

המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ו

פיתוח פרוטוקול כללי להעברת מידע ברכיבי IOT
Development of data transfer protocol for IOT components

מאת: אריק לוי 303175640
גיא מימוני 200481638

מנחה אקדמי: דר' גיא לשם
רכז הפרויקטים: מר אסף שפיינר
אישור: תאריך:
אישור: תאריך:

#	מערכת	מיקום
1	מאגר קוד	https://github.com/arik-le/Chips-Bits
2	יומן	https://trello.com/b/3eGlbeko/project-schedule
3	סרטון גרסת אלפא	https://1drv.ms/v/s!At98xyexrMwMecj3rDIZRg2OJ_I

תוכן עניינים

3.....	תקציר.....
4.....	מילון מונחים.....
5.....	1. מבוא.....
6.....	2. תיאור הבעיה.....
6.....	א. דרישות ואפיון הבעיה.....
7.....	ב. האתגרים הטכנולוגיים.....
7.....	ג. הבעיה מבחינת הנדסת תוכנה.....
7.....	3. תיאור הפתרון.....
7.....	א. מהו הפרויקט?.....
8.....	ב. מטרות.....
8.....	ג. מהלך הפרויקט.....
10.....	ד. מהי המערכת.....
11.....	ה. הפתרון מבחינת הפרויקט.....
12.....	ו. תהליכים ונתוני המערכת.....
13.....	ז. תיאור הכלים המשמשים לפתרון.....
13.....	4. תכנית פיתוח.....
14.....	5. תכנית בדיקות.....
15.....	6. סקירת עבודות דומות בספרות והשוואה.....
15.....	פרוטוקולי IOT להעברת מידע.....
18.....	7. סיכום\מסקנות.....
19.....	8. נספחים.....
19.....	א. רשימת ספרות \ ביבליוגרפיה.....
19.....	ב. תרשימים וטבלאות.....
23.....	ג. תכנון הפרויקט.....
23.....	ד. טבלת סיכונים.....
24.....	ה. טבלת דרישות.....

תקציר

במסגרת לימודים אקדמאיים במכללה להנדסה, הוטלה עלינו המשימה הגדולה מכל שהיא לבצע פרויקט גמר. החלטנו לבצע פרויקט מחקרי, אשר יש לו נפח ומשמעות לשימוש בתעשייה על מנת להתאים את עצמנו לעולם הטכנולוגי של היום ולשאוף להבין לאן נושבת הרוח בעתיד התעשייה והטכנולוגיה. הפרויקט אותו נבצע נוגע במגוון רחב של שימושים טכנולוגיים ובתקווה יגע בעתיד במגוון רחב יותר בזכותנו. בפרויקט זה נחקור ונפתח פרוטוקול ניתוח, סינון והעברת מידע בצורה מינימלית. פרוטוקול כללי זה מותאם לשבבי IOT, יעבוד בשיתוף עם חיישנים שונים, ממחקר מקדים שעשינו גילינו שנושא זה הוא הדבר החם הבא בתעשייה ויש לו כבר מגוון רחב של פיתוחים ושימושים שונים במערכות מתוחכמות. הבחירה שלנו נובעת מכך שהנושא זר לנו והאתגרים הקיימים בפרויקט מחקרי הם לא פשוטים, אנו סבורים שפרויקט זה יכול להכין אותנו לתעשייה ולעולם ההיי טק בצורה המיטבית ביותר.

מילון מונחים

IoT – "האינטרנט של הדברים", הוא רשת של חפצים פיזיים, או "דברים", המשובצים באלקטרוניקה, תוכנה וחיישנים המאפשרים תקשורת מתקדמת בין החפצים ויכולות איסוף והחלפת מידע. רשת זו צפויה להוביל לאוטומציה בתחומים רבים. האינטרנט של הדברים כולל בין השאר את תחומי " הבית החכם "ו"העיר החכמה".

כמו כן, האינטרנט של הדברים כולל טכנולוגיות שמאפשרות למשל: ניטור שתלי לב, שבבים המותקנים על חיות משק לצורכי ניטור ומעקב, כלי רכב המצוידים בחיישנים מובנים, התקני שטח המסייעים לכבאים בפעילויות חילוץ והצלה ועוד. השאיפה הגדולה של הטכנולוגיה היא שבעתיד ניתן לחבר כל התקן חשמלי לאינטרנט וליצור מערכת ורשת חכמה אשר תנתר ותפקח על כל המכשירים.

"האינטרנט של דברים מבטיח מהפך הרבה יותר נרחב, לא רק שימוש באינטרנט כלשלוט רחוק אלא יכולת אוטונומית של מכשירים לתקשר אחד עם השני. דוגמה טובה להתפשטות של חיישנים חכמים היא מוני החשמל החדשים המאפשרים תקשורת דו כיוונית בין הצרכנים לרשת החשמל. בעזרת חיישנים אלו יוכלו הצרכנים לדעת כאשר ישנו מחסור בחשמל ואף לכבות באופן אוטומטי מכשירים זוללי אנרגיה בשעות בהן יש עומס על הרשת."

Arduino - סביבת הפיתוח אשר מאפשרת כתיבת תכניות, הדרתן והבזקתן (Flashing) לרכיבים. אנו נעבוד בסביבה זו על מנת לבדוק את תקינות הפרוטוקול כאשר הוא משדר למרכז הבקרה .

Linkit smart 7688 duo – שבב IOT של חברת Mediatek, השבב בעל מעבד, זיכרון Ram ומטמון, כמו כן לשבב קיימים חיבורי micro-usb, Gpio pins וניתן לחבר אותו לרשת אלחוטית.

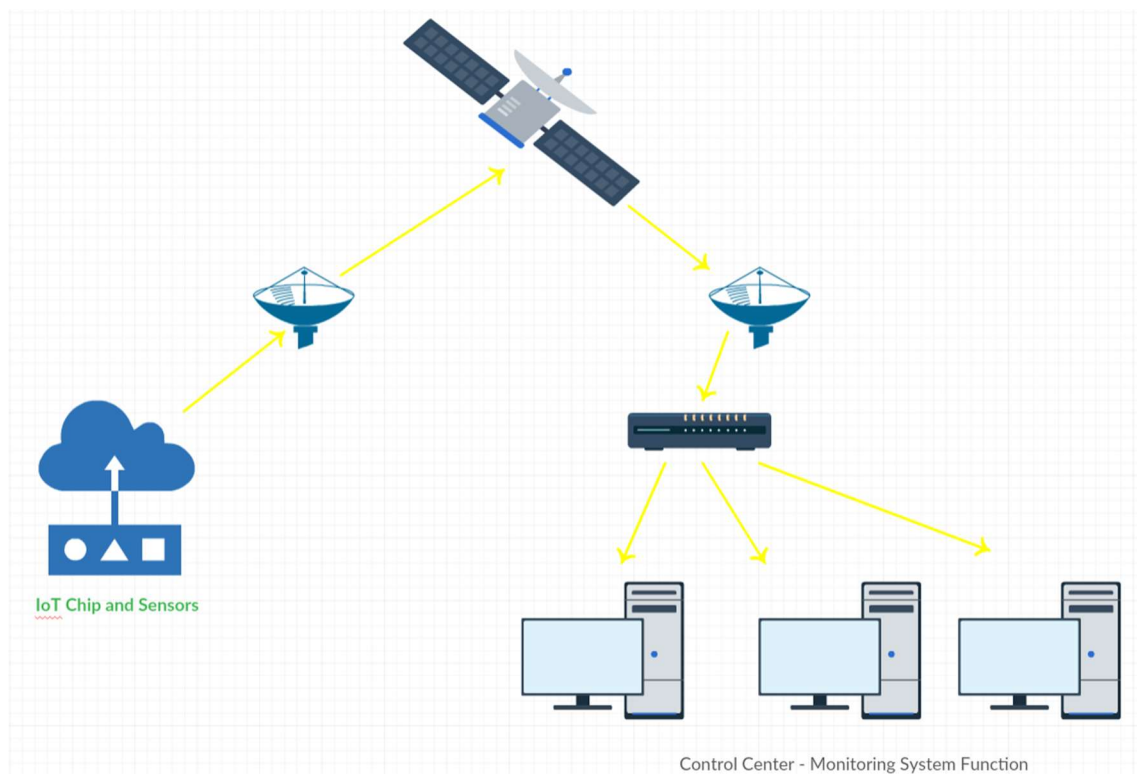
WinSCP - פלטפורמה נוחה וקלה לשימוש המשמשת אותנו להתחברות מהירה עם השבב דרך רשת מקומית משותפת והעברת קבצים לתוכו (בעיקר קבצי תכנית בשפת פיתון).

1. מבוא

המוצר שאנו שואפים לפתח הוא פרוטוקול כללי לרכיב IOT להעברת מידע מינימלית. כפי שאנו מכירים ויודעים יש מחשבים בכל מקום בעולם, גם באזורים מרוחקים ומבודדים שבהם הקליטה הסלולרית בעייתית ואין כמעט באזור חיבורי רשת חוטית ואלחוטית, אך באזורים אלו יש משאבים חיוניים ובעלי חשיבות רבה כגון: צינורות גז, מאגרי מים, אסדות קידוח ועוד. בנוסף, בעקבות התקדמות הטכנולוגיה חלק מהמערכות פועלות היום באופן מרוחק מסיבות שונות.

מה קורה כאשר יש בעיה באחד מהמערכות הללו ויש צורך להעביר את המידע כמה שיותר מהר לגורמים המוסמכים שידעו לבוא ולפתור את הבעיה?

כיום יש פרוטוקולים ורכיבים המבצעים את העבודה ומעבירים מידע למרכזי בקרה אך עלותם יקרה וכמות המידע שמועבר הוא עצום ובעל חסרון של יעילות. רוב השימוש היום, הוא בתקשורת לוויינית שכידוע קיימת בכל נקודה בכדור הארץ אך כמוכן השימוש בה יקר מאד וצורך שימוש בלוויין שיהיה זמין בכל רגע נתון, כי בכל רגע עובר המידע מהרכיבים דרך הלוויין למרכזי הבקרה.



בתרשים שמעל אנו נסביר את המצב הקיים כיום .
במערכות מרוחקות כאשר אין מרכז בקרה מקומי אשר מנתר ומפקח אחר פעילות המערכת
הקיימת (כגון מערכת ניטור נתוני גז וטמפרטורה באסדת קידוח), יש צורך בתקשורת לוויינית .
למערכת מחוברת רשת של חיישנים אשר מחוברים לשבבי IOT אשר משדרים ברשת
המקומית של המערכת את הנתונים מכל החיישנים , נתונים אלו משודרים אל לוויין התקשורת
דרך צלחות לוויין וכמובן הלוויין משדר בצד השני אל המרכז דרך צלחת לוויין שנייה המחוברת
אליו.

מה שאנו רואים פה בתרשים הוא את המידע אשר מועבר בסדר המצוין (חיצים הצהובים) דרך
תקשורת לוויינית .

העניין החשוב בנושא הוא שתהליך עיבוד ואבחון הנתונים נעשה במרכז הבקרה, כל הנתונים
מועברים אליהם ובכמויות רבות ושם נעשה כל התהליך המדובר.
בגלל כל המידע שמועבר יש צורך ברוחב פס גדול של התקשורת הלוויינית היקרה וכמובן שרתי
אחסון רבים וגדולים במרכז הבקרה על מנת להתמודד עם כל המידע שמועבר, כמו כן ניתן
לציין שיש צורך בקווי תקשורת איכותיים ובעלי יכולת העברת מידע במהירות ובכמות רבה, כל
זה מעיד על בזבז רב של כסף שמושקע במשאבים אלו על מנת לאפשר פעילות תקינה של
המערכת.

מטרתנו היא להקטין את העברת המידע למינימלית ככל שאפשר ושתועבר רק בעת הצורך
לדוגמה במקרה חריג שקורה כגון עליית מד הטמפרטורה בצינור גז לרמות מסוכנות.
דבר זה יחסוך בעלויות ככל שאפשר וכמובן שיעזור בייעול המערכת.

2. תיאור הבעיה

א. דרישות ואפיון הבעיה

הדרישות הן שיועבר מידע חיוני על מנת שתהיה פעילות שוטפת של המערכות ככל שאפשר
ומבלי שייוצרו בעיות חדשות וחמורות. הדרישות הטכנולוגיות תמיד עולות ומחמירות ככל
שמתקדמים עם השנים ומפותחות מערכות חדשות וטכנולוגיות חדשות , החברות אשר
משתמשות בטכנולוגיות האלו ירצו לחסוך כמה שיותר ולייעל את המערכות, לכן המצב הקיים
מצוי ולא רצוי, ודורש פתרון.
כפי שהסברנו קצת במבוא , הבעיות העיקריות הן : העלות והייעול .

העלות - בשל הבעיה הגיאוגרפית אשר מונעת מלהתקין כבלים, אנטנות ומכשירי רשת
אלחוטית/חוטית רוב החברות אשר מפעילות מערכות חיוניות ובעלות חשיבות משתמשות
בעיקר בתקשורת לוויינית אשר כמובן יקרה יותר ומצריכה שימוש תמידי בלוויין אשר ימקד את
רוב משאביו לשימוש זה , המידע כמובן עובר בכמויות רבות אשר מצריכות שימוש בשרתים
רבים בעלי כוח אחסון ועיבוד רב ובנוסף מרכזי בקרה אשר יתחזקו את הפעילות ויעבירו את
המידע מסוכן ויותר קונקרטי לגורמים המוסמכים.
עלות רוחב פס בלוויין היא עצומה כמו כן אחזקת שרתים רבים וחזקים בעלי יכולת עיבוד
משמעותית היא לא זולה.

הייעול – העברת המידע עוברת בצורה כמעט תמידית וצורכת שימוש ברוב פס של תקשורת לוויינית בצורה כמעט רצופה, המידע שמועבר הוא מידע רב אשר מתוכו מעט ממנו רלוונטי ונחוץ לשם העבודה השוטפת, רוב הזמן המידע הוא מידע שגרתי אשר מתאר את הפעילות השוטפת של החיישנים ושל המערכות אשר עליהן מותקנים החיישנים. כפי שרואים פעילות זו לא יעילה ולא עובדת בצורה מינימלית.

ב. האתגרים הטכנולוגיים

בגלל המרחק הגיאוגרפי, מערכות מרוחקות משדרות את הנתונים שלהם בזמן אמת דרך תקשורת לוויינית למרכזי בקרה, מערכות אלו בסופו של דבר ניחנות בחוסר יכולת חישוב מקומי, במצב כיום כל החישוב וכל עיבוד הנתונים נעשה במרכז הבקרה המרוחק ממיקום המערכת, בנוסף, כל מערכת מתנהגת בצורה שונה ובעלת חיישנים ורכיבים שונים ולכן דורשת טיפול, ניתור, פיקוח שונה ויותר ספציפי למהותה, האתגר הטכנולוגי פה שאין משהו כללי שידע להתמודד עם כל מערכת ספציפית, אף כלי לא יודע להתמודד עם כל המערכות באופן כללי.

ג. הבעיה מבחינת הנדסת תוכנה

הבעיה הראשונה שאנו צריכים להתמודד אתה עם עבודה עם ממשק וסביבות עבודה שלא היה לנו נגיעה איתם עד כה, יהיה לנו התעסקות רבה במחקר וחיפוש רב באינטרנט אחרי הכלים הכי מתאימים והכי יעילים לפיתוח הפרוטוקול שלנו. הבעיה השנייה שלנו היא לזהות את שינוי מצב המערכת במקרה והתבצע מקרה קצה כלשהו, כלומר מה זה שינוי מצב, מה זה בהתאם לחיישן אותו אנו מפעילים על הפרוטוקול שלנו כי הרי הפרוטוקול שאנו נבנה הוא פרוטוקול כללי להעברת מידע מינימלית והוא לא ספציפי לאף חיישן או כלי מדידה.

3. תיאור הפתרון

א. מהו הפרויקט?

אנו ניצור פרוטוקול כללי להעברת מידע מינימלית בשבבי IOT על מנת לפתור את אותם בעיות אלו.

הרצון שלנו שהפרוטוקול יעביר מידע **מועט ככל שאפשר ורק במקרה הצורך** כמו לדוגמה כאשר קורה משהו, ולא בצורה תמידית ושוטפת.

ב. מטרות

המטרה שלנו היא שנוכל לטפל בבעיה הנוצרת ברמה המקומית שהיא בין ההתקן החיצוני לשבב עליו יותקן הפרוטוקול, אנו לא רוצים לטרוח ולהעביר את המידע ישירות למרכז הבקרה, אלא קודם כל לזהות את השינוי, לעבד ולאבחן אותו, ואם השינוי הוא לא קיצוני (משמע לא מצריך טיפול של גורם אנושי), לטפל בו ולאפשר המשך תפעול תקין של המערכת. אם השינוי מצריך התערבות נוספת, האלגוריתם ישדר את המידע הנתון אל מרכז הבקרה. מה שאנו פותרים פה בעיקר זה התייעלות העברת המידע למינימלית ככל שאפשר ופחות עומס על קווי התקשורת.

ג. מהלך הפרויקט

אנו נממש את פיתוח הפרוטוקול שיורכב משלושה שלבים עיקריים אותם נחקור ונפתח לאורך כל השנה ולפי הסדר הנכון על מנת להגיע לתוצאות הרצויות.

שלושת השלבים:

1. **שלב קליטת הנתונים** - בשלב זה נשתמש במספר חיישנים אשר יחוברו במספר דרכים שונות אל השבב (דרך פני GPIO או ADC, חיבורי USB), האלגוריתם ינסה לקלוט נתונים מהחיישנים ולהציג אותם על המסך ברמת המשתמש על מנת לממש את המשך הפרוטוקול. נחקור רבות באינטרנט כי בשלב הזה אנו מתעסקים רבות עם התקנים חיצוניים שאין לנו נגיעה בהם ברמת החומרה, למעט חיבורם והפעלתם. נממש קטעי קוד ואלגוריתמים מתאימים להרצת החיישנים וקליטת הנתונים המתקבלים מהם.

2. **שלב עיבוד הנתונים** - בשלב זה ניקח את הנתונים שקלטנו מהחיישנים, נרצה לעבד את הנתונים ולאבחן אם נמצא שינוי במצב הקיים. המערכת תדע לעבד ולאבחן את הנתונים המגיעים ולהשוות אותה למצב הקיים, משמע לתבנית התחלתית אשר תהיה קיימת באלגוריתם ותהווה מדד למצב תקין ושגרת.

ההשוואות יבוצעו בצורה מתוככמת תוך מתן תשומת לב למדידה בהתחשב לגורמים שונים אשר יעידו שלא בהכרח יש שינוי מצב, אלא זהו שינוי שגרת שקורה בדרך כלל.

דוגמה למצב כזה היא בשימוש של חיישן אור למול מצב האור במשך היום, השמש כפי שאנו מכירים זורחת בבוקר ושוקעת בערב אך במשך היום מצב האור משתנה משעה לשעה או מזמנים תכופים וקבועים והוא נחלש או מתחזק בהדרגה במשך היום. את השינויים האלו בין מדידת אור אחת לשנייה (כל מדידה תעשה בזמן אחר, סדר מסוים וברצף לאורך היום) נקרא להם דלתא, היות והדלתאות הללו הם שגרתיים ומחוברים למציאות, השינויים הללו צריכים

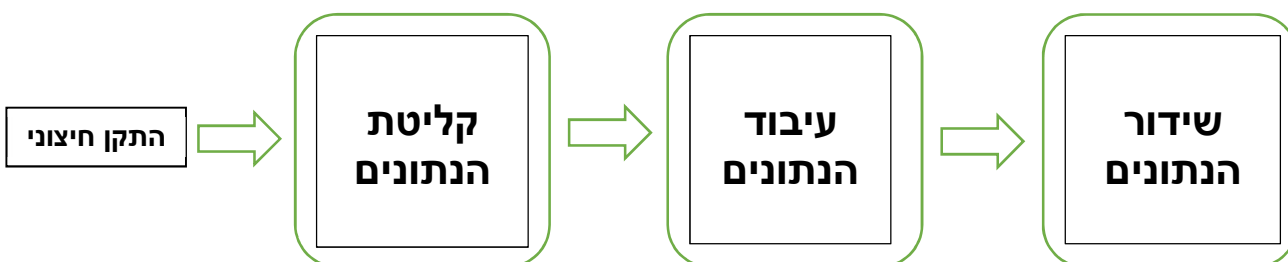
להיתפס על ידי המערכת כשינוי לא חריג ולאפשר למערכת להמשיך בפעילות שוטפת ושגרתית.
שינויים שלא יחשבו כשגרתיים אך לא חמורים, נרצה שהאלגוריתם ינסה לטפל בהם באופן אוטומטי ועצמאי באמצעות ספריית תיקונים או כללים שיימצאו בפרוטוקול.
כל זה יבוצע מכמה סיבות:

- חיסכון בהעברת מידע, הורדת בעומס על קווי התקשורת ורוחב פס הלוויין.
- הפחתת נטל העבודה מהגורמים האנושיים במרכז הבקרה.
- אפשרור המשך פעילות תקינה ושוטפת של המערכת.

הפרוטוקול לא ידע להתמודד עם שינויים חריגים וקיצוניים (כמו למשל עליית טמפרטורה חריגה וקיצונית בצינור גז) ולכן יבחר להעביר את המידע בזמן אמת אל מרכז הבקרה דרך קווי התקשורת המקובלים.

3. **שלב שידור הנתונים** – בשלב זה נשדר את הנתונים אל מרכז הבקרה שהוא הגורם האנושי (רמת האפליקציה).
מכיוון שהפרוטוקול לא ידע להתמודד עם שינוי המצב החריג, הוא יבחר להעביר את המידע הלאה, המידע יועבר בצורה מסודרת ובחבילה מרוכזת עם נתונים רלוונטיים בלבד המעידים על השינוי שנתפס, היכן נתפס והזמן שהוא נתפס. בפרויקט שלנו אנו נדמה את מרכז הבקרה כדי להמחיש את הגורם האנושי, המחשבות שלנו הוא שהמידע יועבר דרך הרשת לענן ומשם נצפה במידע המתקבל מהפרוטוקול.

להלן תרשים זרימה של הפרוטוקול:



ד. מהי המערכת

Application

Transport

Network

Data Process

Physical

המערכת שלנו תהיה פרוטוקול המחולק לשכבות בדומה למודל השכבות של פרוטוקול התקשורת Open System Interconnection.

בכדי להמחיש בצורה הברורה ביותר איך הפרוטוקול שלנו מחולק, אנו נסביר כל שכבה בפירוט:

השכבה הפיזית – זוהי שכבת החומרה, בשכבה זו השבב עליו מותקן הפרוטוקול תחובר לחיישן בחיבורים פיזיים כמו כבלי מתח לפינים או כבל USB ודרכם יועבר הנתונים. חשוב לציין שהנתונים עוברים באופן בינארי ומטרתנו להמיר אותם לנתונים דיגיטליים על מנת להמשיך לשלבים הבאים בהתקדמות הפרוטוקול.

שכבת עיבוד הנתונים – זוהי השכבה העיקרית בתהליך הפרוטוקול, בשכבה זו מבוצעות כלל חישובי הנתונים, אבחון וניתור.

בשכבה זו הפרוטוקול שלנו מבצע את מה שהוא אמור לעשות:

1. לקלוט נתונים.
2. לבדוק אם בוצע שינוי במצב המערכת.
3. אם בוצע שינוי, לקבוע אם השינוי הוא בר תיקון או שינוי שהוא שגרי.
4. אם השינוי לא בר תיקון או שגרי, לעטוף את הנתונים בחבילת מידע מסודרת עם כל הנתונים הרלוונטיים.
5. לשלוח את המידע.

שכבת הרשת – שכבה זו מתפרשת בפרוטוקול שלנו ביכולת שידור הנתונים ברשת אל מרכז הבקרה.

היא עושה זאת באמצעות מיפוי לוגי של הרשת ושל העברת הנתונים, אנו פחות נתעסק בשכבה זו ונשתמש בפרוטוקולים נפוצים כדי להעביר את הנתונים ברחבי הרשת.

שכבת התעבורה – שכבה זו גם מתפרשת בפרוטוקול שלנו בשידור הנתונים במקרה ויש צורך בהתערבות אנושית במרכז הבקרה, אחראית על ניהול התקשורת, אמינות החיבור ואמינות הנתונים.

גם פה פחות נתעסק ונשתמש בפרוטוקול תקשורת נפוץ ואמין שהוא פרוטוקול TCP כדי להעביר את המידע הלאה בצורה אמינה, טובה ומהירה.

שכבת האפליקציה – שכבה זו מדמה את הגורם האנושי, במקרה שלנו הוא מרכז הבקרה, בנגיעה לפרוטוקול שלנו שכבה זו פחות רלוונטית אבל חשבנו שחשוב לציין אותה במודל שלנו כדי להמחיש את מרכז הבקרה כאשר הוא קיבל את הנתונים כתוצאה משינוי המצב.

ה. הפתרון מבחינת הפרויקט

הפתרון יהיה שמרכז הבקרה יקבל מידע אחת לכמה זמן ו/או במקרה חריג יקבל מידע נוסף על שינוי המצב, השאיפה היא שהאלגוריתם ילמד את עצמו לתפעל ולפתור את הבעיות כך שהמערכת תמשיך לשלוח ביט בודד להראות שהמצב תקין, פתרון זה כמובן באופן ישיר יחסוך בעלויות אחסון ותשתיות רחב פס לשימוש במערכת תקשורת לוויינית אשר יקרה במיוחד וכמובן שרתים לאחסון המידע.

בפרק הנספחים ניתן לראות את פעולות הפרוטוקול בתרשים הפעולות (Activity Diagram)

The Protocol Algorithm

```
if initialSample is not define          /* initial pattern to compare with real time data */
    initialSample = start measurements()

loop(forever)
{
    if collect real-time data is different from initialSample
    {
        try check if data changed in expected way
        {
            Every period update InitialSample
            check for fix in known solutions library/history
            transmit 1-bit data /* ensure system correctness */
        }
        catch /*change in data is more different and require further handling*/
        {
            transmit data to control center /* human entity */
        }
    }
    else
    {
        transmit 1-bit data /* ensure system correctness */
    }
} /* End of loop forever */
```

ניתן לראות למעלה את הפסאודו- קוד של הפרוטוקול שלנו .

ו. תהליכים ונתוני המערכת

המערכת תכלול 2 מצבי עבודה עיקריים :

מצב תקין - מצב זה יהיה באופן שוטף כאשר אין לחיישן משהו חדש שהוא זיהה , מצב זה אומר שמצב המערכת תקין והחיישן לא מזהה משהו מיוחד אשר מצריך שינוי מצב ושליחת מידע נוסף , במצב זה ישלח בצורה שוטפת או לסירוגן (לדוגמה: פעם בכמה דקות) ביט בודד ללא מידע נוסף אשר יהווה הוכחה למרכז בקרה שהכל תקין במערכת.

מצב אירוע חריג - מצב זה יקרה כאשר קרה אירוע חריג (לדוגמה: עליית טמפרטורה חריגה מהנורמה), מצב זה יצריך מעבר של ביטי מידע נוספים למרכזי הבקרה באופן מידי וללא דיחוי, כמובן שמצב זה ושליחת מידע נוסף יעיד שמצב המערכת עליה מותקן החיישן – לא תקין, הרצון הוא שיתווסף לפרוטוקול אלגוריתמים וספריות אשר ידעו לתפעל את האירוע ולקבוע האם האירוע/בעיה הוא בר תיקון ברמת המערכת (למשל: טמפרטורה חריגה במעט מהנורמה אך יכולה עדיין להיות נסבלת מבחינת המערכת) וכמובן לתקן את הבעיה, דבר זה לא יצריך שליחת המידע למרכז הבקרה.

בפרק הנספחים ניתן לראות את מצבי המערכת בתרשים המצבים.

ז. תיאור הכלים המשמשים לפתרון

Pycharm – סביבת עבודה מאד נוחה בשפת Python, אנו נשתמש בגרסת Python 2.7 סביבה זו מאפשרת כתיבת קוד, עוזרת בתיקון שגיאות ומהדרת אותם בצורה מהירה.

WinSCP – כלי פיתוח המאפשר חיבור מהיר ברשת לרכיב ה-IOT כלי זה מאפשר העברת קבצים מהירה ברשת מהמחשב המקומי אל שטח אחסון היושב בשבב, כמו כן ניתן לגשת לכל קבצי המערכת בשבב, הכלי מאפשר שינוי גישה לקבצים (כתיבה, קריאה, מנהל) ובנוסף הכלי מאפשר הרצת תכניות וקבצים על השבב.

הפרויקט ינוהל לחלוטין על ידי מערכת ה-GitHub אשר תספק לנו:

- מאגר הקוד.
- מערכת משימות ומטלות.
- מצב הפרויקט.
- מסמכים ומידע נוסף שרלוונטי לפרויקט.

כמו כן אנו ננהל יומן אירועים של מהלך הפרויקט מתחילתו ועד סופו באתר [Trello](https://trello.com/).

4. תכנית פיתוח

לאחר שביצענו את כל המחקר הדרוש לנו לפיתוח הפרוטוקול, אנו נתחיל בחלק השני של השנה לפתח את הפרוטוקול ולממש אותו בקוד, כפי שהסברנו מקודם, אנו נממש את הפרוטוקול בשפת Python.

שלבי עבודה:

1. בכדי שלפרוטוקול יהיה דגימה שהיא תבנית ברירת המחדל, אנו נגדיר בהתחלה תבנית הפרוטוקול יבדוק אם התבנית כבר מוגדרת, אם לא הוא יגדיר אותה באופן אוטומטי לפי כללים והנחות שהוגדרו מבעוד מועד על פי החיישן ואיזה נתונים אנו קולטים ממנו.
2. הגדרת פונקציית קליטת נתוני זמן אמת מהחיישן.
3. הגדרת פונקציית השוואה מתוחכמת בין הנתונים המתקבלים לתבנית, הפונקציה תדע להשוות ולהכריע איזה סוג שינוי זה (שגרת, בר תיקון, חריג).
4. בניה והגדרת פונקציית עדכון התבנית שתעבוד כל זמן מסוים ומוגדר על ידי המשתמש.
5. בניית מנוע חיפוש אחר תיקון אפשרי לשינוי, בשלב ההתחלתי אנו נגדיר כללים וחוקים להתמודדות עם תקלות ושינויים צפויים שאנו רוצים שהאלגוריתם יתפוס ויטפל, המטרה היא שאפשרות תיקון זו תהיה ברת פיתוח והרחבה על ידי משתמשי הפרוטוקול בעתיד . (לדוגמה: אלגוריתם לא תפס את התקלה בפעם הראשונה, מרכז הבקרה טיפל בבעיה והכווין את הפרוטוקול איך לטפל בבעיה בפעם הבאה שהוא ייתקל בו).
6. בנייה של חבילת מידע אשר תורכב מהנתונים רלוונטיים בלבד.
7. הגדרת פונקציית שידור של נתונים, עבודה על חיבור לצד שרת בעבודה עם ענן.
8. הרכבת הפרוטוקול במלואו.

5. תכנית בדיקות

בכדי לבדוק את הפרוטוקול בצורה יסודית, מקיפה ושתביא אותנו למסקנות נחרצות לגבי נכונותו, אנו נבצע תכנית בדיקות אשר תורכב ממספר של חיישנים שונים. הבחירה שלנו בכמה חיישנים היא להראות את עבודת הפרוטוקול בצורה כללית בכדי לענות על המטרות שלנו.

בדיקות הרצת הפרוטוקול על מספר חיישנים ובכל חיישן נבצע מספר רב של הרצות:

1. הרצת הפרוטוקול במצב שגרה ללא תפיסת שינוי.
2. הרצת הפרוטוקול במצב שגרה כאשר קרה שינוי שגרת.
3. הרצת הפרוטוקול במצב שגרה כאשר קרה שינוי חריג.
4. הרצת הפרוטוקול ללא תבנית התחלתית המהווה מדד למצב תקין.

בשלב הסופי אנו נבצע בדיקות על שידור הנתונים למרכז הבקרה, בדיקות אמינות הנתונים כפי שנשלחו מהפרוטוקול וכמובן הצגתן על המסך כפי שהם אמורים להיות.

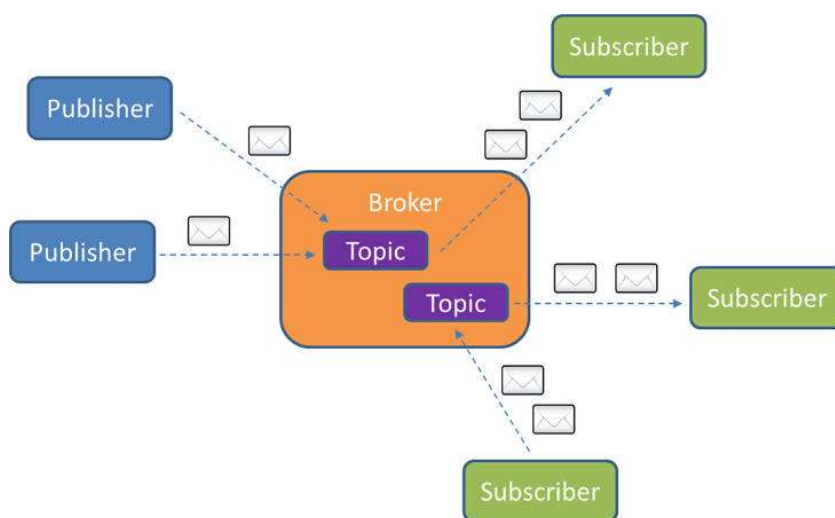
אנו מצפים שהפרוטוקול יעבוד בצורה חלקה במגוון רחב של חיישנים, יצליח לגלות כל שינוי לא רצוי בפעילות המערכת, בנוסף, נרצה שהפרוטוקול ידע להבדיל בין השינויים השונים בכדי לאפשר לתקינות המערכת.

6. סקירת עבודות דומות בספרות והשוואה

היות ואנחנו חוקרים ומפתחים פרוטוקול עיבוד והעברת מידע אשר לא קיים עד כה, הפרוטוקולים הכי דומים שמצאנו הן פרוטוקולים להעברת מידע אשר מעבירים מידע לא מסונן מרכיבי ה IOT אל הרשת. אנו נפרט פה על כמה פרוטוקולים דומים שבעזרתם נקבל פרסקטיבה על מה שאנו רוצים מהפרוטוקול שלנו.

פרוטוקולי IOT להעברת מידע

1. **MQTT** - פרוטוקול חיבוריות ממכונה למכונה (M2M) כמו כן הוא פועל גם ברכיבי IOT, פרוטוקול זה קל מבחינת זיכרון ויכולת עיבוד כך שאפשר להפעיל אותו על רכיבים וחיישנים קטנים, בנוסף הוא אידיאלי לשימוש ברשת חיישנים בחיבור מרחוק ע"י לוויין. הפרוטוקול פותח ב-1999 על ידי אנדי סטנפורד-קלארק וארלן ניפר. ב-2013 חברת IBM הגישה ל OASIS (ארגון הקובע סטנדרטים בתחום התוכנה) את MQTT v3.1 ומאז היא הפכה לסטנדרט עולמי, לפרוטוקול הזה מגוון שימושים בתעשייה וביניהם פייסבוק מסנג'ר אשר משתמש בפרוטוקול זה. המאפיין העיקרי שלו הוא ללא איבוד נתונים ולכן הוא יושב מעל שכבת TCP ובכך מאפשר פשטות ושליחת נתונים אמינה. דפוס העבודה ב MQTT הוא של publish/subscribe, כל מכשיר יכול לפרסם מידע בנושא מסוים לכל שאר המכשירים שעשו מנוי לנושא זה, ולהפך, הוא יכול לעשות מנוי לנושא מסוים. כל המידע הזה עובר דרך המתווך שמקושר לכל המכשירים, וזוהי כל העבודה שלו.



יתרונות:

- קל מבחינת זיכרון ויכולת עיבוד
- מאפשר הפעלה מרחוק על רשת חיישנים ע"י חיבור לוויין
- העברת מידע מהירה מעל TCP

חסרונות:

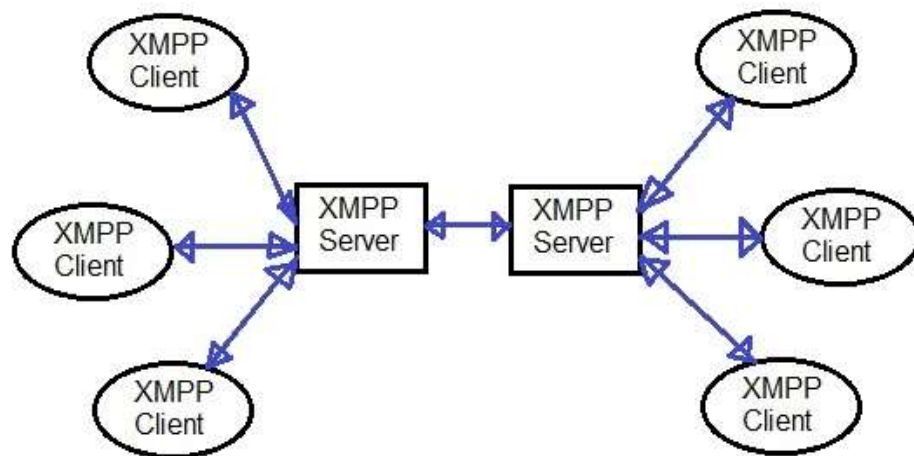
- לא עובד בזמן אמת, ולרוב זמן אמת נחשב אצלו כמה שניות.
- מסובך להפעלה, קשה להתקנה.

הפרוטוקול הזה דומה במהותו למה שאנו רוצים להשיג מהפרוטוקול שלנו , קל זיכרון , מאפשר העברת מידע בצורה פשוטה וניתן להפעיל אותו מרחוק, אך כמובן אצלנו הפרוטוקול מבצע עוד פעולות מרכזיות שהן עיבוד הנתונים .

2. **XMPP – פרוטוקול תקשורת והעברת נתונים**, מאפשר החלפת מסרים כמעט בזמן אמת, במקור נקרא "Jabber", והוא פותח על ידי קהילת קוד-פתוח ששמה ג'אבר בשנת 1999.

הוא תוכן להיות בר הרחבה , והשתמשו בו בין היתר להעברת קבצים , וידאו , משחקים וכמובן IOT ביישומים כגון רשת חכמה. בניגוד לפרוטוקולים אחרים בתחום , XMPP תוכן כפרוטוקול פתוח בעל הפצה חופשית ולכן כל אחד יכול להשתמש בקוד ולבנות לו XMPP מותאם אישית .

מנגנון הפרוטוקול עובד בצורה יותר יעילה בעיקר בזה שהוא עובד על ידי דחיפת ההודעות בניגוד לפרוטוקולים דומים כדוגמת Http שמשמש בבקשות ותגובות , בהקשר של IOT, הוא מאפשר דרך קלה להגיע להתקן וליצור לו כתובת ברשת, זה מאד שימושי אם הנתונים עוברים מרחוק, בדרך כלל בין 2 נקודות לא קשורות. הוא לא נועד להיות מהיר ולרוב השימוש שלו הוא במשיכות (pull) מידע במקרה הצורך , במונחים אנושיים זמן אמת שלו יהיה כמה שניות, הוא מספק דרך מצוינת לחבר מכשירי חשמל ביתיים (כגון: תרמוסטט) לשרת אינטרנט כך שתוכל לגשת אליו מהפלאפון שלך .



יתרונות:

- פרוטוקול פתוח, בר הרחבה.
- יכולת חיבור מכשירי חשמל I – IOT לשרת אינטרנט.
- מחבר בין 2 נקודות בצורה לא ישירה , 2 הנקודות לא חייבות להכיר באופן ישיר אחד את השני.

חסרונות:

- איטי ולא עובד בזמן אמת .

- לא עובד ללא תלוי שרת XMPP .

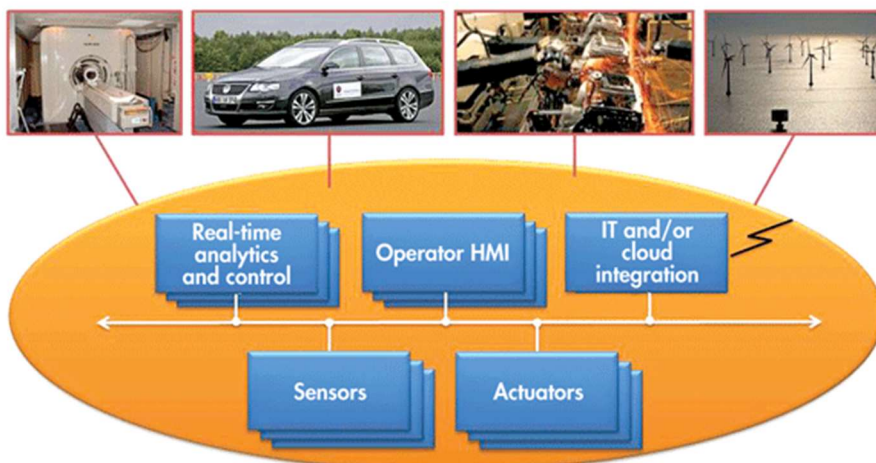
אנו נחקור מעט על הפרוטוקול הזה כי יש בו פיצורים שיכולים לעזור לנו בפיתוח הפרוטוקול שלנו .

3. **DDS** - בניגוד ל MQTT ול- XMPP, שירות הפצת הנתונים (DDS) מתמקד במכשירים שמשתמשים באופן ישיר בזיכרון המכשיר, המטרה העיקרית של הפרוטוקול הזה הוא לחבר מכשירים מסוימים לאחרים, בניגוד לאחרים DDS יכול להעביר מיליוני הודעות לשנייה בצורה יעילה ואמינה לכמה מכשירים בו זמנית.

מנגנון הפרוטוקול עובד בצורת מפרסם/מנוי, היות והוא מציע שירות למלא מכשירים עם הרבה הודעות בו זמנית, חיבור TCP לא ישים פה כי הוא מאפשר בדרך כלל חיבור מוגבל בין 2 נקודות ולכן ב DDS יש שימוש ב quality of service (בקרת איכות מפורטת), אשר דוגלת בשידור בו זמנית לכמה מכשירים (multicast), ניתנת להגדרה ותמיכה יתירה .

פרוטוקול זה מציע דרך עוצמתית לסנן איזה מידע ילך ולאן, וה- "לאן" יכול להיות אלפי מכשירים בו זמנית, הוא מאד דומה למעין אוטובוס-מידע, אוטובוס המידע שולט על כניסת המידע ויש לו דימוי של מסד נתונים כי במסד נתונים יש שליטה על הרשאות הגישה אל המידע (data) המאוחסן, חלק מהמכשירים קטנים ולכן יש גרסאות קלות של DDS שרצות עליהן בצורה מוגבלת.

מערכות משולבות בעלות ביצועים גבוהים משתמשות ב - DDS, זוהי הטכנולוגיה היחידה שמספקת את הגמישות, האמינות והמהירות הדרושים לבניית יישומים מורכבים בזמן אמת. דוגמה לכאלו: מערכות צבאיות, חוות רוח, מערכות אינטגרציה של בתי חולים, מכשירי הדמיה רפואית ועוד.



יתרונות:

- מהיר ומשדר בזמן אמת (מיקרו-שניה).

- יכול לתקשר עם אלפי מכשירים בו זמנית.
- מסוגל לשדר מיליוני הודעות בשנייה.

חסרונות:

- לא קל לשימוש , דורש ידע מעמיק ברשתות אינטרנט ומודל השכבות.
- פחות אמין כי הוא לא משתמש בחיבור TCP.

7. סיכום\מסקנות

חקרנו רבות על רכיב ה IOT ברשותנו, למדנו רבות על המאפיינים שלו, חומרה, דרכי גישה של הנתונים אל המכשיר וממנו, חיישנים שנוכל לחבר אליו ונצליח לקלוט מהם מידע וכמובן איך לחבר אותו למחשב ולהריץ ולפעול עליו בזמן אמת.

היות והמכשיר מריץ קבצי סקריפט בשפת Python , למדנו את השפה בעצמנו ואנחנו עדיין לומדים תוך כדי תנועה, גילינו שהיא שפה מאד נוחה וטומנת בה ספריות רבות שבאו לשפר ולעזור לממשק המשתמש.

לאחר המחקר המקיף התחלנו בעבודה על הפרוטוקול ועבדנו לפי השלבים שהצבנו לעצמנו שהפרוטוקול עובד.

את שלב קליטת הנתונים עברנו לאחר מגוון רחב של אתגרים עם החיישנים השונים והסוגי חיבורים השונים, חשוב להדגיש שהקושי נבע בעיקר בגלל שזה דברים שלא היה לנו נגיעה בהם עד כה והם יותר נגעו בתחום האלקטרוניקה.

אנחנו בעיצומו של שלב עיבוד הנתונים וזהו שלב שיותר נוגע אלינו, יש לנו את הכלים להתמודד אתו בצורה מיטבית ואנו מצפים לסיים אותו בקרוב.

לעוד פרטים ומשימות שאנו מבצעים, נא לפנות למאגר ה GitHub.

8. נספחים

א. רשימת ספרות \ ביבליוגרפיה

Internet of Things (IoT): A vision, architectural elements, and future directions

<https://arxiv.org/ftp/arxiv/papers/1207/1207.0203.pdf>

IoT gateway: Bridging wireless sensor networks into internet of things

<http://yzhang.org/courses/20172018tcpip/papers2read/0301.pdf>

The Internet of Things (IoT): Applications, investments, and challenges for enterprises

https://s3.amazonaws.com/academia.edu.documents/41516575/The_internet_of_things.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1510592709&Signature=wdPAewiQ%2FCUyKUI8MLk8gcmo7tA%3D&response-content-disposition=inline%3B%20filename%3DThe_Internet_of_Things_IoT_Applications.pdf

IoT Standards & Protocols Guide

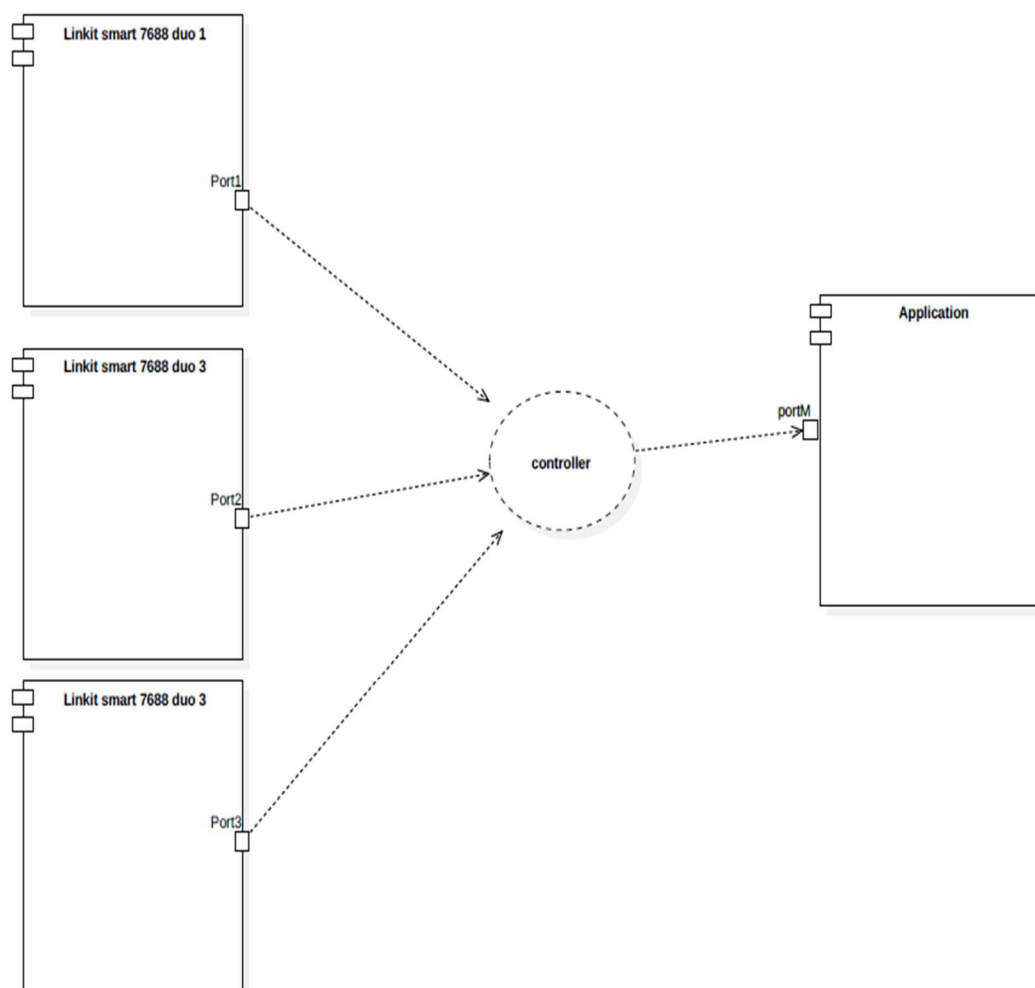
<https://www.postscapes.com/internet-of-things-protocols>

ב. תרשימים וטבלאות

היות וזה נושא מחקרי, לא עומדים לרשותנו כרגע כל הידע והמידע שנצטרך להשלמת הפרויקט ולכן רוב התרשימים שביצענו הן ברמת ההשערה לגבי מה אנחנו מבינים שהמערכת תהיה, תתנהג ותיבנה על ידנו.

תרשים רכיבים – components

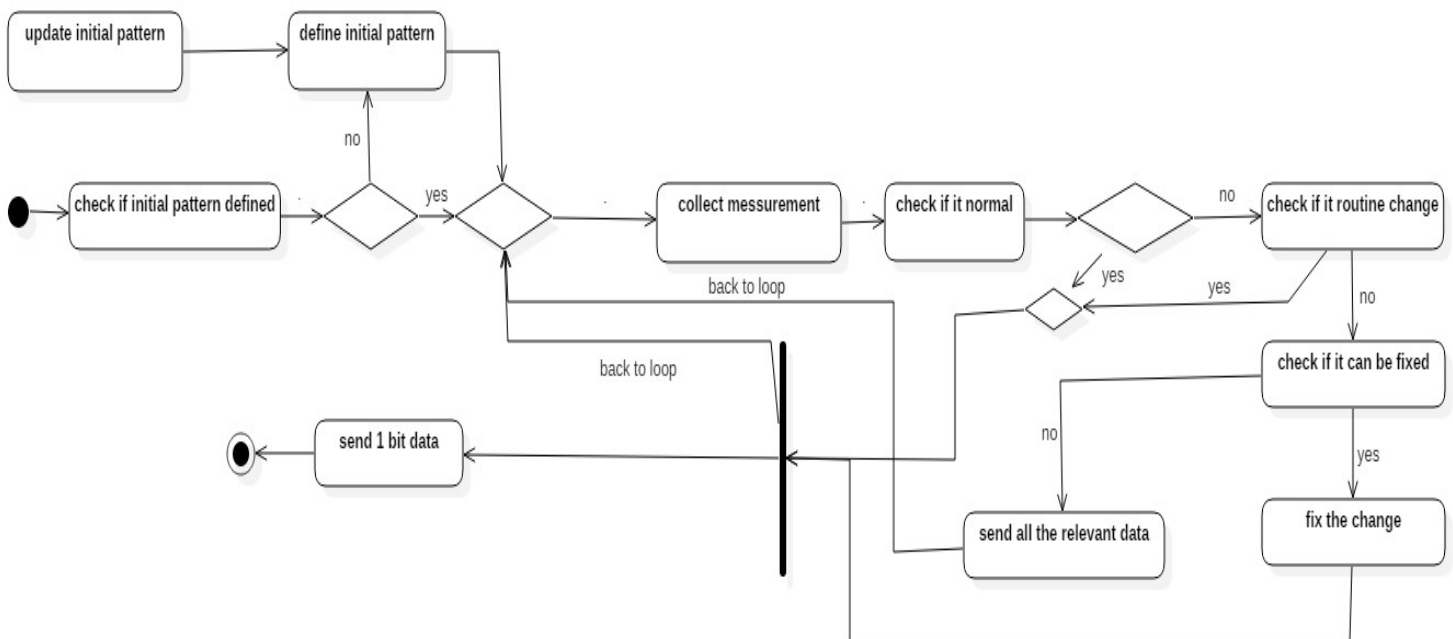
המערכת הכללית תורכב מרשת של חיישנים שתעביר את המידע לשבבי ה IOT , משם יבוצעו תהליכי הפרוטוקול ובמקרים חריגים יועברו הנתונים למרכז הבקרה דרך צינור מידע אשר יעבור במתווך מסוים שיקשר ביניהם .



תרשים פעילות – Activity Diagram

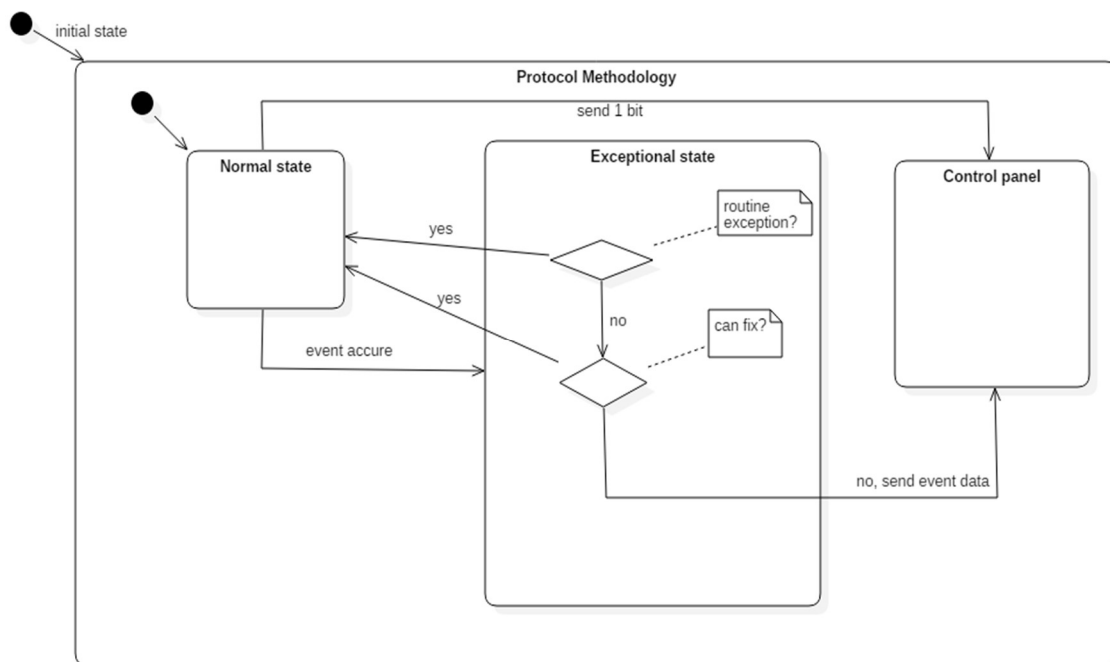
בתרשים הבא ניתן לראות את התנהלות הפרוטוקול בעת פעילות המערכת. באופן התחלתי הפרוטוקול בודק אם קיימת דגימה ראשונית (כמעין ברירת מחדל) עליה הפרוטוקול יסתמך שזה מצב שגרתי ותקין, אם לא הוגדר דגימה שכזו, הפרוטוקול מתחיל במדידות ומגדיר. לאחר מכן, הפרוטוקול נכנס ללולאה תמידית בה הוא מבצע את הפעולות הבאות כל כמה זמן מוגדר מראש, האלגוריתם לוקח דגימות בזמן אמת ומשווה עם הדגימה הראשונית אם בוצע שינוי כלשהו בנתונים. אם לא בוצע שינוי המערכת תשדר 1 ביט כדגל לזה שהמערכת במצב תקין ותחזור לתחילת הלולאה. אם כן נתפס שינוי בנתונים, הפרוטוקול יבדוק האם השינוי הוא שגרתי (כמו למשל בדוגמה עם חיישן האור והשמש שמתחלשת בהדרגה), במקרה וכן הפרוטוקול יחזור לתחילת הלולאה וכמובן ישדר 1 ביט. אם השינוי לא שגרתי, האלגוריתם יבדוק האם ניתן לתקן את השינוי, אם כן הוא יתקן באמצעות כללים שהוגדרו מראש, אם אין ברשותו לתקן, הפרוטוקול ישלח את הנתונים שנקלטו בצורה מסודרת וברורה עם פרטים רלוונטיים למקרה שקרה. באופן דיפולטיבי, כל כמה זמן האלגוריתם יעדכן את הדגימה הראשונית על מנת למזער שגיאות נתונים ופערי סטיות לאורך זמן.

<<precondition>> if checking time past



תרשים מצבים - State Chart

בתרשים הבא נתאר את המערכת במצבי הפעילות שלה. המערכת בברירת מחדל תפעל במצב תקין, במצב זה כדי גם להעיד שהמצב תקין תשלח ביט בודד באופן תמידי או לסירוגין (פעם בכמה דקות למשל), ביט זה כמובן לא יכול מידע נוסף כי פשוט אין צורך. במקרה של אירוע חריג אותו תזהה המערכת, המערכת תעבור למצב אירוע, תאסוף מידע אשר יגיע מהחיישן, המידע יאובחן ובשלב זה האלגוריתם יבדוק האם ביכולתו לתפעל את האירוע ולתקן את הבעיה או לבדוק האם האירוע לא חמור ולא מצריך מעבר של מידע למרכז הבקרה ולגורמים האנושיים. אם כן המערכת חוזרת לשגרה וממשיכה בשליחת ביט בודד, אם אין ביכולת האלגוריתם לתקן את הבעיה המערכת שולחת למרכז הבקרה (גורמים אנושיים) חבילת מידע עם פירוט נרחב על האירוע שקרה.



ג. תכנון הפרויקט

פגישת הכרות עם המנחה	9.10
בחירת נושא והגשה לרכז	10.10
הגשת נושא הפרויקט וטופס ההתנעה	22.10
פגישת עבודה עם המנחה על שלב ההצעה	30.10
הגשת דו"ח שלב ההצעה: סקר שוק, דרישות, מהות הפרויקט, סיכונים ותרשימים.	19.11
הגשת מוצר אלפא – מימוש אב טיפוס, מסמכים נוספים וארכיטקטורת המערכת.	סוף סמסטר א'
הגשת מוצר ביטא – תיכון ומימוש האלגוריתם, גרסת ביטא: ניסוי של שינוי מצב	אמצע סמסטר ב' (תאריך שיקבע ע"י הרכז)
שיפור האלגוריתם לאחר הפקת לקחים ודגשים מהמנחה	שבועיים לאחר הגשת מוצר ביטא
הגשת הפרויקט	סוף סמסטר ב'
פגישת הפקת לקחים וסיכום הפרויקט	תאריך שיקבע עם המנחה

כמו כן יקבעו פגישות רבות עם המנחה במהלך השנה על מנת לתכנן תכניות, משימות ועוד.

ד. טבלת סיכונים

#	הסיכון	חומרה	מענה אפשרי
1	לא נגיע לתוצאה שרצינו בסיום הפרויקט	גבוהה	ייעוץ נרחב עם המנחה, מעקב צמוד אחר ההתקדמות של הפרויקט בהתאם לציפיות מכל שלב, במקרה הגרוע תיאום ציפיות חדש.
2	סביבת העבודה וכלי פיתוח לא יהוו מענה ראוי למטרת הפרויקט	נמוכה	מחקר מעמיק על כלי הפיתוח, התאמת השבב לסביבת העבודה, במקרה הגרוע החלפת שבב.

3	חילוקי דעות עם המנחה לגבי מטרות ותוצאות רצויות	בינונית	עבודה משותפת ופגישות מרובות להכוונה ותיאום ציפיות עם המנחה.
4	האלגוריתם יביא לתוצאות לא צפויות	בינונית	מתן זמן רב לתיקון שגיאות וריג'קטים, בדיקות מרובות במגוון מקרי קצה ועוד.
5	האלגוריתם לא יזהה אירוע חריג שיעיד על שינוי מצב	גבוהה	תיקון שגיאות, כתיבת האלגוריתם בצורה מודולרית ומסודרת שתאפשר חזרה לאחור מבלי לפגוע ביעילותו
6	מוגבלות זיכרון תפגע ביעילות האלגוריתם	בינונית	צמצום האלגוריתם, צמצום ספריות שפחות חשובות לצורכי המערכת, עבודה שוטפת על הקוד בדגש על יעילות מקום וזמן.

ה. טבלת דרישות

מס' דרישה	תיאור	סוג
1	סביבת העבודה תזהה את השבב ותצליח לכתוב את האלגוריתם על השבב	פונקציונאלי
2	השבב יצליח להפעיל את האלגוריתם ביעילות מקום מינימלית	פונקציונאלי
3	השבב ידע לעבוד במצב תקין ולזהות שינוי מצב.	פונקציונאלי
4	האלגוריתם יהיה מודולרי כך שניתן יהיה להוסיף אלמנטים נוספים בעתיד שישדרגו את האלגוריתם	תמיכה לעתיד
5	האלגוריתם יהיה יציב לבדיקות מרובות וימנע קריסות מערכת	יציבות
6	האלגוריתם לעבוד באופן כללי על מגוון חיישנים בעלי פונקציונאליות שונה	פונקציונאלי
7	האלגוריתם ידע לשדר נתונים למרכז הבקרה	פונקציונאלי