

# המחלקה להנדסת תוכנה

## פרויקט גמר – תשע"ו

### זיהוי והימנעות ממכשולים Obstacles Avoidance

בן נקש

מנחה אקדמי: מר שי תבור      אישור: תאריך:  
אחראי תעשייתי: מר טל יצחק      אישור: תאריך:  
רכז הפרויקטים: ד"ר אללוף מרים,  
וד"ר יגל ראובן      אישור: תאריך:

#	מערכת	
1	מאגר קוד	<a href="https://github.com/ben-nakash/Drones-Obstacle-Avoidance-System">https://github.com/ben-nakash/Drones-Obstacle-Avoidance-System</a>
2	יומן	<a href="https://trello.com/b/AuUIDKir/obstacles-detection-avoidance-project">https://trello.com/b/AuUIDKir/obstacles-detection-avoidance-project</a>
3	ניהול פרויקט	Github
4	הפצה	
5	סרטון אב-טיפוס	<a href="https://drive.google.com/file/d/0B-1pi9mMfnD8WmdpMml4Q1RNblE/view?usp=sharing">https://drive.google.com/file/d/0B-1pi9mMfnD8WmdpMml4Q1RNblE/view?usp=sharing</a>

## תוכן עניינים

1.	מבוא	3
2.	תיאור הבעיה	4
4.	דרישות ואפיון הבעיה	4
5.	הבעיה מבחינת הנדסת תוכנה	5
3.	תיאור הפתרון	6
6.	מהי המערכת	6
7.	תיאור הפתרון המוצע	7
9.	תיאור הכלים המשמשים לפתרון	9
4.	תוכנית בדיקות	10
5.	סקירת עבודות דומות בספרות והשוואה	11
6.	נספחים	12
א.	ביבליוגרפיה	12
ב.	תרשימים וטבלאות	13
ג.	תכנון הפרויקט	14
ד.	טבלת סיכונים	15
ה.	טבלת דרישות	16

## 1. מבוא

חברת הסטארט-אפ Airscort מפתחת אפליקציות פורצות דרך בתחום טכנולוגיית הרחפנים בתעשייה. החברה הוקמה ע"י קבוצת מהנדסי מכונות, תוכנה וחשמל שרואים עתיד לטכנולוגיית הרחפנים.

הטכנולוגיה אותה מפתחים ב-Airscort תאפשר מגוון רחב של יישומי רחפנים אזרחיים, ביניהם:

- ❖ ספורט אתגרי - צילום הספורטאי בזמן אמת בצורה קלה, נוחה וללא צורך בגורם נוסף לניווט הרחפן.
- ❖ אבטחה – ביצוע סיורים, התבוננות מלמעלה וסיוע באבטחה שוטפת.
- ❖ בידור – ביצוע מגוון פעילויות מהנות בעזרת הרחפן כמו סיורים.
- ❖ משלוחים – משלוח מהיר ופשוט של מוצרים.
- ❖ חקלאות - ביצוע מעקב אחר גידולים חקלאיים ומתן מידע שימושי למשתמש.
- ❖ התאמה אישית – לצורך האישי של בעל עסק.

בעבר, רחפנים שימשו בעיקר לצרכים צבאיים, אך בשנים האחרונות גוברת מגמת השימוש בהם, וישנו ביקוש רב יותר בתעשייה האזרחית ובתעשייה העסקית.

חברת Airscort מעוניינת לאפשר לרחפן לטוס אוטונומית וללא התערבות אדם. על כן, עליו לזהות מכשולים הנקרים בדרכו ולהימנע מהם עצמאית.

## 2. תיאור הבעיה

נרצה שרחפן יהיה מסוגל להחליף את האדם בביצוע פעולות שונות (או לאפשר פעולות שהאדם לבדו אינו מסוגל לבצע), על מנת לאפשר לאותו אדם להתפנות לעיסוקים האחרים. לדוגמה, החקלאי נדרש להגיע לשדה שלו ולבדוק מה מצבו באופן אישי – דבר המבזבז זמן רב. לכן, נדרוש שהרחפן יעבוד באופן אוטונומי וללא התערבות של גורם חיצוני. בצורה זו, אותו אדם יוכל לשלוח את הרחפן לבדוק את מצב השדה החקלאי, כאשר בו בזמן יוכל להמשיך בעיסוקיו השונים. דוגמה נוספת הינה כאשר אדם מבצע ספורט אתגרי ומעוניין לתעד את עצמו. כיום, אדם אינו יכול לעשות זאת תוך כדי הפעילות בה הוא עוסק ונדרש אדם נוסף שיצלם אותו. לכן, נרצה שהרחפן יתעד את האדם ללא צורך בגורם נוסף שישלוט בו.

לכן, מטרת הפרויקט היא שיפור יכולת הטיסה האוטונומית של הרחפן ע"י זיהוי והימנעות ממכשולים הנקרים בדרכו בזמן אמת באמצעות חיישנים שיורכבו עליו. כיוון שהרחפן טס גבוה באוויר, אצא מנקודת הנחה שהמכשולים שעלולים להיות בדרכו הינם מכשולים סטטיים – כלומר, מכשולים שאינם מסוגלים לזוז.

## דרישות ואפיון הבעיה

נתבונן כעת מנקודת המבט של אדם אשר ישתמש ברחפן ובו המערכת שאפתח. הדרישות מן הרחפן והתוכנה יהיו כדלקמן:

- ❖ נרצה שלא יהיה צורך לכוון את הרחפן מחדש בכל פעם שיש מכשול כלשהו בנתיבו (כיול).
- ❖ הרחפן יזהה מכשולים הנקרים בדרכו ממרחק אשר ייתן לו מספיק זמן להתחמק (וכמובן עוד זמן קצר מעבר לכך כרשת ביטחון).
- ❖ הרחפן יתחמק מהמכשולים אשר זיהה בעזרת אלגוריתם מתוחכם שיזיז אותו לנתיב בטוח.

## הבעיה מבחינת הנדסת תוכנה

בפרויקט זה אעבוד מול מחשב לוח בשם Raspberry Pi המריץ מערכת הפעלה מלאה מבוססת Linux. אותו מחשב יקלוט נתונים מחיישני מרחק שיחוברו אליו, וישלח פקודות הזזה לבקר ה-Pixhawk כדי להימנע ממכשול שהתגלה בדרכו.

ישנן מספר בעיות תכנותיות אשר אאלץ לפתור:

א. קליטת נתונים בזמן אמת לצורך זיהוי מכשולים מחיישני המרחק שיורכבו על גבי הרחפן ויחוברו ללוח ה-Raspberry Pi.

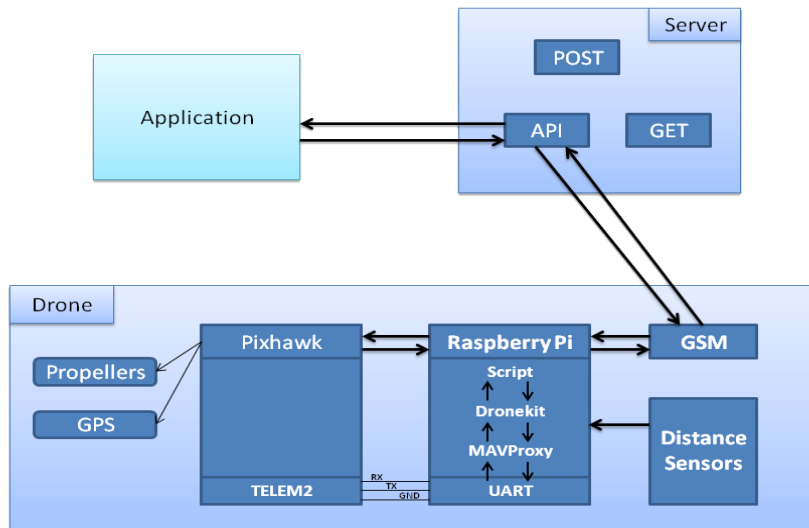
ב. הימנעות מפגיעה במכשול אשר התגלה בדרכו של הרחפן – עליו לפתח סקריפט בשפת Python אשר ישלח פקודות הזזה לרחפן על מנת להימנע מן המכשול בעזרת אלגוריתם מתוחכם.

בשוק כבר קיימים אלגוריתמים לזיהוי מכשולים, אך הם עושים שימוש בטכנולוגיה מתקדמת ויקרה שאינה במסגרת התקציב של החברה. לכן, עליי לפתח ולהתאים אלגוריתם אשר יאפשר את זיהוי מכשולים במסגרת התקציב והחומרה המוגבלים שברשותי.

### 3. תיאור הפתרון

#### מהי המערכת

התרשים הבא מתאר את ארכיטקטורת המערכת:



❖ **Raspberry Pi:** מחשב לוח זעיר שעליו נוכל להריץ סקריפטים ולשלוח פקודות לבקר ב-Pixhawk. אליו אכניס את הסקריפט שאפתח.

❖ **בקר ה-Pixhawk:** מערכת טייס אוטומטי המבוססת על קוד פתוח שפותחה ע"י חברת 3D Robotics. בבקר זה משולבים טכנולוגיית חיישנים ומעבד מתקדמים, מערכת הפעלה בזמן אמת המספקת ביצועים מרשימים, גמישות ושליטה לכלים אוטונומיים.

❖ **חיישני מרחק:** חיישני מרחק יורכבו על הרחפן ויסרקו את השטח שמולם באופן תמידי. נתוני הסריקה ישלחו אל מעבד ה-Raspberry Pi בזמן אמת וינותחו ע"י אלגוריתם שאפתח ונצרוב אל השבב שלו. בחרתי להשתמש בחיישנים מבוססי לייזר, וזאת עקב כמה סיבות:

- טווח ארוך – עד 40 מטר (בניסויים עד כה התברר שהוא קולט אף יותר).
- משקל נמוך – כ-16 גרם.
- צריכת חשמל נמוכה מאוד.
- אפשרות לחבר כמה חיישנים מאותו סוג במקביל.
- מחיר לא גבוה מידי.

❖ **תוכנה לזיהוי והימנעות ממכשולים:** אפתח Script בשפת Python אשר יעבוד על גבי מחשב ה-Raspberry Pi ויקבל כקלט את נתוני המרחק מהחיישנים. בהתאם לקלט ובעזרת אלגוריתם מתקדם הוא ישלח פקודות הזזה לרחפן על מנת להימנע מהמכשולים שזוהו.

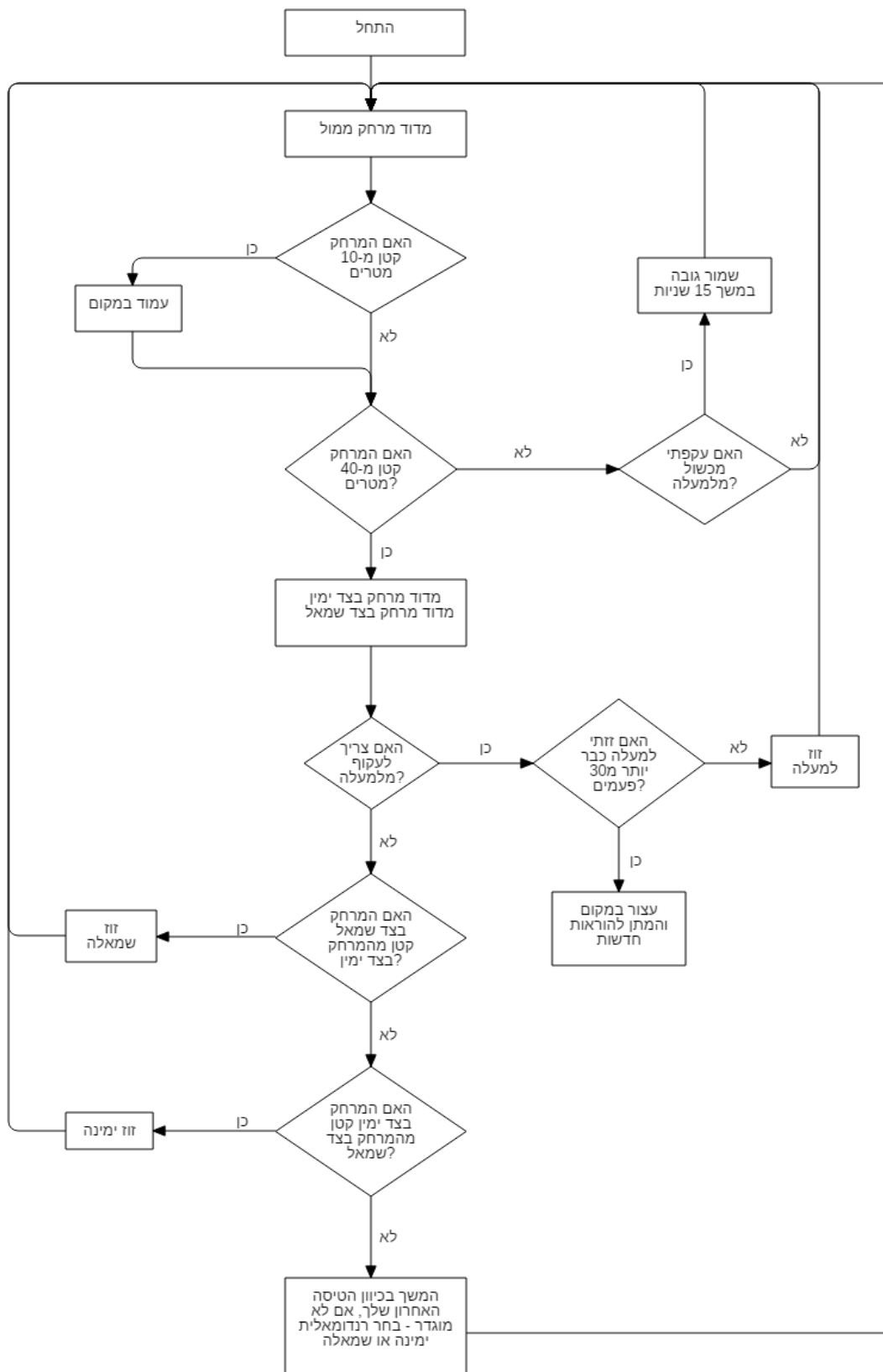
## תיאור הפתרון המוצע:

כאשר התחלתי לעבוד על הפרויקט, התכנון היה לעבוד מול בקר הטיסה של הרחפן שהוא מסוג Pixhawk. לאחר מחקר, ניסיונות רבים והתייעצויות עם גורמים נוספים הוחלט שלא לשכתב את הקוד שבתוך הבקר, כיוון שלמרות שמדובר בקוד פתוח בשפות C++ ו-C, זו משימה קשה מאוד ולא מומלצת. מסיבה זו (ועוד כמה סיבות אחרות שלא רלוונטיות ספציפית לפרויקט זה), הוחלט להיעזר במעבד חיצוני אשר יחובר לבקר ה-Pixhawk והוא ישלח פקודות הזזה לבקר כאשר יאתר מכשול. תחילה, ניסינו לחבר את הבקר למעבד Intel Edison אשר יושב על גבי לוח ארדואינו ייעודי. מעבד זה מריץ מערכת הפעלה מבוססת Linux באופן עצמאי. כיוון שהוא פועל בצורה זו, אוכל להעלות אליו תוכנית אשר תשלח את פקודות ההזזה לבקר ה-Pixhawk כפי שרציתי. עקב בעיות התממשקות בין השניים הוחלט להשתמש במחשב זעיר אחר בשם Raspberry Pi. מחשב זה עובד בדיוק באותו עיקרון כפי שעובד ה-Intel Edison וגם הוא מריץ מערכת הפעלה מסוג Linux. עם לוח זה הצלחנו להתממשק מול הבקר ולבצע תקשורת נתונים תקינה ביניהם (שליחה וקבלה של מידע). לכן, ע"פ המצב הנוכחי אממשי את כתיבת האלגוריתם ע"י כתיבת סקריפט בשפת Python אשר ירוץ על גבי מחשב ה-Raspberry Pi. המחשב יקבל נתונים מן החיישן שיחובר ישירות אליו, והוא ישלח פקודות הזזה לרחפן באמצעות הסקריפט שאכתוב. חשוב לציין שאת פקודות ההזזה עצמן לא אני אפתח, כיוון שזה פרויקט בפני עצמו (עקב הצורך להתממשקות מול הבקר וידע נרחב בפונקציונאליות שה-Pixhawk מספק).

## שלי העבודה במהלך הפרויקט:

- א. חיבור החיישן ללוח ארדואינו תוך כדי כתיבת קוד פשוט לבדיקת תקינות הקלט של החיישן.
- ב. חיבור החיישן למחשב ה-Raspberry Pi ובדיקה נוספת של החיישן לתקינות וקבלת נתונים באופן תקין.
- ג. פיתוח האלגוריתם ראשוני לזיהוי המכשולים והבנת מיקומם ביחס לרחפן.
- ד. סימולציות ראשוניות, הרצת האלגוריתם ובדיקתו לזיהוי תקין של מכשולים.
- ה. שילוב פקודות הזזה לרחפן באלגוריתם בהתאם למיקום המכשול שזוהה.
- ו. בדיקת האלגוריתם בזמן אמת תוך כדי הפעלת הרחפן ע"י הזזת הרחפן לעבר מכשול.
- ז. סימולציות, והרצות נוספות תוך כדי תיקון באגים ושדרוג הקוד בהתאם למסגרת הזמן.

**תרשים זרימה כללי של האלגוריתם (עוד עתיד להשתנות):**





## תיאור הכלים המשמשים לפתרון

### **Mission Planner** ❖

בעזרת תוכנה זו נוכל לבצע מגוון פעולות עם הרחפן, ביניהן:

- מתן נקודות ציון ומסלול מעוף לרחפן בעזרת Google Maps (תכנון משימות טיסה).
- הורדת קבצי Log של המשימה וניתוחם.
- קבלת מידע רב אודות הרחפן – מיקום, גובה, מהירות, זווית, נתיב טיסה ועוד.

### **Arduino Software (IDE)** ❖

סביבת העבודה של Arduino. נוכל להיעזר בה על מנת לשאוב את הנתונים הנמדדים מהחיישנים וביצוע ניסויים שונים בקלות יחסית (ישמש בעיקר בתחילת העבודה עם החיישנים ולבדיקתם).

### **:Notepad++** ❖

עורך טקסט המותאם למתכנתים וניתן באמצעותו לכתוב קוד תוכנה במגוון גדול של תוכנות. בעזרתו אממש את האלגוריתם להימנעות ממכשולים כ-Python Script.

### **StarUML** ❖

תוכנה ליצירת ועריכת דיאגרמות מסוגים שונים, ביניהם תרשימי זרימה ותרשים מחלקות.

### **מערכת הפעלה** ❖

בפרויקט זה אוכל לעבוד גם בסביבת Windows וגם בסביבת Unix. ככל הנראה אשתמש בשתייהן לאורך הפרויקט בהתאם לצרכים של הפרויקט והנוחות שכל אחת ממערכות ההפעלה מספקות.

## 4. תוכנית בדיקות

לאורך פיתוח הפרויקט אבצע כמה וכמה בדיקות:

1. לאחר קבלת החייושן, אחבר אותו ללוח ארדואינו על מנת לבצע בדיקה פשוטה לקבלת קלט ולתפקוד תקין של החייושן.
2. בדיקת החייושן לדיוק וסטייה – אבצע מספר מדידות במרחקים שונים מן החייושן על מנת לבדוק שהוא אכן מחזיר תוצאות מהימנות ושרמת הסטייה אכן זניחה (כפי שטוען היצרן).
3. מדידת התפשטות הקרן – על אף שמדובר בחייושן לייזר, גם לייזר מתפשט על פני שטח מסוים כאשר מדובר במרחק גדול דיו. ארצה לבדוק את מידת התפשטות הקרן על מנת שאוכל להעריך מה רוחבו של השטח הנסרק במרחק של כ-40 מטרים.
4. בדיקה נוספת לתקינות לאחר חיבור החייושן למחשב ה-Raspberry Pi.
5. ביצוע בדיקות רבות לאלגוריתם ההימנעות ממכשולים לאורך הפיתוח. הבדיקות יכללו:
  - a. הרצות יבשות על מנת לוודא שהפקודות הנשלחות לבקר תואמות את התנאים שהצבתי.
  - b. סימולציות חיות עם הרחפן לאחר שנעשו מספיק בדיקות יבשות לתקינות האלגוריתם.

## 5. סקירת עבודות דומות בספרות והשוואה

### ❖ **Radiation Mapping UAV**

רחפן זה נועד לסרוק, למפות ולנתר קרינה באתרי גרעין. הוא עושה גם שימוש בטכנולוגיית הימנעות ממכשולים המתבצעת באמצעות חיישן מרחק מאותו סוג בו אני משתמש בפרויקט זה.

<https://github.com/plusangel/radiationUAV>

### ❖ **Autonomous plane with obstacle detection & avoidance system using a camera.**

רחפן עם מערכת לזיהוי והימנעות ממכשולים אשר מתבססת על מצלמה. אומנם השימוש הוא במצלמה וטכנולוגיה יקרה יותר, אך אלגוריתם ההימנעות עובד מצוין ומאפשר לרחפן לנוע במהירות של 30 מייל לשעה מבלי להיפגע (שזו בערך המהירות הדרושה לרחפן עליו אני עובד). בנוסף, הוא עובד בגישה אשר הוא מקבל יעד, ומתחמק מהמכשול תוך כדי המשך טיסה ליעד – אותה גישה שאני זקוק לה.

<https://github.com/andybarry/flight>

### ❖ **Quadcopter obstacle avoidance system using Pixhawk and a combination of ultrasonic & Infrared sensors (video and article)**

רחפן אשר מיישם גם אלגוריתם להימנעות ממכשולים באמצעות חיישנים אולטרא סוניים. בפרויקט זה יושם האלגוריתם בצורה כזו שאך ורק מתבצעת הימנעות – כלומר אין התחשבות ביעד מטרה אליו הוא צריך להגיע.

- <https://www.youtube.com/watch?v=hD6lgkcLfyl>
- [https://www.researchgate.net/publication/284724052\\_Relative\\_position-based\\_collision\\_avoidance\\_system\\_for\\_swarming\\_UAVS\\_using\\_multi-sensor\\_fusion](https://www.researchgate.net/publication/284724052_Relative_position-based_collision_avoidance_system_for_swarming_UAVS_using_multi-sensor_fusion)

### ❖ **Flying Fast and Low among Obstacles: Methodology and Experiments.**

רחפן אשר מסוגל לטוס מהר ובגובה נמוך ובו בזמן להימנע ממכשולים. ישנה עבודה תיאורטית רבה במאמר זה הכוללת חישובים רבים, אשר אוכל במידת הצורך להיעזר בה ליישום האלגוריתם.

[https://www.ri.cmu.edu/pub\\_files/pub4/scherer\\_sebastian\\_2008\\_1/scherer\\_sebastian\\_2008\\_1.pdf](https://www.ri.cmu.edu/pub_files/pub4/scherer_sebastian_2008_1/scherer_sebastian_2008_1.pdf)

## 6. נספחים

### א. ביבליוגרפיה

#### Sensors:

- ❖ **Exploring the different methods of objects detection**  
<https://phidgets.wordpress.com/2014/05/23/exploring-the-many-methods-of-object-detection/>
- ❖ **Lidar-Lite 2 Laser Rangefinder Specs**  
<http://www.robotshop.com/en/lidar-lite-2-laser-rangefinder-pulsedlight.html>
- ❖ **Lidar-Lite 2 Arduino Library**  
[https://github.com/PulsedLight3D/LIDARLite\\_v2\\_Arduino\\_Library#continuous-mode-wiring](https://github.com/PulsedLight3D/LIDARLite_v2_Arduino_Library#continuous-mode-wiring)

#### Intel Edison

- ❖ **Edison as a Companion Computer**  
<http://dev.ardupilot.com/wiki/edison-for-drones/>
- ❖ **Loading Debian (Ubilinux) on the Edison**  
<https://learn.sparkfun.com/tutorials/loading-debian-ubilinux-on-the-edison#enable-wifi>

#### Pixhawk

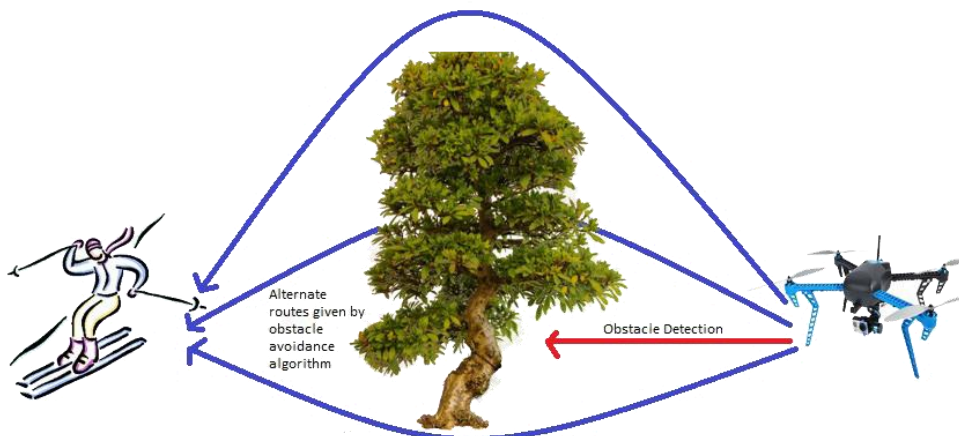
- ❖ **PX4 Github library**  
<https://github.com/PX4/Devguide>
- ❖ **Connecting a Companion Computer**  
[https://pixhawk.org/dev/companion\\_link](https://pixhawk.org/dev/companion_link)
- ❖ **PX4 Developer Guide**  
<http://dev.px4.io/>

#### Others

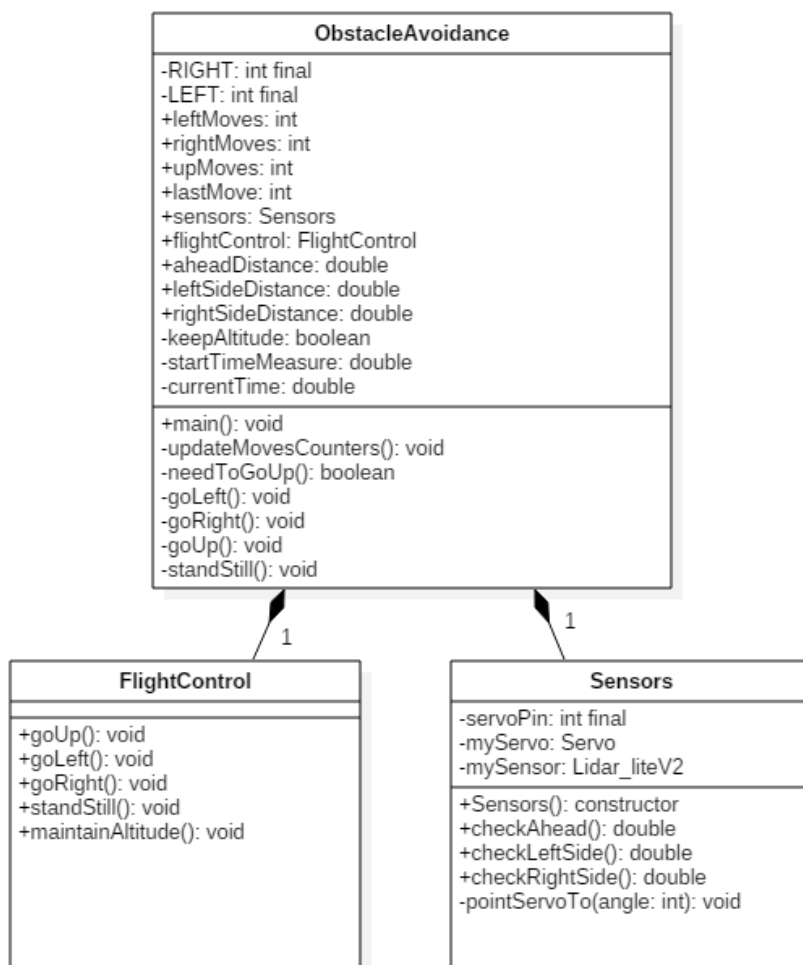
- ❖ **DIY Drones forum**  
<http://diydrones.com/>
- ❖ **Arduino**  
<https://www.arduino.cc/>

## ב. תרשימים וטבלאות

❖ עיקרון פעולת האלגוריתם להימנעות ממכשולים:



❖ תרשים מחלקות עתידי של התוכנה:



## ג. תכנון הפרויקט

היכרות ותחילת עבודה עם Airscort.	<b>04.10.15</b>
התחלת ביצוע מחקר על חיישנים.	<b>01.11.15</b>
הגשת הצעת פרויקט.	<b>22.11.15</b>
רכישת חיישן מרחק ראשון ולמידת סביבת העבודה.	<b>01.12.15</b>
תחילת מחקר על אלגוריתמים וגישות שונות להימנעות ממכשולים ותכנון אלגוריתם ראשוני להימנעות.	<b>01.01.16</b>
קבלת החיישן וחיבורו לבקר ארדואינו. ביצוע בדיקה לתקינות באמצעות קוד פשוט.	<b>10.01.16</b>
הגשת אב טיפוס.	<b>17.01.16</b>
חיבור המערכת לבקר ה-Raspberry Pi, הכרת סביבת העבודה ותחילת עבודה בשפת Python.	<b>20.01.16</b>
העברת פקודות הזזה לרחפן דרך מחשב ה-Raspberry Pi והמשך פיתוח האלגוריתם לזיהוי מכשולים.	<b>10.02.16</b>
סיום פיתוח האלגוריתם לזיהוי המכשולים וביצוע טסטים לאלגוריתם. המשך פיתוח האלגוריתם להימנעות ממכשולים בעזרת פקודות הזזה.	<b>10.03.16</b>
הזמנה וחיבור חיישנים נוספים או מציאת דרך אחרת לסריקת שטח נרחב יותר. בדיקה והמשך פיתוח אלגוריתם להימנעות ממכשולים.	<b>01.04.16</b>
בניה – הרצת סימולציות של התוכנה להימנעות ממכשולים, תיקון באגים ושיפור הקוד.	<b>05.05.16</b>
הרכבת המערכת על גבי הרחפן וביצוע סימולציה מלאה לבדיקת האלגוריתם ותיקונים אחרונים.	<b>1.06.16</b>
מסירה	<b>19.06.16</b>
העברה.	<b>07.07.16</b>

## ד. טבלת סיכונים

#	הסיכון	חומרה	מענה אפשרי
1	אי עמידה בזמנים	5/5	הוספת שעות נוספות
2	תלות בצוות מקביל	3/5	סנכרון, פגישה ותיאום ציפיות עם הצוות המקביל.
3	אי זמינות של הרחפן לסימולציה	2/5	קביעת זמן שימוש ברחפן מספיק זמן מראש
4	עיכוב בפיתוח פקודות תזוזה לרחפן ע"י צוות מקביל	3/5	ביצוע הרצה יבשה – חיבור המערכת על גבי הרחפן, הזנתו מול מכשולים וביצוע הדפסות לבדיקת האלגוריתם.
5	בעיה בהוספת החיישנים לתוכנה בעזרתה נבצע את הסימולציות	4/5	קריאה מרובה של החומר וצפייה בקודים ב github עבור בקר ה arduino. בקשת עזרה מחברי הצוות היותר מנוסים עם תוכנה זו.
6	חוסר ידע באמולטור המתאים לרחפן וסביבות העבודה שלו.	2/5	למידת האמולטור וסביבות העבודה של הרחפן.
7	תכנות לא יעיל שעלול לגרום לעומס חריג על המעבד ולבזבוז סוללה.	5/5	ניתוח זמן הריצה של הקוד שכתבנו וכתיבתו באופן יעיל.
8	חוסר ידע בתכנות בשפת Python	3/5	למידה מרובה, צפייה בסרטונים, התייעצויות עם חברי צוות נוספים.
9	אי התאמה בין רכיבי החומרה	2/5	החלפת מעבד הלוח במעבד אחר על מנת שתהיה התאמה ותקשורת תקינה עם בקר ה-Pixhawk.
10	פגיעה חמורה ברחפן	3/5	הרצת בדיקות וסימולציות יבשות לפני ביצוע ניסויים חיים עם הרחפן.
11	אי הגעה של החיישנים בזמן	4/5	כתיבת קוד עם חיישנים אחרים וניסיון להתאים לחיישנים בזמן הגעתם.
12	אי-יכולת להימנע ב-100% ממכשולים עקב היעדר זמן או חומרה מוגבלת	5/5	השקעת מירב המאמצים לכתיבת קוד הממצא את היכולות של החומרה שברשותי בזמן המוקצב לפרויקט

## ה. טבלת דרישות

#	תיאור
1	מחקר על חיישנים: 1. מהם סוגי חיישני המרחק הקיימים. 2. יתרונות וחסרונות של כל אחד מן הסוגים הקיימים. 3. קבלת החלטה בהתאם לאילוצים שונים: a. משקל. b. צריכת חשמל. c. מחיר. d. טווח פעולה. e. זווית פעולה.
2	חיבור החיישן למחשב ה-Raspberry Pi.
3	כתיבת קוד פשוט לקבלת נתונים מן החיישן ובדיקתו לתקינות (לוודא שאין סטייה גדולה).
3	פיתוח אלגוריתם לזיהוי מכשולים: 1. קליטת נתונים מן החיישן. 2. ניתוח הנתונים. 3. שליחת התראה במידה וזוהה מכשול.
4	פיתוח אלגוריתם להימנעות ממכשולים: 1. עקיפה מימין. 2. עקיפה משמאל. 3. עקיפה מלמעלה. 4. עצירה במקום.
5	הרצות יבשות, חיות וביצוע בדיקות: 1. בדיקת הקלט מהחיישנים. 2. בדיקת האלגוריתם לזיהוי מכשולים. 3. בדיקת האלגוריתם להימנעות ממכשולים. 4. הרצה חיה של הרחפן לבדיקת האלגוריתם להימנעות ממכשולים.