**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 | https://github.com/ben-nakash/Drones-Obstacle-Avoidance-System |
| 2 | יומן | https://trello.com/b/AuUlDKir/obstacles-detection-avoidance-project |
| 3 | ניהול פרויקט | Github |
| 4 | הפצה |  |
| 5 | סרטון אב-טיפוס |  |

תוכן עניינים

[1. מבוא 3](#_Toc440984918)

[2. תיאור הבעיה 4](#_Toc440984919)

[דרישות ואפיון הבעיה 4](#_Toc440984920)

[הבעיה מבחינת הנדסת תוכנה 5](#_Toc440984921)

[3. תיאור הפתרון 6](#_Toc440984922)

[מהי המערכת 6](#_Toc440984923)

[תיאור הפתרון המוצע: 7](#_Toc440984924)

[תיאור הכלים המשמשים לפתרון 8](#_Toc440984925)

[4. תוכנית בדיקות 9](#_Toc440984926)

[5. סקירת עבודות דומות בספרות והשוואה 10](#_Toc440984927)

[6. סיכום / מסקנות 11](#_Toc440984928)

[7. נספחים 12](#_Toc440984929)

[א. ביבליוגרפיה 12](#_Toc440984930)

[ב. תרשימים וטבלאות 12](#_Toc440984931)

[ג. תכנון הפרויקט 13](#_Toc440984932)

[ד. טבלת סיכונים 14](#_Toc440984933)

[ה. ה.טבלת דרישות 15](#_Toc440984934)

# מבוא

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

# 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.

## 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 זעיר 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:

1. קליטת נתונים בזמן אמת לצורך זיהוי מכשולים מחיישני המרחק שיורכבו על גבי הרחפן ויחוברו ללוח ה-Raspberry Pi.
2. הימנעות מפגיעה במכשול אשר התגלה בדרכו של הרחפן – עליי לפתח סקריפט בשפת Python אשר ישלח פקודות הזזה לרחפן על מנת להימנע מן המכשול.

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.

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 שכן למרות שמדובר בקוד פתוח בשפות C וC++, 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 שברחפן והוא ישלח פקודות הזזה לבקר ה-Pixhawk 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 אממש 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.

# C:\Users\Ben\Desktop\תיכון פרויקט גמר.pngתיאור הפתרון

## מהי המערכת

התרשים הקודם מתאר את ארכיטקטורת המערכת ומורכב מכמה חלקים:

* ***בקר ה-Pixhawk*** הינה מערכת טייס אוטומטי המבוססת על קוד פתוח ופותחה ע"י חברת 3D Robotics. בבקר זה משולבים טכנולוגיית חיישנים ומעבד מתקדמים, מערכת הפעלה בזמן אמת המספקת ביצועים מרשימים, גמישות ושליטה לכלים אוטונומיים.  
  דרך בקר זה נשאב את הנתונים מן החיישנים, ואליו נצרוב את האלגוריתמים שנפתח (הפירוט עליהם בסעיפים הבאים).
* ***חיישני מרחק*** – חיישני מרחק יורכבו על הרחפן ויסרקו את השטח שמולם באופן תמידי. נתוני הסריקה ישלחו אל בקר ה-Pixhawk בזמן אמת וינותחו ע"י האלגוריתמים שנפתח ונצרוב אל השבב שלו. בפרויקט זה בחרנו להשתמש בחיישנים העובדים על לייזר, וזאת כיוון שהם מאפשרים סריקה לטווח רחוק יותר.
* ***תוכנה לזיהוי והימנעות ממכשולים*** –תוכנה אשר נפתח ונצרוב לבקר ה-Pixhawk. זו תעבוד על הבקר ותקבל כקלט את נתוני המרחק מהחיישנים, ותשלח לרחפן פקודות תזוזה להימנעות מהמכשולים שזוהו.

## תיאור הפתרון המוצע:

***זיהוי והימנעות ממכשולים:***

1. חיבור החיישנים לבקר ה-Pixhawk באמצעות חיבור I2C, בדיקת החיישנים לתקינות קלט ע"י הזזת חפצים מול הרחפן.
2. פיתוח האלגוריתם ראשוני של זיהוי המכשולים והבנת מיקומם ביחס לרחפן.
3. סימולציות ראשוניות, הרצת האלגוריתם ובדיקתו באמצעות הדפסות למסך.
4. פיתוח פקודות תזוזה לרחפן בהתאם למיקום המכשול שזיהינו באמצעות האלגוריתם שפותח ונבדק בשלב קודם.
5. סימולציות, והרצות נוספות. בדיקת האלגוריתם בזמן אמת תוך כדי הפעלת הרחפן.

## 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.
* ***Arduino Software (IDE):***

סביבת העבודה של Arduino. נוכל להיעזר בה על מנת לשאוב את הנתונים הנמדדים מהחיישנים וביצוע ניסויים שונים בקלות יחסית (ישמש אותנו בעיקר בתחילת העבודה עם החיישנים וצבירת ניסיון, יותר מאוחר נעבור לתכנות בקר ה-Pixhawk).

* ***ArduPilotMega (APM):***

פלטפורמה מבוססת קוד פתוח של כלי טיס בלתי מאוישים שבעזרתה ניתן לשלוט על רחפנים אוטונומיים. בעזרת Eclipse נוכל לערוך את קוד ה-ArduPilot שיתאים לצרכים שלנו ולתת לבקר ה-Pixhawk מטרות.

* ***DroneKit-Python API:***

API המאפשר למפתחים ליצור תוכניות שירוצו על המעבד הנמצא ברחפן. אותו מעבד מדבר ושולט על ה-Autopilot. תוכניות הרצות על מעבד זה מסוגלות לבצע משימות עיבוד אינטנסיביות ומשימות תלויות זמן אמת כמו זיהוי מכשולים ושינוי נתיב טיסה.  
API זה עובד בסביבות העבודה Unix ו- Windows ונותן לנו (המשתמשים) גישה לביצוע משימות כמו:

* התחברות לרחפן (או מספר רחפנים) ע"י סקריפט.
* שינוי וקבלת מצב הרחפן ונתונים שונים לגביו.
* קבלת מסרים אסינכרוניים על שינויי מצבים.
* הדרכת הרחפן לטוס למיקום מסויים.
* שליחת פקודות שרירותיות לשליטה על תנועת הרחפן וחומרה נוספת במידה ויש.

# תוכנית בדיקות

# סקירת עבודות דומות בספרות והשוואה

* Autonomous quadcopter, obstacle detection on Arduino

<https://github.com/bitcoinsoftware/UrbanDroneSystem>

קוד לזיהוי מכשולים של רחפן הכתוב Arduino.

* source code used to control a Quadcopter

<https://github.com/2012SEng466/copter>

מאגר זה מכיל את קוד מקור משמש לשליטה ברחפן מכיל תכנית Arduino לקריאת פקודות סידורי ממחשב ולשלוח פקודות מוטוריות לרחפן.

* Obstacle detection and collision avoidance using ultrasonic distance sensors for an autonomous quadrocopter.

<http://www.informatik.uni-wuerzburg.de/fileadmin/10030800/user_upload/quadcopter/Paper/Gageik_Mueller_Montenegro_2012_OBSTACLE_DETECTION_AND_COLLISION_AVOIDANCE_USING_ULTRASONIC_DISTANCE_SENSORS_FOR_AN_AUTONOMOUS_QUADROCOPTER.pdf>

מאמר על זיהוי מכשולים והימנעות התנגשות בעזרת חיישני אולטרסאונד.

* Pixhawk

<http://copter.ardupilot.com/wiki/common-pixhawk-wiring-and-quick-start/>

הסבר על איך ניתן לשלוט על ה Pixhawk. והסבר על הבקר.

# סיכום / מסקנות

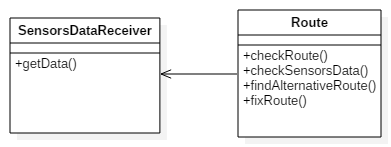
# נספחים

## ביבליוגרפיה

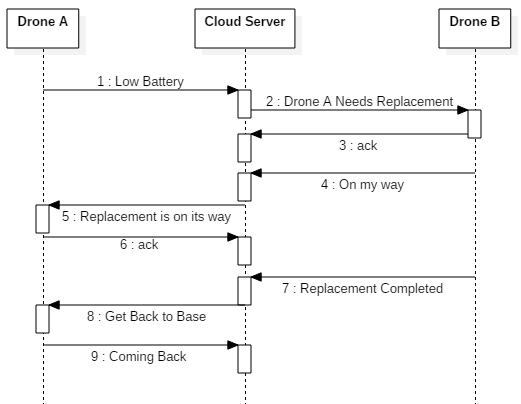
* http://copter.ardupilot.com/
* https://phidgets.wordpress.com/2014/05/23/exploring-the-many-methods-of-object-detection/
* http://diydrones.com/

## תרשימים וטבלאות

* Class Diagram כללית של התוכנה להימנעות ממכשולים וניתוב מחדש של הרחפן:

****

* Sequence Diagram של חילוף בין שני רחפנים:



## תכנון הפרויקט

|  |  |
| --- | --- |
| **04.10.15** | היכרות ותחילת עבודה עם Airscort |
| **01.11.15** | התחלת ביצוע מחקר על חיישנים |
| **22.11.15** | הגשת הצעת פרויקט |
| **01.12.15** | רכישת חיישן מרחק ראשון ולמידת סביבת העבודה |
| **17.01.16** | הגשת אב טיפוס - ביצוע ניסויים ובדיקות ראשונות לחיישן המרחק. |
| **01.03.16** | מתן פקודות לרחפן דרך בקר הpixhawk לתזוזתו.  סימולציות של הרחפן לתגובת הפקודות.  בדיקה ומדידת זמני העברת הנתונים בערוץ השידור בין הרחפן לשרת הענן. |
| **01.04.16** | חיבור חיישנים נוספים לבקר הpixhawk. בדיקה והמשך פיתוח אלגוריתם להימנעות ממכשולים.  מימוש אלגוריתם להזזת הרחפן המוחלף מן הרחפן המחליף על מנת להימנע מהתנגשות. |
| **05.05.16** | בניה – הרצת סימולציות ראשוניות על תוכנת הימנעות ממכשולים ובדיקת התוכנה להימנעות ממכשולים.  בדיקת תקינות האלגוריתם ע"י הדפסת נתוני מיקום של כל אחד הרחפנים תוך כדי חילוף. |
| **1.06.16** | בדיקות התוכנה להימנעות ממכשולים תוך כדי הפעלת הרחפן.  שיפור אלגוריתם להחלפת הרחפנים והרצת סימולציות נוספות בזמן אמת. |
| **19.06.16** | מסירה |
| **07.07.16** | העברה. |

## 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** | חיבור החיישן לא מתאים לבקר ה-Pixhawk | 2/5 | מחקר מראש על החיישן, חיבוריו ואופן פעולתו. |
| **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 ולגרום בנוסף לבזבוז סוללה. | 5/5 | ניתוח זמן הריצה של הקוד שכתבנו וכתיבתו באופן יעיל עד כמה שנוכל. |
| **8** | חוסר ידע בבקר ה-Pixhawk ואופן פעולתו עם הרחפן. | 4/5 | למידת הפקודות שדרכן הבקר מתקשר עם חומרת הרחפן, אופן העברתן וביצוען. |
| **9** | עיכוב בפיתוח צד השרת הדורש קבלה ושליחה ממנו ואליו. | 4/5 | תלות בצוות אחר אשר מפתח את צד השרת. ניתן נקודות ציון משלנו במקום לביצוע הניסויים,  ושליחת נתונים למקור חיצוני. |
| **10** | פגיעה חמורה ברחפן | 3/5 | הרצת בדיקות וסימולציות יבשות לפני ביצוע ניסויים חיים עם הרחפן. |
| **11** | עבודה חופפת בתוך הצוות | 2/5 | משימות מוגדרות ושונות לחברי הצוות לייעול העבודה בפרויקט |
| **12** | אי הגעה של החיישנים בזמן | 4/5 | כתיבת קוד עם חיישנים אחרים וניסיון להתאים לחיישנים בזמן הגעתם. |

## 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:    1. משקל.    2. צריכת חשמל.    3. מחיר.    4. טווח פעולה.    5. זווית פעולה. |
| **2** | עדכון השרת:  1.שליחת מיקום של הרחפן.  2. שליחת מצב סוללה. |
| **3** | פיתוח אלגוריתם לזיהוי מכשולים:   1. קליטת נתוני מדידה של החיישנים דרך בקר ה-Pixhawk. 2. ניתוח הנתונים, ומתן התראה במידה וזוהה מכשול. |
| **4** | פיתוח אלגוריתם להימנעות ממכשולים:   1. בדיקה לקבלת התראת מכשול. 2. פיתוח מספר גישות שונות להימנעות מן המכשול (עקיפה מימין/משמאל/מלמעלה ועצירה לצורך בדיקות). 3. מתן נתיב אלטרנטיבי ושינוי נתיב הטיסה. |
| **5** | פיתוח אלגוריתם להחלפת רחפנים באופן יעיל:   1. קבלת מסר מהשרת להחלפת הרחפן ברחפן אחר. 2. ביצוע החילוף ע"י העלאת הרחפן המוחלף גבוה יותר על מנת לאפשר למחליפו להגיע ללא סכנת התנגשות. 3. שליחת סיום החלפת האלגוריתם לשרת. |
| **6** | הרצות חיות ועשיית מבדקים לכל אורך הפיתוח:   1. בדיקת הקלט מהחיישנים. 2. בדיקת האלגוריתם לזיהוי מכשולים. 3. בדיקת האלגוריתם להימנעות ממכשולים. 4. בדיקת שליחת נתונים של הרחפן למקור חיצוני(שרת). 5. הרצה חיה של הרחפן לבדיקת האלגוריתם להימנעות ממכשולים. 6. בדיקת אלגוריתם להחלפת רחפנים בעזרת הדפסות למסך של מיקום הרחפנים בכל זמן נתון לאחר בקשת חילוף. 7. הרצה חיה ובדיקת האלגוריתם להחלפת הרחפנים. |