

הפקולטה להנדסה
המחלקה להנדסת חשמל ואלקטרוניקה

**רחפן חכם - "MY_Drone":
תכנון, פיתוח ובנייה של מערכת אווירית
מתקדמת עם בקרת קרקע ושלט לשליטה**

פרויקט גמר המהווה חלק מהדרישות לתואר B.Sc.

שמות המבצעים	מס' ת.ז.	דוא"ל
מיכאל מרזיב	208579862	Michael.Marzayev@gmail.com
יהונתן זבידה	206555294	yonizvida@gmail.com

אחראי אקדמי באוניברסיטה : ד"ר דניאל רוזבן

ספטמבר 2025

תשרי ה'תשפ"ו

תקציר

רחפנים מהווים אחת הטכנולוגיות החדשניות והמתקדמות ביותר בעשור האחרון, עם יישומים מגוונים בתחומי תעשייה, ביטחון, חקלאות, מחקר מדעי ועוד. יכולתם לשלב בין ניידות, איסוף נתונים ובקרה מדויקת הופכת אותם לנושא מרכזי בלימודי הנדסה ולטכנולוגיה בעלת פוטנציאל רב. פרויקט זה מתמקד בתכנון, פיתוח ובנייה של מערכת רחפן חכם, המדגים שילוב בין כלי טיס חכם לרשת תקשורת קרקעית מתקדמת. בפרויקט פותחה מערכת משולבת לניטור ושליטה, הכוללת חיישנים ורכיבי קצה הנשלטים באמצעות פרוטוקולי תקשורת שונים, כגון **WebSocket, TCP/IP** ועוד, המאפשרים בקרה וניטור בזמן אמת. הליבה האווירית של המערכת היא רחפן מתקדם שנבנה ותוכנן במלואו במסגרת הפרויקט. הרחפן פועל באמצעות בקר **ESP32** המשמש כבקר הטיסה של המערכת וכן יחידת העיבוד שלה, מקבל ושולח נתונים למחשב קרקעי (**Raspberry Pi 5**) ומנוהל דרך ממשק משתמש ייעודי על גבי טאבלט. תכנון הרחפן כלל מחקר רב בתחומים הנדסיים שונים, בחירה והטמעה של רכיבי טיסה, שילוב מערכות חיישנים, ותכנון לוגיקת בקרה המבטיחה יציבות, אמינות ותגובה מהירה. המערכת מדגימה שילוב בין חומרה, בקרה, תוכנה ותקשורת, תוך התמודדות עם אתגרים הנדסיים כגון אינטגרציה בין רכיבים משובצים, פיתוח פרוטוקולי תקשורת מותאמים ושיפור זמני תגובה. יישומי המערכת מגוונים וכוללים ניטור סביבתי, בקרה במבנים חכמים, תעשייה חכמה, ויכולת הרחבה לניהול יחידות ניידות נוספות. שילוב הרחפן מדגיש את חשיבותה של רשת תקשורת אמינה בהפעלת מערכות דינמיות הדורשות תגובה מהירה, סנכרון מדויק ורמת אוטונומיה גבוהה.



איור מס' 1 – הדמיית חזותית של המערכת

הכרת תודה

ברצוננו להודות מכל הלב לכל האנשים שעמדו לצדנו ותמכו בנו לאורך הדרך בביצוע פרויקט הגמר המאתגר.

בראש ובראשונה, אנו מבקשים להביע תודה עמוקה לבורא עולם, שבזכותו ניתנה לנו היכולת להגיע לרגע זה.

תודה מיוחדת למשפחותינו היקרות על התמיכה, העידוד והסבלנות הרבה שגילו לאורך התהליך – במיוחד ברגעים הקשים בהם חווינו תסכול וייאוש, ועל ההבנה וההתמודדות עם שעות העבודה הארוכות, בסופי השבוע ובשעות בלתי שגרתיות.

כמו כן, אנו מבקשים להודות למנחה האקדמי של הפרויקט, ד"ר דניאל רוזבן, על הליווי, האמון והחופש האקדמי שניתנו לנו, ועל התמיכה המקצועית לאורך כל הדרך.

על כל אלה – תודה גדולה מכל הלב.

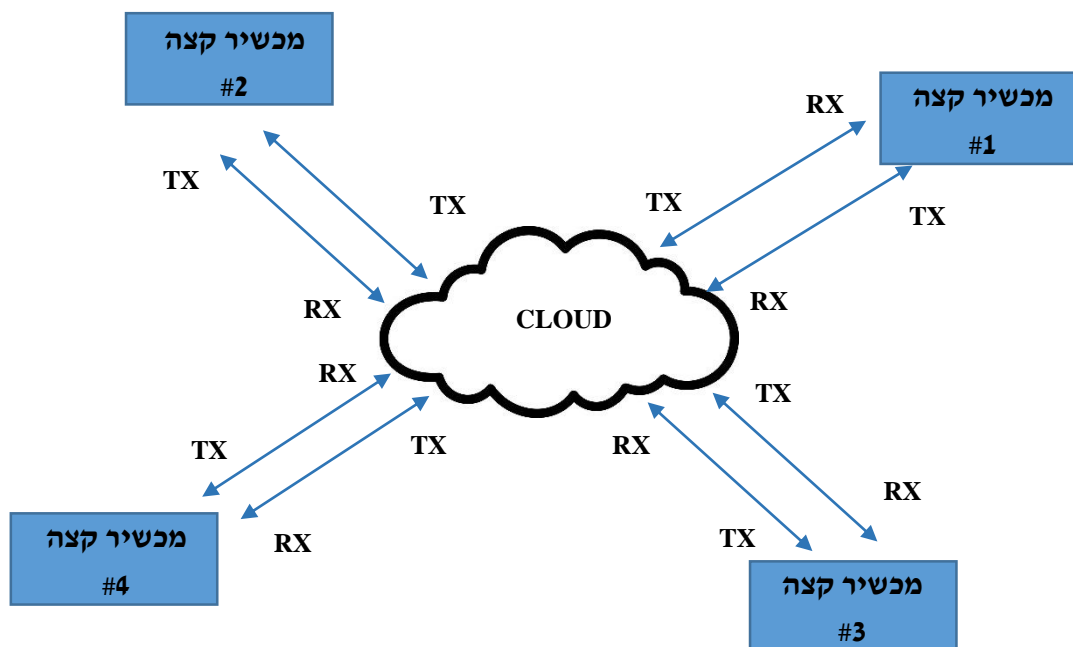
תוכן העניינים

6.....	מבוא ורקע.....	
8.....	מטרת הפרויקט.....	
9.....	תנאי התכנון.....	
10.....	דרישות המערכת.....	
11.....	תיאור המערכת.....	1.
13.....	Raspberry Pi 5 – יחידת שליטה ובקרה (תת מערכת - מכשיר קצה).....	1.1.
14.....	היכרות עם Raspberry Pi 5 והגדרות רשת :.....	1.1.1.
16.....	תכנון החומרה.....	1.1.2.
17.....	תכנון התוכנה.....	1.1.3.
19.....	שלט המשתמש - טאבלט (תת מערכת - User Interface).....	1.2.
20.....	היכרות עם הטאבלט ופלטפורמת MIT App Inventor :.....	1.2.1.
21.....	תכנון החומרה.....	1.2.2.
22.....	תכנון התוכנה - אפליקציית שליטה (MIT App Inventor).....	1.2.3.
26.....	הרחפן - (תת מערכת אווירית).....	1.3.
26.....	תכנון המכניקה.....	1.3.1.
33.....	תכנון החומרה.....	1.3.2.
42.....	תכנון התוכנה.....	1.3.3.
51.....	מימוש המערכת.....	2.
51.....	
51.....	
52.....	ניסויים ובדיקות.....	2.1.
52.....	ניסוי כיול ראשוני ותקינות החיישנים.....	2.1.1.
53.....	ניסוי תקשורת - שידור וקליטה (WebSocket).....	2.1.2.
54.....	ניסוי בקרת יציבות (PID).....	2.1.3.
55.....	ניסוי ובדיקת מנגנוני בטיחות (Fail-Safe).....	2.1.4.
55.....	ניסוי וכיול כיוון סיבוב להבי המנוע.....	2.1.5.
56.....	ניסוי טיסה כולל.....	2.1.6.
57.....	מכשירי המדידה.....	2.2.
58.....	תוצאות ומסקנות.....	3.
58.....	תוצאות הניסויים והמדידות.....	3.1.
59.....	מסקנות הפרויקט.....	3.2.
59.....	סיכום.....	3.3.
60.....	מקורות ספרותיים.....	4.

מבוא ורקע

בעשורים האחרונים חלה התקדמות משמעותית בתחומי התקשורת, האוטומציה והבינה המלאכותית, אשר מאפשרת פיתוח מערכות חכמות לניטור ושליטה מרחוק. מערכות אלו ממלאות תפקיד מרכזי בתעשיות מתקדמות כגון חקלאות חכמה, לוגיסטיקה, רובוטיקה תעשייתית, בטיחות ואבטחה, שבהן נדרשת תקשורת יציבה, אמינה ובעלת השהיה נמוכה בין רכיבי קצה לבין מרכזי בקרה.

פרויקט זה מתמקד בהקמת רשת תקשורת פנימית (LAN) בין מחשבים, המיועדת לניטור חיישנים, שליטה מרחוק והפעלת רכיבי קצה, תוך שילוב רחפן עצמאי כחלק אינטגרלי מהמערכת. הרחפן, אשר נבנה ותוכנן באופן עצמאי, מתפקד כרכיב נייד ברשת התקשורת, מקבל פקודות ממחשב קרקעי (Raspberry Pi 5) מנוהל באמצעות בקר ESP. הבקרה נעשית דרך ממשק משתמש במכשיר קצה, המאפשר שליטה בזמן אמת, קבלת נתוני חיישנים וניתוח המידע המתקבל מהמערכת.



איור מס' 2 – תיאור המערכת בקווים כלליים

רקע טכנולוגי

- **רשתות תקשורת פנימיות (Local Communication Networks) -**

רשתות תקשורת פנימיות משמשות לארגון וחיבור בין מכשירים ללא תלות בתשתיות חיצוניות, כגון אינטרנט ציבורי. טכנולוגיות כמו Wi-Fi, TCP/IP ו-WebSocket מאפשרות חיבור אמין בין חיישנים, בקרים ומחשבים, תוך שמירה על יעילות העברת נתונים וזמני תגובה קצרים. בפרויקט זה, הרשת הפנימית מאפשרת תקשורת ישירה בין מחשב הקרקע, הרחפן והרכיבים הנלווים.

- **שליטה מרחוק ברחפנים -**

שימוש ברחפנים לשליטה מרחוק וליישומים אוטונומיים הולך וגדל, במיוחד בתחומים כגון משלוחי סחורות, סריקות שטח וניטור סביבתי. הרחפן בפרויקט מופעל על ידי בקר ESP, המקבל נתונים מהמחשב הקרקעי ומעביר פקודות למנועים, למערכות הניווט ולחיישנים השונים. התקשורת בין הרחפן לבין המערכת הקרקעית מבוססת על Wi-Fi או על פרוטוקולי רדיו, בהתאם לצרכים הספציפיים.

- **פרוטוקולי תקשורת מתקדמים -**

כדי להבטיח תקשורת אמינה ויעילה בין רכיבי המערכת, נעשה שימוש במספר פרוטוקולים משלימים :

- **WebSocket -** פרוטוקול דריכוני (Full Duplex) המאפשר תקשורת רציפה, יציבה ובזמן אמת בין שרת ללקוח.

- **TCP/IP -** פרוטוקול תקשורת תקני לניהול ושליחה מאובטחת של נתונים בין מחשבים.

- **IoT -** פרוטוקול תקשורת טורי פנימי, המשמש לחיבור יעיל בין בקרי מיקרו וחיישנים, ומאפשר למערכת הרחפן אינטגרציה נוחה עם מגוון רכיבי חומרה.

מטרת הפרויקט

מטרת הפרויקט הינה פיתוח והקמת רשת תקשורת פנימית (LAN – Local Area Network) בין מחשבים, המאפשרת ניטור חיישנים והפעלת רכיבי קצה באמצעות ניתוב חכם ופרוטוקולי תקשורת מתקדמים. הפרויקט ממחיש יישום מעשי של תקשורת אלחוטית ושליטה מבוזרת, תוך שילוב רחפן עצמאי המנוהל באמצעות Raspberry Pi 5 ו-ESP, הנשלט דרך ממשק משתמש ידידותי במכשיר קצה.

שילוב הרחפן במערכת מספק יתרונות נוספים :

1. שימוש במכשירים ניידים כאמצעי שליטה, המרחיב את אפשרויות השימוש במערכת.
 2. תמיכה בחיישנים ורכיבי קצה, המאפשרת ניטור בזמן אמת של נתונים סביבתיים, מכניים ועוד.
 3. פיתוח מערכת מודולרית וגמישה, שניתן ליישם בתחומים כמו אבטחה, חקלאות, תעשייה חכמה ויישומים רובוטיים שונים.
- הפרויקט מאפשר לבחון ולשפר את ביצועי רשתות תקשורת פנימיות, להבין את האינטגרציה בין רכיבי חומרה ותוכנה, וליצור פתרונות חדשניים לניהול יחידות חכמות ומתקדמות. באמצעות יישום זה נבחנים היבטי תקשורת, זמן תגובה ויציבות המערכת, תוך מתן פתרון גמיש לניהול יחידות ניידות עם יישומים אפשריים בתעשייה, אבטחה ומחקר מדעי.

תנאי התכנון

במהלך תכנון, פיתוח והקמת מערכת התקשורת והרחפן בפרויקט זה, נלקחו בחשבון התנאים והדרישות הבאים:

1. **תשתית חומרה מוגבלת:** השימוש ברכיבים זמינים כגון Raspberry Pi 5 ו-ESP בלבד, ללא חיבור לאינטרנט ציבורי.
2. **רשת פנימית (LAN) בלבד:** כל התקשורת בין רכיבי המערכת חייבת להתבצע ברשת מקומית.
3. **זמני תגובה מהירים:** המערכת חייבת לאפשר שליטה בזמן אמת על הרחפן והרכיבים הנלווים.
4. **שליטה ניידת:** ממשק המשתמש חייב להיות זמין במכשירי קצה ניידים (טלפון/טאבלט/מחשב).
5. **אינטגרציה בין רכיבי חומרה ותוכנה:** הרחפן, החיישנים והבקר חייבים לתפקד בצורה מתואמת ומסונכרנת.
6. **בטיחות ויציבות:** הרחפן ומערכת התקשורת חייבים לפעול באופן בטוח ואמין, ללא סיכון למשתמש או נזק לרכיבים.
7. **מודולריות וגמישות:** המערכת חייבת להיות ניתנת להרחבה, החלפת רכיבים או שינוי פרוטוקולי תקשורת בעתיד.

דרישות המערכת

1. המערכת חייבת לאפשר שליטה בזמן אמת על הרחפן והרכיבים הנלווים.
2. המערכת חייבת לבצע ניטור חיישנים באופן רציף, כולל הצגת נתונים בממשק משתמש ובתחנת הניטור.
3. הרחפן (בקר ESP) חייב לקבל ולשלוח פקודות גם מול המחשב הקרקעי (RPI5) וגם מול שלט – ממשק המשתמש (טאבלט).
- (רשת דו כיוונית בין כל המשתמשים גם CLIENTS וגם SERVER).
4. המערכת חייבת להשתמש בפרוטוקולי תקשורת מתקדמים כגון TCP/IP ו-WebSocket.
5. המערכת חייבת להיות מודולרית וגמישה, כך שניתן להוסיף רכיבים נוספים וכן שתהיה פתוחה לעדכונים ושדרוגים בעתיד.
6. הממשק למשתמש חייב להיות ידידותי ונגיש ממכשירי קצה ניידים.
7. המערכת חייבת להבטיח יציבות ובטיחות במהלך הפעלת הרחפן והרכיבים.

טבלה מס' 1 – פרמטרי תכנון נדרשים

דרישת מערכת	זמן מחזור לפעולה	תדר פעילות	פרוטוקולים ופעולות נדרשות
שליטה בזמן אמת	$\leq 10\text{mSec}$	100Hz	שליחת נתונים ע"י – WebSocket בזמן מוגדר קבוע
שליטה בזמן אמת	$\leq 4\text{mSec}$	250Hz	שמירה על יציבות בעזרת בקר PID מתוכנת – בקר טיסה אדפטיבי ESP32
ניטור חיישנים באופן רציף	$\leq 4\text{mSec}$	400KHz	I2C
תקשורת			אפשר FULL-DUPLEX בין כל רכיבי המערכת על גבי WebSocket.
מודולריות וגמישות			<ol style="list-style-type: none"> 1. שימוש ברכיבים עדכניים עם גרסאות מתקדמות כגון: ESP32, RPI5 (לא מוצרי EndOfLife). 2. לשמור קודים אחרונים שנכתבו ולהעלות אותם לתיק פרויקט ב-github 3. – סביבת קוד פתוח (על מנת לאפשר עדכונים באופן קל ונגיש). 3. ליווי הקוד בהערות מובנות כדי לאפשר כניסה קלה לקוד על ידי אדם חיצוני.
אבטחה			שימוש בסיסמאות וחסמים להתחברות לרחפן/לשלט/לתחנת הניטור כדי לשבש שליטה ותקשורת.
יציבות ובטיחות			<ol style="list-style-type: none"> 1. מרווחי פעולה תקינים 2. כפתורי ניתוק תקשורת ומתח ואיפוס מנועים בעת תקלה 3. תנאי חסם בקוד – לא להיכנס למצבים לא מוגדרים ו"נקודות מתות".

1. תיאור המערכת

1.1 רכיבי המערכת

- **רחפן** - מכיל בקר ESP32 כבקר טיסה מרכזי (אלגוריתם טיסה ומערכת ניווט), בקר לעיבוד פעולות כלליות ומימוש התקשורת במערכת.
בנוסף מכיל ESC (לבקרת אספקת הזרם למנועים), מנועים, סוללת ליתיום נטענת וכן חיישנים.
הרחפן מקבל פקודות מהמערכת ומספק נתונים למחשב הקרקעי.
- **תחנת ניטור נתונים (מחשב קרקעי Raspberry Pi 5)** מהווה מרכז בקרה, מקבל ומעבד נתונים מהרחפן ומהחיישנים, ושולח פקודות לשליטה על הרחפן והרכיבים.
- **חיישנים ורכיבי קצה** - מודדים נתונים סביבתיים, מכניים או אחרים, ומעבירים אותם למחשב הקרקעי דרך הרשת.
- **ממשק משתמש** - מאפשר שליטה בזמן אמת, וניהול רחפן והרכיבים באמצעות מכשירים ניידים או מחשבים.

2. תקשורת

- כל הרכיבים מחוברים ברשת פנימית (LAN).
- שימוש בפרוטוקולי **TCP/IP, WebSocket** ו-**I²C** לתקשורת אמינה בין רכיבים.
- שימוש בתקשורת **FULL-DUPLEX** בין המשתמשים (RX ו-TX).

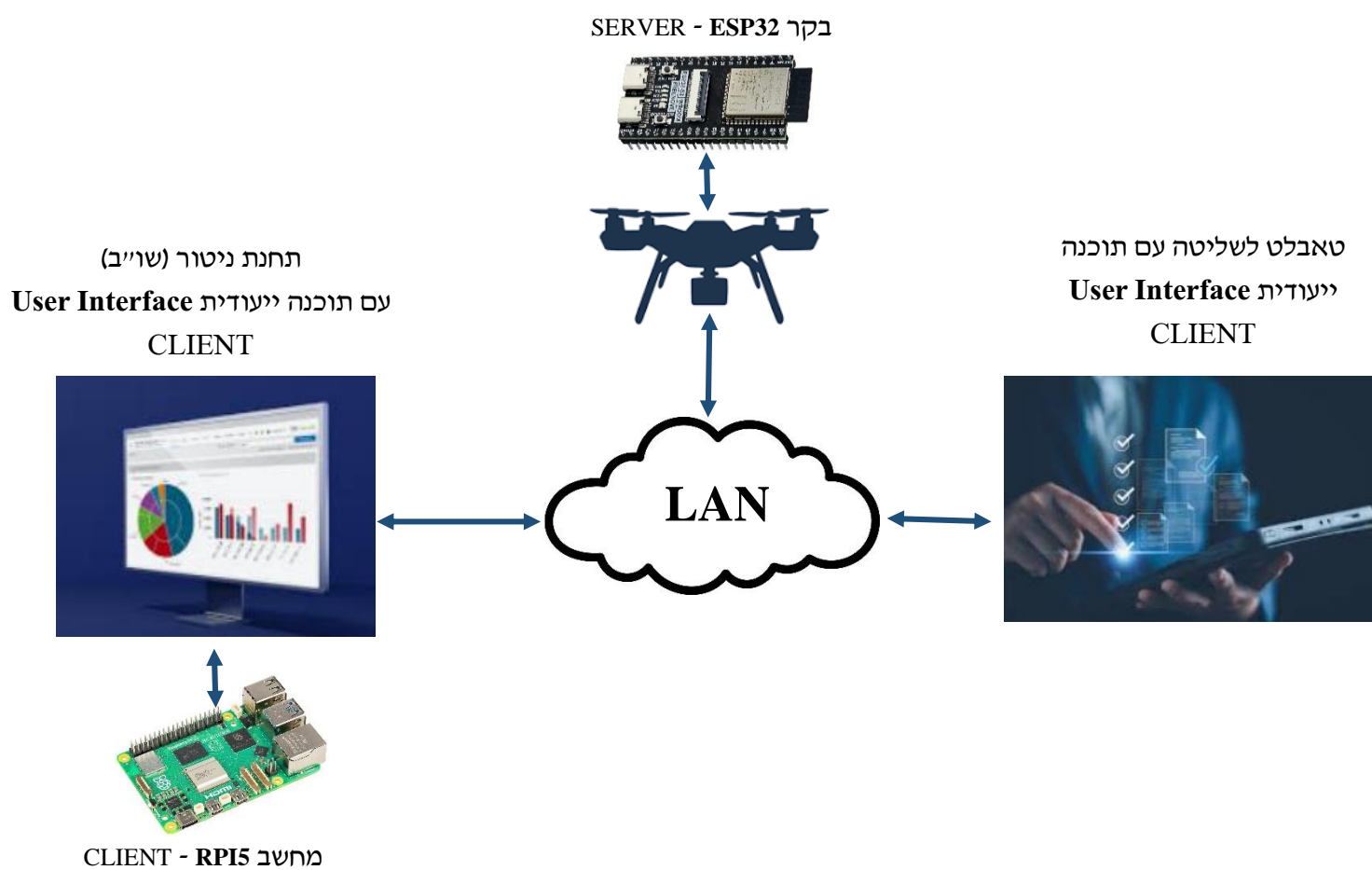
3. בקרת הרחפן

- הרחפן מופעל ונשלחות פקודות אל ה-ESP32 המקבל מידע מהמשתמש.
- בקר ה-ESP32 משמש כבקר טיסה של הרחפן בעזרת אלגוריתם PID לייצוב ובקרת טיסה וכן כבקר האחראי על פעילות של התקשורת והחיישנים.
- הבקר אחראי על ניווט, שליטה במנועים, קריאת חיישנים ושמירה על יציבות בזמן אמת.

4. תכונות עיקריות

- שליטה בזמן אמת על רכיבי קצה ורחפן.
- מודולריות וגמישות להוספת רכיבים חדשים.
- יציבות ובטיחות גבוהה של המערכת.
- יכולת ניטור ושליטה באמצעות מכשירים ניידים.

להלן איור המתאר את מבנה המערכת :



איור מס' 3 – תיאור המערכת

1.1.1. היכרות עם Raspberry Pi 5 והגדרות רשת:

מחשב ה-Raspberry Pi 5 הוא יחידה מרכזית בפרויקט, ומשמש לבקרת רכיבים אחרים באמצעות תקשורת רשת.

מטרת חלק זה היא לתעד את שלבי ההגדרה הראשוניים של ה-Raspberry Pi 5 הכוללים התקנת מערכת הפעלה, בדיקות חומרה ושילובו בתקשורת רשת באמצעות פרוטוקולים.

1.1.1.1. התקנת מערכת האוויר לקירור המעבד ב-Raspberry Pi 5:

על מנת להבטיח תפקוד יציב ויעיל של ה-Raspberry Pi 5-התקנו מערכת קירור ייעודית למעבד. הדגם החדש של ה-RPi5 מציע ביצועים משופרים עם מעבד חזק יותר, אך כתוצאה מכך **מייצר יותר חום** בהשוואה לדורות הקודמים. טמפרטורות גבוהות עלולות לגרום להורדת תדר המעבד (Thermal Throttling) ולפגיעה בביצועים, ולכן התקנת מערכת קירור היא שלב חיוני.

שלבי ההתקנה:

1. בחירת מערכת הקירור – מערכת הקירור שנבחרה כוללת גוף קירור מאלומיניום ומאוורר חיצוני לחיבור ישיר ללוח ה-RPi5.
2. חיבור גוף הקירור למעבד – התקנת גוף הקירור בעזרת דבק תרמי/משחה טרמית לשיפור פיזור החום.
3. חיבור המאוורר ללוח ה-RPi5 – חיבור המאוורר ל-פיני ה-GPIO מתח 5V ו-GND.
4. בדיקת ביצועים לאחר ההתקנה – בדיקה של טמפרטורת המעבד לפני ואחרי התקנת מערכת הקירור, תוך הרצת משימות כבדות (כגון חישובים או עיבוד נתונים) כדי לוודא שהקירור אכן משפר את היציבות.

תוצאות והיתרונות של הקירור:

- הפחתת הטמפרטורה של ה-RPi5 בכ 20°C - 10°C בעת עומס עבודה גבוה.
- שיפור ביצועי המעבד ומניעת האטות הנגרמות כתוצאה מהתחממות יתר.
- הארכת חיי הרכיבים האלקטרוניים ושמירה על פעילות תקינה של המערכת.

1.1.1.2. התקנת מערכת הפעלה והתאמה RPi 5, שלבים לביצוע :

1. הורדת מערכת ההפעלה הרשמית Raspberry Pi OS מהאתר הרשמי.
2. צריבת מערכת ההפעלה לכרטיס microSD באמצעות **Raspberry Pi Imager**.
3. הכנסת הכרטיס ל-Raspberry Pi 5 ואתחול המערכת.

1.1.1.3. בדיקת תקינות GPIO באמצעות שפת תכנות "Python" :

שפת התוכנה שבעזרת נכתבה התוכנה על גבי מחשב RPI5 הינה שפת תכנות בשם - Python. לפני חיבור רכיבים חיצוניים, נבדקה תקינות המערכת וכן תקינות ופונקציונליות של הפינים הדיגיטליים.

קוד הבדיקה:

```
import RPi.GPIO as GPIO
import time
# הגדרת מצב הספינים של ה-Raspberry Pi
GPIO.setmode(GPIO.BCM)

# הגדרת פין 18 כציאה
GPIO.setup(18, GPIO.OUT)
# הדלקה וכיבוי של הפין 5 פעמים
for i in range(5):
    GPIO.output(18, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(18, GPIO.LOW)
    time.sleep(1)
# ניקוי הגדרות ה-GPIO בסיום
GPIO.cleanup()
```

*אם חיבור ל-LED עובד, סימן שהפין תקין.

1.1.1.4. חיבור לרשת ובדיקת קישוריות חיבור (Switch) :

ה-Raspberry Pi 5 חובר פיזית למתג (Switch) באמצעות כבל רשת. המתג מאפשר חיבור בין מספר מכשירים ברשת מקומית ומעביר את נתוני הרשת ביניהם.

• תהליך ובדיקות:

- לאחר החיבור, נבדק שהמחשב מקבל כתובת IP מתאימה או בוצע שינוי ידני של IP (לדוגמה, 192.168.1.1).
- נבדקה הקישוריות לרכיבים אחרים ברשת באמצעות פקודת ping (למשל, לכתובת של הראוטר או מחשבים אחרים ברשת), כדי לוודא שהחיבור פועל כראוי.

• חשיבות:

בדיקה זו מבטיחה שהמחשב מתקשר עם הרכיבים האחרים ברשת, מה שמהווה בסיס לפעילות עתידית עם פרוטוקולי ניתוב ותקשורת מתקדמים.

1.1.2. תכנון החומרה

ה Raspberry Pi 5 - מבוסס על מעבד ARM מתקדם, כולל יכולות עיבוד חזקות יותר, זיכרון RAM משופר, ממשקי קלט/פלט מהירים יותר ותמיכה משופרת בפרוטוקולי תקשורת שונים.

1. **מעבד (CPU):** מעבד **Quad-Core ARM Cortex-A76** במהירות של **2.4GHz** המציע שיפור משמעותי בביצועים.
2. **זיכרון (RAM):** זמינות גרסאות עם **4GB או 8GB LPDDR4X** המאפשרות ריבוי משימות, עיבוד נתונים מתקדם, ושימושים תובעניים יותר.
3. **מעבד גרפי (GPU):** **VideoCore VII**, המספק תמיכה טובה יותר בהאצת חומרה לעיבוד וידאו, גרפיקה תלת-ממדית, ועיבוד תמונה.
4. **אחסון וקלט/פלט:** תמיכה בכרטיס **microSD** לאחסון מערכת ההפעלה והנתונים.
5. **תקשורת וחיבורים:**
 - חיבורי **USB 3.0** ו-**USB 2.0** תומכים בהעברת נתונים מהירה.
 - **WiFi 6** ו-**Bluetooth 5.2** מספקים חיבורים אלחוטיים מהירים ויציבים.
 - **40 פינים GPIO** - לחיבור חיישנים, מנועים ורכיבי חומרה אחרים.

יתרונות השימוש ב- RPi5 בפרויקט:

ה RPi5 - אידיאלי לשימוש בפרויקט מכיוון שהוא משמש כמחשב קרקעי לניהול תקשורת, שליטה ועיבוד נתונים, היתרונות המרכזיים כוללים:

- **עיבוד מהיר של נתונים** - יכול לנתח מידע מחיישנים ולשלוח פקודות לרכיבי קצה.
- **תמיכה בפרוטוקולי תקשורת מתקדמים** MQTT, TCP/IP, UDP, SPI, I2C UART
- **מודולריות ותמיכה בשפות תוכנה מתקדמות** - ניתן להריץ קוד בשפות כמו Python ו C++ לצורך עיבוד נתונים בזמן אמת.
- **GPIO חיבורים רבים** - מאפשר אינטגרציה נוחה עם בקר, ESP חיישנים, מודולי מצלמה, ועוד.

ה Raspberry Pi 5 - מספק פתרון עוצמתי, גמיש וחסכוני לפרויקטים מבוססי IoT, רובוטיקה וניהול תקשורת. בפרויקט הוא מתפקד כמרכז הבקרה שמנתב נתונים בין הרחפן, הבקר (ESP) והשלט הנייד, תוך שימוש בפרוטוקולים מתקדמים לתקשורת אמינה ויעילה.

1.1.3. תכנון התוכנה

תוכנת השליטה והבקרה על רכיב Raspberry Pi 5 נכתבה בשפת Python, תוך שימוש בספריות ייעודיות לניהול תקשורת, GPIO חיישנים וממשק משתמש.

התוכנה נבנתה באופן מודולרי, כאשר כל פונקציונליות נכתבה כיחידה נפרדת, ולאחר מכן שולבה למערכת כוללת.

תהליך הפיתוח התבצע באופן הדרגתי (פרמידה מודול אחר מודול), תוך ביצוע DEBUG ובדיקות שטח כדי להתמודד עם בעיות אמיתיות שעלו, כגון:

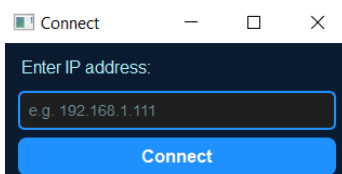
נוחות השימוש והתצוגה בממשק למשתמש, התמודדות עם קלטים בלתי צפויים או מצבי קצה וטיפול בשגיאות חומרה ותקלות תקשורת.

גישה זו איפשרה למערכת להיות יציבה, גמישה ונוחה לתחזוקה, וכן הקלה על הרחבתה בעתיד.

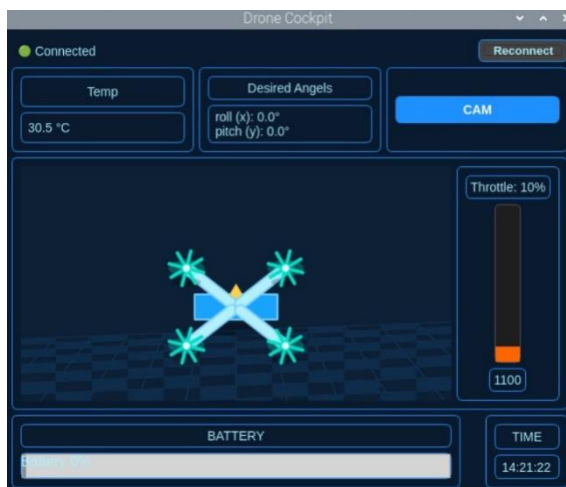
קישור לקוד התוכנה המלא:

https://github.com/yonizvida-gif/Drone-Project/tree/main/RaspberryPi_App

תצוגת מסך הבקרה:



איור מס' 5 – תצוגת מסך הכניסה הראשוני למערכת שוי"ב על גבי RPI5



איור מס' 6 – תצוגת מסך ממשק המשתמש שוי"ב על גבי RPI5

פירוט תצוגת תחנת הניטור והבקרה (איור מס' 5):

התוכנה שפותחה עבור תחנת הניטור והבקרה מציגה ממשק גרפי עשיר ודינמי, המאפשר שליטה מלאה במערכת לצד ניטור בזמן אמת. המסך המרכזי כולל את הרכיבים הבאים:

1. **לד מצב חיבור תקשורת:** תצוגה גרפית של קישוריות החיבור באמצעות פרוטוקול WebSocket. (לד ירוק - מחובר, לד אדום - אין חיבור, לד צהוב - במהלך ניסיון התחברות)
2. **כפתור "Reconnect":** ביצוע חיבור מחדש למערכת במקרה של ניתוק או תקלה.
3. **ניטור טמפרטורה:** הצגת ערכי הטמפרטורה הנמדדים ונשלחים מהרחפן בזמן אמת.
4. **הטיות מבוקשות (Desired Angles):** הצגת ערכי ההטיות הנשלחים משלט המשתמש אל הרחפן, כולל תרגומם לזוויות מספריות.
5. **כפתור צילום (CAM):** מאפשר הפעלה מרחוק של צילום תמונה באמצעות הרחפן ושמירתה במערכת.
6. **מסך מחולק המציג אובייקט:** תצוגת רחפן תלת-ממדית (3D) הדמיה בזמן אמת של הרחפן במרחב, הכוללת שינויים בהטיות בצירים Pitch, Yaw, Roll. בנוסף, מוצגים ארבעת המנועים עם פרופלורים מסתובבים – כאשר מהירות הסיבוב על המסך מתואמת באופן ישיר למהירות המנועים בפועל.
7. **תצוגת מצערת (Throttle):** ערך המצערת המתקבל מהשלט בזמן אמת מוצג במספר אופנים – גם כערך מספרי, גם כערך יחסי (באחוזים מתוך הערך המקסימלי) וגם בתצוגת ויזואלית על גבי BAR שמתמלא בהתאם לכוח המצערת.
8. **חתימת זמן (Timestamp):** שעון עדכני המציג את זמן המערכת בכל רגע נתון.
9. **ניטור מצב הסוללה:** הצגת רמת טעינת הבטרייה של הרחפן בזמן אמת.

1.2. שלט המשתמש - טאבלט (תת מערכת - User Interface)

שלט המשתמש בפרויקט מבוסס על טאבלט המשמש כיחידת שליטה מרכזית עליו מותקנת אפליקציה ייעודית דרכה המפעיל שולט ברחפן ומבצע ניטור בזמן אמת. האפליקציה המותקנת על הטאבלט פותחה באמצעות פלטפורמת MIT App Inventor המאפשרת פיתוח ויזואלי מהיר ונוח של יישומי אנדרואיד, תוך שימוש בבלוקים לוגיים (Visual Programming). השלט מספק ממשק משתמש אינטואיטיבי ונוח, הכולל כפתורים, ג'ויסטיק, סליידרים, גרפים ותצוגות נתונים בזמן אמת. באמצעות השלט ושימוש בפרוטוקול WebSocket נשלחים פקודות שליטה לרחפן (Throttle, Yaw Pitch, Roll) וכן נשלחות פעולות שונות כמו התאמת ערכי PID לבקרה, פתיחת קישוריות לרחפן ולתחנת הניטור וכן פעולות נוספות. כמו כן מתקבלת חזרה אינפורמציה מהרחפן לצורכי ניטור ובקרה וכן לשימוש בקוד של השלט.



איור מס' 7 – תצוגת שלט המשתמש (חומרה ותוכנה)

1.2.1. היכרות עם הטאבלט ופלטפורמת MIT App Inventor:

בפרויקט זה משמש הטאבלט כיחידת שליטה ראשית מרחוק (שלט משתמש), המהווה ממשק קצה ידידותי ונגיש לניהול הרחפן ולניטור הנתונים המתקבלים מהמערכת. הטאבלט נבחר בשל יתרונותיו כפתרון נייד, בעל מסך מגע אינטואיטיבי, המאפשר שליטה נוחה בזמן אמת, ללא צורך בחיבור חוטי או ציוד נוסף. התוכנה המותקנת על גבי הטאבלט פותחה באמצעות פלטפורמת MIT App Inventor סביבת פיתוח ויזואלית מבוססת בלוקים, המיועדת ליצירת יישומים למכשירי Android. MIT App מאפשרת פיתוח מהיר של אפליקציות תוך שילוב ממשק משתמש גרפי אינטראקטיבי עם לוגיקה תכנותית פשוטה, באופן שמקצר את זמני הפיתוח ומקל על שילוב שינויים ותוספות.

יתרונות השימוש בטאבלט עם MIT App Inventor בפרויקט:

- **ניידות מלאה** - מאפשר שליטה מרחוק על הרחפן מכל מקום, באמצעות תקשורת אלחוטית.
- **ממשק משתמש אינטואיטיבי** - מסך מגע גדול וברור המאפשר הצגת נתונים בזמן אמת ושימוש בכפתורים וירטואליים לשליטה.
- **תמיכה בפרוטוקולי תקשורת** - האפליקציה שנבנתה כוללת חיבור ל-WebSocket ופרוטוקולי רשת נוספים, המאפשרים אינטגרציה ישירה עם ה-Raspberry Pi 5 והרחפן.
- **גמישות בפיתוח** - MIT App Inventor מאפשר שינויים קלים ומהירים בקוד ובממשק, כך שניתן להתאים את האפליקציה לצרכים שהתגלו במהלך ניסויי השטח.
- **הרחבה עתידית** - ניתן לשלב בקלות חיישנים וירטואליים, פונקציות נוספות או שליטה באמצעות תנועות (לדוגמה חיישן התאוצה המובנה של הטאבלט).

באמצעות שילוב זה, הטאבלט מתפקד כ"שלט חכם", המספק למשתמש יכולת שליטה בזמן אמת לצד ניטור מלא של המידע המתקבל מהרחפן – בצורה נגישה, ניידת ופשוטה להפעלה.

1.2.2. תכנון החומרה

שלט המשתמש מבוסס על טאבלט אנדרואיד סטנדרטי.

מרכיבי החומרה העיקריים:

1. מסך מגע - מאפשר שליטה נוחה למשתמש באמצעות אצבעות/מחוות.
2. מעבד ARM - מבצע עיבוד מקומי של ממשק האפליקציה.
3. תקשורת WiFi מובנת - משמשת לקישוריות מול ה-Raspberry Pi 5 והרחפן (ESP32).
4. סוללה נטענת - מאפשרת עבודה ניידת לאורך זמן.

יתרונות השימוש המרכזיים בשימוש בטאבלט בפרויקט:

- **ניידות גבוהה** - אין חיבור פיזי של טעינה או כבלי תקשורת.
- **ממשק גרפי נוח וידידותי למשתמש** - תצוגת משתמש נוחה, מסך גדול נוח לתצוגה ותפעול.
- **עיבוד מהיר של נתונים** - יכול לנתח מידע מהמשתמש ולשלוח פקודות לרכיבי קצה.
- **מודולריות וגמישות** - אפשרות הרחבה ושדרוג מהיר באמצעות עדכון האפליקציה



איור מס' 8 – תצוגת חומרת השלט (טאבלט)

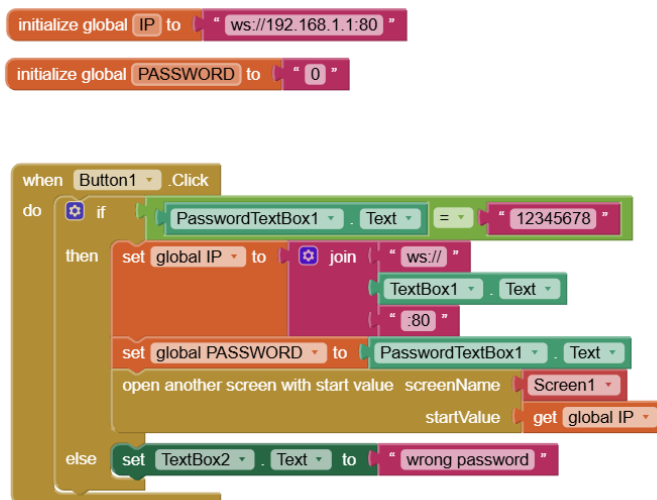
1.2.3. תכנון התוכנה - אפליקציית שליטה (MIT App Inventor)

האפליקציה נכתבה ונבנתה בסביבת MIT App Inventor המאפשרת פיתוח קל ומהיר של יישומי אנדרואיד באמצעות בלוקים של קוד תוכנה מובנה. מאפייני הפיתוח:

- **מודולריות:** כל פונקציונליות (שליחה/קבלה/תצוגה) פותחה כמודול עצמאי וחבורה למערכת הכוללת.
- **שימוש בפרוטוקולים:** התממשקות אל ה-Raspberry Pi 5 וכן אל ה-ESP32 באמצעות WebSocket לשם שליחת פקודות וקבלת מידע.
- **פעולות בזמן אמת:** כל הנתונים המתקבלים מהמשתמש (טמפרטורה, מצב סוללה, זוויות, מצלמה וכו') מוצגים מיידית על המסך ונשלחים לרחפן.
- **התמודדות עם תקלות:** טיפול בניתוקים אוטומטיים, כפתור "Reconnect" ואינדיקציה צבעונית למצב החיבור, כפתור בטיחות לניתוק המנועים מהמערכת, בנוסף אפשרות עדכון תוכנה ושדרוג מהיר שמאפשר טיפול מהיר בתקלות באמצעות עדכון האפליקציה.

קישור לקוד התכנות המלא:

https://github.com/yonizvida-gif/Drone-Project/tree/main/Remote_Controller



איור מס' 9 – דוגמא לבלוק תכנות במסך הכניסה בשלט

פירוט תצוגת שלט המשתמש - User Interface:

התוכנה שפותחה עבור שלט השליטה למשתמש על הרחפן מציגה ממשק גרפי עשיר ודינמי, המאפשר שליטה מלאה במערכת לצד פונקציות מתקדמות לכיול ובטיחות המשתמש.

מסך כניסה:

מסך זה נבנה מתוך הצורך לספק גמישות בשינוי כתובת ה-IP של המערכת ולעבוד מול מחשבים שונים. בנוסף, מסך זה מהווה שכבת אבטחה ראשונית, המונעת גישה של גורמים לא מורשים למערכת. המסך כולל:

- שדה קלט לכתובת IP לצורך התחברות ל-WebSocket.
- שדה להזנת סיסמה שהוגדרה מראש, למניעת התחברות לא רצויה (אבטחה).

OPENscreen

IP:

Password:

Connect

איור מס' 10 – תצוגת מסך הכניסה הראשוני באפליקציה

מסך אינטרקציה:

מסך זה מהווה את לב המערכת ומאפשר שליטה מלאה ברחפן למשתמש לצד תצוגה בזמן אמת של נתונים חיוניים.

בין הרכיבים המרכזיים במסך:

1. **מתג off/on (חיבור/ניתוק תקשורת):** מאפשר פתיחה או ניתוק של התקשורת באמצעות פרוטוקול WebSocket. המצב מוצג באמצעות חיווי גרפי (לד ירוק - מחובר, לד אדום - מנותק).

2. **כפתור "SHUTDOWN":** מאפשר ניתוק כללי של המערכת לצורכי בטיחות, נועל את הרחפן במצב SAFETY כאשר המנועים כבויים ולא ניתן לתת פקודות עד לשחרור המצב.

3. **כפתור "Stop":** כפתור שמשבית את אפשרות לגעת במסך ולשלוח פקודות ע"י המשתמש לרחפן, כמו כן נותן ערך קבוע של CutOff למנועים של 1000uSec. בעת לחיצתו הכפתור מוצג בצבע אדום במצב פעיל.

4. **כפתור "Start":** כפתור שמאפשר חידוש שליחת פקודות לרחפן ושחרור נעילת המסך, מבצע ARMING למנועים ושולח להם ערך Idle התחלתי למנועים של 1100uSec. בעת לחיצתו הכפתור מוצג בצבע ירוק במצב פעיל וכפתור Stop משתחרר.

5. **כפתור צילום (CAM):** מפעיל מרחוק את מצלמת הרחפן ושומר את התמונה במערכת.

6. **ג'ויסטיק וירטואלי:** באמצעות הג'ויסטיק ניתן לשלוט על תנועת הרחפן וההטיות שלו בצירים ROLL ו- PITCH (קדימה/אחורה/ימינה/שמאלה). פעולת המשתמש מתורגמת מקורדינאטות על המסך להטייה במעלות של הרחפן והפקודה לביצוע נשלחת באמצעות תקשורת WebSocket, בעת עזיבה של הג'ויסטיק הרחפן מתייצב לערך קבוע ונשאר בנקי שיווי משקל.

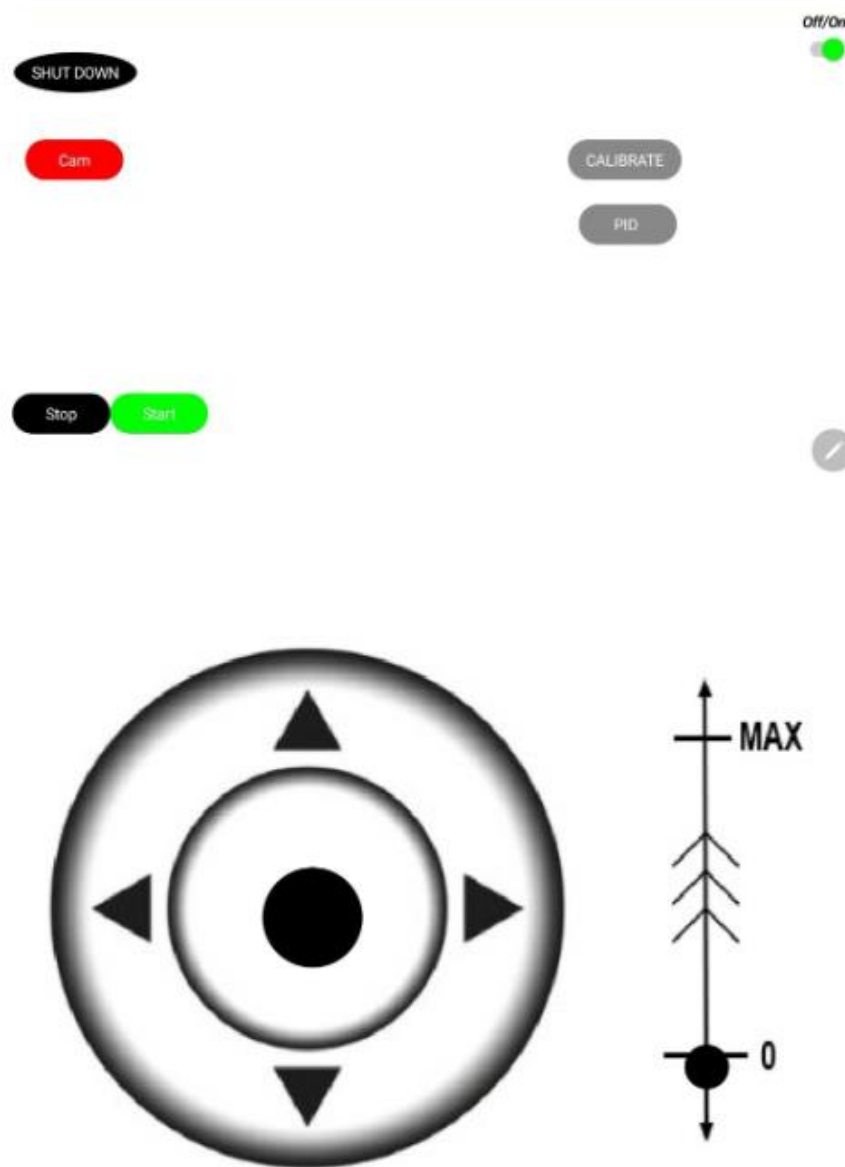
7. **תצוגת מצערת (Throttle):** באמצעות מצערת הגלילה על המסך המשתמש קובע את כוח הדחף שנשלח למנועים, ערך המצערת המתקבל מהמשתמש נשלח בזמן אמת כפקודה לביצוע אל הרחפן. כאשר מתבצעת עזיבה של המצערת כוח הדחף של המנועים נשאר קבוע על הערך האחרון שנשלח.

8. **כפתור PID (מצב כיול):** מאפשר מעבר למסך אחר, מצב כיול מיוחד והתאמת פרמטרים לבקר pid - נצרך לשלב כיול הבקר לייצוב הרחפן באוויר.

9. **כפתור CALIBRATE (מצב כיול):** מאפשר ביצוע כיול מיוחד בעזרת ביצוע מס' רב של מדידות וחישוב ממוצעים של חיישן הגיירו והאקסלומטר.

* **כפתורי כיול-PID Calibrate שימשו בעיקר בשלבי הפיתוח, ניתן לנעול אותם בשימוש גרסת קצה.**

תצוגת מסך האינטרקציה הראשי:



איור מס' 11 – תצוגת המסך הראשי באפליקציה

1.3. הרחפן - (תת מערכת אווירית)

מבט על ארכיטקטורת הרחפן:

מטרת פרק זה היא לתאר את רכיבי הרחפן (תת מערכת מרכזית), האינטגרציה, יחסי הגומלין ביניהם, מחקר ושיקולי התכן המרכזיים לתת מערכת זו.

הרחפן הינו בצורת X-Quadroper מבוסס שלדה F-450, כולל 4 מנועי Brushless, בקרי מהירות (ESC), סוללת Li-Po 3S, בקר טיסה הממומש באמצעות אלגוריתמים של חיישנים, חומרה ותוכנה על גבי בקר ESP32, מודול Wi-Fi לתקשורת, WebSocket ותאורת LED לחיווי מצבים.

הדגש ההנדסי הוא על יציבות, תקשורת אמינה, זמני תגובה נמוכים ואמינות אספקת-הכוח.

1.3.1. תכנון המכניקה

1.3.1.1. שלדה וגוף הרחפן

לצורך הפרויקט נבחרה שלדת F450 פלטפורמת X נפוצה (Quad) המתאימה לפרויקטי R&D בזכות:

- זמינות חלקים: זרועות/פלטות חלופיות וזולות, זמן השבתה מינימלי בתקלות.
- מרחב עבודה: פלטות עליונה/תחתונה רחבות למיקום בקר, ESC חיישנים וסוללה.
- קשיחות/משקל: שילוב זרועות פלסטיק קשיח ופלטות PCB, איזון טוב בין קשיחות למשקל.

מבנה כללי: 4 זרועות פלסטיק מוברגות בין שתי פלטות (עליונה/תחתונה).

הפלטת התחתונה משמשת גם כ־ **PDB** (לוח חלוקת הספק) באמצעות משטחי נחושת להלחמת הזנות.

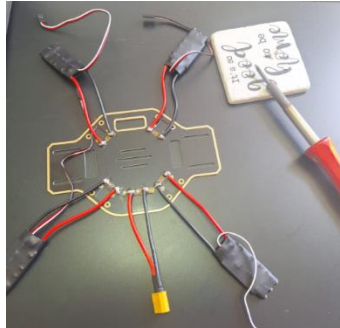


איור מס' 12 – שלדת הרחפן בחלקים

חיווט והלחמות על הגוף

בשלדת ה-F450 בוצעו הלחמות בדיל לפלטת ה-PDB לצורך הזנת כוח מרכזית:

- **כניסת סוללה:** מחבר XT60 מולחם לפדים הראשיים (+/-).
- **יציאות ל-ESC:** זוגות חיבורי הספק מולחמים לפדים שיועדו לכך, עם סימון קוטביות.
- **הגנת קצוות:** כיסוי נקודות חשופות בסיליקון למניעת קצרים.
- **ניהול כבלים:** קשירות הכבילה בעזרת אזיקונים ויצירת תעלות חיווט לאורך הזרועות.



איור מס' 13 – תיאור הלחמות לגוף הרחפן

חומרים ושיטת הרכבה

חומרים: זרועות פלסטיק (לבן/שחור לזיהוי קדימה/אחורה), פלטת PCB.
חיבורים מכניים: ברגי M2.5/M3 עם אומי נעילה.
כיוון זרועות: זרועות קדמיות בצבע שונה (לבן/שחור) לשיפור אוריינטציה חזותית.
סידור שכבות:

- תחתונה: לוח חלוקת הספק PDB, הכולל כניסת סוללה ויציאות ל-ESC.
- עליון: לוח ישר להנחת בקר ה-ESP32 ורכיבי החומרה.
- חלל בין עליון לתחתון: מיקום לסוללה עם רצועות להידוק + משטח נגד החלקה.



איור מס' 14 – שלדת הרחפן לאחר הרכבה

1.3.1.2. מנועים + פרופלורים

בחירת סוג מנוע - Brushless (BLDC):

בפרויקט נבחרו מנועי Brushless DC בשל יתרונות מהותיים לעומת מנועי DC:

- **יעילות גבוהה:** פחות הפסדים לחיכוך/חימום - זמן טיסה ארוך יותר לאותה קיבולת סוללה.
 - **יחס הספק/משקל מצוין:** הספק גבוה במארז קטן וקל - מתאים לרחפנים.
 - **אמינות ועמידות:** תחזוקה מינימלית ויציבות בביצועים לאורך זמן.
 - **שליטה מדויקת במהירות/תגובה:** יחד עם ESC מתקדם מתקבלת תגובה מהירה - קריטי לייצוב.
- לרחפן מרובה-להבים (Quad) נדרש סט של 4 מנועים קלים, יעילים ואמינים - ולכן מנועי BLDC הם הבחירה הסטנדרטית והנכונה.

דירוג KV והמשמעות לבחירה (נבחר 920KV):

דירוג KV מבטא את מהירות הסיבוב ללא עומס ביחס למתח:

$$KV \approx [RPM/V] \text{ מהירות מנוע ללא עומס}$$

$$K_t \approx \frac{9.55}{KV} \text{ הקשר בין KV למומנט}$$

כלומר K_t ביחס של ניוטון לאמפר מבטא את הכח שמופעל.

- מתאים לפלטפורמות בגודל 450 mm כמו F450 עם פרופלורים בקוטר 9–10 אינץ'.
- מספק איזון טוב בין מומנט (להנעת פרופלור גדול יחסית), צריכת זרם ויכולת נשיאה.
- עובד היטב עם סוללות LiPo תוך שמירה על זרמים/טמפרטורות סבירים.

כלל אצבע: ככל שהרחפן גדול וכבד יותר – נעדיף KV נמוך יותר עם פרופלורים גדולים יותר; לרחפנים קטנים וקלים – KV גבוה ופרופלורים קטנים/מהירים.



איור מס' 15 – Brushless Motor 920KV

בחירת פרופלורים – קוטר וחומרים:

הפרופלורים שנבחרו הם בקוטר 10 ופיץ' 4.5 : (10x4.5 mm), השיקולים:

- קוטר משפיע בעיקר על ספיקת האוויר והדחף המרבי.
- פיץ' (Pitch) משפיע על "צעד" האוויר לסיבוב - על המהירות הקדמית והזרם.
- התאמה ל-KV920 וסל"ד העבודה הרצוי כך שצריכת הזרם תישאר בגבולות המנוע/ESC והחום יהיה נמוך.

החומר שנבחר הינו פלסטיק/ניילון מחוזק – חסכוני, סלחני לפגיעות, קל להשגה.

CW/CCW: יש להרכיב זוגות נגדיים (שניים מסתובבים עם כיוון השעון ושניים נגד) לפי דיאגרמת הסיבוב של הבקר כדי לבטל מומנט נטו.



איור מס' 16 – Propellers 1045

1.3.1.3 סוללה 3S LiPo

נבחרה סוללת LiPo S3 (Lithium-Polymer) (שלושה תאים בטור), מאחר ש-S3 מספקת שילוב טוב בין מתח, משקל, זמינות ורמת סיכון ניהולית.



איור מס' 17 – 1045 Propellers

- יחס אנרגיה/משקל גבוה (Wh/kg) - זמן טיסה טוב יחסית למשקל.
- גודל יחסית קטן ומתאים לגודל הרחפן.
- יכולת פריקה גבוהה - יכולות לספק זרמים שיא גבוהים הדרושים למנועי Brushless.
- זמינות מסחרית רחבה, עלות סבירה והתאמה לרוב מחברי כוח סטנדרטיים.

טבלה מס' 2 – מפרט הסוללה שנבחרה

LiPo 3S (3-cell)						
מידות	משקל	מחבר	דירוג פריקה [C]	קיבולת	מתח טעון מלא	מתח נומינלי
94*34*28 (mm)	182 (g)	XT60	45C	3000 (mAh)	12.6 (V)	11.1 (V)

נוסחאות חשובות בעת תכנון בחירת הסוללה:

1. חישוב זרם מקסימלי תאורטי (Peak Discharge):

כאשר C = דירוג הפריקה

$$\frac{Capacity_{mAh}}{1000} * C = I_{max}$$

$$\frac{3000}{1000} * 45 = 135$$

$$I_{max} = 135(A)$$

מסקנה: סוללה זו יכולה לספק תאורטית עד **135(A)**.

2. חישוב צריכה ממוצעת וזמן טיסה תיאורטי:

$$I_{AVG_mA} = \frac{N_{motors} * I_{avg1motor}}{1000}$$

$$\frac{Capacity_{mAh}}{I_{AVG_mA}} * 60 = t_{min}$$

$$\frac{3000}{4 * X} * 60 = t_{min}$$

1.3.1.4 Electronic Speed Controller – ESC

ESC הוא רכיב שמתווך בין הסוללה למנועים של הרחפן. הוא מתרגם את אותות השליטה מבקר הטיסה (FC – Flight Controller) לאות חשמלי שמניע את המנוע בצורה מדויקת ובטוחה.

ב-ESC שולטים דרך בקר הטיסה בעזרת אות PWM בטווח בין 1000-2000(uSec). הרכיב שנבחר להשתמש בפרויקט הינו ESC מסוג 30A בזכות הפרמטרים והיתרונות שלו:

- **אמפר גבוה:** ה-30A מייצג את המקסימום שה-ESC יכול לספק למנוע. במקרה שלנו, אם המנועים דורשים עד 25A, בשיא, ESC של 30A נותן שוליים בטוחים.
- **בטיחות:** ה-ESC שנבחר אינו קרוב למגבלת הצריכה של המנוע, כלומר הבחירה בוצעה מתוך התחשבות למזעור סיכון להתחממות או כיבוי פתאומי. השוליים שמספק 30A מגנים על הרכיב.
- **תאימות למתח:** ה-ESC מתאים למתח של הסוללה שנבחרה. LiPo S3.



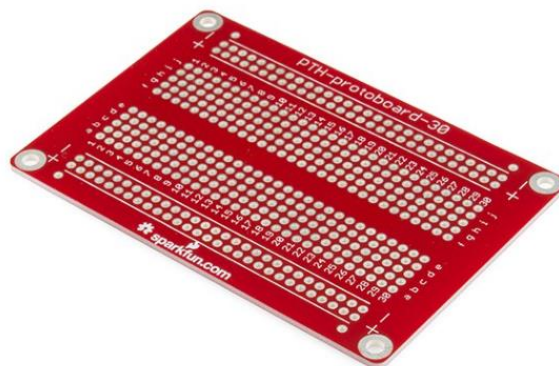
איור מס' 18 – ESC Brushless 30A

1.3.2. תכנון החומרה

תכנון החומרה בפרויקט זה בוצע במטרה ליצור מערכת רחפן יציבה, אמינה ובטיחותית, תוך שמירה על יעילות אנרגטית ונוחות תחזוקה. כל רכיב נבחר בהתאמה לדרישות המנועים, הסוללה ובקר הטיסה, תוך התחשבות במתח, זרם, משקל וגישה לניסויים בשטח. החומרה הוכנה והולחמה תחילה על גבי מטריצה ניסיונית (Breadboard/PCB ניסיונית) כדי לאפשר בדיקות, התאמות ותיקונים מהירים לפני ההתקנה הסופית. בתכנון נשקלו גם עקרונות נוספים, כגון:

- סידור רכיבים: הצבת הרכיבים בצורה שמקלה על ניהול כבלים ומקטינה הפרעות חשמליות.
- קירור והגנה: שמירה על פיזור חום מתאים והגנה מפני עומס יתר או קצר חשמלי.
- גמישות ניסיונית: תכנון המאפשר החלפה או עדכון של רכיבים בהתאם לצרכי ניסוי או שדרוג עתידי.
- בטיחות ותפעול: מקסום נוחות החיבור והניתוק של הרכיבים תוך שמירה על בטיחות המשתמש ורכיבי המערכת.

גישה זו מבטיחה כי החומרה לא רק תעמוד בדרישות הבסיסיות של הרחפן, אלא גם תאפשר פיתוח עתידי, תחזוקה נוחה ובדיקות אמינות בשטח.



איור מס' 19 – תיאור מטריצת החומרה

1.3.2.1. בחירת החומרה והבקר המרכזי של המערכת

בשלב זה של הפרויקט, מטרתנו הייתה להתחיל בבניית ליבת הרחפן ולבחון את הפלטפורמות האפשריות להטמעת הבקר המרכזי.

שתי אפשרויות עיקריות שנבחנו הן בקר Arduino ו-ESP כאשר המטרה הייתה לבחור את הבקר האופטימלי שיאפשר שליטה מיטבית ברחפן, תקשורת רציפה עם מחשב הקרקע (Raspberry Pi 5) ושילוב עם אפליקציה ייעודית במכשיר הסלולרי.

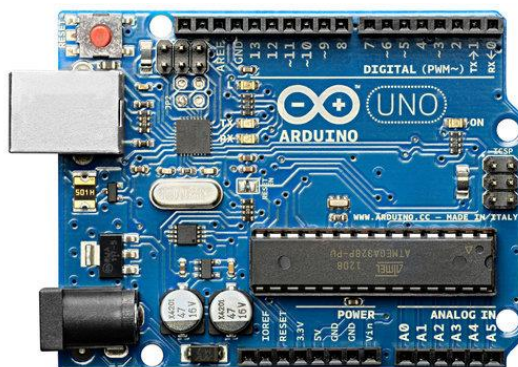
בחירת פלטפורמות הבקרה:

Arduino

היא פלטפורמה פופולרית מאוד בקרב מפתחי חומרה בזכות הפשטות שלה, התמיכה הרחבה והיציבות. השימוש בארדואינו לצורך שליטה ברחפן נבחן באמצעות חיבור חיישנים ורכיבים חיוניים, כולל גייروسקופ, מד תאוצה ובקרי מהירות (ESCs).

עם זאת, במהלך הבדיקות התגלו מספר מגבלות:

- קישוריות מוגבלת -חיבוריות WiFi מובנית אינה זמינה, מה שמחייב שימוש במודולים חיצוניים כגון ESP8266
- ביצועים מוגבלים -קצב עיבוד נתונים נמוך יחסית, מה שעלול להשפיע על התגובה בזמן אמת.
- תמיכה בפרוטוקולי תקשורת -דורש הרחבות חיצוניות לתקשורת מתקדמת.

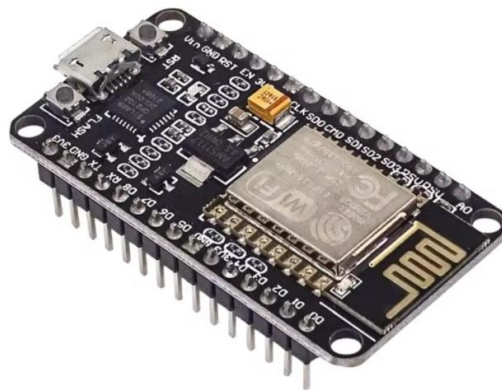


איור מס' 20 – בקר ARDUINO

ESP (ESP32/ESP8266)

לעומת Arduino בקרי ESP הציגו יתרונות משמעותיים בעת הבדיקות:

- קישוריות WiFi ו Bluetooth-מובנית -מאפשרים תקשורת ישירה ומהירה.
- עיבוד מהיר יותר, ESP32- לדוגמה, כולל מעבד חזק יותר ומאפשר עיבוד נתונים בזמן אמת.
- תמיכה בפרוטוקולי תקשורת מתקדמים MQTT, WebSockets ו HTTP מאפשרים תקשורת רציפה עם הרחפן והמערכות הסובבות אותו.
- צריכת חשמל נמוכה יחסית -דבר התורם ליעילות המערכת ולשיפור חיי הסוללה של הרחפן.



איור מס' 21 – בקר ESP32

ניסויים ובחירת הבקר:

כדי לקבל החלטה מושכלת, בוצעו ניסויים עם שני הבקרים.

במסגרת הניסוי, הופעלו ונלמדו 2 לוחות הבקרה Arduino ו ESP- תוך כדי ביצוע בדיקות לשליטה מרחוק, ביצוע פעולות פונקציונליות שונות, עיבוד נתוני חיישנים, תגובה לפקודות מהאפליקציה וביצוע תקשורת עם ה-RPI5.

רכיבים דרושים

- מחשב Raspberry Pi 5
- בקר Arduino Uno
- בקר ESP
- חיישן מרחק Ultrasonic
- נוריות LED בשלושה צבעים (אדום, צהוב, ירוק)
- נגד 220Ω לכל נורת LED
- חוטי קצר ומטריצת חיבורים

1. ניסוי בדיקת פונקציונליות ועיבוד נתוני חיישנים:

בניסוי זה נבדק תהליך חיבור בקר ESP ArduinoUno למדידת נתונים המתקבלים מחיישנים שונים וכמו כן ביצוע פעולות שונות בעזרת חיוויים אלו.
כמו כן בניסוי זה נבחנה הפונקציונליות ונוחות כתיבת הקוד והשימוש בכל אחד מלוחות הבקרה הנתונים.

2. ניסוי בדיקת קישוריות:

בניסוי זה נבדק תהליך חיבור בקר ESP ArduinoUno / למחשב Raspberry Pi 5 באמצעות תקשורת טורית (Serial).
המטרה היא אפשרות תקשורת רציפה וחלקה בין שני הרכיבים כך שניתן יהיה לשלוח פקודות מהמחשב לבקר ולבצע פעולות בהתאם.
בתחילה חובר לבקר נורת LED אשר נדלקת ונכבית על פי פקודות הנשלחות מהמחשב. בהמשך הוספנו חיישן מרחק על מנת ליצור מערכת מבוססת מרחק, כאשר כל טווח מרחק מפעיל נורה בצבע אחר, והמרחק מנוטר ומוצג במחשב דרך שפת Python.
כמו כן בניסוי זה נבחנה אופן וביצוע התקשורת של כל אחד מהבקרים אל מול ה-RPI5 וזמן התגובה לתקשורת בין המערכות.

מסקנה וממצאי הניסויים:

- בקר הארדואינו תפקד כראוי אך הצריך תוספות חיצוניות משמעותיות לצורך תקשורת אלחוטית.
- ה-ESP הציג תגובה מהירה יותר, שילוב מובנה של תקשורת WiFi, ויכולת עיבוד גבוהה יותר, מה שהפך אותו לאופציה העדיפה.

בהתאם לתוצאות הניסוי והבדיקות, הוחלט להשתמש ב-ESP כמעבד הראשי של הרחפן.
הבחירה בו התבססה על יתרונותיו הרבים בתחום הקישוריות, הביצועים וההתאמה לדרישות הפרויקט.
כמו כן עוד יתרון מרכזי הוא המשקל הקל יחסית ללוח הארדואינו ולכך שאין צורך לחבר לו רכיבים נוספים לביצוע התקשורת.

37

טבלה מס' 3 – פירוט הפינים המחוברים לבקר ה- ESP

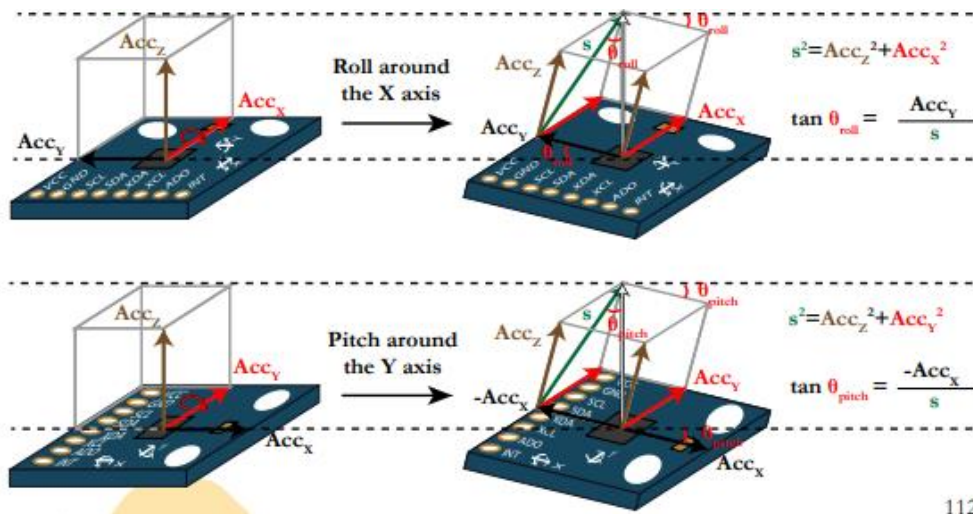
פירוט	TYPE	ייעוד	Pinout
חיבור כל פין מסוג PWM בהתאם ליציאה מהבקר טיסה ל- ESC וממנו מועבר למנוע המתאים. האות יוצא בטווח בין 1000-2000uSec. הרזולוציה של הפינים האלו מוגדרת ל- 12 ביט.	PWM	ESC 1 (MOTOR1)	#5
	PWM	ESC 2 (MOTOR2)	#6
	PWM	ESC 3 (MOTOR3)	#9
	PWM	ESC 4 (MOTOR4)	#10
פינים דיגיטלים המחוברים מהבקר לנורות הLED לחיווי תקשורת. מקבלים ערך לוגי – '0' או '1'.	Digital Pin	RED LED	#12
	Digital Pin	GREEN LED	#13
פין אנאלוגי בעל רזולוציה של 12 ביט לחישוב וניטור מתח הסוללה. מקבל מתח מסוללת הליתיום בעזרת מחלק מתח לשמירת על טווח מתחים תקין בכניסה לבקר.	ADC	ניטור סוללה	#14
פינים שבעזרת ניתן לנהל ולהעביר תקשורת בפרוטוקול I2C בין חיישן הגייירו לבין הבקר. קצב הנתונים מוגדר למקסימלי : 400KHz	I2C	SCL	#19
	I2C	SDA	#20
פינים קבועים במערכת המספקים מתח ברמה קבועה ומוגדרת או לחילופין פין המייצג אדמה. דרך פין המתח של ה- 5V גם מתבצעת הזנת המתח מה- ESC שמקבל מהסוללת ליתיום הראשי אל ה- ESP.	Voltage	אספקת מתח	5V
	Voltage	אספקת מתח	3.3V
	Ground	אדמה	GND

1.3.2.3. חומרה משלימה – חיישנים, לדים ורכיבי קצה (GPIO)

כפי שניתן לראות באיור מס' 23 המתאר את הסכמה החשמלית לבקר המרכזי (ESP) מחוברים מספר רכיבי חומרה משלימים ביניהם מקורות מתח, חיישנים, לדים, נגדים, קבלים ורכיבי קצה נוספים. בעזרת רכיבים משלימים אלו ניתן לבצע ולעבד את והפעולות הנצרכות השונות על גבי הבקר למען הפעולות המרכזיות שהיא להפעיל בקר טיסה מתוחכם ובקר תקשורת מרכזי במערכת.

1. חיישן MPU6050 - חיישן תאוצה וג'ירו

אחד מרכיבי הקצה המרכזיים במערכת הוא חיישן ה-MPU6050, המשלב בתוכו אקסלרומטר (Accelerometer) וג'ירוסקופ (Gyroscope) הפועלים ב3 צירים X,Y,Z. החיישן מודד את התאוצה הליניארית של הגוף ואת קצב הסיבוב הזוויתי, ובכך מאפשר חישוב מדויק של זווית הנטייה, התנועה והכיוון במרחב. בנוסף, החיישן כולל מדידת טמפרטורה פנימית לצורך כיול ובקרת יציבות הנתונים. החיישן מתקשר עם הבקר (ESP) באמצעות פרוטוקול התקשורת I²C המאפשר העברת נתונים דיגיטליים בצורה יעילה ומהירה. במסגרת המערכת בוצעה קריאה של הנתונים מהרגיסטרים הפנימיים של החיישן, המרת ערכים גולמיים (Raw Data) ליחידות פיזיקליות (תאוצה במטר לשנייה בריבוע, מהירות זוויתית במעלות לשנייה).



112

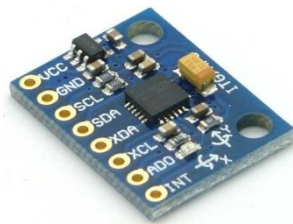
איור מס' 24 – המחשת חישובי טריגונומטריה לצירי 3D מהחיישן

בנוסף יושמו אלגוריתמים לעיבוד נתונים לצורך חישוב מיקום, זווית ויציבות בזמן אמת.
להלן נוסחאות החישוב והמרת נתוני החיישן לקורדינאטות בצירים, כפי שהומחש באיור מס' 24 :

$$AngleRoll = atan(AccY / sqrt(AccX * AccX + AccZ * AccZ)) * 1/(3.142/180)$$

$$AnglePitch = -atan(AccX / sqrt(AccY * AccY + AccZ * AccZ)) * 1/(3.142/180)$$

באמצעות ה־ MPU6050 מתקבלת תמונה דינמית ומדויקת של תנועת המערכת במרחב, דבר המאפשר שליטה ובקרה חכמה בהתאם ליישום הנדרש בפרויקט.



איור מס' 25 – חיישן MPU6050

2. נורות חיווי (LED)

רכיב חומרתי נוסף במערכת המתקשר לבקר המרכזי הינו נורות הלד (Led).
נורות הלד במערכת נוספו למטרת הצגת חיווי ויזואלי למשתמש להצגת מצב המערכת ומצב התקשורת במערכת וכן להוספת רובד בטיחות נוסף למשתמש.
הצבעים שנבחרו לנורות החיווי הינם ירוק ואדום אשר אלו צבעים סטנדרטים המסמלים מצב תקין/לא תקין.



איור מס' 26 – נורות LED

3. ניטור סוללה חומרתי

במערכת בוצע ניטור חומרתי ותוכנתי של מתח סוללת הליתיום המרכזית במטרה לעקוב אחר רמת הטעינה ולזהות מצבי ירידת מתח בזמן אמת.

לצורך כך, נבנה מחלק מתח המורכב משני נגדים המחוברים בטור בין הדק הסוללה לבין פין הכניסה של ממיר האנלוגי-לדיגיטלי (ADC) בבקר ה-ESP, כאשר המתח הנקרא בפין ה-ADC מחושב לפי הנוסחה:

$$V_{ADC_pin} = V_{batt} \cdot \frac{R_2}{R_1 + R_2} = V_{batt} \cdot 0.1639$$

כאשר מתקיים:

$$10.6(V) \leq V_{batt} \leq 12(V)$$

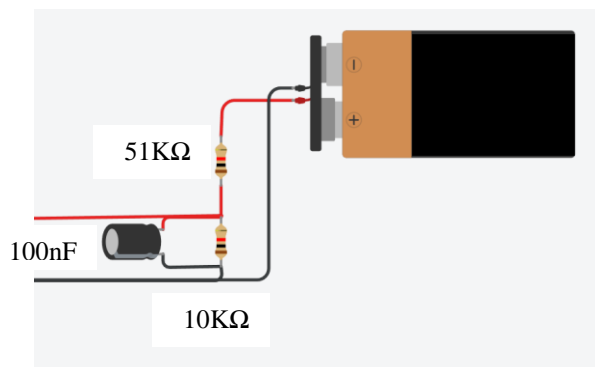
כלומר:

$$1.74(V) \leq V_{batt} \leq 1.967(V)$$

מחלק המתח מאפשר הקטנת מתח הכניסה לערך בטוח המתאים לטווח העבודה של כניסת ה-ADC (עד 3.3 וולט לכל היותר ומעל 1 וולט) ובכך ניתן למדוד את מתח הסוללה הכולל באופן עקיף אך מדויק. בנוסף, חובר קבל (קבל משאבה) במקביל לפין ה-ADC לצורך החלקת האות ומניעת תנודות ורעשים הנובעים משינויים רגועים במתח, באופן זה מתקבלת קריאה יציבה ואמינה של מתח הסוללה. בצד התוכנה, פותחה פונקציה המבצעת קריאה תקופתית של ערך ה-ADC, המרה לערך מתח ממשי (וולט), לפי הנוסחה:

$$V_{Digital_SW} = \frac{V_{ADC_pin} \cdot 3.3}{4095}$$

בנוסף בוצעה בתוכנה השוואה לספים שהוגדרו מראש לצורך התרעה והצגת מצב טעינה למשתמש.



איור מס' 27 – ניטור סוללה חומרתי

1.3.3. תכנון התוכנה

התוכנה שנכתבה על הבקר נכתבה בשפת התוכנה C בפלטפורמת IDE. תכנון התוכנה של בקר הטיסה (ESP32) מבוסס על עקרון מודולריות וזמן אמת (Real Time). כל פונקציונליות מרכזית מיושמת כמודול נפרד והריצה מתבצעת באמצעות משימות FreeRTOS שמורצות עלפי עדיפויות ויכולות Multi-Core של ה-ESP32. המטרה היא להשיג תגובתיות בזמן אמת, בידוד תקלות וקלות תחזוקה/כיוול.

להלן קישור לקוד המלא לבקר הטיסה ולבקר המרכזי שנכתב על ESP32 :

https://github.com/yonizvida-gif/Drone-Project-Michael-Yoni/tree/main/ESP32_FlightController/ESP32_Flight_Controller

1. ארכיטקטורת קוד (Multi-Core/Task Separation)

בקוד יש הפרדה ברורה בין משימות מדידה וקריאה שונות, בקורות לולאה (PID), תקשורת בין הרכיבים השונים ושליחת אותות (PWM) למנועים. בקוד בוצעה שימוש ב-FreeRTOS ע"י 2 ליבות (Core0, Core1) להרצת משימות מפוצלות כדי לייעל ולעמוד בזמנים מדויקים וקצובים לביצוע לולאות מסוימים והתקרבות לזמן אמת לפעולות אלו. כמו כן בוצעה תקשורת פנימית בין 2 הליבות באמצעות queues ו-ring buffers למניעת חסימות זמן ארוכות.

המשימות מחולקות לשני הליבות על גבי הבקר כפי המתואר בטבלה הבאה :

טבלה מס' 4 – פירוט חלוקת הליבות בבקר

תדירות	תפקיד	משימה (Task)	ליבה (Core)
2mSec	קריאות מהחיישן – I2C	Sensor Task	CORE 1
20uSec	חישוב מדידות מחושבות – LPF ופילטר קלמן	Filter Task	
35uSec	חישובי PID ושליחת ערכי PWM בהתאם למנועים	Control Task	
RealTime	האזנה וקבלת הודעות – Server	WebSocket	CORE 0
3mSec	שליחת הודעות ל- Clients		
10mSec	ניטור סוללה קבוע ושליחת הנתונים בBroadcast	Battery Monitor	

2. מודול קריאת חיישנים והמרת יחידות פיזיקליות

קריאת הנתונים מחיישן ה־MPU6050 מתבצעת באמצעות פרוטוקול התקשורת I²C, הפועל על שני קווי תקשורת עיקריים: SDA (Serial Data) קו הנתונים ו-SCL (Serial Clock) קו השעון. בצד התוכנה, בוצעה התקשורת ישירה לכתובת החיישן על גבי קווי התקשורת, לצורך שליפת הנתונים הגולמיים מהרגיסטרים הפנימיים של ה־MPU6050. הקריאה בוצעה בצורה ישירה (Direct Register Access) כלומר, ללא שימוש בספריות חיצוניות או במעטפות תוכנה מוכנות מראש, בחירה זו נועדה להשיג:

- זמן תגובה מהיר יותר - על ידי ביטול עיכובים מיותרים של פונקציות ספרייה.
- גבוהה יותר - שליטה מלאה ברצף הקריאות ובניהול הזמנים.
- דיוק משופר - קבלת נתוני גיירו ואקסלרומטר "נקיים" ללא עיבוד נוסף.

באופן זה, התוכנה ניגשת ישירות אל כתובות הרגיסטרים של החיישן, קוראת את ערכי ה־LSB וה־MSB מכל ערוץ וממירה אותם לערכים פיזיקליים של תאוצה ומהירות סיבוב. להלן תיאור שורות הקוד המרכזיות לביצוע הקריאה מהחיישן:

```
int16_t AccXLSB = Wire.read() << 8 | Wire.read();
int16_t AccYLSB = Wire.read() << 8 | Wire.read();
int16_t AccZLSB = Wire.read() << 8 | Wire.read();
int16_t GyroX=Wire.read()<<8 | Wire.read();
int16_t GyroY=Wire.read()<<8 | Wire.read();
int16_t GyroZ=Wire.read()<<8 | Wire.read();
```

איור מס' 28 – תיאור שורות קוד לקריאה מרגיסטרים

כמו כן, הנתונים שנקראו מהרגיסטרים של החיישן נשמרו במשתנים שונים בתוכנה, ועליהם בוצעה המרה ליחידות פיזיקליות אמיתיות (תאוצה במונחי g, ומהירות סיבוב במעלות לשנייה). בנוסף, יושמו על הנתונים אלגוריתמים וחשובים מתמטיים לצורך הפקת קורדינאטות, טמפרטורה, וכן חישוב זוויות הנטייה והגלגול של הרחפן בזמן אמת. המרה משפת המכשיר לערכים פיזיקליים של מהירות סיבוב ($s/^{\circ}$):

$$\text{RateRoll} = (\text{float})\text{GyroX} / 65.5$$

$$\text{RatePitch} = (\text{float})\text{GyroY} / 65.5$$

$$\text{RateYaw} = (\text{float})\text{GyroZ} / 65.5$$

החיישן מודד תאוצה על שלושת הצירים X,Y,Z -
כאשר הרחפן במצב סטטי, תאוצת הכובד (g) מתפצלת בין הצירים לפי הזוויות שלו.
לכן ניתן להעריך את הזוויות ביחס לאופק בעזרת נוסחאות טריגונומטריות:

$$AngleRoll = atan(AccY / \sqrt{AccX * AccX + AccZ * AccZ}) * 1/(3.142/180)$$

$$AnglePitch = -atan(AccX / \sqrt{AccY * AccY + AccZ * AccZ}) * 1/(3.142/180)$$

התוצאה שמתקבלת מחישוב זה הינה מיקום גוף הרחפן במישור 3D והמרת התוצאה מרדיאנים למעלות.

מיקום הרחפן במרחב 3D מיוצג ע"י מערכת הצירים:



איור מס' 29 – תיאור מערכת צירים לרחפן

Pitch – הטייה אחורה או קדימה

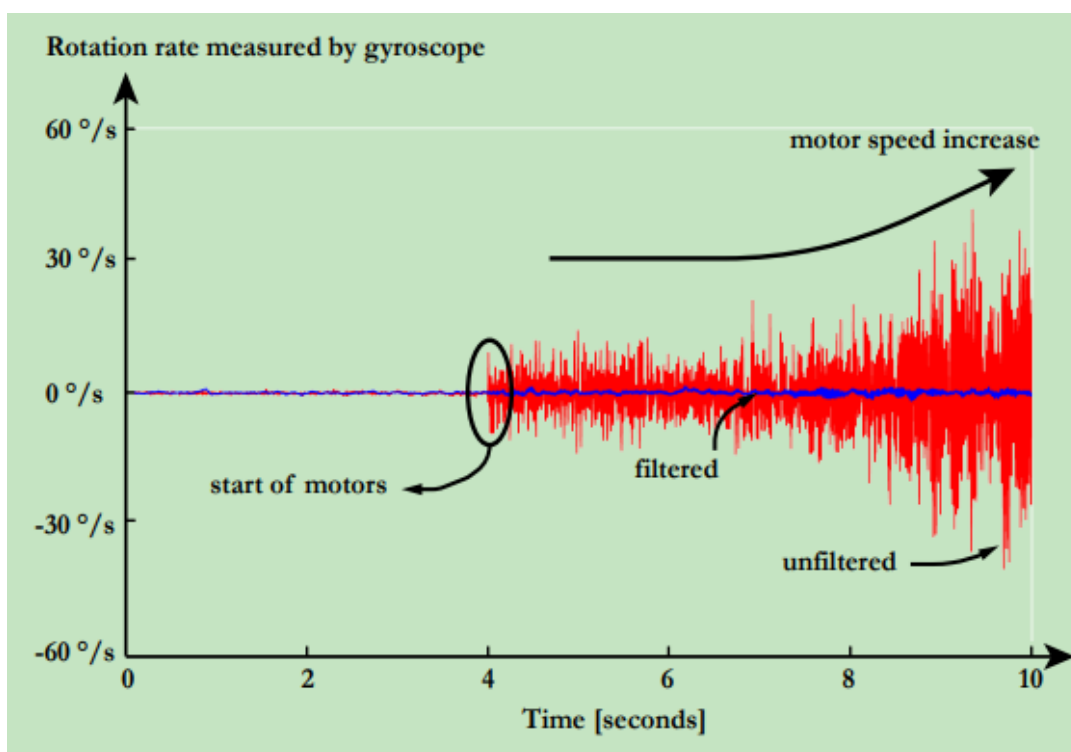
Roll – הטייה שמאלה או ימינה

Yaw – הטייה בציר הסיבובי

3. בידוד ופילטר רעשים – LPF ופילטר קלמן

בקרת טיסה מדויקת מחייבת שימוש בנתונים נקיים ויציבים המתקבלים מחיישני ה-IMU. המדידות הגולמיות מהחיישן מכילות רעש חשמלי ומכאני הנובע מרעידות המנועים, שינויי טמפרטורה, הפרעות אלקטרו מגנטיות ורעשי תקשורת ב-I²C. לכן בוצעה בתוכנה שכבת פילטרים שמטרתם לנקות את האותות לפני שימושם בלולאות הבקרה (PID).

הסינון הראשוני של הנתונים התבצע בעזרת פילטר ממוצע נע - Low Pass Filter (LPF). פילטר LPF נועד לעבור על מדידות רצופות ולשכך תנודות מהירות (High Frequency Noise) תוך שמירה על השינוי האיטי והרצוי של האות.



איור מס' 30 – הצגת פילטר באמצעות LPF

בנוסף לכך במערכת הרחפן, לכל אחד משלושת הצירים X, Y, Z מופעל פילטר נפרד עבור תאוצה (Acc) ומהירות סיבוב (Gyro) והערכים המסוננים משמשים כקלט לפילטר הקלמן ולבקר ה-PID.

פילטר קלמן (Kalman Filter):

לאחר סינון בסיסי באמצעות LPF, עדיין קיים פער בין נתוני התאוצה ונתוני הגיירו. מד התאוצה מספק אינדיקציה יציבה אך איטית, בעוד הגיירו מספק נתון חלק ומהיר אך נוטה לסטייה. לכן נדרש מנגנון אינטליגנטי לאיחוד שני המקורות - זהו **פילטר קלמן**. פילטר קלמן הוא אלגוריתם הערכה סטטיסטית המחשב בכל רגע את המצב האמיתי של המערכת, על סמך מדידות מרובות ורמות אי-ודאות שלהן, האלגוריתם מתבצע בשני שלבים מחזוריים:

- Prediction (חיזוי):

מתבסס על המהירות הזוויתית מהגיירו כדי לחשב מה צפויה להיות הזווית החדשה.

$$\theta^{pred} = \theta^{prev} + Rate_{gyro} \cdot dt$$

- Correction (תיקון):

משתמש במדידה מהאקסלרומטר כדי לתקן את הערך המשוער לפי רמת האמון.

$$K = \frac{P_{pred}}{R + P_{pred}}$$

$$\theta^{new} = \theta^{pred} + K \cdot (\theta_{acc} - \theta^{pred})$$

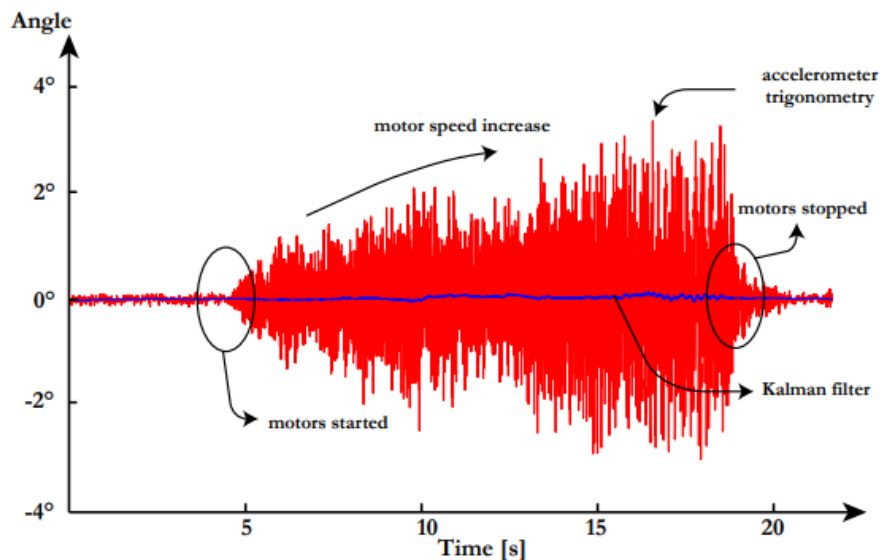
$$P_{new} = (1 - K) \cdot P_{pred} + Q$$

Q - שונות הרעש של תהליך

R - שונות הרעש של מדידה

P - מטריצת השונות (הערכת אי הוודאות)

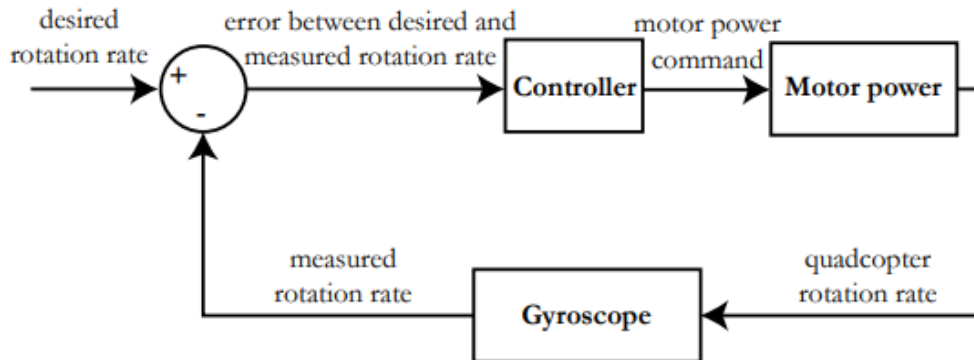
K - Kalman Gain מקדם תיקון - משקל יחסי בין הגיירו לאקסלרומטר



איור מס' 31 – הצגת פילטר באמצעות KALMAN

4. מודול בקרת יציבות (PID)

בקרת היציבות היא הליכה של מערכת הטיסה - תפקידה לשמור על זוויות ה-Roll/Pitch/Yaw הרצויות, להמיר פקודות טייס (setpoints) לאותות מנוע יציבים, ולחזור למצב יציב במהירות ובבטיחות לאחר הפרעות. במערכת זו מיושמת אסטרטגיית (Cascade Control) לולאת זווית חיצונית + לולאת מהירות פנימית, המשפרת יציבות ותגובתיות ביחס ליישום PID יחיד לכל ציר.



איור מס' 32 – חוג הבקרה במערכת

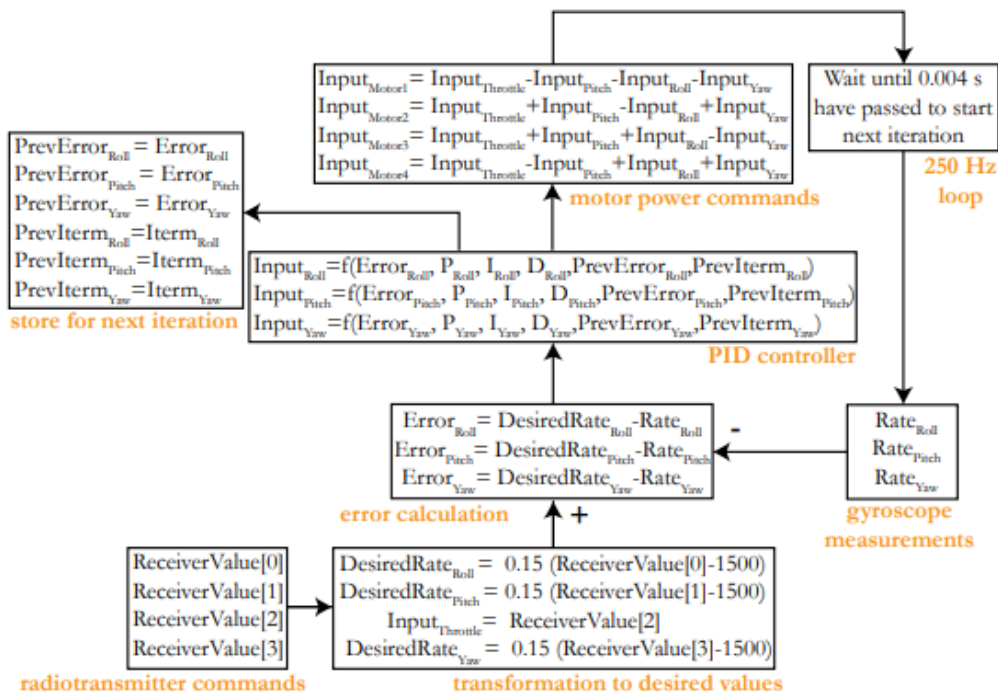
- לולאת זווית (Outer loop): מקבלת את ה-setpoint של זווית מייצרת פלט שהוא setpoint של מהירות סיבוב.
 - לולאת מהירות (Inner loop / Rate loop): מקבלת את ה-rate setpoint ומחסרת ממנו את מדידת הגיירו כדי לחשב את הפלט הרצוי למנועים. זו הלולאה המפקחת ישירות על פקודות ה-PWM / ESC.
- היתרון: בקרת זווית מדויקת ותגובה מהירה להפרעות - כשלולאת המהירות מטפלת ברעשים ובדינמיקה המהירה של המערכת.
- תיאור והגדרת כל אחד משלושת ערכי הבקרה בבקר:

$$P = K_p \cdot e(t)$$

$$I = I_{prev} + K_i \cdot e(t) \cdot dt = I_{prev} + K_i \cdot e(t) \cdot dt$$

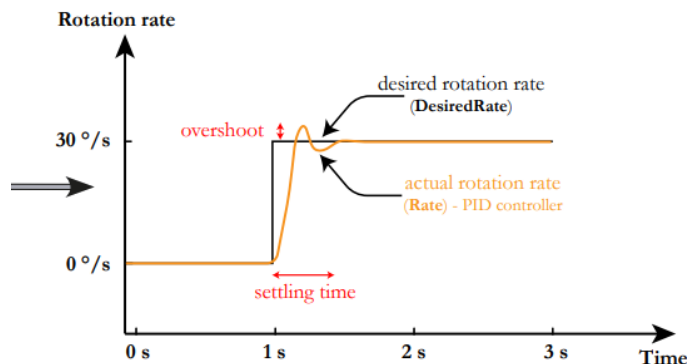
$$D = K_d \cdot dt \cdot d(measurement)$$

- Kp (Proportional)** - רכיב זה אחראי על עוצמת התגובה של המערכת לשגיאה הנוכחית.
- Ki (Integral)** - רכיב זה סוכם לאורך זמן את השגיאות הקטנות ומתקן סטייה קבועה.
- Kd (Differential)** - רכיב זה מתחשב בשינוי של השגיאה לאורך הזמן, ומבצע ניבוי קדימה כדי לבלום תנודות פתאומיות.
- בהתאם לחוג הבקרה במערכת ולכלל ההגדרות שהגדרנו בנינו את האלגוריתם המיוצג בעזרת חוג הבקרה. להלן חוג הבקרה המלא של בקר הטיסה :



איור מס' 33 – אלגוריתם הבקרה

כמו כן לאחר תכנון ומימוש האלגוריתם של בקר הPID הייעודי בוצעה התאמה עבור שלושת ערכי הבקרה של הבקר ששולטים על ביצועי המערכת כדי להגיע למצב אופטימלי בין תגובה מהירה ליציבות.



איור מס' 34 – מצב אופטימלי רצוי לתגובת PID

5. מודול התקשורת - WebSocket (קבלת ושליחת פקודות)

מודול זה אחראי על ניהול התקשורת הדו־כיוונית בין בקר הטיסה (ESP32) לבין תחנת הבקרה הקרקעית (GUI במחשב) ולבין שלט המשתמש הממומש ע"י טאבלט. התקשורת מתבצעת בפרוטוקול WebSocket מעל רשת Wi-Fi ומאפשרת חיבור רציף בזמן אמת להעברת נתונים ללא עיכובים מיותרים. מודול זה ממומש כתהליך נפרד (Task) הרץ על אחת מליבות הבקר תוך כדי התחשבות בזמן הריצה ומנוהל ע"י FreeRTOS. העברת הנתונים מתבצעת בפורמט JSON מה שמאפשר קריאה קלה בצד התוכנה במחשב, ניתוח בזמן אמת ותיעוד נתונים.

6. מודול ניטור סוללה תוכנתי (ADC)

מודול זה אחראי למדידת מתח הסוללה הראשית של הרחפן לצורך בקרה, ניטור והגנה. המדידה מתבצעת באמצעות ממיר ADC פנימי של ה־ESP32, אשר קורא את המתח האנלוגי דרך מחלק מתח כדי להתאים את הטווח למתח הקריאה המותר בבקר. כמו כן, הצד התוכנתי לוקח את הנתונים הגולמיים שמסופקים לבקר בעזרת התמיכה החומרתית ומסננת ומבצעת מספר פעולות על ערכי המתח הנמדדים בפין האנלוגי המומר לדיגיטלי. ראשית כל, לאחר הורדת המתח באופן חומרתי כל קריאה עוברת סינון למניעת קפיצות רגעיות. לאחר מכן מבוצע בתוכנה חישוב מחדש של ערך המתח בסוללה ע"י הכפלת מחלק המתח ההפוך והחזרת ערך הסוללה המקורי ובכך ערך הסוללה מוצג ומומר שוב ליחידות של וולט. ערך זה מחולק למספר מקטעים ועל ידי סיווג הערך המתקבל נשלחת למשתמש באמצעות הטלמטריה מצב הסוללה בממשק המשתמש. בנוסף, קיימת לוגיקה של התרעת מתח נמוך - כאשר מתח הסוללה יורד מתחת לסף שהוגדר $10.8V$ מופעלת הודעת אזהרה על המסך.



איור מס' 35 – תצוגת סוללה 4 מקטעים בתחנת הניטור 25 עד 100 אחוז

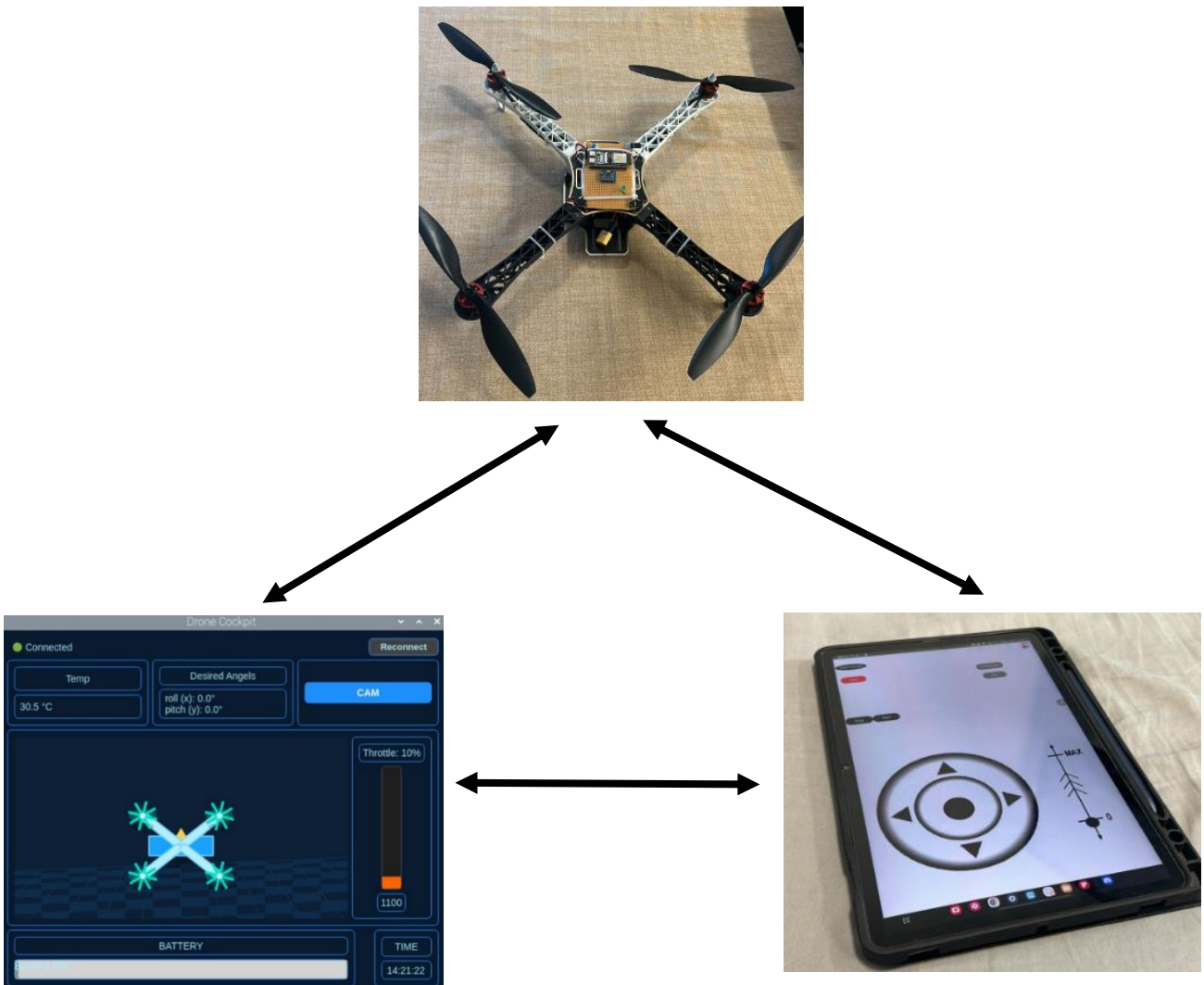
7. מנגנון Fail-Safe

מנגנון זה נועד להבטיח פעולה בטוחה של הרחפן במצבי תקלה, אובדן תקשורת או הפעלה לא נכונה של המערכת, ולמנוע סיכונים הן לרחפן והן למשתמש. המערכת כוללת מספר שכבות הגנה הפועלות במקביל ומכסות תרחישים שונים של כשל אפשרי. שכבות ההגנה העיקריות:

- מנגנון ניתוק חירום ראשי (Master Fail-Safe Switch): כפתור ייעודי בשלט הרחוק מאפשר כיבוי מיידי של כלל המערכת - כולל ניתוק אותות ה-PWM מהמנועים ונעילת הבקר עד לשחרור ידני של המשתמש. מנגנון זה מהווה "מעטפת הגנה" פיזית מפני הפעלה לא מבוקרת או טעות אנוש.
- מנגנון תוכנתי לזיהוי אובדן תקשורת: בבקר הטיסה מתבצע ניטור רציף של קצב קבלת הפקודות משלט המשתמש. אם לא מתקבלת כל פקודה במשך פרק זמן שהוגדר מראש, המערכת נכנסת אוטומטית למצב Fail-Safe תוכנתי הכולל שמירת יציבות ועצירת מנועים במידת הצורך.
- חיווי ויזואלי (LED Status Indication): מערכת לדים בבקר וגם על גבי שלט המשתמש מספקת למשתמש חיווי מיידי על מצב הקשר והמערכת. לד אדום - אין תקשורת. לד ירוק - תקשורת פעילה ותקינה. מנגנון זה מאפשר למשתמש לזהות בזמן אמת אם המערכת פעילה, תקינה או במצב חירום.
- הגנת מתח נמוך (Battery Fail-Safe): כאשר מתח הסוללה נמדד כמתחת לסף הבטיחות שהוגדר, מתבצעת התראה ויזואלית למשתמש.

2. מימוש המערכת

בשלב זה בוצעה אינטגרציה מלאה של כלל תתי המערכות ורכיבי המערכת לכדי יחידה אחת מתפקדת. הרכבת המערכת כללה חיבור פיזי של כל החומרה והבקר המרכזי על גבי מטריצת החיבורים לגוף המכני של הרחפן, חיבור המונעים וכן חיוטים וחיוזקים מתאימים. בנוסף בשלב זה בוצע חיבור תוכנתי וקישוריות תקשורת על בסיס Websocket באמצעות Wifi בשכבת TCP/IP בין שלט המשתמש על גבי הטאבלט ושל תחנת השו"ב על גבי ה-RPI5 לבין הרחפן.



איור מס' 36 – מימוש המערכת השלמה

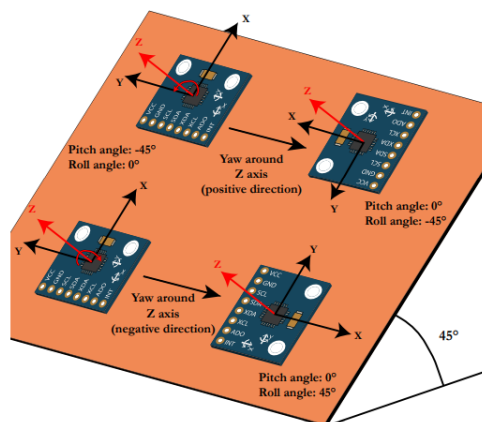
2.1. ניסויים ובדיקות

לאחר תכנון המערכת, בניית כל תתי-המערכות ואיחודם למערכת כללית ומתפקדת, בוצעו סדרת ניסויים ובדיקות שנועדו לוודא את תקינות הפעולה הכוללת של הרחפן, את יציבותו, את אמינות התקשורת, ואת בטיחות המשתמש.

הניסויים נועדו לאמת את תפקוד כל אחד מהרכיבים - החל מקריאת חיישנים, דרך אלגוריתם בקרת ה-PID ועד לתקשורת בזמן אמת בין הבקר למחשב הקרקע ולשלט הרחוק. במהלך הניסויים נמדדו פרמטרים פיזיקליים שונים (זוויות, מצערת, טמפרטורה, מתח סוללה) ונבחנה התנהגות הרחפן במצבי עומס שונים. בנוסף, נבדקו תרחישים של תקלות מבוקרות (כגון ניתוק תקשורת) לצורך בחינת מנגנוני ההגנה.

2.1.1. ניסוי כיוול ראשוני ותקינות החיישנים

בשלב הראשון בוצעה בדיקת תקינות לחיישן ה-MPU6050 וכן לאלגוריתם ולחיישנים שמומשו עליו. נבדקה תגובתו לתזוזות ידניות של הרחפן, ווידוא כי הערכים של Roll, Pitch ו-Yaw משתנים באופן רציף. כמו כן, נבדקו ערכי האפס לאחר כיוול (Offset) כדי לוודא שהחיישן מייצג מצב אופקי תקין.



איור מס' 37 – ניסוי קבלת ערכי מיקום מהחיישן

בניסוי התגלה כי קיימים ערכי היסט קבועים למיקום החיישן שעל הרחפן במרחב. לכן בוצעה תיקון בקוד עבור ערכים אלו וערך קבוע הוחסר או חובר לכל ציר מדידה. בנוסף בניסוי זה גילינו שעקב רעשים אלימי או הפרעות רגעיות התחלת החיישן לא תמיד נותנת ערכים מדויקים ואמינים באשר למיקומו ההתחלתי במרחב לכן בעת הפעלת ועליית המערכת המיקום הראשוני שלו מחושב על ידי מדידת 100 מדידות מיקום מהחיישן וחישוב ממוצע שלהם כדי להפחית ולצמצם שגיאות ולהגביר את דיוק ואמינות החיישן.

טבלה מס' 5 – ערכי כיוול ACC

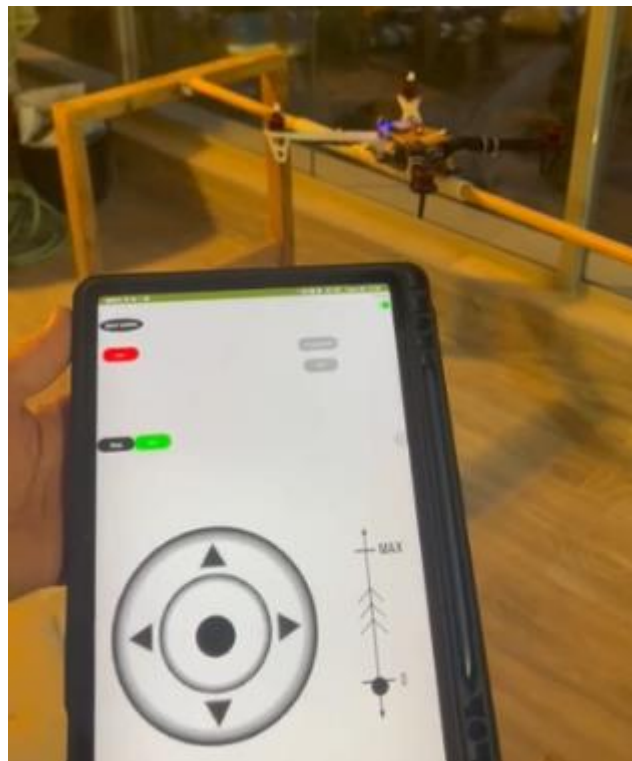
AccX	AccY	AccZ
-0.04	0	-0.05

2.1.2. ניסוי תקשורת - שידור וקליטה (WebSocket)

בשלב זה נבדקה תקשורת Wi-Fi שמומשה באמצעות פרוטוקול Websocket בין כלל תתי המערכת במערכת השלמה כאשר ה-ESP32 הינו מחולל הרשת "Server" ושלט המשתמש על גבי הטאבלט ותחנת הבקרה הקרקעית הינם "Clients".

התקשורת מתבצעת בשכבת TCP/IP בין המכשירים השונים ברשת על גבי PORT 80. המערכת נבדקה תחת מספר עומסי רשת, תוך כדי שליחת נתונים רציפה בקצב גבוה, ובדיקת שידור וקליטת חבילות נתונים ללא השמטות. נמדד זמן ההשהיה בין שליחת פקודה מהמחשב לבין תגובת הרחפן שהיה ברמת mSec הבודדות.

כמו כן, נבדקה תקשורת השלט (באמצעות טאבלט) מול בקר הרחפן דרך WebSocket. במהלך הבדיקה נשלחו פקודות שינוי מצערת, Roll ו Pitch ונמדד פרק הזמן עד שהרחפן הגיב לשינוי. הממוצע עמד על תגובה כמעט מיידית, מה שמעיד על סנכרון יעיל בין שלושת רכיבי התקשורת.



איור מס' 38 – ניסוי בדיקת תקשורת וזמני תגובה בין המערכות

2.1.3. ניסוי בקרת יציבות (PID)

בבדיקה זו נבדקה תגובת הרחפן לשינויים בזוויות ובתנודות חיצוניות, לצורך קביעת ערכי הבקרה האופטימליים (K_p, K_d, K_i).
לצורך ניסוי זה הוקמה מערכת SETUP ייעודית לרחפן אשר שמרה על בטחוננו בעת ביצוע הניסויים וקביעת ערכי הבקר.
כמו כן המערכת הורכבה על מנת שנוכל לקבוע את ערכי הבקרה ללא כל הפרעות וללא גרימת נזק לרחפן ולמערכת בעת התרסקות עקב ערכי בקרה שגויים.

להלן קישור לסרטונים המתעדים את הגדרת פרמטרי הכיול של בקר המערכת:

https://github.com/yonizvida-gif/Drone-Project-Michael-Yoni/tree/main/ESP32_FlightController/PID_Calibration



איור מס' 39 – ניסוי בדיקת יציבות וכיול ערכי PID

לאחר בניית מערכת הבדיקה והרכבת הרחפן על גבי המערכת בוצעו מספר הרצות בהן שונה כל פרמטר בנפרד, ונמדדה התגובה של הרחפן בזמן אמת באמצעות ראייה ויזואלית של התייצבות הרחפן על גבי המערכת וכן הוצאת פלט לגרפים בזמן אמת המדגים את עוצמת המנועים.

המטרה הייתה להגיע לערכים שמאפשרים תגובה מהירה אך ללא תנודות יתר (Overshoot).
במהלך הניסוי:

- K_p הוגדל בהדרגה עד להופעת תנודות קלות.
- K_i הותאם לשמירה על שגיאה אפסית לאורך זמן.
- K_d שימש להחלקת המעברים ולהפחתת רגישות לרעשים.

טבלה מס' 6 – ערכי בקר PID

K_p (Rate,Pitch)	K_i (Rate,Pitch)	K_d (Rate,Pitch)	K_p (Yaw)	K_i (Yaw)	K_d (Yaw)
1.1	0.65	0.003	2	12	0

2.1.4. ניסוי ובדיקת מנגנוני בטיחות (Fail-Safe)

ניסוי זה נועד לבדוק את תגובת המערכת בתרחישים של תקלה או איבוד תקשורת:

- לכפות ניתוק המערכת על ידי מתג SAFETY ראשי – הגנה חיצונית
 - ניתוק תקשורת Wi-Fi פתאומי
 - חריגה במתח סוללה מתחת לסף – התרעת חיווי וניתוק מנועים
- המערכת עמדה בכל תרחישי הבדיקה בהצלחה והפסיקה את פעולת המנועים באופן מבוקר ובטוח ונכנסה למצב הגנה או נחיתה במידת הצורך.

2.1.5. ניסוי וכיול כיוון סיבוב להבי המנוע

לאחר חיבור ארבעת המנועים וה- ESCs לבקר הטיסה, נדרש לוודא שכל מנוע מסתובב בכיוון הנכון ובהתאם למבנה הרחפן שבתצורת X - כדי לגרום לעילוי תקין ויציבות. הכיוון הנכון של כל מנוע הוא גורם קריטי ליציבות הרחפן - סיבוב שגוי של אחד מהם יגרום לחוסר איזון ולחוסר שליטה בטיסה.

מטרת הניסוי הינה לאמת שכל אחד מהמנועים מסתובב בכיוון הנדרש על פי תצורת הרחפן, ולבצע תיקון במידת הצורך ע"י שינוי חיבורי פאזה או עדכון תוכנה.

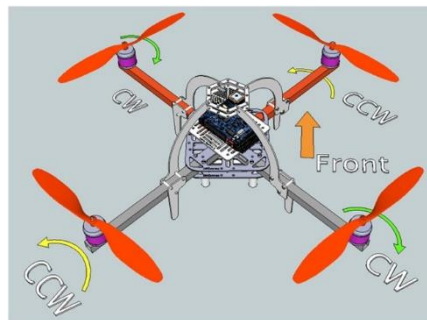
בבדיקה הראשונית, כל מנוע הופעל בנפרד בקצב קבוע בעזרת רכיב SERVO TESTER המספק אות PWM מתאים לרכיב ה- ESC תוך צפייה חזותית על כיוון הסיבוב של המנוע.

בנוסף בהתאם לסיבוב כיוון המנוע הנכון של כל ציר הורכב על אותו המנוע הלהב המתאים עם כיוון השעון CW או נגדו CCW.

במקרה שאחד המנועים הסתובב בכיוון הפוך, בוצע היפוך בין שני חוטי פאזה מתוך שלושת חוטי ה- ESC מה שגרם להפיכת כיוון הסיבוב.

לאחר התיקון בוצעה הפעלה מחדש ואימות חוזר.

ניסוי זה הוכיח כי כיוון הסיבוב של המנועים הותאם במדויק לתצורת הרחפן, וכי מערכת הבקרה מסוגלת להפעילם באופן מתואם, השלמת שלב זה מהווה תנאי חיוני לפני ביצוע ניסויי טיסה מלאים.



איור מס' 40 – תיאור כיוון סיבוב להבי הרחפן CW/CCW

2.1.6. ניסוי טיסה כולל

בשלב הסופי בוצעה טיסת ניסוי מלאה בהפעלת כל המודולים ברזמנית, הניסוי כלל את כלל המרכיבים הבאים :

- קישוריות תקשורת אלחוטית מלאה בין הרחפן לתחנת השו"ב (Raspberry Pi 5)
- תקשורת דו־כיוונית בין בקר הטיסה ESP32 לשלט המשתמש (טאבלט).
- שליחת פקודות טיסה בזמן אמת (Throttle, Pitch, Roll, Yaw)
- איסוף טלמטריה רציפה (טמפרטורה, מתח סוללה, מדדי זווית ותאוצה).
- פעולת בקר ה־PID ואלגוריתם הבקרה בזמן אמת לשמירה על יציבות הרחפן.

במהלך הניסוי נבדקה יציבות הרחפן במצב ריחוף ובמעברים, והוא שמר על גובה וזווית יציבה לאורך זמן.



איור מס' 41 – טיסת מבחן לרחפן

להלן קישור המכיל סרטונים המתעדים את טיסת הניסוי :

https://github.com/yonizvida-gif/Drone-Project-Michael-Yoni/tree/main/ESP32_FlightController/Test_Flights

לאחר איסוף הנתונים מהניסויים, בוצע ניתוח גרפי של תגובת הרחפן באמצעות תחנת השו"ב על גבי ה-RPI5 ובאמצעות חיזוי ויזואלי בהתנהגות הרחפן בשטח. המערכת הציגה עמידה מדויקת ביעדי היציבות, השהיה נמוכה בתקשורת, וזמני תגובה קצרים בין שליחת פקודות לבין ביצוע בפועל. התוצאות הראו כי הרחפן מצליח לשמור על יציבות גם תחת שיבושי רוח קלים, וה־PID מגיב באופן חלק ללא תנודות יתר.

2.2. מכשירי המדידה

במהלך הניסויים נעשה שימוש במגוון מכשירי מדידה, בקרה ותיעוד לצורך כיוול, ניתוח ביצועים ואימות תקינות רכיבי הרחפן. המכשירים שימשו הן במהלך הניסויים המעבדתיים והן בניסויי הטיסה בפועל:

- **מחשב נייד עם סביבת פיתוח (IDE / Serial Monitor / GUI) (ייעודי):**

שימש לניטור ערכי הטלמטריה בזמן אמת, קבלת נתוני חיישנים, הצגת ערכי PID וביצוע כיוולים ישירות מבקר הטיסה (ESP32). בנוסף בוצע תיעוד גרפי של פרמטרים כגון Pitch, Roll, Throttle וערכי המנועים.

- **מצלמת תיעוד:**

שימשה לתיעוד ניסויי השטח והטיסות מזוויות שונות, לצורך ניתוח ויזואלי של תגובת הרחפן בזמן אמת והשוואה מול הנתונים הנמדדים.

- **מולטימטר דיגיטלי מדויק:**

נועד למדידת זרמים ומתחי מערכת במהלך הטעינה, הפריקה וההפעלה של הסוללות והמנועים. באמצעותו נבדקו גם תקינות החיוטים על גבי גוף הרחפן וכן את כל חיוטי הבקר על מטריצת החיבורים.

- **אוסצילוסקופ דרערוצי:**

שימש למדידת תדר ועוצמת אותות ה-PWM היוצאים מהבקר אל בקרי ה-ESC וכן לבדיקת יציבות האותות וחוסר רעידות.

- **Servo Tester:**

מכשיר זה שימש לכיוול כיוון סיבוב המנועים ולהתאמת ערכי ה-Throttle לכל ESC לפני חיבור לבקר הראשי, על מנת להבטיח סנכרון מלא בין כל ארבעת המנועים ולמנוע תופעות של סבסוב לא רצוני.

- **מערכת כיוול לבקר (PID Tuning Station):**

תוכנה ייעודית שנכתבה כחלק ממערכת ה-GUI אפשרה שינוי בזמן אמת של מקדמי הבקר (K_p , K_i , K_d) ובדיקת השפעתם על יציבות הטיסה. בנוסף הורכבה עמדת בדיקה ייעודית לחיבור הרחפן ובדיקות ערכי בקר PID אלו. הנתונים נותחו בזמן אמת והוצגו בגרפים לצורך קביעת ערכי Tuning מיטביים.

3. תוצאות ומסקנות

פרק זה מציג את ממצאי הניסויים שנערכו במהלך מימוש המערכת ובדיקת תפקודה הכולל של הרחפן, שלט המשתמש ותחנת השו"ב. מטרת הניסויים הייתה לוודא את עמידת המערכת בדרישות התכן - דיוק, יציבות, תגובתיות, בטיחות, אמינות ויכולת שליטה בזמן אמת - כפי שהוגדרו בשלבי הפיתוח.

3.1. תוצאות הניסויים והמדידות

מערכת הבדיקות והניסוי הופעלה בסדרת בדיקות בתנאים שונים, שכללו שינויי מתח, זווית, משקל ופרמטרים של בקר ה-PID. בתחילה נבדקה תקינות החיישנים, חיבורי התקשורת והפעלת המנועים, ובהמשך נבחנו תפקודי בקרת היציבות, התקשורת בזמן אמת, ומנגנוני ההגנה – כפי שפורט בחלק 2.1.

תוצאות עיקריות:

1. ניסוי בקרת יציבות (PID):

נמצא כי הרחפן מצליח להגיע למצב יציב בתוך פרק זמן קצר לאחר שינוי זווית פתאומי. כמו כן, בקר היציבות שלו מצליח להתמודד עם שינוי סביבה ורוח ולהישאר במצב טיסה יציב. ערכי הבקרים שנבחרו לאחר אופטימיזציה (K_p , K_i , K_d) הציגו איזון טוב בין תגובתיות לרעש מדידה.

2. ניסוי תקשורת דרכיונית (WebSocket):

המערכת הציגה השהיה ממוצעת נמוכה בין שליחת פקודה מהשלט לבין תגובת הרחפן (ברמת RealTime), דבר המאפשר שליטה בזמן אמת גם בטיסות מדויקות. נמדדה אמינות תקשורת של מעל 90% לאורך 10 דקות של שידור רציף.

3. ניסוי מנגנוני בטיחות (Fail-Safe):

בעת ניתוק מכוון של תקשורת ה-Wifi הרחפן עצר את פעולת המנועים תוך פחות מ-1 שניות. נוריות החיווי הציגו התראה מיידית, מנגנון זה מנע טעויות הפעלה ושמר על בטיחות המפעיל. כמו כן מתג עצירה מאולץ עצר מיידית את פעולת המנועים להבטחת פעולת המפעיל ומניעת נזק למערכת.

4. ניסוי ניטור מתח הסוללה:

ה-ADC סיפק קריאה מדויקת עם טעות ממוצעת של 0.1V בלבד לעומת מדידה במולטימטר. המערכת זיהתה ירידת מתח בזמן אמת והתריעה לפני ירידה מתחת לסף המוגדר.

3.2. מסקנות הפרויקט

מניתוח התוצאות ניתן להסיק כי :

- המערכת השיגה יציבות ותגובה בזמן אמת, בהתאם לדרישות התכן שהוגדרו בשלב התכנון.
- בקר ה-PID שנבנה וכויל סיפק שליטה עדינה ויעילה גם בשינויים פתאומיים.
- פרוטוקול התקשורת WebSocket הוכיח יעילות גבוהה עם השהיה מינימלית ואמינות גבוהה.
- מנגנוני ה-Fail-Safe פעלו כנדרש ומנעו תאונות אפשריות בזמן ניסויים.
- שילוב המודולים (ESP32, RPi5, טאבלט, חיישני IMU, מערכת PID, ממשק המשתמש ו-FreeRTOS) פעל בהרמוניה מלאה, תוך ניהול משימות מדויק ויעיל.
- המערכת הציגה חסינות גבוהה לרעשים אלקטרומגנטיים ולחוסר יציבות באספקת המתח, בזכות סינון חומרתי ותוכנתי (LPF + Kalman).
- תצוגת הנתונים בממשק המשתמש (GUI) הוכיחה כי ניתן להשיג ניטור מלא בזמן אמת של כלל פרמטרי הרחפן בצורה ויזואלית ונוחה למשתמש.

יחד עם זאת, נמצאו מספר נקודות לשיפור עתידי :

- שדרוג בקר הטיסה בעזרת חיישן ברומטר BMP לשיפור יציבות הרחפן בגובה קבוע והוספת ממד גובה לאלגוריתם הטיסה בפילטר קלמן (Altitude Hold).
- שילוב GPS במערכת הרחפן לצורך ניווט עצמאי, שמירת מסלול טיסה ויכולת Return to Home
- החלפת התקשורת בין שלט המשתמש לרחפן ממסדר Wi-Fi למודול RF ייעודי (מסדר-מקלט) להגדלת טווח ואמינות השידור.

3.3. סיכום

הרחפן שפיתחנו במסגרת פרויקט זה הדגים בהצלחה מערכת אינטגרטיבית מתקדמת המשלבת בין תחומי אלקטרוניקה, בקרה, תקשורת ותוכנה. המערכת עמדה ביעדים המרכזיים שהוגדרו בתחילת הדרך :

- שליטה בזמן אמת עם תגובתיות גבוהה.
- יציבות טיסה ומנגנוני הגנה יעילים.
- תקשורת דו-כיוונית רציפה בין כלל הרכיבים.
- ממשק משתמש אינטואיטיבי לניטור ובקרה.

הפרויקט מהווה הוכחה ליכולת לממש מערכת אווירית חכמה המבוססת על חומרה זמינה, תוך תכנון ובקרה הנדסית מלאה, ופותר פתח להמשך מחקר ופיתוח בתחום הרחפנים החכמים והרובוטיקה האווירית.

4. מקורות ספרותיים

במהלך ביצוע הפרויקט נעשה שימוש במגוון מקורות ידע, ספרותיים ודיגיטליים, לצורך לימוד, מחקר והעמקת ההבנה של עקרונות הבקרה, התקשורת והחומרה בתת מערכת הרחפן ובמערכת הכללית. המידע שנאסף שימש לבניית התכנן, כתיבת התוכנה ולביצוע הניסויים השונים.

4.1 מקורות מקוונים ולימודיים

- סרטוני הדרכה מתוך פלטפורמת YouTube אשר סקרו עקרונות בקרה מבוססי PID, כיול חיישני IMU (MPU6050) ויישומי טיסת רחפן מבוססי ESP32.
- מדריכים בנושא פרוטוקול WebSocket, תקשורת Wi-Fi בזמן אמת.
- מאמרים מקוונים בנושא אלגוריתם קלמן (Kalman Filter) לפילטר רעשים ויישומם במערכות מבוססות חיישני תנועה.
- מקורות לימוד בתחום בקרת יציבות ובקרה בזמן אמת ממקורות חינוכיים טכנולוגיים ואתרי קוד.

4.2 דפי נתונים טכניים (Datasheets)

- MPU6050 - MEMS Accelerometer & Gyroscope
- ESP32 - Dual-Core Wi-Fi & Bluetooth MCU
- ESC - PWM Motor Driver
- Li-Po Battery Specifications נתוני מתח, זרם וטעינה בטוחה.
- Servo Tester Calibration Device הוראות כיול והפעלה.

4.3 מקורות עיוניים ותיאורטיים

- Wikipedia - לעיון והבנת עקרונות כלליים של בקר PID, אלגוריתם קלמן, רעשי מדידה, פרוטוקול I2C ופרוטוקול תקשורת WebSocket.
- מקורות תיאורטיים מתוך חומרי קורסים בתחום מערכות בקרה, מערכות משובצות (Embedded Systems), קורסי תכנות, מעבדת CISCO ומעבדות האלקטרוניקה השונות.
- מאמרים טכניים בנושאי Real-Time Control Systems.

4.4 מקורות תוכנה וקוד

- דוגמאות קוד מאתרי קוד פתוח כגון GitHub ו-ESP32 Forums ששימשו להבנת מבנה הקוד והטמעת משימות FreeRTOS.
- קטעי קוד דוגמה מתוך פרויקטים קיימים ששימשו להשראה בתכנון הארכיטקטורה ובמימוש מערכת הניטור והבקרה.