

המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ז

פרוטוקול דחיסת נתונים בהתקן IoT

IoT Device Compress Protocol

מאת:

אורי חליו- 201633781
איתמר גולדשטיין- 200562593

תאריך:
תאריך:

אישור:
אישור:

מנחה אקדמי: ד"ר גיא לשם
רכז הפרויקטים: מר אסף שפינר

מערכות ניהול הפרויקט :

#	מערכת	מיקום
1	מאגר קוד	https://github.com/itamargs/Iot_Project
2	יומן	https://calendar.google.com/calendar/embed?src=e0luturcbaalb57knbt17hq83k@group.calendar.google.com&pli=1&ctz=Asia/Jerusalem
3	ניהול פרויקט (אם בשימוש)	https://github.com/itamargs/Iot_Project/projects/1
4	הפצה	https://github.com/itamargs/Iot_Project/releases
5	סרטון	https://youtu.be/HEch1aGKY_A

תקציר :

במסגרת פרויקט סיום לימודי תואר בהנדסת תוכנה, נדרשנו לממש את הידע שרכשנו במהלך שנת הלימודים.

בתחילה חשבנו לפתח אפליקציה, רצינו להעמיק את הידע בסביבת Android ולחוות את עולם פיתוח האפליקציות. לאחר שנפגשנו עם המנחה האקדמי שלנו - ד"ר גיא לשם, הבנו שעולם האפליקציות מעניין ורחב אך נוכל לבצע פרויקט יותר ראוי ובתקווה, כזה שגם יתרום לתעשייה בתחום ה-IoT.

ד"ר לשם הציע לנו לפתח פרוטוקול כללי להתקני IoT, התקנים אלו נמצאים היום בשימוש רב הן בבתי מגורים והן בתעשייה.

יצירת הפרוטוקול היא בפני עצמה אתגר לא פשוט, אך נכנסו אליו מתוך ידיעה שבעזרת השם ועם הרבה כוח רצון נוכל לעמוד בו.

נתעסק עם עולם ה-IoT שתופס תאוצה רבה, נלמד שפת תכנות חדשה Python שגם כן נכנסה לעולם הפיתוח ותופסת מקום רב, נתעסק בשירותי הענן שתפקידם לקבל ולאכסן את הנתונים, וניגע בעולם התקשורת.

בעיקר נעסוק בדחיסה ואופטימיזציה של נתונים נושא שלא נגענו בו באופן מעשי כלל בעבר.

הפרוטוקול שאנו מפתחים, ייתן מענה לבעיית המידע הגולמי שתופס רוחב תקשורת גדול ולהתמודדות עם בעיית מגוון סוגי המידע שעוברים בו ובכך בתקווה יקדם בעוד מדרגה את השימוש של התקני IoT בתעשייה.

תוכן העניינים :

2	תקציר :
3	תוכן העניינים :
4	מבוא
4	תיאור הבעיה :
4	דרישות ואפיון הבעיה
4	הבעיה מבחינת הנדסת תוכנה
5	תיאור הפתרון
5	מהי המערכת
5	תהליכים ונתוני המערכת
5	תיאור הפתרון המוצע
6	תיאור הכלים המשמשים לפתרון
8	סקירת עבודות דומות בספרות והשוואה
8	נספחים
9	א. רשימת ספרות \ ביבליוגרפיה
9	ב. תרשימים וטבלאות
9	תרשימי תיכון
9	מצבי המערכת השונים מתוארים בצורה כרונולוגית
	Error! Bookmark not defined.
	מסכים
12	תכנון הפרויקט
12	טבלת סיכונים
13	רשימת טבלת דרישות

1. מבוא

התקני IoT נכנסים לתודעה הציבורית באופן הולך וגובר במהלך השנים האחרונות. הקונספט של הפיכת מכשירים "טיפשים" לחכמים ושימוש במכשירים אלו ע"מ לקבל מידע מדויק, עדכני וקל לניתוח תופס תאוצה מסיבות ברורות.

מעבר לשימוש הביתי, התקני IoT נכנסים לשימוש בתעשייה במקומות כגון מפעלים בקנה מידה גדול. קבלת המידע מההתקנים הפרוסים בשטח, תורמת לניהול חכם ופרודוקטיבי יותר של תהליך הייצור.

עם זאת, כאשר משתמשים בהתקני IoT בקנה מידה גדול (למשל שדה גידול תבואה בעל שטח גדול שמרושת כולו בהתקני IoT שמסדרים נתונים ברוחב תעבורה גדול) ניהול המידע הופך למסובך ויקר בשל העובדה שאנו מקבלים מידע גולמי שלפעמים לא רלוונטי או לא נחוץ. כמויות המידע נעשות עצומות ואיתן מגיעות בעיות נלוות כמו קושי להעביר את הנתונים במהירות, עלויות רוחב פס, גדילה בצריכת החשמל ועוד. כמו כן קיימת בעיה שלא קיים פרוטוקול כללי לטיפול בנתונים אלו ביעילות. כלומר קיים פרוטוקול שמטפל בדחיסת וידאו או תמונה אך אם ההתקן יידרש להעביר נתונים מסוג אחר או שיידרש לנו רק חלק מהמידע המופיע בוידאו או בתמונה אין לפרוטוקולים הקיימים פתרון.

לכן קיים הצורך להוציא את פיסות המידע הרלוונטיות מתוך כל המידע שמיועד לתשדורת, כלומר קיים צורך בפרוטוקול לדחיסת הנתונים המתקבלים ושידור המידע לאחר דחיסתו ומעבר לכך נדרש פרוטוקול כללי שיידע לטפל במגוון סוגי התקנים ומידע.

2. תיאור הבעיה :

2.1 דרישות ואפיון הבעיה

במצב הנוכחי, ההתקנים משדרים את כל המידע המגיע אליהם ללא סינון ובקרה, כלומר, גודל המידע הנשלח זהה לגודל המידע שקיים בהתקן גם אם הוא אינו נחוץ לנו כרגע. למשל, במידה וההתקן מקבל מידע על מיקום של חפץ כלשהו, לא נרצה שהוא יעביר את המידע, אלא אם כן התרחש שינוי במיקום החפץ. בנוסף, לא קיים פרוטוקול שיידע לטפל בסוגי המידע השונים שמתקבלים אצלו, אלא עבור כל סוג מידע נצטרך פרוטוקול שונה.

2.2 הבעיה מבחינת הנדסת תוכנה

נדרש להתמודד עם בעיות בשני מישורים :

1. דחיסת המידע :

- a. החלטה איזה מידע מוגדר כנחוץ ואיזה לא.
- b. שמירה על תמונת מצב עדכנית למרות מיפוי של חלק מהמידע.

2. יצירת פרוטוקול ג'נרי :

- a. חוסר הידיעה של איזה סוג מידע מתקבל בהתקן, יקשה על הטיפול בנתונים.
- b. ההבחנה בין סוגי המידע המתקבלים וטיפולם.
- c. התאמת פונקציות לסוגי מידע שונים.

3. תיאור הפתרון

3.1 מהי המערכת

פרוטוקול, אשר בשלב ראשון יבחין בין סוגי המידע השונים המתקבלים בהתקן, ינתח אותם ובהתאם יחליט איזה מידע נחוץ לצורך שליחה ואיזה מידע ניתן לסנן כך שבסופו של דבר נקבל מקסימום אינפורמציה תוך כדי מינימום העברה של מידע וזאת תוך שמירה על הרלוונטיות שלו.

3.2 תהליכים ונתוני המערכת

קיימים שני מצבי עבודה עיקריים- עיבוד מידע, ושליחתו. בנוסף יש חלוקה לצד לקוח וצד שרת.

צד לקוח:

- קליטת המידע- קבלת מידע מהסנסורים שמחוברים להתקן הIoT
- עיבוד המידע- החלטה איזה מידע לשמור ואיזה לזרוק.
- דחיסת המידע- העברת המידע לפורמט/תצורה שבא יוכל להיקרא בצד השרת תוך כדי קידוד הנתונים לגודל קטן יותר.
- שליחת המידע- שליחת המידע מההתקן דרך WIFI אל מסד הנתונים בענן.

צד שרת:

- ניתוח המידע ושמירתו במסד הנתונים.

3.3 תיאור הפתרון המוצע

דחיסת המידע לפני שליחתו אל מחוץ להתקן.

3.4 אלגוריתמים

נרצה להציג את שני האלגוריתמים העיקריים לפתרון,

האלגוריתם "הסטנדרטי" כלומר דרך הפעולה השוטף של המערכת, פועל בצורה הבאה, המערכת מקבלת מידע ראשוני מהחיישן לצורך נוקדת ייחוס, אליה נשווה את המידע החדש שהתקבל, עבורו נבדוק האם אכן המידע החדש שהתקבל מוגדר כשינוי מבחינת בעל המערכת, במידה וכן, ההתקן ישלח את המידע החדש אל הנתב ומשם אל הענן, במידה ולא, ההתקן "יזרוק" את המידע.

אלגוריתם לטיפול בעומס

אלגוריתם זה יימומש במקרה שנוזהה עומס על רוחב הפס, האלגוריתם ימומש בצורה הבאה :

- ההתקן שולח חיבלת גישוש לראות שהקו בינו לבין הנתב פנוי
- אם פנוי, ויש מקום בחוצץ שליחה -> שולח את המידע הרלוונטי, מעדכן את ההתקן.
- זוהתה התנגשות -> מתריאים לכל ההתקנים בשטח, ההתקן עם השינוי נתונים מקבל העדפה, ההתקן שביקש לשלוח אות חיים מקבל זמן המתנה.
- לא פנוי -> מחכה זמן מוגדר מראש ושולח מחדש.
- פנוי ואין מקום בחוצץ השליחה -> כלומר נשלחה כמו מאסיבית של פקטות או החבילות לא נשלחו מהנתב לענן.

- X החבילות הראשונות מקבלו TTL ונכנסות לתור, שאר החבילות שמגיעות נזרקות
- אם נשלחו לפני שנגמר TTL -> הענן מעדכן את הנתב שמעדכן את ההתקן
- אם עדיין לא נשלחו ועבר פרק הזמן המוגדר -> החבילה נזרקת, הנתב לא מאשר את החבילה להתקן.
- ההתקן רואה שלא התקבלה תגובה מהנתב, ממתין פרק זמן מסויים ושולח שוב (במידה והמידע רלוונטי).
- חוזר לראש האלגוריתם.

3.5 תיאור הכלים המשמשים לפתרון

לפרויקט שלנו החלטנו כי נשתמש בשפת הפיתוח Python וכל הספריות הרלוונטיות. סביבת העבודה שלנו תהיה Arduino Playground. PyCharm של JetBrains הפיתוח יבוצע על גבי ההתקן "Linkit Smart 7688 Duo" שיתפקד כרכיב המשדר את הנתונים המתקבלים מהמשדרים הרלוונטיים. לצורך שמירת המידע המתקבל- נשתמש בفلטפורמת "FireBase" של גוגל במידה ואפשרי. נשתמש בFireBase הן לצורך מסד נתונים והן לצורך קבלת התשדורות מההתקן. הפרויקט כולו ינוהל במערכת ה-GitHub אשר יכלול וינהל את:

- יומן הפגישות
- מאגר הקוד
- לוח מטלות

4. תוכנית בדיקות

הבדיקה	תיאור
בסיסית	נרצה לבדוק את המקרה בסיסי של המערכת של שליחת המידע הרלוונטי לנתב הרלוונטי
עומס תקשורת	נפעיל עומס על פס התקשורת, כלומר, נייצר מצב בו כל ההתקנים והחיישנים ישלחו מידע בו זמנית.
התנגשות	נרצה לבדוק מצב בו מתרחשת התנגשות של שני חיישנים אשר שידרו במקביל, האם המערכת יודעת לטפל במצב כזה.
זיהוי שינוי רלוונטי	נגדיר מאיזה אחוז שינוי במידע מגדיר אכן שינוי, ונראה לראות שאכן המערכת מזהה את השינוי.
בדיקת דופק	נרצה לוודא שהמערכת אכן יודעת לזהות כי ההתקן אכן במצב תקין.

5. סקירת עבודות דומות בספרות והשוואה

Wind Spring:

כיוון ואופטימיזציה להתקני IoT.

קישור:

www.windspring.com

שני מאמרים של ארגון IEEE בנוגע לדחית נתונים בהתקני IoT:

דחיסת נתונים ייעודית להתקני IoT

קישור:

<http://ieeexplore.ieee.org/document/7149287/?reload=true>

<http://ieeexplore.ieee.org/abstract/document/7179026/>

6. סיכום ומסקנות

לסיכום עד כה ניתן להגיד כי ביססנו תשתית טובה להמשך העבודה. בנינו חלק מהאלגוריתמים שישרתו אותנו לצורך המשך הפיתוח, כמו כן את ארכיטקורת המערכת, ואופן הפעלתה. כעת אנו יכולים לנווט את רוב משאבי האנרגיה והלמידה שלנו לצורך השלב העיקרי בפרוייקט, במקביל נעמיק את הידע בPython. הבנו שיש מגבלה לרמת הג'נריות שהפרוטוקול יכול לספק ובהתאם שינינו את הארכיטקטורה של גירסאת האלפא. (לארכיטקטורה מבוססת Interfaces) אין ספק שבמהלך תהליך ההתנעה למדנו הרבה בין אם זה על צורת העבודה המתאימה לנו ובין אם זה על עולם הIoT. נרצה לתכנן את לוחות הזמנים שלנו בצורה יותר טובה, במתן דגש על הפיתוחים לצורך שחרור הגרסאות הקרבות.

7. נספחים

ספרות, תרשימים נוספים, תכנון הפרויקט, טבלת ניהול סיכונים, טבלת דרישות (URD),

א. רשימת ספרות \ ביבליוגרפיה

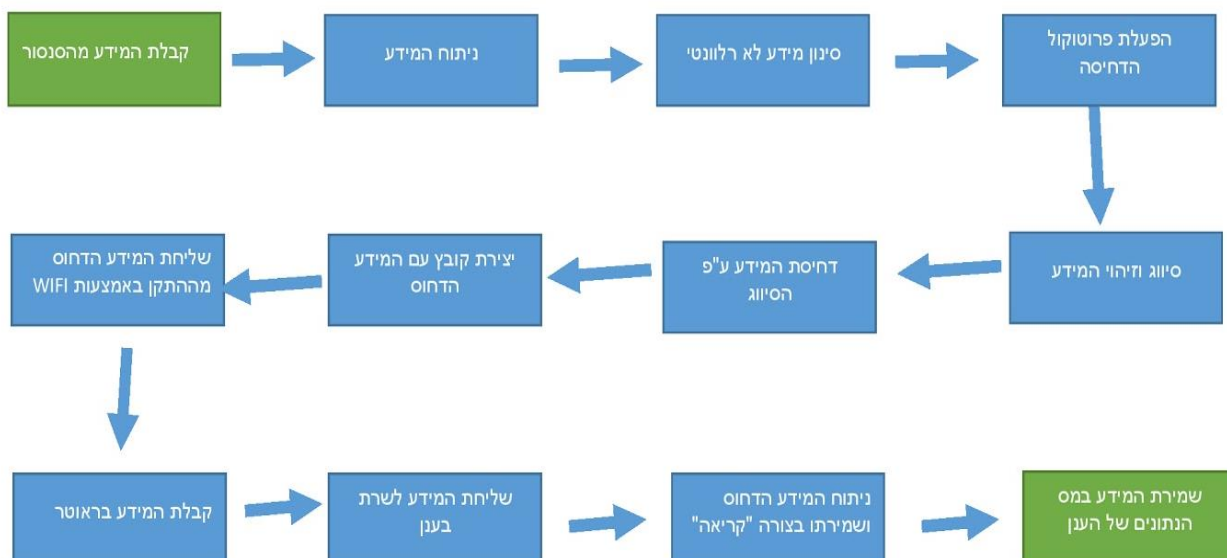
- IoT Data Compression: Sensor-Agnostic Approach **by** Arijit Ukil, Soma Bandyopadhyay and Arpan Pal
- Adaptive Sensor Data Compression in IoT systems: Sensor data analytics based approach **by** Arijit UkilSoma BandyopadhyayArpan

ב. תרשימים וטבלאות

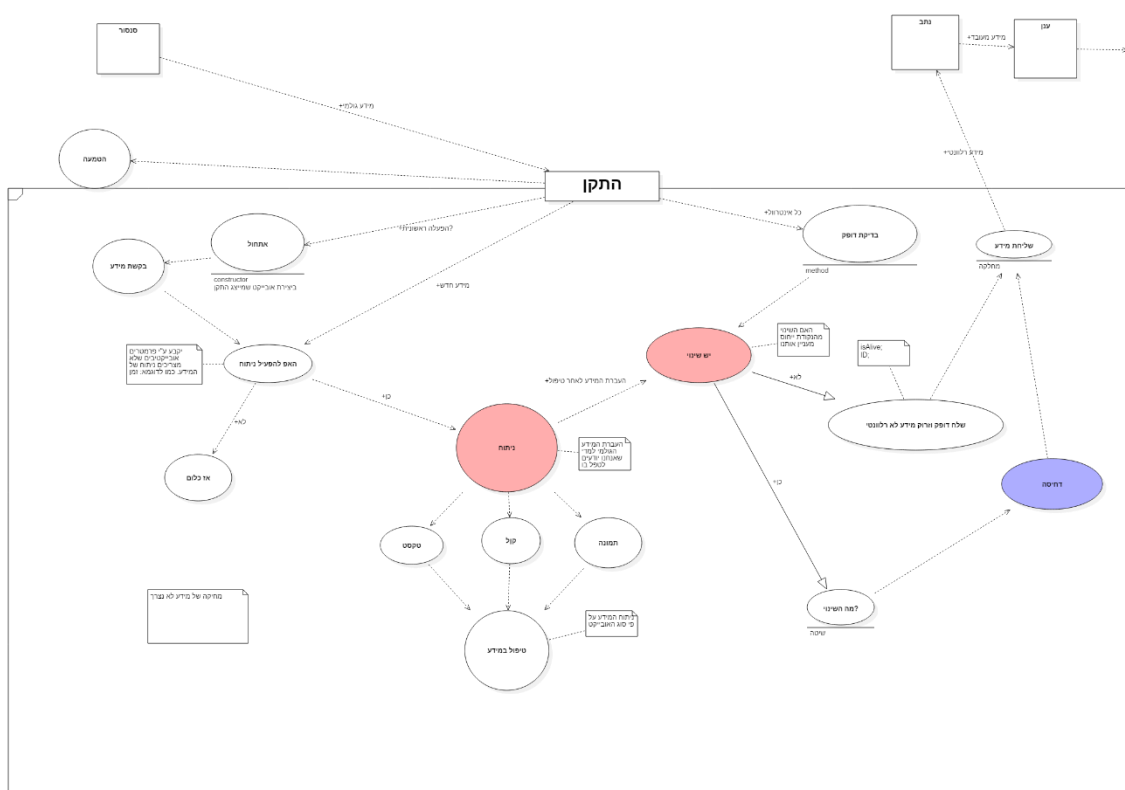
7.1 תרשימי תיכון

דיאגרמת רכיבים \ הפצה (UML), דיאגרמת ישויות, טבלאות במסד נתונים.

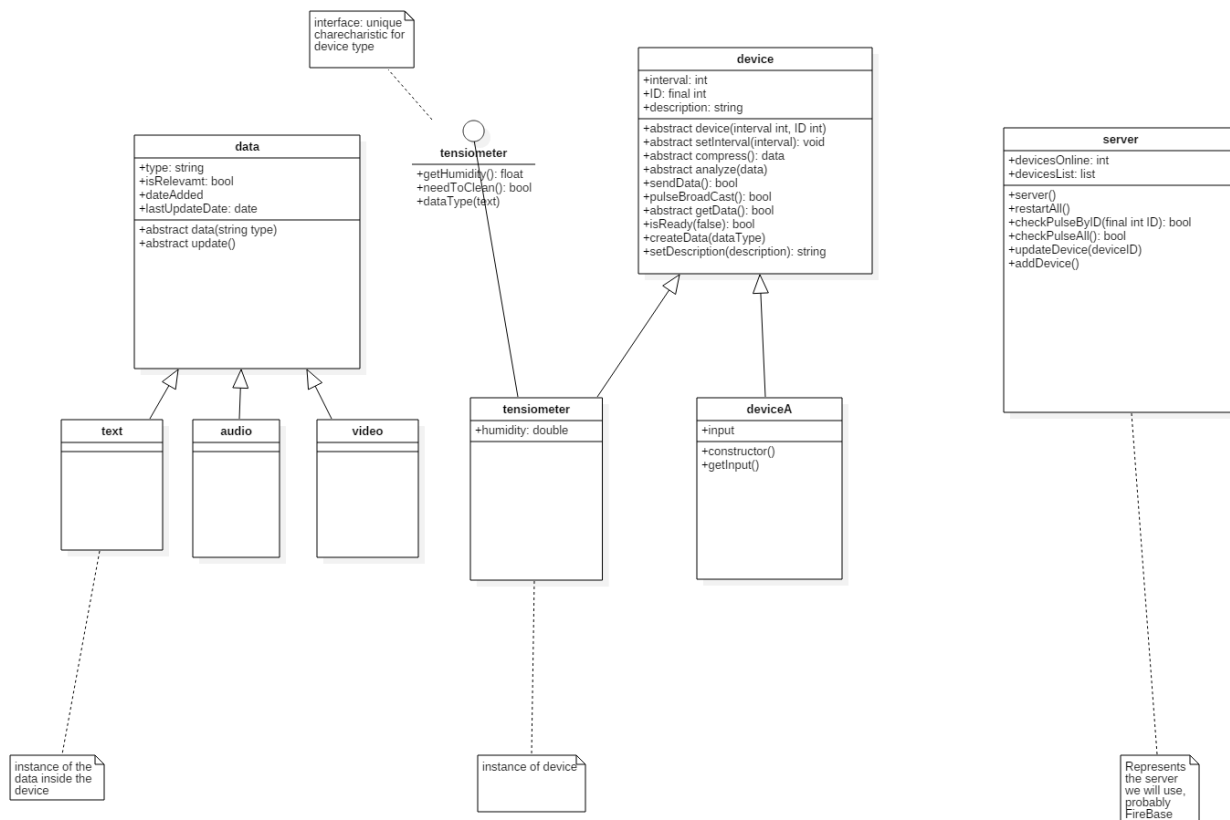
מצבי המערכת השונים מתוארים בצורה כרונולוגית.



UseCase diagram



Class Diagram



7.2 תכנון הפרויקט

פגישת היכרות עם המנחה	16.10.
פגישה נוספת עם המנחה לצורך תיאום ציפיות והבנת דרישות	12.11
הגשת דו"ח הצעה למנחה	19.11
<ul style="list-style-type: none"> הגשת מוצר אלפא, ע"פ דרישת רכו הפרויקט הגשת דו"ח אמצע הגשת סרטון 	1.2
הצגת מוצר בטא והגשת דו"ח	[תרם נקבע תאריך ע"י הרכז]
הצגת תיקונים של מוצר הבטא ובדיקה של המוצר המוגמר	[תרם תואם תאריך מול המנחה]
<ul style="list-style-type: none"> הגשת הפרויקט דו"ח סופי סרטון 	[תרם נקבע תאריך ע"י הרכז]
<ul style="list-style-type: none"> פוסטר דו"ח סיכום 	[תרם נקבע תאריך ע"י הרכז]

7.3 טבלת סיכונים

#	הסיכון	חומר ה	מענה אפשרי
1	פיתוח בשפת תכנות חדשה	4	לימוד ע"י שימוש באינטרנט וסרטוני הדרכה של Python
2	חוסר התאמה של הרכיב	2	סקר שוק וייבוא של התקן חדש יותר מותאם
3	האטת התקשורת ע"י הפרוטוקול שנפתח	5	מעקב תמידי לאורך הפיתוח בכדי לראות שקצב התקשורת עונה על הציפיות, אופטימיזציה של הפרוטוקול.
4	בעיית תאימות לסוגי מידע שונים	3	שימוש בפרוטוקולים קיימים תואמים לסוג המידע המועבר
5	חוסר הכרות מוחלט עם פיתוח פרוטוקולים	5	למידת הנושא באינטרנט והכוונה מהמנחה
7	אי עמידה בלוח זמנים	5	הבנת הדרישות לעומק ויצירת לוח זמנים עבור כל דרישה
8	אי התכנות של טיפול בסוגים שונים של מידע במסגרת פרוטוקול יחיד	4	חקירת הנושא, הכוונה מהמנחה ובמקרה של חוסר הצלחה, עבודה עם מספר פרוטוקולים שונים תחת מערכת יחידה.

7.4 רשימת טבלת דרישות

טבלת דרישות (User Requirement Document)

#	הדרישה	סוג
1	המערכת תתמוך בקבלת מידע מ N סוגי סנסורים שייקבעו במהלך העבודה על הפרוייקט (ככל הנראה 2-3)	פונקציונאלית
2	המערכת תדע לסנן מידע גולמי מיד לאחר קבלתו מהסנסור (זיכרון עבודה מוגבל בהתקן)	לא פונקציונאלי
3	המערכת תתמוך בפרוטוקול דחיסת נתונים בפיתוח עצמי שידע לדחוס את הנתונים שנמצאים בהתקן ולשלוח אותם בצורה יעילה	פונקציונאלי
4	הפרוטוקול לא יגרום לאיבוד מידע חיוני	לא פונקציונאלי
5	הפרוטוקול ישמור על רלוונטיות המידע	לא פונקציונאלי
6	המערכת תדע לקבל את המידע הדחוס בצד השרת, תשמור ותציג אותו בצורה "קריאה".	פונקציונאלי
7	המערכת תיתן אינדיקציה תמידית על תקינותה	פונקציונאלית