

המחלקה להנדסת תוכנה

מערכת לזיהוי והימנעות ממכשולים עבור רחפן אוטונומי

חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת:

בן נקש

יולי 2016

תמוז התשע"ו

המחלקה להנדסת תוכנה

מערכת לזיהוי והימנעות ממכשולים עבור רחפן אוטונומי

חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת:

בן נקש

מנחה אקדמי: מר שי תבור	אישור:	תאריך:
אחראי תעשייתי: מר טל יצחק	אישור:	תאריך:
רכז הפרויקטים: דר' יגל ראובן	אישור:	תאריך:

מערכת ניהול פרויקט ובקרת תצורה

#	מערכת	מיקום
1	מאגר קוד	https://github.com/ben-nakash/Drones-Obstacle-Avoidance-System
2	יומן	https://trello.com/b/AuUIDKir/obstacles-detection-avoidance-project
3	ניהול פרויקט	Github
4	הפצה	
5	סרטון	https://drive.google.com/open?id=0B-1pi9mMfnD8ZGFPbm9aMEV2UmM

תקציר

פרויקט זה מתמקד בעולם הטכנולוגי של הרחפנים ומטרתו לסייע לשילובם במגזר התעשייתי. רחפן הינו כלי טייס בלתי מאויש המהווה יחידה אינטגרטיבית המשלבת בתוכה מערכות שונות לרבות חיישנים, בקר טיסה, GPS ואחרות. בשנים האחרונות גברה מגמת השימוש ברחפנים בעקבות ההתפתחות הטכנולוגית והוזלת המחירים וכיום הם תופסים חלק גדול יותר בחיינו במגוון רחב של תחומים ואפליקציות – החל ממשחק ילדים, צילום ומשלוחים ועד לפיתוחים אישיים של אנשים פרטיים למטרות שונות.

לאחר סקירת שימושי הרחפנים כיום ניתן להסיק כי לעת עתה הרחפנים אינם לוקחים חלק מהותי במגזר השוק העסקי אך זוהה הפוטנציאל הגדול הטמון בשימוש בהם. על כן, נעשה מחקר על ידי מייסדי חברת הסטארט-אפ Airscort בדבר הגורם המונע את שילובם הרחב בתעשייה. תוצאות המחקר העלו כי הגורם המרכזי לכך נעוץ בזמן חיי הסוללה הקצר של הרחפנים, אך בנוסף לכך הובן כי על הרחפן להיות מסוגל לבצע משימות מורכבות באופן עצמאי. על מנת לאפשר לרחפן להוות יחידה אוטונומית עליו להיות מסוגל לטוס לבדו. צורך זה העלה עמו בעיה חדשה – כיצד יוכל הרחפן לטוס בבטיחות ללא השגחה ולא להיתקל במכשולים?

פרויקט זה בא לתת מענה לבעיה על ידי פיתוח מערכת להימנעות ממכשולים עבור הרחפן.

חברת הסטארט-אפ Airscort מפתחת פתרונות פורצי דרך בתחום הרחפנים על מנת לסייע לשילובם במגזר התעשייה העסקית, ופרויקט זה מבוצע בחסותם. לצורך פיתוח אב טיפוס של רחפן אוטונומי, הגדירה החברה דרישות לפיתוח מערכת להימנעות ממכשולים שתותקן ברחפן המיועד לצלם אדם בעת גלישת סקי. בתיאום מול מייסדי החברה הוחלט על שתי הנחות יסוד עליהן יבוסס הפיתרון המערכתי:

- ❖ הרחפן יטוס בגובה קבוע של כ-10 מטרים.
- ❖ המכשולים מהם הרחפן ימנע יהיו סטטיים בלבד, כגון עצים ועמודים.

דרישות החברה נותחו ובוצע מחקר אנליטי בדבר החומרה הדרושה למערכת אותה יש לפתח במקביל לסקר ספרות על פרויקטים דומים. על פי ממצאי המחקר הוחלט כי המערכת תורכב ממערך של כארבעה חיישני מרחק שיחוברו באמצעות חיבור טורי למחשב לוח יחיד, ומחשב זה יחובר ישירות לבקר הטיסה של הרחפן. בצורה זו, יקבל המחשב נתוני מרחק מן החיישן, יפענחם בעזרת אלגוריתם מורכב וישלח פקודות הזזה מתאימות לבקר הטיסה של הרחפן לצורך תיקון מסלול הטיסה שלו והימנעות מן המכשול שזוהה.

מפאת תקציב מוגבל ומחירו הגבוה של החיישן הייתה באפשרות החברה לרכוש רק חיישן אחד. אילוץ זה יצר בעיית חוסר מידע - בדיקת המערכת כאשר היא מקבלת קלט מחיישן אחד במקום משלושה. על כן, לצורך בדיקת המערכת פותחה מחלקה נוספת בתוכנה אשר מזינה נתוני מרחק למערכת מקבצי טקסט המכילים נתוני מדידה אמיתיים שנלקחו מתוך מסלול שתוכן מראש ונטען למערכת.

כיוון שלא ניתן היה לבצע בדיקה מלאה של המערכת על גבי הרחפן, לא ניתן להבטיח באופן גורף את הצלחת המערכת בסיטואציות שונות, אך בהתבסס על הסימולציה שבוצעה ניתן לראות כי המערכת עובדת כמצופה לשלב זה של הפיתוח ומספקת תוצאות טובות.



הצהרה

העבודה נעשתה בהנחיית מר שי תבור במחלקה להנדסת תוכנה, עזריאלי – המכללה
האקדמית להנדסה ירושלים ובחברת הסטארט-אפ Airscort.
החיבור מציג את עבודתי האישית ומהווה חלק מהדרישות לקבלת תואר ראשון בהנדסה.

תודות

ברצוני להביע רב תודות למספר אנשים אשר סייעו לי במהלך הפרויקט ותמכו בי לכל אורך הדרך.

בראש ובראשונה ברצוני להודות למנחה הפרויקט: מר שי תבור, על עזרתו המקצועית, על מתן הכוונה ועזרה בפיתרון בעיות שונות, על הייעוץ ועל ההדרכה לכל אורך הפרויקט.

ארצה להודות לשי פאר על מתן הכוונה, עזרה בפיתרון בעיות ועל מתן סיוע רב בפיתרון בעיות הקשורות לרכיבי החומרה השונים איתם עסקתי לאורך הפרויקט.

תודה מיוחדת לחברתי, לחבריי, להוריי ולאחיי אשר עזרו, תמכו ועודדו אותי לאורך כל מהלך הפרויקט.

לסיום, ברצוני להודות לאיתי שטראוס וליצחק טל – מייסדיה של חברת הסטארט-אפ Airscort בה מימשתי את פרויקט זה, על מתן האמון והוספתי לצוותם, על יחסם החם ועל העידוד והסיוע בכל היבט אשר הייתי זקוק.

תוכן עניינים

4.....	תקציר	
5.....	הצהרה	
6.....	תודות	
9.....	מילון מונחים	
11.....	מבוא	1.
11.....	מהו רחפן	1.1
12.....	התפתחות הרחפנים	1.2
13.....	תיאור מסגרת הפרויקט	1.3
14.....	תיאור הבעיה	2.
14.....	תיאור כללי	2.1
15.....	דרישות ואפיון הבעיה	2.2
15.....	הבעיה מבחינת הנדסת תוכנה	2.3
16.....	תיאור הפיתרון	3.
16.....	תיאור כללי	3.1
16.....	תכנון החומרה	3.2
17.....	תכנון התוכנה	3.3
19.....	תיאור המערכת שמומשה	4.
19.....	תיאור הכלים אשר שימשו לפיתרון	4.1
20.....	תיאור החומרה שמומשה	4.2
20.....	רכיבי החומרה בהם נעשה שימוש	4.2.1
20.....	אפיק תקשורת I2C	4.2.2
21.....	מחשב לוח מסוג Raspberry Pi 2	4.2.3
22.....	מיקרו-בקר מסוג Arduino	4.2.4
23.....	מיקום החיישנים על גבי הרחפן	4.2.5
25.....	תיאור התוכנה שמומשה	4.3
25.....	תיכון המערכת שמומשה	4.3.1
26.....	מחלקות המערכת	4.3.2
28.....	אופן הפעלת המערכת	4.3.3
29.....	פירוט השינויים שיש לבצע בעת מעבר למערכת מלאה	4.3.4

30.....	תרשים זרימה המתאר את אופן פעולת האלגוריתם	4.3.5
33.....	תוכנית בדיקות	5.
33.....	בדיקת תקינות קלט	5.1
33.....	מחלקת ObstacleAvoidance	5.1.1
33.....	מחלקת FlightCommands	5.1.2
34.....	מחלקת FlightData	5.1.3
34.....	סימולציית טיסה	5.2
36.....	סקירת עבודות דומות בספרות והשוואה	6.
37.....	מסקנות מהמימוש ומהפרויקט	7.
37.....	חומרה	7.1
38.....	תוכנה	7.2
39.....	נספחים	8.
39.....	ביבליוגרפיה	8.1
40.....	תרשימים וטבלאות	8.2
42.....	תכנון הפרויקט	8.3
43.....	טבלת סיכונים	8.4
44.....	טבלת דרישות	8.5
45.....	חישוב זמן-מרחק-דרך	8.6
45.....	חישוב מרחק בין שתי נקודות ציון במרחב	8.7
46.....	חיישני מרחק	8.8
46.....	קביעת טווח הפעולה הדרוש	8.8.1
47.....	סוגי חיישני מרחק	8.8.2
48.....	בחירת סוג החיישן המתאים ביותר	8.8.3
49.....	רשימות	8.9
49.....	רשימת איורים	8.9.1
49.....	רשימת טבלאות	8.9.2
50.....	רשימת משוואות	8.9.3
50.....	רשימת סימנים	8.9.4
51.....	Summary	9.

מילון מונחים

עברית		
1	בקר טיסה	מחשב ובתוכו מערכת טייס אוטומטי של רחפן.
2	ג'ירוסקופ	מכשיר המשמש למדידה או שמירה של יציבות.
3	כטב"מ	כלי טייס בלתי מאויש.
4	מדידה שגויה	קלט לא הגיוני מן החיישן, כלומר ערך קטן או שווה ל-0.
5	מזל"ט	מטוס זעיר ללא טייס.
6	מיקרו-בקר	רכיב מתוכנת שהוא בעצם מערכת מחשב פונקציונאלית שלמה הפועלת על גבי רכיב אלקטרוני בודד. לבקר רכיבי כניסות ויציאות I/O המשמשים לבקרה ולתקשורת עם העולם החיצוני (חיישנים לדוגמא).
7	סביבת עבודה	אוסף של רכיבי מחשב המאפשרים לבצע משימות מוגדרות לאותו השלב בפרויקט המחשוב.
8	פלטפורמה	מערכת המורכבת מסוג כלשהי של ארכיטקטורת חומרה יחד עם תשתית תוכנה. החיבור בין שני אלו אמור לאפשר הרצה של תוכנות ויישומי מחשב.
9	פרוטוקול בטיחות	פרוטוקול המונע מן הרחפן לטפס גבוה מידי באוויר לשם עקיפת מכשול.
10	קואדקופטר	רחפן בעל 4 מנועים.
11	קמ"ש	קילומטר לשעה
12	רחפן	כלי טייס בלתי מאויש בעל 4 מנועים או יותר.
13	רחפן אוטונומי	רחפן המסוגל לטוס באופן עצמאי ללא עזרה מגורם אנושי חיצוני.
14	תקשורת טורית/סריאלית	תהליך שליחת מידע במחשב בו שליחת המידע מתבצעת על ידי שליחת סיבית (bit) אחת בכל פרק זמן נתון.

אנגלית		
1	API	Application Programming Interface או בעברית ממשק תכנות יישומים. זהו כינוי לערכות של ספריות קוד, פקודות, פונקציות ופרוצדורות מוכנות בהן יכולים מתכנתים לעשות שימוש פשוט מבלי לכתוב אותן בעצמם.
2	Computer Bus	אפיק נתונים. זוהי תת מערכת המשמשת להעברת מידע בין רכיבי מערכות ממוחשבות.
3	DroneKit	ערכת פיתוח תוכנה (SDK) וממשק לתכנות יישומים (API) לפיתוח קל של אפליקציות לרחפנים.
4	GPS	Global Positioning System או בעברית מערכת ניווט לוויינית.
5	I2C	Inter Integrated Circuit. אפיק תקשורת טורית להעברת נתונים.
6	IDE	Integrated Development Environment, סביבת פיתוח משולבת המסייעת למתכנתים לפתח תוכנה.
7	Intel Edison	מחשב לוח יחיד המריץ מערכת הפעלה מלאה מבוססת Linux.
8	Pixhawk	בקר טיסה המשלב מערכת טייס אוטומטית המריץ מערכת הפעלה יעילה ביותר בזמן אמת.
9	Raspberry Pi	מחשב לוח יחיד המריץ מערכת הפעלה מלאה מבוססת Linux.
10	SDK	Software Development Kit. ערכת פיתוח תוכנה. זהו סט של כלים לפיתוח תוכנה המאפשרים יצירת יישומים עבור פלטפורמה מסוימת כגון סביבת עבודה, מחשב או מערכת הפעלה.

1. מבוא

1.1 מהו רחפן

רחפן הינו כלי טיס בלתי מאויש (כטב"מ) המהווה יחידה אינטגרטיבית המשלבת בתוכה מערכות שונות לרבות חיישנים, בקר טיסה, מערכת מיקום גלובאלית (GPS), מנועים ועוד. מדובר במעין הליקופטר זעיר בעל 4 מנועים או יותר אשר מסוגל לנוע באוויר בחופשיות בכיוונים שונים תודות לריבוי מנועיו, זאת בשונה לכלי טיס עם כנף קבועה ומנוע יחיד.

על פי מספר המנועים המורכבים על גבי הרחפן נקבע שמו וסוגו. פרויקט זה עוסק ברחפן מסוג קואדקופטר (Quadcopter) שהוא רחפן בעל ארבעה מנועים.



איור 1 : רחפן מסוג IRIS-M של חברת 3DR בעל 4 מנועים אשר ברשות החברה

רחפן מסוג קואדקופטר הינו קל יותר לשליטה מכלי טיס בעלי כנף קבועה והוא נשלט מרחוק על ידי בקר טיסה. תנועתו מתאפשרת על ידי שינוי מהירות הסיבוב של מנועיו השונים הגורר שינוי בדחף המיוצר על ידי כל אחד מן המנועים. בצורה זו מתאפשרת הזזתו של הרחפן בכל כיוון, בהבדל מכלי טיס בעל כנף קבועה שכיוון תנועתו הוא ישר בלבד.

רחפנים נחשבים לכלי טיס בלתי יציבים מבחינה אווירודינמית. לכן, על מנת לשמור על דיוק תנועתם וכיוון הטיסה שלהם, משלב בקר הטיסה את הפקודות ממפעילו של הרחפן יחד עם נתונים המתקבלים ממדי תאוצה וג'ירוסקופים אשר מובנים בו.

1.2 התפתחות הרחפנים

בשנת 1969 בעת המתיחות הרבה בין ישראל ומצרים, החלו הרוסים לספק לצבא המצרי מערכות טילים נגד מטוסים (נ"מ). כתוצאה מכך, נולד איום חדש על מטוסי הקרב הישראליים, וגיוחות הצילום שנערכו עד אותה עת מעבר לתעלת סואץ הפכו למסוכנות אף יותר. בעקבות כך, עלה הצורך במציאת פיתרון לבעיה. על הפיתרון עלו חבורה של אוהבי טיסנאות אשר הציעו פיתרון מקורי - הם הרכיבו מצלמת סטילס על טיסן ממונע ובנוסף חיברו מנוע סרוו ללחיצה על מתג הצילום, כאשר אלו נשלטים על ידי שלט רחוק. לאחר מספר ניסויים מוצלחים יצא הטיסן למשימתו הראשונה ולאחר ששב בשלום ותמונות הפילם שצילם פותחו בהצלחה היה זה ברור כי נולד המטוס הזעיר ללא טייס (מזל"ט).

לאחר כפחות מעשר שנים הוקמה יחידת המזל"טים הראשונה בצה"ל אשר עשתה שימוש במזל"ט חדיש יותר אשר היה מסוגל לטוס למרחק של עשרות קילומטרים ונשא עימו מצלמת וידאו שסיפקה סרטי וידאו אשר נצפו בזמן אמת בקרון השליטה והבקרה. בחלוף השנים, גבר הביקוש למזל"טים והשתכללו יכולותיהם - החל מטיסות לטווחים רחוקים יותר, דרך מצלמות בעלות זום גדול יותר וכלה בצילומי צבע וצילומים בלילה (אינפרה אדום).

המעבר של הכטב"מ לעולם היישומים האזרחיים היה צפוי ומתבקש אך היצרנים נתקלו בשני חסמים עיקריים אשר מנעו את התפתחות הכטב"מים בשוק האזרחי:

❖ **רגולציה:** באף מדינה לא התפתחה חקיקה מסודרת לגבי הפעלתם החוקית של כטב"מים במרחב האזרחי ונערכו אינספור דיונים בסוגיה כיצד מערכות אלו ישולבו במרחב האזרחי באופן שלא יסכן את ביטחונם של אזרחים תמימים. כתוצאה מכך הוסכם כי כל עוד אין תשתית רגולטורית והסדר בחוק אין טעם להשקיע בפיתוח של מערכות אזרחיות.

❖ **מחיר:** כל עוד הייצור של אותן טכנולוגיות היה נחלתן של מדינות בודדות בלבד, מחירן של מערכות אלו היה יקר ברמה שלאזרחים כבר לא היה משתלם לרכוש כטב"מ.

בינתיים, מחוץ למפעלים הביטחוניים התחוללה מהפכה עליה היו אחראים שוב חובבי טיסנאות. בדומה לאותם חובבי טיסנאות ותיקים אשר המציאו את המזל"ט הצבאי לפני 45 שנה, חזרו חובבי הטיסנים והמציאו אותו שוב - בזאת נולד הרחפן. הם ישבו בבית, פירקו והרכיבו, ניסו ובדקו, הטיסו וריסקו. אך הפעם להבדיל משנות השבעים, הם לא עשו זאת בסודי סודות, אלא בשיתוף פעולה עם אנשים רבים מרחבי העולם על ידי שימוש ברשת האינטרנט. הם שיתפו את הידע והניסיון שצברו, שאלו שאלות בפורומים משותפים, רכשו רכיבים משוכללים וזולים בחנויות אינטרנטיות בהונג-קונג ופיתחו תוכנה בקוד פתוח אשר זמין לכולם.

הזמינות של הידע והטכנולוגיה הובילה מהר מאוד לפיתוחם של רחפנים קטנים ומשוכללים. עלות הייצור והפיתוח של רחפנים אלו, אשר פשוטים יותר לשימוש ממטוסים בעלי כנף קבועה, הייתה אחוז בודד מעלותם של כטב"מים צבאיים ובכך נפרץ מחסום המחיר.

כשהרחפנים החלו להימכר ברחבי העולם ונוצרה אלטרנטיבה אמיתית בשוק האזרחי נוצר ביקוש. הביקוש הוביל לשימוש, והשימוש יצר מציאות חדשה שחייבה התייחסות וחקיקת חוקים מתאימים ובכך נפרץ גם המחסום הרגולטורי.

כיום, גוברת מגמת השימוש ברחפנים ונעשה בהם שימוש בקשת רחבה של יישומים וביניהם צילום, חקלאות, פנאי ועוד.

1.3 תיאור מסגרת הפרויקט

פרויקט גמר זה מבוצע במסגרת חברת הסטארט-אפ Airscort. חברה זו הוקמה על ידי קבוצת מהנדסי מכונות, תוכנה וחשמל אשר רואים עתיד לטכנולוגית הרחפנים ומפתחים פיתרון פורץ דרך בתחום טכנולוגיית הרחפנים בתעשייה האזרחית.

הטכנולוגיה והאפליקציות אותן מפתחים ב-Airscort יאפשרו מגוון רחב של יישומי רחפנים אזרחיים, ביניהם:

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

על מנת לאפשר לרחפן לבצע את מגוון היישומים הנראים לעיל, על הרחפן להיות מסוגל לטוס ולבצע משימות מורכבות תוך כדי טיסה אוטונומית ללא מגע יד אדם. על מנת לאפשר זאת, על הרחפן להיות מסוגל לזהות מכשולים שונים בדרכו ולהימנע מהם באופן עצמאי. על כן, הצורך עליו עונה פרויקט זה הוא פיתוח מערכת לזיהוי והימנעות ממכשולים עבור הרחפן.



איור 2 : לוגו החברה

2. תיאור הבעיה

2.1 תיאור כללי

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: חקלאי נדרש לבדוק באופן קבוע את מצב השדה שלו באופן אישי – פעולה המבזבזת זמן רב לחקלאי. לכן, נדרוש שהרחפן יעבוד באופן אוטונומי וללא התערבות של גורם חיצוני. בצורה זו, אותו חקלאי יוכל לשלוח את הרחפן לבדוק את מצב השדה שלו, כאשר בו בזמן יוכל להמשיך בעיסוקיו השונים.

דוגמה נוספת הינה כאשר אדם מבצע ספורט אתגרי כדוגמת סקי, ומעוניין לתעד את עצמו. כיום, אדם אינו יכול לעשות ונדרש אדם נוסף שיצלמו.

על פי דרישת החברה, פרויקט זה מתרכז בפיתוח מערכת להימנעות ממכשולים עבור רחפן העוקב ומצלם אדם בעת גלישת סקי.

בתיאום מול מייסדי החברה הוחלט על שתי הנחות יסוד עליון יבוסס הפיתרון המערכתי:

- ❖ הרחפן יטוס בגובה קבוע של כ-10 מטרים.
- ❖ המכשולים בהם יתקל הרחפן הם סטטיים ואינם מסוגלים לזוז.



איור 3 : רחפן העוקב אחר אדם בעת גלישת סקי וסורק את נתיב הטיסה שלו לצורך הימנעות ממכשולים

2.2 דרישות ואפיון הבעיה

קיימות מספר דרישות מבחינתו של אדם המעוניין להשתמש ברחפן ובו המערכת שתפותח במסגרת פרויקט זה:

- ❖ לא יהיה צורך לכוון את הרחפן מחדש בכל פעם שמתגלה מכשול כלשהו במסלולו (כיול).
- ❖ מערכת החיישנים המורכבת על גבי הרחפן תספק טווח זיהוי מכשולים גדול דיו על מנת לספק לרחפן מספיק זמן תגובה על מנת לשנות את מסלולו ולהימנע מהתנגשות.
- ❖ אלגוריתם מתוחכם יעבד את נתוני החיישנים ויסיט את הרחפן ממסלולו המקורי בעת זיהוי מכשול על מנת להימנע מפגיעה.
- ❖ לא יהיה צורך להפעיל את המערכת באופן נפרד לשאר מערכות הרחפן.

2.3 הבעיה מבחינת הנדסת תוכנה

ישנן מספר בעיות תכנותיות שפרויקט זה מגיע לפתור:

- ❖ קליטת נתונים בזמן אמת לצורך זיהוי מכשולים מחיישני המרחק שיורכבו על גבי הרחפן ויחוברו ללוח ה-Raspberry Pi.
- ❖ פיתוח אלגוריתם לניתוח נתוני החיישנים וקבלת החלטה כיצד לתקן את מסלול הטיסה של הרחפן על מנת להימנע ממכשול שהתגלה. יש לציין כי קיימים אלגוריתמים לזיהוי מכשולים, אך אלו עושים שימוש בטכנולוגיה מתקדמת ויקרה שאינה במסגרת התקציב של החברה ולכן על האלגוריתם לאפשר את זיהוי המכשולים והימנעות מהם במסגרת התקציב והחומרה המוגבלים.
- ❖ קבלת נתוני מיקום של הרחפן לצורך קבלת החלטות באלגוריתם ההימנעות.
- ❖ התממשקות עם מחלקה חיצונית המפותחת בפרויקט אחר במסגרת החברה על מנת לבצע שליחה של פקודות טיסה לבקר ה-Pixhawk.

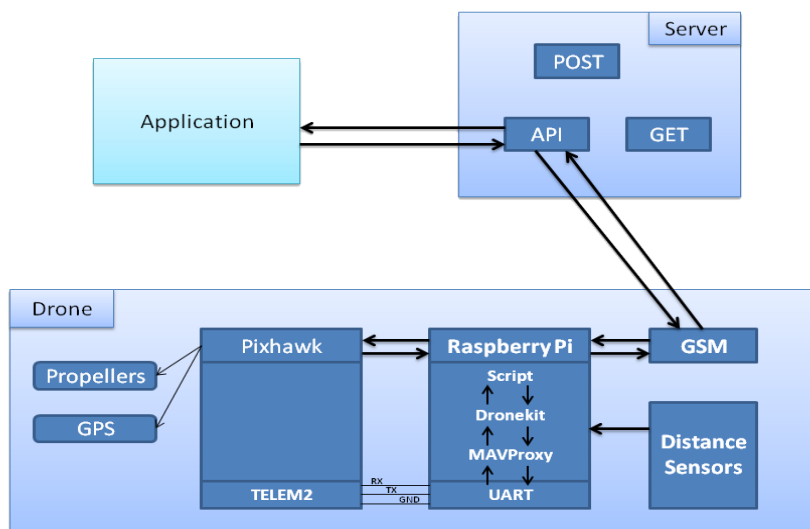
3. תיאור הפיתרון

3.1 תיאור כללי

מערכת ההימנעות ממכשולים מורכבת הן מרכיבי תוכנה והן מרכיבי חומרה. בבסיס הרחפן ישנו בקר טיסה מסוג Pixhawk ובו מערכת טייס אוטומטי המריצה מערכת הפעלה יעילה ביותר המתוכננת לעבודה ועיבוד נתונים בזמן אמת. לבקר זה יחובר מחשב לוח בשם Raspberry Pi. על גבי מחשב זה תרוץ המערכת שתפותח במסגרת פרויקט זה יחד עם שאר מערכות הרחפן.

למחשב לוח זה יחוברו כארבעה חיישנים, אשר יסרקו באופן קבוע את השטח שממול הרחפן ויזרימו לו נתוני מדידות. בעזרת אלגוריתם מתקדם, יוכל המחשב לזהות מכשולים בזמן הטיסה ולשלוח פקודות מתאימות לבקר הטיסה על מנת לתקן את מסלולו של הרחפן ולהימנע מהתנגשות.

3.2 תכנון החומרה



איור 4: ארכיטקטורת המערכת

❖ **Raspberry Pi:** מחשב לוח עליו אשר ביכולתו לשלוח פקודות שונות לבקר ה-Pixhawk. על מחשב זה רצה התוכנה שפותחה במסגרת הפרויקט. ישנן כמה סיבות לבחירת מחשב לוח זה על פני מחשבים אחרים – מהיר, זול, מריץ על גביו מערכת הפעלה מלאה מבוססת Linux ומציע מגוון רחב של יציאות לשימושים שונים, ביניהן USB, GPIO, Full HDMI ועוד.

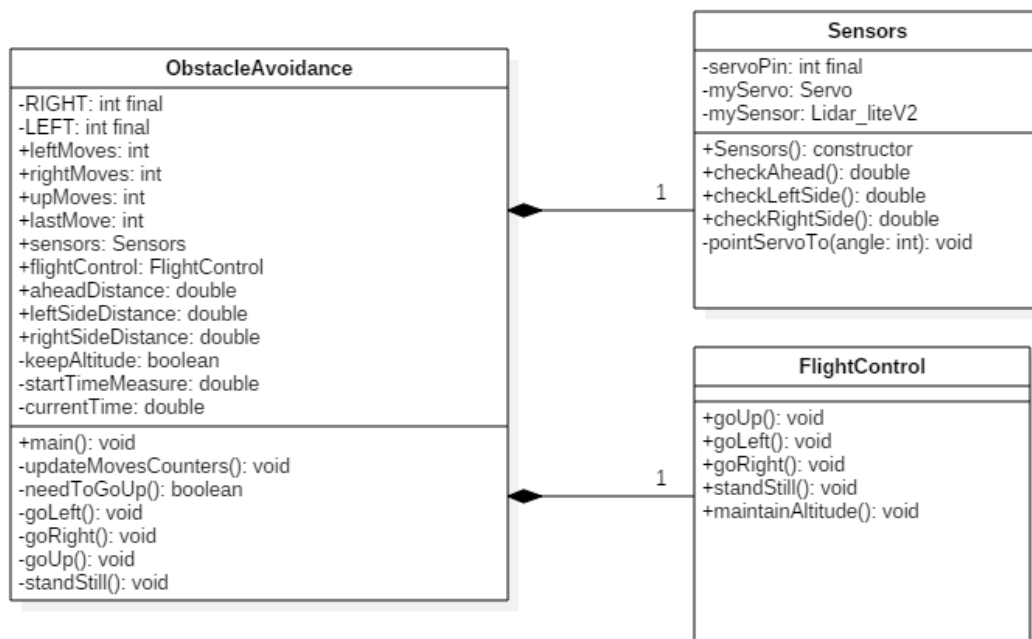
❖ **בקר ה-Pixhawk:** מערכת טייס אוטומטי המבוססת על קוד פתוח שפותחה ע"י חברת 3D Robotics. בבקר זה משולבים טכנולוגיית חיישנים ומעבד מתקדמים, מערכת הפעלה בזמן אמת המספקת ביצועים מרשימים, גמישות ושליטה לכלים אוטונומיים.

❖ **חיישני מרחק:** חיישני מרחק יסרקו את נתיב הטיסה של הרחפן באופן תמידי. נתוני המדידות ישלחו אל מחשב ה-Raspberry Pi בזמן אמת וינותחו על ידי האלגוריתם שיפותח.

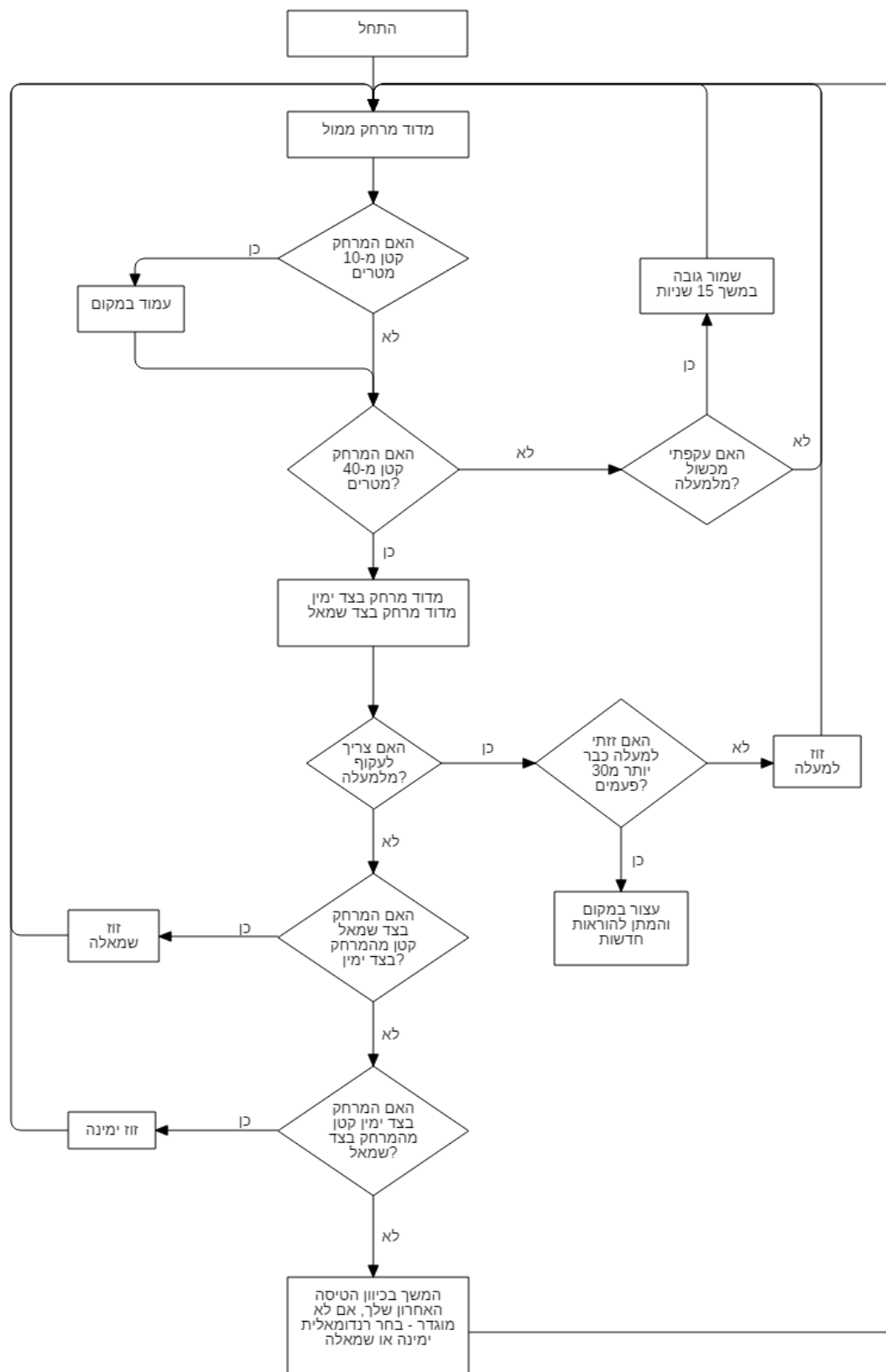
3.3 תכנון התוכנה

מערכת התוכנה תפותח בשפת Python ותרוץ על גבי מחשב ה-Raspberry Pi. התוכנה תקבל כקלט את נתוני המרחק מן החיישנים ותשלח פקודות הזזה לרחפן בהתאם על מנת להימנע מהמכשולים שזוהו על פי אלגוריתם מתאים. ישנן מספר סיבות לבחירת שפת Python בפרויקט זה:

- ❖ עובדת כנדרש עם כל סוגי הבקרים בהם נעשה שימוש במהלך פיתוח הרעיון הן בפרויקט זה והן בפרויקטים מקבילים בחברה.
- ❖ עובדת כנדרש עם ערכת הפיתוח אפליקציות לרחפנים Dronekit.
- ❖ שפת תכנות זו לא נלמדה לעומק במסגרת קורסים במהלך לימודי התואר והיה רצון ללמוד ולהתמקצע גם בשפה זו.



איור 5 : תוכן עתידי של מערכת התוכנה



איור 6 : תרשים זרימה המתאר את האלגוריתם המתוכנן להימנעות ממכשולים

4. תיאור המערכת שמומשה

4.1 תיאור הכלים אשר שימשו לפיתרון

❖ **Arduino Software (IDE)**: סביבת פיתוח קוד Arduino. בעזרת סביבה זו פותח קוד התוכנה המקבלת מדידות מחיישן ה-LidarLite v2 ומדפיסה אותן ל-Serial Port לצורך שמירתן לקובץ אשר ישמש לבדיקת האלגוריתם וביצוע סימולציה.

❖ **Notepad++**: עורך טקסט המותאם למתכנתים. ניתן לכתוב באמצעותו קוד תוכנה במגוון גדול של שפות תכנות.

❖ **PyCharm**: סביבת פיתוח קוד בשפת Python. בסביבה זו פותחה מערכת התוכנה של פרויקט זה.

❖ **StarUML**: תוכנה אשר שימשה ליצירת תרשים המחלקות של מערכת התוכנה.

❖ **מערכות הפעלה**: בפרויקט נעשה שימוש גם בסביבת Windows וגם בסביבת Linux. העבודה מול מחשב הלוח Raspberry Pi 2 נעשתה דרך סביבת העבודה של Linux, זאת מכיוון סיפקה כלים נוחים יותר להתחברות למחשב זה. לצורך פיתוח האלגוריתם ומערכת התוכנה בוצע שימוש בסביבת העבודה של Windows בעזרת סביבת פיתוח קוד Python בשם PyCharm.

❖ **Putty**: תוכנה בה נעזרתי בסביבת עבודה של Windows ליצירת חיבור עם מחשב ה-Raspberry Pi. זהו אמולטור חינמי מבוסס קוד-פתוח של טרמינל דרכו ניתן לדבר ישירות עם מחשב ה-Raspberry Pi 2 ובכך להעביר אליו קבצים, לבצע מחיקה של קבצים, לבצע התקנות/הסרות של אפליקציות ועוד.

❖ **CoolTerm**: בעזרת תוכנה זו ניתן לבצע חיבור ל-Serial Port של ה-Arduino. על ידי ביצוע החיבור ניתן לקחת את הנתונים המודפסים ולהעבירם לקובץ טקסט. תוכנה זו שימשה לשמירת הנתונים לצורך בדיקת האלגוריתם וביצוע הסימולציה.

4.2 תיאור החומרה שמומשה

4.2.1 רכיבי החומרה בהם נעשה שימוש

במסגרת פרויקט זה ופיתוח המערכת נעשה שימוש בכמה רכיבי חומרה:

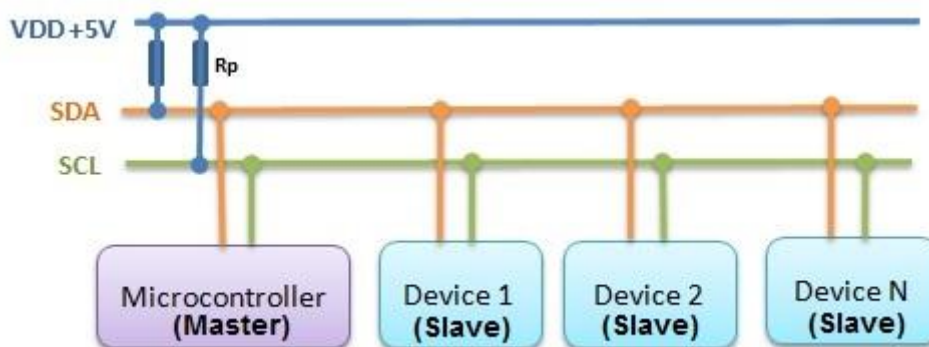
- ❖ מיקרו-בקר מסוג Arduino.
- ❖ מחשב לוח מסוג Raspberry Pi 2.
- ❖ חיישן מרחק מבוסס לייזר מסוג Lidar Lite v2. לצורך בחירת חיישן זה בוצע מחקר בנושא חיישני מרחק. את מחקר זה ואת נתוני החיישן ניתן לראות בפרק הנספחים.

למימוש הפיתרון המערכתי בפרויקט זה היה צורך לבצע חיבור בין חיישן המרחק לבין המיקרו בקר ומחשב הלוח שציונו לעיל. חיבורם נעשה באמצעות אפיק תקשורת בשם I2C עליו יורחב כעת, ואופן החיבור יוצג בהמשך.

4.2.2 אפיק תקשורת I2C

אפיק תקשורת מסוג I2C הינו אפיק תקשורת המאפשר העברת נתונים באופן טורי בין מעבד לרכיבים שונים. המעבד מנהל את התקשורת ונקרא Master בזמן שהרכיבים המחוברים אליו נקראים Slaves. בתקשורת זו ישנם שני קווים – קו נתונים דו-כיווני טורי (SDA) וקו שעון טורי חד-כיווני המופעל על ידי המעבד (על ידי ה-Master) בשם SCL.

כיוון שמדובר באפיק תקשורת טורית, לא ניתן לקחת דגימות משלושת החיישנים בו זמנית, אך מהירות העבודה הגבוהה של הרכיבים מפצה על כך ולא לא נשקפת סכנה של אי-איתור מכשול בזמן כתוצאה מכך.

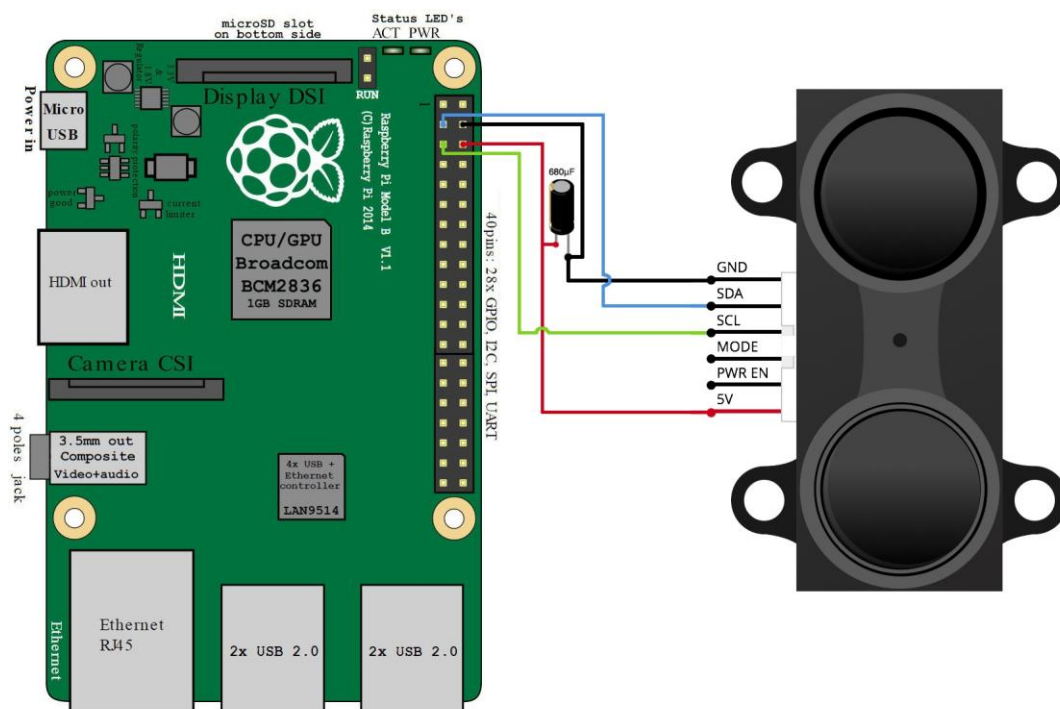


איור 7 : דיאגרמה זו מתארת את קווי התקשורת ואופן החיבור של מספר רכיבים באפיק תקשורת מסוג I2C

4.2.3 מחשב לוח מסוג Raspberry Pi 2

Raspberry Pi 2 הינו מחשב לוח עליו אשר ביכולתו לשלוח פקודות שונות לבקר הטיסה מסוג Pixhawk שברשות החברה. על גבי מחשב זה עובדת מערכת התוכנה שפותחה במסגרת הפרויקט המריצה את האלגוריתם להימנעות ממכשולים. ישנן כמה סיבות לבחירתו של מחשב לוח זה על פני מחשבי לוח אחרים – מהיר, זול, מריץ על גביו מערכת הפעלה מלאה מבוססת Linux ומציע מגוון רחב של יציאות לשימושים שונים, ביניהן USB, GPIO, Full HDMI ועוד.

מערכת התוכנה שפותחה במסגרת פרויקט זה תוכננה לעבוד עם מערך של כארבעה חיישנים מסוג Lidar Lite v2, אך עקב מחירו הגבוה של החיישן היה באפשרות החברה לרכוש חיישן רק יחיד. מסיבה זו, הדיאגרמה הבאה מתארת את אופן חיבורו של חיישן יחיד למחשב לוח זה. אך ניתן למצוא בנספחים דיאגרמה המתארת את אופן החיבור של כארבעה חיישנים למחשב הלוח.



איור 8 : דיאגרמה זו מתארת את אופן החיבור של חיישן יחיד מסוג LidarLite V2 למחשב ה- Raspberry Pi 2 באמצעות פס תקשורת מסוג I2C

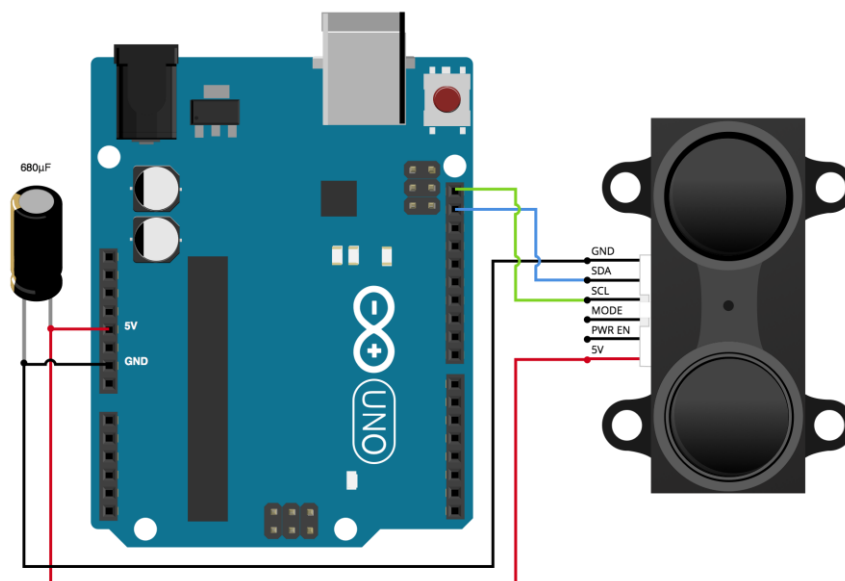
4.2.4 מיקרו-בקר מסוג Arduino

Arduino הינו מיקרו-בקר עם סביבת פיתוח משולבת (IDE) העושה שימוש בקוד פתוח ומטרתה לספק סביבה נוחה וזולה לפיתוח פרויקטים המשלבים תוכנה עם רכיבי אלקטרוניקה.

מעבר לחיבור החיישן למחשב ה-Raspberry Pi 2, היה צורך לחברו גם לבקר ה-Arduino.

עקב מחירו הגבוה של החיישן, לא הייתה באפשרות החברה לרכוש יותר מחיישן אחד. כתוצאה מכך נוצרה בעיית חוסר מידע במערכת, כיוון שזו תוכננה לעבוד עם כארבעה חיישנים. עקב בעיה זו עלה צורך למצוא דרך אחרת לבדיקת נכונות האלגוריתם. לצורך מימוש הפיתרון שנמצא לבעיה זו, בוצע חיבור בין החיישן לבקר זה. זאת על מנת לקלוט נתוני מדידות מן החיישן ושמירתם לתוך קובץ.

Basic I2C Wiring

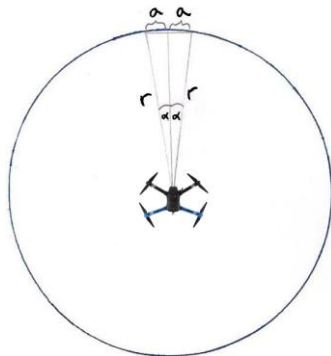


איור 9 : דיאגרמה זו מתארת את אופן חיבור חיישן ה-LidarLite V2 למחשב ה-Arduino באמצעות פס תקשורת מסוג I2C

4.2.5 מיקום החיישנים על גבי הרחפן

הרחפן בו נעשה שימוש בחברה ובמסגרת פרויקט זה הינו רחפן מסוג Iris+ ומיוצר על ידי חברת 3DR. רוחבו של הרחפן הוא כ-550mm שהם כ-0.55m. כלומר, בהתבוננות ממרכז הרחפן, הוא בולט כ-0.275m מכל צד.

קרן הלייזר של חיישן ה-Lidar Lite v2 היא בעלת זווית פתיחה של כ- $3^\circ \pm$. בהינתן זווית $\alpha = 3^\circ$ ורדיוס $r = 40$, נוכל לחשב את a על פי המשוואה הבאה:



$$\sin \alpha = \frac{a}{r} \Rightarrow a = r * \sin \alpha$$

$$\alpha = 3^\circ ; r = 40m$$

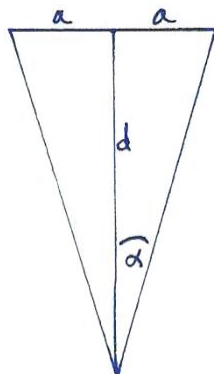
$$\Rightarrow a = 2.093m$$

משוואה 1 : חישוב רוחב השטח הנסרק על ידי חיישן מרחק יחיד

כלומר, בהסתמך על טווח הפעולה של החיישן וזווית פתיחת הקרן, ניתן להסיק כי רוחבו של השטח הנסרק על ידי החיישן הינו כ-4.18 מטרים. בהתחשב ברוחבו של הרחפן, חיישן זה מספק מסלול טיסה בטוח עם מרחב בטיחות נוסף של כ-1.8 מטרים מכל צד.

על מנת לקבוע את אופן מיקום החיישנים על גבי הרחפן, עלה צורך לביצוע מחקר נוסף בדבר רוחבו של עץ ממוצע הגדל באזורי סקי. ממצאי המחקר העלו כי עצים במקומות אלו נוטים להיות בעלי מבנה צר וגבוה, זאת על מנת למנוע הצטברות שלג על ענפיהם. על כן, נצא מנקודת הנחה כי רוחבו של עץ בוגר ממוצע עומד על כ-5 מטרים.

על מנת לבצע התחמקות יעילה ממכשולים, יש לקבוע את המרחק המינימאלי בו ייקלט מכשול על ידי 2 חיישנים לפחות. המקרה הגרוע ביותר הינו בעת שהרחפן עומד ישירות מול המכשול ובמרכזו. נתבונן על שני מקרים, אחד בו המרחק בין הרחפן למכשול עומד על כ-30 מטרים, ואחד בו המרחק עומד על כ-20 מטרים:



$$\alpha = \tan^{-1} \left(\frac{a}{d} \right)$$

$$a = 2.5m$$

$$d_1 = 20m ; d_2 = 30m$$

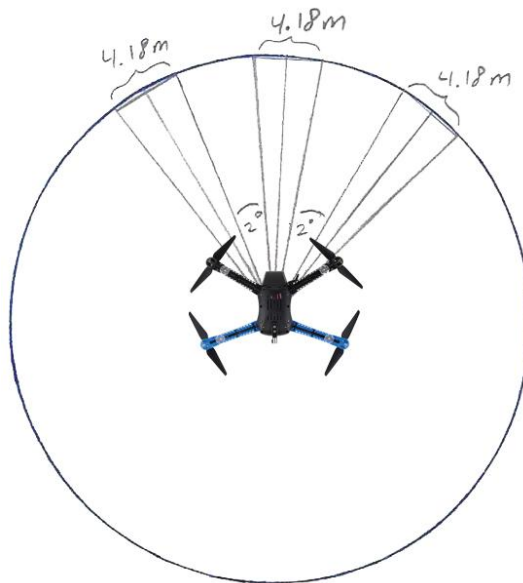
$$\alpha_1 = 7.125^\circ ; \alpha_2 = 4.763^\circ$$

משוואה 2 : חישוב זווית המרווח הדרושה בין חיישני המרחק לזיהוי מכשולים והימנעות מהם באופן יעיל

ניתן להסיק מחישובים אלו כי אם נדרוש זיהוי של מכשול על ידי 2 חיישנים לפחות במרחק של כ-30 מטרים, החיישנים ימוקמו בהפרש זווית של כ- 4.7° אחד מן השני. זווית זו הינה זווית צרה מאוד, ועלולה ליצור בעיות במרחקים קצרים יותר. מסיבה זו נדרוש כי המרחק המינימאלי לזיהוי מכשול על ידי 2 חיישנים יעמוד על כ-20 מטרים, בו הזווית בין החיישנים תעמוד על כ- 7° . כיוון שמדובר גם בחיישן לייזר בעל טווח חישה ממוקד, יש לקחת בחשבון "איזורים מתים" בהם החיישנים לא יקלטו מכשול.

כיוון שלקרן הלייזר בחיישן זה ישנה פתיחות של כ- $\pm 3^\circ$, נמקם את החיישנים בזוויות $82^\circ \pm 3^\circ$, $90^\circ \pm 3^\circ$, $98^\circ \pm 3^\circ$ וזאת בהתאם לדרישות לעיל.

פריסה זו תכסה את כמעט את כל השטח ממול הרחפן, עם שטחים מתים מינימאליים הנפרסים על כ- 2° כל אחד. בחישוב מהיר על פי הנוסחאות שהוצגו קודם לכן ניתן לדעת כי על פני מרחק של כ-40 מטרים, 2° יוצרים רוחב של כ-1.4 מטרים של "שטח מת", אך מכשולים בתחום זה יתגלו בחיישן נוסף במרחק של כ-20 מטרים מהם – מרחק מספק לביצוע תמרוני התחמקות מתאימים והימנעות מפגיעה.



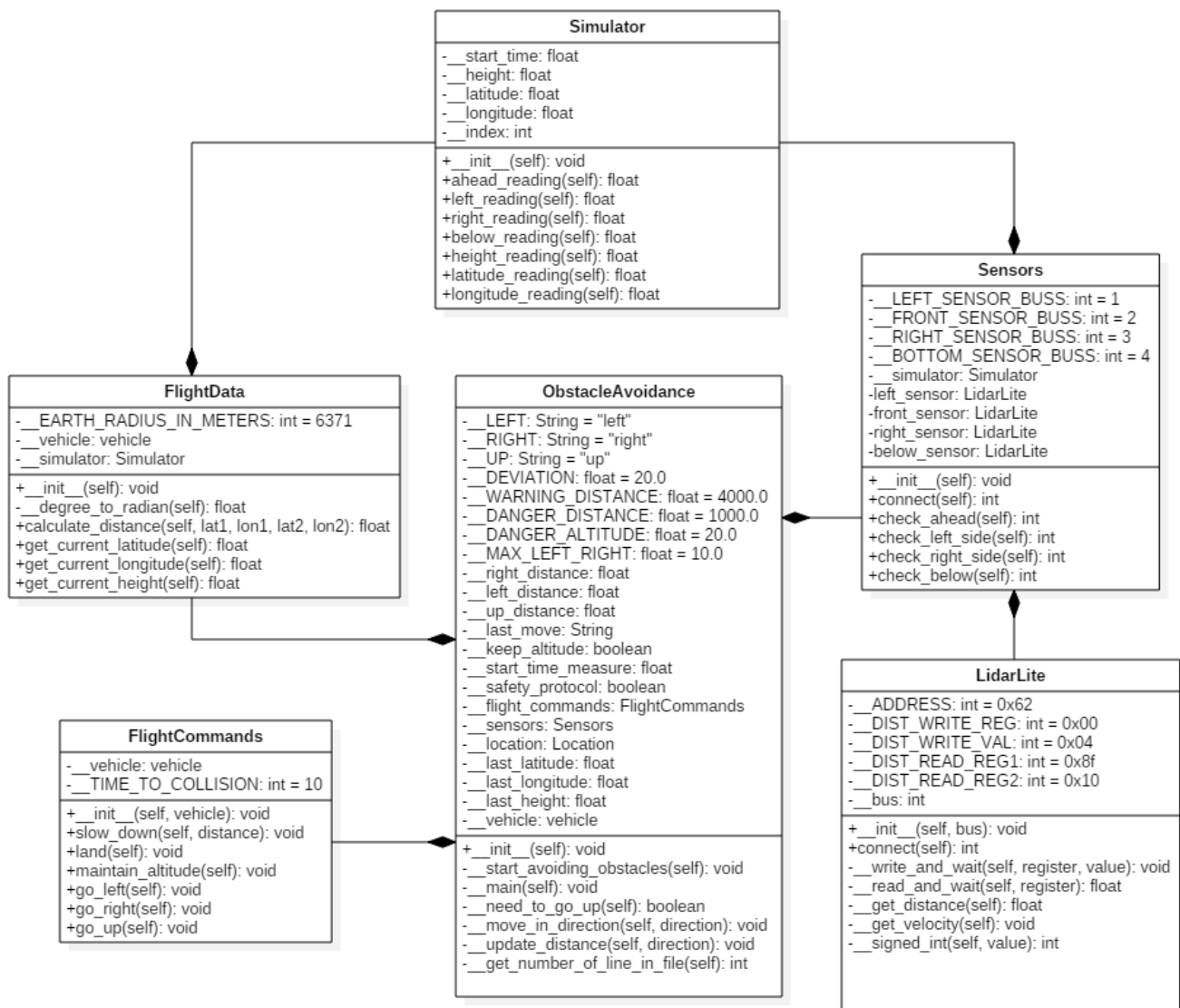
איור 10 : דיאגרמה זו מתארת את מיקום החיישנים על גבי הרחפן

בנוסף לשלושת החיישנים הללו, ישנו חיישן נוסף אשר נמצא בתחתית הרחפן ומכוון לקרקע. חלק תפקידיו של חיישן זה הם לשמור על גובה טיסה קבוע ולהתריע ממכשולים בעת הצורך לבצע נחיתה.

4.3 תיאור התוכנה שמומשה

4.3.1 תוכן המערכת שמומשה

הפיתרון התוכנתי שמומש במסגרת הפרויקט פותח בשפת Python ומורכב מכ-6 מחלקות המופרדות ל-6 מודולים שונים ולכל אחת מהן תפקיד מוגדר. ניתן לראות את מבנה המערכת בתרשים המחלקות הבא:



איור 11 : דיאגרמה זו הינה תוכן של המערכת שמומשה
(ייתכנו שינויים קלים עקב המשך פיתוח המערכת)

4.3.2 מחלקות המערכת

❖ **Obstacle Avoidance**: מחלקה זו הינה המחלקה הראשית במערכת ובה ממומש אלגוריתם ההימנעות ממכשולים. בהמשך פרק זה ניתן לראות תרשים זרימה המתאר את אופן פעולת האלגוריתם. כקלט, מקבלת מחלקה זו אובייקט מסוג vehicle אשר מייצג את הרחפן, אותו היא מעבירה למחלקות אחרות שיוצרת על מנת לקבל נתונים שונים אודות הרחפן ושליחת פקודות טיסה לתיקון מסלולו. יש לציין כי מחלקת vehicle עדיין בשלבי פיתוח על ידי גורם אחר בחברה. מסיבה זו לא ניתן לבצע בה שימוש ממשית, כעת, ופקודות העושות בה שימוש באלגוריתם מופיעות תחת הערה.

תפקודה של מערכת זו הוא קריטי עבור הרחפן, שכן עקב תפקוד לא תקין של המערכת הרחפן עומד בפני סכנת התרסקות. על כן, מערכת זו צריכה לפעול במקביל לשאר מערכות הרחפן ומסיבה זו המחלקה מייצרת Thread חדש המריץ לולאה אינסופית ובה האלגוריתם שפותח במסגרת הפרויקט.

❖ **Flight Commands**: תפקידה של מחלקה זו הוא לרכז את כלל פקודות הטיסה הדרושות במקום יחיד. המחלקה הראשית מייצרת אובייקט מסוג מחלקה זו ומעבירה אליה כפרמטר את אובייקט ה-vehicle שקיבלה. כאשר האלגוריתם במחלקה הראשית מזהה מכשול ועולה צורך לתקן את מסלולו של הרחפן, הוא קורא לפקודה המתאימה במחלקה זו, אשר קוראת לפקודה המתאימה דרך אובייקט ה-vehicle. הסיבה העיקרית ליצירת מחלקה זו היא לצורך כתיבת קוד יעיל ומוכן לשינויים עתידיים. במידה ויתבצעו שינויים בפקודות ההזזה במחלקת ה-vehicle, לא יהיה צורך לחפש את כל פקודות ההזזה במחלקה הראשית ולשנותן, אלא לבצע את השינויים המתאימים רק במחלקה זו.

❖ **Simulator**: מחלקה זו פותחה לצורך בדיקת נכונות האלגוריתם ופעולתו התקינה בלבד. על מנת שמערכת התוכנה תוכל לעבוד באופן תקין עליה לקבל כקלט נתוני מדידה מארבעה חיישני מרחק ונתונים שונים אודות הרחפן לרבות גובהו ומיקומו. עקב רכישת חיישן יחיד ואי-שלמות מחלקת vehicle נוצרה בעיית חוסר מידע. על כן, תפקידה של מחלקה זו הוא לשלוף את נתוני החיישנים מקבצים שנוצרו למען מטרה זו, ולייצר את שאר הנתונים הדרושים בעצמה. כל זאת על מנת לאפשר את פעולת המלאה של האלגוריתם לצורך בדיקת תקינותו ונכונותו.

❖ **Flight Data**: תפקידה של מחלקה זו הוא לספק נתוני גובה ומיקום של הרחפן ולעבד מידע העושה שימוש בנתונים אלו לטובת האלגוריתם המרכזי. מחלקה זו מקבלת אובייקט מסוג vehicle המייצג את הרחפן, וממנו אמורה לקבל את הנתונים להם היא זקוקה. אך מכיוון שמחלקת vehicle עוד לא הושלמה, המחלקה יוצרת אובייקט מסוג Simulator, ממנו מקבלת את הנתונים הדרושים ומעבירה למחלקה הראשית.

❖ **LidarLite**: תפקידה של מחלקה זו הוא לדלות נתונים מחיישן ה-LidarLite v2. עבור כל אחד מן החיישנים יוצרים אובייקט מסוג מחלקה. המחלקה מקבלת כפרמטר מספר המתאר את אפיק התקשורת (bus) דרכו מחובר החיישן למחשב ה-Raspberry Pi 2. במחלקה זו ישנן שתי שיטות עיקריות – אחת ליצירת החיבור עם החיישן והשנייה לקבלת המרחק הנמדד. מחלקה זו סופקה על ידי Github repository.

❖ **Sensors**: תפקידה של מחלקה זו הוא לבצע את החיבור עם שלל החיישנים המותקנים במערכת, לדלות מהם את נתוני המרחק ולהעבירם לאלגוריתם במחלקה הראשית. עקב רכישת חיישן יחיד נוצרה בעיית חוסר מידע כיוון שבמערכת הפיזית ישנו רק חיישן אחד במקום ארבעה. מסיבה זו המחלקה מייצרת אובייקט מסוג Simulator על מנת לקבל את הנתונים הדרושים ולהעבירם למחלקה הראשית.

4.3.3 אופן הפעלת המערכת

מערכת התוכנה שפותחה במסגרת פרויקט זה הינה מערכת עצמאית הפועלת בנפרד לשאר מערכות הרחפן ורצה על Thread נפרד.

כיוון שהמערכת אינה בנויה כעת עם מערך של כארבעה חיישנים ומחלקת ה-vehicle עדיין נמצאת בשלבי פיתוח על ידי גורם אחר בחברה, בוצעו בה שינויים לצורך בדיקתה. לכן, קיימים בקוד התוכנה מספר קטעים אשר יש לשנותם כאשר יעלה הצורך להפעילה כמערכת שלמה בעת רכישת חיישנים נוספים. שורות הקוד הנדרשות לפעולה מלאה ותקינה של המערכת עם ארבעה חיישנים כבר כתובות במקום בו הן צריכות להיות ומסומנות כהערה על מנת למנוע בעיות בעת בדיקת המערכת במצבה הנוכחי. לצידן ישנו תיעוד מלא כיצד יש לבצע את השינוי. פירוט על השינויים אותם יש לבצע בקוד בעת רכישת חיישנים נוספים ניתן לראות בטבלה המופיעה בדף הבא.

על מנת להפעיל את המערכת כל שיש לעשות הוא ליצור אובייקט מסוג ObstacleAvoidance באלגוריתם הכללי אשר מריץ גם את שאר מערכות הרחפן ואין צורך בביצוע פעולות נוספות מעבר לכך.

קלטים לא נכונים במערכת עלולים להוביל לקריסתה, לביצוע פעולות שגויות או לאי ביצוע פעולות כלל. לכן, עבור כל קלט לא תקין תיזרק שגיאה מתאימה ובאחריות המתכנת של שאר יישומי הרחפן להחליט כיצד לטפל בתקלות אלו.

טבלה 1 : פירוט התקלות שהמערכת יכולה לזרוק

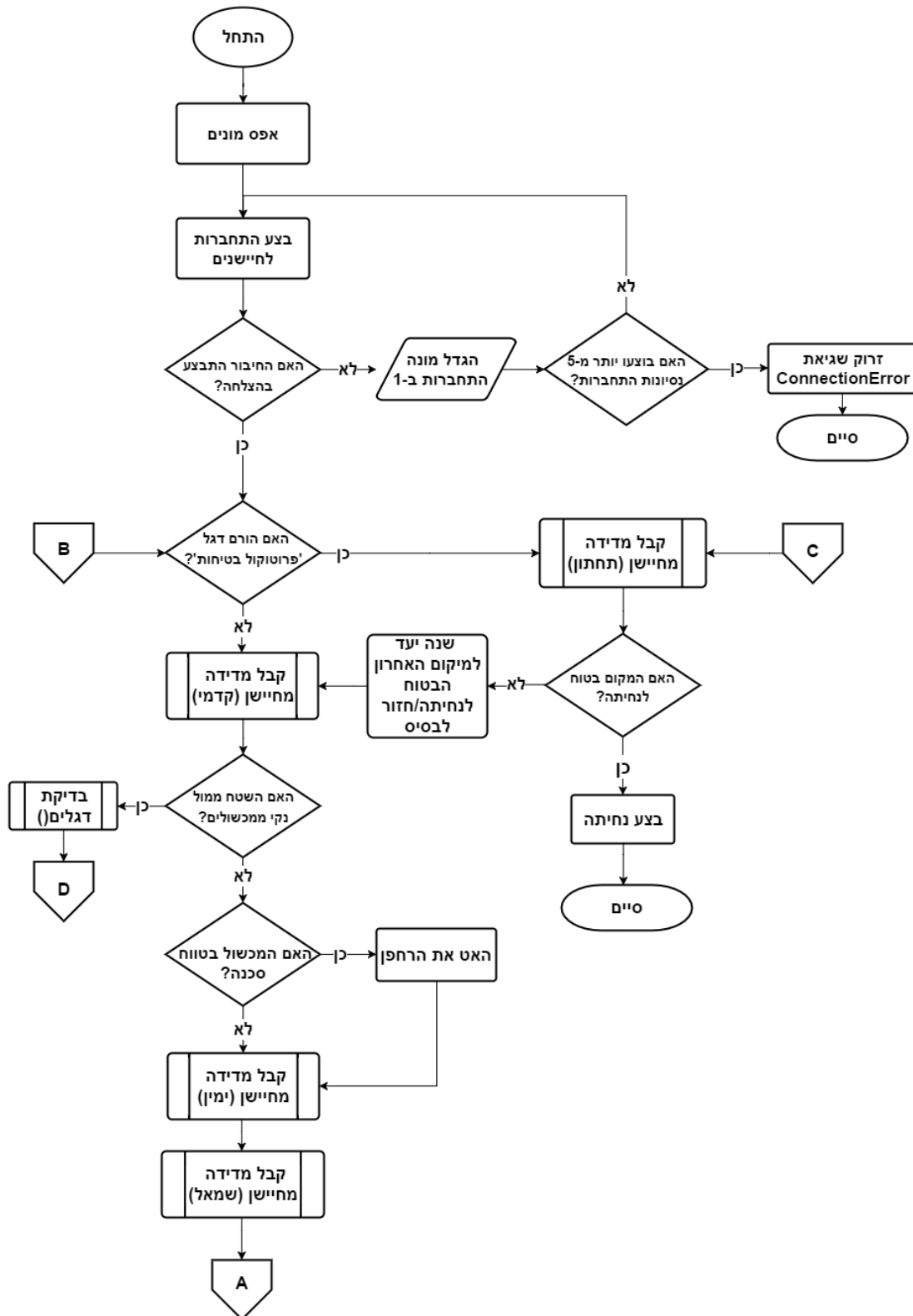
סוג התקלה	גורם התקלה
ConnectionError	תקלה בהתחברות לחיישנים
TypeError	קלט לא תקין לפונקציה/מתודה – סוג האובייקט שהועבר אינו תואם את הדרישה.
ValueError	קלט לא תקין לפונקציה/מתודה – הערך שהועבר אינו חוקי.
SystemError	התקבל מספר רב של קלטים לא חוקיים מן החיישנים ויש לבדוק את תקינותם.

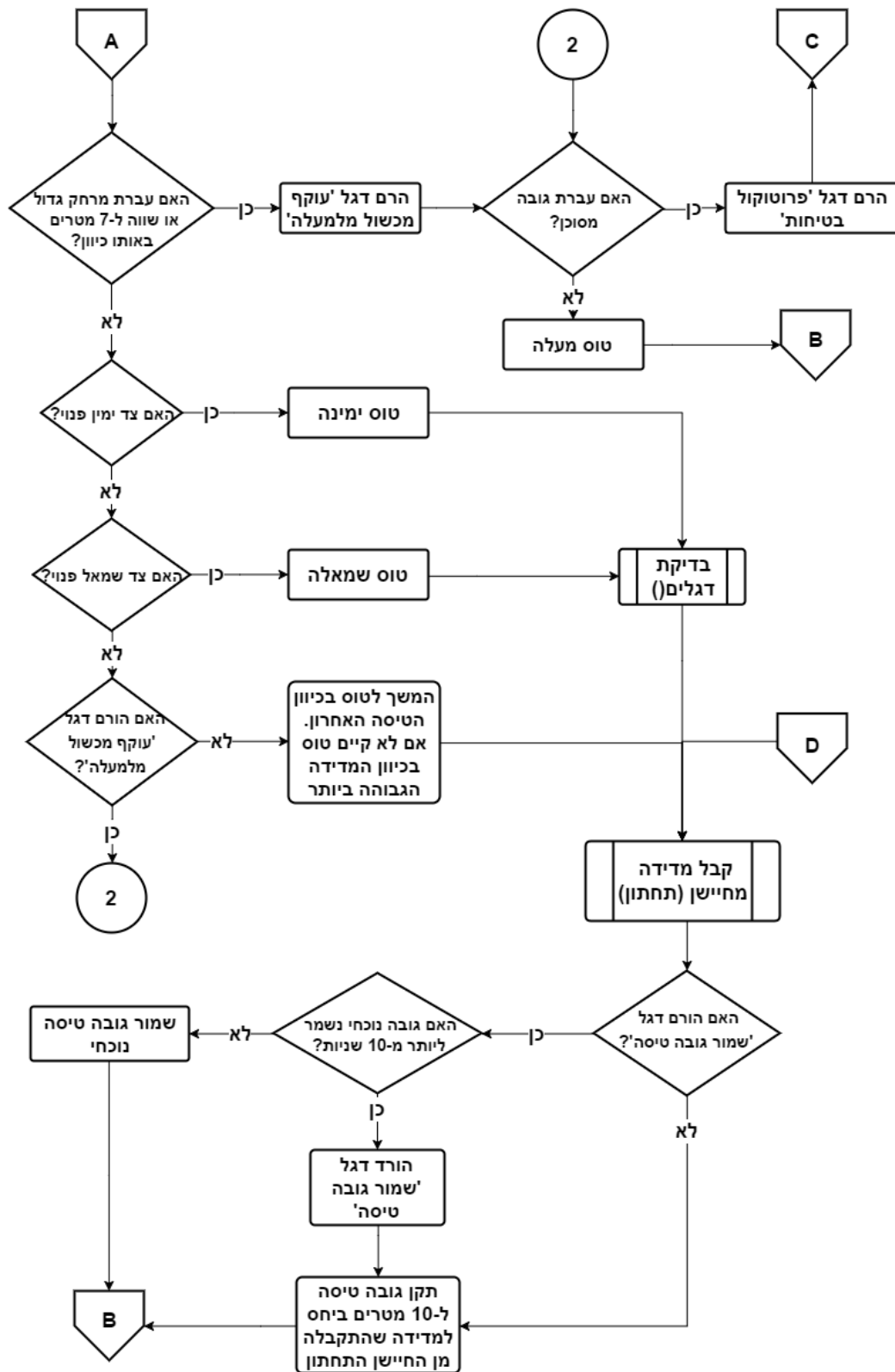
4.3.4 פירוט השינויים שיש לבצע בעת מעבר למערכת מלאה

טבלה 2 : פירוט השינויים שיש לבצע בקוד התוכנה בעת הפעלתו על מערכת מלאה עם כארבעה חיישנים

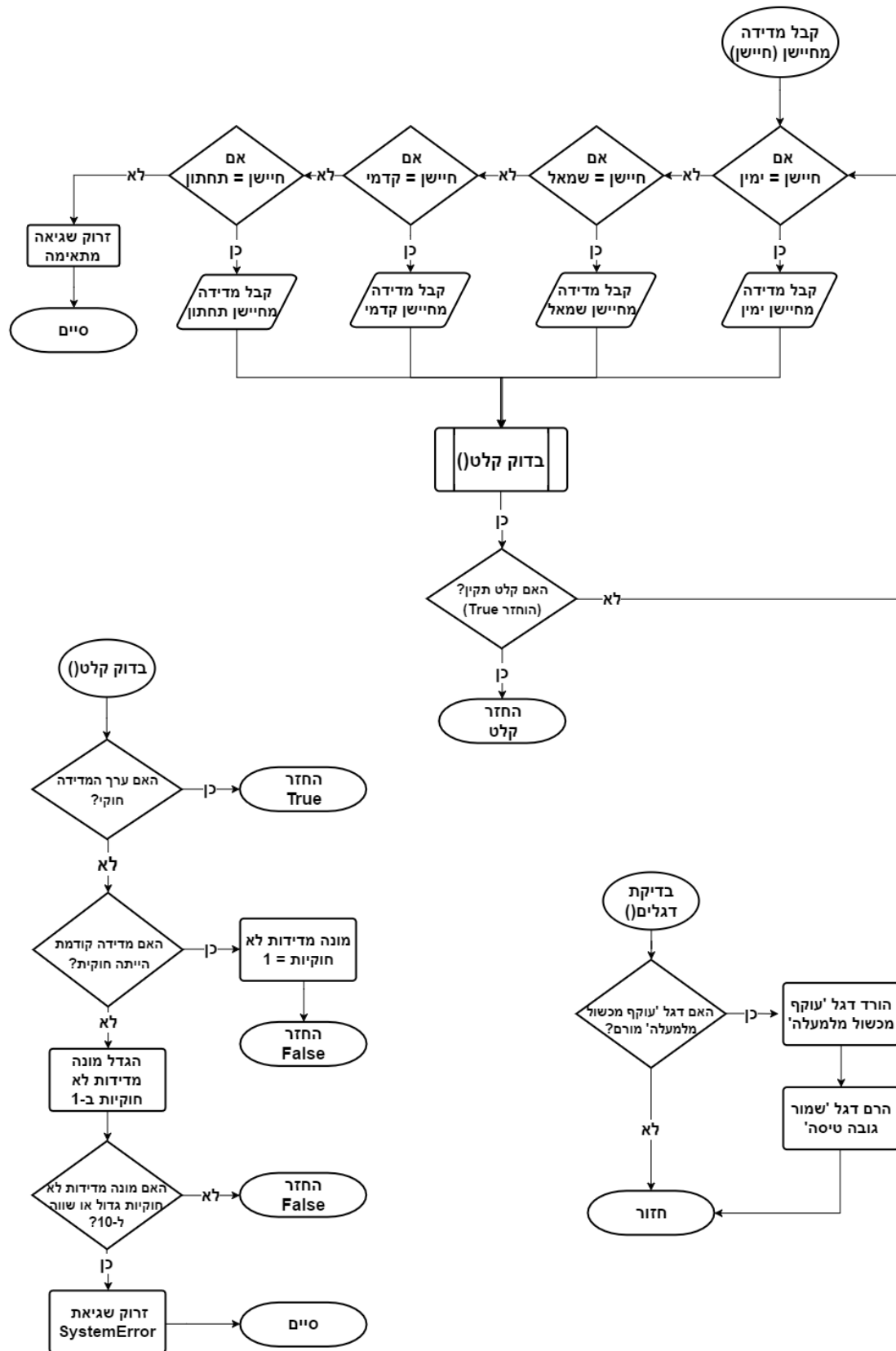
מחלקה	פונקציה/מתודה	תיאור השינוי
Obstacle Avoidance	__init__	יש לשנות את חתימת הבנאי של המחלקה כך שיקבל אובייקט vehicle.
		בעת יצירת אובייקט ה-FlightCommands יש להעביר לו כפרמטר את אובייקט ה-vehicle.
		בעת יצירת אובייקט ה-FlightData יש להעביר לו כפרמטר את אובייקט ה-vehicle.
	__start_avoiding_obstacles	יש למחוק את שורה זו. משמשת לצורך בדיקת האלגוריתם.
	__get_number_of_line_in_file	יש למחוק פונקציה זו.
Flight Commands	__init__	יש להוציא את הבנאי מההערה.
	land	יש למחוק את פקודות ההדפסה ולהוציא מהערה את פקודות ההזזה של הרחפן בעזרת אובייקט ה-vehicle.
	maintain_altitude	
	go_left	
	go_right	
	go_up	
	slow_down	למחוק את פקודת ההדפסה ולהוציא מהערה את פקודת חישוב המהירות ואת פקודת האטת הרחפן בעזרת אובייקט ה-vehicle.
FlightData	__init__	יש למחוק את הבנאי הקיים ולהוציא מהערה את הבנאי הכתוב.
	get_current_latitude	יש למחוק את השורה ששואבת נתונים מן הסימולטור ולהוציא מהערה את השורה המקבלת נתוני מיקום דרך אובייקט ה-vehicle.
	get_current_longitude	
	get_current_height	
Sensors	-	להוציא מהערה את ה-import של מחלקת LidarLite.
	__init__	להוציא מהערה את השורות המגדירות את החיישנים.
	connect	למחוק את ה-return הנוכחי ולהוציא מהערה את שאר הפקודות הרשומות.
	check_ahead	למחוק את השורה השואבת נתונים מן הסימולטור ולהוציא מהערה את זו שמקבלת נתונים מן החיישן.
	check_left_side	
	check_right_side	
	check_below	

4.3.5 תרשים זרימה המתאר את אופן פעולת האלגוריתם





איור 13: חלק ב' של תרשים זרימה המתאר את אופן פעולת האלגוריתם



איור 14 : חלק ג' של תרשים זרימה המתאר את אופן פעולת האלגוריתם

5. תוכנית בדיקות

לבחינת תפקוד המערכת נערכו סדרת ניסויים לבדיקת תקינות קלט וטיפול בקלט לא תקין וסימולציה המדמה תפקוד מלא של המערכת.

5.1 בדיקת תקינות קלט

סדרת ניסויים אלו באה לבדוק את עמידות המערכת בפני קלטים שאינם נכונים ואינם תואמים לאלגוריתם. טיפול לא תקין בקלטים לא נכונים עלול להוביל לקריסת המערכת, לביצוע פעולות שגויות או לאי ביצוע פעולות כלל. לכן, עבור כל קלט לא תקין תיזרק שגיאה.

5.1.1 מחלקת ObstacleAvoidance

פעולה	תוצאה צפויה	תוצאה בפועל
העברת אובייקט שאינו vehicle לבנאי: __init__(self, vehicle)	המערכת תזרוק שגיאת TypeError.	לא ניתן לבדיקה עקב אי שלמות מחלקת vehicle.
העברת ערך שונה מן הקבועים המסמלים את הצדדים (LEFT,RIGHT,UP) למתודה: __move_in_direction(self, direction)	אם הערך המוזן הוא מסוג String אך ערכו לא תואם, המערכת תזרוק שגיאת ValueError. אחרת המערכת תזרוק שגיאת TypeError	בהתאם לציפיות
העברת ערך שונה מן הקבועים LEFT,RIGHT,UP למתודה: __update_distance(self, direction)	אם הערך המוזן הוא מסוג String אך ערכו לא תואם, המערכת תזרוק שגיאת ValueError. אחרת המערכת תזרוק שגיאת TypeError	בהתאם לציפיות

5.1.2 מחלקת FlightCommands

פעולה	תוצאה צפויה	תוצאה בפועל
העברת אובייקט שאינו vehicle לבנאי: __init__(self, vehicle)	המערכת תזרוק שגיאת TypeError.	לא ניתן לבדיקה עקב אי שלמות מחלקת vehicle.
העברת ערך קטן מ-0 או שאינו מסוג float או int למתודה: slow_down(self, distance)	עבור ערך קטן מ-0 תזרוק המערכת שגיאת ValueError. עבור ערך שאינו מסוג int או float תזרוק המערכת שגיאת TypeError	בהתאם לציפיות

5.1.3 מחלקת FlightData

תוצאה בפועל	תוצאה צפויה	פעולה
לא ניתן לבדיקה עקב אי שלמות מחלקת vehicle.	המערכת תזרוק שגיאת TypeError.	העברת אובייקט שאינו vehicle לבנאי: __init__(self, vehicle)
בהתאם לציפיות	המערכת תזרוק שגיאת ValueError	העברת ערך מספרי לא תקין למתודה calculate_distance(self,lat1,lon1,lat2,lon2)
בהתאם לציפיות	המערכת תזרוק שגיאת TypeError.	העברת ערך שאינו מסוג float למתודה calculate_distance(self,lat1,lon1,lat2,lon2)

במחלקות Simulator, Sensors, LidarLite אין מתודות/פונקציות המקבלות ערכים ממקור חיצוני ועל כן אין צורך בביצוע בדיקות קלט למחלקות אלו.

תוצאות ניסוי זה מראות כי המערכת עמידה כנדרש בפני קלטים לא תקינים ושולחת שגיאות מתאימות לאופי התקלה.

5.2 סימולציית טיסה

בעיית חוסר המידע שעלתה בעקבות רכישת חיישן יחיד העלתה צורך לבצע בדיקה למערכת בצורה תוכניתית. לכן, לצורך בדיקת האלגוריתם בוצעה הדמיה של מסלול טיסה עם מכשולים. לביצוע המשימה חובר חיישן המרחק למיקרו-בקר מסוג Arduino, זאת על מנת לבצע קליטה של נתוני החיישן לתוך קובץ טקסט.

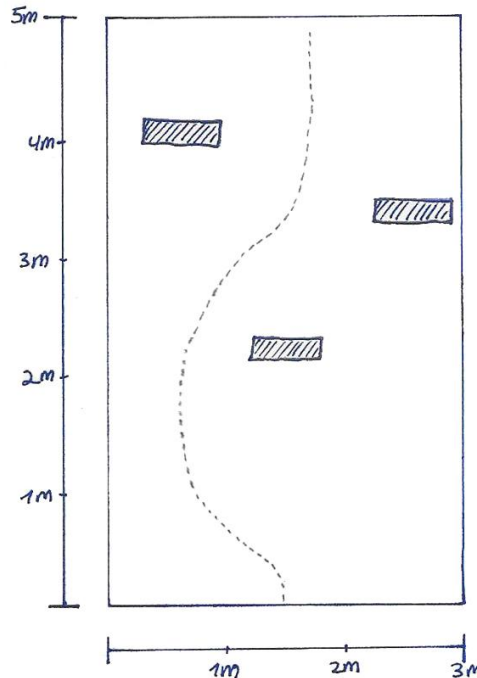
החיישן יוזז יחד עם מחשב נישא לאורכו של מסלול שתוכנן מראש המדמה מסלול טיסה אפשרי. הזזת החיישן במסלול תבוצע שלוש פעמים כאשר בכל פעם ישמרו נתונים המייצגים חיישן אחד בקובץ טקסט נפרד (כמספר החיישנים המתבוננים קדימה במערכת).

לאחר סיום איסוף הנתונים ובעת הרצת הבדיקה יקבל האלגוריתם את הנתונים על ידי מחלקת Simulator אשר שואבת את הנתונים המתאימים מן הקבצים ומזרימה אותם לאלגוריתם.

יצירת הסימולציה בקנה מידה מלא היא משימה קשה מאוד עד בלתי אפשרית, על כן יש לצמצם את קנה המידה כך שיאפשר את ביצוע הבדיקה. עקב אילוצי מקום לביצוע הבדיקה, היא נעשתה בקנה מידה של 1:8 ועל פי קנה מידה זה תוכננו והותאמו גם שאר פרטי הניסוי:

- ❖ טווח הזיהוי של החיישן מוגדר כעת על כ-5 מטרים במקום 40 מטרים.
- ❖ מרחק הסכנה ממכשול מוגדר כעת על כ-1.25 מטרים במקום 10 מטרים.
- ❖ רוחבו של מכשול יעמוד על כ-0.5-0.6 מטרים במקום 5 מטרים.

השרטוט הבא מדגים את אופן ביצוע הבדיקה ומיקום המכשולים. הקו המקווקו מציין את המסלול הנצפה מן האלגוריתם ללכת לפיו ואת המסלול בו הוזז החיישן לקליטת הנתונים.



איור 15 : דיאגרמה זו מתארת את המסלול שתוכנן לסימולציית הטיסה

תוצאות הבדיקה מובאות בצורת הדפסות על גבי ה-Console של סביבת העבודה המתארות את התנהגות האלגוריתם והפקודות ששולח לרחפן.

סקירת תוצאות הבדיקה מעלות כי האלגוריתם מתפקד בדיוק כפי שנצפה ממנו. הרחפן המדומה החל את דרכו בטיסה ישרה. משהבחין במכשול המתקרב בדק ומצא כי צידו השמאלי פנוי יותר לעומת צידו הימני ובהתאם לכך החל לתקן את מסלולו וטס שמאלה. לאחר עקיפת המכשול משמאל הפסיק הרחפן המדומה לקלוט כי ישנו מכשול מלפניו והמשיך בכיוון הטיסה שלו. בעת שזיהה מכשול נוסף, הבחין כי צידו הימני פנוי יותר והחל לפנות לעברו. משחלף את המכשול השני הבחין כי אין מולו מכשול נוסף והמשיך בכיוון הטיסה שלו.

6. סקירת עבודות דומות בספרות והשוואה

- ❖ **Radiation Mapping UAV:** <https://github.com/plusangel/radiationUAV>
פרויקט זה מציג רחפן שיועד לסרוק, למפות ולנתר קרינה באתרי גרעין. נעשה בו שימוש בטכנולוגיית הימנעות ממכשולים המתבצעת באמצעות חיישן מרחק מבוסס לייזר בו נעשה שימוש בפרויקט זה.
- ❖ **Autonomous plane with obstacle detection & avoidance system using a camera:** <http://github.com/andybarry/flight>
בפרויקט זה מוצג טיסן בעל מערכת לזיהוי והימנעות ממכשולים המתבססת על מצלמה. במערכת זו נעשה שימוש במצלמה יקרה, אך זו מאפשרת לטיסן לנוע בביטחון במהירות של כ-30 מייל לשעה (המהירות הדרושה בפרויקט זה).
- ❖ **Quadcopter obstacle avoidance system using Pixhawk and a combination of ultrasonic & Infrared sensors (video and article):** https://www.researchgate.net/publication/284724052_Relative_position-based_collision_avoidance_system_for_swarming_UAVS_using_multi-sensor_fusion
במאמר זה מוצג הפיתוח של רחפן המיישם אלגוריתם להימנעות ממכשולים מ-4 כיווני טיסה על ידי תמרוני טיסה במרחב דו-מימדי תוך שימוש בשילוב של חיישני מרחק מבוססי אולטרא-סוני וחיישני מרחק מבוססי אינפרא אדום.
- ❖ **Flying Fast and Low among Obstacles: Methodology and Experiments:** https://www.ri.cmu.edu/pub_files/pub4/scherer_sebastian_2008_1/scherer_sebastian_2008_1.pdf
בפרויקט זה מוצג רחפן שביכולתו לבצע טיסה מהירה גובה נמוך תוך כדי הימנעות ממכשולים. ישנה עבודה תיאורטית רבה במאמר זה הכוללת חישובים רבים ובמידת הצורך אוכל להיעזר בה ליישום האלגוריתם.

7. מסקנות מהמימוש ומהפרויקט

תהליך פיתוח המערכת לווה בחשיבה רבה ותכנון ארוך על מנת ליצור מערכת יעילה העונה על דרישות החברה. לאורך הפיתוח בוצעו מחקרים בקשת רחבה של נושאים תוך התחשבות בדרישות החברה ובאילוצים שונים לרבות תקציב וחומרה מוגבלים. כיוון שהפרויקט פותח במסגרת חברת סטארט-אפ, תהליך הפיתוח לווה בקשיים רבים עקב השינויים התכופים שהיו, בעיקר בחומרת המערכת.

לאחר סיום תהליך הפיתוח של המערכת ניתן להסיק מסקנות המתייחסות להיבטים שונים בפרויקט, עליהן יפורט בפרק זה.

7.1 חומרה

בתחילת העבודה על הפרויקט היה ברשות החברה מחשב לוח מסוג Intel Edison, אך לא היה ברשותם את הידע כיצד לעבוד עמו. על כן היה צורך לבצע מחקר ותהליך למידה אשר גזל זמן רב על אופן העבודה עם מחשב לוח זה. לאחר תהליך הלמידה הארוך, הוחלט על ידי גורם אחר בחברה לעבוד עם מחשב לוח אחר מסוג Raspberry Pi 2 בעקבות קשיים בהתממשקות עם בקר הטיסה של הרחפן. בעקבות זאת עלה צורך לבצע את כל תהליך למידת החומרה שנית, דבר שגרר עיכוב נוסף. ניתן להסיק כי תהליך פיתוח הפרויקט התעכב רבות עקב חוסר הידע במחשבי לוח אלו. ייתכן כי אם ניתן היה למצוא מומחה המכיר את מחשב ה-Intel Edison, תהליך הלמידה היה פשוט יותר, גורר עמו פחות עיכובים וייתכן שכלל לא היה עולה צורך להחליפו במחשב ב-Raspberry Pi 2.

בוצע מחקר על חיישני מרחק במטרה לבחור את החיישן המתאים ביותר לדרישות הפרויקט ולהזמין. הזמנת החיישן מחו"ל גררה עיכוב נוסף בתהליך הפיתוח שכן לא ניתן היה לקבל נתונים ולבדוק את התנהגות האלגוריתם שהחל להתפתח. בנוסף, עלותו של החיישן שנבחר כמתאים לדרישות הפרויקט הייתה גבוהה מן המצופה. על כן, מערך של כארבעה חיישנים יגרור מחיר גבוה מאוד לצרכן המעוניין לרכוש את המערכת ועלול לבטל את עניינו בה. ניתן להסיק כי ייתכן שהתייעצות עם מומחים בנושא החיישנים הייתה מובילה לרכישת חיישנים מסוג אחר וזול יותר אשר הייתה גורמת למערכת להיות יותר אטרקטיבית עבור המשתמש.

בעקבות העלות הגבוהה של החיישן נרכש רק חיישן יחיד. פיתוח הפרויקט עם חיישן יחיד גרר בעיה בבדיקת נכונות האלגוריתם, שכן לא הייתה אפשרות להרכיב את המערכת על גבי הרחפן ולבצע בדיקות חיות לנכונותה. ניתן להסיק כי הנמכת הדרישות על ידי החברה מן החיישן לטווח פעולה קצר יותר היה מוביל לבחירת חיישן זול יותר. בבחירת חיישן זול ניתן היה לרכוש כארבעה חיישנים כפי שנדרש לתפקוד מלא של המערכת ובכך לבצע בדיקה יסודית יותר לנכונות האלגוריתם.

7.2 תוכנה

הקשיים שליוו את הפרויקט בהיבט החומרה גררו עיכוב גדול בפיתוח מערכת התוכנה, זאת עקב אי ידיעת שפת התכנות וסביבת העבודה הדרושות. בשלבים הראשונים של הנעת הפרויקט והגשת ההצעה, התכנון היה לתכנת בשפת C++ ו C++ כיוון שהפלטפורמה איתה נעשתה העבודה הייתה בקר ה-Pixhawk. לכן נעשו ניסיונות התממשקות עם הבקר ובוצע ומחקר על אופן העבודה איתו שלא צלח. לאחר זמן מה הובא לידיעתם של מייסדי החברה כי שיטת העבודה בה יש לשנות את קוד הבקר אינה נכונה וכי יהיה קשה לבצע משימה זו. על כן הוחלט על עבודה עם מחשב הלוח מסוג Intel-Edison שלאחר פרק זמן נוסף הוחלף גם הוא במחשב לוח מסוג Raspberry Pi. בעקבות שינויי הפלטפורמות הרבים היה קושי רב בכתיבת מערכת התוכנה הדרושה. ניתן להסיק כי אם היה נעשה מחקר מקדים בנושא על ידי מייסדי החברה תסבוכות אלו היו נחסכות מלכתחילה.

בתחילת העבודה על הפרויקט בשלבי ההתנעה וההצעה הוא תוכנן להתבצע בזוג. בעקבות דרישות חדשות שצצו לחברה, הפרויקט תפח למימדים גדולים מדי, מה שגרר פיצול של הפרויקט לשני פרויקטים נפרדים. ככל שהתקדם תכנון האלגוריתם וכתיבת הקוד התברר עד כמה אלגוריתמים להימנעות ממכשולים מסובכים. פיצול הפרויקט הוביל לקושי גדול בפיתוח שלו כיוון שלא הייתה אפשרות לסייעור מוחין ולהתייעצות של ממש עם אדם נוסף ברמת הקוד והאלגוריתם. המסקנה הנובעת מכך היא שבמידה ויהיו פרויקטים דומים בעתיד, מומלץ לצוות להם צמד מפתחים ולא אדם יחיד.

עקב עלותו הגבוהה של חיישן המרחק שנבחר לפרויקט זה נרכש רק חיישן יחיד על ידי החברה. דבר זה הוליד עימו בעיה תוכנתית גדולה – בדיקת האלגוריתם תחת בעיית חוסר מידע. על כן הושקע זמן רב על מציאת פיתרון מתאים ויישומם בדמות סימולציה. ניתן להסיק מכך כי על ידי הנמכת דרישות החברה לחיישן עם טווח חישה נמוך יותר ניתן היה לבצע רכישה של מספר חיישנים ולבצע בדיקה מלאה של המערכת ללא צורך בפיתוח פיתרון תוכנתי בדמות סימולציה. בדרך זו ניתן היה להגיע לתוצאות טובות יותר, לבצע בדיקה יותר מקיפה לתפקוד האלגוריתם בסיטואציות שונות וייתכן שאף להרכבת המערכת פיזית על גבי הרחפן.

8. נספחים

8.1 ביבליוגרפיה

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

Laser Sensors for Robotic Applications:

<http://www.smashingrobotics.com/laser-sensors-for-robotic-applications/>

Style guide for Python code: <https://www.python.org/dev/peps/pep-0008/>

Learn Python: <http://www.learnpython.org/>

Intel Edison as a companion computer: <http://ardupilot.org/dev/docs/edison-for-drones.html>

Loading Debian (Ubilinux) on the Intel Edison:

<https://learn.sparkfun.com/tutorials/loading-debian-ubilinux-on-the-edison#enable-wifi>

Comparison of Aerial Collision Avoidance algorithms:

<http://etd.auburn.edu/bitstream/handle/10415/2976/holtThesis.pdf?sequence=2&ts=1452085785924>

I2C Communication port installation guide for Raspberry Pi:

<http://skpang.co.uk/blog/archives/575>

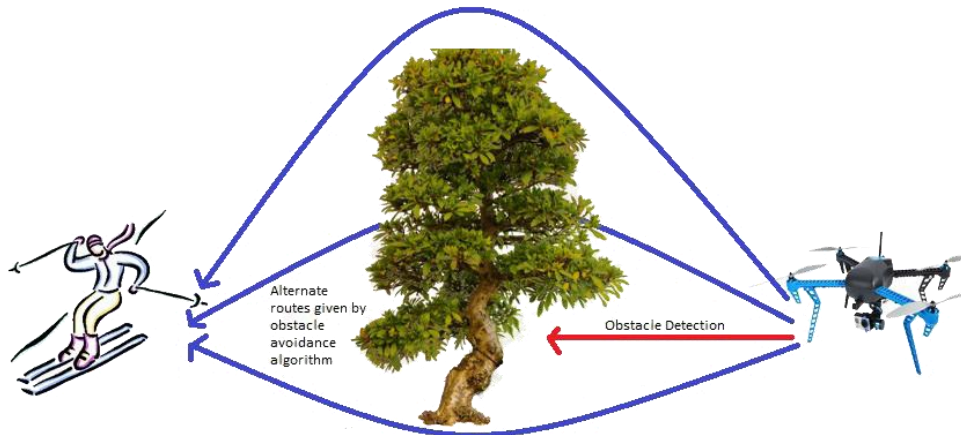
Connecting Raspberry Pi and Arduino with serial USB cable:

<https://oscarliang.com/connect-raspberry-pi-and-arduino-usb-cable/>

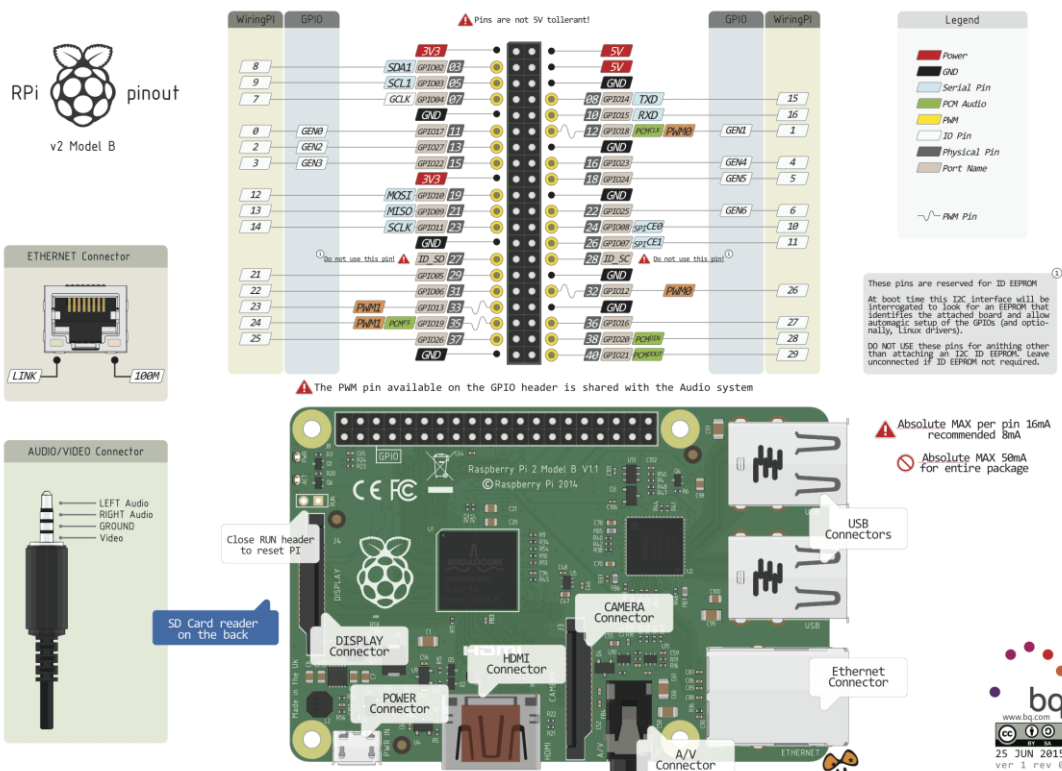
Calculate distance between two GPS coordinates: <http://www.movable-type.co.uk/scripts/latlong.html>

Online tool for creating flow charts: <https://www.draw.io/>

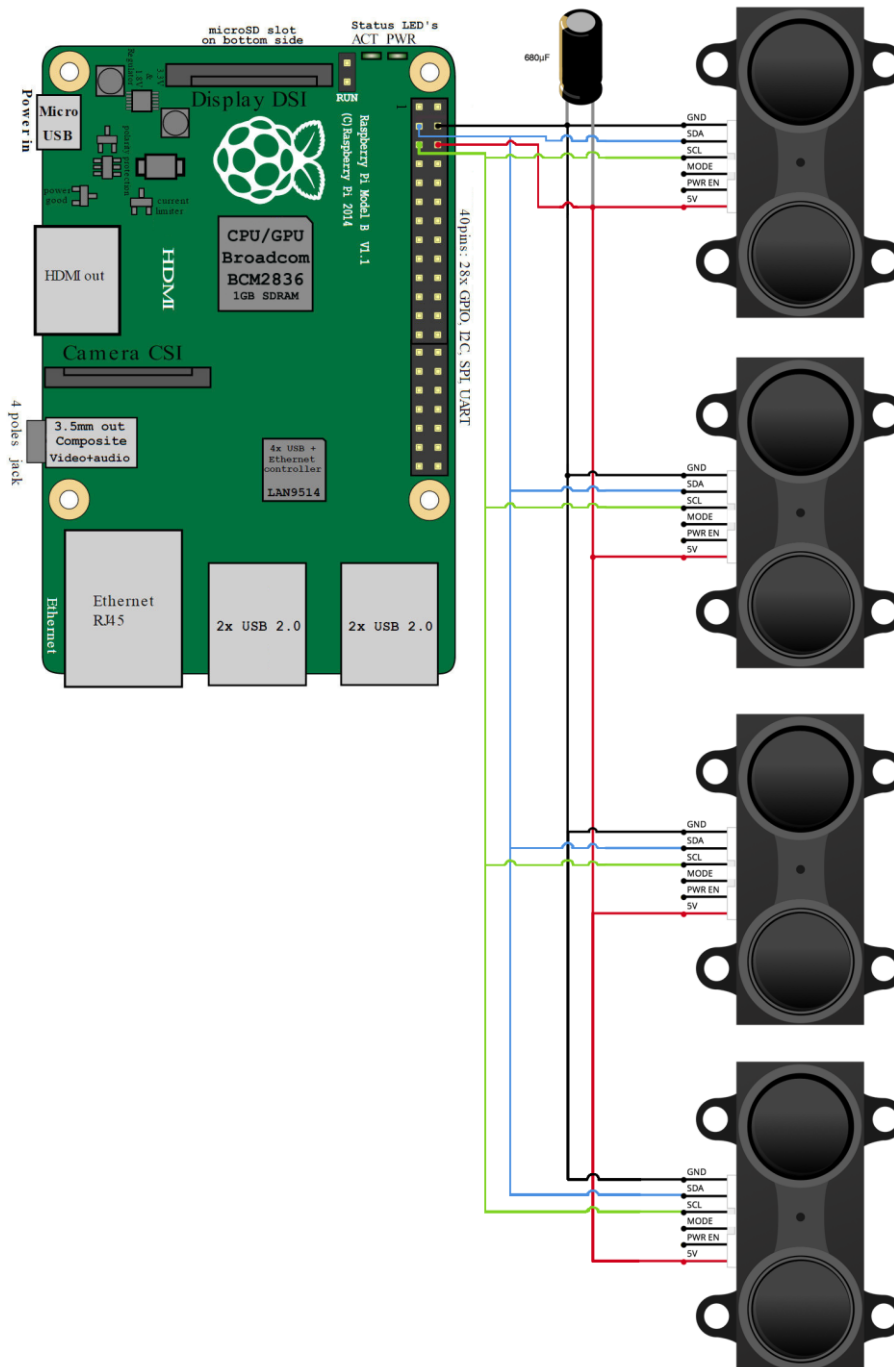
8.2 תרשימים וטבלאות



איור 16 : עיקרון פעולת האלגוריתם להימנעות ממכשולים. ראשית ינסה הרחפן לעקוף את המכשול מצדדיו, ובמידה ולא הצליח, ינסה לעקוף אותו ממעל.



איור 17 : דיאגרמה זו מתארת את חיבורי לוח ה-Raspberry Pi 2



איור 18 : דיאגרמה זו מתארת את אופן חיבור החיישנים ומחשב ה-Raspberry Pi 2 במערכת המשלבת כארבעה חיישנים

8.3 תכנון הפרויקט

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

8.4 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 ולבזבז סוללה.	5/5	ניתוח זמן הריצה של הקוד שכתבנו וכתביבתו 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.

8.5 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.

8.6 חישוב זמן-מרחק-דרך

בעת שזיהה הרחפן מכשול במרחק מסוכן (פחות מ-10 מטרים), יאט הרחפן את מהירותו בהתאם על מנת לספק לו כ-10 שניות לפגיעה. זמן זה צריך להספיק לו לעקוף את המכשול בבטחה. ברגע שעבר את המכשול, יחזור הרחפן למהירותו המקורית.

את המהירות הדרושה ניתן לחשב על פי הנוסחא הבאה:

משוואה 3 : נוסחאת חישוב מהירות-זמן-דרך

$$T = \frac{D}{V} \Rightarrow V = \frac{D}{T}$$

בנוסחא זו ידועים ערכי המשתנים המייצגים את הזמן ואת המרחק. על כן חישוב המהירות הדרושה הינו חישוב פשוט.

8.7 חישוב מרחק בין שתי נקודות ציון במרחב

במערכת התוכנה שפותחה עלה צורך לביצוע חישוב המרחק שעבר הרחפן בעת שטס בכיוון מסוים. יש צורך בחישוב זה על מנת לקבוע האם נעשה כל ניסיון אפשרי לעקוף את המכשול מצדדיו ואם יש צורך לנסות לעקוף אותו מלמעלה.

על מנת לבצע את החישוב הנדרש יש לקבל את נתוני המיקום של הרחפן בעת הנוכחית ומיקומו הקודם בו טס באותו כיוון.

כל נקודת ציון במרחב מאופיינת על ידי 2 פרמטרים – קו רוחב וקו אורך (latitude ו-longitude בהתאמה) וחישוב המרחק ביניהן נעשה על פי הנוסחא הבאה:

משוואה 4 : נוסחת חישוב מרחק במטרים בין שתי נקודות ציון במרחב

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi)_1 * \cos(\varphi)_2 * \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c$$

8.8 חיישני מרחק

חיישנים הם עצמים שמטרתם היא לזהות אירועים או שינויים בסביבתם ולספק פלט מסוים בהתאם לדרישה. ישנם סוגים רבים של חיישנים המיועדים למטרות שונות ואופן פעולתם משתנה על פי הצורך והמטרה אשר לשמה הם יועדו. פרויקט זה מתמקד בחיישנים אשר מודדים מרחק, וזאת על מנת לזהות עצמים ומכשולים בדרכו של הרחפן. ישנם סוגים שונים של חיישני מרחק אשר ייעודם ויכולותיהם משתנות בהתאם לסוגם העיקרון הפיסיקלי על פיהם הם עובדים. פרויקט זה עוסק ברחפן אשר טס באופן אוטונומי ומצלם אדם הגולש סקי. על כן, על מנת לבחור את החיישן המתאים ביותר יש לקחת בחשבון מספר פרמטרים:

- ❖ קביעת טווח הפעולה הדרוש.
- ❖ מחיר.
- ❖ גודל ומשקל.
- ❖ מגבלות פעולה.

8.8.1 קביעת טווח הפעולה הדרוש

על מנת לקבוע את טווח הפעולה הנדרש מן החיישן, יש לקבוע מהי מהירות תנועתו של גולש סקי ממוצע. לאחר בדיקה העלו הממצאים כי גולש סקי ממוצע נע במהירות של כ-43 קילומטרים בשעה (קמ"ש) עם סטייה אפשרית של כ-11.2 קמ"ש. לאחר התייעצות עם גורמים רלוונטיים בחברה התקבלה החלטה כי הרחפן ינוע בעקבותיו של גולש סקי ממוצע במהירות של כ-43 קמ"ש וכי ניתן להתעלם מן הסטייה בפרויקט זה. טווח החיישן נמדד במטרים, לכן יש לקבוע כמה מטרים יעבור הרחפן בשנייה במהירות טיסה מירבית.

משוואה 5 : המרת יחידות מקילומטרים לשעה למטר לשנייה

$$vKph = v * \frac{1000m}{3600s} = \frac{v}{3.6} Mps$$

v - Velocity | Kph - Kilometer Per Hour | m-meter | s-Second | Mps-Meter Per Second

כלומר:

משוואה 6 : חישוב מהירות הרחפן במטרים לשנייה

$$43Kph = 43 * \frac{1000m}{3600s} = \frac{43}{3.6} Mps = 11.944 \approx 12Mps$$

מסקנה: 43 קילומטרים לשעה הם שווי ערך ל-12 מטרים בשנייה. כעת משידוע כי הרחפן עובר כ-12 מטרים בשנייה במהירות מירבית, נדרוש כי טווח הפעולה המינימאלי של החיישן יהיה כ-36 מטרים, זאת על מנת לספק לרחפן לפחות כ-3 שניות להגיב ולתקן את מסלולו.

8.8.2 סוגי חיישני מרחק

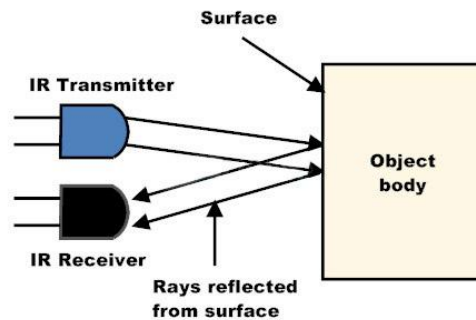
❖ חיישן אינפרא אדום (Infrared)

האור הנראה לעין האנושית משתרע על פני תחום הצבעים בין אדום לסגול, אך בנוסף לצבעים אלו קיימים שני תחומי אור אשר העין האנושית אינה מסוגלת לראות. אחד מתחומי אור אלו הוא תחום האינפרא אדום.

חיישנים מסוג זה פועלים על פי עיקרון של שידור וקליטה של גלי אור. לחיישן שני רכיבים – משדר ומקלט, האחד משדר את קרן האור האינפרא אדומה והשני קולט אותה. המרחק מן העצם מחושב על פי הפרש הזמן שבין הגל המשודר לגל החוזר הנקלט.

חיישנים מסוג זה מתחלקים לכמה סוגים כתלות במוצא שלהם, נתמקד בשניים עיקריים:

1. מוצא בינארי – מתריע רק במידה והוא מזהה עצם כלשהו בטווח הקליטה שלו, אך לא מספק את המרחק בינו לבין אותו העצם שזיהה.
2. מוצא אנלוגי – מספק את המרחק הנמדד בין החיישן לעצם הנמדד.

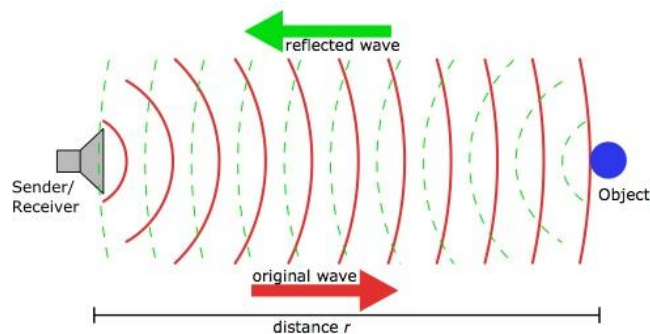


איור 19 : עיקרון הפעולה של חיישן אינפרא אדום. המשדר שולח קרני אינפרא אדום אשר פוגעות בעצם, מוחזרות ממנו ונקלטות חזרה במקלט

❖ חיישן אולטרה-סוני (Ultrasonic)

כשם שישנם תחומי אור אשר העין האנושית אינה מסוגלת להבחין בהם, ישנם גם תחומי קול אשר האוזן האנושית אינה מסוגלת לשמוע.

קול נע באוויר כגלים, וכאשר הוא נתקל בעצם כלשהו הוא מוחזר כהד. חיישנים אולטרה סוניים עובדים על פי עיקרון זה. לחיישן שני רכיבים – משדר אשר משדר גלי קול בתדר גבוה למשך פרק זמן קצר ומקלט אשר קולט את ההד החוזר של אותם גלי קול.



איור 20: עיקרון הפעולה של חיישן אולטרה-סוני. המשדר בצד שמאל שולח גל קול בתדר גבוה אשר פוגע בעצם כלשהו בצד ימין אשר מוחזר חזרה כהד ונקלט במקלט של החיישן.

❖ חיישן לייזר

כשמו, חיישן זה עושה שימוש בלייזר על מנת למדוד את המרחק לעצם כלשהו. ישנן מספר טכניקות שונות לחישוב המרחק בעזרת לייזר בהתאם לרמת הדיוק הנדרשת וישנם מספר גורמים המשפיעים על כך כמו רוחב הקרן, התפשטות הקרן ועוצמת הקרן. הטכניקה הנפוצה ביותר יכולה להגיע לרמת דיוק של מילימטר והיא נשענת על עיקרון זמן מעוף הקרן. החיישן שולח פולסים של לייזר בקרן מרוכזת לעבר עצם כלשהו, ועל פי הזמן שלקח ללייזר לפגוע בעצם ולחזור למקלט ניתן לחשב את המרחק ממנו.

8.8.3 בחירת סוג החיישן המתאים ביותר

טבלה 3 : השוואת חיישנים מסוגים שונים על פי קריטריונים

חיישנים			
לייזר	אולטרה-סוני	אינפרה-אדום	קריטריונים
גבוהה	בינונית	נמוכה	עלות
גבוה, יכול להגיע גם לקילומטרים	בינוני, עד 1000 ס"מ	נמוך, 60-810 ס"מ	טווח פעולה
צר, עובד על אור	רחב יותר, עובד על קול	צר, עובד על אור	רדיוס סריקה
אין	עצם סופג קול (שלג לדוגמה) עלול לשבש מדידות	אור שמש יכול לשבש מדידות	מגבלות

על פי הטבלה הנראית לעיל ניתן להסיק כי סוג החיישן המתאים ביותר הוא חיישן מסוג לייזר, זאת מכיוון שזהו הסוג היחיד המסוגל לתת מענה לטווח הפעולה הדרוש ולא מוגבל על ידי אור או עצמים סופגי קול.

לאחר סקירת חיישני לייזר רבים והתייעצות עם גורמים רלוונטיים הוחלט לרכוש חיישן לייזר מסוג Lidar Lite v2 מכמה סיבות:

- ❖ עונה על דרישת טווח הפעולה – עד 40 מטרים (לאחר בדיקות בשטח התגלה כי הוא מספק אפילו טווחים עד כ-50 מטרים, אך פרויקט זה אינו מסתמך על כך).
- ❖ דיוק מירבי – סטייה של עד $\pm 0.025m$.
- ❖ משקל נמוך – כ-20.5 גרם ליחידה.
- ❖ חיבור מסוג I2C אשר מתאים לחומרה שברשותנו.
- ❖ צריכת חשמל נמוכה מאוד המתאימה במיוחד לתנאי סוללה מוגבלת.
- ❖ על פי תיאור היצרן – זהו חיישן המיועד בעיקר עבור רובוטים ורחפנים.

8.9 רשימות

8.9.1 רשימת איורים

- 11..... איור 1 : רחפן מסוג IRIS-M של חברת 3DR בעל 4 מנועים אשר ברשות החברה.
- 13..... איור 2 : לוגו החברה.
- איור 3 : רחפן העוקב אחר אדם בעת גלישת סקי וסורק את נתיב הטיסה שלו לצורך הימנעות ממכשולים.
- 14..... איור 4: ארכיטקטורת המערכת.
- 16..... איור 5 : תיכון עתידי של מערכת התוכנה.
- 17..... איור 6 : תרשים זרימה המתאר את האלגוריתם המתוכנן להימנעות ממכשולים.
- 18..... איור 7 : דיאגרמה זו מתארת את קווי התקשורת ואופן החיבור של מספר רכיבים באפיק תקשורת מסוג I2C.
- 20..... איור 8 : דיאגרמה זו מתארת את אופן החיבור של חיישן יחיד מסוג LidarLite V2 למחשב ה- Raspberry Pi 2 באמצעות פס תקשורת מסוג I2C.
- 21..... איור 9 : דיאגרמה זו מתארת את אופן חיבור חיישן ה- LidarLite V2 למחשב ה- Arduino באמצעות פס תקשורת מסוג I2C.
- 22..... איור 10 : דיאגרמה זו מתארת את מיקום החיישנים על גבי הרחפן.
- 24..... איור 11 : דיאגרמה זו הינה תיכון של המערכת שמומשה (ייתכנו שינויים קלים עקב המשך פיתוח המערכת).
- 25..... איור 12 : חלק א' של תרשים זרימה המתאר את אופן פעולת האלגוריתם.
- 30..... איור 13: חלק ב' של תרשים זרימה המתאר את אופן פעולת האלגוריתם.
- 31..... איור 14 : חלק ג' של תרשים זרימה המתאר את אופן פעולת האלגוריתם.
- 32..... איור 15 : דיאגרמה זו מתארת את המסלול שתוכנן לסימולציית הטיסה.
- 35..... איור 16 : עיקרון פעולת האלגוריתם להימנעות ממכשולים. ראשית ינסה הרחפן לעקוף את המכשול מצדדיו, ובמידה ולא הצליח, ינסה לעקוף אותו ממעל.
- 40..... איור 17 : דיאגרמה זו מתארת את חיבורי לוח ה- Raspberry Pi 2.
- 40..... איור 18 : דיאגרמה זו מתארת את אופן חיבור החיישנים ומחשב ה- Raspberry Pi 2 במערכת המשלבת כארבעה חיישנים.
- 41..... איור 19 : עיקרון הפעולה של חיישן אינפרה אדום. המשדר שולח קרני אינפרה אדום אשר פוגעות בעצם, מוחזרות ממנו ונקלטות חזרה במקלט.
- 47..... איור 20: עיקרון הפעולה של חיישן אולטרה-סוני. המשדר בצד שמאל שולח גל קול בתדר גבוה אשר פוגע בעצם כלשהו בצד ימין אשר מוחזר חזרה כהד ונקלט במקלט של החיישן.
- 47.....

8.9.2 רשימת טבלאות

- 28..... טבלה 1 : פירוט התקלות שהמערכת יכולה לזרוק.
- טבלה 2 : פירוט השינויים שיש לבצע בקוד התוכנה בעת הפעלתו על מערכת מלאה עם כשלושה חיישנים.
- 29.....
- טבלה 3 : השוואת חיישנים מסוגים שונים על פי קריטריונים.
- 48.....

8.9.3 רשימת משוואות

- משוואה 1 : חישוב רוחב השטח הנסרק על ידי חיישן מרחק יחיד 23
משוואה 2 : חישוב זווית המרווח הדרושה בין חיישני המרחק לזיהוי מכשולים והימנעות מהם
באופן יעיל 23
משוואה 4 : נוסחת חישוב מרחק במטרים בין שתי נקודות ציון במרחב 45
משוואה 3 : נוסחת חישוב מהירות-זמן-דרך 45
משוואה 5 : המרת יחידות מקילומטרים לשעה למטר לשנייה 46
משוואה 6 : חישוב מהירות הרחפן במטרים לשנייה 46

8.9.4 רשימת סימנים

#	הסימן	הגדרה	יחידות
1	$\Delta\varphi$	הפרש בין שתי נקודות ציון של קו-רוחב	[rad]
2	$\Delta\lambda$	הפרש בין שתי נקודות ציון של קו-אורך	[rad]
3	R	רדיוס כדור הארץ	[m]
4	V	מהירות	[m/s]
5	T	זמן	[s]
6	D	מרחק	[m]

9. Summary

This project focuses on the technological world of multirotor drones and aims to help integrate them in the industrial sector. Drone is an unmanned aerial vehicle (UAV). In recent years there has been increase in the use of drones due to technological development and prices lowering. As of today, drones are taking a bigger part in our lives and used in a variety of fields including photography, video filming, delivery and more.

Reviewing the current uses of drones suggests that drones are not taking a significant part in the business market segment these days, but their big potential has been identified. Therefore, a study has been conducted by the founders of the start-up company 'Airscoart' about the cause which prevents their integration in the industrial sector. The findings showed that the main cause for this is the short battery life of drones, but in addition they understood that drones must be able to carry out complex tasks independently. In order to enhance the drone's capabilities into an autonomous unit he must be able to fly by himself. This need has created a new problem - how could a drone fly safely without supervision and avoid crashing into obstacles?

This project comes to solve this problem by developing an obstacle avoidance system for drones in order to enhance its capabilities and make it able to perform complex tasks autonomously.

This project is carried out under the auspices of 'Airscoart' Start-up company. For the development of an autonomous drone prototype designed to film a person while skiing, the company required the development of an obstacle avoidance system. Coordinated with the company's founders, it's been decided that the system will be created while having two assumptions:

- ❖ The drone will fly at a constant altitude of 10 meters.
- ❖ The obstacles which the drone would avoid will be static, as trees and poles.

The company's requirements have been examined and an analytic research has been conducted about the needed hardware for the system in parallel to reviewing similar projects. According to the findings, it was decided that the system will consist four distance sensors which will be connected to a single board computer, and the computer will be connected directly to the drone's flight controller. This way, the computer will receive data from the distance sensors, decipher them using a complex algorithm and then send appropriate commands to the drone's flight controller in order to correct its flight path and avoid the identified obstacle.

Due to low budget and high cost of the sensor, the company was able to purchase only one sensor. This constraint has created a new problem – testing the system with a lack of input. Therefore, for the purpose of system testing, another class has been developed within the software. This class feeds the needed data to the system from text files which holds measurements taken with the purchased sensor in a pre-planned route.

Since it was impossible to run a full system test on the drone, the sweeping success of the system cannot be guaranteed in different situations, but based on the simulation carried out its possible to see that the system works as expected to the current stage of development and delivers good results.



Software Engineering Department

Obstacle Avoidance System For Autonomous Drone

By:

Ben Nakash

Academic Supervisor:

Shai Tavor



Software Engineering Department

Obstacle Avoidance System For Autonomous Drone

By:

Ben Nakash

July 2016

Tamuz 5776