עבודת גמר במדעי המחשב

נושא: טריאנגולציה של מכשירים בתוך שטח בנוי

א. פרטים על התלמיד:

שם התלמיד: גלעד אוזן.

מספר זהות: 325914117.

שם בית הספר: ישיבת צביה אשקלון.

סמל בית הספר: 641183.

מספר יחידות במדעי המחשב: 10 .

ב. פרטים על המדריכים:

שם המדריכ/ה: גל בר-און.

תואר אקדמי: B.sc/M.sc.

תחומי התמחות: הנדסת תוכנה ומדעי המחשב.

עיסוק: מורה ומרצה הנדסת תוכנה ומדעי המחשב.

שם המוסד בו תתבצע העבודה: המרכז למצויינות עירונית אשקלון.

דואל: gal.baron23@gmail.com

מספר טלפון נייד: 0523499554

איכון מכשירים בעזרת בלותות'

# מבוא

## תיאור הפרוייקט

הפרוייקט יעסוק באיכון מכשירים סלולארים באמצעות טכנולוגיית ה Bluetooth בתוך שטח בנוי. החיפוש ייעשה במאצעות מספר מכשירי Raspberry Pi המפוזרים בשטח הנתון. כל מכשיר ימדוד את עוצמת האות אל המכשיר המיועד וימיר את חוזק האות למרחק, אותו ישלח לPi רביעי שיחשב את המיקום המדוייק של המכשיר בעזרת טריאלטרציה ויציג את המיקום על מסך ראשי.

## מטרת הפרוייקט

בזמן החקירה על רעיון הפרויקט וכיצד ייראה התוצר הסופי, היה ידוע לי שהמטרה תכלול שימוש במערכות ניווט או מיפוי, התוצר הסופי יכול לעבוד במקומות כגון חנויות או קניונים, או אפילו מגרשי חנייה לרכבים בעלי תמיכה בבלוטות', המערכת שלי לא מחייבת את הלקוח להתחבר אל שרת מסוים או לאשר חיבור, הדרישה היחידה היא השארת מתג הבלוטות' במכשיר דולק, המערכת יכולה לזהות מיקומים ולשדר אותם בשידור חי, כך שיהיה ניתן לראות את תנועת הנמצאים בשטח, לאסוף מידע על כך ולעבוד בהתאם. (המערכת גם יכולה לתפקד כמערכת למציאת המכשיר אם אבד ונמצא בשטח).  
בעזרת כל הנתונים שהמערכת מוציאה, ניתן לבנות מערכת מיפוי ולראות על מפה מדומת GPS ספציפית עבור השטח (כלומר לראות מפה של פנים המבנה) ולאסוף סטטיסטיקות שונות על תנועות הנמצאים באזור, כך למשל במבנה של קניון ניתן לדעת מה הם המקומות היותר שכיחים בקרב הקונים.

הפרויקט הוא פרויקט שבחלקו מחקרי, לפני שהתחלתי לעבוד על הקוד והפרויקט המעשי היה צורך בחקירה של הפרוטוקולים השונים שאין לי ניסיון איתם, כדוגמת בלוטות', חקרתי ברשת על אופן הפעולה שלו, השוני בינו לבין פרוטוקולים דומים כמו Wi-Fiותת פרוטוקולים בתוכו כגון RFCOMM וRSSI על מנת לדעת האם הוא הפרוטוקול הנכון לפרויקט, בחרתי בנושא זה מכיוון שהוא כולל בתוכו המון תחומים, שימוש במכשירים, מחקר רב, מתמטיקה ולבסוף גם יש פלט על המסך שמראה את תוצאת הפרויקט (מיקומים של המכשירים).

## תוכן

[מבוא 2](#_Toc60078854)

[תיאור הפרוייקט 2](#_Toc60078855)

[תוכן 2](#_Toc60078856)

[מבנה הפרויקט 3](#_Toc60078857)

[סקירת ספרות 5](#_Toc60078858)

[Bluetooth (בלוטות') 5](#_Toc60078859)

[תקן חיבור מקביל: Wi-Fi 7](#_Toc60078860)

[בלוטות' לעומת תקן חיבור מקביל: Wi-Fi 8](#_Toc60078861)

[הפעלת בלוטות' על מכונה וירטואלית 9](#_Toc60078862)

[מציאת כתובות ה-IP של צמתי הקצה 12](#_Toc60078863)

[מערכות הפעלה מבוססות לינוקס 13](#_Toc60078864)

[טריאלטרציה וטריאנגולציה 15](#_Toc60078865)

[RSSI (Received signal strength indicator) 18](#_Toc60078866)

[raspberry pi 19](#_Toc60078867)

[הסברים על הקודים 19](#_Toc60078868)

[קוד1: client.py 19](#_Toc60078869)

[קוד 2: Adapted\_Server.py 23](#_Toc60078870)

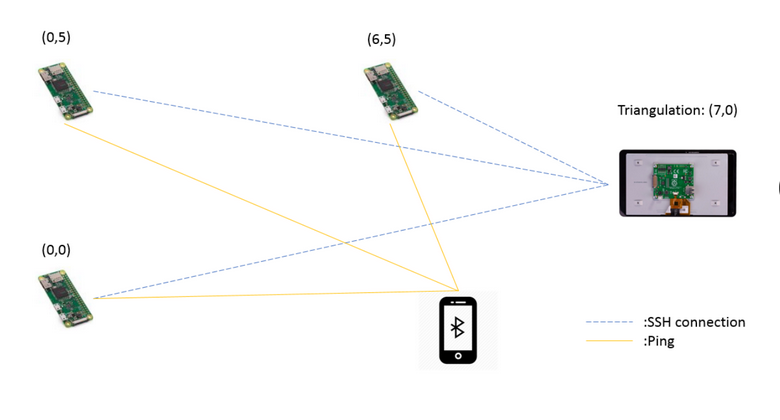
[קוד 3: Trialteration.py 26](#_Toc60078871)

[ייצוג גרפי בסיסי של מכשירי הבלוטות' שמיקומם נמצאו 29](#_Toc60078872)

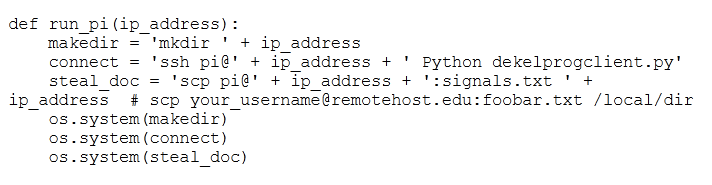
[רפלקציה 31](#_Toc60078873)

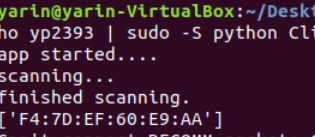
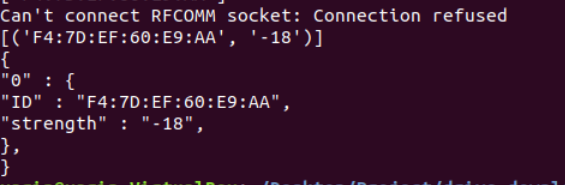
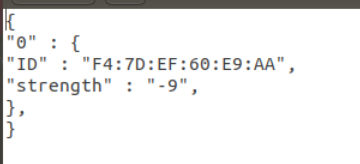
[ביבילוגרפיה 32](#_Toc60078874)

## מבנה הפרויקט

הפרויקט בנוי משלושה יחידות עיקריות:  
1. 3 מכשירי RPi בתור צמתי קצה (הכנס תמונה) ומכשיר RPi בתור צומת ראשית  
2. מכשירי בלוטוס שאת מיקומם לבסוף נרצה לדעת (מכשירים כגון סמארטפונים, מחשבים, טלוויזיות וכו')  
3. אלגוריתם הטריאלטרציה.  
תחילה, נרצה להרים "רשת" פרטית של צמתים, שבהם 3 צמתי קצה המסודרים בתור שטח וצומת ראשית במרחק עם מסך שיכול להראות פלט.   
  
עבור כך, נחבר את הצמתים לאותה רשת תקשורת על מנת שיוכלו לתקשר אחד עם השני.  
בכל צומת קצה יימצא קובץ בשם end\_code.py, ובצומת הראשית יימצא הקובץ main\_code.py.  
על מנת להריץ את התוצר הסופי נרצה להפעיל את main\_code.   
חלקי המערכת:

הצומת הראשית (הקלט: כתובות IP של צמתי הקצה, הפלט: הצגת המיקומים של מכשירי הבלוטות' בשטח)

כאשר נריץ אותו הפעולה run\_end תופעל, הפעולה מפעילה 3 פקודות משורת הפקודה שקיימות בלינוקס:  
1. Mkdir, הפקודה תיצור תיקייה בשם המשוייך לצומת הקצה הנוכחית (התוכנית משייכת את עצמה 3 פעמים, כל אחד לצומת אחרת על פי כתובת IP.  
2. SSH, הפקודה תתחבר בעזרת כתובת הIP של הצומת המשויכת בפרוטוקול SSH ותפעיל דרכו את הקובץ **end\_code.py**  
3. SCP, גם היא פקודה המבוססת על פרוטוקול SSH, הפקודה תבצע התחברות אל צומת הקצה המשויכת ותעתיק ממנו את הקובץ הכולל את הנתונים הסופיים שאסף (יפורט בהמשך על הקוד של צומת הקצה)  
  
על מנת לחסוך בזמן, אשתמש בספריית multiprocessing על מנת להפעיל את הפעולה run\_end עבור 3 הצמתים במקביל.  
  
(Multiprocessing- ספרייה הקיימת בPython, בדומה לספריית הThreading, מולטי-פרוססינג היא חבילה המאפשרת יצירת תהליכים במחשב, ותמיכה בהרצת מספר תהליכים דומים בבת-אחת, ניתן לתת לכל אובייקט של תהליך פעולה שהוא יבצע, בנוסף לספרייה אפשרויות סנכרון רבות ושיתוף הזיכרון ביניהם, פיצול הרצת פעולות לתהליכים שונים מקצר משמעותית את הזמן שלוקח למחשב להריץ את הקוד, זאת מכיוון שבמקום להריץ את הקודים אחד אחרי השני, יווצרו כשלושה תהליכים שכל אחד יבצע את הקוד התואם.)  
  
לאחר שכל הנתונים נאספו וסודרו אוטומטית בצומת הראשית, הקוד יבצע איטרציה מסודרת על הנתונים ועבור כל נתון (נתון = מכשיר בלוטוס שנקלטו נתוניו) תופעל הפעולה power\_to\_distance, הפעולה מתרגמת את חוזק האות שצומת הקצה אספה ומתרגמת אותה למספר שנוכל לקרוא לו גם "מרחק", כלומר הוא מדיד ביחידות אורך.  
לבסוף, יבוצע אלגוריתם טריאלטרציה וממנו נעבד לבסוף נקודות מיקום משוערות של המכשיר בלוטוס בתוך השטח הבנוי.  
צמתי הקצה  
התפקיד של צמתי הקצה הוא לאסוף נתונים ממכשירי בלוטוס, כל אחד ייקח מאותם מכשירים אך עקב השינוי במיקומים יהיה שינוי יחסי גם בנתונים שהתקבלו.

התוכנית מתחילה עם הפעולה scan\_to\_file, הפעולה תפעיל את פקודת לינוקס הנקראת hcitool, הפקודה משתמשת בפרוטוקול בלוטוס ובעצם מפעילה סריקת בלוטוס רגילה, נשמור את התוצאות בקובץ טקסט פשוט, לכל מכשיר בלוטוס יש id נוכחי שגם אותו אנו מקבלים ושומרים לקובץ וכך משייכים כל קובץ ומזהים אותם.  
    
לאחר מכן הפקודה get\_signal תופעל, הפקודה עוברת על קובץ מכשירי הבלוטוס, ומפעילה את פקודת הלינוקס rfcomm connect, הפקודה מנסה לשלוח נתונים קטנים על מנת לבדוק אם יש אפשרות חיבור, כמובן שמכשיר הבלוטוס יידחה את החיבור אבל אנחנו בכל מקרה מקבלים חיבור רגעי שניתן להשתמש בו (כלומר, אם שלחנו בקשה, אנחנו בעצם בכל זאת יצרנו תקשורת עם המכשיר, בין אם אישר זאת או לא) לאחר מכן תופעל פקודת rssi עבור אותו מכשיר, ונקבל ערך איכות חיבור אל אותו מכשיר.  
  
  
  
לאחר שיש לנו את ערך זה אנחנו יכולים לשמור את הנתונים בצורה מסודרת לקובץ טקסט, הקוד מעבד גם את הקובץ כך שיהיה מותאם לשיטת הקריאה שבקוד הצומת הראשית.  
 

מבנה הנתונים- הנתונים נשמרים בקובץ מסוג txt, הכתיבה נעשית מתוך הקוד באמצעות קוד הפייתון, נעשה שימוש בפקודות הקשורות למחלקת File בפייתון.

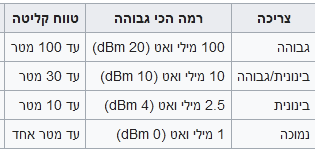
# סקירת ספרות



רקע תיאורטי, מושגים ותכונות הפרויקט

## Bluetooth (בלוטות')

בלותות' היא טכנולוגיית שידור וקליטה אלחוטית בקצב והספק נמוכים, המאפשרת חיבור של טווח קצר בין מכשירים. טכנולוגיית הבלוטות' מאפשרת להקים רשת בין התקנים תומכים ולבצע העברה של נתונים. טווחי הקליטה של בלוטוס מתחלקים לרמות על פי צריכת אנרגיה, ומובעים בעזרת מילי ואט (שניתן לתרגם ליחידות dBm)



בלותות' עובדת בתדרים של 2.4-2.4835Ghz. כדי שלא יהיו בעיות בשידור השידור נעשה בטכנולוגית SSFH (Spread Spectrum Frequency Hopping) שבה התדר משתנה כ-1600 פעמים בשנייה באקראיות וכך אין סיכוי לתדר להתערבב עם שיחה אחרת.

|  |  |
| --- | --- |
| יתרונות | חסרונות |
| הספק חשמל נמוך מאוד העוזר לחיסכון בצריכת הסוללה של המכשיר | אפשרות להפרעות וסטיות בטווחי הקליטה מצד מכשירים אלקטרוניים, או עצמים אחרים העומדים בדרך |
| יכולות הצפנה ואבטחה ברמה סבירה מספיק על מנת להבטיח חיבור מוגן ובטוח | רוחב פס נמוך המקשה על העברת נתונים גדולים (לא רלוונטי לפרויקט מכיוון שנעביר קבצי txt פשוטים). |
| יכולת חיבור אוטומטי בין מכשירים בנוסף לאל-חוטיות. | טווח חיבור קצר |
| נפוצה מאוד. הפרוטוקול מיושם כמעט בכל מכשיר נייד מהעשור האחרון (מחשבים ניידים, סמארטפונים ואף טלוויזיות ושלטים). |  |

לטכנולוגיית הבלוטות' יתרונות על רשתות אלחוטיות אחרות, אך עם זאת גם חסרונות.

ארכיטקטורת רשת בלוטות'  
בלותות' עובדת בשיטת עבד מאסטר(master-slave), שיטה שבה המאסטר אחראי על השעון ועל הסנכרון בין הרשת שנוצרה עם כל המשרתים. כל מכשיר בלותות' יכול להיות חלק מכמה רשתות בלותות' נפרדות ואפילו להיות מאסטר ברשת אחת, וברשת אחרת משרת. מצב זה ממש יעזור לנו בארכיטקטורת העבודה.

אבטחה  
בלותות', כחיבור על חוטי חשוף יותר לפרצות אבטחה מאשר חיבורים חוטיים למיניהם, מכיוון שהמידע עובר באוויר וגורם צד שלישי יכול לתפוס אותו. בלותות' היא תקשורת מוצפנת אך בנוסף יש לטכנולוגיית הבלותות' מגוון מנגנוני אבטחה כמו האפשרות להגביל את התקשורת למספר מסויים של מכשירים. רמת אבטחה זו מספיק חזקה בשביל המשתמש הסטנדרטי כולל אותנו.

## תקן חיבור מקביל: Wi-Fi

WiFi (Wireless Fidelity) גם היא טכנולוגיית שידור וקליטה אלחוטית המשתמשת בגלי מיקרו ומשמשת בעיקר כחיבור אלחוטי לרשת האינטרנט במכשירים ניידים כגון טלפון, לפטופ וכו'.

טכנולוגיית הWiFi משתמשת בתקני 802.11 IEEE שזה בעצם שם נרדף לWiFi. תקני 802.11 מאפשרים פריסת רשת תקשורת אלחוטית למרחק של כמה עשרות מטרים. התקן הספציפי הוא בעצם זה שקובע את התדר והמהירות.

הקמת רשת Wi-Fi:  
הדרישות לכך הן:

1. נתב (ראוטר) המחובר בכבל: לרשת האינטרנט
2. חומת אש (יכול להיות מוגדרת בראוטר, או על ידי ספק האינטרנט, או במערכת גם ההפעלה.)
3. נקודת גישה אלחוטית.
4. מכשיר בעל מתאם Wi-Fi שבעזרתו נתחבר.

יתרונות וחסרונות:

|  |  |
| --- | --- |
| יתרונות | חסרונות |
| טווח חיבור בינוני עד ארוך גם כאשר יש הפרעות פיזיות | צריכת חשמל גבוהה(בעייתי במיוחד במכשירים ניידים מה שפוגע בחיי הסוללה) |
| מהיר מאוד בייחס לחיבורים אלחוטיים אחרים |  |
| קיים כמעט בכל מכשיר מודרני |  |
| אבטחה ברמה גבוהה |  |

ארכיטקטורת רשת WiFi

מתאם הWiFi ממיר את המידע לגלי מיקרו ושולח אותם לראוטר בעזרת אנטנה. הראוטר מקבל את הגלים. הוא ממיר אותם בחזרה למידע דיגיטלי ומעביר אותו לרשת האינטרנט. התהליך יכול להתבצע בשני הכיוונים - הראוטר מסוגל לקבל מידע דיגיטלי מהאינטרנט. הוא ימיר אותו לגלי מיקרו וישלח אל המתאם שעל המכשיר.

אבטחה:

גם WiFi הוא חיבור אלחוטי ולכן גם הוא חשוף לפרצות אבטחה כמו ציטוטים והחדרות מידע כי המידע עובר באוויר. בעבר היו כל מיני פרוטוקולי אבטחה כמו WEP וWPA שננטשו מכל מיני סיבות. כיום פרוטוקול WPA2 הוא הנמצא בשימוש כיום. גם בפרוטוקול WPA2 נמצאו חולשות אבטחה ובעקבות זה WPA3 נמצא בשלבי פיתוח. פרצות אבטחה אלו קשות למימוש ולכן WPA2 עדיין מתאים לשימוש סטנדרטי.

## **בלוטות' לעומת תקן חיבור מקביל: Wi-Fi**

שתי הדרכים הנפוצות ביותר ליצירת רשת פרטית בצורת peer to peer הן Wi-Fi ובלוטות'. לכל אחד מהתקנים היתרונות והחסרונות שלו. על מנת להחליט מי מתאים יותר מבין השניים, חקרתי על שני התקנים ומאפייניהם. לאחר מכן סיכמתי את היתרונות והחסרונות של כל אחד, ולבסוף בחרתי לממש את אחד הפרוטוקולים בפרויקט.

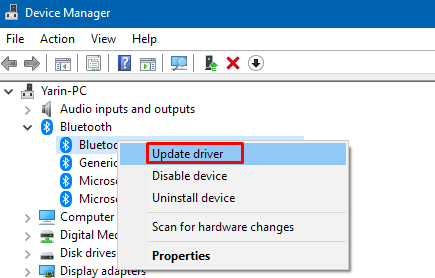
|  |  |  |
| --- | --- | --- |
| פרמטר/תקן חיבור | Bluetooth | Wi-Fi |
| רוחב-פס | נמוך, כ- 2 עד 5 Mbps | גבוה, כ11Mbps |
| מרחק | עד 50 מטר | עד 100 מטא |
| זלילת אנרגיה (סוללה) | נמוך מאוד, השארת המתג דולק כמעט ולא תגרום להפרעה | גבוה יחסית |
| אמינות המידע | אמין, אך פחות מWIFI | אמין, ויותר מבלוטות'. |

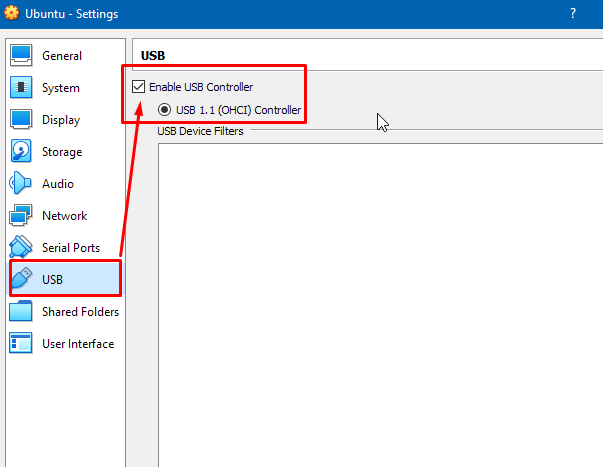
הטבלה מראה כי תכונות המפתח של תקן הבלוטות' הן: שימוש נמוך באנרגיה ,גבולות מרחקים קבועים ומאוזנים לעומת תקן הWi-Fi שאמנם נותן לנו רוחב פס גבוה, אמינות, בתוך מבנים כוחו יורד וזלילת האנרגיה שלו גבוהה מאוד לעומת בלוטות'.

לבסוף בחרתי בבלוטות' מכיוון שאין לי צורך ברוחב פס גבוה ובלותות' זול יותר וצורך פחות חשמל.

## הפעלת בלוטות' על מכונה וירטואלית

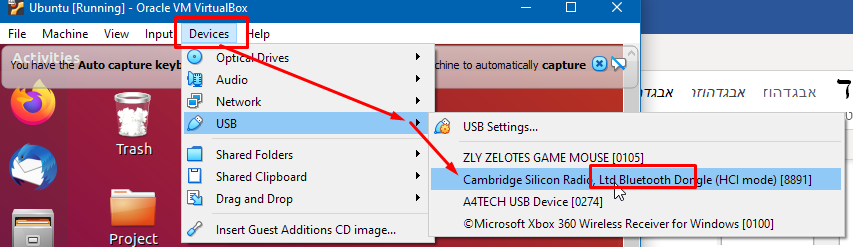
בתוכנת ה-Virtual Box מתאמים לא יהיו מופעלים כברירת מחדל, ויש לבצע הפעלה ידנית שלהם על מנת שהתוכנה תקלוט את המתאם ומערכת ההפעלה הוירטואלית תאפשר הפעלה של בלוטות'.  
  
על מנת להפעיל את מתאם הבלוטות' במערכת וירטואלית יש לעקוב אחרי השלבים הבאים:  
1. **לפני** הפעלת ה-Virtual Box, וודאו שמתאם הבלוטות' מחובר. האינדיקציה לכך היא סמל הבלוטות' המופיע בשורת המשימות  
  
 

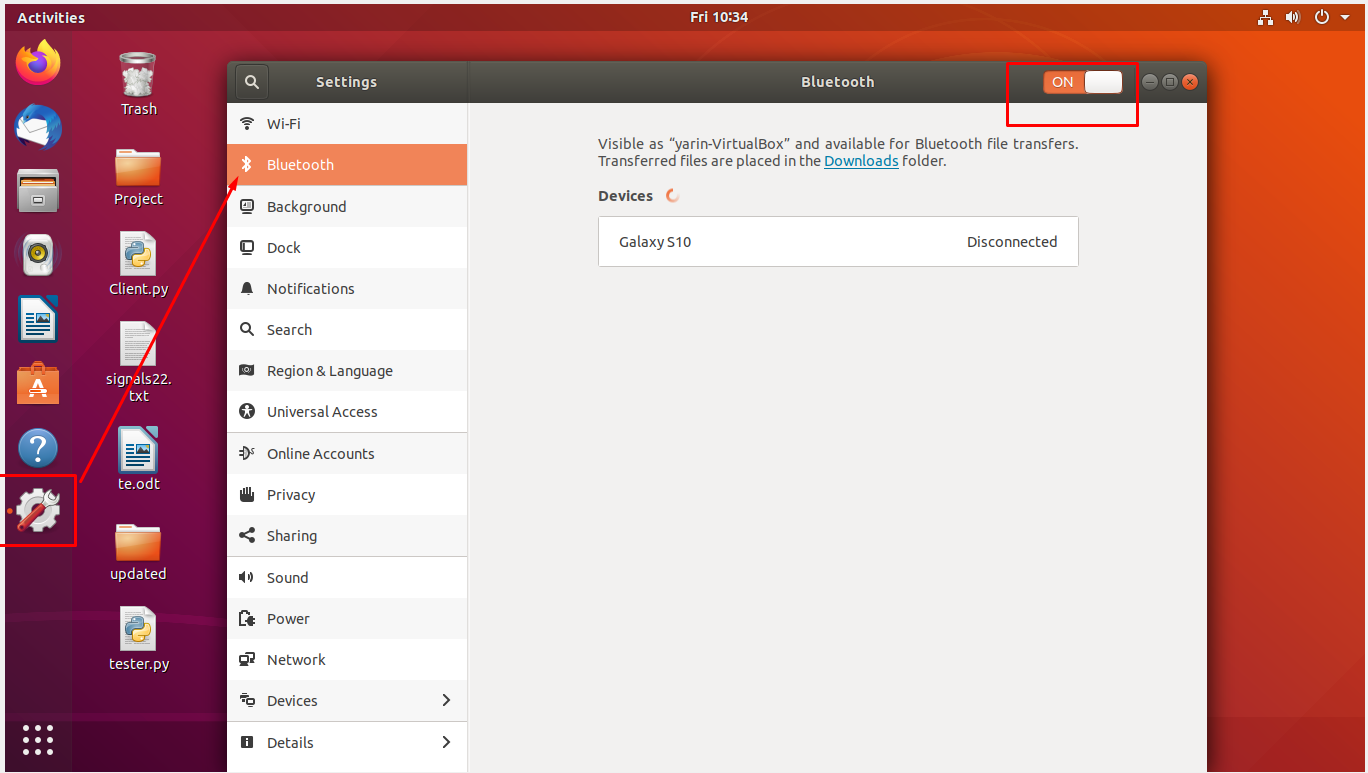
במידה והסמל לא מופיע, פתחו את "מנהל ההתקנים" על ידי חיפושו בחיפוש של "התחל", פתחו את הלשונית בלוטות' וודאו שהמתאם שלכם נמצא שם (אם לא סימן שהמחשב לא זיהה חיבור חדש), לאחר מכן לחצו עליו מקש ימני וUpdate Driver לאחר מכן המתינו לסיום ההתקנה.  
  
 

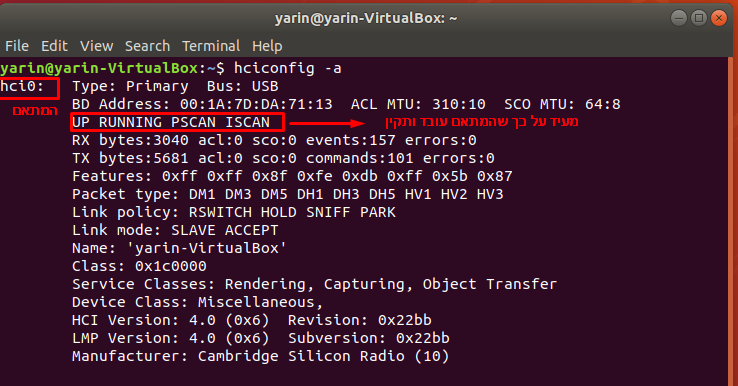
2. פתחו את תוכנת ה-Virtual Box, בחרו את המערכת הרצויה ופתחו את הלשונית Settings, שם בחרו בלשונית USB וודאו שאפשרות הUSB מופעלת:  
 

3. הפעילו את המערכת הוירטואלית והמתינו להפעלה המלאה שלה (עד ששולחן העבודה מופיע על המסך)

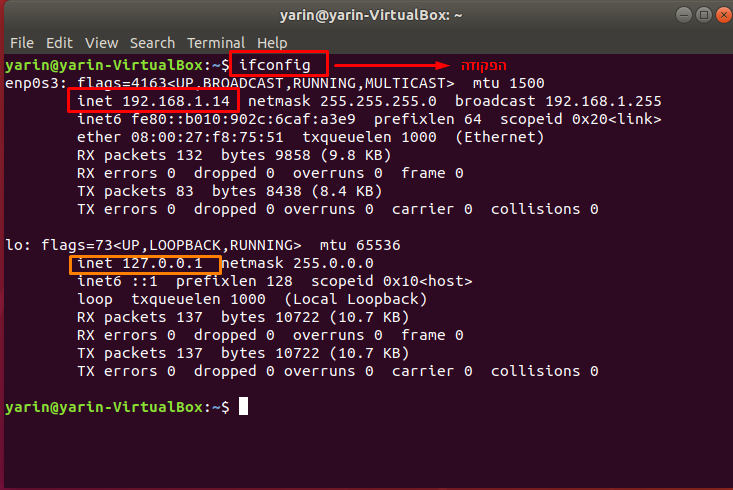
4. לאחר שמערכת ההפעלה הוירטואלית הופעלה, בשורת האפשרויות של Virtual Box למעלה, בחרו בלשונית Devices, הצביעו עם העכבר על הלשונית USB, מצאו את מתאם הבלוטות' שלכם ולחצו עליו.

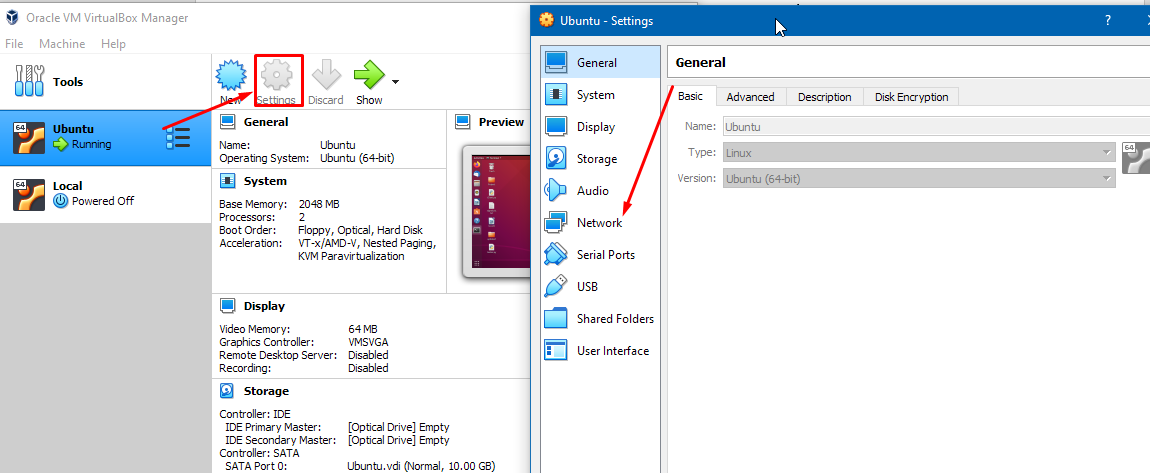


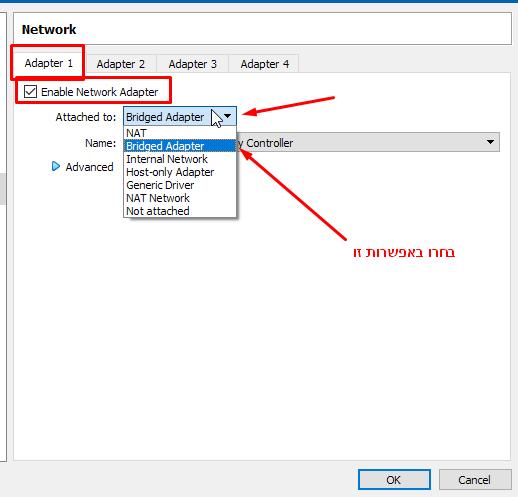
(שימו לב שלכל מתאם שם שונה, אך ניתן לזהות לרוב על פי כך שהמילה Bluetooth תופיע.  
  
5. כעת לאחר שבחרתם במתאם, סמל הבלוטות' יעלם ממחשב המקור, פתחו את לוח ההגדרות, בחרו בבלוטות' והפעילו את המתג.  
  
 

6. וודאו את תקינות הבלוטות' על ידי פתיחת שורת הפקודה, והקלדת הפקודה הבאה:  
hciconfig -a  
  
 

## מציאת כתובות ה-IP של צמתי הקצה

אחד מהפרמטרים הדרושים להפעלת קובץ הפרוייקט הראשי הם כתובות ה-IP ברשת של צמתי הקצה, ניתן לעשות זאת כך:  
פתחו את שורת הפקודה של מערכת ההפעלה (מקש ימני -> open in terminal)  
  
כעת, הקלידו את הפקודה הבאה: ifconfig ולחצו אנטר:  
  
 

כעת תצטרכו לזהות את כתובת ה-IP שלכם מן חלון המידע, הכתובת כתובה בסמוך למילה inet, מורכבת מ4 מספרים מ0-255 כאשר ביניהם נקודות.  
**חשוב לשים לב** שהכתובת המסומנת באדום היא הדרושה ולא האחת המסומנת בכתום, שהיא כתובת ברירת-מחדל לרשת מקומית בכדי שהמחשב יתקשר עם עצמו.  
  
במידה ואתם משתמשים במכונה וירטואלית, יש לבצע כמה צעדים לפני שניתן למצוא את כתות ה-IP:  
  
1. גשו אל הגדרות התוכנה של המערכת הוירטואלית, ופתחו את חלון הNetwork:  


כעת, בחרו במתאם הראשי שלכם, סמנו ב-V את תיבת הסימון ובחרו באפשרות Bridged Adapter כמתואר בתמונה.  
 

לאחר שביצעתם זאת תוכלו להשתמש בצעדים המפורטים מעלה על מנת להשיג את כתובת ה-IP שלכם.

## **מערכות הפעלה מבוססות לינוקס**

לינוקס היא משפחה של מערכות הפעלה, הבנויות על ליבת לינוקס. ליבת לינוקס זמינה לשימוש שינוי והפצה חינם לכל אחד. למעשה, מדובר בתוכנה חופשית עם קוד פתוח. מערכות הבנויות על ליבת לינוקס נקראות גם הפצות לינוקס. הפצות נוצרות על ידי משתמשים או קבוצות המעוניינים לתחזק מערכת הפעלה התואמת את צרכיהם.

מכיוון שהקוד הוא חופשי וחינמי המערכת קלה לשינוי בהתאם לצרכים השונים של כל משתמש. דוגמאות להפצות מוכרות הן: Ubuntu, Mint, Arch ולכל אחת מטרה עיקרית שונה. הפצות גם יכולות להיות מבוססות על הפצות קיימות.  
הממשק העיקרי שעבורו נצטרך הפצה מבוססת לינוקס בפרויקט זה, היא שורת הפקודה (Terminal). שורת הפקודה היא ממשק נפוץ בכל מערכת הפעלה. בין היתר יתרונות של לינוקס שיעזרו לנו בפרוייקט זה הם:

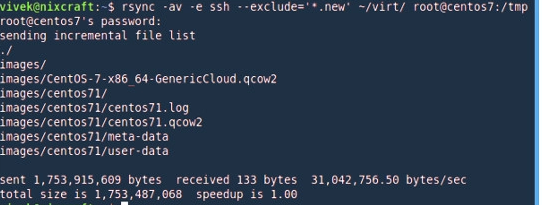
1. הממשק מינימליסטי שמצמצם את השימוש במשאבי המחשב ולא יוצר תעבורה כבדה
2. ממשק הפקודה במערכות מבוססות לינוקס לרוב עשיר בפקודות היעילות למשתמשים המעוניינים לבצע תקשורות בין מערכות (SSH, (SCP

הטרמינל(שורת הפקודה) של לינוקס  
בפרויקט זה נעשה שימוש עיקרי בטרמינל של מערכת ההפעלה שבמקרה זה היא מבוססת לינוקס. הפקודות ירוצו דרך הקוד בעזרת הספרייה OS.  
הטרמינל כולל בתוכו פקודות רבות. ניתן להוסיף אליו פקודות על ידי הורדת ספריות נוספות. הפקודות יכולות להיות מבוססות מערכת ותומכות שליטה על מערכת ההפעלה, וחלקן מממשות פרוטוקולים שונים כגון רשת האינטרנט, בלוטות' וכו'.  
לרוב פקודות המבצעות שינויים משמעותיים במערכת דורשות הרשאות מנהל. ניתן לקבל הרשאת מערכת על ידי הוספת המילה sudo לפני כתיבת הפקודה, אך לאחר מכן יש חובה להכניס את סיסמת המשתמש.

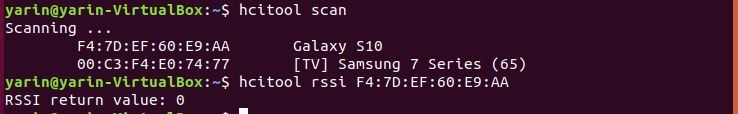
בעזרת שורת הפקודה אני יוצר תקשורת בין צמתי הקצה לצומת הראשית, שליטה על מערכת מרוחקת, העתקת קבצים ותקשורת עם מכשירי בלוטות'.

המחלקות והפקודות העיקריות בהם אני משתמש לצורך הפרויקט הן בלוטות' (hcitool) וSSH (תקשורת בין מערכות).(לשנות ניסוח)

1) SSH, פקודות המאפשרות תקשורת בין לקוחות ושרתים, ואף שליטה עליהם.  
Secure Shell- הוא פרוטוקול לתקשורת מחשבים המאפשר למפעיל הפקודה לבצע פקודות נוספות על המחשב המרוחק. SSH פועל מעל לשכבת TCP (SSH הוא חלק משכבת השיחה) ולרוב רץ על פורט 22, אך זה ניתן לשינוי.  
SSH גורם לתנועת תעבורה בין מחשבים באופן מאובטח על גבי ערוץ תעבורה, גם אם הרשת אינה מאובטחת. החיבור נעשה בצורת שרת-לקוח, לרוב משומש לקבלת גישה למערכות מבוססות לינוקס מרוחקות.  
החיבור דורש תהליך הזדהות של מפתחות RSA וסיסמת המערכת.  
חוץ מהפעלת פקודות מרחוק, ניתן להשתמש בSSH גם לביצוע תיעול, העברת קבצים (בעזרת פקודת SCP המבוססת על חיבור SSH). התיעול גם יכול להביא לאבטחה באינטרנט.

Secure Copy- פרוטוקול תקשורת מחשבים התומך בהעברת קבצים מלקוח לשרת, שרת ללקוח או משרת לשרת. הפרוטוקול מבוסס על הפרוטוקול SSH. אופן ההעתקה מתבצע בצורה מאובטחת, ודורש תהליך הזדהות של סיסמא.  
 

2) פרוטוקול הבלוטות' מכיל פקודות המאפשרות שינוי מאפיינים והפעלת תקשורת עם מכשירי בלוטות'. הפקודות העיקריות בהן ייעשה שימוש הן:  
hcitool - פקודה המשמשת לשינוי מאפייני חיבורי בלוטוס בנוסף לפקודות מיוחדות אחרות שניתן לשלוח למכשירי בלוטוס   
scan- פקודה המבצעת סריקה למכשירי בלוטוס בטווח הקליטה, יודפסו גם מספרי ה-ID של כל מכשיר ושמו.



## טריאלטרציה וטריאנגולציה

טריאנגולציה

טריאנגולציה או שילוש היא תהליך חישוב קואורדינטות של נקודה במרחב. החישוב מתבצע באמצעות משולש שידוע אורך אחת מצלעותיו ומיקום שני הקודקודים שבקצות צלע זו, והזוויות הנוצרות בשני קודקודים אלה עם הצלעות המחברות אותם לנקודה המבוקשת. לחישוב משמש משפט הסינוסים.

טריאלטרציה

טריאלטרציה היא אלגוריתם המבוסס על הטריאנגולציה. אלגוריתם זה משתמש במרחקים ידועים של 3 נקודות קבועות במרחב.

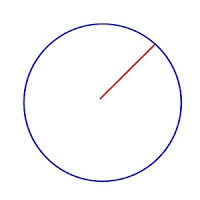
אלגוריתם זה משמש אותי בפרוייקט למציאת מרחקים במרחב הדו מימדי. בעזרת חוזק האות אני יכול למצוא את המרחקים במרחב הדו ממדי, ולכן אלגוריתם זה משרת את העבודה שלי.

בנוסף הרבה ממערכות הניווט היום ואפילו מערכת ה GPS (Global Positioning System) מבוססת על אלגוריתם זה, מה שמעיד על התאמת אלגוריתם זה לעבודה שלי.

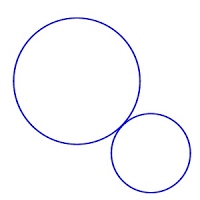
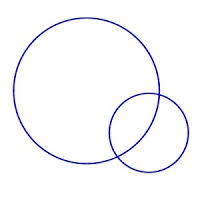
איך האלגוריתם עובד

שלושת הבסיסים שלנו יוצרים צורת משולש, אך נוכל למצוא גם נקודות הנמצאות **מחוץ** למשולש, כל עוד הן נמצאות בטווח הקליטה של שלושת הבסיסים.

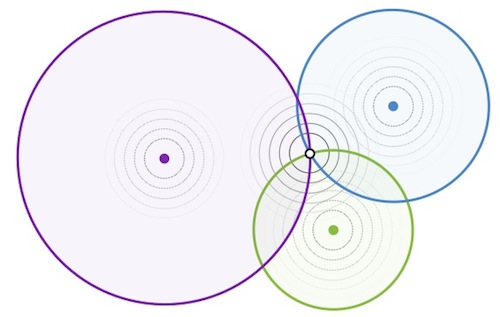
מרחק מנקודה לא ידועה לבסיס שמיקומו ידוע מראש הוא בעצם רדיוס לנקודות אופציונליות, שיוצרות מעגל. אחת מהנקודות על המעגל היא הנקודה שלנו. כך כשנתקדם נדע שהנקודה היא לא בתוך המעגל, ולא מחוצה לו אלא על המעגל.



המרחק לנקודה השנייה הוא רדיוס נוסף למעגל שני של נקודות אופציונליות. נקודה הנמצאת על שני מעגלים יכולה להיות נקודת חיתוך בין שני מעגלים או נקודת השקה בניהם. במקרה שנקודה זו היא נקודת השקה אנו כבר יודעים את מיקומה, אך כאשר נקודה זו היא נקודת חיתוך קיימות שתי נקודות אופציונליות שיכולות להיות הנקודה שלנו. חשוב לציין המעגלים לא יכולים להיות חופפים כיוון שהבסיסים שהצבנו לא באותו המקום.



המרחק לנקודה השלישית הוא רדיוס נוסף למעגל ומעגל זה יקבע לנו (במקרה שבו המעגלים חותכים) איזה מן הנקודות היא הנקודה אותה אנו מחפשים.



איך האלגוריתם עובד בקוד

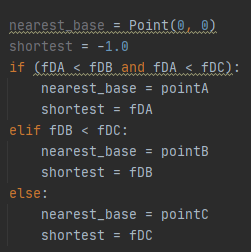
ראשית נגדיר את הקורדינטות שלנו שיהיו בייחס של 1:1 כלומר כל מטר הוא יחידה אחת במרחב שלנו. נרצה להגדיר אותו כמספר עשרוני כיוון שאולי נרצה להשתמש במקומות יותר קטנים למשל בחצי מטר



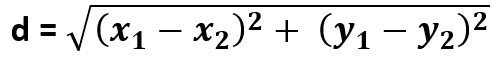
לאחר מכן נרצה למצוא את נקודת האמצע של המישור הדו מימדי שבטווח הקליטה שלנו ע"י ממוצע ערכי הx וערכי הy.



אחרי שיש לנו את המרכז נמצא את הבסיס שהכי קרוב לנקודה שאנו מחפשים. חשוב לשים את הטווח הכי קצר כשלילי כי אם הנקודה שאנו מחפשים נמצאת על בסיס המרחק הוא 0.



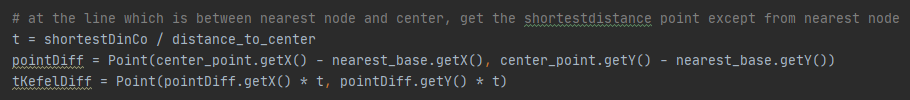
את המרחק נחשב באמצעות הנוסחה לחישוב מרחק בין 2 נקודות במרחב קורדינטות.



לאחר מכן נגדיר משתנה t ולתוכו נכניס את המרחק הקצר ביותר מצומת קצה אל מכשיר ונחלק ערך זה במרחק בין הצומת לנקודת האמצע כדי לקבל את היחס בניהם.

ניצור נקודה בשם pointDiff שערך הX- וערך ה-Y שלה הם ההבדל בערכי ה-X וה-Y של נקודת האמצע ונקודת הצומת שנמצאה כהכי קרובה למכשיר הנסרק. ניצור עוד נקודה שהיא ערכי ה-X וה-Y של הנקודה PointDiff רק שכעת נכפול אותה ביחס t.

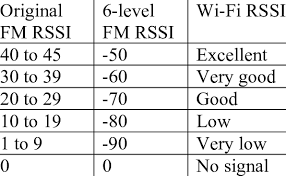
משם ניקח את הנקודה הסופית , שהיא ערך נקודת הצומת הקרובה ביותר למכשיר, בחיבור עם הנקודה האחרונה שיצרנו (יחס כפול ההבדל בערכי X ו-Y)

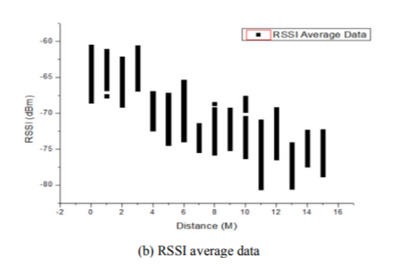


## RSSI (Received signal strength indicator)

RSSI הוא מדד לחוזק האות המתקבל באנטנה. RSSI נמדד באחוזי חוזק, ככל שערכו המספרי גדול יותר כך חוזק האות גם גדל. ניתן למדוד מרחקים באמצעות חוזק האות כיוון שחוזק האות מתחזק כשמתרחקים ונחלש כאשר מתרחקים.

דוגמה לערכי RSSI ולמשמעות שלהם אפשר לראות בתמונה לקמן. תמונה זו מראה את ערכי ה RSSI (משמאל) ואת חוזק האות (מימין). בטכנולוגיית ה WiFi.





## raspberry pi

Raspberry Pi הוא מחשב זול וקטן (בגודל של פלאפון) שמסוגל להתחבר למסך מחשב או לטלוויזיה ויש לו אפשרות חיבור של מקלדת ועכבר סטנדרטיים. על אף הגודל שלו מגוון רחב של פעולות כמו שמחשב רגיל מסוגל לבצע בנוסף לניהול משימות ייעודיות לתכנות פיזי (Physical computing). למחשב זה 2 שימושים עיקריים:

שימוש במעבד מידע מרכזי לפרויקטים אלקטרונים – בזכות היותו מחשב עם אפשרות לחבר רכיבים אלקטרונים בקלות, Raspberry Pi מאפשר לפתח פרוייקטים אלקטרונים בקלות ועם רכיבים וזולים ובקלות.

שימוש במחשב Raspberry Pi המחשב יעודי – ישנם המון תוכנות (או יותר נכון ספריות) יעודיות Raspberry Pi מה שהופך את המחשב הזעיר למחשב יעודי עבור חלק מהמשתמשים.

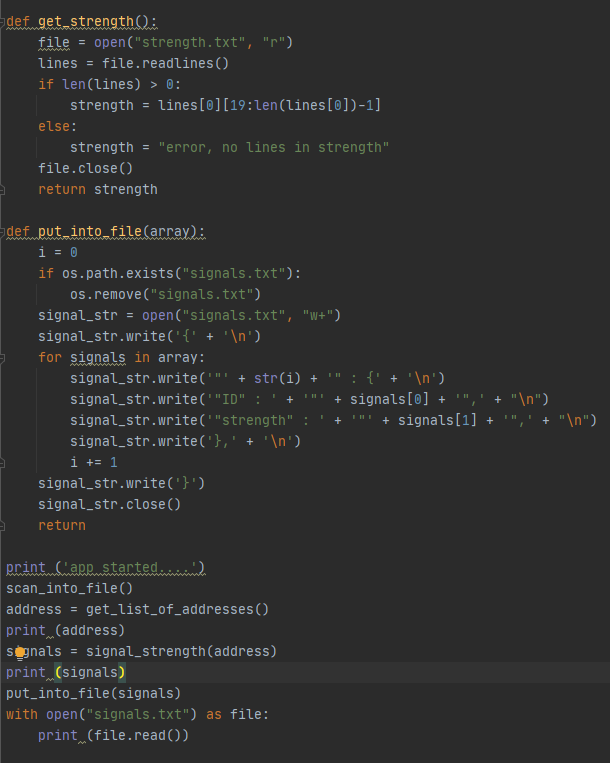
ה Raspberry Piמשתמש במערכת ההפעלה linox החינמית עליה הרחבנו קודם.



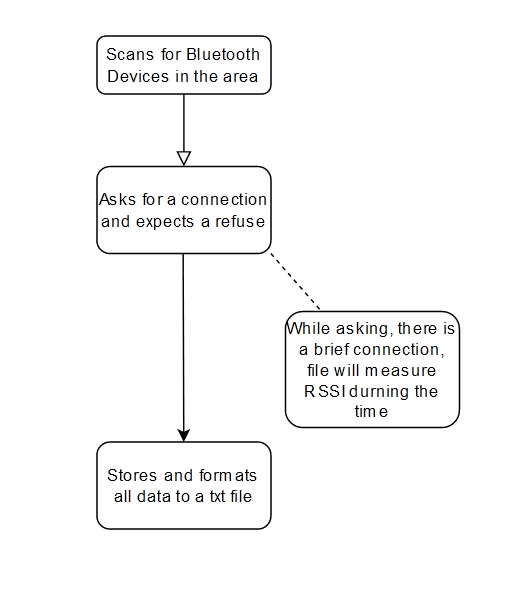
# הסברים על הקודים

## קוד1: client.py



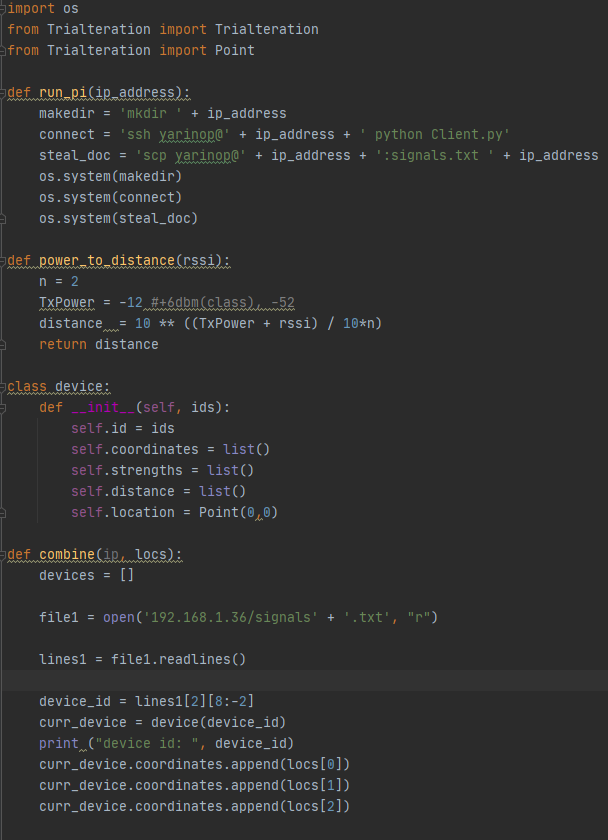


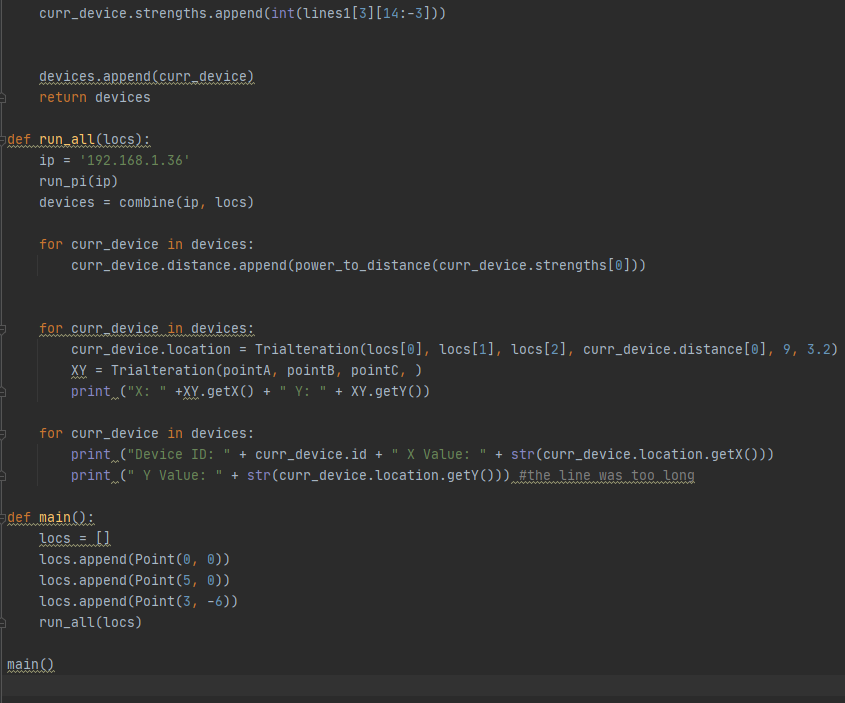
|  |  |  |
| --- | --- | --- |
| שם הפעולה | טענת כניסה | טענת יציאה |
| Scan\_into\_file | אין | הפעולה מפעילה פונקציית לינוקס המשתמשת בפרוטוקול בלוטות' על מנת לסרוק עבור מכשירי בלוטות' שונים וכתובותיהן, התוצאות נשמרות לקובץ addresses.txt |
| get\_list\_of\_addresses | אין | מייצא רקאת כתובות הבלוטות' מתוך הקובץ מהפעולה הקודמת אל רשימה |
| signal\_strength | רשימה של כתובות בלוטות' | הפעולה מחזירה רשימה שכל תא בא הוא tuple המחולק לשניים, כתובת המכשיר וערך הRSSI שלו. בנוסף נוצר קובץ txt בשם strength.txt ושם בו את ערכי הRSSI |
| get\_strength | אין | הפעולה פותחת את הקובץ strength.txt ומייצאת ממנו אך ורק את הערך המספרי של RSSI |
| put\_into\_file | רשימה הכוללת בתוכה טאפלים של כתובות וערכי RSSI | הפעולה מעבדת את הנתונים הסופיים וכותבת אותם אל קובץ txt בפורמט קל לקריאה עבור המשתמש |

תרשים זרימה עבור פעולות הקובץ:  
  
 

תפקיד הקובץ: קובץ הכולל בתוכו את הפעולה המתמטית שמחשבת נקודה באמצעות אלגוריתם הטריאלטרציה, יבוצע import לפעולות אל הקובץ הראשי (יפורט בהמשך), הסיבה להפרדה היא בכדי לבצע סדר בקוד והפרדה בין החלק המתמטי לחלק שעוסק בפקודות ובקוד.

## קוד 2: Adapted\_Server.py

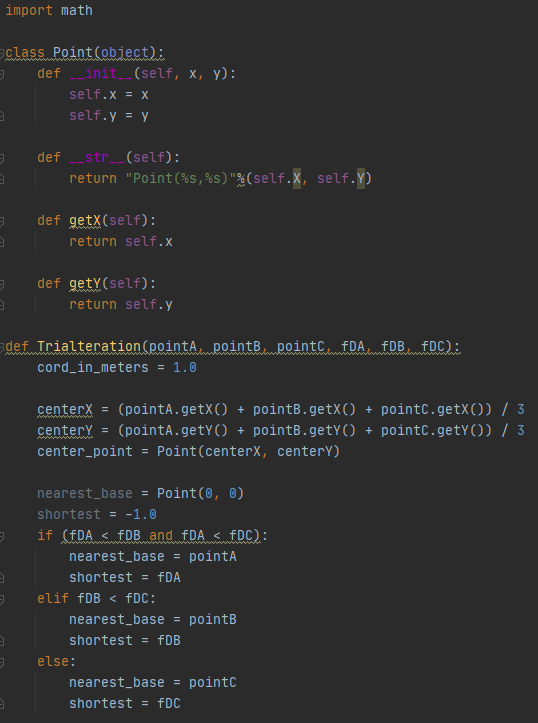


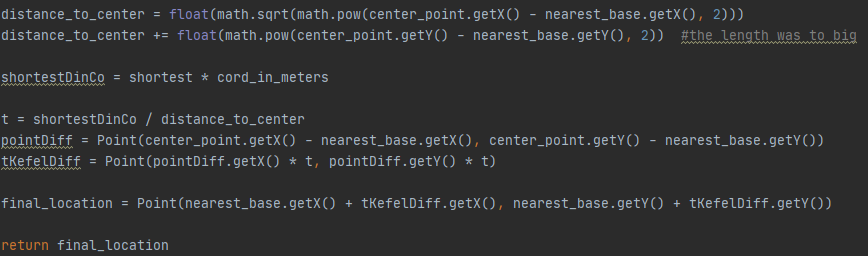


|  |  |  |
| --- | --- | --- |
| שם הפעולה/מחלקה | טענת הכניסה | טענת היציאה |
| run\_pi | כתובת IP של מחשב המריץ לינוקס | הפעולה יוצרת במחשב המקומי תיקייה בשם כתובת ה-IP, מריצה על המחשב המרוחק את קובץ הפייתון הדרוש ומעתיקה מהמחשב המרוחק אל המקומי את קובץ התוצאות. |
| power\_to\_distance | ערך RSSI ממכשיר בלוטות' מסוים. | הפעולה תמיר את ערך הRSSI למרחק במטרים. |
| combine | שלושת כתובות ה-IP של צמתי הקצה, שלושת המיקומים שלהם | הפעולה מאחדת את כל הנתונים מצמתי הקצה לעצמים מסוג מכשיר, כל אחד ונתוניו בהתאם |
| do\_everything | שלושה כתובות IP של צמתי הקצה ושלושת מיקומיהם. | הפעולה הראשית בקובץ, מפעילה את כל הפעולות הצדדיות בסדר המתאים ולבסוף מחזירה ערך נקודתי של כל מכשירי הבלוטות' בשטח. |

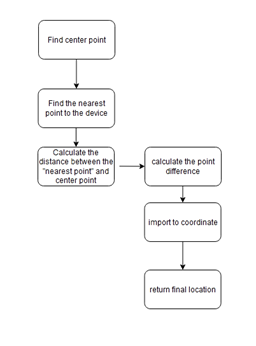
תפקיד הקובץ: לבצע את כל הרצת ולוגיקת הפרויקט, לבסוף להחזיר מיקום סופי של מכשירי הבלוטות' באזור.

## קוד 3: Trialteration.py





|  |  |  |
| --- | --- | --- |
| שם הפעולה/מחלקה | טענת כניסה | טענת יציאה |
| Class Point | \_\_init\_\_  ערך X  ערך Y | עצם בעל הערכים (תכונות) שנבחרו  (x וy במקרה שלנו) |
| \_\_str\_\_  אין | מחרוזת המתארת את העצם ותכונותיו |
| getX  אין | הפעולה מחזירה את ערך X |
| getY אין | הפעולה מחזירה את ערך Y |
| Trialteration | 3 נקודות שונות המייצגות צמתי קצה, והמרחק של כל נקודה אל המכשיר בלוטות' הרלוונטי בהתאמה | הפעולה תחזיר נקודה המייצגת את מיקום מכשיר הבלוטות' שאותו אנו מחפשים |

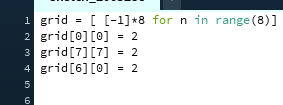
תרשים זרימה עבור פעולות הקובץ:  
  
 

תפקיד הקובץ: קובץ הכולל בתוכו את הפעולה המתמטית שמחשבת נקודה באמצעות אלגוריתם הטריאלטרציה, יבוצע import לפעולות אל הקובץ הראשי, הסיבה להפרדה היא בכדי לבצע סדר בקוד והפרדה בין החלק המתמטי לחלק שעוסק בפקודות ובקוד.

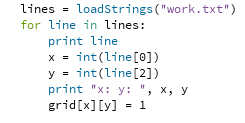
הקוד מוסבר בהסבר על טריאנגולציה להסביר שוב?

## ייצוג גרפי בסיסי של מכשירי הבלוטות' שמיקומם נמצאו

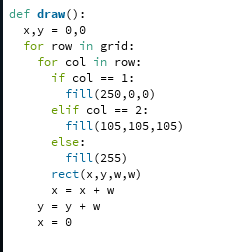
נרצה לייצג את הנקודות שמצאנו (נקודות המכשירים שמצאתי בעזרת המוצר), בצורה בסיסית על מנת לקבל ראייה כללית של היכן נמצא כל מכשיר ביחס לאחר, בשביל לייצג שטח דו-ממדי בסיסי נשתמש במנוע Processing עם שפת פייתון, התוכנה Processing מאפשרת לנו לצייר מערכים בסיסיים, לצבוע אותם ולייצג בעזרת צבע על פי ערך.  
  
איך נייצג צבע בקוד? על פי תכונת RGB, לדוגמא- הצבע אפור: (105,105,105)

כל קטע קוד הבנוי בProcessing מכיל שתי פעולות עיקריות:  
setup: הרכבת "המסך" שלנו והמשתנים הבסיסיים שנרצה להגדיר לפני העיבוד הגרפי.  
draw: העיבוד הגרפי, איטרציה על מבנה נתונים וצביעתו על פי ערכים.  
  
setup():  
תחילה נגדיר את גודל המסך שלנו על ידי הפעולה size, נכניס פרמטרים מספריים פשוטים שיקבעו את רזולוציית המסך שייפתח.  
נגדיר את מבנה הנתונים שלנו, מערך דו ממדי בגודל 8 על 8, הגודל ניתן לשינוי חופשי בהתאם לרצון כותב הקוד.  
נכניס כפרמטרים את מיקומי צמתי הקצה שלנו אל תוך מבנה הנתונים, ונגדיר להם ערך אחיד- במקרה זה בחרתי במספר "2" (אין לערך חשיבות כלשהיא, הערך הוא אינדיקטור להפרדה בין כל חלק במבנה הנתונים).  
דוגמא: במידה ואלו מיקומי צמתי הקצה שלנו:   
A(0,0)  
B(7,7)  
C(6,0)  
נגדיר את המערך הדו-ממדי ונשייך את הערכים כך:   
 

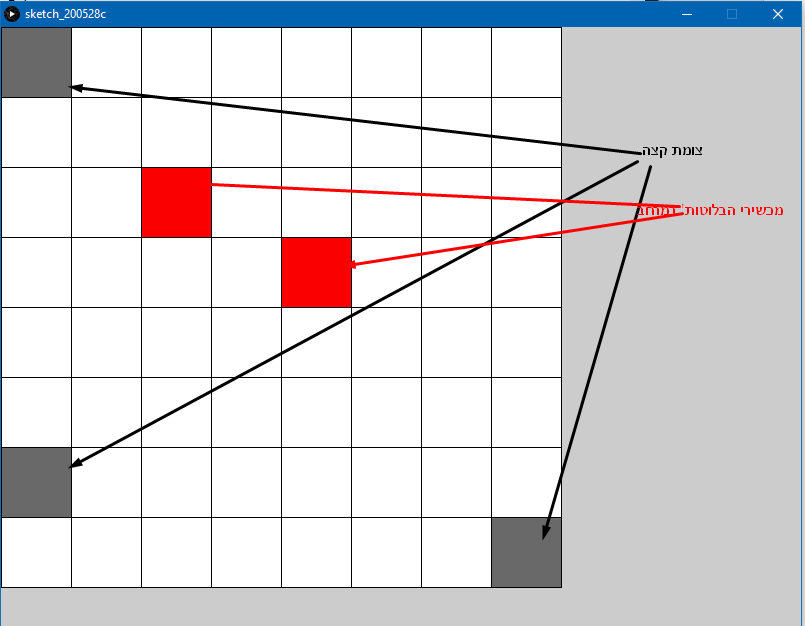
בפעולת setup מתבצעת גם איטרציה על קובץ שיכיל את נתוני מיקומי מכשירי הבלוטות' במרחב, כך שאני משייך אליהם גם ערך, במקרה זה הספרה "1".



Draw():   
בפעולה זו נבצע איטרציה על המערך הדו-ממדי שלנו, נעבור על כל ערך ונבדוק אותו באמצעות if statements, נבדוק את הערך ונשייך לו לצבע.  
דוגמא: אם התא שנמצא במיקום [2][3] משויך לערך 1, צבע אותו ב-אדום ((250,0,0



כפי שניתן לראות, אל צמתי הקצה שייכתי את הצבע אפור ואל מכשירי הבלוטות' שייכתי את הצבע אדום, נביא דוגמה של נתונים ונראה את התוצאה הגרפית:  
צמתי קצה (אפור): מיקומיהם הם (0,0) (7,7) (0,6)  
מכשירי בלוטות' (אדום) מיקומיהם הם (4,3) (2,2)  
נריץ את קוד הProcessing ונקבל את התוצאה:



# רפלקציה

העבודה על הפרויקט הייתה עבורי אתגר מהנה ומלמד, למדתי לתכנת בצורה קצת שונה, בחופשיות לעבוד עם כל כלי קיים על הפרויקט, עשיית הפרויקט, כתיבת הקוד והמחקר עליו, והסברתו בספר הרגישו כמו עבודה אמיתית על מוצר אמיתי שמפתחים.

קיבלתי ידע רב על נושא מערכות הפעלה מבוססות לינוקס, למדתי לתפעל אותן בעזרת פקודות, בנוסף למדתי המון על נושאי בלוטות' ו-Wi-Fi, ראיתי את היתרונות והחסרונות של כל אחד מהם ולפי כך בחרתי את הפרוטוקול המתאים עבורי.

הכלים שאני לוקח עם עצמי הם כתיבת קוד ופיתוח מוצר, תיאור קוד בעברית והסבר אלגוריתם, חקירה על מושגים שאיני מכיר וסיכומם, ועבודה עם מערכות זרות עבורי.

הקשיים והאתגרים שעמדו בפניי היו בעיקר שגיאות בכתיבת הקוד, תחילה הידע שלי במערכות לינוקס שאף לאפס ובקושי הצלחתי לתפעל אותה, זה התבטא בדברים בסיסיים כמו חיבור בלוטות' או חיבור לרשת, בנוסף חששתי מכך שלא אצליח לפרט בצורה מספקת על הקוד שלי ואופן פעולתו, אך עם המשך כתיבת הקוד ראיתי שיש המון עומק לשורות והן דורשות ידע על מנת לתפעל אותן.

במידה והייתי מתחיל היום, הייתי משנה את סדר העבודה שלי, במקור כשהתחלתי לעבוד מיהרתי לכתוב קוד בלי ידע קודם רב על הנושא שלי, לאחר שראיתי שאני לא מצליח להסתדר עם כך החלטתי לחקור על המושגים הקשורים לנושא, צפיתי בסרטונים, קראתי סיכומים ודוגמאות קוד קצרות – דברים שעזרו לי לבסוף להרכיב מוצר סופי שגם אני מרוצה ממנו.

# ביבילוגרפיה

Goran J.(24/9/2018). How To Use SSH To Connect To A Remote Server In Linux Or Windows. Retrieved June, 19,2020, from:

<https://phoenixnap.com/kb/ssh-to-connect-to-remote-server-linux-or-windows>

Chris H.(10/7/2017), How to Connect to an SSH Server from Windows, macOS, or Linux. Retrieved December,28,2019 from:

<https://www.howtogeek.com/311287/how-to-connect-to-an-ssh-server-from-windows-macos-or-linux/>

N.A. (n.d). SSH Login Without Password .Retrieved June, 14,2020, from:

<https://www.linuxtrainingacademy.com/ssh-login-without-password/>

N.A. (n.d). HOW TO SET UP WIFI ON THE RASPBERRY PI 3.Retrieved June, 3,2020, from:

<http://www.circuitbasics.com/how-to-set-up-wifi-on-the-raspberry-pi-3/>

N.A, (n.d). Raspberry Pi Documentation. Retrieved June, 11,2020, from:

<https://www.raspberrypi.org/documentation>

N.A. (n.d). Connecting to Raspberry Pi without a monitor for Beginners 3.Retrieved December, 20,2019, from:

<https://www.dexterindustries.com/howto/connecting-raspberry-pi-without-monitor-beginners/>

N.A. (12/11/2016). Trilateration vs Triangulation – How GPS Receivers Work.Retrieved 12,December,2020, from:

[https://gisgeography.com/trilateration-triangulation-gps](https://gisgeography.com/trilateration-triangulation-gps/)