

## המחלקה להנדסת תוכנה

# פיתוח רשת IoT לזיהוי אנומליות אקוסטית

## ומתן חיווי לגורמים הרלוונטיים

חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת

טלי כהן: 206326050

חני גרבר: 206114688

יוני 2018

תמוז תשע"ח

## המחלקה להנדסת תוכנה

# פיתוח רשת IoT לזיהוי אנומליות אקוסטית ומתן חיווי לגורמים הרלוונטיים

חיבור זה מהווה חלק מהדרישות לקבלת

תואר ראשון בהנדסה

מאת

טלי כהן: 206326050

חני גרבר: 206114688

מנחה אקדמי: ד"ר גיא לשם

אישור:

תאריך:

רכז הפרויקטים: מר אסף שפנייר

אישור:

תאריך:

## תקציר

כאשר מעוניינים לרשת סביבה אורבנית לצורכי אבטחה באמצעי חיווי אקוסטיים, אשר יכולתם לחוש, לעבד ולשדר מידע גבוהה, מהירה ומדויקת, קשה מאוד ליישם זאת משום העלות הגבוהה, הן מבחינה כלכלית והן מבחינת סיבוכיות. פרויקט זה הינו פיתוח מערכת חישה באמצעות התקן IoT אשר מבצעת הקלטת אודיו ועיבוד מידע. או במילים פשוטות יותר, בפרויקט זה "נאזין" לסביבה אורבנית, על מנת לחקור ולנתח את האירועים המתרחשים בה, כבסיס למערכת אבטחה ונבצע זאת באמצעות מערכת מבוססת רשת IoT.

IoT (Internet of Things – "האינטרנט של הדברים") - זוהי 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. [1] התקני IoT הם רכיבים אשר באמצעותם ניתן לתקשר באמצעות Wifi, GSM, Bluetooth וכו', אשר עליהם תרוץ המערכת וישמשו לתקשורת.

כל התקן צמוד לחיישן בהתאם למטרתו, אשר יבצע מדידות ויעביר את המידע הגולמי להתקן לעיבוד. ההתקנים הם דלי משאבים בעלי יכולת לחוש, לעבד ולשדר מידע אקוסטי.

התקן IoT שממשש אותנו בפרויקט זה הינו Raspberry pi, זהו מחשב זעיר שעלותו זולה מאוד, בגודל פחות מכרטיס אשראי שפיתחו בקרן ראספברי פאי. בשונה ממחשב רגיל יש לו יציאות וכניסות פלט/קלט כך שאפשר להתקין עליו חיישנים למיניהם ולהפעיל איתו התקנים חיצוניים.

באמצעות שימוש ופיתוח על מכשיר כזה תיפתר בעיית העלות הכלכלית והפונקציונאלית.

הפרויקט מתחלק לשלושה שלבים:

1. הכנה לסימולציה - התאמת התקן דל משאבים בעל יכולת לחוש, לעבד ולשדר מידע אקוסטי.
2. שלב הסימולציה:

- הקלטה בסביבה אורבנית - חצר בית ספר, בבסיס לתבנית אקוסטית נורמלית.
  - הקלטת אנומאליות אקוסטית בסביבה אורבנית.
3. שלב האנליזה - פיתוח אלגוריתם לזיהוי אנומאליות אקוסטית.

## הצהרה

העבודה נעשתה בהנחיית ד"ר גיא לשם במחלקה להנדסת תוכנה, "עזריאלי"  
– המכללה האקדמית להנדסה ירושלים.

העבודה התבצעה בזוג, משום שלפרויקט מחקרי זה נדרשה חקירה  
משמעותית ומקיפה המצריכה יותר מכוח אדם אחד.

החיבור מציג את עבודתנו האישית ומהווה חלק מהדרישות לקבלת תואר ראשון  
בהנדסה.

## תודות

ראשית נודה לבורא עולם שהביאנו עד הלום.

תודה מיוחדת למנחה הפרויקט ד"ר גיא לשם על העזרה הרבה  
ההכונה, העצות והרעיונות והכל מתוך שיתוף פעולה, סבלנות וסובלנות.

תודה רבה למשפחותינו האהובות שעזרו, עודדו, האמינו ותמכו בנו לאורך כל  
הדרך.

ותודה לחברותינו שהתעניינו ותרמו את השכלתם במהלך הפרויקט.

**על הכול תודה !**

## תוכן עניינים

3.....	תקציר
5.....	הצהרה
6.....	תודות
8.....	מילון מונחים
9.....	1. תיאור מסגרת הפרויקט
10.....	2. תיאור הבעיה
11.....	3. תיאור הפתרון
21.....	פירוט אלגוריתם עיבוד המידע
23.....	4. תיאור המערכת
27.....	5. תיאור בדיקות
27.....	5.1 בדיקות שמישות תוכנה
27.....	5.2 בדיקות פונקציונליות
27.....	5.3 בדיקות מערכת
27.....	5.4 בדיקות תאימות
27.....	5.5 בדיקות תחזוקה (Maintainability)
27.....	5.6 ניתוח תוצאות
28.....	6. ספרות
29.....	7. מסקנות
30.....	8. רשימת מקורות
31.....	9. מערכות ניהול הפרויקט
32.....	10. נספחים
32.....	10.1 Use Case של המערכת
32.....	10.2 תרשים זרימה – תיאור שלושת התהליכים העיקריים במערכת
33.....	10.3 תרשים זרימת אלגוריתם עיבוד המידע

## מילון מונחים

**דגימה:** המרת אות זמן רציף לתוך אות זמן בדיד. תהליך הדגימה יוצר נתוני אותות דיגיטליים במרווחי זמן בציר הזמן. תדירות דגימה מקובלת היא שניה.

**מיקרופון קליטת קולות:** ממיר גלי קול (תנודות זרם האוויר) לאות חשמלי (ספרתי או אנלוגי) קליטת הקולות מתבצעת דרך המיקרופון והוא ממיר לאותות, ואז מבצעים על הנתונים עיבוד ספרתי או אנלוגי.

**עיבוד אותות קול:** עיבוד אותות קול או עיבוד קול הוא עיבוד או שינוי מכוון של אותות קול או שמע בצורה ספרתית או אנלוגית המתבצע לרוב על ידי התקנים אלקטרוניים, ומערכות מחשב, אשר קולטים את הקול באמצעות מיקרופון, ממירים לתחום הספרתי ומבצעים על הנתונים עיבוד אותות ספרתי או עיבוד אותות אנלוגי, מאחסנים את הפלט, דוחסים או משדרים ופולטים אותו לאחר העיבוד על ידי רמקולים.

**FFT - (Fast Fourier Transform):** המרת הצליל מתחום הזמן לתחום התדרים. התמרת פורייה מאפשרת ניתוח של הקלטה ובידוד התדרים המרכיבים אותה. היא בעצם מאפשרת לאתר בתוך פונקציה רכיבים מחזוריים. הפלט של ניתוח פורייה מציג את התדרים החזקים ביותר לאורך זמן המזהים בקלט.

**IoT:** האינטרנט של הדברים (Internet of Things) 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". [2]

**Raspberry Pi:** מיקרו פרוססור (Microprocessor) בגודל של פחות מכרטיס אשראי אשר מסוגל לבצע מגוון רחב של פעולות כמו שמחשב רגיל מסוגל לבצע, בנוסף לניהול משימות ייעודיות לתכנות פיזי (Physical computing).

**Threads:** תהליכון או פתיל ריצה או נים הוא מושג במדעי המחשב המשמש במערכות הפעלה כדי לתאר הקשר ריצה במרחב כתובות. מערכות הפעלה מודרניות מאפשרות לנהל במסגרת ריצה של תהליך (Process) מספר תהליכונים הרצים במקביל במרחב כתובות אחד. במערכות אלו כל תהליך חדש מתחיל את ביצועו באמצעות 'תהליכון ראשי' אשר עשוי בהמשך ליצור תהליכונים נוספים. מנגנון הריצה באמצעות תהליכונים מאפשר לספק למשתמש במערכת ההפעלה מהירות תגובה ורציפות פעולה כאשר התהליך (יישום) מבצע כמה משימות במקביל.



## 1. תיאור מסגרת הפרויקט

במסגרת הפרויקט נדרשנו לחקור ולפתח אלגוריתמים אקוסטיים לזיהוי אנומליות לעיבוד דל משאבים ולתקשורת מבוססת מערכת IoT מבוזזת. תחילה ניתחנו את דרישות העיבוד הנדרשות לאלגוריתמים האקוסטיים והגדרת התקני העיבוד הדל. במהלך המחקר על אודיו ומרכיביו למדנו כיצד מנתחים קובץ אודיו ואלו נתונים ומידע אפשר לחלץ ממנו, על מנת שעל כל המידע הנאסף ניתן יהיה להפעיל אלגוריתם למידה חישובית.

בהמשך המחקר הסתבר כי נזדקק להשתמש בעוצמת חישוב גבוהה ובפונקציות פיתון "כבדות" להקלטת אודיו וניתוח של המידע.

בשלב הבא, תרנו אחר התקן מתאים לדרישות המחקר כך שיהיה בעל יכולת לחוש, לעבד ולשדר מידע. שידור המידע יבוצע באמצעות חיבור wifi בין ההתקן הרגיל להתקן מסטר שמחובר בסופו של התהליך לענן. לשם כך, בדקנו את ההתקן **LinkIt Smart 7688 Duo**, אך ניסויים שבצענו הראו כי ההתקן אינו מתאים לעבודה חישובית בעוצמה גבוהה, ולכן התקדמנו ל- **Raspberry pi** היכול להריץ פונקציות פיתון המבצעות הקלטה ועיבוד אודיו. התקן זה הוא מחשב זעיר שעלותו זולה, המבוסס על מערכת הפעלה של לינוקס, בגודל פחות מכרטיס אשראי שפותחו בקרן ראספברי פאי בבריטניה. בשונה ממחשב רגיל יש לו יציאות וכניסות פלט/קלט כך שאפשר להתקין עליו חיישנים למיניהם ולהפעיל איתו התקנים חיצוניים. [3]

להתקן חיברנו מיקרופון חיצוני לקליטת אודיו, והקלטנו סביבה אורבנית בפורמט wav בבסיס למחקר. ביצענו ניתוח של עיבוד אותות ע"י פונקציות פיתון.

התוצאה של הניתוח היא: אין שינוי/אנומליה ולכן נשלח מידע מינימלי למרכז נתונים, או יש שינוי/אנומליה ולכן נשלח את קובץ האודיו לעיבוד נוסף ברמה גבוהה יותר.

הניתוח כלל 3 אלמנטים:

א. מציאת האמפליטודה של הקול - ומעל ערך סף שנקבע, תזוזה אנומליה.

ב. ניתוח פורייה של המידע (Fast Fourier Transform).

ג. חיפוש תבניות אודיו והשוואה מול תבנית סטנדרטית או תבנית "בעייתית" (Sound Pattern Recognition with Python).

## 2. תיאור הבעיה

על מנת לעקוב אחר אירועים המתרחשים באזורים אורבניים, ולאכוף מקרים מסוימים הדורשים התייחסות ותשומת לב מגורמים רלוונטיים, יש צורך לקלוט מהסביבה האורבנית את האירועים המתרחשים בה ולנתח אותם ע"פ פרמטרים מסוימים הנקבעים ע"י הגורמים הרלוונטיים המעוניינים במידע.

רישות סביבה אורבנית במצלמות אבטחה יקר מאד הן מבחינת עלות כלכלית של ההתקנים והן מבחינת ניתוח המידע הנקלט בהם ושמירתו/שידורו. אפשרות אחרת לרישות סביבה אורבנית יכול להתבצע ע"י ביזור מיקרופונים ש"יאזינו" למתרחש בעיר ויהיו מחוברים למחשב אשר יעבד את המידע הנקלט וישדר אותו לגורמים הרלוונטיים במקרה הצורך.

### 2.1 עיקרי הבעיה

1. עלותם הגבוהה של ההתקנים המסוגלים לחוש מידע לעבדו ולשדרו
2. סיבוכיות זמן הריצה של האלגוריתמים

### 2.2 דרישות

פיתוח רשת IoT לזיהוי אנומליות אקוסטיות ומתן חיווי לגורמים הרלוונטיים דרישה זו כוללת בתוכה את הדרישות הבאות:

1. לימוד ומחקר המונחים הקשורים לאקוסטיקה ולאודיו היכולים לייעל ולקדם את המחקר.
2. חקירת התקנים היכולים להתאים למטרה, לימוד מרכיביהם ותכונותיהם ואופן פעולתם.
3. ניסוי בפועל של התקנת חיישנים על ההתקן הנבחר והשימוש בהם.
4. פיתוח אלגוריתם אקוסטי בהסתמך על המידע שרכשנו אודות אודיו ואקוסטיקה.

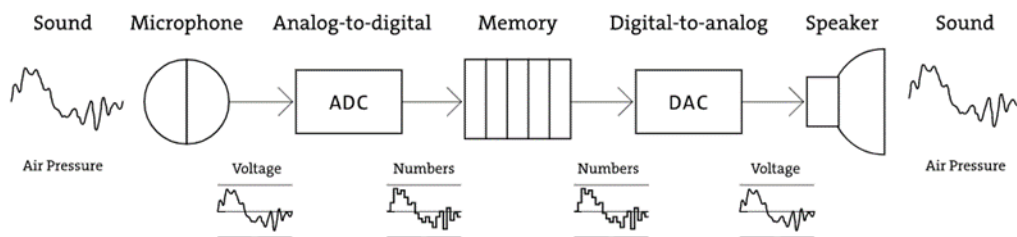
### 2.3 הבעיה מבחינת הנדסת תכנה

הבעיה המרכזית היא פיתוח יחידות קצה המסוגלות לחוש מידע בדרך זולה, לעבד אותו בסיבוכיות נמוכה ולשדר אותו לגורמים רלוונטיים.

- בחירת התקן זול אשר יכול לעבד את המידע המתקבל מהחיישן בדרך זולה ומהירה
- פיתוח אלגוריתם יעיל ומהיר לניתוח ועיבוד המידע

### 3. תיאור הפתרון

התחלנו את הפרויקט בשלב של מחקר ולמידה על כל נושא האודיו לתחומיו וניתוחו; בתרשים הבא מופיע התהליך שעובר על קול, החל מאז שהוא נקלט במיקרופון, דרך עיבודו במחשב ולבסוף – השמעתו דרך הרמקול. [4]



במחקר הגענו גם לתחום עיבודי הקול שבו ניתן למנות גם סינון רעשים שמתבצע כך: ניתן להסתכל על כל אות בציר הזמן, כלומר עוצמה כתלות בזמן, וכן ניתן להסתכל עליו בציר התדר: עוצמה כתלות בתדר מסוים. כך למעשה נוח לראות כי אותות שמע מורכבים מכמה תדרים. ישנו תחום תדרים צר אשר אוזן האדם מסוגלת לשמוע. תדרים קיצוניים בתוך תחום זה, למשל צפצוף בתדר גבוה, עלולים להישמע כרעשים. לצורך כך ניתן לבנות מסננים ספרתיים או אנלוגיים, חלקם אפילו משתמשים רק בדגימות של האות, כך שאלה מעבירים (מגבירים) את התדרים שאותם אנו מעוניינים לשמוע, ואינם מעבירים (מנחיתים) את התדרים הלא רצויים.

ישנם 4 סוגים בסיסיים של מסנני תדרי אודיו:

**LPF - Low Pass Filter**, מסנן המעביר תדרים נמוכים בלבד ומנחית תדרים גבוהים. (לאחר סינון נשמע קול בס עמוק)

**HPF - High Pass Filter**, מסנן המעביר תדרים גבוהים בלבד ומנחית תדרים נמוכים. (לאחר סינון נשמע קול גבוה וצפצפני)

**BPF - Band Pass Filter**, מסנן המעביר רק תדרים מסוימים ואת היתר לא מעביר. אפשר למשל להיעזר בו כדי "להיפטר" ממידע בתדרים שאוזן האדם בין כה אינה שומעת ולכן שמירתם נחשבת "בזבוז".

**BSF - Band Stop Filter**, מסנן המנחית רק תחום תדרים מסוים ואת היתר מעביר. לדוגמה, אם הרעש מופיע בתדר לא קיצוני, הנשמע לאוזן, ונרצה "להיפטר" ממנו תוך ויתור על מידע חיוני שחלקו נמצא גם הוא בתדר זה, נשתמש במסנן זה. [5]

### פורמט אודיו מועדף

בסקירת פורמטים שונים הקיימים ומאפייניהם, אופן העבודה איתם וניתוחם בהנדסת תכנה, הגענו להחלטה שקובצי האודיו עליהם נעבוד ואותם ננתח בפרויקט שלנו יהיו בפורמט WAV של

מיקרוסופט ו-IBM - זהו קובץ מסוג Pulse Code Modulation, שמכיל מידע בתחילתו ובסופו. לרוב הוא דגום בתדירות של 44.1Khz וברזולוציה גבוהה מאוד - 1411Kbps.

הסיבות לבחירת פורמט זה:

- שיטה טובה לשמירת מידע והמידע לא עובר דחיסה. היתרון במידע לא דחוס הוא שהמידע יכול להיות מעובד בנקל על ידי מעבדים חלשים, מאחר והוא לא זקוק לפריסה מחדש לשם השמעת הצליל. כיום, כוח העיבוד של מחשב ממוצע מאפשר לעבוד עם קבצים מכווצים אלו.
- פורמט שאינו מאבד מידע, מה שמאפשר לצליל של קובץ WAV להיות העתק קרוב למקור השמע המקורי.
- אידיאלי לשימור הקלטות בארכיון, שבו שטח אחסון אינו מהווה בעיה.
- יכול לשמש אמצעי החלפה עבור פלטפורמות מחשב אחרות, כגון מקינטוש מה שמאפשר למפתחים להעביר באופן חופשי קבצי אודיו בין פלטפורמות שונות.
- יכול להיות מומר לסוגי קבצים אחרים כגון AAC, MP3 ו-AIFF [6]

## חילוץ תכונות מאודיו

קיימים שני שלבים במתודולוגיית החילוץ של תכונות אודיו: [7]

חילוץ תכונות לטווח קצר: פעולה זו מיושמת בפונקציה - `stFeatureExtraction()` של הקובץ `audioFeatureExtraction.py`. זה מחלק את האות הנקלט לתוך frames ומחשב מספר תכונות עבור כל מסגרת. תהליך זה מוביל לרצף של וקטורים המכילים תכונות עבור האות כולו.

חילוץ תכונות של אמצע טווח: במקרים רבים, האות מיוצג על ידי נתונים סטטיסטיים של רצף תכונות המופקות לטווח קצר כמו שתואר לעיל. הפונקציה `mtFeatureExtraction()` מקובץ `audioFeatureExtraction.py` מחלצת מספר סטטיסטיקות (למשל ממוצעים וסטיית תקן) על כל רצף תכונות קצרות טווח.

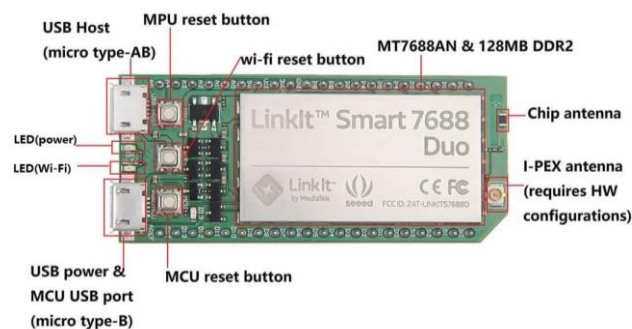
בטבלה הבאה מוצגת רשימת מאפייני אודיו עיקריים:

מזהה תכונה	שם התכונה	תיאור
2	Energy	סכום הריבועים של ערכי האות מנורמל ע"י אורך המסגרת בהתאמה
4	Spectral Centroid	מרכז הכובד של הספקטרום
34	Chroma Deviation	סטיית התקן של 12 מקדמי Chroma

## שלב בחירת ההתקן המתאים בתכונותיו לדרישות המחקר

תחילה בדקנו את המכשיר LinkIt Smart 7688 Duo מכיוון שהוא דל משאבים, בעל יכולת לחוש, לעבד ולשדר מידע ועלותו נמוכה במיוחד.

LinkIt Smart 7688 Duo הוא לוח פיתוח התואם את הסקריפטים של Arduino, המבוססים על ATmega32u4 ו- MT7688, OpenWrt Linux distribution. הלוח תוכנן במיוחד כדי לאפשר את אב הטיפוס של יישום עשיר ב- IoT לבית חכם או למשרד. בגלל שהוא תואם Arduino, הוא מאפשר להשתמש בתכונות שונות מ- Arduino ו- LinkIt Smart 7688 Duo, אשר מסייעים לבנות יישומים עשירים שמתבססים על סכמות שונות, חזקות ומהדרים של Arduino. הלוח מציע זיכרון ו- packet storage, כדי לאפשר עיבוד וידאו חזק. הפלטפורמה מציעה גם אפשרויות ליצירת יישומי התקנים בפיתוח, שפת Node.js ו- C. [8]



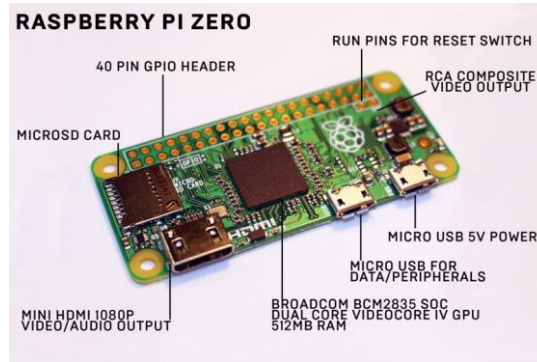
אך ניסויים שבצענו הראו כי ההתקן LinkIt smart 7688 Duo אינו מתאים לעבודה חישובית ולכן התקדמנו ל- Raspberry pi היכול להריץ פונקציות Python שמבצעות הקלטה ועיבוד אודיו.

הרספברי פיי (באנגלית: Raspberry pi) הוא מיקרו פרוססור (Microprocessor) בגודל של פחות מכרטיס אשראי אשר מסוגל לבצע מגוון רחב של פעולות כמו שמחשב רגיל מסוגל לבצע, בנוסף לניהול משימות ייעודיות לתכנות פיזי (Physical computing).

למחשב רספברי פיי 2 שימושים עיקריים:

שימוש במעבד מידע מרכזי לפרויקטים אלקטרוניים – בזכות היותו מחשב עם אפשרות לחבר רכיבים אלקטרוניים בקלות, הרספברי פיי מאפשר לפתח פרויקטים אלקטרוניים בקלות ועם רכיבים זולים אשר מסופקים על ידי יצרני ציוד לרספברי פיי בכל העולם. במילים פשוטות; פיתוח פרויקטים אלקטרוניים יותר בקלות ועם גישה מהירה לציוד.

שימוש במחשב הרספברי פיי כמחשב ייעודי – ישנם המון תוכנות (או יותר נכון ספריות) ייעודיות לרספברי פיי מה שהופך את המחשב הזהיר למחשב ייעודי עבור תוכנה מסוימת. [9]



השפה איתה עובדים עם ההתקן היא Python ולכן הפיתוח שלנו ייעשה גם הוא בשפה זו, בסביבת פיתוח Thonny. זוהי סביבת פיתוח ידידותית עבור Python, ובמיוחד נוח לעבוד איתה מכיוון שהיא מהירה ולא כבדה. כמו כן, קל מאוד לדאג איתה את הקוד ולכן היא נוחה לעבודת מחקר.

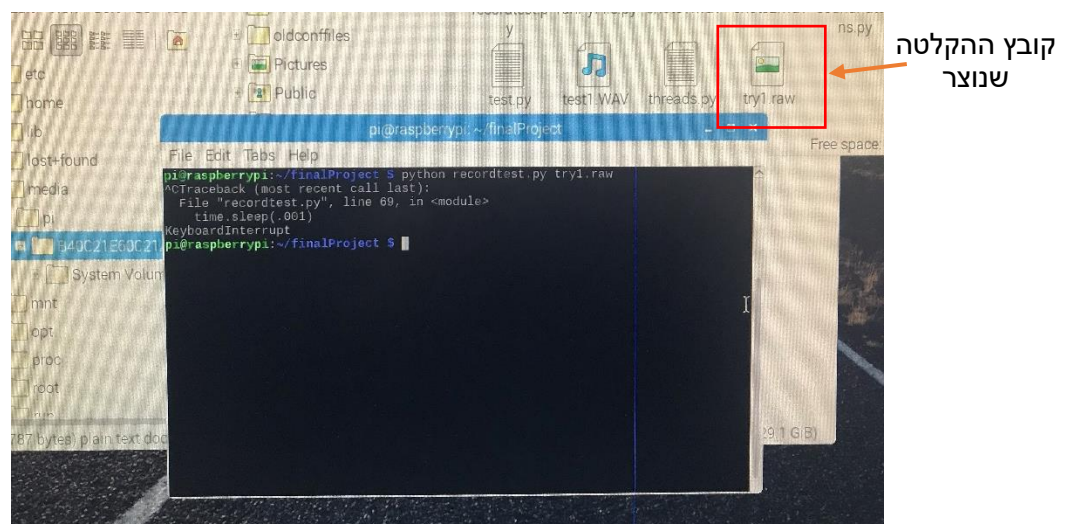
על המכשיר התקנו את הספריות הנדרשות לפיתוח עם פיתון.

התחלנו לחקור כיצד מקליטים מהמיקרופון ומקבלים קובץ אודיו לתוך המכשיר. לשם כך, מצאנו בספרית אלגוריתמי פיתון של גיטהאב [10] אלגוריתם לקליטת קול והרצנו אותו במכשיר.

האלגוריתם פותח את ההתקן המחובר למכשיר שנועד להקלטה, ומחזיר את המידע ב frames, כך שגודל כל frame הוא 2 בתים. הרצת האלגוריתם תתבצע בעזרת הפקודה הבאה להרצת קבצי פיתון דרך ה shell:

```
python recordtest.py out.raw # talk to the microphone
```

דוגמה להרצה דרך ה shell של הפקודה



קובץ האודיו של ההקלטה התקבל בפורמט raw. ובעזרת הפקודה הבאה דרך ה shell המרנו אותו לפורמט wav איתו בחרנו לעבוד.



```
sox -r 44100 -e signed -integer -b 16 -c 1 <nameOfTheRecordedFile.raw>
<nameOfTheNewFile.wav>
```

עקרונית, ההעדפה היא לעשות את ההקלטה וההמרה דרך האלגוריתם.

### את תהליך הסימולציה שלנו בחרנו לבצע על בית ספר פעיל בסביבה אורבנית.

תחילה יש לבדוק, האם הקלט שהתקבל הוא בזמן "קריטי" כפי שהוגדר ע"י הגורם הרלוונטי. זמן "קריטי" בסימולציה (הנוכחית) הינו זמן שבו לא אמורה להיות פעילות בשטח בית הספר, ולכן סטייה של רמת דציבלים מהממוצע שנקבע – תזוזה מיד כחריגה אנומאלית של דציבלים. לצורך כך, מתוך הפרמטרים שניתן לקבל מקובץ אודיו, תאריך ההקלטה מהווה פרמטר קריטי לניתוח המידע בסביבה זו:

**חודש** - האם זהו חודש פעיל בבית הספר (חודש שבו מתקיימים לימודים או זמני חופשה)

**יום** - באיזה יום בשבוע נקלט האודיו

**זמן** - באיזה זמן ביום נקלטו הרעשים (שעות הלימודים או שעת לילה / בוקר מוקדמת)

במקרה והקלט התקבל בזמן שאינו "קריטי" נמשיך בחקירתו באמצעות שלושת הפרמטרים של ניתוח מידע כפי שכתבנו בתחילת המסמך.

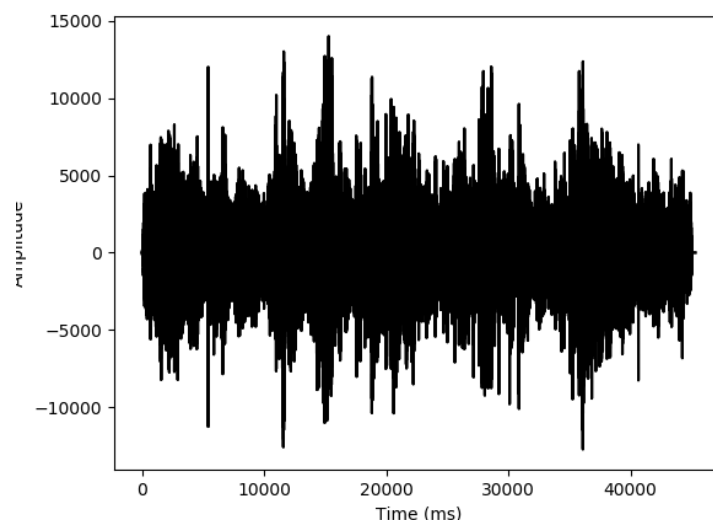
### ניתוח המידע המתקבל מקובץ אודיו

א. מציאת האמפליטודה של הקול - מעל ערך סף שנקבע ע"פ האמפליטודה, תזוזה אנומליה

על מנת לדעת את עוצמות הדציבלים (גובה הקולות) של הסביבה האורבנית בסימולציה, אותם ניתן להפיק מהמידע הנקלט, יש צורך בהצגת אמפליטודה.

דוגמא לאמפליטודה שיצרנו עבור הקלטה מסביבה אורבנית - חצר בית ספר בשעות פעילות

סטנדרטית:



באמפליטודה זו שהוגדרה בבסיס הסימולציה, ניתן לראות שבסביבה אורבנית נורמאלית גובה הקולות בממוצע הינו 8000 , וגובה הקולות המקסימאלי הוא כמעט 15000 .  
מנתונים אלו נסיק, שכשאר תתקבל הקלטה עם דציבלים הגבוהים מרמת הממוצע הנ"ל, הקולות ייחשבו כקולות **אנומאליים**.

### ב. ניתוח פורייה של המידע (Fast Fourier Transform)

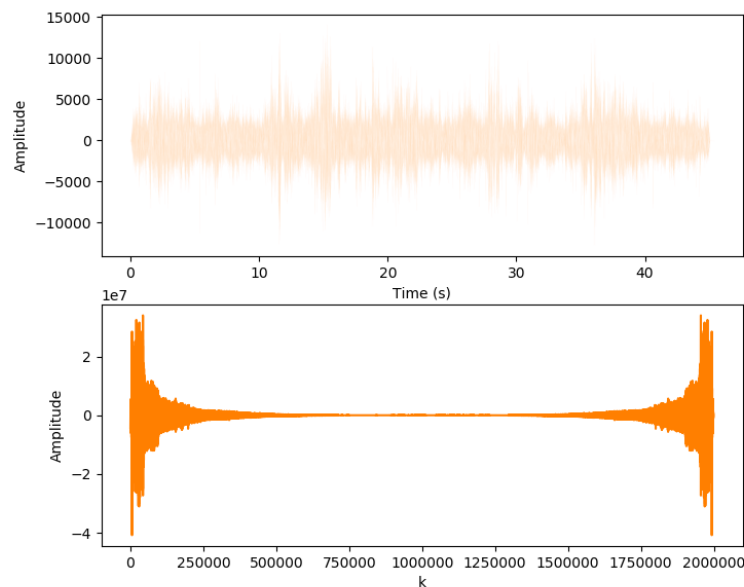
אחת הדרכים הנפוצות לביצוע ניתוח נתונים שלא נראים לעין ממבט ראשוני, היא להשתמש ב-Fast Fourier Transform (FFT). כדי להמיר את הצליל מתחום הזמן לתחום התדרים. התמרת פורייה מאפשרת ניתוח של הקלטה ובידוד התדרים המרכיבים אותה, ובעצם מאפשרת לאתר בתוך פונקציה רכיבים מחזוריים.

**הפלט של ניתוח פורייה מציג את התדרים החזקים ביותר לאורך זמן המזוהים בקלט (FFT) נותן עוצמה בתדר מסוים).**

בניתוח זה הקצב אינו חשוב כי מה אנו מעוניינים לדעת את ערכי האודיו. ערכי האודיו נמדדים לפי POWER - חוזק ניגון הצליל ו FREQUENCY – תדירות הצליל.

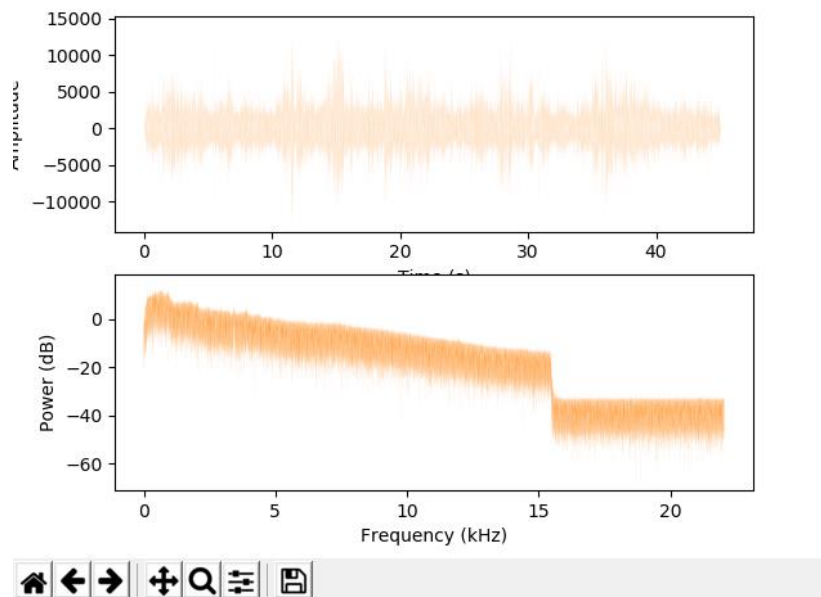
הגרפים הבאים התקבלו מהרצת אלגוריתם FFT על קובץ האודיו של הקלטת חצר בית הספר – בסיס הסימולציה:

איור 1





איור 2

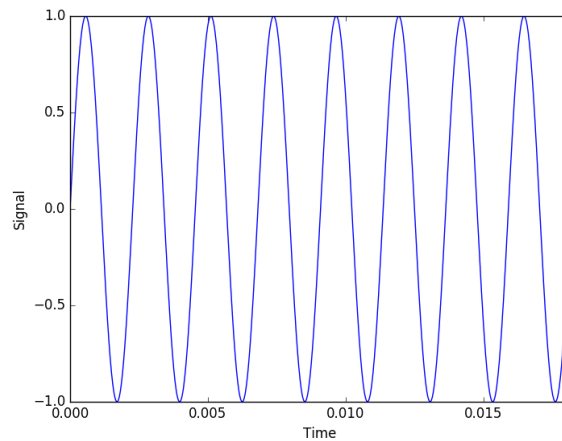


באיור 1 – הגרף העליון הוא האמפליטודה כתלות בזמן ללא הרצת אלגוריתם כלשהו, והתחתון הוא הפעלת ה-FFT, כאשר הגרף מחולק לשניים: צד שמאל בציור הוא החלק האמיתי של תוצאת האלגוריתם, וצד ימין הוא החלק המדומה.

באיור 2 – הציור העליון הוא שוב האמפליטודה כתלות בזמן, והתחתון הינו הפעלת FFT על החלק האמיתי בלבד של תוצאת ריצת האלגוריתם. מה שמתקבל הינו עוצמה כתלות בתדירות (ביחידות של קילו הרץ).

### ג. חיפוש תבניות אודיו והשוואה מול תבנית סטנדרטית או תבנית "בעייתית" (Sound Pattern Recognition with Python) [11]

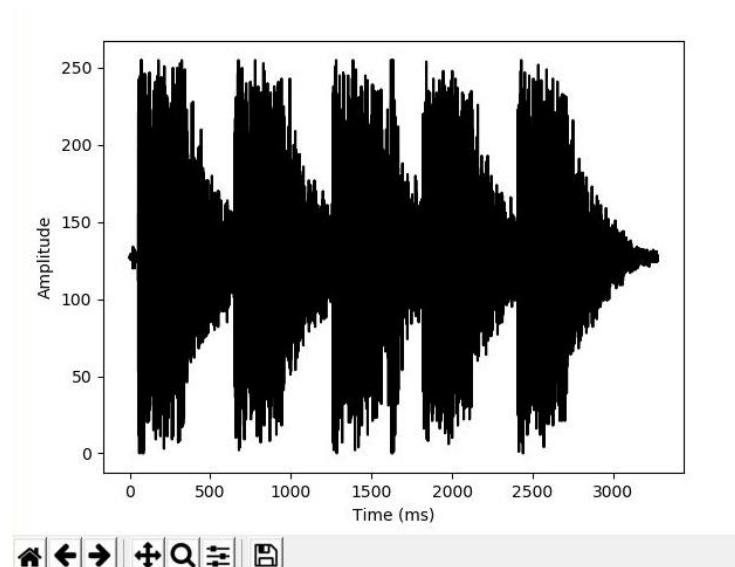
דרך נוספת ויעילה לניתוח ולמידה על קבצי אודיו היא חיפוש תבניות אודיו. עבור כל קובץ אודיו נמצא תבנית, ונשווה בין תבניות לצורך למידה ואיסוף נתונים. בעזרת האלגוריתם של השוואת תבניות ניתן בעצם לזהות דפוס צלילים. ניתן לחשוב על הצליל כפונקציה של לחץ לאורך זמן, כך שאפשר לייצג צליל בגרף, כמו בדוגמה הבאה:



בגרף מיוצג גל סינוס עם גודל תדר של 440 הרץ ושיעור דגימה של 44100 הרץ.

על מנת לשמור צליל במחשב אנו דוגמים אותו, תהליך הדגימה יתבצע כך: ניקח גודל של תדר אופטימאלי וניצור וקטור של גודל התדר, בכל שניה אנו לוקחים דגימות כאשר התדר הוא קצב הדגימה. עכשיו הצליל הוא מערך מספרי גדול. מה שמגדיר צליל מסוים הוא הזמן בין צליל אחד למשנהו.

ניקח לדוגמא את הגרף הבא שמייצג צליל יריות:



ע"פ הגרף ניתן לראות, שיש צורך למצוא את הערכים הגבוהים של הצלילים והמרחקים ביניהם במערך המידע שמתקבל מקובץ האודיו, על מנת לזהות דפוסי צלילים.

על מנת לזהות דפוסים מסוימים נזדקק לשני דברים:

1. ערך מינימלי שעבורו האות הוא גבוה.

לדוגמא עבור האיור: `min_val = 250`

2. זמן מיזוג - מרחק זמן אותו מגדירים עבור קול מסוים שייחשב לצליל אחד. (ייקבע לפי הערך המינימלי שהוגדר)

לדוגמא עבור האיור: `focus_size = 650`

על מנת לחשב שיא, נחפש ערכים במערך העולים על ערך המינימום, ובכל פעם שיימצא ערך כזה, כל ערך שאחריו בתוך זמן המיזוג יהווה ירייה בודדת.

ביצענו זאת כך:

```
while idx < len(data):
    if data[idx] > min_val:
        mean_idx = idx + focus_size // 2
        focuses.append(float(mean_idx)/data_size)
        if len(focuses) > 1:
            last_focus = focuses[-2]
            actual_focus = focuses[-1]
            distances.append(actual_focus - last_focus)
            idx+=focus_size
        else:
            idx+=1
    return distances
```

זמן התרחשות הירייה יהיה נקודת האמצע של זמן המיזוג. לבסוף נחשב את המרחקים בין כל נקודות זמני ההתרחשות וניצור וקטור של מרחקים, זו תהיה בעצם התבנית.

וקטור המרחקים:

```
1 print calc_distances('knock2.wav')
2 # Outputs
3 [0.1978195192433994, 0.18560357283593854, 0.19873899908052017, 0.22172599500853807]
```

כרגע יש לנו תבנית מרחקים מובנה, שניתן להשוות אליה תבניות אחרות. וזאת על מנת לזהות דפוס דומה/שונה.

מהשוואת התבניות נוכל ללמוד האם המדידה החדשה שקבלנו כעת, כבר קרתה בעבר. אם כן, נוכל לדעת מה בדיוק קרה, ואם לא הרי שיש מצב חדש שיש לבדוק אותו.

לאחר המחקר על דרכי ניתוח הקול השונות, התחלנו לפתח באמצעותן את האלגוריתם שירוצ על נתוני הסימולציה.

## מהלך האלגוריתם

באלגוריתם ישנם 3 תהליכים עיקריים:

1. הקלטה – תהליך המריץ תוכנית להקלטת אודיו, ומפיק את קובץ האודיו שעליו ניתן יהיה לעבוד.

2. עיבוד המידע – תהליך ניתוח המידע לפי הדרכים שנחקרו: פרמטרים לבדיקה, מידע מהאמפליטודה, מידע מהרצת FFT על הקובץ והשוואת תבניות אודיו.
  3. שידור - בעת זיהוי אנומליה אקוסטית - שליחת קובץ האודיו לגורמים הרלוונטיים.
- עקב דלות המשאבים של ההתקן איתו אנו עובדות, נחסוך במקום ונמחק כל הקלטה שגמרנו לעבד אותה.
- את 3 התהליכים ביצענו בעזרת **Threads** – תהליכונים / פתילים. השתמשנו בשיטה זו, שאופן מימושה מהווה יתרון עבור הפרויקט, משום שהפעלת מספר תהליכי Threads דומה להפעלת מספר תכניות שונות בו זמנית, אך עם היתרונות הבאים –
1. Threads מרובים בתוך תהליך חולקים את אותו שטח נתונים עם התהליך הראשי, ולכן יכולים לשתף מידע או לתקשר אחד עם השני בקלות רבה יותר מאשר אם היו תהליכים נפרדים.
  2. Threads נקראים לפעמים תהליכים קלי משקל, אינם דורשים הרבה זיכרון וזולים יותר מאשר תהליכים.
- ל-Thread יש התחלה, רצף ביצוע ומסקנה. ויש לו מצביע שעוקב אחר המקומות שבהם הוא נמצא כרגע.

#### מהלך ריצת האלגוריתם היוצר את התהליכונים:

```

3  import _thread
4  import time
5
6  # Define a recording thread
7  def recording():
8      import recordtest
9
10 # Define a processing thread
11 def soundProcessing(file):
12     import audioAnalysis
13
14 # Define a transmission thread
15 #def transmission():
16
17
18 # Create 3 threads as follows
19 try:
20     _thread.start_new_thread( recording )
21     _thread.start_new_thread( soundProcessing, 'test1.py' )
22     _thread.start_new_thread( soundProcessing, 'test1.py' )
23 except:
24     print ("Error: unable to start thread")
25
26 while 1:
27     pass

```

בקוד הנ"ל יצרנו עבור כל תהליך - פתיל, ובסה"כ יצרנו 3 פתילים. [תרשים בנספח 10.2]  
**Thread ראשון:** עבור ההקלטה - מריץ את קובץ ההקלטה recordetest.py ומפיק ממנו קובץ אודיו.  
**Thread שני:** הפלט של thread ההקלטה נשלח ל thread הבא המעבד את המידע הנקלט.

#### פירוט על תהליך עיבוד המידע

- הניתוח כולל 3 אלמנטים: [תרשים בנספח 10.3]
- א. מציאת האמפליטודה של הקול - ומעל ערך סף מזהה אנומליה.
  - ב. ניתוח פורייה של המידע.
  - ג. חיפוש תבניות אודיו והשוואה מול תבנית סטנדרטית או תבנית "בעייתית".

#### פירוט אלגוריתם עיבוד המידע

- בדיקת פרמטרים חיוניים
  - בעזרת הפונקציה ctime() מהספרייה time, מציאת תאריך ההקלטה, וחילוץ הפרמטרים שצריך לבדוק: חודש, יום, זמן.
  - שליחת הפרמטרים לפונקציות ולידציה:
  - בדיקה עבור החודש האם רלוונטי עבור הסביבה האורבנית הנוכחית.
  - בסימולציה הנוכחית, חודשים שאינם רלוונטיים עבור הסביבה - חצר בית ספר - הינם יולי ואוגוסט. אם החודש לא רלוונטי, הבדיקה תעבור להשוואת תבניות וכן תופסק בדיקת הפרמטרים.
  - בדיקה עבור היום, האם זהו יום שבת שאינו רלוונטי לסביבה האורבנית.
  - בדיקה עבור הזמן, האם זמן ההקלטה נמצא בטווח הזמנים בהם בית הספר פעיל.
- אם כל הפרמטרים נמצאו רלוונטיים - בדיקת האודיו תימשך.
- המרת הקובץ למערך נתונים ושרטוט הנתונים לאמפליטודה. מהאמפליטודה יוגדר ערך הסף אשר כל קול שעובר אותו יוגדר כאנומלי.
- ניתוח פורייה FFT על קובץ האודיו - התמרת פורייה לשם קבלת עוצמת הדיצילים בכל תדירות.

– השוואת תבניות אודיו:

לשם השוואה וניתוח, יצרנו תבנית של קול נורמאלי מהקלטה של סביבה אורבנית סטנדרטית (בסימולציה שלנו- בית ספר פעיל) ותבנית אנומאלית עבור הסביבה (עבור הסביבה שלנו – יריות בסביבת בית ספר פעיל).

מציאת תבנית עבור הקובץ והשוואת התבניות הבאות אליה:

1. תבנית אודיו נורמאלית עבור הסביבה הנוכחית השמורה באלגוריתם.
2. תבנית "בעייתית".

בדיקה האם קיימת חפיפה בין הנתונים המתקבלים מהשוואת התבניות, וקטלוג התבנית בהתאם לממצאים: בעייתי/לא בעייתי.

על המידע הנאסף באלגוריתם ניתן להפעיל אלגוריתם למידה חישובית.

התוצאה של ניתוח המידע

- אין שינוי / אנומליה ולכן נשלח מידע מינימלי למסטר (מרכז נתונים).
- יש שינוי ולכן הקובץ האודיו יישלח לעיבוד נוסף ברמה גבוהה יותר.

לאחר עיבוד המידע וחקירת הקובץ, תיערך בדיקה האם הקובץ מכיל קול אנומאלי עבור הפרמטרים שהוגדרו לסביבה הנוכחית. אם כן, נשמור את תבנית הקובץ כתבנית אנומאלית, שתשמש להשוואה ולימוד לתבניות הבאות שיגיעו בהמשך ריצת האלגוריתם.

**Thread שלישי:**

שידור המידע לגורם הרלוונטי

יחידת הקצה אותה מימשנו מהווה חלק מרשת התקני IoT החשים מידע אקוסטי, מעבדים אותו ומשדרים לגורם הרלוונטי. ברשת זו קיים מאסטר מרכזי שתפקידו לנהל את ההתקנים, מיקומם, לקבל מהם מידע אותו הם משדרים ולעבד אותו. הוא אחראי לשליחת המידע לגורמים הרלוונטיים עבורם המערכת נועדה. הקשר בין יחידת הקצה לבין המאסטר שמחובר בסוף התהליך לענן, נעשה באמצעות חיבור wifi.

יחידת הקצה שולחת בכל פרק זמן מסוים את חיווי למאסטר על פעילות תקינה. רק קבצי קול שזוהתה בהם אנומליה ע"י האלגוריתם ישודרו בשלימותם למאסטר.

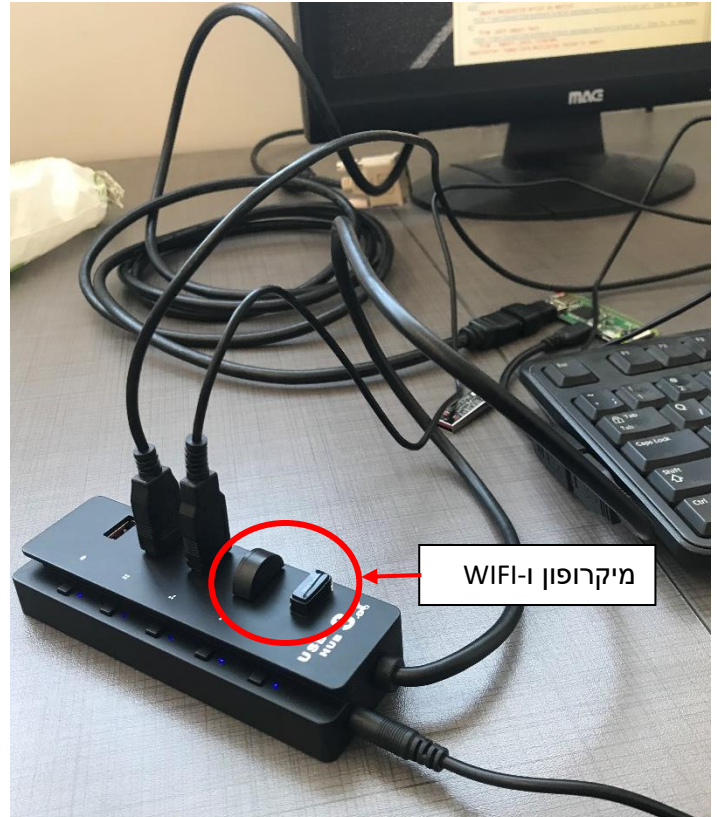
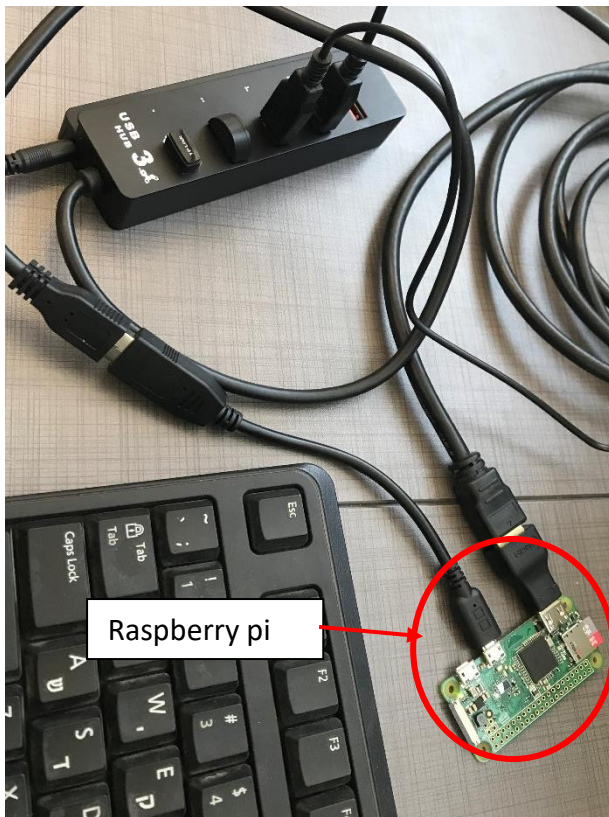
לסיכום, המאסטר מקבל מכל אחת מיחידות הקצה ברשת את הנתונים הבאים:

1. אות חיווי על פעילות תקינה של יחידת הקצה
2. קובץ אודיו המכיל אנומאליות אקוסטיות – במקרה שהתרחשה.



## 4. תיאור המערכת

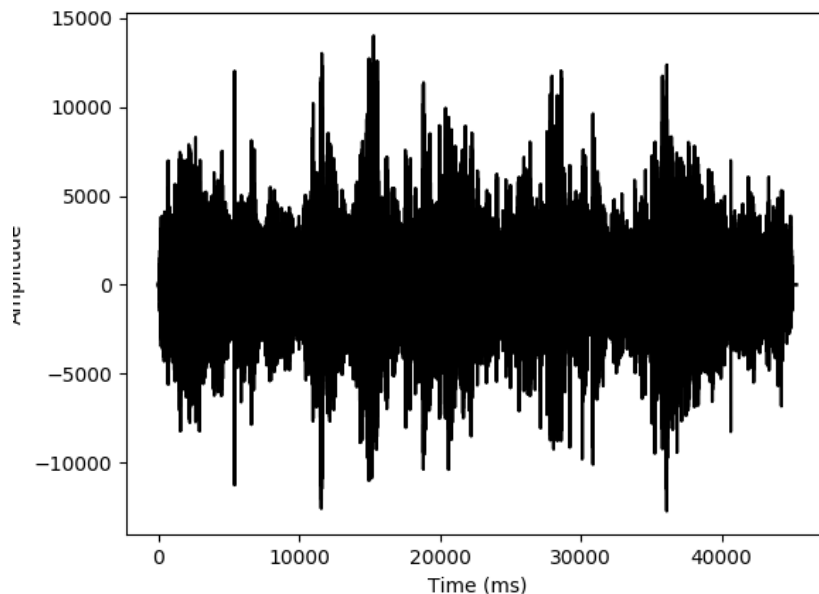
תמונות של מערכת ה-Raspberry pi כאשר מחוברים אליה: מסך, מקלדת, עכבר, מיקרופון להקלטה wifii 23



## דוגמא להרצת המערכת עבור בית ספר פעיל בסביבה אורבנית:

הצגת אמפליטודה עבור קול נורמאלי בסביבה אורבנית:

Figure 1

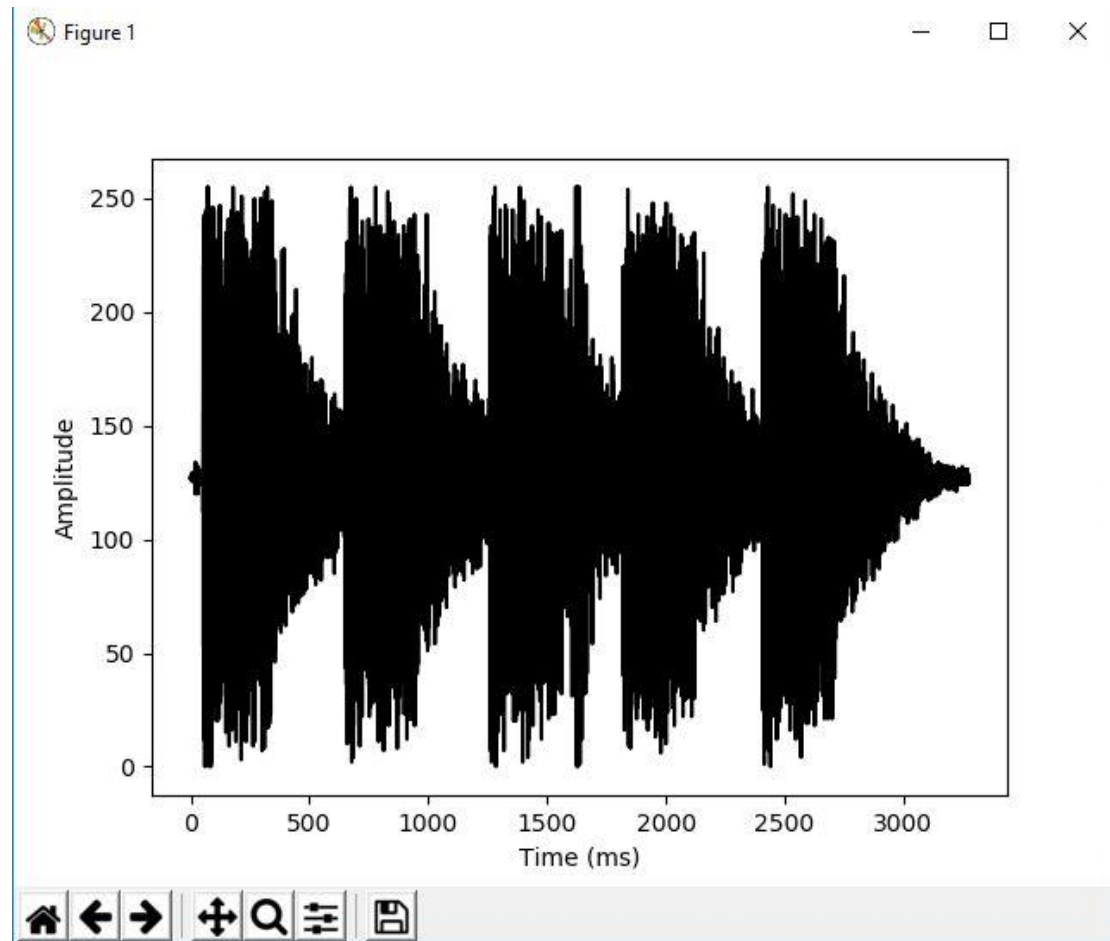


באמפליטודה זו שהוגדרה בבסיס הסימולציה, ניתן לראות שבסביבה אורבנית נורמאלית גובה הקולות בממוצע הינו 8000 , וגובה הקולות המקסימאלי הוא כמעט 15000 . מנתונים אלו נסיק, שכשאר תתקבל הקלטה עם דציבלים הגבוהים מרמת הממוצע הנ"ל, הקולות ייחשבו כקולות **אנומאליים**.

את התבנית של קובץ אודיו זה נשמור כ"תבנית סטנדרטית" עבור הסביבה בסימולציה.

לצורך בדיקה, ניקח למשל הקלטת צרור יריות בסביבת בית ספר פעיל. האמפליטודה של קובץ זה תיראה כך:

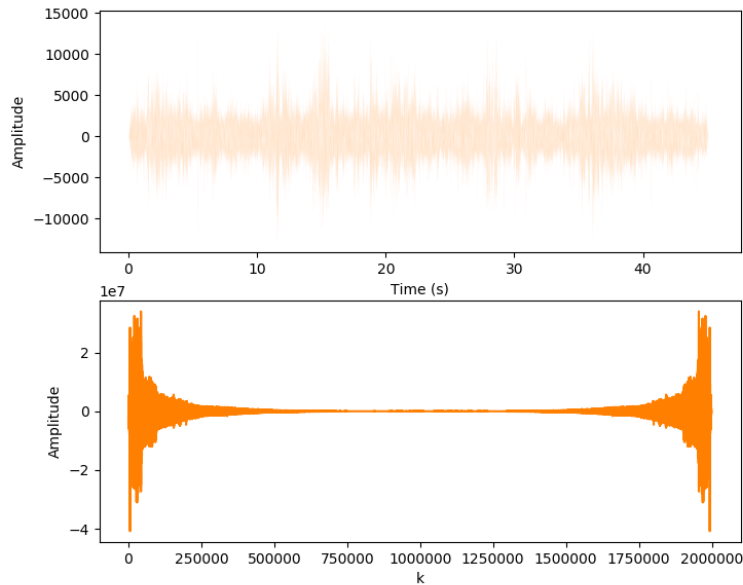




נוכל לראות שצירור היריות הזה נמצא בגובה דציבלים של 25000- מה שעובר בהרבה את גובה הקולות בסביבה הנורמאלית. -> מכך ניתן להסיק **אנומאליות**.

את התבנית של קובץ אודיו זה נשמור כ"תבנית בעייתית" עבור הסביבה בסימולציה.

על מנת לחלץ את הקולות החזקים ביותר הקיימים בקובץ האודיו נריץ **ניתוח פורייה** על המידע שהתקבל מהקלטת צרור היריות בסביבה האורבנית. הגרף המתקבל מניתוח זה יראה כך:



נחלץ ממנו את הקולות החזקים ביותר הנשמעים בקובץ.

**בחיפוש התבניות שנערוך**, נשווה כל תבנית של קובץ אודיו לתבנית ששמרנו כנורמאלית – "סטנדרטית", ולתבנית ששמרנו כאנומאלית – "בעייתית".

מהנתונים הנ"ל נוכל להסיק אנומאליות אקוסטיות עבור הקובץ שנבדק.

בזיהוי אנומאליות אקוסטיות **נשדר** את ההקלטה למאסטר האחראי על יחידת הקצה.

## 5. תיאור בדיקות

### 5.1 בדיקות שמישות תוכנה

בדיקות מקיפות של כל פונקציות המערכת ע"מ לוודא נכונות, מקרי קצה, מקרים חריגים וכו'.

### 5.2 בדיקות פונקציונליות

- בדיקת מבשיר Raspberry pi Zero W
  1. תקינות המכשיר
  2. האם המידע מעובד ומשודר
  3. האם המכשיר לא "נופל" בזמן הקלטות ממושכות ורצופות של אודיו
- בדיקות על אלגוריתמים לעיבוד אודיו שמומשו בפרויקט:
  1. אלגוריתם לחיפוש תבניות אודיו Sound Pattern Recognition with Python – הרצת האלגוריתם על תבניות ברורות עם מידע קצר, והשוואה עם מידע דומה בדיוק.
  2. ביצוע ניתוח פורייה של המידע (Fast Fourier Transform) – הרצת האלגוריתם על יחידה קטנה של מידע.

### 5.3 בדיקות מערכת

1. אינטגרציה נכונה של כל השלבים במערכת
2. האם המערכת מתפקדת בזמן אמת
3. האם המערכת אינה קורסת כשיש עומס של מידע

### 5.4 בדיקות תאימות

סנכרון בין wi-fi של המכשיר למחשב השרת

### 5.5 בדיקות תחזוקה (Maintainability)

- האם ניתן לעדכן או לתקן את התוכנה אחרי הוצאתה לאור
- האם הקוד כתוב בצורה פשוטה, ברורה ומתועדת

### 5.6 ניתוח תוצאות

בדיקות מקיפות וניתוח תוצאות האירוע; האם המידע התקבל באופן תקין, עובר עיבוד נכון במכשיר, ואם יש צורך – האם נשלח והגיע ליעדו בצורה מדויקת.

## 6. ספרות

קיים מאמר [12] העוסק במחקר שנערך ע"י סטודנטים מספרד, ובו הם ביצעו ניתוח סטטיסטי מרחבי של נתוני רעש עירוני מ WASN שנאסף על ידי מערכת IoT לעיר קטנה.

תקציר המאמר: המחקר נערך ע"פ הנחיית האיחוד האירופי 49/2002 והחוק הספרדי 37/2006 הקוראים לערים לפתח מפות רעש אסטרטגיות ותוכניות פעולה להערכת חשיפת הרעש וליצור נהלי הפחתת רעשים באזורים קריטיים. עם זאת, מיפוי רעש כרוך בהליכי מדידה יקרים ומסורבלים שיכולים להפוך לבעיה אמיתית בפועל. מאמר זה מתאר מערכת ניטור רעש מבוססת המבוססת על WASN (Wireless Acoustic Sensor Network) ויישום מתודולוגיה גיאוגרפית סטטיסטית לחיזוי סטטיסטי מרחבי-טמפורלי של רמות רעש באזורים פתוחים למחצה, כגון עיר טיפוסית, (Algemesí, ולנסיה, ספרד). מתודולוגיה זו מיושמת בחקר האבולוציה המרחבית בזמן זיהום הרעש. לשם כך, מודל סטטיסטי מרחבי פותח באמצעות מדידות זיהום רעש שהושגו על קבוצה של נקודות הממוקמות בכמה מיקומים אסטרטגיים. מודל הזמן הגיאומטרי-סטטיסטי מאפשר לאמוד רמות רעש ספציפיות ומאפיין את השונות המרחבית-טמפורלית של זיהום הרעש. התוצאות מראות כי המודל המפותח מספק קירוב טוב של המדידות המתקבלות בניסוי.

בהמשך המאמר הם מציינים את הכלים שבעזרתם ביצעו את המחקר: ...מטרת מסמך זה היא לנתח את ההתאמה של מערכת החומרה למדידת רעש, המבוססת על צמתים של Raspberry pi עם מיקרופון כחיישן אקוסטי ומערכת לאיסוף נתונים, בשילוב עם טכניקות גיאומטרי-סטטיסטיות החלות על בעיית מיפוי רעש בעיר קטנה בקהילת ולנסיה שבספרד.

במאמר זה, הסטודנטים ביצעו עבודה דומה; הם שתלו בכמה צמתים מכשירי Raspberry pi מבוססי IoT. אך מטרת המחקר שלהם הייתה ניטור הרעש בעיר, ולא זיהוי קולות חריגים / חשודים, על מנת להתריע לגורמים הרלוונטיים.

## 7. מסקנות

1. בפרויקט מחקרי עדיף להשקיע המון זמן לחקירת המוצר והמשאבים לפני שמתחילים לכתוב קוד ולפתח כדי שיהיה בסיס רחב עליו יהיה ניתן לעבוד.
2. בפרויקט מחקרי שבו צריך להתחיל הכול מאפס, כולל בחירת מכשיר והתאמתו לפרויקט עם טכנולוגיות שיפעלו כראוי, נדרש רב זמן להתאמת מכשיר ולהכרתו, כך שזמן רב אחרי תחילת התהליך עדיין לא נראות תוצאות בשטח. מה שיכול לגרום לתסכול מ"אי התקדמות" לכאורה בפרויקט.
3. יש לתעד ולפשט ככל הניתן את ההנחיות שמעביר המנחה, כדי שלא ייווצרו אח"כ בשעת העבודה חילוקי דעות ואי הבנות.
4. עבודה בזוג בפרויקט מחקרי, מקילה הן על חלוקת העבודה והן בחשיבה משותפת המגוונת ותורמת רבות לפיתוח ושדרוג הפרויקט.
5. למידה לעומק של טכנולוגיה חדשה ולא מוכרת – מצריכה זמן רב. לכן, חלוקת זמנים מפורטת ועבודה בצמוד ללו"ז, עם השארת "עודף" זמן לתקלות לא צפויות או הערכות שגויות, תאפשר עבודה מסודרת ויעילה לאורך הפרויקט.
6. עבודה מסודרת, תיעוד סדיר ומפורט של המפגשים והידע הנרכש משבחים את העבודה וממהרים את קצב העבודה

## 8. רשימת מקורות

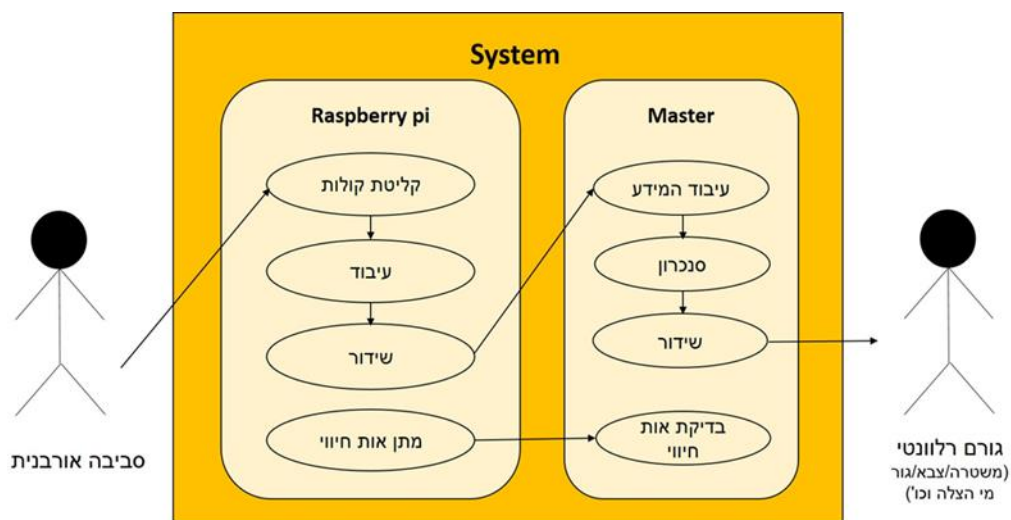
- [1] [https://he.wikipedia.org/wiki/%D7%94%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98\\_%D7%A9%D7%9C\\_%D7%94%D7%93%D7%91%D7%A8%D7%99%D7%9D](https://he.wikipedia.org/wiki/%D7%94%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98_%D7%A9%D7%9C_%D7%94%D7%93%D7%91%D7%A8%D7%99%D7%9D)
- [2] [https://he.wikipedia.org/wiki/%D7%94%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98\\_%D7%A9%D7%9C\\_%D7%94%D7%93%D7%91%D7%A8%D7%99%D7%9D](https://he.wikipedia.org/wiki/%D7%94%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98_%D7%A9%D7%9C_%D7%94%D7%93%D7%91%D7%A8%D7%99%D7%9D)
- [3] [https://he.wikipedia.org/wiki/%D7%A8%D7%90%D7%A1%D7%A4%D7%91%D7%A8%D7%99\\_%D7%A4%D7%90%D7%99](https://he.wikipedia.org/wiki/%D7%A8%D7%90%D7%A1%D7%A4%D7%91%D7%A8%D7%99_%D7%A4%D7%90%D7%99)
- [4] <https://processing.org/tutorials/sound>
- [5] [https://he.wikipedia.org/wiki/%D7%A2%D7%99%D7%91%D7%95%D7%93\\_%D7%90%D7%95%D7%AA%D7%95%D7%AA\\_%D7%A7%D7%95%D7%9C](https://he.wikipedia.org/wiki/%D7%A2%D7%99%D7%91%D7%95%D7%93_%D7%90%D7%95%D7%AA%D7%95%D7%AA_%D7%A7%D7%95%D7%9C)
- [6] <https://www.remsoftware.com/glossary/wav-file>
- [7] <https://github.com/tyiannak/pyAudioAnalysis/wiki/3.-Feature-Extraction>
- [8] <https://www.seeedstudio.com/LinkIt-Smart-7688-Duo-p-2574.html>
- [9] <https://piitel.co.il/about-raspberry-pi>
- [10] <https://github.com/larsimmisch/pyalsaaudio/blob/master/recordtest.py>
- [11] <https://medium.com/@almeidneto/sound-pattern-recognition-with-python-9aff69edce5d>
- [12] <http://www.mdpi.com/2076-3417/6/12/380/htm>

## 9. מערכות ניהול הפרויקט

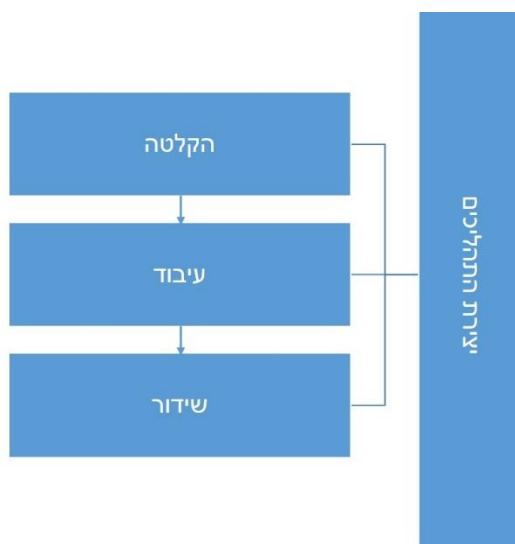
#	מערכת	מיקום
1	מאגר קוד	<a href="https://github.com/TaliYa/Final-Project">https://github.com/TaliYa/Final-Project</a>
2	יומן	<a href="https://trello.com/b/OjTevadw/welcome-board">https://trello.com/b/OjTevadw/welcome-board</a>
3	סרטון סופי	<a href="https://drive.google.com/open?id=1NW4s_flH9FWK81AyZcERgjhvwMvWZkUg">https://drive.google.com/open?id=1NW4s_flH9FWK81AyZcERgjhvwMvWZkUg</a>
4	סרטון אב טיפוס	<a href="https://drive.google.com/open?id=1eNDD5rH3UHjyqxgRmq80Y1TVmFjv8Wvx">https://drive.google.com/open?id=1eNDD5rH3UHjyqxgRmq80Y1TVmFjv8Wvx</a>

## 10. נספחים

### Use Case 10.1 של המערכת

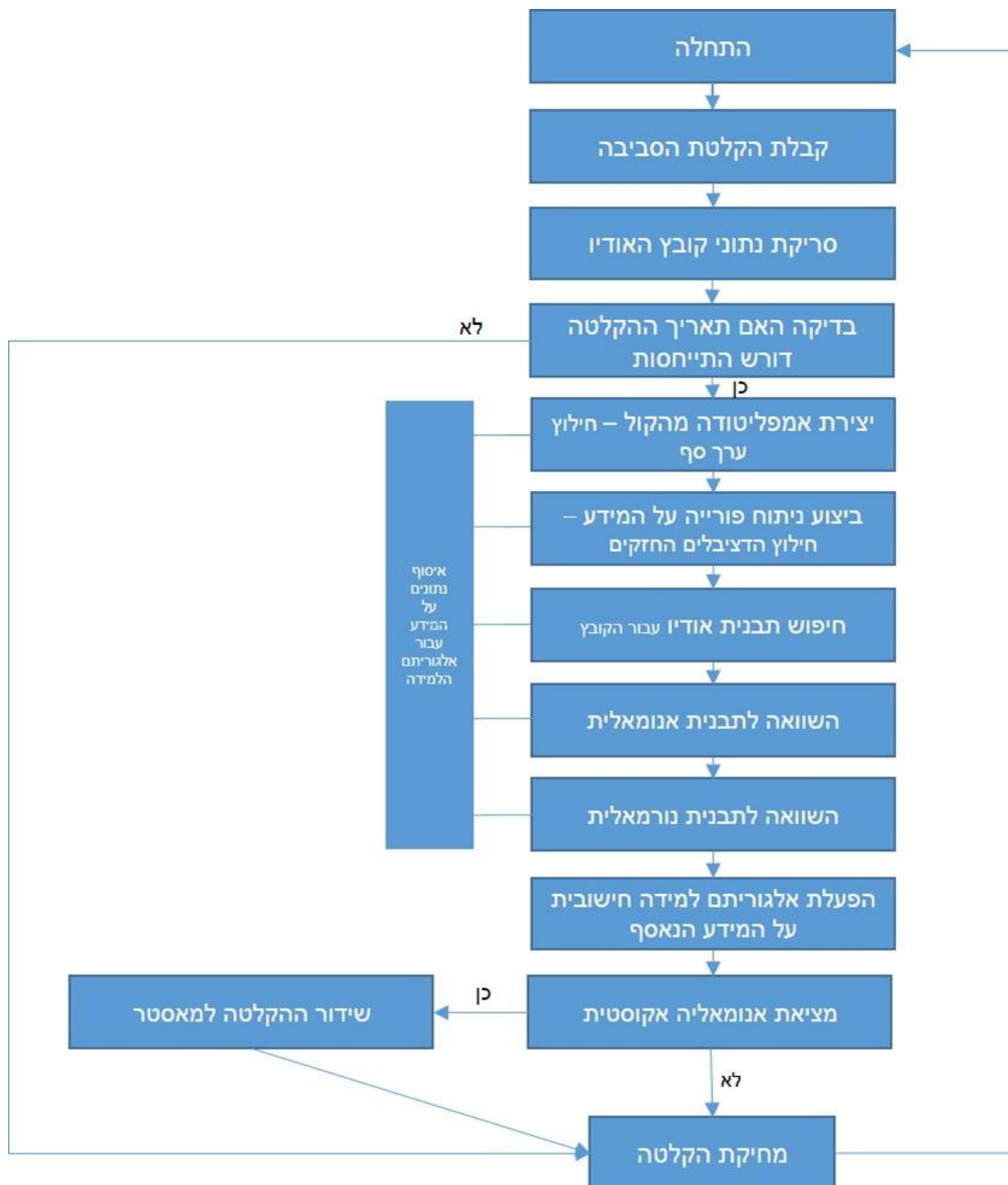


### 10.2 תרשים זרימה – תיאור שלושת התהליכים העיקריים במערכת





### 10.3 תרשים זרימת אלגוריתם עיבוד המידע



## Abstract

When you want an urban environment for security by means of acoustic signals, whose ability to sense, process and transmit high information is fast and accurate, it is very difficult to implement this because of the high cost, both economically and in terms of complexity.

This project is the development of a sensing system using an IoT device that performs audio recording and data processing. Or, more simply, in this project we will "listen" to an urban environment in order to investigate and analyze the events taking place in it, as a basis for a security system and we will do so using an Internet-Things-based system (IoT) "Things" embedded in electronics, software and sensors that enable advanced communications between the objects and the ability to collect and exchange information. This network is expected to lead to automation in many areas. IoT devices are components through which Wifi, GSM, Bluetooth, and so on can be communicated, which the system will run and be used for communication. Each device is attached to the sensor according to its use, which will measure and transfer the raw information to the device for processing. The devices are low resources capable of sensing, processing and transmitting acoustic information.

The IoT device we use in this project is the Raspberry pi, a tiny computer that is very cheap, less than a credit card developed by the Raspberry Pie Foundation. Unlike a conventional computer, it has outputs and input / output inputs so that it can be installed with different sensors and external devices.

By using and developing such a device, the problem of economic and functional cost will be solved.

The project is divided into three stages:

1. Preparation for simulation - adapting a low-resource device capable of sensing, processing and transmitting acoustic information.
2. Simulation phase:
  - Recording in an urban setting - a school yard, as a basis for a normal acoustic pattern.
  - Record acoustic anomalies in urban environments.
  - Analysis phase - development of an algorithm to identify acoustic anomalies.

## **Software Engineering Department**

# **Development of the IoT network to identify acoustic anomalies and provide an indication of the relevant factors**

**By**

**Tali Cohen & Chani Garber**

**Academic supervisor:**

**Dr. Guy Leshem**

**June 2018**

## **Software Engineering Department**

# **Development of the IoT network to identify acoustic anomalies and provide an indication of the relevant factors**

**By**

**Tali Cohen & Chani Garber**

**June 2018**