



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)

חיישני אינפרא-אדום:

החיישן GPD12 נותן אות אנלוגי מ-0-5v בהתאם למרחק שהוא "רואה". האות האנלוגי עובר עיבוד על ידי רכיב ה- Analog to Digital והוא מגיע לאחד הכניסות 0-7. 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.

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

© כל הזכויות שמורות לאסף פוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



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 a/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 תבצע מערוץ אנלוגי מספר X לכתובת בזיכרון (0 ל-0, 1 ל-1 וכדומה). לאחר מכן אנו מחכים עד שההמרה תושלם. הסימן לכך שההמרה הסתיימה היא שהביט השמאלי באוגר ATDSTAT מוגדר כ-1 ולכן אנו משתמשים בפקודה Branch If Clear ששמה "מסכה" על כל הביטים אשר מוגדרים בפקודה כ-0 ובודקת אם הביטים שמוגדרים כ-1 מוגדרים כ-0. במידה וכן הפקודה תקפוץ לתווית המוגדרת כ-AD_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 ב-8 אנו תמיד מעלים רק את ה High 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 $\pm 2/3$ מהקריאה האמיתית. לאחר שהעברנו את המידע לאקומולאטור ניתן לבצע פעולות חישוב כמו השוואה לנתון מסוים ותגובה בהתאם (במקרה זה, בדיקת מרחק מהקיר הקדמי).

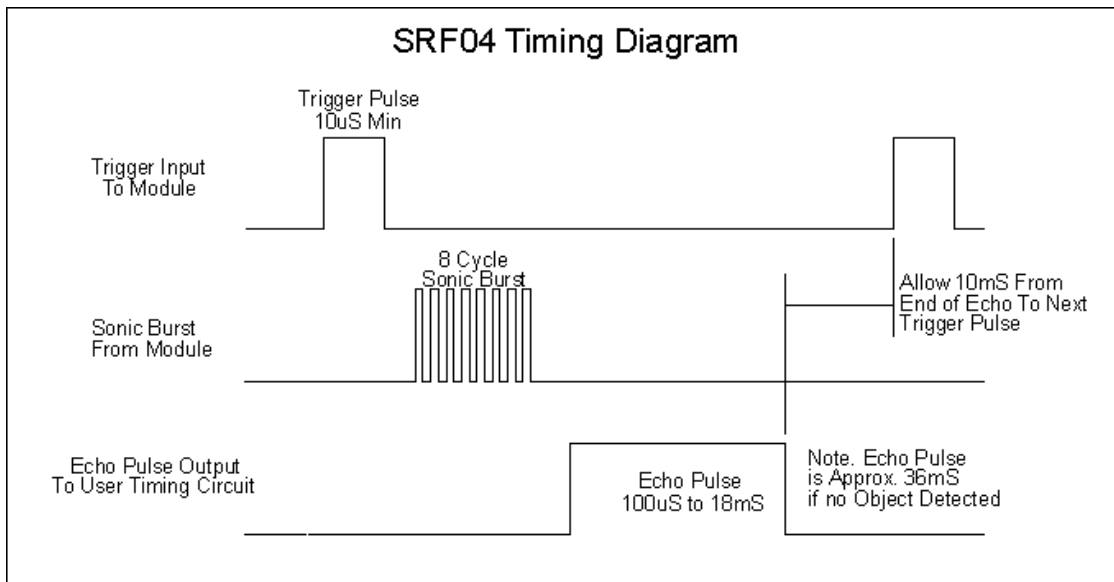
בעלי אתר הרובוטיקה הישראלי לא ישאו באחריות כלשהי לכל נזק, כספי או אחר שייגרם במישורין או בעקיפין משימוש במידע המצוי באתר זה

© כל הזכויות שמורות לאסף מוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



חיישנים אולטרא סוניים

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 Trigger גל שאורכו הוא 10 מיקרו שניות.
 - לחכות עד אשר תתקבל קריאה ברגל Echo 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 PortDLC כפורט יציאה ברגליים בהן אנו משתמשים:

```
PDLC_INI :  
          MOV B # $CF, DDRDLC  
          RTS
```

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

© כל הזכויות שמורות לאסף פוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



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 Output.

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 Triggern 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:	LDA	#\$20	
PRERD:	DECA		
	BNE	PRERD	
	DEX		
	BNE	PREDAL	
	LDA	PDLC	; Set PDLC leg = +5v
	ORA	US_INIT	
	STAA	PDLC	

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

© כל הזכויות שמורות לאסף פוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



```
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 10 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 Trigger החיישן אמור לתת קריאה ברגל Echo וכן 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 שמראה 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

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

© כל הזכויות שמורות לאסף מוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



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
TFLG1*\$80	1	0	0	0	0	0	0	0

ברגע שהביט עלה, התוצאה משתנה מהערך 0 ואז יודעים שהערך ב TFLG1 השתנה.

הערה: 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.

הערה: למעשה לא היה צורך להגדיר בתוכנה את הגל היורד, היה ניתן פשוט להעלות את ערך הגל
 העולה לאקומולטור מסוים ועליו לבצע את הפעולה ASRA שמזיזה את כל הביטים מקום אחד
 מימין.

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 דוגמה 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
...			
...			

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

© כל הזכויות שמורות לאסף מוניס ולגיא יונה
 אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



חיישני פס לבן

חיישני הפס הלבן נותנים אות דיגיטלי. את החיישנים ניתן לחבר בשתי דרכים:

חיבור PortLT – דורש מקום באותו הפורט שלא היה לנו.

חיבור PortDLC – 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 לא 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 לא. 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 לא 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 BNE – Branch If Not Equal To Zero).

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

© כל הזכויות שמורות לאסף פוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



דוגמא ל-PortDLC כאשר החיפוש 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 חיפוש הפס הלבן השני שמחובר לרגל מספר 4.

דוגמא לתפעול חיפוש 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 בחיפוש האחורי.

חיפושני UvTron

החיפושנים מחזירים אות דיגיטלי ומחוברים ל-PortT. עוד פירוט על חיפושנים אלו בפרק על פסיקות.

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

© כל הזכויות שמורות לאסף פוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



חיישני Pyro

חיישנים אלו מחזירים אות אנלוגי ומחוברים לפורט האנלוגי.
על מנת לקרוא מהחיישנים הללו השתמשנו בממיר ה־Analog 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 ל־\$10 ומעל ל־\$E8. 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.

חיישן המיקרופון

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

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

© כל הזכויות שמורות לאסף פוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר



חישן המקודד (Encoder)

חישן זה מחזיר אות דיגיטלי והוא מצוי בתוך המנוע. חישן 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 ל- L_Enco או R_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.

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

© כל הזכויות שמורות לאסף פוניס ולגיא יונה
אין להעתיק תכנים מאתר זה ללא רשות בכתב ממנהלי האתר