**המחלקה להנדסת תוכנה**

**פרויקט גמר – תשע"ו**

**מתן פקודות תזוזה לרחפן דרך לוח מחשב לבקר ה Pixhawk**

**Giving commands to drone through by board computer to the Pixhawk**

**עדי גרינשפן**

**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/adiGr/FinalProjectDrone |
| 2 | יומן | https://trello.com/b/HEelaCir/final-project |
| 3 | ניהול פרויקט | Github |
| 4 | סרטון אב-טיפוס |  |

הקדמה: הפרויקט קיבל תפנית לפי בקשות ששונו במהלך העבודה ע"י חברת הסטארט-אפ Airscort, הדבר עודכן ואושר ע"י רכזי הפרויקטים- ד"ר ראובן יגל, ד"ר מרים אללוף,

המנחה - מר שי תבור ואחראי תעשייתי - מר טל יצחק.

# 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 שהרחפן יהיה מסוגל לטוס בצורה אוטומטית ללא התערבות אדם בין אם בכדי להימנע ממכשולים או כדי לטוס אחר אובייקט,

לשם כך הם מעוניינים שתהיה ספריה המקשרת בין ה Pixhawk למחשב לוח להעברת פקודות תזוזת הרחפן שתשרת את שאר הצוותים העובדים בצוות.

# 2.תיאור הבעיה

לחברת הסטארט-אפ Airscort יש כיום מספר צוותים שעובדים איתה, מטרתם הכוללת שהרחפן יעבוד באופן אוטונומי וללא התערבות של אדם, כל צוות ומטרתו שלו.

לשם כך החברה מעוניינת בתוכנה אשר תימצא על לוח מחשב.

הלוח מחשב יהיה על הרחפן באופן פיזי, ויחובר עם בקר ה Pixhawk.

בקר ה Pixhawk הינו בקר התעופה, האחראי על אופן טיסתו של הרחפן.

הבעיה היא שכרגע לחברה אין חיבור בין ה Pixhawk לבין לוח מחשב הכל נעשה ישירות לרחפן דרך שלט או דרך תוכנת מחשב ולא באופן אוטונומי כמו שהם רוצים.

כעת לחברת Airscort אין תוכנה אשר יודעת לבצע פקודות הזזה לרחפן דרך הלוח מחשב.

תוכנה זאת של נתינת פקודות לרחפן צריכה להיות זמינה בסוף הפרויקט לכל הפרויקטים שעובדים ב Airscort , שגם הם יוכלו להשתמש בה לאלגוריתמים השונים.

# דרישות ואפיון הבעיה

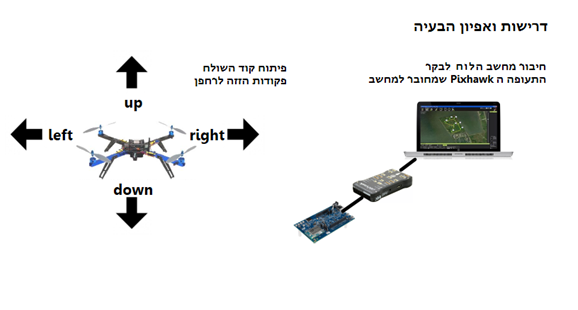
חברת הסטרט-אפ Airscort מעוניינת שאפתח מחלקה שתשמש את יתר הצוותים בחברה.

המחלקה הנדרשת, אחראית על התזוזה של הרחפן מלוח מחשב אל בקר התעופה של הרחפן, ה Pixhawk.

לשם כך, בשלב הראשון אעשה מחקר תכנון אופן שמירת הנתונים בלוח מחשב כדי שיהיו זמינים לצוותים האחרים.

בנוסף, דרישת החברה היא לשלוח פקודות לבקר ה Pixhawk דרך מחשב הלוח מחשב על מנת להקל עליהם בהמשך עם ההתחברות בין 2 החומרות הללו.

לאחר שהושגה שליחת פקודות, ויש תקשורת העובדת, אצטרך לרשום תוכנה השולחת פקודות הזזה לרחפן כדי ששאר הצוותים העובדים בחברה יוכלו להתמקד בבעיית האלגוריתם שלהם.



# הבעיה מבחינת הנדסת תוכנה

בפרויקט זה אעבוד מול בקר הטיסה של הרחפן מסוג Pixhawk. בקר זה מתבסס על סביבת קוד פתוח, ולכן נתכנת בשפות C וC++. ישנן מספר בעיות תכנותיות אשר אנו נאלצים לפתור:

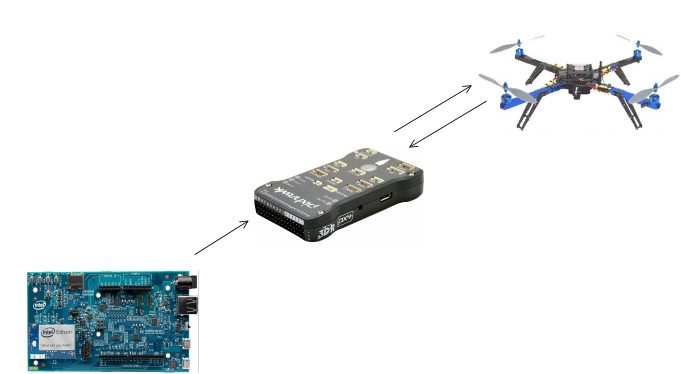
1. התקנת סביבת עבודה וללמידה על הבקר באופן מעמיק.
2. צריבת תוכנית והרצתה בshell המתאים לבקר
3. שאיבת נתונים בזמן אמת לצורך זיהוי מכשולים מחיישני פנימיים שנמצאים על בקר ה-Pixhawk.

כל אלה יסיעו ללימוד מעמיק ותכנותי על הבקר Pixhawk שיקלו בהמשך על העבודה עם לוח המחשב מול בקר התעופה ,בלוח המחשב צריכה להיות מותקנת מערכת הפעלה, המחלקה אותה אכתוב לכתיבת פקודות הזזה לרחפן יהיו בשפת Python ואשתמש ב API שאיתו ניתן לתקשר עם בקר ה Pixhawk מרחוק, הבעיות התכנותיות שאיתן אתמודד:

1. למידת שפה שהוכרה באופן שטחי מלפני : Python
2. עבודה עם חומרה חדשה, לוח מחשב .
3. למידה על דרכי תקשורת והתחברות עם ה Pixhawk דרך לוח המחשב
4. כתיבת אלגוריתם להזזת הרחפן לכיוונים שונים- המתאים לרחפן בשוק כבר קיימים אלגוריתם להזזת הרחפן אך הם עושים שימוש בטכנולוגיה מתקדמת ויקרה שאינה במסגרת התקציב והחומרה המוגבלים שברשותנו, את תזוזת הרחפן נכתוב לפי דרישות ספציפיות של החברה Airscort

# תיאור 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).
* ***לוח מחשב*** היא פלטפורמת מחשוב קטנה במיוחד שיצור על ידי אינטל שמחובר ללוח arduino , החיבור ללוח ה arduino מאפשר ל לוח מחשב לתמוך גם ב Arduino Sketch, Linux, Wi-Fi, Bluetooth., הלוח מאפשר יציאות רבות ונכנסות שדרכן להעביר נתונים ופקודות,ולקבל נתונים בנוסף אפשר להתחבר דרך ה USB . את מחשב הלוח נחבר לבקר ה Pixhawk ועליו נרשום את התוכנה, שהחברה מעוניינת בדרך כזאת שתהיה נגישה לשאר הצוותים שיכולו להשתמש בה, ובתוכנה זאת התמצא על לוח מחשב נרשום תוכנה שדרכה נעביר פקודות לרחפן דרך הבקר תעופה ה Pixhawk .
* ***תוכנה השולטת על תזוזת הרחפן*** –תוכנה אשר אפתח ואצרוב למחשב לוח שיתחבר לבקר ה-Pixhawk השולטת על הטיסה, התוכנה תשלח לרחפן פקודות לתזוזה, התוכנה תצרב במקום אשר תהיה זמינה לשאר חברי הפרויקט כדי שיוכלו להשתמש בה.

## תיאור הפתרון המוצע:

לשם הכרה טובה יותר של בקר ה Pixhawk אצרוב תוכנית על הבקר המתחברת אל החיישנים פנימיים וזאת לשם הכרה והקלת העבודה בהמשך הפרויקט עם החיבור של מחשב הלוח.

***תוכנית המתחברת לחיישנים פנימיים של בקר ה Pixhawk:***

1. הקמת סביבת עבודה מתאימה ב Linux.
2. התחברות לבקר ה Pixhawk.
3. כתיבת תוכנית פשוטה המוציאה פלט התחלתי.
4. צריבת תוכנית פשוטה על בקר ה Pixhawk.
5. הרצת התוכנית וקבלת פלט מתאים לתוכנית.
6. כתיבת תוכנית המתחברת אל החיישנים הפנימיים
7. צריבה והרצת התוכנית ולבסוף בדיקת וקבלת פלט מתאים לתוכנית.

***תוכנית המאפשרת להזיז את הרחפן:***

1. כתיבת קוד פשוט אשר ישמר על לוח המחשב.
2. התחברות לבקר ה Pixhawk דרך לוח מחשב.
3. הרצת תוכנית הראשונית.
4. בדיקת הפלט של התוכנית .
5. תחילת כתיבת הקוד התאפשר תזוזה לרחפן.
6. כתיבת מחלקה אשר תבדוק את התוכנה והרצתה ותיקוני באגים.
7. המשך כתיבת המחלקה, הרצה מחלקה בודקת ותיקונים נוספים.

## תיאור הכלים המשמשים לפתרון

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

API המאפשר למפתחים ליצור תוכניות שירוצו על הלוח מחשב המחובר אל בקר התעופה הנמצא ברחפן. אותו לוח מדבר ושולט על ה-Autopilot. תוכניות הרצות על מעבד זה מסוגלות לבצע משימות עיבוד אינטנסיביות ומשימות תלויות .  
API זה עובד בסביבות העבודה Unix ו- Windows .

* ***Mission Planner:***

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

1. מתן נקודות ציון ומסלול מעוף לרחפן בעזרת Google Maps (תכנון משימות טיסה).
2. הורדת קבצי Log של המשימה וניתוחם.

* ***Arduino Software (IDE):***

סביבת העבודה של Arduino. אוכל להיעזר בה כתחליף לעבודה עם לוח מחשב במידה ויהיה תפוס, ובכך לנצל זמן גם כאשר צוות אחר עובד על מחשב הלוח .

* ***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.

## 4.תוכנית בדיקות

במהלך עבודתי עם הPixhawk עשתי בדיקות במהלך כל העבודה:

חיבור של הבקר למחשב- כאשר חיברתי את הבקר למחשב שלי עשיתי מספר בדיקות לוודא שאכן הבקר מחובר והמחשב מכיר את הבקר, כמו כן ברגע בדקתי כי הבקר יודע שמחובר אליו USB ע"י הרצת פקודות כמו: make px4fmu-v4\_default שבפעם הראשונה שהוא רץ הוא מוכרח לקבל חיבור עם הרחפן , ובמחשב יכולתי לראות את החיבור שנעשה.

צריבת תוכנית על הבקר Pixhawk- במהלך הצריבה הייתי צריכה לבדוק שאכן יש לי את היכולת לצרוב על הבקר תוכנית אשר כתבתי , ברגע שהבקר מזהה צריבה שנעשתה עליו הוא משמיע צפצוף, הבדיקה תעשה שברגע שאצרוב תוכנית שלי ואראה פלט אוכל לדעת שהצריבה נעשה בוודאות, ולכן כתבתי תוכנית פשוטה המדפיסה hello adi .

הורדת shell המאפשר הרצה של התוכנית באופן רציף עם הבקר- את הshell בדקתי ע"י הרצה של הפקודה: screen /dev/ttyACM0 57600 8N1

הרצתי את התוכנית hello adi וברגע שהודפסה המילה וידאתי שאכן צרבתי את התוכנית.

כעת כתבתי תוכנית אשר מתחברת לחיישנים ומדפיסה את הפלט שלהם על מנת שאוכל לבדוק שאכן התוכנית מתחברת אל החיישנים הפנימיים בתוך ה Pixhawk ,

הרצתי את התוכנית וראיתי שהפלט הינו הזווית של הבקר

בשביל לבדוק את התוכנית הרצתי שוב את התוכנית רק שהפעם שינית את הזווית כל פעם של ה Pixhawk ובדקתי שאכן הפלט משתנה כל פעם לזווית אחרת.

## סקירת עבודות דומות בספרות והשוואה

* 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.

https://pixhawk.org/dev/start

מדריך התקנה והסבר על ה Pixhawk, הסבר על צריבה, סביבת עבודה.

* PIXHACK

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

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

* Autonomous quadcopter, obstacle detection on Arduino

<http://events.linuxfoundation.org/sites/events/files/slides/Edison%20Takes%20Flight.pdf>

הסבר על Edison כאשר נעשה שימוש עם רחפנים. (דוגמה למחשב לוח)

* guiding and Controlling Copter

<http://python.dronekit.io/examples/guided-set-speed-yaw-demo.html>

מדריך תזוזות לרחפן

* trying to read mavlink messages from pixhawk with my arduino

<http://diydrones.com/forum/topics/trying-to-read-mavlink-messages-from-pixhawk-with-my-arduino>

מארדואינו לפיקסהוק

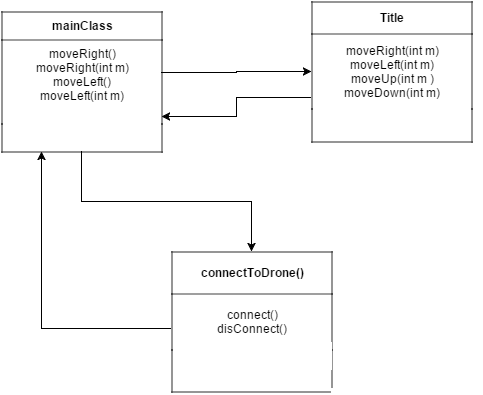
# נספחים

## ביבליוגרפיה

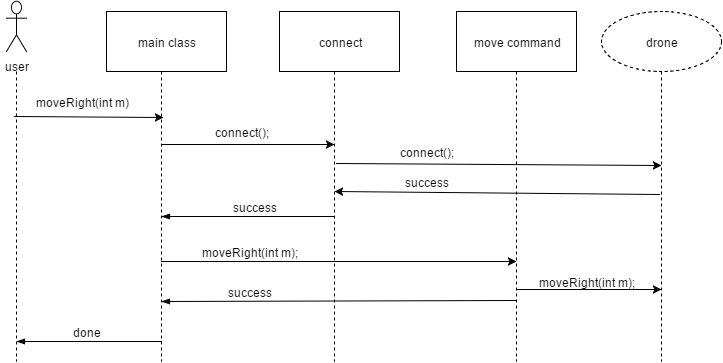
* http://copter.ardupilot.com/
* https://phidgets.wordpress.com/2014/05/23/exploring-the-many-methods-of-object-detection/
* <http://diydrones.com/>
* [https://learn.sparkfun.com/tutorials/edison-getting-started-guide?\_ga=1.42893285.1336304550.145149438](https://learn.sparkfun.com/tutorials/edison-getting-started-guide?_ga=1.42893285.1336304550.1451494387)
* <https://learn.sparkfun.com/tutorials/loading-debian-ubilinux-on-the-edison>

## תרשימים וטבלאות

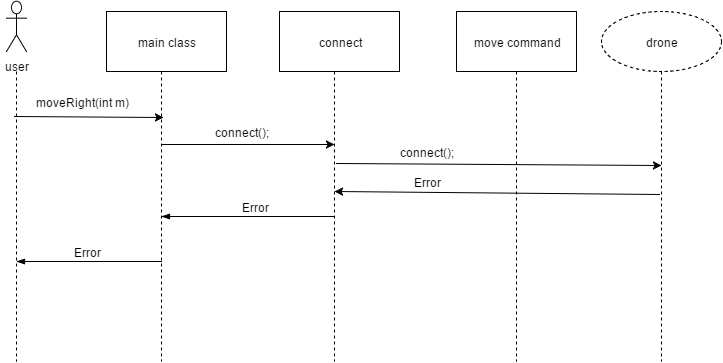
* Class Diagram כללית של התוכנה למחלקת פקודות הזזה לרחפן והתחברות אליו:

****

* Sequence Diagram – פקודת הזזה ימינה הנמצאת במחלקה



* Sequence Diagram – פקודת הזזה ימינה הנמצאת במחלקה- במידה ולא הצליחה התחברות לרחפן



## תכנון הפרויקט

|  |  |
| --- | --- |
| **04.10.15** | היכרות ותחילת עבודה עם Airscort |
| **01.11.15** | התחלת ביצוע מחקר על הpixhawk |
| **22.11.15** | הגשת הצעת פרויקט |
| **01.12.15** | רכישת חיישן מרחק ראשון ולמידת סביבת העבודה |
| **24.01.16** | הגשת אב טיפוס – התקנת סביבת עבודה, כתיבת תוכנית וצריבתה לבקר הpixhawk , קבלת נותנים מחיישנים פנימיים הנמצאים בתוך הבקר |
| **01.03.16** | חיבור ה לוח מחשב ל Pixhawk,אתחול הלוח מחשב לתקשר עם ה Pixhawk, כתיבה ושמירת תוכנית בלוח מחשב, הרצת התוכנית,למידה על מתן פקודות ל Pixhawk ותחילת כתיבת הספרייה |
| **01.04.16** | בדיקת התוכנית שנשמרה על לוח מחשב, בדיקתה , המשך כתיבת ספריית הפקודות |
| **05.05.16** | בניה –סיום כתיבת הספרייה, כתיבת תוכנית שתשתמש בפקודות הספרייה לבדיקתו,הרצת התוכנית על סימולאטור mission planner ,תיקון באגים ,הרצה על הרחפן עצמו. |
| **1.06.16** | תיקוני באגים שהתגלו בהרצה הקודמת והרצה נוספת של התוכנית על הרחפן עצמו. |
| **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** | שינוי דרישות | 5/5 | בעבודה עם חברת סטארט-אפ יש שינוי בדרישות וצריך להיות גמישים ככל שניתן,  מתן עדיפות לדרישות הפרויקט המקורי |
| **5** | חוסר ידע ב סימולאטור המתאים לרחפן וסביבות העבודה שלו. | 2/5 | למידת הסימולאטור וסביבות העבודה של הרחפן. |
| **6** | תכנות לא יעיל שעלול לגרום לעומס חריג על המעבד ולגרום בנוסף לבזבוז סוללה. | 3/5 | ניתוח זמן הריצה של הקוד שכתבנו וכתיבתו באופן יעיל עד כמה שנוכל. |
| **7** | חוסר ידע בבקר ה-Pixhawk ואופן פעולתו עם הרחפן. | 4/5 | חקירת הבקר Pixhawk, לחקור היטב על הצריבה, כתיבת התוכנית שעליו, והוצאת נתונים ממנו. |
| **8** | עיכוב בפיתוח של צוות האחר האמור להשתמש בפקודות התזוזה לרחפן. | 4/5 | כתיבת תוכנית המשתמשת בפקודות התזוזה בספרייה. |
| **9** | פגיעה חמורה ברחפן | 3/5 | הרצת בדיקות וסימולציות יבשות לפני ביצוע ניסויים חיים עם הרחפן. |
| **10** | אי הגעה של החיישנים בזמן | 4/5 | כתיבת קוד עם חיישנים אחרים וניסיון להתאים לחיישנים בזמן הגעתם. |
| **11** | חוסר ידע בחיבור מחשב לוח .  והעברת נתונים ופקודות דרכו | 4/5 | מחקר מרובה על לוחות מחשבים ובחירת לוח מחשב ספציפי.  בדיקה לגבי פרויקטים אחרים אשר למדו וחקרו. |
| **12** | התעכבות במשלוח ותקינותם | 3/5 | לעבוד עם חלקים חלופיים הנמצאים במקום. |

## ה.טבלת דרישות

|  |  |
| --- | --- |
| **#** | **תיאור** |
| **1** | מחקר על Pixhawk:   1. סביבת עבודה מתאימה שדרכה ניתן לעבוד על Pixhawk . 2. חיבור של הבקר למחשב ומחקר על החיבורים שנמצאים עליו. 3. שפות שניתן לתכנת על ה Pixhawk . 4. חקירת חיישנים פנימיים הנמצאים על הבקר. 5. למידה על חיישנים פנימיים וגישה אליהם בתוכנית הרצה על Pixhawk. |
| **2** | צריבת תוכנית לבקר Pixhawk :  1. הקמת סביבת עבודה ב Linux.  2. התחברות לבקר ה Pixhawk.  3. צריבת תוכנית על הבקר Pixhawk. |
| **3** | שאיבת נתונים מהבקר Pixhawk באופן ישיר:   1. הורדת shell המאפשר הרצה של התוכנית באופן רציף עם הבקר. 2. כתיבת תוכנית פשוטה לבדיקת ה shell וצריבת התוכנית. 3. כתיבת תוכנית המתחברת אל החיישנים המובנים שנמצאים על הבקר: חיישני זווית. 4. כתיבת תוכנית המדפיסה את הנתונים של הזוויות. 5. הרצת התוכנית,והדפסת הפלט לפי שינוי זווית ה Pixhawk . |
| **4** | מחקר על הלוחות מחשבים שונים:   1. חיבור הלוח מחשב ל Pixhawk. 2. אתחול לוח המחשב לתקשר עם ה Pixhawk. 3. כתיבה ושמירת תוכנית בלוח מחשב. 4. הרצת התוכנית 5. קבלת נתונים מה Pixhawk ל לוח מחשב . 6. תכנון אופן שמירת הנתונים בלוח מחשב כך שיהיו זמינים לצוותים האחרים |
| **5** | בדיקת תקשורת בין לוח מחשב לבקר ה Pixhawk:   1. כתיבת תוכנית על לוח מחשב . 2. הרצת התוכנית. 3. בדיקת הפלט. |
| **6** | כתיבת ספרייה תזוזה לבקר ה Pixhawk:   1. כתיבת ספרייה :פקודות ההזזה לבקר ה Pixhawk 2. כתיבת תוכנית שתשתמש בפקודות הספרייה לבדיקתו. 3. הרצת התוכנית על סימולאטור mission planner 4. תיקון באגים שהתגלו בהרצת הסימולאטור. 5. הרצה של התוכנית המשתמשת בספרייה על הרחפן עצמו. |