



אוניברסיטת בן-גוריון בנגב  
Ben-Gurion University of the Negev

# פרויקט גמר

קורס: מבנה מחשבים ספרתיים

הנדסת מחשבים - שנה ב

הסבר מפורט בליווי תמונות

21/05/2017

**הקדמה:**

בקורס DCS להנדסת מחשבים, פרויקט הגמר מבוסס על פלטפורמת רכב רובוט אוטונומי **בעל יכולות רבות** אשר חלקן הרלוונטי לפרויקט זה, יפורט בהמשך הקובץ.

רכב רובוט אוטונומי, פועל ללא התערבות חיצונית ומבצע את פעולותיו באופן עצמאי.

בכל שנה הגדרת המשימה הניתנת לסטודנטים בתכנות הרובוט אוטונומי משתנה, מטרתה לפתח יכולות תיאורטיות ומעשיות בהבנת המחשב הספרתי (מיקרו-בקר) בתחום הרובוטיקה האוטונומית.

הגדרת המשימה זהה לכל הקבוצות בקורס (עבודה בזוגות) כך שבסופה מתקיימת תחרות בין הקבוצות לביצוע משימה בתנאים זהים בזמן מינימאלי.

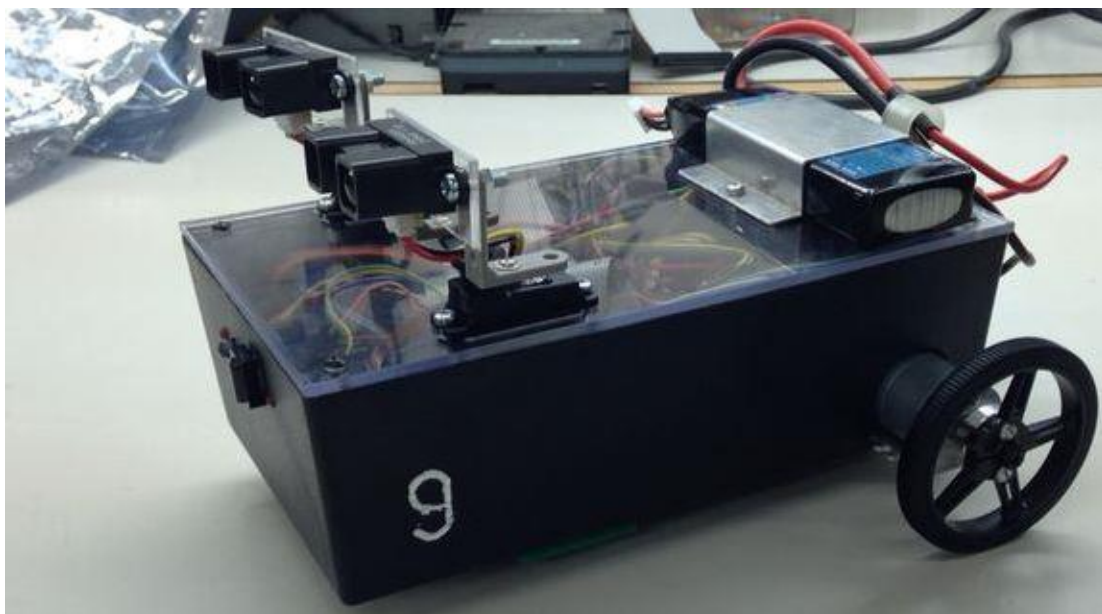
לצורך הגברת המודעות באוניברסיטה לתחום זה, אנו נקיים תערוכה פתוחה לקהל הרחב אשר תיקרא DCS CUP ובזמן תתקיים התחרות של פרויקט הגמר.

בסיום התחרות יוכרזו שלושת המקומות הראשונים אשר יזכו בבנוס בציון (יפורט בהמשך) ולחשיפה (המטרה להזמין אנשים מתעשיית ההייטק מתחום ה- Embedded ולחשוף אותם לסטודנטים).

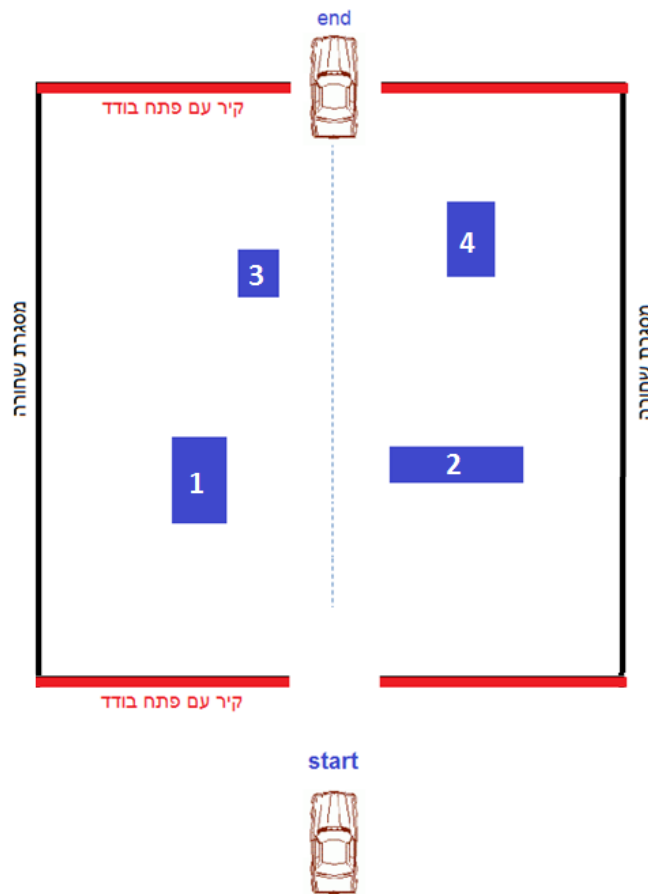
סרטון התערוכה הרשמי יפורסם ב-YouTube.

**פלטפורמת הרכב (חומרה מכניקה וכו') תוכננה ויוצרה במכון האלקטרוני**

**הנמצא במחלקה להנדסת חשמל ומחשבים 123/33**



איור 1 – צילום צדי של רכב רובוטי

**תיאור הפרויקט :****איור 2 - תיאור כללי של מסלול הנסיעה**

1. משימת הרכב לצאת מנקודת ההתחלה (start) לנקודת הסיום (end) בזמן מינימאלי ללא התערבות חיצונית (באופן אוטונומי). שתי הנקודות start, end נמצאות על אותו קו אווירי החוצה לשניים את זירת התחרות (צורת מלבן). מנקודת התחלה עד לנקודת סיום, הרכב נדרש לזהות את מספר המטרות הנמצאים בזירה ולפגוע בכל אחד מהם בתורו. בשלב ראשון, יש צורך לעבור דרך פתח בקיר (רוחב כל פתח הוא כפול מרוחבו של הרכב). לאחר מכן ישנן מטרות פזורות בדרכו לנקודת הסיום בצורת תיבות מלבניות בגדלים שונים. על הרכב לסרוק את הזירה ולהחליט בכל שלב, באיזו מטרה הוא רוצה לפגוע וכך קובע את מסלול הנסיעה. נקודת הסיום היא למעשה פתח בקיר ישר שיש צורך לעבור דרכו. הזירה המלבנית של התחרות מוקפת בפס איזולירבנד שחור המהווה גבולות גזרה של הנסיעה ואסור לחצות אותו (קנס מבחינת זמן יקבע על כל חציית גבול הגזרה).  
2. במהלך הנסיעה הרכב צופה וסורק בעזרת חיישני המרחק (סעיף 4) ובכך צופה בזירה, מעבד נתונים ומקבל החלטות למסלול נסיעת הרכב. נדרש מכם לשלוח ל-PC בזמן אמת (סעיף 5) מה "רואה" הרכב וכך לבנות ב-PC את מבנה מסלול נסיעת הרכב. בדומה למה שנתבקשתם לבצע כעבודת הכנה ללמידת שפת C#, זירת התחרות המלבנית ניתן לייצג בעזרת מערכת צירים (רביע ראשון בלבד) ובשימוש ה-ENCODERS תוכלו לנתח את נקודת המיקום של הרכב (x,y) וממנה, בעזרת חיישני המרחק תוכלו לנתח את מיקום המטרות. סימון בגרף למסלול הנסיעה, קיר, פתח ומכשול יסומנו בצורה הבאה:  
מסלול נסיעה - קו שחור, קיר - קו אדום עבה, פתח - נקודות אדומות, מכשול - כוכביות שחורות.  
3. המידע הנשלח מהרכב ל-PC (בזמן אמת) יש צורך לאגור אותו ב-PC בצורת מטריצה דו ממדית, כך שניתן יהיה להציג את מבנה מסלול נסיעת הרכב כשגרפה.

## הכוונת עבודה על הפרויקט

הכוונת העבודה על הפרויקט חשובה הן מבחינת הגישה לכתיבת קוד נכון ואופטימאלי השולט באופן ישיר על חומרה כך שכל שורת קוד משמעותית ביותר עבור ביצועי המערכת.

בנוסף, חלוקת עבודה בצורה נכונה הגוזרת את המשימה לתתי משימות שנותר לבצע בניהן אינטגרציה (חיבור בין תתי המשימות לצורך ביצוע מטרת הפרויקט).

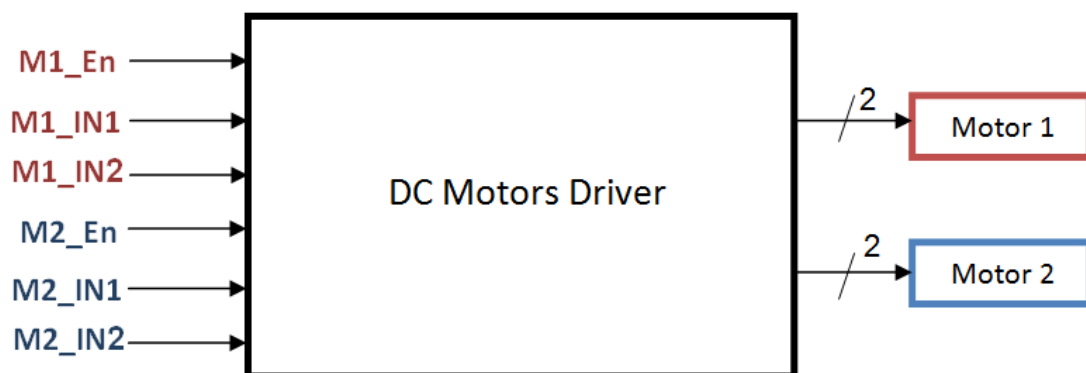
בקובץ זה, אגדיר לכם את השליטה על מודולי החומרה (הרלוונטיים לפרויקט עבור שנה זו) מבחינת קופסאות שחורות שהגישה אליהן תהיה מבחינת כניסה ומוצא (Input and Output).

בתור המלצה, אגדיר לכם את הפונקציה המתאימה למודול אותה תצטרכו לממש מבחינת תת-משימה. המטרה הראשונה תהיה בדיקת המימוש של תתי-המשימות (פונקציות לכל מודול) אותה תכינו לקראת מפגש המעבדה בהן יחולקו הרכבים.

צורת עבודה זו תבטיח יעילות מבחינת למידה נכונה ניצול זמן וצמצום שגיאות.

**הארה:** כיווני הנסיעה בקובץ זה הם ביחס לכיוון נסיעת הרכב (כאילו המשתמש נוהג ברכב).

### 1. הפעלת DC Motors:



Label Name	MCU Signal	Description	FRDM-KL25Z Pin No.
M1_En	PWM	שולט על כניסת EN של Motor1	PTE23 – TPM2_CH1 (J10-7)
M1_IN1	Output	שולט על כיווניות Motor1	PTC5 - Output (J1-9)
M1_IN2	Output	שולט על כיווניות Motor1	PTC6 - Output (J1-11)
M2_En	PWM	שולט על כניסת EN של Motor2	PTE20 – TPM1_CH0 (J10-1)
M2_IN1	Output	שולט על כיווניות Motor2	PTC7 - Output (J1-1)
M2_IN2	Output	שולט על כיווניות Motor2	PTC10 - Output (J1-13)

### ● כיווניות המנוע:

כיווניות Motor1 (כך גם עבור Motor2) נקבעים ע"י הערכים במוצא הבקר ברגליים M1\_IN1 ו-M1\_IN2.

✓ כאשר M1\_IN1=0 , M1\_IN2=1 - כיוון נסיעה קדימה.

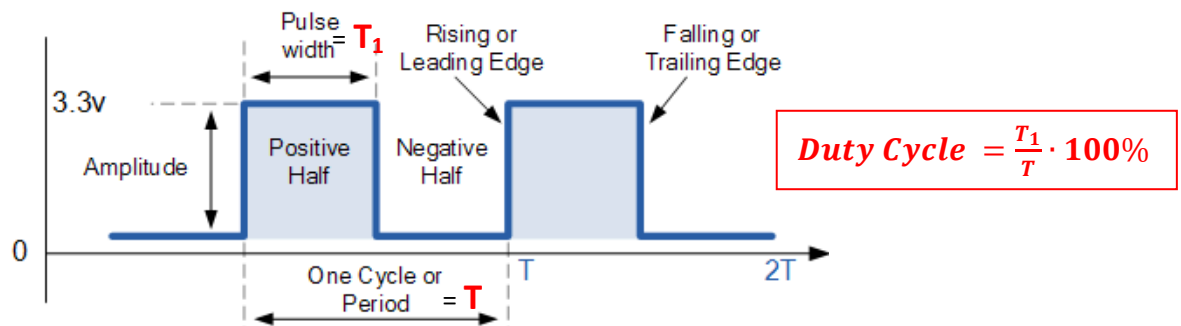
✓ כאשר M1\_IN1=1 , M1\_IN2=0 - כיוון נסיעה אחורה.

✓ כאשר M1\_IN1=1 , M1\_IN2=1 - עצירה.

✓ כאשר M1\_IN1=0 , M1\_IN2=0 - עצירה.

### ● מהירות המנוע:

רגל Enable של המנועים (M2\_En , M1\_En) מאפשרת שליטה על מהירותם ע"י מוצא PWM מהבקר לכל אחד מ-2 רגליים אלו. ערכי ה-Duty Cycle של אות ה-PWM קובע את מהירות (ערך ה-Duty Cycle קובע את המתח DC הממוצע שבין הדקי המנוע) המנוע. Duty Cycle=0% המנוע עוצר, Duty Cycle=100% המנוע בשיא המהירות.



לצורך כך היעזרו בהגדרות הבאות:

**בניסוי מספר 3** למדתם לעבוד עם מנועים מהסוג שנמצאים ברכב.

תדר אות PWM יהיה  $f_{PWM} = 40Hz$  (שווה ל**תדר** פעולה של מנועי SERVOS – ראה סעיף 3).

### ● חיכוך סטטי וחיכוך דינאמי של המנוע:

מעבר ממצב של עצירה למצב נסיעה, יש לעבור את **החיכוך סטטי** של המנוע.

ערך ה-Duty Cycle לצורך נסיעה הכי איטית אפשרית תהא גבוהה יותר כאשר הרכב מתחיל לנוע מעצירה מוחלטת. על כן מומלץ מאוד להגדיר בהתחלה **Duty Cycle=100%** וכאשר מתחילה תזוזת הרכב (זיהוי ע"י מודול ENCODER – סעיף 2) להוריד את ה-Duty Cycle כרצוננו לצורך מהירות איטית נדרשת מינימאלית אשר תבדקו אותה בעצמכם.

### ● שליטה על כיוון נסיעת הרכב:

בעזרת שליטה על מהירות סיבובי המנועים (**לכל מנוע בנפרד**) נוכל לשלוט על כיוון נסיעת הרכב.

✓ נסיעה **ישרה** – באופן תיאורטי 2 המנועים יפעלו במהירות זהה. מאחר ובייצור המנועים לא ניתן להגיע לרמת דיוק זהה. לכן עליכם לראות עבור כל מנוע בנפרד מהו ה-Duty Cycle לצורך נסיעה ישרה.

✓ פנייה **ימינה** – מנוע ימין (Motor1) יהיה איטי ממנוע שמאל (Motor2) – היחס בניהם יחסי לזווית הפנייה ימינה.

✓ פנייה **שמאלה** – מנוע שמאל (Motor2) יהיה איטי ממנוע ימין (Motor1) – היחס בניהם יחסי לזווית הפנייה שמאלה.

● פונקציות מומלצות – כתיבת דרייברים:

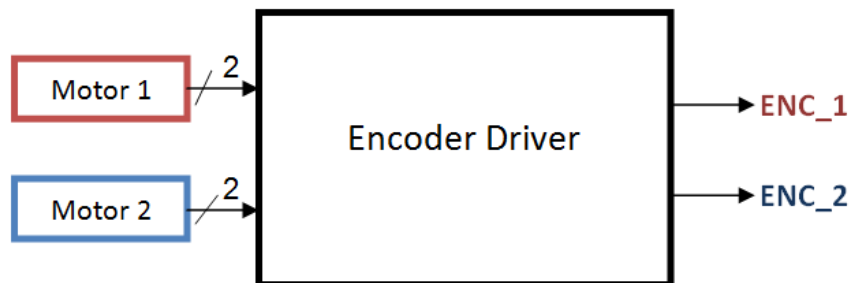
✓ פונקציית קינפוג למנועים - `Void MotorsConfig ()`

✓ פונקציית שליטה על מהירות וכיוון הרכב - `Void MotorsDir_Speed (int Direction, int Speed)`

כאשר Speed היא מהירות הרכב ביחידות של cm לשנייה.

כאשר Direction הוא כיוון נסיעת הרכב במעלות.

2. חיווי סיבובי גלגלים בעזרת ENCODER:



Label Name	MCU Signal	Description	FRDM-KL25Z Pin No.
ENC_1	Input Capture	סופר סיבובי מנוע Motor1	PTC3 – TPM0_CH2 (J1-5)
ENC_2	Input Capture	סופר סיבובי מנוע Motor2	PTC4 – TPM0_CH3 (J1-7)

כל אחת מהרגליים **ENC\_1, ENC\_2** נותנת חיווי למספר סיבובי הגלגל אליו היא מחוברת.

המתח בכניסת כל אחת מהרגליים הוא אות PWM ב- Duty Cycle=50%.

מספר המחזורים (זיהוי של עליות או ירידות באות PWM) בכל סיבוב של גלגל הוא 408, כלומר רמת הדיוק של

היא  $\frac{1}{408}$  של סיבוב הגלגל. **קוטר הגלגל כולל צמיג הינו 2.36" = 60 mm.**

ע"י שימוש בתכונת Input Capture, אפשר לחשב את מהירות הרכב תאוצתו ומיקומו וכמובן לשלוט על כיוון הרכב.

● פונקציות מומלצות – כתיבת דרייברים:

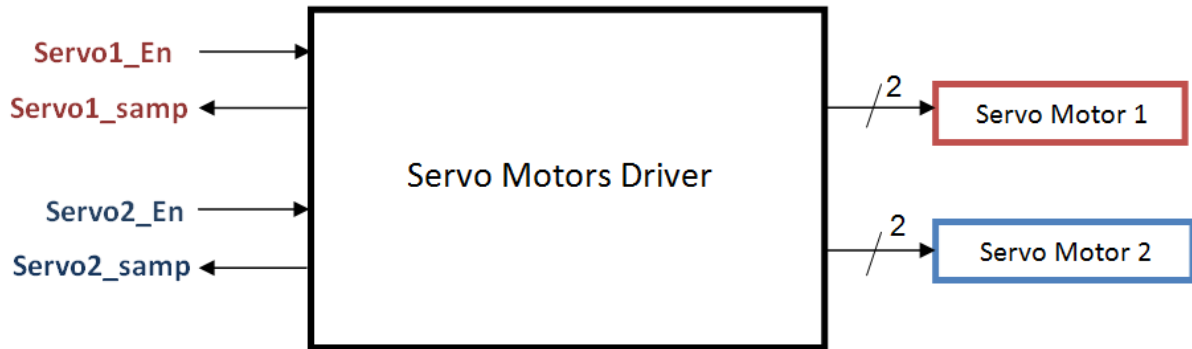
✓ פונקציית קינפוג - `Void EncodersConfig ()`

✓ פונקציית חיווי למהירות הרכב ולכיוונו - `Int* EncodersSensing ()`

הפונקציה מחזירה פוינטר למערך בגודל 2 תאים, כאשר תא ראשון מכיל את כיוון הרכב ותא שני את מהירותו.

### 3. הפעלת Servo Motors:

שימוש במנועי Servo נועד לצורך שליטה על סיבוב ידית המנוע (הזזת זרוע המחוברת לידית המנוע) בתחום זוויות בין  $0^\circ \div 180^\circ$  מעלות. מטרת השימוש במנועי ה-Servo בפרויקט זה (ישנם 2 מנועי Servo בגג הרכב) הינה לצורך שליטה על כיוון "ההסתכלות" של חיישני המרחק המותקנים על מנועי ה-Servo לצורך עקיבה במעקף מטרות.



Label Name	MCU Signal	Description	Value range	FRDM-KL25Z Pin No.
Servo1_En	PWM	שולט על זווית זרוע שמאל Servo Motor1	0v , 3.3v	PTE21 – TPM1_CH1 (J10-3 )
Servo1_samp	ADC	דגימת זרם מנוע שמאל Servo1	0v , 1.28v	PTE29 – ADC0_SE4b (J10-9 )
Servo2_En	PWM	שולט על זווית זרוע ימין Servo Motor2	0v , 3.3v	PTE31 – TPM0_CH4 (J2-13 )
Servo2_samp	ADC	דגימת זרם מנוע ימין Servo2	0v , 1.28v	PTE30 – ADC0_SE23 (J10-11 )

#### ● מיקום זוויתי של מנועי ה-Servo:

רגל Enable של המנועים (**Servo2\_En** , **Servo1\_En**) מאפשרת שליטה על מיקום זוויתי של המנועים (מיקום בזווית שבין  $0^\circ \div 180^\circ$ ) בעזרת מוצא PWM מהבקר לכל אחד מ-2 רגליים אלו. ערכי ה-Duty Cycle של אות ה-PWM קובע את מיקום הזרוע. בהגדרות הבאות יצוין ה-PWM המינימאלי עבור מיקום זוויתי של  $0^\circ$  ו-PWM מקסימאלי עבור מיקום זוויתי של  $180^\circ$ . טווחי ביניים של PWM ייתן זוויות בתחום  $0^\circ \div 180^\circ$ .

#### Servo Motors

$$f_{PWM} = 40Hz \rightarrow T_{PWM} = 25msec$$

$$T_{on} = 0.568msec \rightarrow \text{זווית של } 0 \text{ מעלות}$$

$$T_{on} = 2.2msec \rightarrow \text{זווית של } 180 \text{ מעלות}$$

### ● ביצוע סריקה ע"י מנועי ה-Servo:

- במצב של  $0^\circ$  - שני המנועים מכוונים קדימה (חיישני המרחק המורכבים עליהם מסתכלים קדימה).
- ✓ עבור מנוע שמאל (Servo Motor 1) – בטווח של  $0^\circ \div 180^\circ$  המנוע סורק את **צד שמאל** של הרכב.
- ✓ עבור מנוע ימין (Servo Motor 2) – בטווח של  $0^\circ \div 180^\circ$  המנוע סורק את **צד ימין** של הרכב.
- ✓ לצורך **סריקה** תקינה של מנועי ה-Servo יש לשנות זוויות בתחום של  $0^\circ \div 180^\circ$  ע"י שינוי הרגיסטרים השולטים על ה-Duty Cycle, בצעדים של בערך  $\pm 9$  עם השהייה של 4msec בין צעד לצעד (תלוי לכמה צעדים תרצו לחלק את  $180^\circ$  של טווח סיבוב המנוע). הסיבה לכך, **בשימוש רב** של תנועות חדות המנוע עלול להשתגע ולצאת משליטה. כמובן **שאינן לחצות את גבולות** הרגיסטרים השולטים על ה-Duty Cycle עבור גבולות של  $0^\circ \div 180^\circ$ , **אחרת המנוע יתחיל לרעוד**.
- ✓ כדי להגיע ל**זווית ספציפית** (בשונה מסריקה) אפשר להכתיב PWM עם Duty Cycle מתאים **ללא** צורך בהגעה לזווית בעזרת צעדים קטנים.

### ● מערכת בקרה והגנת מנועי ה-Servo:

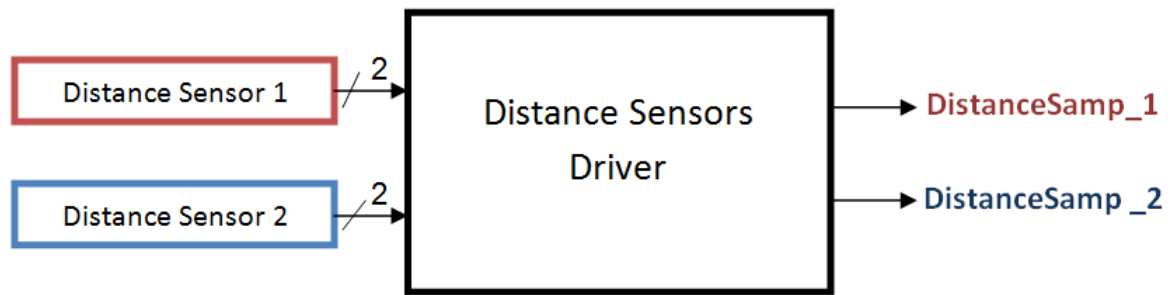
- רגלי **Servo1\_samp, Servo2\_samp** של המנועי ה-Servo דוגמים את הזרם העובר דרך המנועים.
- ✓ במצב שהמנועים נעולים על זווית מסוימת, המתח ברגליים אלו שווה 0v.
- ✓ במצב של תנועה תקינה של המנועים האות ברגליים אלו ריבועי בתדר הסיבוב, מה שחשוב הוא שהאמפליטודה של אות זה הוא **1.28v**, כאשר ערך האמפליטודה גדול מ-1.28v זהו סימן שהמנוע לא פועל כראוי (מנוע "משתולל" - עקב כתיבת קוד לא נכונה) ועליכם לתקן זאת. משוב זה לחיווי מצב פעולת המנוע מאפשר בקרה על פעילות תקינה של המנוע.
- ✓ כאשר המנוע משתולל ואתם לא מצליחים לתקן זאת בזמן אמת, **כאשר המתח ברגליים Servo1\_samp, Servo2\_samp עובר את המתח 1.505v**, נכנסת לפעולה מערכת ההגנה של מנועי ה-Servo אשר מנתקת את המנועים למשך כ-10 שניות כדי שהמנוע Servo לא ייהרס.

### ● פונקציות מומלצות – כתיבת דרייברים:

- ✓ פונקציית קינפוג למנועי ה-Servo - `Void ServosConfig ()`
- ✓ פונקציית שליטה על **מיקום זוויתי של המנועים** –
- מיקום זוויתי בין  $0^\circ$  (מבט קדימה) לזווית  $180^\circ$  – (מבט שמאלה) - `Void Servo1SetPos (int Servo1Position)`
- מיקום זוויתי בין  $0^\circ$  (מבט קדימה) לזווית  $180^\circ$  (מבט ימינה) - `Void Servo2SetPos (int Servo2Position)`
- כאשר `Servo1Position`, `Servo2Position` הן זוויות רצויות של מיקום המנוע במעלות.
- יש להגיע מזווית נוכחית של מנועי ה-Servo (ערך Duty Cycle נוכחי) לזוויות רצויות אלו (בעזרת ערך Duty Cycle רצוי, מומלץ מאוד לצעוד כמפורט לעיל בצעדים קטנים).



#### 4. חיישני מרחק:



מטרות שימוש בחיישני המרחק:

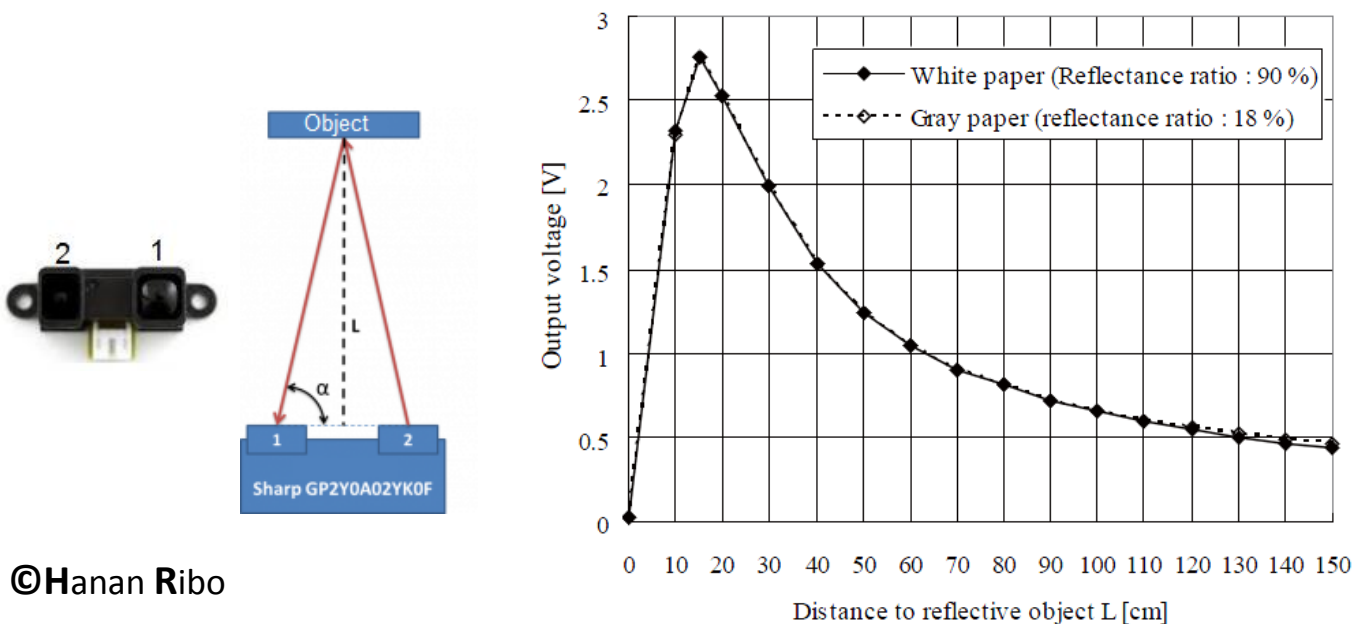
- ✓ שני חיישני המרחק מכל אחד מהסוגים הבאים, ממוקמים על הרכב (ומורכבים על 2 מנועי Servo) הינם לצורך עקיפת מטרות הנמצאים במסלול נסיעת הרכב.
- ✓ הגנה על הרכב מפני התנגשויות חזיתיות. כאשר חיישן אחד בשימוש לצורך עקיפה, חיישן שני מביט קדימה למניעת התנגשות הרכב.
- ✓ חשוב לציין, בזמן ניסויים לצורך ביצוע הפרויקט כאשר הרכב נוסע בצורה אוטונומית, עליכם לדאוג שאחד החיישנים מביט קדימה ובעזרתו אתם נמנעים מכך שהרכב יתנגש בקיר וכדומה.

#### סוג 1) חיישני מרחק מבוססי IR - IR Distance Measuring Sensors:

Label Name	MCU Signal	Description	Value range	FRDM-KL25Z Pin No.
DistanceSamp_1	ADC	חיישן מרחק שמאלי	2.8v – 0.4v	PTD6 – ADC0_SE7b (J2-17)
DistanceSamp_2	ADC	חיישן מרחק ימין	2.8v – 0.4v	PTD5 - ADC0_SE6b (J2-4)

#### ● המרה מערך דגימת מתח אנלוגי למרחק ביחידות cm:

החיישן מרחק "יורה" קרניים של אינפרא אדום וההחזרות מהחפץ שממול החיישן מתורגמות למתח אנלוגי לפי הגרף הבא (דפי המידע והמפרט נמצאים באתר הקורס). **טווח המדידה 0cm ÷ 150cm**



לעבודה נכונה, נדרש לכייל את חיישני המרחק באחת מ-2 הצורות הבאות:

✓ נדגום את הערך האמיתי בנקודות בגרף המסומנות במעין (בקפיצות של 10cm) ע"י הרחקת החלק הקדמי של הרכב באיטרציות של 10cm מקיר/משטח ודגימת ערכי ADC המתאימים ל-Flash בעזרת לחיצה על לחצן שתבחרו.

✓ אפשר לעבוד במצב DEBUG ולהרחיק משטח מהחלק הקדמי של הרכב באיטרציות של 10cm תוך רשימת ערכים מדויקים של ערכי הדגימה במיקרו-בקר).

ערכי הדגימות בין נקודות איטרציה ייעשו ע"י משוואת ישר בין 2 נקודות סמוכות. מאחר והפונקציה המתוארת בגרף הנ"ל היא בחלקה דו-ערכית כדי להחליט מה המרחק המתאים יש צורך לזהות את פיק המעבר וכך לקבוע את הערך המתאים.

● פונקציות מומלצות – כתיבת דרייברים:

✓ פונקציית קינפוג לחיישני המרחק – `Void DistSensConfig ()`

✓ פונקציית תרגום ערך דגימה דיגיטאלי (מוצא מתח אנלוגי הנדגם מחיישן המרחק) למרחק –

`Double DistanceMeasuring1 (int DisSamp1)`

`Double DistanceMeasuring2 (int DisSamp2)`

כאשר DisSamp1 ו-DisSamp2 ערכי מתח הנדגמים מרגליים PTD5, PTD6.

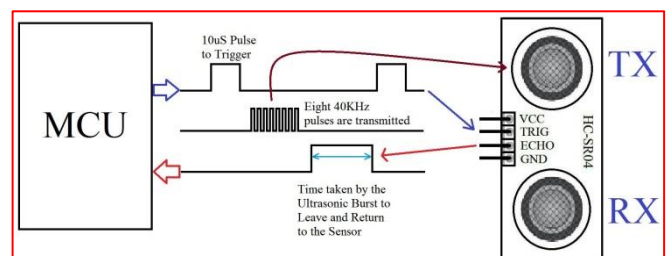
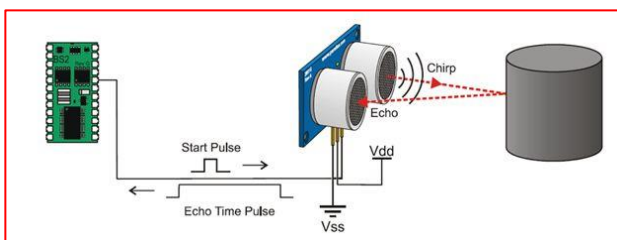
הערך המוחזר מהווה את המרחק הפיסי ביחידות של cm. במידה והכיוול של 2 חיישני המרחק זהה אפשר להשתמש בפונקציה אחת עבור שניהם.

**סוג 2) חיישני מרחק מבוססי גלי קול – Ultra-Sonic Distance Measuring Sensors**

כאשר ניזום דרך הבקר פולס ברוחב של **לפחות** 10uSec המהווה טריגר דרך רגל **Trigger** של החיישן (מרווח מינימאלי בין טריגר לטריגר הוא 60mSec, כלומר תדר עבודה מקסימאלי של 16.7Hz), **בסיום** הפולס חיישן המרחק "יורה" גל קול באורך שמונה מחזורים בתדר 40kHz לכיוון האובייקט וקולט את ההחזרים המגיעים ממנו. מעגל חשמלי הנמצא בחיישן ממיר את החזרי גל הקול כתוצאה מהפולס וממיר אותו לפולס היוצא מרגל **Echo** של החיישן ונכנס לרגל הבקר בעל יכולת פסיקה (דפי המידע והמפרט נמצאים באתר הקורס). טווח המדידה  $2cm \div 400cm$  (במקרה שלנו, הטווח המעשי  $2cm \div 150cm$  עקב החזרות מהרצפה). מידת המרחק תתבצע ע"י אחת משתי הנוסחאות הבאות:

$$1. \quad Range[cm] = Echo \text{ high level time} \cdot 17000 \quad \text{when } 17000 \left[ \frac{cm}{sec} \right] \text{ is sound velocity}$$

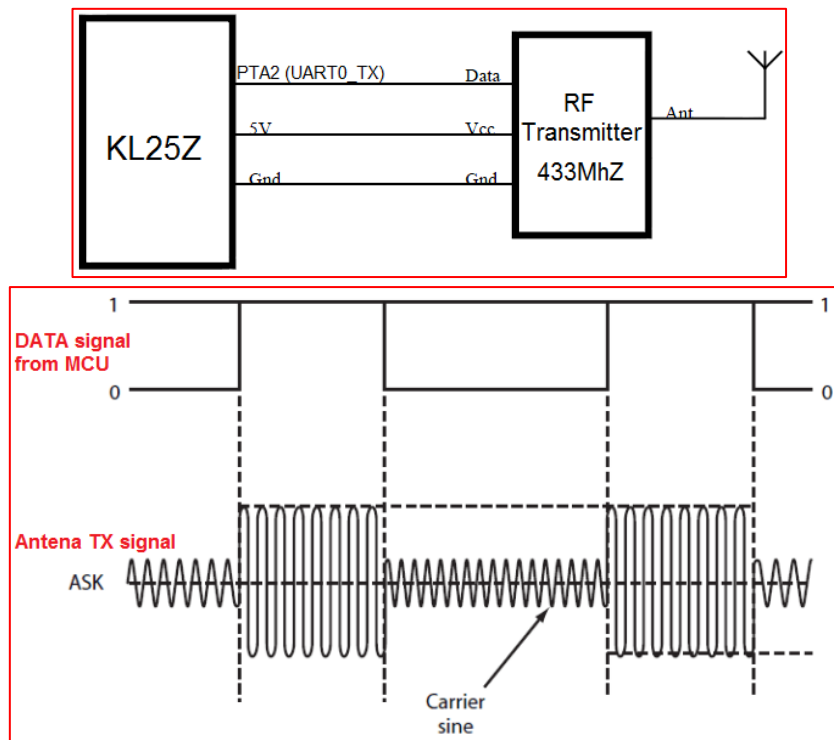
$$2. \quad Range[cm] = \frac{\text{time interval between sending trigger signal and receiving echo signal}}{58}$$



Label Name	Description	FRDM-KL25Z Pin No.
UltraSonic1_Echo	אות ECHO המגיע מחיישן מרחק שמאלי	PTD2 (Digital Input Interrupt capability)
UltraSonic2_Echo	אות ECHO המגיע מחיישן מרחק ימין	PTD3 (Digital Input Interrupt capability)
UltraSonic_Trigger	אות Trigger משותף לשני החיישנים	PTD1(TPM0_CH1)

## 5. משדר RF בתדר 433MHz המשדר אות דיגיטלי באפנון ASK:

חלק מדרישות הפרויקט היא לשלוח ל- PC בזמן אמת את מה "רואה" הרכב וכך לבנות ב- PC את **מבנה מסלול נסיעת הרכב**. שליחת המידע תשודר בצורה טורית למקלט (את המקלט אינכם נדרשים לממש) בתקשורת RF. המשדר מחובר לבקר ברגל **PTA2 (UART0\_TX)** כך שנוכל להתייחס אליו כתקשורת טורית בפרוטוקול UART. לידע כללי, במוצא הבקר לרגל DATA של המודול נכנס אות דיגיטלי ובהתאם לכך המודול משדר גל אלקטרומגנטי בתדר 433MHz באמפליטודה נמוכה למשך זמן שקו DATA ב- '0' ובאמפליטודה גבוהה למשך זמן שקו DATA ב- '1'. זהו למעשה הסבר קצרצר לאפנון הנקרא ASK, ראה איורים הבאים.



מאחר שהרגישות של מודול המקלט היא גבוהה ( $-103\text{dBm}$ ) המשמעות היא שהמודול יכול לקלוט אותות ממש חלשים ולתרגם אותם למידע בינארי. כדי להתגבר על הפרעות **נשתמש בסנכרון מידע בתוכנה**, ע"י עטיפת המידע בסיסמת פתיחה וסיסמת סיום המהוות רצף של תווים המוגדרות מראש באורכן ובתוכנם. שליחת רצף של מידע (רצף בתים) תיעשה עם סיסמת פתיחה וסיסמת סיום כך המקלט יידע להינעל רק על מידע בין שתי הסיסמאות ולהתעלם מכל מידע אחר שאינו עונה על התנאים. במצב זה ישנה תקורה למידע אך היא הכרחית לצורך קליטה תקינה.

## להלן מספר הבהרות טכניות:

- כדי לבצע שידור תקין של פרוקול UART דרך אנטנה (ולא כבל המחובר לצד השני) דרך רגל PTA2, יש צורך להגדיר רק את רגל הבקר דרכה יש שידור.

```
PORTA_PCR2 = PORT_PCR_MUX(2) | PORT_PCR_DSE_MASK; // TX
```

- קצב שידור יהיה Baud Rate = 4800.
- שידור של רצף תווים בגודל כלשהו יתבצע בצורה הבאה. ראשית משדרים סיסמה לצורך סנכרון תוכנית, לאחר מכן משדרים את המידע המורכב מרצף תווים (data frame) ולסיום משדרים את הערך 0x00.

ניתן להשתמש בהגדרת הפונקציה הבאה:

```
void RF433MHz_transmitter(UART_MemMapPtr channel, char* Buffer){
    UARTprintf(UART0_BASE_PTR, "qwer"); // qwer is the Preamble password
    UARTprintf(UART0_BASE_PTR, Buffer); // send data frame buffer
    uart_putchar (UART0_BASE_PTR, 0); // End data frame character
}
```

```
RF433MHz_transmitter(UART0_BASE_PTR, Buffer);
```

הפונקציות UARTprintf, uart\_putchar מוגדרות בקוד לדוגמה של ניסוי 5.

## 6. עקיבה של מסלול שחור בעזרת מערך של 5 חיישני צבע (Reflectance Sensor):

לצורך עקיבה על מסלול שחור מותקן בתחתית הרכב מערך של 5 חיישני צבע במרחק של  $\frac{3}{4}$  רוחב פס איזולירבנד. יוצא מכך שפס איזולירבנד שחור יכול להראות ע"י חיישן בודד או ע"י 2 חיישנים סמוכים. על הרכב ישנו מעגל הדואג לכך שכאשר מול אחד מהתאים מופיע צבע שחור ומוצא הרגל (Digital Input) המתאימה לתא זה ערכה 1 לוגי, אחרת מוצא הרגל 0 לוגי. לאור הנ"ל, ישנם 9 מצבי זיהוי של הפס השחור ע"י מערך של 5 חיישני צבע (אחד לכל חיישן ואחד לכל 2 סמוכים).

Label Name	Description	FRDM-KL25Z Pin No.
DistanceSensor1	מוצא חיישן צבע 1 מצד ימין לרכב	PTE2 - Input (J9-9)
DistanceSensor2	מוצא חיישן צבע 2 מצד ימין לרכב	PTE3 - Input (J9-11)
DistanceSensor3	מוצא חיישן צבע 3 מצד ימין לרכב	PTE4 - Input (J9-13)
DistanceSensor4	מוצא חיישן צבע 4 מצד ימין לרכב	PTE5 - Input (J9-15)
DistanceSensor5	מוצא חיישן צבע 5 מצד ימין לרכב	PTA16 - Input (J2-9)

כאשר הרכב נמצא על פס ישר המשמעות שחיישן הצבע האמצעי DistanceSensor3 (ערך הרגל PTE4) ערכו במוצא 1 לוגי, אחרת ערכו 0 לוגי (כאשר כל רגלי הבקר בטבלה הנ"ל נמצאים במצב של pull-down). ישנם 9 מצבים המפורטים לעיל המאפשרים פלטפורמה לאלגוריתם בקרה חכם לשמירת הרכב על הפס שחור.

● פונקציות מומלצות – כתיבת דרייברים:

- ✓ פונקציית קינפוג לחיישני הצבע – `Void ColorSensConfig ()`
- ✓ פונקציית שמירת הרכב על פס שחור – `Void BlackLineKeeping ()`

## 7. חיווי מצב סוללה:

סוללת הליתיום המחוברת לרכב במצב מלא ערכה הממוצע הוא 7.5v כאשר מתג הפעלת הרכב במצב ON כנסת הסוללה לשימוש. הערך המינימאלי המותר במוצא הסוללה הינו 6v. על כרטיס הרכב ישנו מעגל הגנה על הסוללה, כאשר מתח הסוללה יורד ל-6.05v המנועים מתנתקים, נדלקת נורה ומופעל זמזם. סדר הפעולות המידי שיש לבצע במצב זה,

★ חורדת מתג הפעלת הרכב למצב OFF

★ חיבור הסוללה למטען (ראה נספח שימוש במטען ליתיום).

לצורך חיווי של מתח הסוללה, מאופשר ע"י דגימת מתח הסוללה ברגל PTE22.

בגב הרכב מותקנים 3 לדים אדומים המאפשרים לחלק את טווח המתח בין 7.5v ל-6.05v ל-8 דרגות.

במצב זה תוכלו לדעת בכל רגע נתון את מצב הסוללה ולהשתמש נכון בתכונת LPM של הבקר וכו'.

Label Name	Description	FRDM-KL25Z Pin No.
<b>BattSamp</b>	דגימת מתח הסוללה בין ערכים 1.02v – 1.23v	PTE22 – ADC0_DP3/ADC0_SE3 (J10-5)
<b>BattLed1</b>	לד 1 מתוך 3 – חיווי חיצוני למצב סוללה	PTB8 - Output (J9-1)
<b>BattLed2</b>	לד 2 מתוך 3 – חיווי חיצוני למצב סוללה	PTB9 - Output (J9-3)
<b>BattLed3</b>	לד 3 מתוך 3 – חיווי חיצוני למצב סוללה	PTB10 - Output (J9-5)

● פונקציות מומלצות – כתיבת דרייברים:

- ✓ פונקציית קינפוג ללדים למצב הסוללה – `Void BattLedConfig ()`
  - ✓ פונקציית להדלקת לדים לחיווי מצב הסוללה חלוקה ל-8 רמות - `Void BattLedState (int BattSamp)`
- כאשר BattSamp זהו ערך הדגימה אנלוגית של הסוללה בין ערכים 1.02v – 1.23v יחסי לטווח המותר של הסוללה 7.5v – 6.05v

## 7. מודול DTMF:

בעזרת חיבור של מכשיר סלולרי למעגל חשמלי מבוסס רכיב DTMF הנמצא ברכב הרובוטי (שתקבלו לקראת פרויקט הגמר), החיבור אליו הינו דרך חיבור אוזניית הטלפון במקרה של חיבור טלפון סלולרי. דרך התקשרות ממרחק עם הטלפון המחובר לכרטיס (או לחילופין בעזרת אפליקציית DTMF בטלפון החכם) נוכל להפעיל כל דבר שנרצה, זאת ע"י כך שבכל לחיצה על אחד מתוך 12 לחצני המכשיר המתקשר (כמתואר למטה תחת כותרת תיאור הפרויקט), ייווצר צליל שונה המופק מחיבור 2 טונים בתדרים שונים, מוצא הכרטיס ממיר כל צליל לספרה המתאימה בעזרת מוצא בינארי של 4 ספרות, כך למעשה נוצרת התקשרות לכיוון אחד בלבד. ישנו פין נוסף המהווה פין לפסיקה לכל פעם שנלחץ במכשיר המתקשר על לחצן מחדש.



Label Name	Description	FRDM-KL25Z Pin No.
DTMF_Q1	Q1 מוצא DTMF	PTC11 - Input (J1-15)
DTMF_Q2	Q2 מוצא DTMF	PTC12 - Input (J2-1)
DTMF_Q3	Q3 מוצא DTMF	PTC13 - Input (J2-3)
DTMF_Q4	Q4 מוצא DTMF	PTA17 - Input (J2-11)
DTMF_Interrup	DTMF מוצא לפסיקה	PTD7 - Input Interrupt(J2-19)

שימוש ב- DTMF אפשרי ורצוי בשלב עבודה על נסיעת הרכב לצורך מניעת התנגשות הרכב בקיר וכדומה. כאשר הרכב במצב אוטונומי נדרשת לעיתים התערבות חיצונית **במהלך פיתוח הקוד**.

## 8. כתיבת תוכנת בדיקה (לפני חלוקת הרכב):

כאשר הרכב מונח על ספוג (תקבלו בערכת הרכב) כך שהגלגלים באוויר הבדיקות תתבצענה באופן הבא: בעזרת תפריט בחירה הבא המוצג במסך המחשב (נשלח דרך הבקר בתקשורת טורית) ומכיל את בחירת הבדיקות המפורטות בקובץ זה. ניתן לבדוק את כל המודולים של הרכב, הנדרשים לפרויקט שלכם.

<u>Vehicle checking menu</u>
1. DC MOTORS and ENCODERS
2. SERVO MOTORS
3. IR Distance Measuring Sensors
4. Ultra-Sonic Distance Measuring Sensors
5. Reflectance Sensor
6. Battery
7. DTMF

תנאי לקבלת הרכב, הוא ביצוע תוכנית הבדיקה בשימוש סקופ (לבדיקת צורת האות במוצא הבקר), מחולל (ליצירת אות PWM), נגד משתנה (הכנסת מתח משתנה לצורך מדידתו), וולט מטר (למידת מתח).

פירוט הנדרש לביצוע עבור כל שורה בתפריט, מופיע בעמוד הבא.

### ● DC MOTORS and ENCODERS

- לאחר בחירה מצב זה (תו '1') עוברים לאופן פעולה של שליטה על נסיעת הרכב וקבלת מידע מה-Encoders באופן הבא:
- ✓ בלחיצה על תו **S** - שני המנועים יפעלו בשיא המהירות (Duty Cycle = 100%).
- ✓ בלחיצה על תו **P** - שני המנועים יעצרו.
- ✓ בלחיצה על תו **U** - נדרש להעלות את המהירות של 2 המנועים ב- 10% עד למקסימום, לחיצה נוספת על **U** עוברת למינימום מהירות של 2 המנועים, ועל זה הדרך בצורה מחזורית.
- ✓ בלחיצה על תו **D** - נדרש להוריד את המהירות של 2 המנועים ב- 10% עד למינימום, לחיצה נוספת על **D** עוברת למקסימום מהירות של 2 המנועים, ועל זה הדרך בצורה מחזורית.
- ✓ בלחיצה על תו **E** - מתבצעת בדיקת ה-ENCODERS. יש לרשום על המסך את המהירות העכשווית של הגלגלים (רק בשינוי מהירות הרכב לעדכן את המהירות על המסך ע"י דריסה של המהירות הקודמת).

### ● SERVO MOTOR

- לאחר בחירה מצב זה (תו '2') עוברים לאופן פעולה הבא:
- ✓ לחיצה על תו **F** - שני מנועי ה-Servo **ימוקמו בחזית הרכב**. זווית 0 מעלות למנוע ימין וזווית 180 מעלות למנוע שמאל.
- ✓ לחיצה על תו **M** - מזיזה את 2 מנועי ה-Servo ב-10 מעלות (Servo1 שמאלה ו-Servo2 ימינה). וכשמגיעים המנועים לקצה, לחיצה נוספת על **M** מזיזה את 2 מנועי ה-Servo ב-10 מעלות לכיוון חזרה עד להגעה לזווית 0 מעלות, ועל זה הדרך.

### ● Distance Measuring Sensors

- לאחר בחירה מצב זה (תו '3', '4'), יש להדפיס על המסך את המרחק ממשטח (דף נייר) עד לחיישנים כאשר שני מנועי ה-Servo ימוקמו בחזית, זו למעשה מדידת מרחק מקדמת הרכב. רישום מרחק בנפרד לכל חיישן מרחק (רק בשינוי מרחק יש לעדכן את המרחק על המסך ע"י דריסה של המרחק הקודם).

### ● Reflectance Sensor

- לאחר בחירה מצב זה (תו '5'), יש להדפיס על המסך את הערך המתקבל ממערך של 5 חיישני הצבע.

### ● Battery

- לאחר בחירה מצב זה (תו '6'), יש להדפיס על המסך את מתח הסוללה (ביחידות volt).

### ● DTMF

- לאחר בחירה מצב זה (תו '7'), ניתן ללחוץ על מקשים במכשיר הנייד והצגתם על מסך המחשב.

## 9. מבנה הציון בפרויקט

- הציון יינתן על-פי קריטריונים של עמידה במשימות, זמן ביצוע המסלול כולל הקפת המטרות, דו"ח מסכם והגנה על הפרויקט. בהגנה הסטודנט אמור לענות לשאלות בע"פ ולהפגין את ידע והבנה שנרכשו תוך ביצוע הפרויקט (חומרה ותוכנה) על כל מרכיביו באופן מעמיק.
- כל קבוצה תגיש דו"ח מסכם המתעד את החומר והתוכנה שפותחו במסגרת הפרויקט. תיאור התוכנה ילווה בתרשים זרימה מפורט, תיאור הרוטינות העיקריות, פסיקות, תיאור האלגוריתם וכו'.

● הדו"ח ידגיש גם את הנושאים הבאים:

- (1) פירוט אלגוריתם הבקרה לעקיבה וסריקת מטרות
- (2) פירוט אלגוריתם למסלול פגיעה במטרות
- (3) תהליך קבלת החלטות כפונקציה של דגימת החיישנים
- (4) פירוט תרשימי זרימה או PSEUDO CODE המתארים את הסעיפים (כל סעיף בנפרד)
- (5) תיעוד קטעי הקוד המתארים את מימוש הסעיפים (כל סעיף בנפרד)

● כל קבוצה תתעד את הפרויקט בסרט וידאו ותעלה קישור ליו-טיוב !!

● בנוסף לציון הפרויקט גמר:

הבנוסף לציון יינתן על פי זמן הביצוע היחסי (לקבוצות אחרות) של עקיבה אחר המסלול כולל עקיפת מטרות.

✓ לקבוצה שתשיג את זמן הביצוע המינימאלי יוענק בonus של 10% לציון הפרויקט.

✓ לקבוצה שתגיע במקום השני יוענק בonus של 5%.

✓ לקבוצה שתגיע במקום השני יוענק בonus של 2.5%.

✓ לשאר הקבוצות לא יוענק בonus.

תאריך הגשת הפרויקט:

הסבר על חלוקת הצגת הפרויקט לפי תחנות מבוקרות של כל שלב בנפרד, יפורסמו בתקשוב.

הצגה סופית (תחרות בין הקבוצות והגנה על הפרויקט) יתקיימו בתערוכה בתאריך 29/6/17 בנוכחות קהל.

**בהצלחה לכולם!**