



**שנקר-בי"ס גבוה להנדסה ולעיצוב**

**הפקולטה להנדסה**

**המחלקה להנדסת תוכנה**

**FollowThatCar**

**פרוייקט גמר**

מאת:

נועם רונס 38827903

אורי גרינברג (פרץ) 39771035.

מוגש כחלק מהדרישות לקבלת תואר ראשון

בוגר במדעים (B.Sc.).

4.9.2013

## הכרת תודה

אנו רוצים להודות ליאיר שעזר לנו רבות עם בניית כל הרכיבים האלקטרוניים, לזוהר שהיה תמיד זמין לעזרה ותמיכה. כמובן תודה למחלקה בכלל ולאמנון בפרט שסיפקו לנו את כל הציוד והרכיבים האלקטרוניים אשר נדרשו לנו לצורך השלמת הפרויקט.

## תקציר מנהלים

בשנים האחרונות אנו עדים לכך שרכבינו הופך ליותר ויותר רובוטי. המטרה של הפרויקט שלנו היא להשתלב במגמה המפחיתה את המעורבות האנושית בנסיעה לטובת הגדלת מעורבות מחשב הרכב. בפרויקט שלנו אנו מעוניינים לדמות נסיעה בפקק, דהיינו רכב שעוקב באופן כמעט קבוע אחר רכב אחר. אנו נבנה ונתכנן רכב שיודע לעקוב אחר הרכב שלפניו במצב של נהיגה בפקק, ובכך יעביר את תפקיד השליטה ברכב לנהג הרובוטי וישחרר את הנהג מנסיעה זו.

בפרויקט זה ייושם: זיהוי כלי רכב מלפנים ע"פ זיהוי לוחית הרישוי (בעזרת מצלמה), עיבוד הנתונים מהמצלמה והחלטה איך והאם להתקדם לעבר הרכב, שליטה פיזית ברכב ע"י נתינת פקודות.

בכדי לממש מטרות אילו הרכבנו על גבי רכב מצלמת רשת פשוטה, מצלמה זאת חוברה למחשב. במחשב התקנו את חבילת התוכנה OpenCV, זוהי ספריית פונקציות שנועדה לעזור לפתח יישומים של ראייה ממוחשבת, הכוללת בתוכה פונקציות לעיבוד תמונה רבות.

במחשב עיבדנו את הנתונים שהתקבלו מהמצלמה. ע"י פונקציות של מציאת קווי מתאר של עצמים בתמונה (אלגוריתם Canny), איסוף המבנים סגורים, מציאת מרובעים, ולבסוף מציאת פורפורצייה מדויקת של לוחית זיהוי. לאחר זיהוייה איתרנו את מיקומה של לוחית רישוי במרחב.

ע"פ הזיהוי, העברנו פקודות למיקרו-בקר לצורך הנעת הרכב. המיקרו-בקר הפעיל בפועל את הרכב ע"י שני מנועים, מנוע DC פשוט השולט על הנעת הרכב ובעזרת זרמים למגנט שלטנו בכיוון הרכב.

מטרות הפרויקט ייושמו והושגו. המערכת אשר הורכבה על הרכב עקבה בצורה מוצלחת אחר הרכב שלפניה. למרות הצלחת הפרויקט, ישנן כמה נקודות שניתן לשפר: בעקבות אי שליטה במהירות הנסיעה הרכב נסע בצורה לא רציפה – מקטעים קצרים של עקיבה. בנוסף, המגנט שלט בגלגלים בצורה בינארית (ימינה או שמאלה) ולא בזוויות מדויקות, בעקבות זו נדרשו תמרונים/תיקונים רבים בכדי להתיישר מאחורי הרכב שלפניו.

## תוכן העניינים

1	המטרות.....	1
2	תקציר מסמך האפיון (SRS).....	2
2.1	מטרת הפרויקט.....	2
2.2	היקף הפרויקט ומאפייני המוצר.....	2
2.3	תיאור כללי.....	2
2.4	תכונות מוצר.....	3
2.5	ממשקי משתמש.....	3
2.6	סבילות לשגיאות משתמש.....	3
2.7	דרישות שאינן פונקציונליות.....	3
3	תקציר מסמך התיכון (FRS).....	4
3.1	סקירה כללית.....	4
3.2	תרחישים.....	4
3.3	מטרות שאינן חלק מהפרויקט.....	4
3.4	תרחישי שימוש.....	5
3.5	אפיון מסכים.....	6
3.6	נושאים פתוחים.....	6
4	מבוא.....	7
5	סקר ספרות/מערכות.....	8
8	מכוניות ללא נהג:.....	8
10	לקריאה נוספת:.....	10
6	שיטות.....	11
11	מחקר ופיתוח המוצר.....	11
7	היישום.....	12
7.1	תאור כללי של היישום.....	12
7.2	מתאר החומרה:.....	12
7.3	תאור התוכנה:.....	13
7.4	ארכיטקטורת המעגל.....	16
7.5	מיקרו – בקר:.....	17

19	7.6 מנהל התקן המנועים (motor driver carrier):
20	8. בדיקות, תוצאות והערכה
20	8.1 תוצאות
21	8.2 הערכה
21	9. דיון: מסקנות ופיתוחים עתידיים
23	10. מקורות
23	11. נספחים
23	דוגמה לריצת מבחן:

## רשימת הטבלאות והאיורים

5	איור 1: תרחישי שימוש
8	איור 2: תיאור מיכשור המכונית של גוגל
12	איור 3: מתאר חומרה
13	איור 4: דיאגרמת מצבים זיהוי לוחית
16	איור 5: ארכיטקטורת המעגל
17	איור 6: דיאגרמת מצבים של תוכנת המיקרו-בקר
19	איור 6: מעגל motor driver carrier

## מילון מונחים

**עיבוד תמונה (Image processing):**

עיבוד תמונה הוא תחום העוסק בעיבוד של תמונה או וידאו ובמניפולציות שונות עליה. לרוב העיבוד נעשה בתחום הדיגיטלי לאחר שהתמונה כבר נדגמה. המטרות העיקריות הן שיפור התמונה והסקת מסקנות על התמונה. הפלט של עיבוד תמונה יכול להיות גם תמונה או סט של מאפיינים או פרמטרים הקשורים לתמונה (ע"פ וויקיפדיה)

**OpenCV (Open Source Computer Vision)**

OpenCV היא ספרייה של פונקציות תוכנת לראייה ממוחשבת בזמן אמת. הספרייה היא בקוד פתוח (BSD) והיא נועדה לשימוש אקדמי ומסחרי. יש לה ממשקים ב C, C++, Python, הרצים על Windows, לינוקס, אנדרואיד ו-Mac. בספרייה יש למעלה מ-2500 אלגוריתמי אופטימיזציה. (מתוך האתר <http://opencv.org/>)

**מיקרו בקר (Microcontroller)**

מיקרו בקר הוא מחשב קטן במעגל משולב יחיד המכיל את ליבת מעבד, זיכרון, וציוד היקפי לקלט / פלט הניתן לתכנות. בקרים נמצאים בשימוש במוצרים ומכשירים, כגון מערכות רכב מנועי בקרה, התקנים רפואיים מושתלים, וכו' (ע"פ וויקיפדיה)

**מערכת משובצת מחשב (Embedded system)**

מערכת משובצת מחשב היא מערכת עם פונקציה ייעודית בתוך מערכת מכאנית או חשמלית רחבה יותר, לעתים קרובות במערכות real time. המערכת לרוב היא חלק ממכשיר מלא, המערכת כוללת חומרה וחלקים מכאניים. מערכות משובצות נמצאות במרבית מחשירי החשמל הנמצאים סביבנו.

**Motor Driver Carrier**

התקן לשליטה על מנועים משתמש לרוב ב H bridge.

## **H-BRIDGE**

הוא מעגל אלקטרוני המאפשר למתח להיות מיושם על פני עומס בשני הכיוונים. מעגלים אלה משמשים לעתים קרובות ברובוטיקה כדי לאפשר למנועי DC לפעול בשתי בכיוונים.

## **LIDAR** (Light Detection and Ranging)

זוהי טכנולוגיית חישה אופטית מרחוק, מקבילה לרדאר. הטווח (מרחק) לאובייקט, נקבע על ידי מדידת עיכוב הזמן שבין שידור של דופק וזיהוי של האות המוחזר.

## **Kinect**

"זהו בקר משחקים שמיוצר ע"י מיקרוסופט... הוא מאפשר שליטה ללא שלט או בקר אלא על ידי חישה של תנועות הגוף. ה Kinect-מבוסס על טכנולוגיה של חברת פריים-סנס הישראלית."

(Wikipedia)

## **ROS** (Robot Operating System)

היא מערכת הפעלה רובוטית והיא סביבת עבודה, המספקת פונקציונליות בדומה למערכת הפעלה במחשב, כגון הפשטת חומרה, בקרה על מכשירים ברמה נמוכה, העברת הודעות בין תהליכים וכדומה.

## **Arduino**

הוא מיקרו בקר בעל מעגל מודפס יחיד (Single Board MicroController), עם סביבת פיתוח משולבת (IDE) ברישון קוד פתוח, אשר מטרתה ליצור סביבה נוחה וזולה לפיתוח פרויקטים המשלבים תוכנה עם רכיבי אלקטרוניקה.

## **אלגוריתם Canny**

"אלגוריתם קני הוא אלגוריתם רב-שלבי בתחום עיבוד תמונה שמטרתו היא זיהוי קצוות בתמונה"

(Wikipedia)

## 1. המטרות

- בשנים האחרונות אנו עדים לכך שרכבינו הופך ליותר ויותר רובוטי. זה מתחיל בתיבה האוטומטית וממשיך בחיישנים הרבים שמתווספים כיום לרכבים הן לצורכי נוחות והן לצורכי בטיחות. אך ברור שהמגמה המסתמנת היא מעבר לרכב אוטומטי לגמרי, ואנו מתקדמים לכך בצעדי ענק.
  - בפרויקט שלנו אנו מעוניינים לדמות נסיעה בפקק, (מצב שרק נהיה שכיח יותר ויותר, עם התרבות כלי הרכב בכבישים) דהיינו רכב שעוקב באופן כמעט קבוע אחר רכב אחר. (מיותר לציין שנסיעה זאת גוררת לא מעט תסכול אצל הנהג האנושי)
  - המטרה של הפרויקט שלנו היא להשתלב במגמה המפחיתה את המעורבות האנושית בנסיעה לטובת הגדלת מעורבות מחשב הרכב. אנו נבנה ונתכנן רכב שיודע לעקוב אחר הרכב שלפניו במצב של נהיגה בפקק, ובכך יעביר את תפקיד השליטה ברכב לנהג הרובוטי וישחרר את הנהג מנסיעה זו.
- אם כך מטרות הפרוייקט הן:
- זיהוי כלי רכב מלפנים ע"פ זיהוי לוחית הרישוי (בעזרת מצלמה).
  - עיבוד הנתונים מהמצלמה והחלטה איך והאם להתקדם לעבר הרכב (ימינה/שמאלה/גז/ברקס).
  - שליטה פיזית ברכב ע"י נתינת פקודות.



## 2. תקציר מסמך האפיון (SRS)

### 2.1 מטרת הפרויקט

פיתוח מערכת אשר תאפשר נהיגה אוטומטית מלאה של הרכב במצב של גודש תנועה. המערכת תאפשר (בלחיצת כפתור פשוטה) לכל נהג שאיבד את שעותיו בעקבות העצבים היומיומיים של נסיעה בפקק להעביר את רכבו למצב של נהיגה אוטומטית. ובכך להשתחרר מהעול שבנהיגה איטית, מונוטונית ומייגעת שהוא הנסיעה בפקק. ברגע הגעה לפקק כל שנדרש לעשות הוא ללחוץ במוניטור שברכב על אפליקציית ה-FollowThatCar הקיימת וללחוץ על כפתור ה-On לחכות שמהערכת תזהה את הרכב מלפנים ולאחר מכן להישען לאחור ולהתרווח.

### 2.2 היקף הפרויקט ומאפייני המוצר

נפתח מערכת הנשענת על ספריות OpenCV (שפותחו ע"י Intel) ומאפשרת מעקב אחרי רכב אחר(מלפנים) ע"י עיבוד תמונה וזיהוי לוחית הרישוי של הרכב.

### 2.3 תיאור כללי

#### פרספקטיבת מוצר:

המוצר הינו מערכת המאפשרת מעקב אחרי הרכב שמלפנים בעת נסיעה בפקק באמצעות זיהוי ומעקב אחר לוחית הזיהוי של הרכב באמצעות עיבוד תמונה.

#### מאפייני משתמש:

המערכת מותאמת לכל נהג באשר הוא אשר בעל רכב וסובל מנהיגה בפקק.

#### סביבה מבצעית:

סביבות השימוש במוצר מתבצעת אך ורק בתנאי עומסי תנועה בהם הנסיעה איטית.

#### עיצוב ואילוצי יישום:

בגרסה הראשונית של המערכת לא תוכנן ממשק משתמש. אך בעתיד המטרה היא אפליקציה בעלת GUI פשוט שדומה ללחיצה על כפתור ה-cruise ברכבים רבים בימינו. המערכת עצמה(שהיא החלק הארי של הפרויקט) תיושם ע"י שפת C/C++ ובעזרת ספריות OpenCV המספקות פונקציות לעיבוד תמונה.

## 2.4 תכונות מוצר

- ניתן להשתמש במערכת בכל רכב בעל מוניתור מובנה המאפשר ממשק משתמש בסיסי התומך במערכת.
- ע"מ להשתמש במערכת נדרש המשתמש להיות במצב של עומס תנועה כלשהו(המערכת עובדת עד מהירות מסוימת – אחרת זה לא עומס).
- בכדי שהמערכת תוכל לעבוד, חובה שברכב שבו המערכת מותקנת תהיה מותקנת מלצלמה בעלת רזולוציה אשר תומכת בסטנדרטים של התוכנה.
- חובה שיהיה מותקן ברכב מחשב אשר תומך בצריבת התוכנה עליו.

## 2.5 ממשקי משתמש

אפיון המסכים של המוצר:

- מסך הבית.
- הפעלה תקינה(צילום בזמן אמת של הרכב מלפנים).
- מסך שגיאת זיהוי רכב.
- התראת יציאה מהעקיבה(שינוי מהירות וכו').
- מסך סיום נסיעה אוטונומית.

## 2.6 סבילות לשגיאות משתמש

במקרה שהמשתמש מנסה להפעיל את המערכת במצב שאינו עומס, כלומר מהירות הנסיעה עולה על המותר(כפי שמוגדר במערכת) המערכת תיתן הודעת שגיאה שתסביר שהסיטואציה אינה מאפשרת את מצב הנהיגה האוטונומית.

יכול להיווצר מצב שבו לרכב שמלפנים אין לוחית זיהוי או שהלוחית זיהוי שלו מעוות או יוצרה בצורה שונה מהמוגדר במערכת כך שהמערכת לא תצליח לזהות את לוחית הזיהוי ובכך המערכת לא תינעל עליו ולא תכנס למצב הנהיגה האוטונומית.

## 2.7 דרישות שאין פוקנציונליות

המערכת מתוכננת לפעול על כל רכב בעל מחשב רכב המאפשר צריבה של תוכנה זו, מצלמה וממשק משתמש המאפשר גישה לאפליקציה של המערכת.

### 3. תקציר מסמך התיכון (FRS)

#### 3.1 סקירה כללית

בשנים האחרונות ישנה מגמה ברורה, שהיא הפיכת הרכב לאוטומטי, ללא כל שיתוף של האדם.

הפרויקט שלנו ממחיש נהיגה בפקק, שבו רכב נוסע באיטיות ובקרבה אחר האוטו שלפניו.

המטרה של הפרויקט שלנו היא להשתלב במגמה המפחיתה את המעורבות האנושית בנסיעה לטובת הגדלת מעורבות מחשב הרכב. אנו נבנה ונתכנן רכב שיודע לעקוב אחר הרכב שלפניו במצב של נהיגה בפקק, ובכך יעביר את תפקיד השליטה ברכב לנהג הרובוטי וישחרר את הנהג מנסיעה זו.

#### 3.2 תרחישים

כמעט לא קיים אדם בארץ שיש לו אותו שלא נתקע על כביש איילון בשעת עומס. זה מייגע, מעייף ואף מעצבן.

המערכת הזו מונעת את כל הרגשות האלה וכל העבודה הסיזיפית שבנסיעה בפקק.

כעת, לאחר שהמערכת מותקנת ברכבו ומקוונת, כאשר אדם מגיע כל בוקר לפקק הקבוע הוא מפעיל את המערכת. מערכת זו בעזרת מצלמה מזהה את הרכב שלפניו, עוקבת אחריו תוך כדי שמירה על מרחק קבוע ובטיחותי.

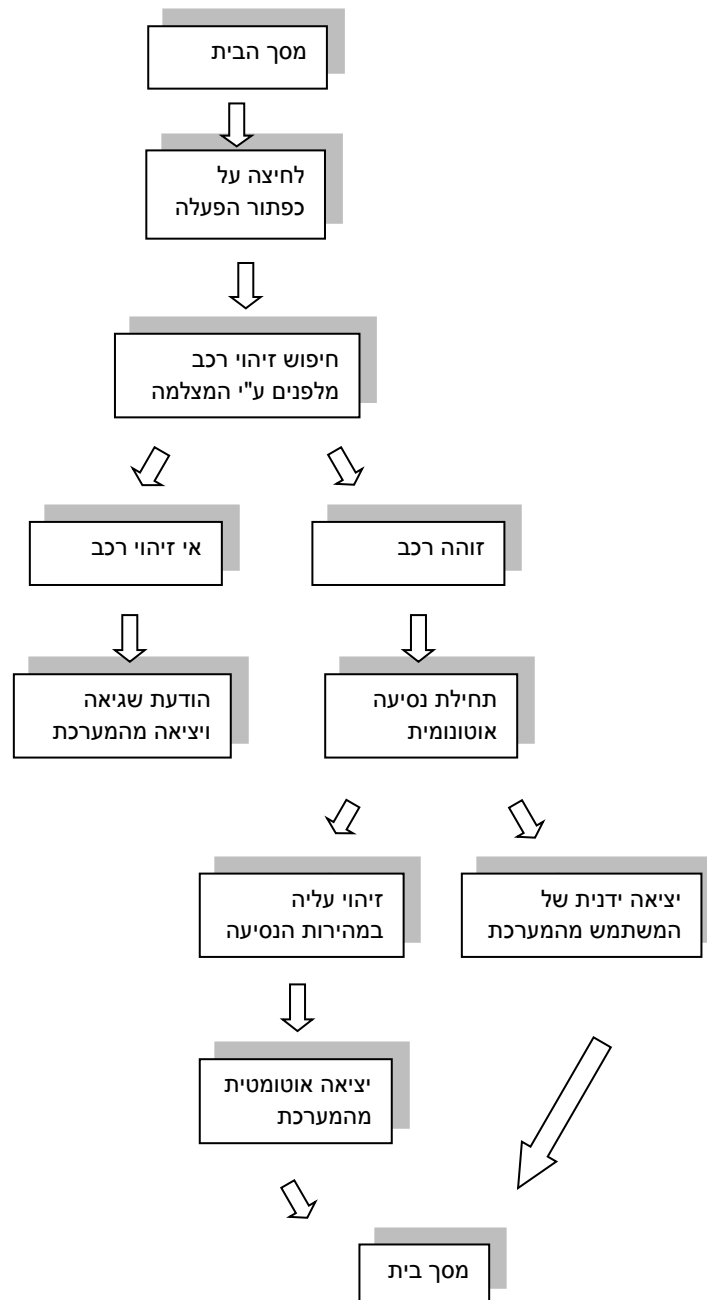
אם המערכת מזהה עלייה במהירות מעל סף מסוים היא תתריע על כך ותמליץ/תעבור לנסיעה ידנית.

#### 3.3 מטרות שאינן חלק מהפרויקט

המערכת אינה תפתור את התרחישים הבאים:

- לא פותרת בעיית עקיפה תוך כדי בדיקת פניות נתיב אחר.
- לא מזהה עצמים שנכנסים באופן מפתיע לנתיב הנסיעה (הולכי רגל, רכב שחותך וכו').

### 3.4 תרחישי שימוש



איור 1: תרחישי שימוש

### 3.5 אפיון מסכים

- מסך הבית.
- הפעלה תקינה (צילום בזמן אמת של הרכב מלפנים).
- מסך שגיאת זיהוי רכב.
- התראת יציאה מהעקיבה (שינוי מהירות וכו').
- מסך סיום.
- נסיעה אוטומטית.

### 3.6 נושאים פתוחים

- זיהוי מהירות נסיעה.
- זיהוי מרחק מדויק מהרכב שמלפנים.

#### 4. מבוא

מטרת העל של הפרוייקט היא בניית כלי רכב שיעקוב אחר רכב אחר.  
ניתן לפרק את זה לכמה יעדים:

החלק הראשון הינו בעיקרו עיבוד התמונה (Image processing):  
זיהוי כלי רכב במרחב, המרחק שלו מאיתנו, ומיקומו המדויק.

למטרה זו נעזרנו בספריית CV Open Source Computer Vision) היא חבילת תוכנה שנועדה לעזור לפתח יישומים של ראייה ממוחשבת. הספרייה מוכוונת בעיקר עבור יישומי ראייה ממוחשבת בזמן אמת.

בחלק השני, לאחר זיהוי מיקומו המדויק של הרכב מלפנים, צריך להחליט האם אנו מעוניינים לעקוב אחריו ובמידה וכן לאיזה כיוון בדיוק יש לפנות.

עיבוד הנתונים הללו נעשה בקוד C++

יעד מרכזי נוסף הוא שליטה על החומרה עצמה והממשק עם התוכנה:

קריאת הנתונים מהמצלמה, לאחר עיבוד הנתונים "דיבור" התוכנה עם הרכיבים הפיזיים, ולבסוף הנעה עצמה של הרכב במדויק לפי צרכינו.

השתמשנו במצלמת רשת שמחברת ישירות למחשב. המחשב מעביר את הפקודות לתנועה למיקו בקר (Microcontroller). מיקרובקר הוא מחשב קטן במעגל משולב יחיד המכיל את ליבת מעבד, זיכרון, וציוד היקפי לקלט / פלט הניתן לתכנות. מיקרובקרים מיועדים ליישומים משובצים. מיקרובקרים נמצאים בשימוש במרבית המוצרים החשמליים הסובבים אותנו. למיקאובקרים צריכת הספק נמוכה משמעותית.

המיקרובקר בעצמו היה מחובר לעוד מנהל התקן (driver) לשליטה על המנועים. ההתקן קיבל מהמיקרובקר 4 מתחים ועל פיהם הוא שלט בשתי מנועים. מנוע DC שזה מנוע שמקדם או עוצר את רכב. ושלט על מנוע מגנטי שביצע עבורינו את ההגוי.

## 5. סקר ספרות/מערכות

מכוניות ללא נהג:

ישנם כיום הרבה פרוייטי פיתוח של רכב ללא נהג, אך המוכר והמתקדם ביותר הינו פרוייקט הרכב של גוגל. פרוייקט זה מנוהל ומבוסס על פרוייקט של סבסטיאן ת'ראן, מנהל המחלקה לבנינה מלאכותית באוניברסיטת סטנפורד. הוא פיתח רכב שנבנה עבור המרוץ האתגר הגדול של דארפ"א (The DARPA Grand Challenge)

בתחרות ב-2005 היה צריך לעבור דרך של 212 ק"מ, מנהרות ופניות חדות. למקום הראשון (בזמן של 6:54 ש') הגיע סטנלי, הרכב של אוניברסיטת סטנפורד. רכב זה הוא הבסיס לרכב של גוגל.

### Google driving to be driverless

Google's modified Toyota Prius uses an array of sensors to navigate public roads without a human driver. Other components, not shown, include a GPS receiver and an inertial motion sensor.

#### Laser-guided mapping

A rotating sensor with lasers called a LIDAR on the roof scans more than 200 feet in all directions to generate a precise three-dimensional map of the car's surroundings.

#### Video camera

A camera mounted near the rear-view mirror detects traffic lights and helps the car's onboard computers recognize moving obstacles—such as pedestrians and bicyclists.

#### Position estimator

A sensor mounted on the left rear wheel measures small movements made by the car and helps to accurately locate its position on the map.

#### Radar

Four standard automotive radar sensors, three in front and one in the rear, help determine the positions of distant objects.

Source: Google

NEW YORK TIMES: PHOTOGRAPHS BY RAMIN RAHIMIAN FOR THE NEW YORK TIMES

איור 2: תיאור המיכשור על המכונית של גוגל

כמו שניתן לראות על רכב זה מותקנות מערכות רבות: LIDAR - העיקרית שבהם, היושבת על גג הרכב, היא מערכת לייזרים הממפה את הדרך ע"י יצירת תמונה תלת מיימדית. בנוסף ישנם 4 חיישני ראדר העוזרים לקבוע מיקום ומרחק עצמים זרים. חיישן מיקום ומצלמה לקריאת רמזורים ולמנעית פגיעה בהולכי רגל.

טכנולוגיה זו מאוד בשלה אך אם זאת מורכבת מאוד בפרוייקט זה חיפשנו פתרון הרבה יותר פשוט שאינו מצריך כמות רבה כל כך של חיישנים.

טכנולוגיה נפוצה נוספת למיפוי הסביבה היא טכנולוגיית ה- **Kinect**. הממפה את הסביבה באמצעות חיישני אינפרא אדום ושתי מצלמות

ה Kinect יכל להיות פתרון נהדר, אבל הבעיה העיקרית היא, שהמרחק שלו מוגבל לכ- 3.5 מטרים, דבר שמבטל את האפשרות להשתמש בו במעקב אחרי תנועת רכב המתקיים במרחקים גדולים יותר. בסופו של דבר החלטנו למפות את הסביבה באמצעות מצלמה בודדת. מטרה זו מצריכה שימוש רחב בעיבוד תמונה. חבילת תוכנה אשר מתמחה בנושא עיבוד התמונה היא ה- **OpenCV**. בזכות מאגר הפונקציות הרחב שהיא מספקת נוכל למפות את הסביבה ולזהות כלי רכב בחזית.

כשניגשנו לחומרה בחנו מספר טכנולוגיות. טכנולוגיית ה- **ROS**, טכנולוגיה מתפתחת של מערכות הפעלה רובוטיות. טכנולוגיה זו היכולה לשלוט בחיישנים ומנועים רבים מותאמת למחשבים מבוססי לינוקס אך עדיין אינה נמצאת בשימוש רחב. לעומתה המיקרו בקר - **Arduino** היה עבורינו פתרון נח בהרבה, מכיוון שלא נדרש בפרוייקט זה שליטה מורכבת במנועים רבים מדי ולבקר זה יש קהילת משתמשים רחבה המספקת פונקציות רבות ותמיכה פשוטה. בנוסף, השימוש במיקרו בקר קל מאוד ליישום.



## לקריאה נוספת:

- המרוץ של דרפ"א

<http://www.darpagrandchallenge.com/>

- הרכב סטנלי

<http://cs.stanford.edu/papers/thrun.stanley05.pdf>

- סבסטיאן ת'ראן מדבר על הרכב של גוגל

[http://www.ted.com/talks/sebastian\\_thrun\\_google\\_s\\_driverless\\_car.html](http://www.ted.com/talks/sebastian_thrun_google_s_driverless_car.html)

- ספר על ה OpenCV : Learning OpenCV

[http://books.google.de/books?id=seAgiOfu2EIC&pg=PA9&dq=openclv&hl=de&sa=X&ei=T2NNUu\\_CM4mY4wTWxoCwCg&ved=0CEUQ6AEwAg#v=onepage&q=openclv&f=false](http://books.google.de/books?id=seAgiOfu2EIC&pg=PA9&dq=openclv&hl=de&sa=X&ei=T2NNUu_CM4mY4wTWxoCwCg&ved=0CEUQ6AEwAg#v=onepage&q=openclv&f=false)

- ROS

<http://www.willowgarage.com/pages/software/ros-platform>

- Arduino

<http://www.arduino.cc/>

### מחקר ופיתוח המוצר

ראשית, לפני שניגשנו בכלל לחקור על הנושא ולבדוק איך מיישמים רעיון שכזה, היה לנו בראש מושג כללי של איך זה עובד. היה לנו היכרות עם הביטוי הזה "עיבוד תמונה" והיה לנו מושג קלוש איך זה עובד. ומכאן התחלנו לעבוד:

- ❖ כמובן בימינו כשאדם רוצה לקבל מידע על משהו, הדבר הראשון שהוא עושה אם יש לו מחשב ואינטרנט זה להיכנס לGoogle ולהתחיל לחפש. זה כמובן מה שעשינו. התחלנו להתעמק בנושא עיבוד התמונה ומאגר התיקיות שישנו באינטרנט (OpenCV).
- ❖ אחרי שספגנו מספיק ידע על הנושא הנ"ל התחלנו לראות עשרות סרטוני הדרכה באינטרנט של התקנה ועבודה עם ספריות עיבוד תמונה.
- ❖ לצורך עבודה עם הספריות התקנו את סביבת הפיתוח של Visual Studio 2012 – Microsoft אשר מצוינת לצורך פיתוח בשפת ++C שאיתה פיתחנו את המערכת.
- ❖ האינטגרציה של הספריות בתוך VS12 היה כאב ראש לא קטן רווי בתקלות ובהתקנות מחדש.
- ❖ במקביל לעבודה עם סביבת הפיתוח על המחשב עבדנו עם המחשב הזעיר Arduino Uno שאחראי על נידוד המכונות. ה-Arduino אחראי על כל הצד הפיסי של המערכת. הוא אחראי על נתינת פקודות שגורמות לרכב לנוע, אם ע"י הזזת הגלגלים לצורך סיבוב ואם לצורך האטה ע"י הפחתה במהירות הנסיעה.
- ❖ כמובן שכל הפקודות שהגיעו מה-Arduino נעשו לאחר קבלת נתונים מהמצלמה וניתוח/עיבוד עם הרבה בעיות של התמונה. היו הרבה טלטלות בעבודה עם live feed ממצלמת רשת.
- ❖ הפקודות שה-Arduino העביר למכונות הינן כמובן אלקטרוניות. בכדי להגיע לתוצאות הרצויות היינו צריכים לעבוד עם מעגל חשמלי שהרכבנו (עם עזרה מאדם או שניים) בכדי לשלוט בעדינות/דיוק הפקודות כמה שיותר.
- ❖ אחרי הרבה כאבי ראש בעבודה עם וידאו חי הצלחנו להגיע לתוצאות עיבוד תמונה יפות. ואז התחלנו לעבוד על השיפורים והגימורים של המוצר בעבודה עם שתי מכונות עד אשר הגענו לתוצאה שרצינו.

## 7. היישום

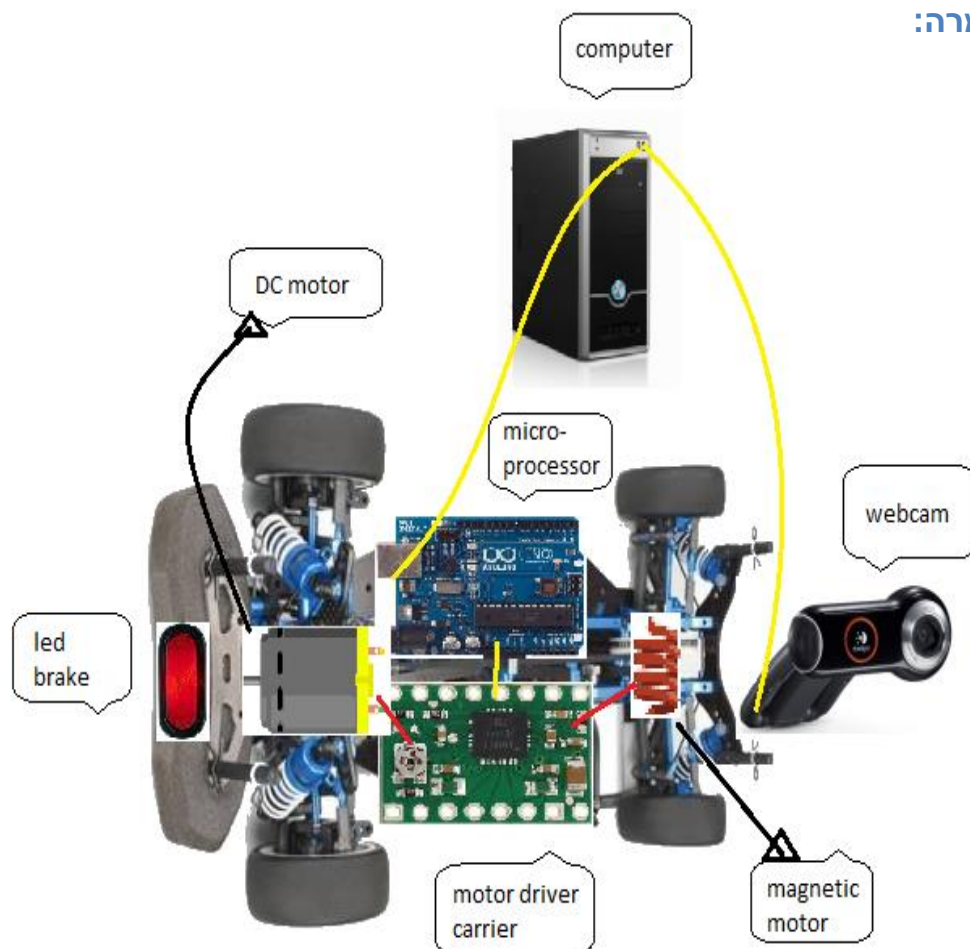
### 7.1 תאור כללי של היישום

החלטנו להשתמש במצלמת רשת פשוטה אשר מורכבת על חזית הרכב ומצלמת את הסביבה הקדמית. המצלמה מחוברת למערכת ממוחשבת שמעבדת את הנתונים, ממפה את הסביבה ומחליטה בהתאם לכך אילו פעולות הרכב נדרש לבצע. המערכת הממוחשבת הופכת את ההחלטות הללו לסדרת פקודות אותן היא שולחת למיקרו-בקר.

כעת המיקרו-בקר מתרגם את הפקודות שקיבל מהמערכת לאותות חשמליים שמועברים ל- Motor Driver אשר ע"י אותו אלו מפעיל שני מנועים:

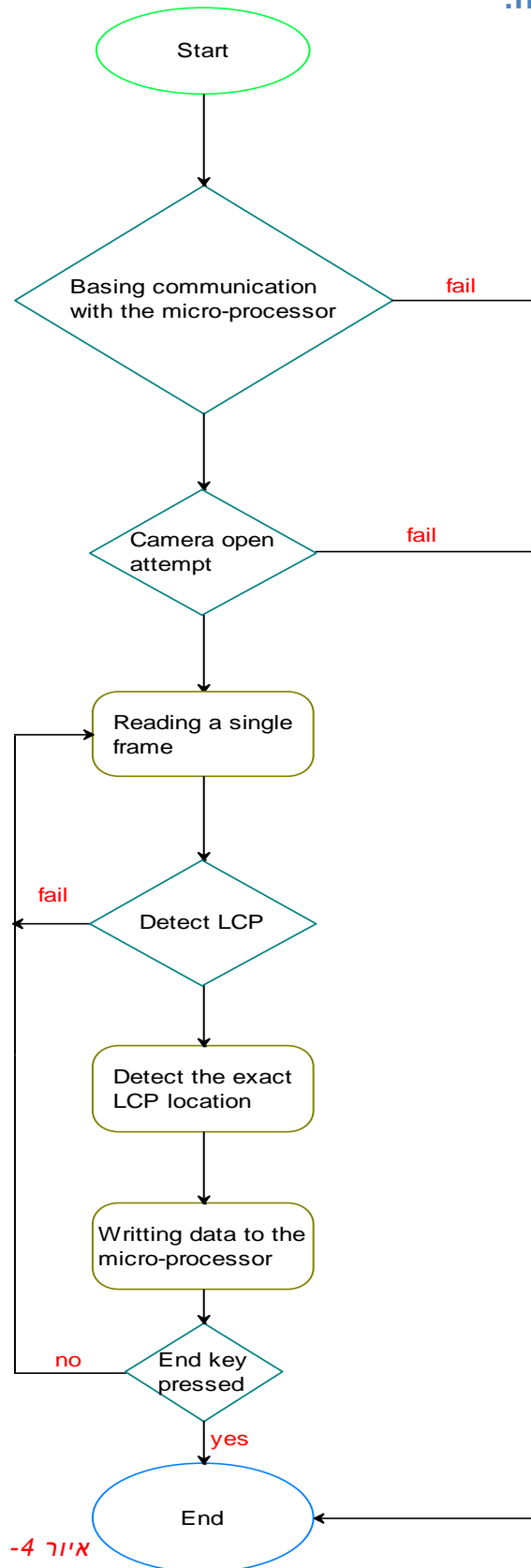
- המנוע האחורי – מנוע DC, אשר מבצע את פעולת הנעת הרכב (גז וברקס).
- המנוע הקדמי – מגנטי, אשר מדמה הגה ושלוט בכיוון התנועה.

### 7.2 מתאר החומרה:



איור 3- מתאר חומרה

### 7.3 תאור התוכנה:



איור 4- דיאגרמת מצבים: זיהוי לוחית

בתחילת התוכנית המערכת מנסה לבסס קשר עם המיקרו-בקר. במקרה של כישלון, המערכת תדפיס הודעת שגיאה ותצא מהתוכנית. עתה, שבוסס קשר בהצלחה, יש נסיון פתיחת המצלמה. גם כאן במקרה של כישלון, התוכנית תסתיים. לאחר פתיחת המצלמה, התוכנית נכנסת ללולאה קבועה (כל עוד לא נלחץ מקש יציאה), שבא נעשה קריאה של פריים בודד, זיהוי לוחית רישוי, זיהוי מיקומה המדויק ובמידת הצורך מועברת פקודה למיקרו-בקר.

## מילון נתונים (התוכנה במחשב):

### משתנים

משתנה	הסבר
Mat frame	מחזיק פריים בודד מהמצלמה
VideoCapture cap	מבנה (class) שמחזיק את הקישור למצלמה
vector<vector<Point> > squares	ווקטור שמחזיק את המלבנים שזוהו
HANDLE hDevice	מחזיק את הקישור לפורט (קישור למיקרופרוססור)
double edges[2]	אורך של זוג צלעות
vector<vector<Point> > contours;	מערך של צורות סגורות
Mat canny_output	מטריצה שמחזיקה את התמונה לאחר הפעלת אלגוריתם Canny
vector<Point> approx	ווקטור של נקודות המייצגות מצולע

### מתודות

חתימת הפונקציה	הסבר
void InitPort()	ביסוס קשר עם המיקרו בקר דרך פורט USB שמוגדר מראש. (כאן פורט 5)
static double angle( Point , Point ,Point )	מקבל 3 נקודות ומחזיר את קוסינוס הזווית שביניהן
static void drawSquares( Mat&, const vector<vector<Point> >& )	מקבל תמונה ומתאר של מלבן ומוסיף את המלבן במיקומו בתמונה
double *getEdges(vector<Point> )	מקבל 4 נקודות ומחזיר את בסיס וגובה המרובע
bool TestLCPRatio(vector<Point> )	מקבל 4 נקודות ובודק האם היחס בין הנקודות תואם ליחס לוחית הרישוי

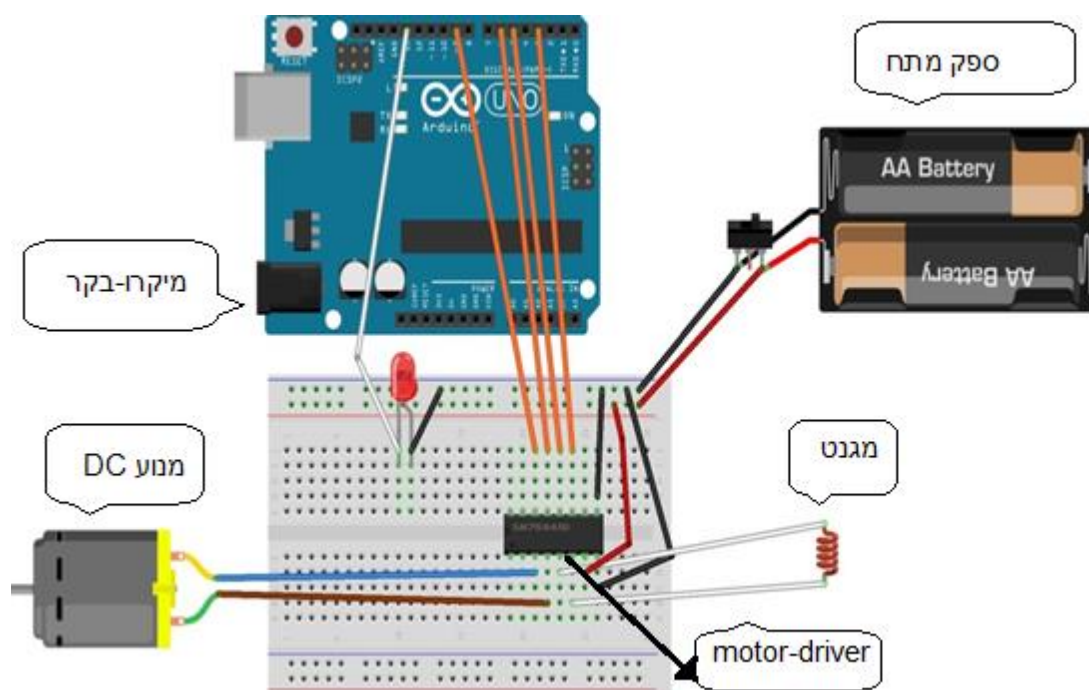
void FrameProcessing(Mat )	מקבל תמונה בודדת ומחפש מצולעים בכדי לבדוק התאמה ללוחית רישוי
void DetectLCP(vector<vector<Point> >)	מקבל מערך של מצולעים ובודק התאמה ללוחית רישוי
void CheckLCPLocation(vector<Point>)	מקבל נקודות של לוחית רישוי, בודק את מיקומם ולפיכך מעביר פקודות לתזוזת הרכב

## מתודות של OpenCV

חתימת הפונקציה	הסבר
void <b>cvtColor</b> (InputArray , OutputArray , int )	מקבל תמונה צבעונית ומחזיר תמונה בשחור לבן
void GaussianBlur(InputArray , OutputArray , Size , double , double)	מעביר את התמונה בפילטר גאוסיאני(ע"מ להפחית רעש)
void Canny(InputArray , OutputArray , double , double , int <b>apertureSize</b> =3)	מציאת קצוות של כל אובייקט בתמונה ע"פ אלגוריתם Canny
void <b>dilate</b> (InputArray , OutputArray , InputArray , Point <b>anchor</b> =Point(-1,-1))	חידוד התמונה ע"י פעולת הרחבה
imshow("edges", canny_output)	הצגת התמונה
void <b>findContours</b> (InputOutputArray , OutputArrayOfArrays , int , int , point <b>offset</b> =Point())	מציאת מתארים של מצולעים בתמונה בינארית
void approxPolyDP(InputArray , OutputArray , double , bool );	משווה בין מצולעים נתונים לפי רמת דיוק נתונה
double contourArea(InputArray )	מחזיר שטח של מצולע
bool isContourConvex(InputArray )	בודק האם במצולע נתון יש קימור
void polylines(Mat& , const Point** , const int* , int , bool , const Scalar& , int , int)	שרטוט מספר קווים במצולע

**\*\* הגדרות הפונקציות מבוסס על האתר <http://docs.opencv.org/>**

## 7.4 ארכיטקטורת המעגל



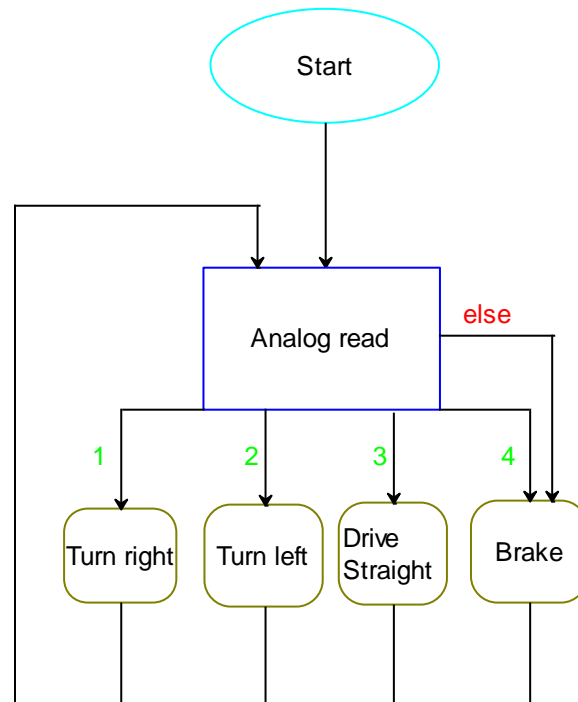
איור 5: ארכיטקטורת המעגל

המנועים ובקר התקן המנוע מחוברים בעזרת מטריצה לאותו ספק מתח. עליו יש מפסק פשוט.

בקר המנוע מחובר ב-4 כניסות למיקרו בקר. בנוסף קיימת נורה המשמשת כאור בלם.

## 7.5 מיקרו – בקר:

מטרתו לקבל פקודות מהמערכת הממוחשבת ולתרגם אותם לפקודות מעשיות למנועים (ע"י אותות חשמליים).



איור 6: דיאגרמת מצבים של תוכנת המיקרו בקר

בשלב ה-Analog read המערכת נמצאת בלולאה קבועה שבא היא קוראת קלט מהכניסה האנלוגית.

- קלט '1' - המערכת מופנת למצב של פניה ימינה.
- קלט '2' - המערכת מופנת למצב של פניה שמאלה.
- קלט '3' - המערכת מופנת למצב של נסיעה ישרה.
- קלט '4' - המערכת מופנת למצב של קרבה לרכב שמלפנים והאטת הרכב עד כדי בלימה.
- כל קלט אחר - לצורכי בטיחות המערכת מופנת למצב '4'.



## מילון נתונים (מיקרו-בקר):

### משתנים:

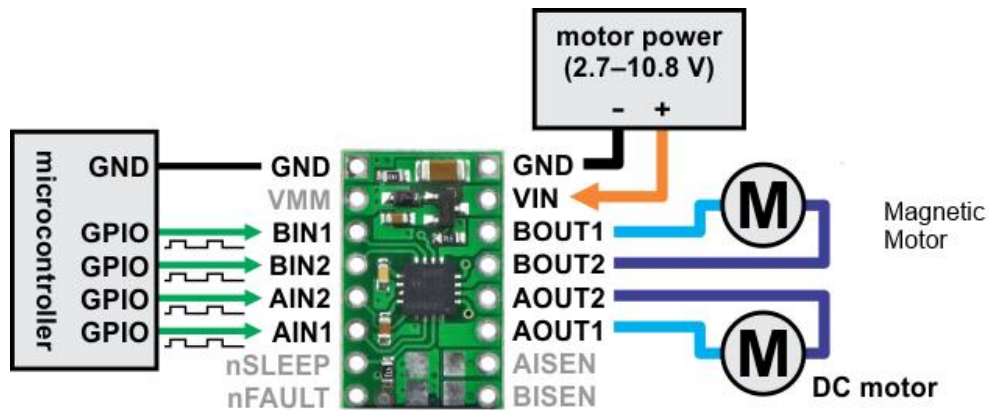
משתנה	הסבר
int DCMotor1	מספר הפין השולט במנוע DC
int DCMotor2	מספר הפין השולט במנוע DC
int leftWheel	מספר הפין השולט בגלגל השמאלי
int rightWheel	מספר הפין השולט בגלגל הימני
int breaksLed	מספר הפין השולט בLED הבלם
char input	קלט הפקודה מהמחשב
int prevInput	קלט הפקודה הקודמת מהמחשב

### מתודות:

מתודה	הסבר
Void setup()	אתחול המשתנים
Void loop()	לולאה אינסופית שקוראת נתונים ופועלת על פיהם
Void breaks()	הפעלת הבלם

## 7.6 מנהל התקן המנועים (motor driver carrier):

מקבל אותות חשמליים מהמיקרו-בקר ועל פיהם שולט במנועים עצמם.



איור 6: מעגל motor driver carrier

**\*\* האזור נלקח מהאתר <http://www.pololu.com/catalog/product/2130>**

AIN2, AIN1 אילו פני הכניסה השולטים בפלט של AOUT1, AOUT2 בהתאמה; באופן דומה, BIN1, BIN2, פני הכניסה השולטים בפלט של BOUT1 וBOUT2.

**טבלת האמת של ההתקן:**

xIN1	xIN2	xOUT1	xOUT2	FUNCTION
0	0	Z	z	Coast/Fast decay
0	1	L	H	Reverse
1	0	H	L	Forward
1	1	H	H	Brake/Slow decay

**\*\* הטבלה נלקחה מה DataSheet של הרכיב.**

במילים אחרות, כשרצינו שהרכב יתקדם, העברנו מתח גבוה לכניסה AIN2 (שחובר ל pin6 במיקרו בקר), ונמוך לכניסה AIN1 (שחובר ל pin9 במיקרו בקר). כשרצינו לבלום העברנו מתח נמוך בשתי הכניסות.

בכדי לשלוט על ההגה העברנו מתח גבוה או נמוך בכניסות BIN, שחוברו למגנט. ולפי כיוון הזרם השתנו קטבי המגנט. ומכיוון שהגלגלים חוברו למגנט הם פנו בהתאם לקטבי המגנט.

## 8. בדיקות, תוצאות והערכה

### 8.1 תוצאות

הותקנה בהצלחה סביבת הפיתוח – OpenCV והרצתה בעזרת Visual Studio 2012.

בנוסף התוכנה מותאמת לעבוד בו זמנית עם כמה התקני I/O: המצלמה והמיקרו-בקר.

התוכנה עצמה מזהה לוחיות זיהוי באחוז גבוה מזמן הריצה. אך עם זאת, בתנאים של תזוזות חדות ומהירות, דבר שגורם לתמונה להטשטש. במצב זה המערכת לא זיהתה באופן עיקבי את לוחית הרישוי.

לאחר זיהוי מוצלח של לוחית הרישוי, המערכת זיהתה וניתחה את מיקומה המדויק של לוחית הרישוי במרחב.

בעקבות ניתוח הנתונים, המערכת מצליחה לתקשר עם המיקרו בקר ולהעביר לו פקודות ברורות.

בד בבד, תוכנת המיקרו-בקר מצליחה לקבל נתונים מהמערכת הממוחשבת. כעת נעשה ניתוח של נתונים אילו ובהתאם מועברים סיגנלים חשמליים ל-Driver Controller.

בסופו של דבר, בעזרת סיגנלים חשמליים אילו, מניע ה-Driver Controller את הרכב בכיוון הנדרש.

## 8.2 הערכה

אב טיפוס של המערכת הורכב על גבי מכונת צעצוע אלקטרונית, שמטרתה הייתה לעקוב אחרי מכונת זהה שחובר לה לוחית רישוי. הרכב זיהה את הרכב שמלפניו ע"י זיהוי לוחית הרישוי ועקב אחריו באופן אוטונומי.

תוצאות הניסוי באב טיפוס הראו שאכן נעשה מעקב אוטונומי, אך זה נעשה לא תמיד באופן רציף ולעיתים למערכת היה צורך בכמה תיקונים/תמרונים בכדי להגיע למצב של עקיבה מדויקת.

## 9. דיון: מסקנות ופיתוחים עתידיים

בפרק התוצאות עלו שלוש נקודות עיקריות לשיפור המערכת:

- **זיהוי חלקי של לוחית הרישוי בתנאי תנועה מורכבים** – מצלמה פשוטה בעת נסיעה מהירה ו/או בעלת שינויי כיוון חדים, לא מסוגלת להתמקד ומספקת תמונות מטושטשות אשר מקשות על המערכת בזיהוי לוחית הרישוי.  
לצורך שיפור נקודה זו, יעשה שימוש במצלמה ברמה גבוהה יותר אשר תצליח להנפיק תמונות חדות יותר גם במצבים של שינויים מהירים. פתרון אחר אפשרי הוא נסיון זיהוי ועקיבה אחרי גוף הרכב עצמו, ולא בלוחית הרישוי.
- **תנועה לא חלקה/רציפה של הרכב** – ברגעים שהמערכת לא זיהתה לוחית רישוי, גם אם זה היה לזמן קצר מאוד, המערכת מיד בלמה, דבר אשר יצר נסיעה במקטעים ולא רציפה.  
לצורך שיפור נקודה זו, נדרשת שליטה בקצב ההתקדמות (מהירות הנסיעה) ולא רק בכיוונה, ע"י שליטה בעוצמת זרמי החשמל שמופנים למנוע ה-DC. בנוסף, בכדי למנוע או להמעיט את הנסיעה הלא רציפה ניתן להפעיל טיימר אשר יגרום לעצירה רק במידה ולא זוהתה לוחית רישוי לאחר פרק זמן מוגדר.
- **תמרון גס של גלגלי הרכב הקדמיים** – בכדי להגיע למצב שהרכב עומד בצורה ישרה מאחורי הרכב שמלפנים נעשה שימוש בהרבה תמרונים קטנים ומיותרים. זאת מכיוון שלצורך כיוון הגלגלים נעשה שימוש במגנט אשר לו שתי אפשרויות בלבד של ימינה או שמאלה. למרות שהמערכת יכלה לחשב זווית מדויקת של הפניה הרצויה, החומרה עצמה לא יכלה לפעול בהתאם לרגישות הרצויה.

לצורך שיפור נקודה זו, ישנה אפשרות של שימוש במנוע סרבו אשר נע בזוויות ולא באופן בינארי (ימינה או שמאלה) וע"י כך המערכת תוכל לשלוט במדויק על כיוון הנסיעה.

## מסקנות

במהלך עבודה על הפרוייקט נוכחנו לדעת כמה נושא זה (רכב אוטונומי) מפותח ומעסיק חברות רבות והמעבר לרכב עצמאי יגיע מהר משאנו משארים. אך עם זאת, פיתוח רכב אוטונומי הוא נושא יקר מאוד ומורכב מבחינות של רישוי והטמעה. אז עד שנגיע לנקודה שהרכבים ינהגו באופן עצמאי לחלוטין יש מקום להטמעת פתרונות ביניים כדוגמת פרוייקט זה.

הפרוייקט הראה שניתן להגיע למעקב אחר רכב אחר באמצעות מערכת פשוטה יחסית הכוללת מצלמה בלבד. אך כדי ליישם מערכת זו ב "עולם האמיתי" יש צורך לתרגם את התוכנה ולהתאימה למחשב הרכב. בנוסף יש צורך בפיתוח יישומים נוספים וחשובים כגון: זיהוי עצמים מתפרצים (הולכי רגל, רכב סוטה וכו'), בידוד הרכב הנעקב משאר הרכבים.

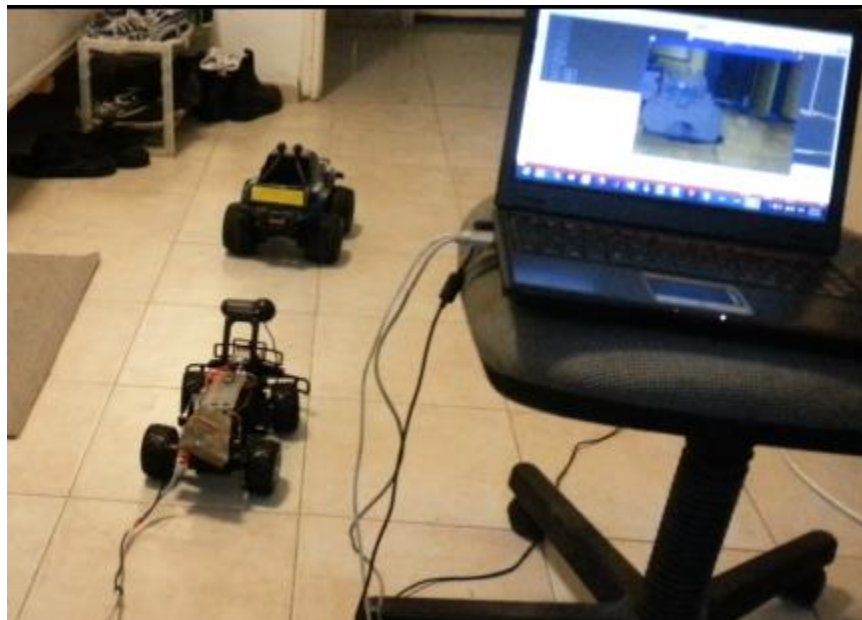
## 10. מקורות

- [www.arduino.cc/](http://www.arduino.cc/)
- [books.google.de](http://books.google.de)
- [www.robotics.usc.edu](http://www.robotics.usc.edu)
- [www.darpa-grandchallenge.com](http://www.darpa-grandchallenge.com)
- [www.opencv.org](http://www.opencv.org)
- [www.pololu.com](http://www.pololu.com)
- [www.ted.com](http://www.ted.com)
- [www.willowgarage.com](http://www.willowgarage.com)
- [www.wikipedia.com](http://www.wikipedia.com)
- [www.youtube.com](http://www.youtube.com)

**\*\* כל האתרים מעודכנים לתאריך 03/10/13**

## 11. נספחים

### דוגמה לריצת מבחן



<http://www.youtube.com/watch?v=NTTrbnIL-5Vg&feature=youtu.be>