUMR
            MOVB #$0C,TFLG1
            MOVB #$80,TFLG2
            SEI
            RTS

PROP_OFF:
            LDAA  PDLC
            ANDA  #$BF
            STAA  PDLC
            RTS

UV_INIT:
          LDAA  TCTL4
          ORAA  #$F0
```

```
            STAA   TCTL4
            MOVB   #$0C,TFLG1
            CLR    UV_FLAG
            RTS

;
;       End Of Initializations
;
```

Ir_ovf.asm

```
T_OVF:
        PSHD
        PSHX
        PSHY
        PSHC

SIUM:
CHK_WALL:                          ; Check for front wall
             CLRA
             CLRB
             JSR     READ_ATD     ;MEMOTZA 4 READS FOR RIGHT FRONT
SENSOR
             LDAB    ADR0H
             LDAA    ADR1H
             JSR     READ_ATD
             ADDB    ADR0H
             ADDA    ADR1H
             LSRA
             LSRB
             CMPA    WALL
             BMI     SIOM
             CMPB    WALL
             BMI     SIOM

NEARWALL:
        MOVB    #$FF,PLZ_TURN         ; Make the flag and jump to end
SIOM:
        MOVB   #$80,TFLG2
        PULC
        PULY
        PULX
        PULD
        RTI

UV_IS_GAY:
        MOVB   #$FF,UV_FLAG
        MOVB   #$0C,TFLG1
        RTI

AKOV_INT:
          PSHX
          PSHD
          PSHC
          MOVB   #$DD,WHO_SENT
          CLR    PAS_FLAG
          LDAA   PDLC
          ORAA   #$DF
          INCA
          LBNE   TURN_OFF
          LDAA   PDLC
          ORAA   #$EF
          INCA
          LBNE   TURN_OFF
          JMP    PYRO_CHK

END_INT_F:
           PULC
           PULD
           PULX
```

```
            MOVB   #$80,TFLG2
            RTI
```

Motors.asm

```
Q_ENGAGE:
            MOVB   #$03,PWEN
            MOVB   #$03,PORTPP
            MOVB   #PWDREG,PWDTY1
            MOVB   #PWDREG,PWDTY0
            RTS
ENGAGE:
            JSR    Q_ENGAGE
            LDX    #$FFFF
            LDY    #$7
            JSR    DLOOP
            RTS

ENG_TRY:
            MOVB   #$03,PWEN
PRENG_F:    MOVB   #$81,PORTPP
            MOVB   #PWDREG,PWDTY1
            MOVB   #PWDREG,PWDTY0
            LDX    #$A2FF
            JSR    LOOP
            JSR    ENGAGE
            RTS

W_BRAKE:
            MOVB   #$60,PWDTY1
            MOVB   #$60,PWDTY0
            LDAA   PORTPP
            EORA   #$50
            STAA   PORTPP
            LDX    #$8FFF
            LDY    #$2
            JSR    DLOOP
            MOVB   #$00,PORTPP
            MOVB   #$00,PWEN
            RTS
M_BRAKE:
            MOVB   #$90,PWDTY1
            MOVB   #$90,PWDTY0
            LDAA   PORTPP
            EORA   #$50
            STAA   PORTPP
            LDX    #$FFFF
            JSR    LOOP
            MOVB   #$00,PORTPP
            MOVB   #$00,PWEN
            RTS

VW_BRAKE:
            MOVB   #$00,PWDTY1
            MOVB   #$00,PWDTY0
            MOVB   #$00,PORTPP
            MOVB   #$00,PWEN
            LDX    #$FFFF
            JSR    LOOP
            RTS
BRAKE:
            MOVB #$A0,PORTPP
            LDX  #$4
            LDY  #$FFF
```

```
            JSR   DLOOP     ; Delay for brk
            MOVB #$00,PWEN
            RTS
L_BRAKE:
            MOVB #$A0,PORTPP
            LDX   #$A
            LDY   #$FFF
            JSR   DLOOP        ; Delay for brk
            MOVB #$00,PWEN
            RTS


L_ENCO:
            PSHA
            LDAA   #ENC_PUL1
            STAA   ENC_PULS
            LDAA   #ENC_FLG1
            STAA   ENC_FLAG
            PULA
            RTS

R_ENCO:
            PSHA
            LDAA   #ENC_PUL0
            STAA   ENC_PULS
            LDAA   #ENC_FLG0
            STAA   ENC_FLAG
            PULA
            RTS
TRN_R_F:
            MOVB   #$13,PORTPP
            MOVB   #PWDTRNS,PWDTY1
            MOVB   #PWDTRNW,PWDTY0
            JSR    L_ENCO
            LDX    E_TURN          ; Load encoders for turn
            BRA    TRN_BEGN
TRN_L_F:
            MOVB   #$43,PORTPP
            MOVB   #PWDTRNS,PWDTY0
            MOVB   #PWDTRNW,PWDTY1
            JSR    R_ENCO          ; Inits for Right Encoder
            LDX    E_TURN          ; Load encoders for turn
            BRA    TRN_BEGN
TRN_L_B:
            MOVB   #$43,PORTPP
            MOVB   #PWDTRNS,PWDTY0
            MOVB   #PWDTRNW,PWDTY1
            JSR    L_ENCO
            LDX    E_TURN          ; Load encoders for turn
            BRA    TRN_BEGN
TRN_R_B:
            MOVB   #$13,PORTPP
            MOVB   #PWDTRNS,PWDTY1
            MOVB   #PWDTRNW,PWDTY0
            JSR    R_ENCO
            LDX    E_TURN          ; Load encoders for turn
            BRA    TRN_BEGN

TRN_BEGN:
            JSR    PLS_CNT

END_TURN:
```

```
            JSR     BRK_TURN
            RTS

TURN_180:
            MOVB   #$43,PORTPP
            MOVB   #$05,PWDTY0
            MOVB   #PWDTRNW,PWDTY1
            JSR    L_ENCO
            LDX    #ENC180T
            JSR    PLS_CNT
            BRA    END_TURN

PLS_CNT:
            MOVB   TCTL4,TCTLPLCE
            MOVB   ENC_PULS,TCTL4
            MOVB   ENC_FLAG,TFLG1
C_P_CNT:    LDAA   TFLG1
            ANDA   ENC_FLAG
            BEQ    C_P_CNT
            MOVB   ENC_FLAG,TFLG1
            DEX
            BNE    C_P_CNT
            MOVB   TCTLPLCE,TCTL4
            RTS

TRN_R_FS:
            MOVB   #$13,PORTPP
            MOVB   #PWDTRNW,PWDTY0
            MOVB   #PWDTRNW,PWDTY1
            JSR    L_ENCO
            LDX    #ENCTURNS
            JSR    PLS_CNT
            MOVB   #$13,PORTPP
            MOVB   #90,PWDTY0
            MOVB   #90,PWDTY1
            LDX    #$AEFF
            LDY    #$2
            JSR    DLOOP
            JSR    MOVBYTES
            RTS

;U_TURN:
;          MOVB     #$43,PORTPP
;          JSR      L_ENCO
;          LDX      #$2200
;          JSR      PLS_CNT
;          MOVB     #$E0,PWDTY0
;          MOVB     #$30,PWDTY1
;          MOVB     #$13,PORTPP
;          LDX      #$AFFF
;          LDY      #$2
;          JSR      DLOOP
;          MOVB     #$03,PORTPP
;          MOVB     #$E0,PWDTY1
;          MOVB     #$E0,PWDTY0
;          RTS
;O_TURN:
;          MOVB     #$43,PORTPP
;          JSR      L_ENCO
;          LDX      #$4000
;          JSR      PLS_CNT
;          RTS
```

```
X_L:        MOVB  #$03,PWEN
            MOVB  #$03,PORTPP
            MOVB  #$68,PWDTY1
            MOVB  #$68,PWDTY0
ACC_2:      INC   PWDTY0
            INC   PWDTY1
            LDX   #$1200
            JSR   LOOP
            LDAA  PWDTY1
            CMPA  #$F0
            BLO   ACC_2
            LDX   #$FFFF
            LDY   #$4
            JSR   DLOOP
            MOVB  #$E0,PWDTY1
            MOVB  #$E0,PWDTY0
            RTS


TURN_RIGHT_B:
           PSHC
           PSHX
           SEI
           MOVB   #$03,PWEN
           MOVB   #$23,PORTPP
           MOVB   #PWDREG,PWDTY1
           MOVB   #$00,PWDTY0
           JSR    L_ENCO
           LDX    E_TURN
           JSR    PLS_CNT
           JSR    BRK_TRNB
           CLR    PLZ_TURN
           PULX
           PULC
           RTS

TURN_LEFT_B:
           PSHC
           PSHX
           SEI
           MOVB   #$03,PWEN
           MOVB   #$83,PORTPP
           MOVB   #PWDREG,PWDTY0
           MOVB   #$00,PWDTY1
           JSR    R_ENCO
           LDX    E_TURN
           JSR    PLS_CNT
           JSR    BRK_TRNB
           CLR    PLZ_TURN
           PULX
           PULC
           RTS

TURN_LEFT:
           PSHC
           PSHX
           SEI
           JSR   R_ENCO
           MOVB  #$43,PORTPP
           MOVB  #$F0,PWDTY0
           MOVB  TURNWEAK,PWDTY1
```

```
            LDX    E_TURN          ; Load encoders for turn
            JSR    PLS_CNT
            JSR    BRK_TURN
            CLR    PLZ_TURN
            PULX
            PULC
            RTS

TURN_RIGHT:
            PSHC
            PSHX
            SEI
            JSR    L_ENCO
            MOVB   #$13,PORTPP
            MOVB   #$F0,PWDTY1
            MOVB   TURNWEAK,PWDTY0
            LDX    E_TURN          ; Load encoders for turn
            JSR    PLS_CNT
            JSR    BRK_TURN
            CLR    PLZ_TURN
            PULX
            PULC
            RTS

BRK_TURN:
            PSHA
            LDAA   PWDTY0
            LDAB   PWDTY1
            STAB   PWDTY0
            STAA   PWDTY1
            LDAA   PORTPP     ; Brake
            EORA   #$50       ; Brake
            STAA   PORTPP     ; Brake
            LDY    #$2
            LDX    #$AFFF     ; Brake  WAS  FFFF
            JSR    DLOOP      ; Brake
            JSR    MOVBYTES
            PULA
            RTS

MOVBYTES:
            MOVB   #$03,PWEN
            MOVB   #$03,PORTPP
            MOVB   #PWDREG,PWDTY1
            MOVB   #PWDREG,PWDTY0
            RTS

BRK_TRNB:
            PSHA
            LDAA   PWDTY0
            LDAB   PWDTY1
            STAB   PWDTY0
            STAA   PWDTY1
            LDAA   PORTPP     ; Brake
            EORA   #$A0       ; Brake
            STAA   PORTPP     ; Brake
            LDY    #$2
            LDX    #$AFFF     ; Brake  WAS  FFFF
            JSR    DLOOP      ; Brake
            JSR    MOVBYTES
            PULA
            RTS
```

Sensors.asm

```
US0:     MOVB #$40,USUPW
         MOVB #$80,USFLG
         MOVB #$80,USDNW
         MOVB #$08,US_INIT
         JMP  USENG

US1:     MOVB #$10,USUPW
         MOVB #$40,USFLG
         MOVB #$20,USDNW
         MOVB #$04,US_INIT
         JMP  USENG

US2:     MOVB #$04,USUPW
         MOVB #$20,USFLG
         MOVB #$08,USDNW
         MOVB #$08,US_INIT
         JMP  USENG

US3:     MOVB #$01,USUPW
         MOVB #$10,USFLG
         MOVB #$02,USDNW
         MOVB #$04,US_INIT
         JMP  USENG


USENG:
          PSHC
          PSHD
          PSHX
          SEI
PREREAD:  LDX  #$0400   ;10 mili seconds delay
PREDAL:   LDAA #$20
PRERD:    DECA
          BNE  PRERD
          DEX
          BNE  PREDAL

          LDAA PDLC
          ORAA US_INIT
          STAA PDLC
          LDAA #$20
USSHEV:   DECA
          BNE  USSHEV

          LDAA PDLC
          LDAB US_INIT
          COMB
          STAB US_INIT
          ANDA US_INIT
          STAA PDLC
          LDX  #$0000


USREAD:
          LDAA  TCTL3
          ORAA  USUPW
          STAA  TCTL3
          MOVB  USFLG,TFLG1
SHUV:     LDAA TFLG1
```

```
         ANDA USFLG
         BEQ  SHUV
         LDAA TCTL3
         EORA USUPW
         ORAA USDNW
         STAA TCTL3
         MOVB USFLG,TFLG1
         LDX  #$0000

MONE:   INX
        LDAA TFLG1
        ANDA USFLG
        BEQ MONE
        STX US_READ
        PULX
        PULD
        PULC
        RTS

;========================================
;                   A2D
;========================================

READ_ATD:  MOVB #$50,ATDCTL5
AD_PROC:   BRCLR ATDSTAT $80 AD_PROC;
           RTS

IR_AVG_F:
          PSHD
          JSR       READ_ATD
          LDAA      ADR0H
          LDAB      ADR1H
          JSR       READ_ATD
          ADDA      ADR0H
          ADDB      ADR1H
          ABA
          LSRA
          LSRA
          STAA      IR_AVG
          PULD
          RTS

IR_AVG_B:
          PSHD
          JSR       READ_ATD
          LDAA      ADR4H
          LDAB      ADR5H
          JSR       READ_ATD
          ADDA      ADR4H
          ADDB      ADR5H
          ABA
          LSRA
          LSRA
          STAA      IR_AVG
          PULD
          RTS

;========================================
;                PAS LAVAN
;========================================

PAS_LAV:
```

```
          PSHD
          CLR   PAS_FLAG
          LDAA  PDLC
          ORAA  #$DF
          INCA
          BNE   PAS_1
PAS_0:    MOVB  #$00,PAS_FLAG
          PULD
          RTS
PAS_1:    MOVB  #$FF,PAS_FLAG
          PULD
          RTS

PAS_LAV2:
          PSHD
          CLR   PAS_FLAG
          LDAA  PDLC
          ORAA  #$EF
          INCA
          BNE   PAS2_1
PAS2_0:   MOVB  #$00,PAS_FLAG
          PULD
          RTS
PAS2_1:   MOVB  #$FF,PAS_FLAG
          PULD
          RTS

;========================================
;              ULTRA VIOLET
;========================================

UV_ON:
          MOVB  #$0C,TFLG1
          LDAA  TMSK1
          ORAA  #$0C
          STAA  TMSK1
          RTS

UV_ON_L:
          MOVB  #$08,TFLG1
          LDAA  TMSK1
          ORAA  #$08
          STAA  TMSK1
          RTS

UV_OFF:
          MOVB  #$0C,TFLG1
          LDAA  TMSK1
          ANDA  #$F3
          STAA  TMSK1
          RTS

CHECK_UV:
          TST   UV_FLAG
          LBNE   PRE_AKOV
          RTS
;CHK_UV:
;          BRCLR TFLG1 $04 CHK_UV2
;          BRA   YES_UV
;CHK_UV2:
;          BRCLR TFLG1 $08 NO_UV
;YES_UV:   MOVB  #$FF,UV_FLAG
```

```
;           MOVB  #$0C,TFLG1
;           RTS
;
;NO_UV:     MOVB  #$00,UV_FLAG
;           RTS
```

Print.asm

```
SPILL:   STAA     $0850

         LDAA     $0850
         RORA
         RORA
         RORA
         RORA
         ANDA     #$0F
         CMPA     #$09
         BPL      ADD1
MORE1    ADDA     #$30
         STAA     $0851
         LDAA     $0850
         ANDA     #$0F
         CMPA     #$09
         BPL      ADD2
MORE2    ADDA     #$30
         STAA     $0852
         LDAA     #EOT
         STAA     $0853

         LDX      #$0851
         JSR     OUTSTRG ; send it out serial port
         JSR     OUTCRLF ; output carriage-return

WAITL:   PSHX
         PSHB
         LDAB  #$06
SB0:     LDX   #$FFFF
SB1:     DBNE X,SB1
         DBNE B,SB0
         PULB
         PULX

         RTS

ADD1     ADDA    #$07
         BRA     MORE1

ADD2     ADDA    #$07
         BRA     MORE2


ENDLESS  BRA ENDLESS
         RTS



ONSCI:
        ldaa    #$34    ; get baud rate constant
        staa    SC0BDL  ; store low byte
        clr     SC0BDH  ; clear high byte
        ldaa    #$00    ; configure SCI0 control registers
        staa    SC0CR1
        ldaa    #$0C    ; enable transmit and receive
        staa    SC0CR2
        RTS

; Output string of ASCII bytes starting at x until end of text ($04).
```

```
OUTSTRG:
        JSR  OUTCRLF    ; output carriage-return
OUTSTRG0:
        PSHA           ; save a
OUTSTRG1:
        LDAA 0,X        ; read char into a
        CMPA #EOT      ; is this end of text?
        BEQ  OUTSTRG3   ; jump if yes
        JSR  OUTPUT     ; output character
        INX            ; incriment pointer
        BRA  OUTSTRG1  ; loop
OUTSTRG3:
        PULA            ; restore a
        RTS

; Output a Carriage return and a line feed.  Returns a = cr.
OUTCRLF:
        LDAA #$0A       ; get LF
        JSR  OUTPUT     ; send it
        LDAA #$0D       ; get CR
        JSR  OUTPUT     ; send it
        LDAA #$00
        JSR  OUTPUT     ; output padding
        LDAA #$0D
        RTS

; Output A to SCI0
OUTPUT:
OUTSCI2:
        LDAB    SC0SR1  ; read status
        BITB    #$80    ; test Transmit Data Register Empty bit
        BEQ     OUTSCI2 ; loop if TDRE=1
        ANDA    #$7F    ; mask parity
        STAA    SC0DRL  ; send character
        RTS

;**************
; TEXT TABLES *
;**************
MSGIL   FCC    'IN LINE'
        FCB    EOT
MSGTL   FCC    'METAKEN LEFT'
        FCB    EOT
MSGTR   FCC    'METAKEN RIGHT'
        FCB    EOT
MSGFX   FCC    'GOING STRT'
        FCB    EOT
MSGCL   FCC    'CHECK LEFT'
        FCB    EOT
```

ביבליוגרפיה

<u>ספרות כתובה:</u>

שם המחבר	שם הספר	עמודים	הוצאה
ד"ר יורם אשל	מכניקה לתיכון ולאוניברסיטה	516-517	אשל
מיכל צלטנר	הכימיה: אתגר – כימיה של תאים חשמליים	26-33	המחלקה להוראת המדעים מכון ויצמן
Ingo Cyliax	מאמר: Power Systems in Autonomous Robots	24-34	Circuit Cellar Ink #92
Motorola	M68HC12B Family – Advance Information	1-484	Motorola
Motorola	M68HC12B Family – Reference Manual	1-540	Motorola

<u>אתרי אינטרנט:</u>

כתובת האתר	מחבר
http://www.abrobotics.com	Alex Brown
http://www.robotica.co.il	אתר הרובוטיקה הישראלי
http://www.seattlerobotics.org/encoder	Seattle Robotics