

שם הפרויקט: נווט אוטונומי

מגישים: נאור ידעי (302568894)

יוסף זוהר (021512884)

מנחה הפרויקט: ד"ר בועז בן משה

תיאור הפרויקט

כיום ישנה מגמה להעדיף כלי טיס בלתי מאויישים (מזל"ט) על פני מטוסים מאויישים. למזל"ט יתרונות רבים:

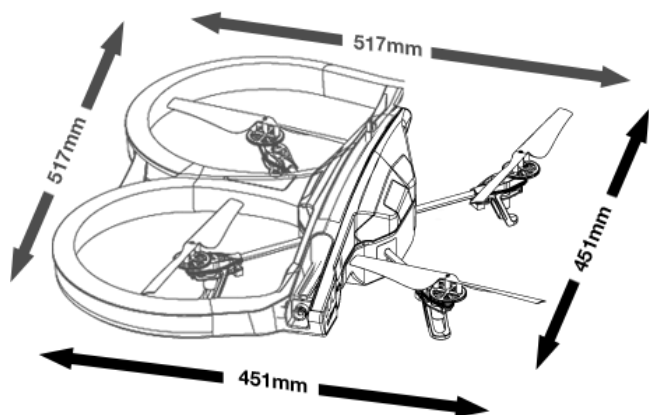
- בטיחות - אין סיכון לחיי אדם.
- עלות - עלות כלי הטיס הבלתי מאוייש נמוכה בהשוואה למטוס רגיל.
- ביצועים - מימדיו הקטנים של המזל"ט מאפשרים לו לבצע משימות מגוונות ואיכותיות יותר מכלי טיס רגיל.

אולם המזל"ט מופעל ע"י אדם. ולכן, למרות היתרונות שנמנו לעיל, קיימים בשיטת המפעיל האנושי כמה חסרונות:

- תקציבים גבוהים - מעבר להשקעה במזל"ט עצמו, נדרשת השקעה של תקציבים גבוהים כדי להקים סביבה תומכת, כגון: סימולטורים, צוות טכני רחב מאוד.
- הכשרה - יש צורך להכשיר צוות שיטיס את המזל"ט וישמור על כשירות.
- טעויות - מזל"ט המופעל ע"י אדם עלול לטעות בעת ביצוע המשימה בהתאם לשיקול דעת מוטעה של מפעילו. לעומת זאת מזל"ט המופעל על פי אלגוריתם מדויק יותר, והסיכוי לטעות נמוך יותר.

מטרת עבודה זו, הנווט האוטונומי, היא לשדרג את מערכת כלי הטיס הבלתי מאוייש הקיימת כיום. היעדים הם:

- המזל"ט יפעל באופן אוטונומי ללא התערבות אנושית.
- הנווט האוטונומי יסייע לכלי הטיס לבצע את משימתו תוך הימנעות מהיתקלות במכשולי דרך שונים.
- המערכת המתוכננת בפרויקט זה מותאמת למזל"ט הפשוט ביותר, שהוא קל וקטן בהרבה ממזל"ט רגיל (ראה בתרשים המצורף). לכן המזל"ט יהיה מסוגל לבצע באופן מדויק ומושלם "משימות איכות", שאינן ניתנות לביצוע ע"י כלי גדול יותר.



תיאור המערכת

המערכת המתוכננת כוללת ארבעה מרכיבים עיקריים:

א. AR.Drone 2 - מזל"ט

נתונים כלליים:

- מהירות מרבית: 11.11 m/s
- גובה מקסימאלי: 199.03m
- מידות: 451x451x62mm
- משקל: 380g

Indoor Hull :
14.82oz / 420g

Outdoor Hull :
13.40oz / 380g



ב. חיישני מרחק - מיועד לזיהוי עצמים במהלך הטיסה.

אנו נשתמש בחיישן IR מסוג GP2Y0A60SZ0F של SHARP. החיישן מזהה עצמים ממרחק 10 ס"מ עד למרחק של 150 ס"מ. קומפקטי - $7.2 \times 8.0 \times 22.0$ מ"מ.

קישור לגיליון הנתונים - http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a60szxf_e.pdf



ג. **רכיב GPS** - מיועד לומר למטס את מיקומו הנוכחי ולאן הוא צריך להגיע.

אנו נשתמש ברכיב u-blox MAX-7.

רכיב קומפקטי ומדויק מאוד.

קישור - <http://www.u-blox.com/en/gps-modules/pvt-modules/max-7.html>

ד. **בקר טיסה** - מקבל את הנתונים מהחיישנים ומה-GPS ונותן הוראות טיסה למזל"ט.

אנו משתמש בבקר מסוג Teensy 3.1. רכיב שמימדיו קטנים וכוח עיבוד הנתונים

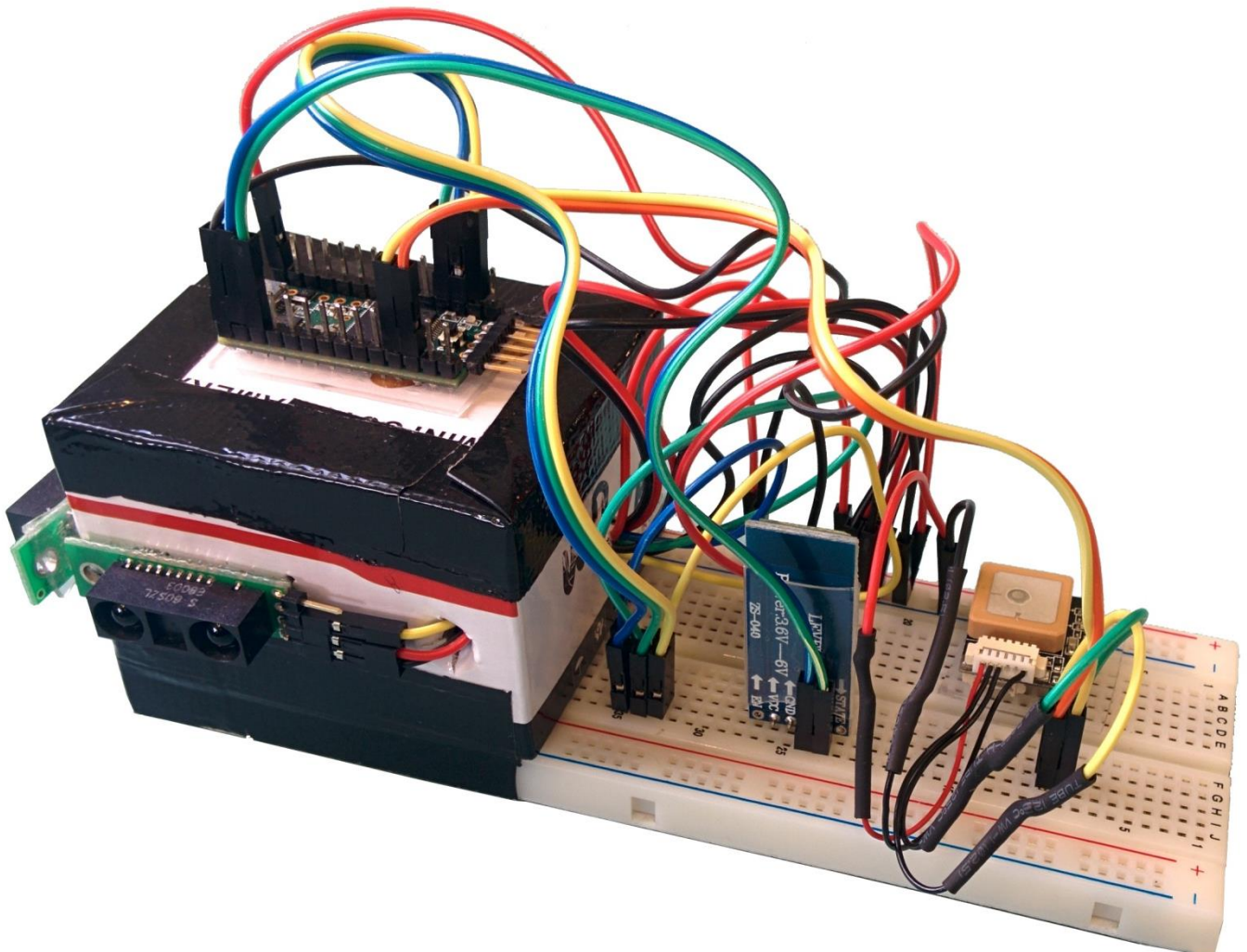
שלו חזק יחסית.

קישור - <https://www.pjrc.com/teensy/teensy31.html>

ה. העברת נתונים מהבקר טיסה למזל"ט.

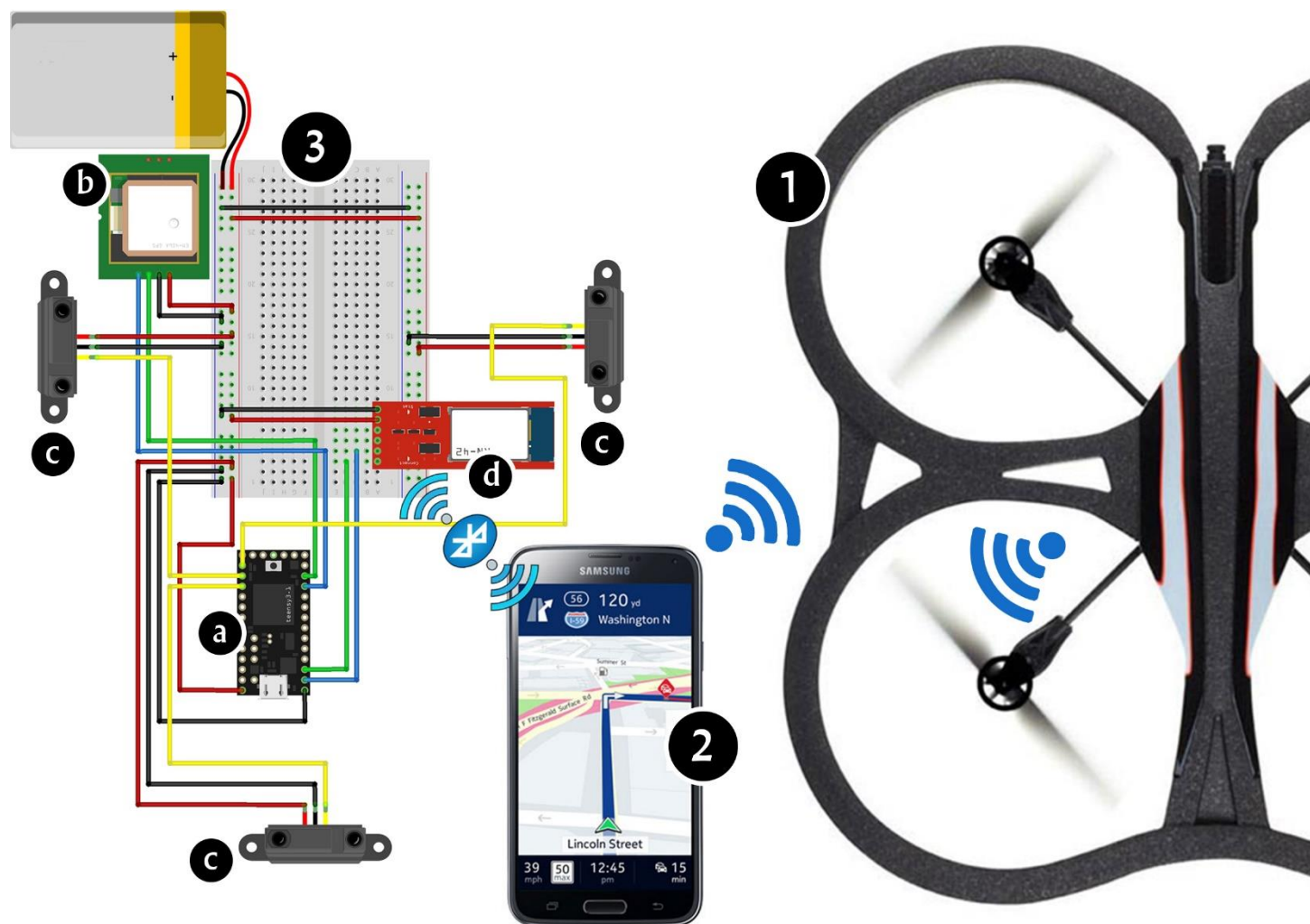
נשתמש ברכיב Bluetooth.

מערכת משולבת שהרכבנו במהלך העבודה :



רכיבי הפרויקט

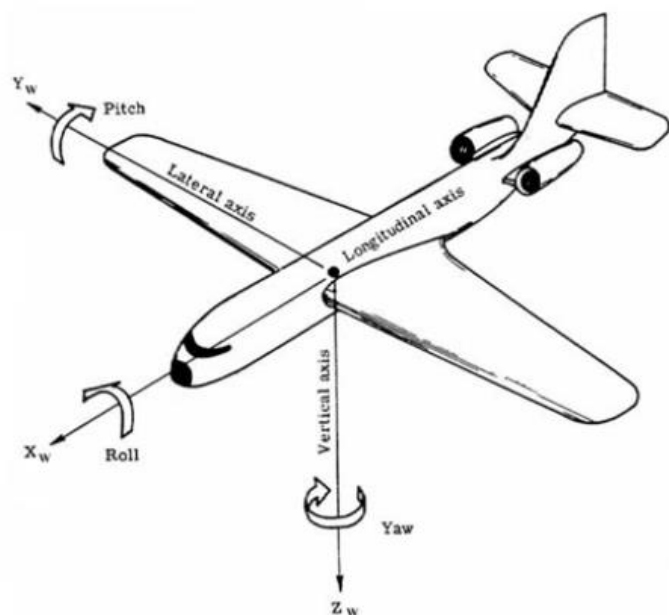
סכמת המערכת :



אפיון המערכת - המערכת תפעל באופן הבא :

- א. הכרטיס החכם (a) יאסוף את הנתונים הבאים :
 - a. מדי מרחק - הסנסורים (c) יספקו מידע - האם קיימים עצמים במרחב של המזל"ט.
 - b. מיקום - ה-GPS (b) יספק את קו האורך וקו הגובה.
- ב. הנתונים מסעף א' יועברו ב-stream בעזרת רכיב ה-Bluetooth (d) אל רכיב ה-Bluetooth המובנה בטלפון החכם (2).

ג. נתוני NavData ישלחו מהמזל"ט (1) אל הטלפון החכם, הנתונים יכללו:



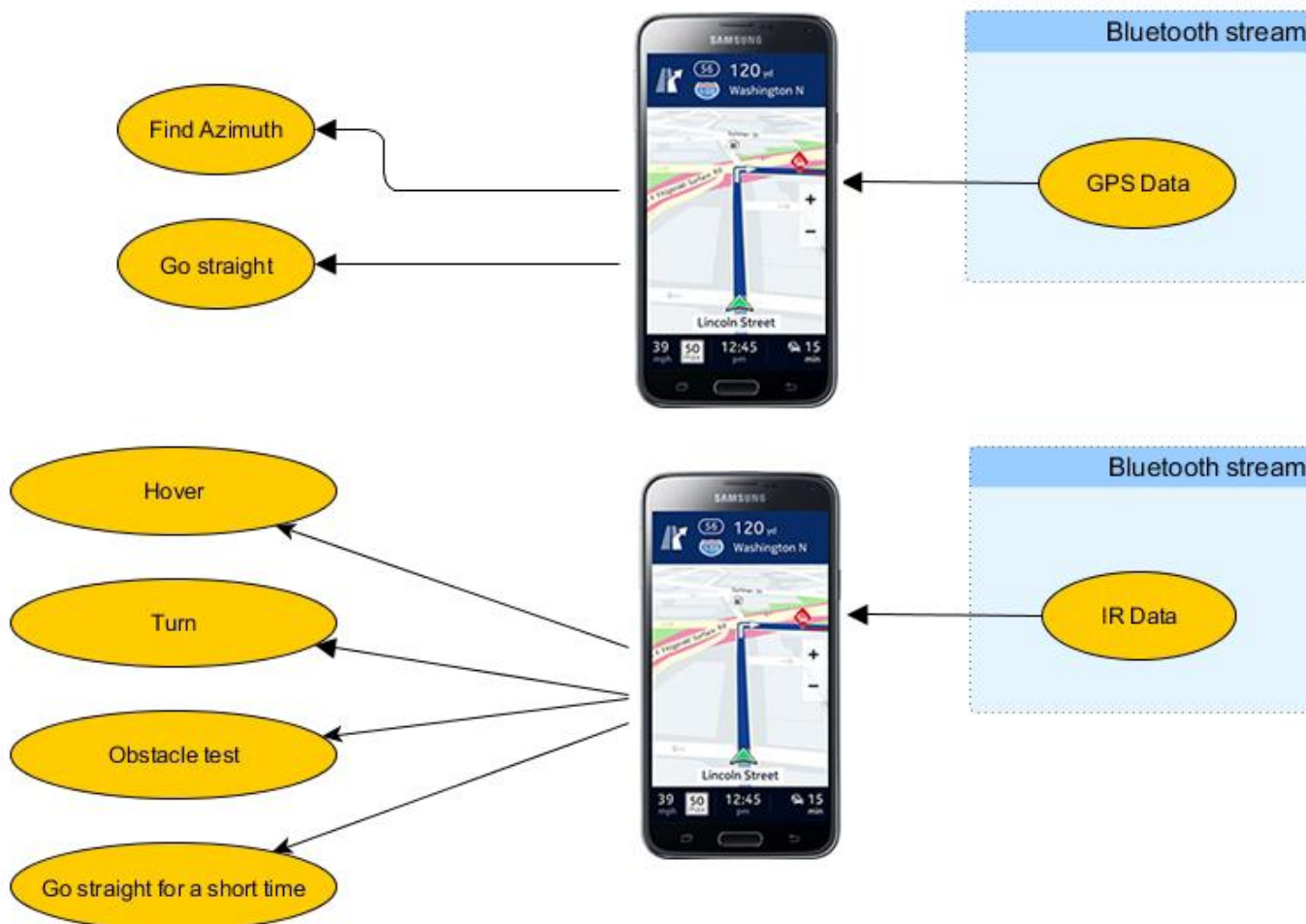
- a. גובה מעל פני הקרקע
- b. אוריינטציה:
 - i. yaw
 - ii. roll
 - iii. pitch
- c. מהירות טיסה
- d. מצב סוללה

ד. הנתונים מהסעיפים הנ"ל (ג'-ד') יעברו עיבוד בטלפון החכם.

ה. על סמך עיבוד הנתונים, הטלפון החכם ישלח פקודות אל המזל"ט (1) בעזרת רכיב ה-WiFi שמובנה בטלפון החכם.

ו. פקודות חירום יעשו בעזרת שימוש בטכנולוגיית ה-3G המובנה בטלפון החכם.

: Use Case

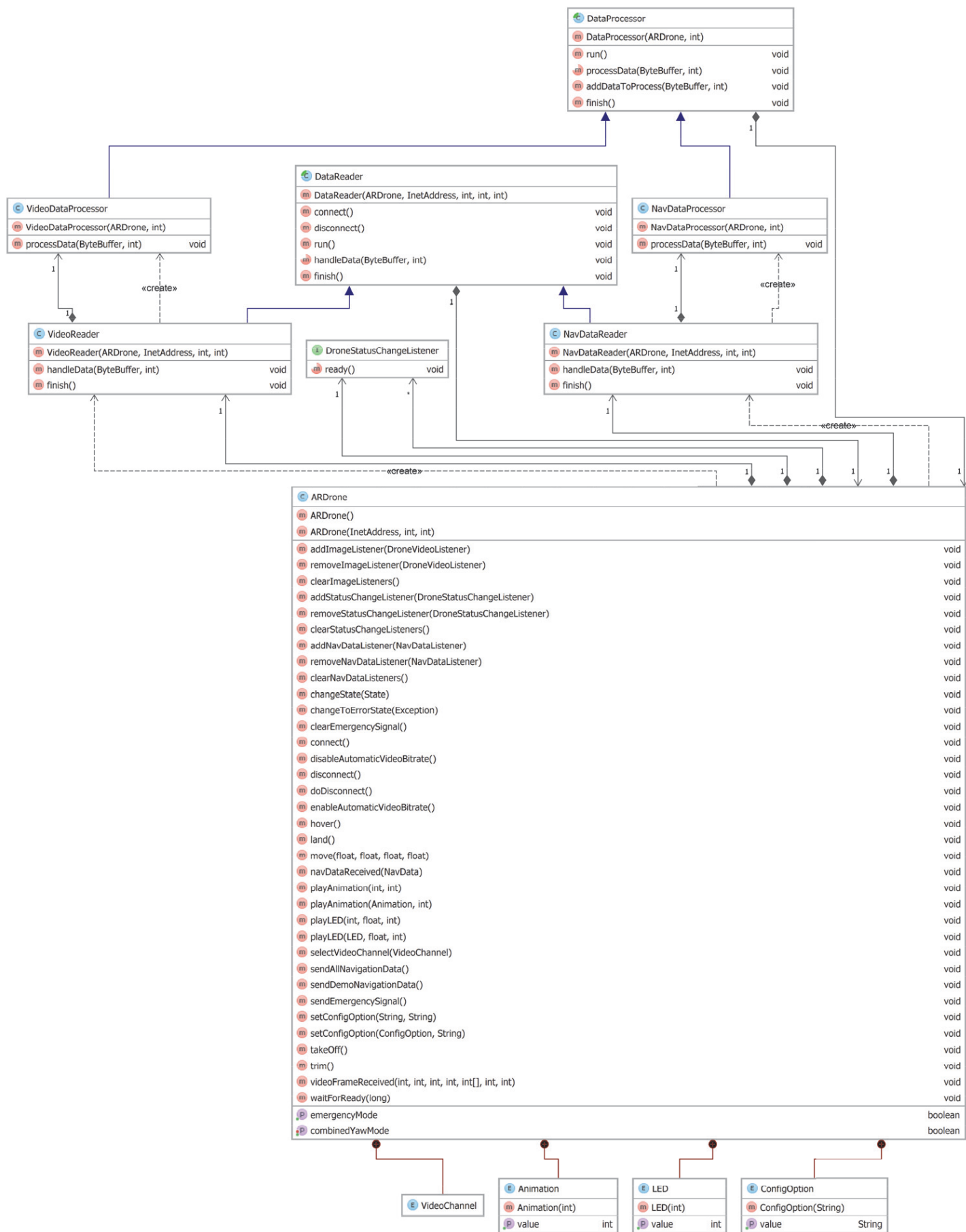


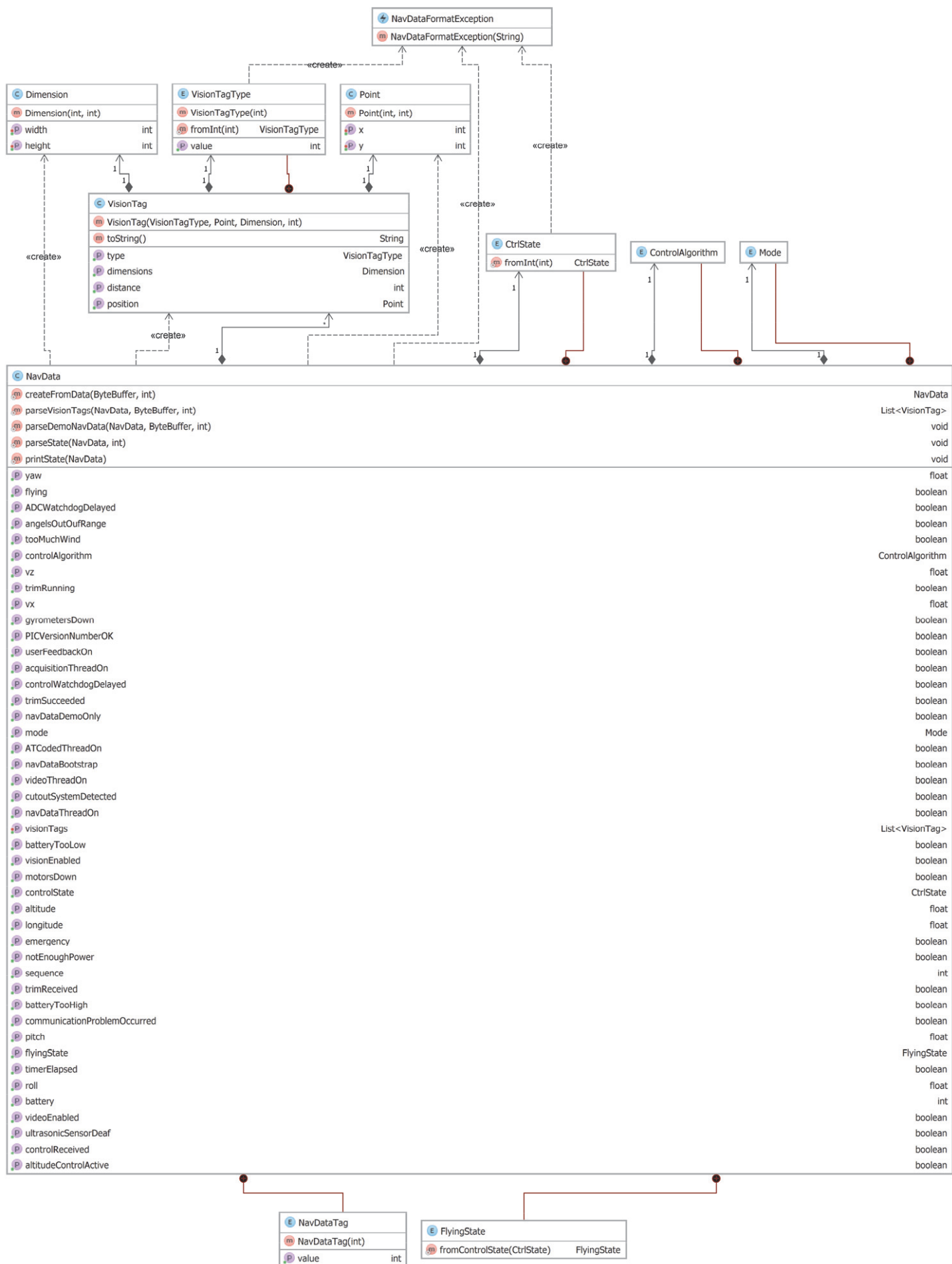


: UML - Class diagram

[חשוב לשים לב : מפאת חוסר מקום - חילקנו את המחלקות העומדות ביחס עם המחלקה "ARDrone" ל-2 דיאגרמות]







פלטפורמת הפיתוח

האתגר הטכנולוגי ובחירת החומרה

בחירת מזל"ט המתאים לצרכים שלנו. אנו חיפשנו מזל"ט שיספק לנו כמה שיותר פונקציונאליות. שיהיה קל משקל וקטן. בחרנו ב-AR.Drone 2 מבית Parrot. המטיס מתחבר למזל"ט באמצעות WiFi, כך שניתן להתחבר אליו ביתר קלות. בנוסף, המזל"ט שולח נתונים בזמן אמת אודות הטיסה (גובה, אוריינטציה, מהירות טיסה וכו'). למזל"ט מחוברים 2 מצלמות, קדמית ותחתונה. בעזרת קבלת נתוני הטיסה והמצלמות אנו יכולים לנתח בזמן אמת את התנהגות המטוס בזמן ניסויים. למזל"ט האפשרות להינעל (Optic Flow) על אובייקט מסוים בקרקע, כך הוא יכול להישאר יציב במהלך הטיסה. בנוסף, AR.Drone 2 ניתן לתכנות וקיים תיעוד מלא ומפורט. יש הרבה ספריות לשליטה במזל"ט ברחבי הרשת.

בחירת בקר טיסה. בקר הטיסה הינו המח המקבל את כל הנתונים מהמזל"ט, מהסנסורים ומה-GPS ומבצע את כל החישובים ע"מ לתת למזל"ט פקודות בכל זמן נתון. יש הרבה בקרים בשוק, ואנו היינו צריכים לבחור בקר שיענה על כל הדרישות, ביניהם, קל לתפעול, דיוק וכוח עיבוד.



PCduino 2

התחלנו להתנסות ב-PCduino 2, כרטיס המכיל בתוכו mini PC וגם Arduino. הוא יכול להריץ מערכת Android ואת מערכת ההפעלה Ubuntu. אך התכוונת החזקות שבו לא התאימו לפרויקט.

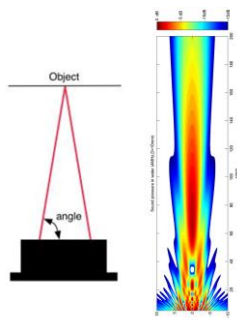


teensy

המשכנו עם Arduino uno. בקר זה ענה לדרישותינו אך רצינו להחליפו עם בקר יותר קטן שלא נופל ממנו מבחינת כוח עיבוד. לכן, בחרנו לעבוד עם teensy. בקר קטן מאוד, שכוח העיבוד שלו עולה על כרטיסי ה-Arduino.

בחירת חיישני מרחק. יש הרבה סוגים של חיישני מרחק, אנו התנסינו בשני סוגים:

1. IR Sensor - חיישן שולח קרן אור של אינפרא אדום לאובייקט והקרן מוחזרת לחיישן. החיישן מודד את זווית החזרה של קרן האור ומחשב את המרחק לאובייקט.
2. Ultrasonic Sensor - חיישני Ultrasonic עובדים על עיקרון דומה לרדאר או סונאר, המעריכים מרחק לאובייקט על ידי פירוש ההדים מהרדיו או גלי קול בהתאמה. חיישנים קוליים פעילים ליצור גלי קול בתדר גבוה ומעריכים את ההד שהתקבל בחזרה על ידי החיישן. החיישן מודד את מרווח הזמן בין שליחת האות וקבלת ההד כדי לקבוע את המרחק לאובייקט.



IR vs Ultrasonic

בחרנו את חיישני ה-IR (ראה תיאור המערכת) מכיוון שמצאנו דגם ספציפי שמאוד קטן, מדויק יחסית ומשקלו אפסי.

בחירת רכיב GPS. רכיב ה-GPS מהווה חלק מרכזי במערכת הניווט האוטונומי. כיום יש הרבה רכיבי GPS קטנים ומדויקים. אנו בחרנו את רכיב ה-u-blox MAX-7 GPS. הוא מאוד מדויק ובמודול הקיים יש אפשרות לקבל את המיקום ביחס לצפון באופן פשוט יחסית.

שפת פיתוח וסביבת עבודה

נושא זה מתחלק ל-2 חלקים:

א. פיתוח הקוד ל: Teensy 3.1.

הפיתוח נעשה בסביבת עבודה של Arduino עם תוסף שיש להתקין בשביל לעבוד עם הכרטיס החכם. שפת תכנות: C.

ב. פיתוח קוד לטלפון החכם מבוסס Android.

הפיתוח נעשה ב: Eclipse ADT.

שפת תכנות: Java.

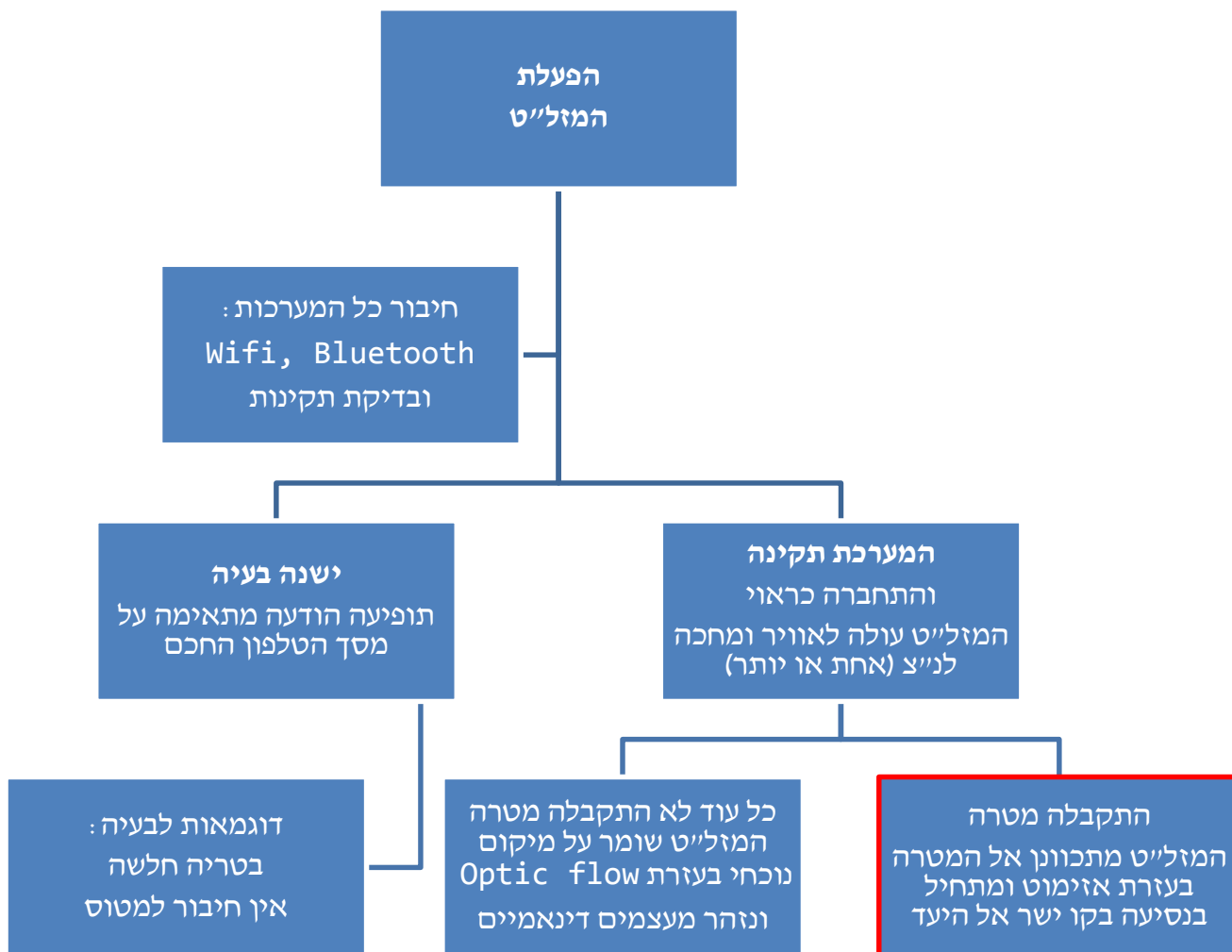
אלגוריתמים

נחלק את פעולות המטוס ל"מצבים" (mode) שונים:

- א. מצב "קרקע"
- ב. מצב "המתנה"
- ג. מצב "טיסה אל היעד"
- ד. מצב "סכנה" (מיידי / לא מיידי)
- ה. מצב "התגברות על מכשול"

נחלק את המצבים לשלושה תרשימים:

תרשים מס' 1: תרשים זה מתאר את המזל"ט בתחילת התהליך: המזל"ט במצב "קרקע". לאחר בדיקת תקינות המזל"ט עובר למצב "המתנה". הריבוע המסומן באדום הינו מעבר לתרשימים הבאים (2-3)



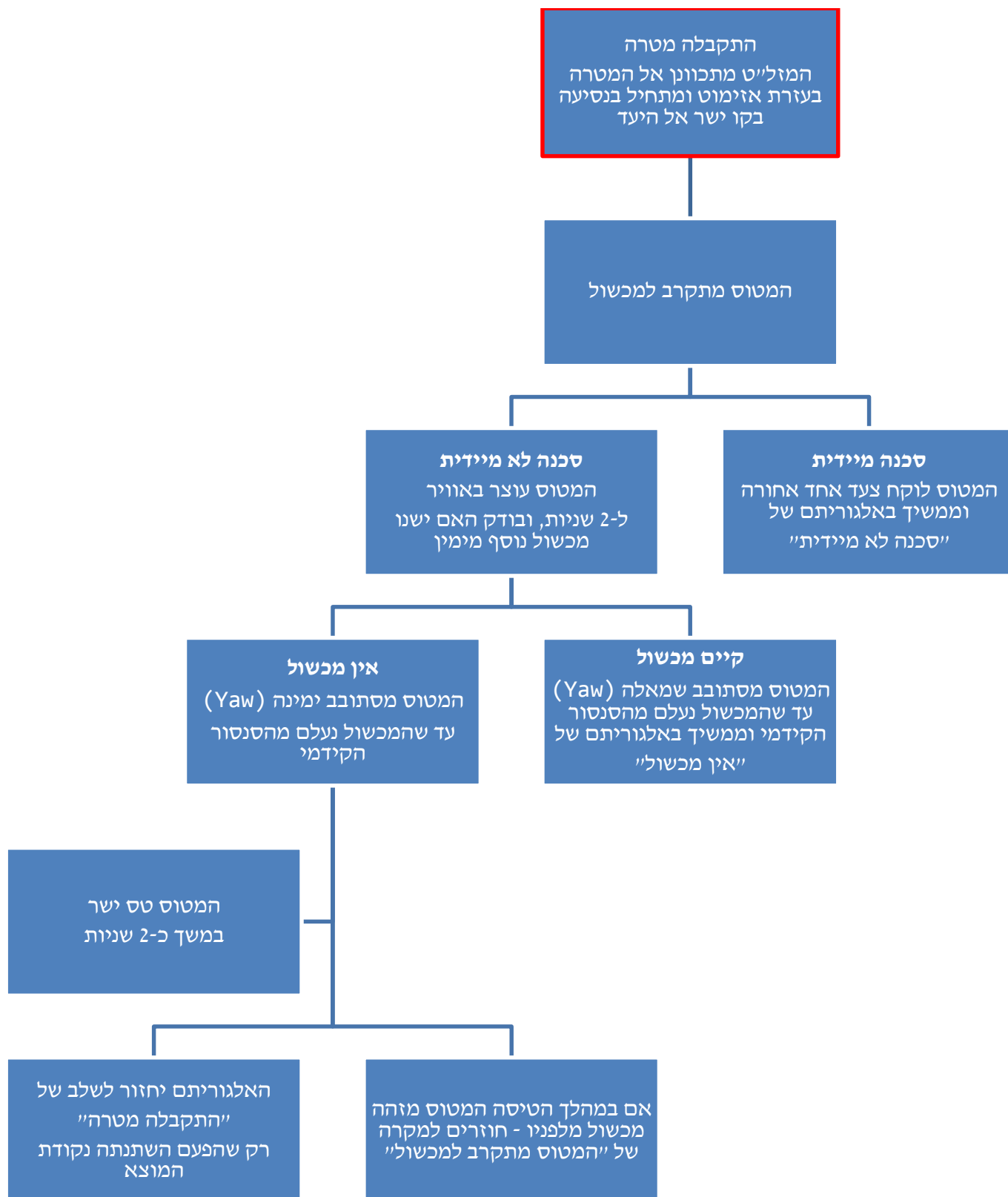
תרשים מס' 2 : תרשים זה מתאר את המצב "טיסה אל היעד", בהנחה שלא קיימים מכשולים בין נקודת המוצא אל נקודת היעד.

חשוב לשים לב: ייתכן ובשלב ראשון לא יהיו מכשולים בדרך, ובעצם נעבוד לפי תרשים זה. אך בהמשך התרשים ישנם מצבים המחזירים אותנו למצב "התקבלה מטרה", ומצב זה מתפצל ל-2 מצבים - ללא מכשולים (תרשים זה) ועם מכשולים (תרשים 3, מובא להלן).



תרשים מס' 3 : תרשים זה מתאר את המצב "סכנה" (מיידי / לא מיידי), ומצב "התגברות על מכשול".

חשוב לשים לב: ייתכן ובשלב ראשון נגיע למכשול, ובעצם נעבוד לפי תרשים זה. אך בהמשך התרשים ישנם מצבים המחזירים אותנו למצב "התקבלה מטרה", ומצב זה מתפצל ל-2 מצבים - כולל מכשולים (תרשים זה) וללא מכשולים (תרשים 2, הובא לעיל).



ייתכנות

האתגר העומד לפנינו בפרויקט זה הוא ליצור מערכת שניתן להרכיב אותה על מזל"ט והוא יוכל לטוס מנקודה לנקודה בלי התערבות יד אדם. ובנוסף, היא תצליח לנווט את המזל"ט כך שיצליח להתחמק ממכשולים (המזל"ט טס בגובה נמוך בהגדרה).

הבעיה **הראשונה** שמתעוררת היא, כאשר המזל"ט מזהה מולו מכשול במרחק כלשהו כיצד הוא יגיב. דבר ראשון, הוא יעצור. לאחר מכן ישנם כמה אפשרויות:

1. להסתובב 360 מעלות במקום ולמפות את השטח בנקודה שבה המזל"ט עומד. לאחר שידע היכן המכשול נמצא הוא יוכל לנווט ולעקוף אותו.
2. להסתובב לכיוון מוגדר מראש (לדוגמא, לצד ימין) עד שהוא לא מזהה אף מכשול שעומד מולו. לאחר מכן המזל"ט ימשיך ישר כמה מטרים, יעצור ויכוון עצמו מחדש לכיוון הנקודה הבאה שעליו להגיע.
3. לטוס לצד המכשול עד שהוא מסתיים. ברגע שהמכשול הסתיים המזל"ט יעצור ויכוון עצמו מחדש לכיוון הנקודה הבאה שעליו להגיע.

בחירת האלגוריתם של ההימנעות ממכשולים מאוד קריטי, יכול להיות שהמזל"ט לא יגיע לנקודה בכלל!

בעיה **שנייה** היא חישוב המרחק והמיקום שעל המזל"ט לטוס. המזל"ט אמור לזהות את מיקומו הנוכחי ולקבל נקודת GPS (latitude, longitude). על המזל"ט לחשב את הזווית בין הנקודה הנוכחית לנקודה הבאה, להסתובב כך שיהיה מול הנקודה ולהתחיל לטוס ישירות לנקודה.

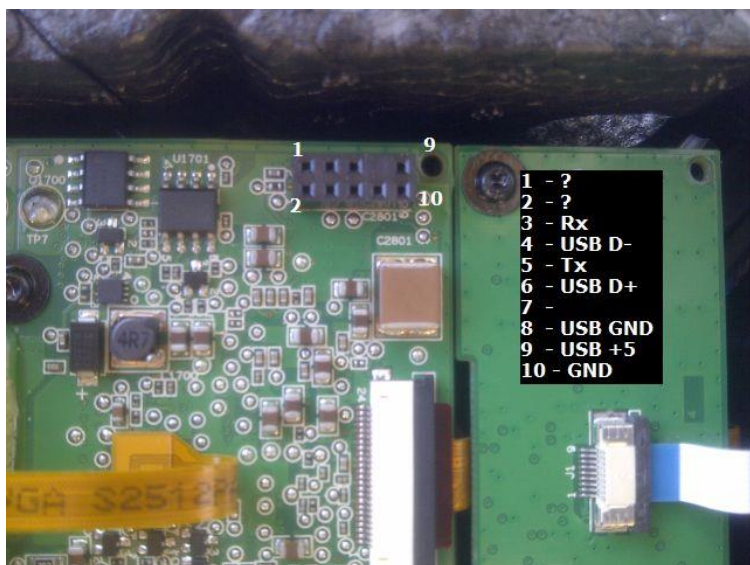
בעיה **שלישית** היא מיקום החיישנים על המזל"ט. מיקום בעייתי עלול לגרום לאי זיהוי של מכשול שנמצא מול או לצידו המזל"ט ולהוביל להתרסקותו.

בעיה **נוספת**, הנחתת המזל"ט במקרה חרום. כאשר המזל"ט יכנס למצב בעייתי (קליטת WiFi חלשה, התנהגות בלתי רצויה) אנו נרצה שתהיה אפשרות להנחית אותו בלחיצת כפתור.

עומדות לפנינו מספר אפשרויות:

אפשרות אחת: שימוש בטכנולוגיית 3G. מכשיר הסמארטפון יהיה מחובר באמצעות כרטיס sim לרשת, וברגע שהמכשיר מזהה שיחה נכנסת הוא פוקד על המזל"ט לנחות באופן מיידי.

אפשרות שנייה הפוכה לראשונה: מרגע שנכנסת שיחה המזל"ט עולה לאוויר ומחכה למשימה. ברגע שהשיחה מתנתקת (מסיבות שונות כמו: אין קליטה באיזור או שהמפעיל החליט להנחית אותו) המזל"ט באופן מיידי.



אפשרות שלישית: לחבר למזל"ט RC receiver

(החיבור הוא פיזי ל-board של המזל"ט, ראה תמונה). ברגע שמתקבלת תשדורת מסוימת המזל"ט נוחת באופן מיידי.

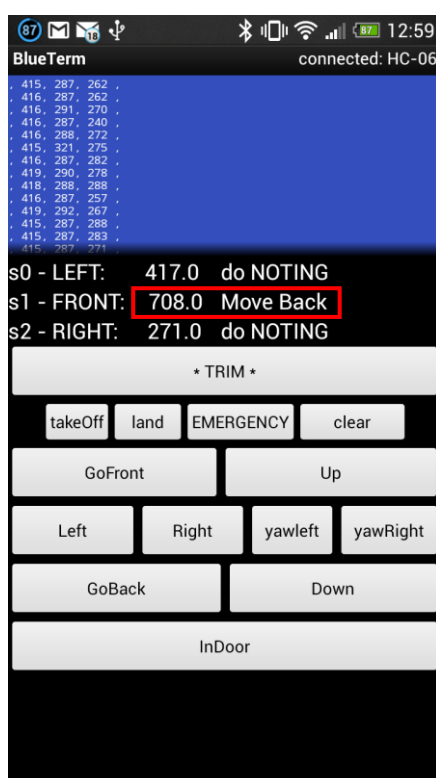
בהתחלת העבודה חיברנו את החיישנים לבקר teensy. בנוסף חיברנו למערכת רכיב Bluetooth. פלט החיישנים עובר ב: stream: לאפליקציה שיצרנו בטלפון החכם. האפליקציה מקבלת את הנתונים ושולחת פקודות למזלי"ט על פי הנתונים המתקבלים ב: stream:.

לאחר מכן חיברנו גם את ה-GPS לבקר ה-teensy וקיבלנו את הנתונים באפליקציה.

תמונות מתוך האפליקציה (גרסה ראשונית):

תמונה שמאלית מתארת מצב בו אין שום מכשול.

תמונה ימנית מתארת זיהוי מכשול בסנסור הקדמי של המטוס (השינוי מסומן במסגרת אדומה)



לוח זמנים

מועד אחרון לביצוע	מטלה
30.01.15	הגשה סופית להצעה לביצוע פרויקט
31.03.15	הצגת הפרויקט בגרסה ראשונית עובדת
31.05.15	הצגת פרויקט ביום הפרויקטים
30.09.15	הגשת המצגת, הפוסטר והציוד הנדרש

סקר ספרות

- Project Name:** Node Copter
Control of the AR.Drone using node.js.
Nodecopter team create client protocol with Node.js, platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications
Link: <http://www.nodecopter.com/>
- Project Name:** Autonomous People Tracking using Parrot AR-Drone 2.0
Using the front camera so that the plane will follow a certain character, People detection is done using Histogram of Gradient (HOG) feature and tracking with a particle filter.
Link: <https://www.youtube.com/watch?v=SJY1zxVtWsU>
- Project Name:** AR Drone Target Tracking with OpenCV - Optical Flow
Using optic flow to follow a signal
Link: <https://www.youtube.com/watch?v=C95bngCOv9Q>
- Project Name:** Helping Robots See
Explore computer-vision-based algorithms and machine-learning methods to help an ARDrone quad-rotor helicopter autonomously navigate
Link: https://www.youtube.com/watch?v=o_O2o2ly-34&list=PL63FC304EF5263230&index=13
- Project Name:** AR Drone Face Tracking
Quadcopter programmed to detect and follow a face. A Haar feature detector is used to identify a face and a camshift algorithm is used to track the face. A controller reorients the quadcopter to center the face within the video frame.
Link: <https://www.youtube.com/watch?v=VghVtljvWew>
- Project Name:** quadcopter project
Obstacle Avoidance System For Parrot AR.Drone Quadcopter, very good work with the IR sensors. Work with Node.js and JavaScript.
Link: <https://tsouthprojects.wordpress.com/category/2-quadcopter/>,
<https://www.youtube.com/watch?v=WUs2IV1kdU>
- Project Name:** Extended Wifi Signal
Antennae kit serves to increase wifi range and reliability. The antenna is connected directly to the drone and increases the WiFi range. It seems in Ardrone2 no antenna connector as in Ardrone 1.
Link: <http://ardroneshow.com/ar-drone-2-0-modded-main-board-wifi-antennae-kit/>



8. Project Name: Autonomous robotic plane flies indoors

The MIT researchers have completed a series of flight tests in which an autonomous robotic plane running their state-estimation algorithm successfully threaded its way among pillars in the parking garage under MIT's State Center.

Link: <http://newsoffice.mit.edu/2012/autonomous-robotic-plane-flies-indoors-0810>

Sensors

9. Sharp infrared IR ranger comparison

Sharp infrared detectors and rangers boast a small package, very low power consumption and a variety of output options. In order to maximize each sensor's potential, it is important to understand how these types of IR sensors work, their effective ranges, and how to interface to them.

Link: <http://www.acroname.com/articles/sharp.html>

10. Ultrasonic Sensors

Datasheets, Application Notes, Pictures, & Other Documents

This site has tutorials on how to operate the sensors. In addition, there are programs that show how to work with Arduino cards.

Link: <http://www.maxbotix.com/downloads.htm>

GPS

11. u-center GNSS evaluation software

The u-center GNSS evaluation software provides a powerful tool for evaluation, performance analysis and configuration of u-blox GNSS receivers. Its unique flexibility makes the u-center GNSS evaluation software an invaluable tool for evaluation, analysis and configuration of u-blox GNSS receivers. u-blox GNSS receivers can be configured using the u-center evaluation software.

Link: <http://www.u-blox.com/en/evaluation-tools-a-software/u-center/u-center.html>

12. NMEA Parser - Java Language

Use for parse the GPS data.

Link: <https://github.com/HvB/UsbGps4Droid/blob/master/src/org/broeuschmeul/android/gps/nmea/util/NmeaParser.java>