

## 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 שפיינר

## מערכות ניהול הפרויקט :

#	מערכת	מיקום
1	מאגר קוד	<a href="https://github.com/itamargs/Iot_Project">https://github.com/itamargs/Iot_Project</a>
2	יומן	<a href="https://calendar.google.com/calendar/embed?src=e0luturcbaalb57knbt17pli=1&amp;ctz=Asia/Jerusalem&amp;hq83k@group.calendar.google.com">https://calendar.google.com/calendar/embed?src=e0luturcbaalb57knbt17pli=1&amp;ctz=Asia/Jerusalem&amp;hq83k@group.calendar.google.com</a>
3	ניהול פרויקט (אם בשימוש)	<a href="https://github.com/itamargs/Iot_Project/projects/1">https://github.com/itamargs/Iot_Project/projects/1</a>
4	הפצה	<a href="https://github.com/itamargs/Iot_Project/releases">https://github.com/itamargs/Iot_Project/releases</a>

## תקציר :

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

בתחילה חשבנו לפתח אפליקציה, רצינו ללמוד שפת חדשה- Android ולחוות את עולם פיתוח האפליקציות שבשנים האחרונות תופס תאוצה רבה. לאחר שנפגשנו עם מנחה הפרויקט שלנו ד"ר גיא לשם, הבנו שעולם האפליקציות הוא מעניין ורחב אך נוכל לבצע פרויקט יותר ראוי מעניין ומקיף בתחום 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 :
5	מבוא
5	תיאור הבעיה :
5	דרישות ואפיון הבעיה
5	הבעיה מבחינת הנדסת תוכנה
6	תיאור הפתרון
6	מהי המערכת
6	תהליכים ונתוני המערכת
6	תיאור הפתרון המוצע
6	תיאור הכלים המשמשים לפתרון
7	סקירת עבודות דומות בספרות והשוואה
7	נספחים
7	א. רשימת ספרות \ ביבליוגרפיה
7	ב. תרשימים וטבלאות
7	תרשימי תיכון
8	מצבי המערכת השונים מתוארים בצורה כרונולוגית
8	מסכים
8	תכנון הפרויקט
9	טבלת סיכונים
9	רשימת טבלת דרישות



- מילון מונחים-בשלב מאוחר יותר
- סימנים וקיצורים- בשלב מאוחר יותר

## 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 שמשדרים 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.

## תיאור הבעיה :

### 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.

### 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.

## תיאור הפתרון

### 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.

### 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.

### תיאור הפתרון המוצע

דחיסת המידע לפני שליחתו אל מחוץ להתקן.

### תיאור הכלים המשמשים לפתרון

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

- יומן הפגישות
- מאגר הקוד
- לוח מטלות

## סקירת עבודות דומות בספרות והשוואה

### Wind Spring:

כיווץ ואופטימיזציה להתקני IoT.

קישור:

[www.windspring.com](http://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/>

## נספחים

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

א. רשימת ספרות \ ביבליוגרפיה

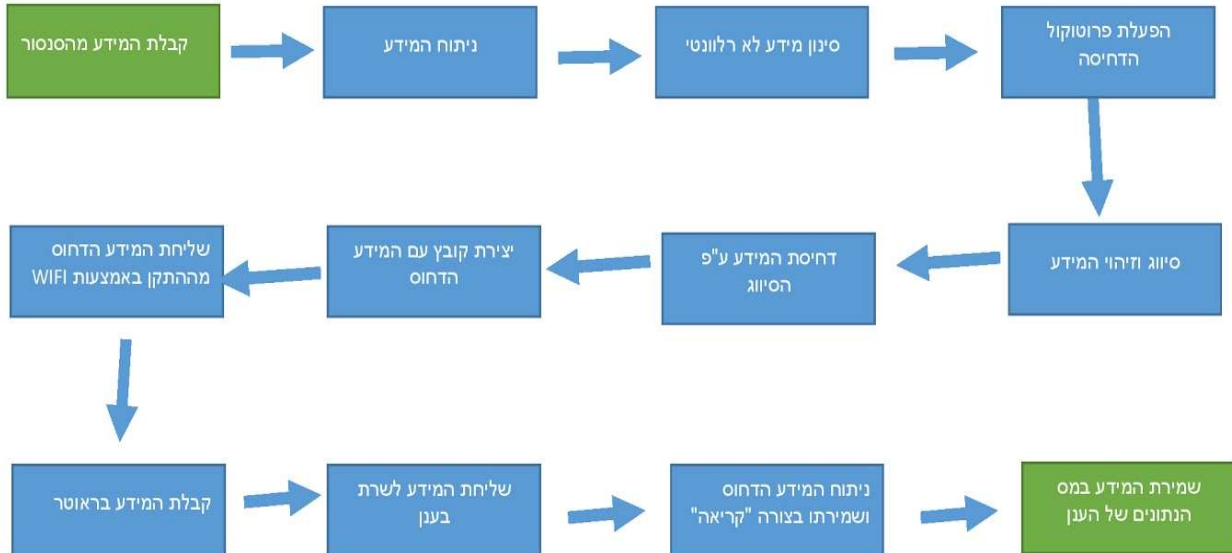
יעודכן בהמשך

ב. תרשימים וטבלאות

תרשימי תיכון

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

מצבי המערכת השונים מתוארים בצורה כרונולוגית.



**מסכים**  
**יעודכן בהמשך**

**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"> <li>הגשת מוצר אלפא, ע"פ דרישת רכז הפרויקט</li> <li>הגשת דו"ח אמצע</li> <li>הגשת סרטון</li> </ul>	[תרם נקבע תאריך ע"י הרכז]
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"> <li>הגשת הפרויקט</li> <li>דו"ח סופי</li> <li>סרטון</li> </ul>	[תרם נקבע תאריך ע"י הרכז]
<ul style="list-style-type: none"> <li>פוסטר</li> <li>דו"ח סיכום</li> </ul>	[תרם נקבע תאריך ע"י הרכז]



## טבלת סיכונים

#	הסיכון	חומרה	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.

## 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