The Iby and Aladar Fleischman
Faculty of Engineering
Tel Aviv University

הפקולטה להנדסה
ע"ש איבי ואלדר פליישמן
אוניברסיטת תל אביב

מערכת חישה לאלגוריתמיקה מתקדמת

פרויקט מס' 22-1-1-2667

דו"ח סיכום

מבצעים:

טל אריאלי

שחר הנובר

מנחה:

שמחה ליבוביץ אוניברסיטת ת"א

מקום ביצוע הפרויקט: אוניברסיטת ת"א

תוכן עניינים:

רשימת איורים:

רשימת טבלאות:

רשימת גרפים:

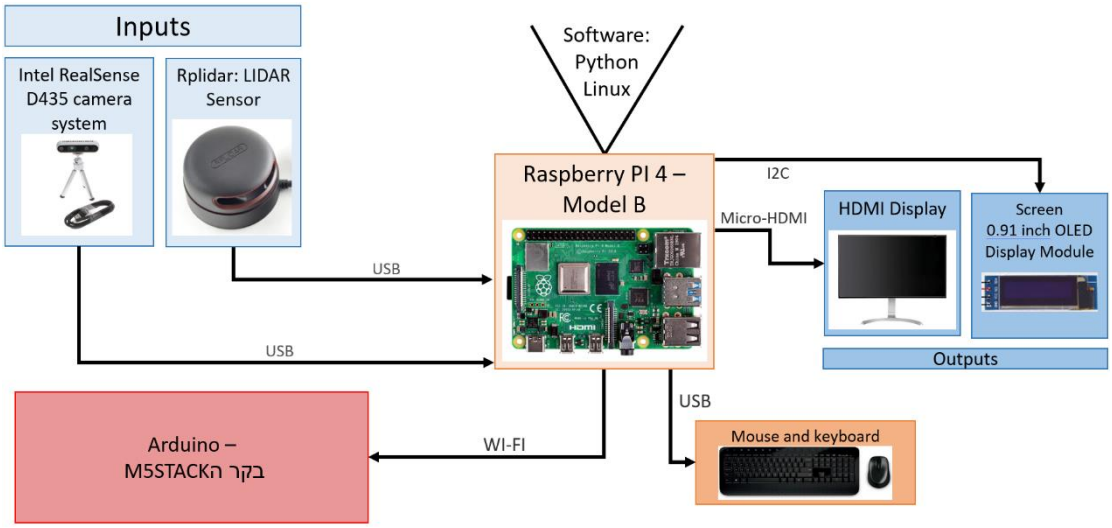
תקציר

פרויקט גמר זה עוסק בבניית תשתית מבוססת Raspberry Pi עבור פרויקטים עתידיים בהנדסת חשמל בדגש על פרויקטים בתחום הפיזיותרפיה. מהות הפרויקט היא ליצור סביבת עבודה ידידותית אשר כניסותיה יהיו סנסורים (כדוגמת מצלמה וחיישן LIDAR) ויציאותיה יהיו חיווי של תאורה או קול בהתאם לדרישת המשתמש. במרכז סביבת העבודה יעמוד הRaspberry Pi ודרכו תתבצע קליטת אות הסנסורים, עיבודם והעברתם לחיווי הרלוונטי. בנוסף, פרויקט זה יכיל תיעוד מפורט של החומרה והתוכנה והממשק בניהם, זאת על מנת להקל על אלו שישתמשו בתשתית זו וירצו לשפרה. התיעוד יכיל פירוט של סביבת העבודה, הרכיבים השונים בה, כל חיבור וחיבור שיעשה בסביבת העבודה וכל התוכנה שבה נעשה פירוט לרבות הקוד להפעלת וההתמודדות עם התרחישים השונים. כחלק מהעבודה על הפרויקט, ביצענו מחקר והשוואות בין סוגי סנסורים שונים ובעבור כל סנסור ביצענו עבודת השוואה והתאמה לRaspberry Pi שלנו על מנת למצאו את החיישנים התואמים ביותר.

בנוסף נעשתה עבודה משותפת עם הצוות המקביל שעבד על פרויקט משלים לפרויקט שלנו המבוסס על תשתית ארדואינו מסוג M5Stack וחיישנים משלימים לחיישנים שעשינו בהם שימוש. התקשורת בין שני הפרויקטים נעשתה באמצעות WIFI.

תוצרי הפרויקט מכילים את שרטוט ובניית התשתית החומרתית של רכיב הRaspberry PI והחיישנים אשר מחוברים אליו, מסמך בחירת החיישנים אשר מכיל עבור כל קטגוריה של חיישן את החומרה שיכולה להתאים לצרכי הפרויקט (מבחינת חומרה ומבחינת מבדקים שניתן לבצע). בנוסף, תוצרים נוספים של הפרויקט הינם מדריכי התקנה מפורטים שמכילים שלב אחר שלב את השימוש בכל חיישן (התקנה ראשונית ושימוש שגרתי) וכן גם תוכניות בדיקה (סקריפטים) ייעודיים לבדיקות שקבענו להראות את פונקציונליות החיישנים ואת השימוש בהם על פי צרכים שכיחים של התשתית.

דיאגרמת בלוקים :



איור 1- דיאגרמת בלוקים

הקדמה

מטרות הפרויקט הוא ליצור תשתית נוחה למשתמש וקלה לתפעול ולאדפטציה בהתאם לצרכי הפרויקט העתידי שיעשה בה שימוש. התשתית מפורטת על כל רכיביה ועל כן ניתן יהיה להשתמש בה ב"קיצור דרך" בעת בנית מערכת חדשה לניטור בעבור צורך נקודתי.

תשתית זו תוכל לסייע במגוון דרכים בתחום הפיזיותרפיה כמו למשל בניית מערכת חיישנים על כיסא גלגלים אשר תתריע בפני מכשולים או מיקום התשתית וחיישניה על גבי הליכון וליצור מערכת ניתור של הסביבה.

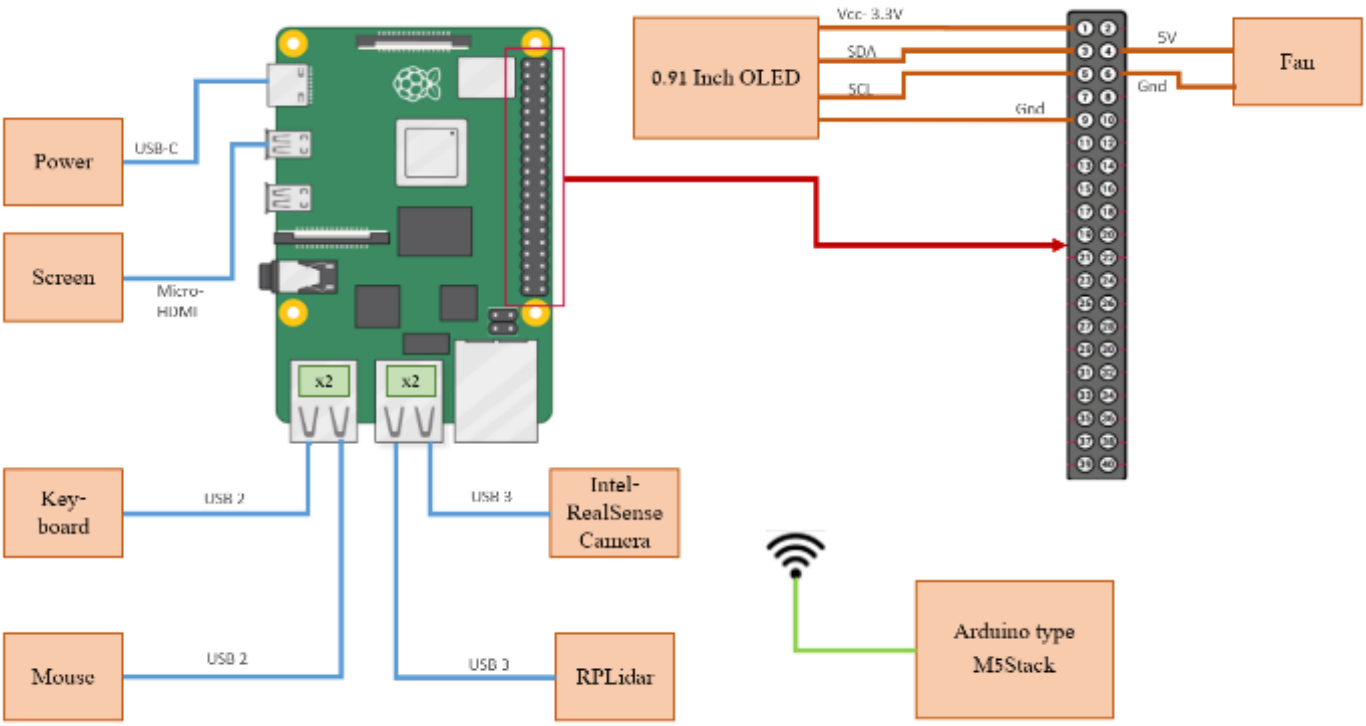
הגישה שננקטה בעת העבודה על הפרויקט הינה הבנת כל רכיב מסביבת העבודה באופן עצמאי, הבנת מערכת ההפעלה, ההתקנה והשימוש שלו.

הדרישה הסופית היא ליצור בסיס תשתיתי שבסופו של דבר יהיה ניתן לקחת אותו וממנו לבצע את ההתאמה לצרכים הנקודתיים של הפרויקטים שיעשו בו שימוש ועל כן העבודה על כל חיישן בנפרד הייתה חשובה ויסודית מאוד. חשוב היה לנו ליצור מדריכים מפורטים על ההתקנה של כל חיישן ומה ניתן להשיג באמצעותו. באופן הזה, יצרנו גם כן מדריכים שיתארו את המבדקים הבסיסיים והשכיחים ביותר עבור פעולתו של כל חיישן ועל כך נפרט בהמשך.

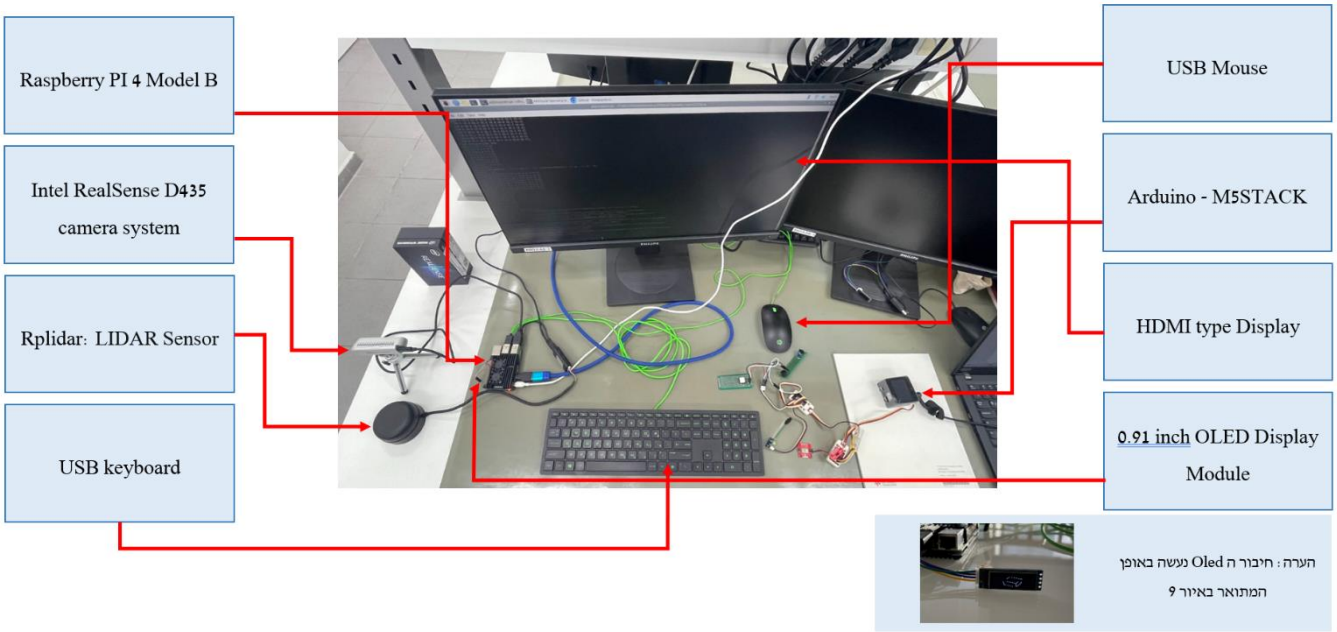
חשוב לציין כי ישנם המון מדריכים שקיימים באינטרנט ושהעבודה עם החיישנים שנבחרו היא עבודה נוחה ונגישה גם באמצעות כלים שאפשר להשיג מהרשת. בחרנו להנגיש בעבודה זו כמה וכמה מקורות שעשינו בהם שימוש ואלו עוברים עדכונים מפעם לפעם ולכן בעת שימוש עתידי בחיישנים שקיימים בתשתית זו חלק מהפקודות יכולות לעבור התאמה ועדכון ולכן חשוב להישאר מעודכנים עם הגרסאות שקיימות ברשת.

מימוש

תחילה, נציג את מבנה החיבור של החיישנים יחד עם הRaspberry PI .



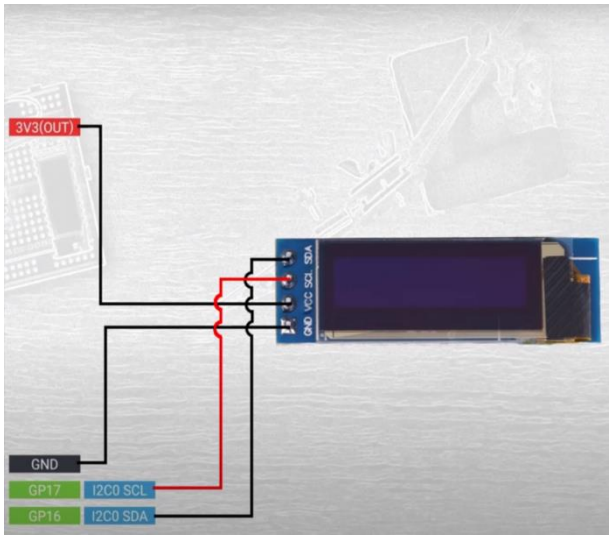
איור 2- תצורת חיבור תשתית הפרויקט

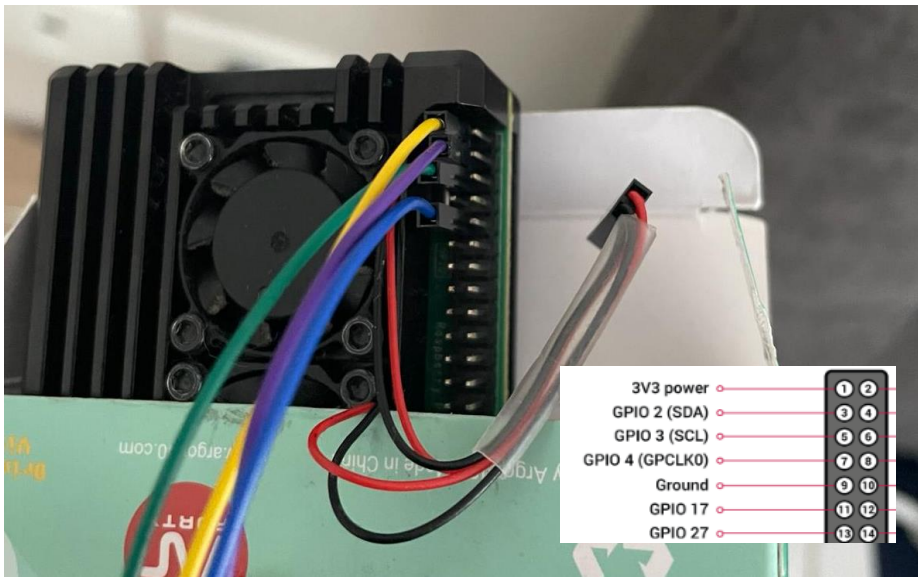


איור 3- צילום התשתית ומקרא של כלל הרכיבים

ניתן לראות כי הRaspberry PI מחובר לחיישני המצלמה והLIDAR, וכן גם למסך, עכבר ומקלדת בו זמנית. ניתן באותו הזמן לחבר אותו גם למסך הOLED כפי שמופיע בתרשים. בנוסף ניתן לראות את החיבור לתשתית הM5stack יחד עם החיישנים שלו. התקשורת בין שתי התשתיות נעשית באמצעות חיבור WIFI.

<u>חיבור מסך הOLED אל תשתית ה Raspberry PI :</u>





איור 4- צילום ומקרא חיבורים צג הOLED

OLED Pin	PI PGIO Pin	Notes
Vcc	1	3.3 V
Gnd	9	Ground
SCL	5	I2C Clock
SDA	3	I2C Data

טבלה 1- חיבורי צג הOLED

בעבור כל אחד מהחיישנים נדרשנו לבצע התקנה תוכנתית, קישור החיישן לRasppberry PI, הורדת הספריות הרלוונטיות להפעלת המבדקים והתמודדות עם שגיאות התקנה ושדרוגי גרסאות. עבור כל אחד מהחיישנים ישנו מדריך התקנה מפורט. חוברת מדריכים זו נמצאת בתוצרי הפרויקט.

כעת נעבור על הרכיבים השונים שבהם נעשה שימוש וניתן את הרקע אודותיהם:

Raspberry Pi 4 Computer- Model B:

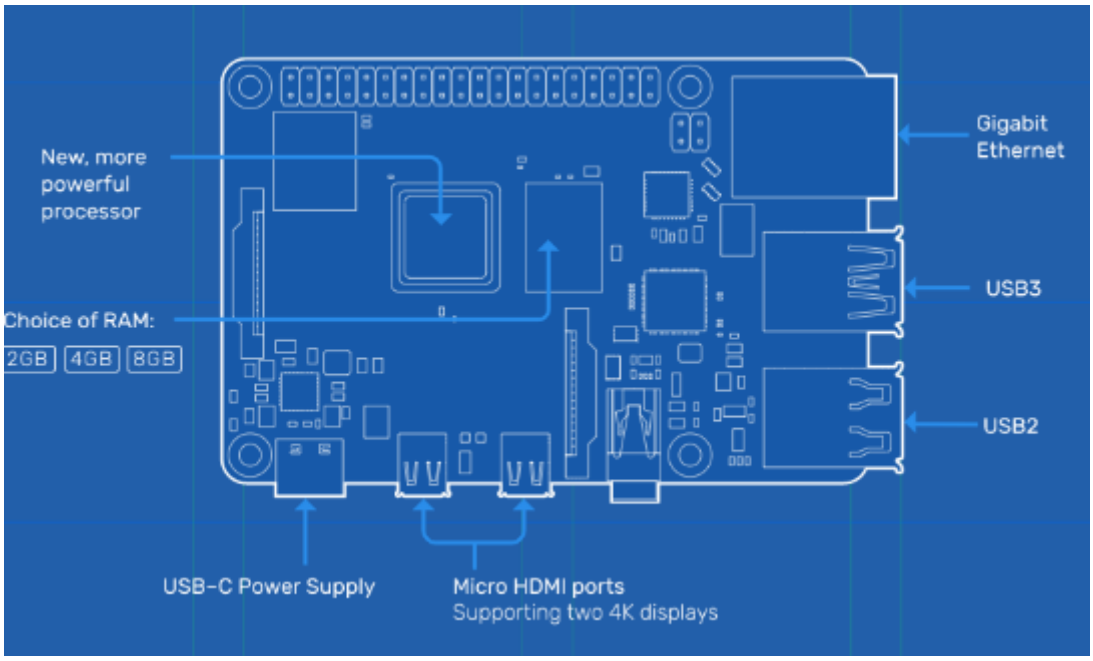
ה-Raspberry Pi 4 Model B הוא מחשב בעל לוח יחיד שפותח על ידי חברת Raspberry Pi.

התכונות העיקריות של Raspberry Pi 4 דגם B כוללות:

- מעבד ארבע ליבות: ה-Pi 4 מצויד במעבד Broadcom BCM2711 ארבע ליבות 64 Cortex-A72 (ARMv8-A) סיביות, המציע ביצועים משופרים בהשוואה לקודמיו.
- אפשרויות זיכרון: הוא זמין עם תצורות RAM שונות, כולל 2GB, 4GB ו- 8GB LPDDR4-3200 SDRAM, המספק למשתמשים גמישות רבה יותר עבור הפרויקטים והיישומים שלהם. עבור הרכיב שלנו יש לנו זיכרון RAM של 4GB וכרטיס SD בגודל 64 GB.
- קישוריות: הלוח כולל LAN אלחוטי בפס כפול (GHz2.4 ו-GHz5), Bluetooth 5.0, Gigabit Ethernet, שתי יציאות USB 3.0, שתי יציאות USB 2.0 ושתי יציאות מיקרו HDMI לתמיכה בצגים כפולים.
- פלט וידאו: ה-Pi 4 תומך בפלט וידאו K4, מה שהופך אותו למתאים ליישומי מולטימדיה ומשמש כמרכז בידור ביתי רב-תכליתי.
- Pins GPIO: הוא שומר על כותרת GPIO (כניסה/פלט לתכלית כללית) בעלת 40 פינים, המאפשרת למשתמשים להתממשק עם מגוון רחב של רכיבים אלקטרוניים וציוד היקפי.
- תמיכה ב-Power over Ethernet (PoE): ** ניתן להפעיל את ה-Raspberry Pi 4 Model B באמצעות Ethernet באמצעות כובע PoE (חומרה מחוברת למעלה), מה שמפשט את הגדרות אספקת החשמל.
- מחבר חשמל USB-C: ** ה-Pi 4 הציג מחבר מתח USB-C, שהחליף את מחבר המיקרו-USB ששימש בדגמים קודמים.

איור 5 - Raspberry Pi 4 Model B

ה-Raspberry Pi 4 Model B פופולרי ביישומים מקצועיים שונים, כולל אוטומציה ביתית, מרכזי מדיה, אחסון מחובר לרשת (NAS) ופריסות שרתים בקנה מידה קטן בשל הביצועים המשופרים והמורחבים שלו.



איור 6 - תרשים חיבורים של הRaspberry Pi

GPIO ולוח 40 הפינים:

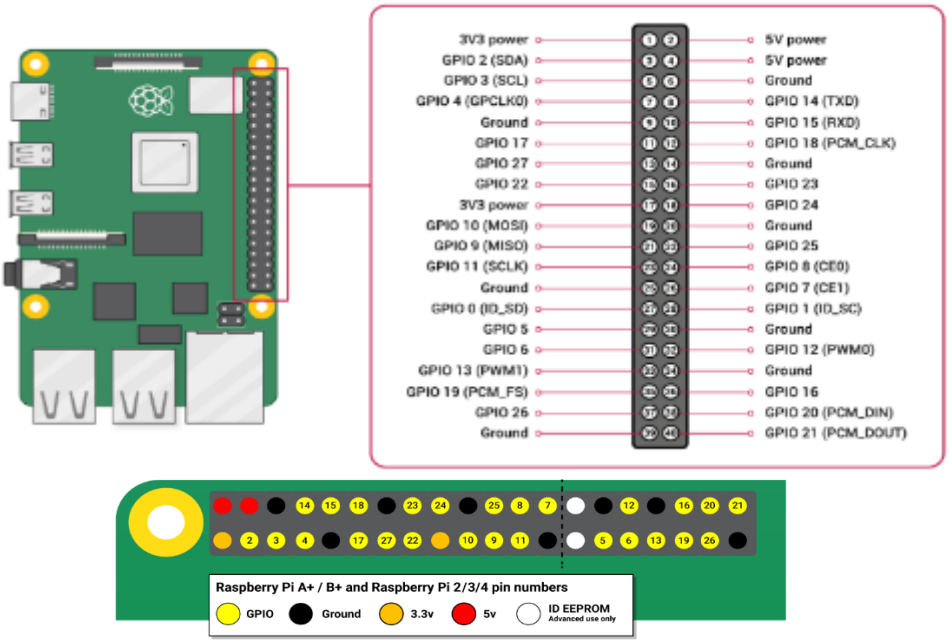
אחד הפיצ'רים העוצמתיים של ה Raspberry PI הוא שורת פיני ה-GPIO (כניסה/פלט לשימוש כללי) לאורך הקצה העליון של הלוח. תת הלוח GPIO של 40 הפינים נמצאת בכל הלוחות הנוכחיים של Raspberry Pi (לא קיים ב-Raspberry Pi Zero, Raspberry Pi Zero W ו- Raspberry Pi Zero 2 W). ללוח ה-GPIO יש גובה פינים של 0.1 אינץ' (2.54 מ"מ). כל אחד מפיני GPIO יכול להיות מוגדר (בתוכנה) כקלט או פלט וניתן להשתמש בו למגוון רחב של מטרות.

מתחים: על הלוח קיימים שני פינים של 5V ושני פינים של 3.3V, כמו כן גם מספר פינים של הארקה (0V) שאינם ניתנים להגדרה. הפינים הנותרים הם כל פיני 3.3V למטרות כלליות, כלומר היציאות מוגדרות ל 3.3V הכניסות הן 3.3V עד לערך משתנה.

מוצאים: ניתן להגדיר פין GPIO המיועד כפלט לרמה גבוהה (3.3V) או נמוכה (0V).

כניסות: פין GPIO המיועד כקלט יכול להיקרא כגבוה (3.3V) או נמוך (0V). זה נעשה קל יותר עם שימוש בנגדים פנימיים (Pull-up or Pull-down). לפינים GPIO2 ו-GPIO3 יש נגדי Pull-up קבועים, אך עבור פינים אחרים ניתן להגדיר זאת בתוכנה

מידע וחיבורים נוספים נמצאים בנספח 1



איור 7 - מקרא לוח פינים GPIO

סביבת העבודה שבה אנו עובדים הינה **Raspian OS**.

Raspian OS היא מערכת הפעלה מבוססת דביאן (Debian) שתוכננה במיוחד עבור מחשבי Raspberry Pi עם לוח יחיד. זוהי מערכת ההפעלה הרשמית למכשירי Raspberry Pi והיא מותאמת לארכיטקטורת ARM המשמשת במחשבים אלו.

מאפייניה של Raspbian OS כוללים:

- ממשק משתמש: Raspbian משתמשת באופן מסורתי ב-LXDE קל משקל (Lightweight X11 Desktop Environment) כסביבת שולחן העבודה המוגדרת כברירת מחדל. עם זאת, חלו שינויים, וגרסאות חדשות יותר עשויות להשתמש בסביבות שולחן עבודה שונות.
- ניהול חבילות תוכנה: Raspbian משתמשת במערכת ניהול החבילות של דביאן, מה שמקל על התקנה, עדכון והסרה של תוכנות באמצעות מנהל החבילות APT (Advanced Package Tool).
- ביצועים אופטימליים: Raspbian מותאם לארכיטקטורת ARM של ה-Raspberry Pi, מה שמבטיח ניצול יעיל של משאבים וביצועים טובים יותר במכשירים בעלי הספק נמוך אלה.
- תוכנה מותקנת מראש: Raspbian מגיעה בדרך כלל עם מגוון תוכנות מותקנות מראש, כולל כלי תכנות, משאבים חינוכיים, דפדפני אינטרנט ונגני מדיה, מה שהופך אותה לפלטפורמה רב-תכליתית לפרויקטים שונים.
- גרסאות שולחן עבודה ושרתים: Raspbian זמין הן בגרסת שולחן העבודה והן בגרסאות לייט (ללא ראש/שרת), המתאים למקרי שימוש שונים. גרסת הלייט מתאימה לפרויקטים שאינם דורשים ממשק משתמש גרפי.

- שילוב עם חומרת Raspberry Pi: Raspbian נועד להשתלב בצורה חלקה עם התכונות הייחודיות של חומרת Raspberry Pi, כגון פיני GPIO (General Purpose Input/Output), מודולי מצלמה וציוד היקפי אחר.

מערכת ההפעלה ממשיכה להתפתח, עם עדכונים ושיפורים שמסופקים על ידי חברת Raspberry Pi. ביצענו הורדה של מערכת ההפעלה ל-Raspberry Pi כמתואר בנספח 2.

<u>חיישנים שבהם נעשה שימוש בפרויקט:</u>

<u>LIDAR: Slamtec RPLIDAR A2M8 360° Laser Scanner</u>

מאפייניו של החיישן:

- סורק טווח לייזר 360 מעלות בעלות נמוכה
- תדר דגימה: 2000 - 8000 הרץ
- קצב סריקה: 5 - 15 הרץ
- טווח מרחק: 0.15 - 12 מ'
- רזולוציה זוויתית: 0.45 - 1.35°
- מחבר "חבית" זמין, מכלול מחבר חבית קטן 0.7x2.35 מ"מ

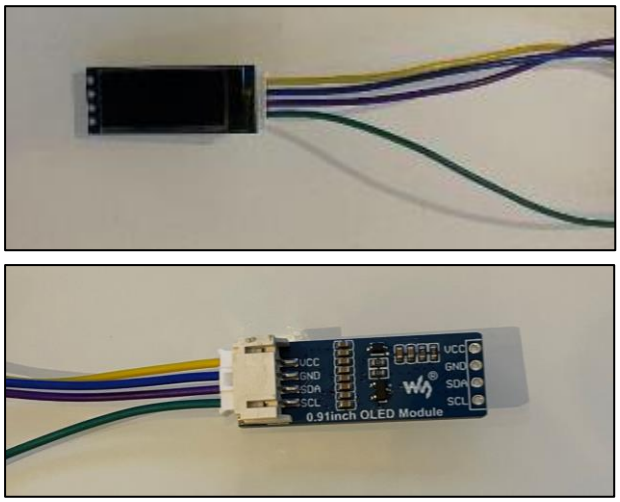
RPLIDAR A2 מאמצת מערכת מדידת טריאנגולציה בלייזר בעלות נמוכה שפותחה על ידי SLAMTEC, ולכן יש לו ביצועים מצוינים במגוון רחב של סביבות פנימיות וסביבות חיצוניות ללא חשיפה ישירה לאור שמש.

החיישן מסוגל להעביר עד 8000 דגימות לייזר בשנייה במהירות סיבוב מהירה, הוא משתמש בטכנולוגיית OPTMAG המוגנת בפטנט של SLAMTEC כדי לעבור את מגבלות תוחלת החיים של מערכות LIDAR קונבנציונליות, מה שמבטיח פעולה יציבה וממושכת.

איור 8- RPLidar

חיישן תצוגת OLED 0.91 אינץ'

צגי הOLED הם צגים בעלי ניגודיות גבוהה וברזולוציה גבוהה, כך שהם מספקים קריאה טובה למשתמשים. לא קיימת תאורה אחורית במסך הOLED והוא משתמש בפיקסלים מוארים באופן עצמאי, כך שהוא דק יותר ואלגנטי בהשוואה למסכי LCD. הdriver chip שלו הוא SSD1306, המספק תקשורת I2C. SSD1306 הוא מודול דרייבר למערכות תצוגה dot-matrix. נעשה בו שימוש עבור צגים "מסוג קתודה" והוא מתאים לישומים ניידים רבים כמו טלפונים ניידים, נגני MP3 ומחשבונים.



איור 9 - צג הOLED מלפנים ומאחור

למודול יש ארבעה פינים:

- **VCC**: Module power supply
- **GND**: Ground
- **SDA**: I2C data
- **SCL**: I2C clock

איור 10 - מקרא חיבורים של צג הOLED

שניים מהפינים הם עבור כוח (VCC וGND) והשניים האחרים הם עבור ממשק I2C (SDA and SCL). ישנם חיישנים מסוג זה שייתכן שיהיה צורך להלחים את החיבורים שלהם לחוטים מתאימים טרם השימוש.

Intel® RealSense™ Depth Camera D435

מצלמת Intel RealSense D435 היא מכשיר חישת עומק המיועד ליישומים שונים, החל מרובוטיקה ועד ראייה ממוחשבת. מצלמה זו היא כלי רב עוצמה עבור יישומי חישת עומק וראייה ממוחשבת, המספקת למפתחים את האמצעים ליצור פתרונות חדשניים הדורשים מודעות מרחבית מדויקת. תכונות מרכזיות של החיישן:

- חישת עומק: ה-D435 משתמש במצלמות סטריאוסקופיות כדי ללכוד מידע עומק בנוסף לתמונות RGB, מה שמאפשר תפיסת עומק מדויקת.
- רזולוציה גבוהה: הוא מספק תמונות צבעוניות ברזולוציה גבוהה (עד 1920x1080) ומפות עומק, מה שמאפשר לקבל נתונים חזותיים מפורטים ומדויקים.
- מודולי סטריאו: המצלמה משתמשת בזוג מודולי סטריאו הפועלים במקביל ליצירת ייצוג תלת מימדי של הסצנה, תוך שיפור ההבנה המרחבית.
- רבגוניות (ורסטיליות): עם העיצוב הקומפקטי שלו, ה-D435 מתאים לשימוש במקומות פנימיים (תחת קורת גג) ומקומות חיצוניים (צילומי חוץ) כאחד, מה שהופך אותו להתאמה למגוון רחב של יישומים.
- RealSense SDK: אינטל מספקת את RealSense SDK, המציעה סט מקיף של כלים וספריות למפתחים כדי למנף את יכולות המצלמה ביישומים שלהם.
- דיוק עומק: למצלמה יש יכולת ללכוד מידע עומק מדויק, מה שמקל על משימות כמו זיהוי עצמים, בקרת מחוות וסריקה תלת ממדית.
- התאמה אישית: משתמשים יכולים להתאים אישית את הגדרות המצלמה כגון חשיפה, רווח וטווח עומק כדי לייעל את הביצועים למקרי שימוש ספציפיים.
- תמיכה ב-SDK: ה-RealSense D435 נתמך על ידי שפות תכנות שונות דרך ה- RealSense SDK, מה שהופך אותו לנגיש למפתחים המשתמשים בשפות כמו Python ,++C ועוד.

איור 11 - מצלמת Intel RealSense D435

אלגוריתמי הבדיקה בעבור הסנסורים השונים:

אלגוריתמי הבדיקה של המצלמה:

- אלגוריתם חישוב המרחק של הפיקסל האמצעי בתמונה הנלקחת במצלמה:

(ממומש על ידי הסקריפט: Calculate_distance_center.py)

1. אתחול הגדרות המצלמה: יצירת צינור של RealSense והגדרת פרמטרי צבע ועומק. הפעלת pipline וייצוב המצלמה.
2. התאם חשיפת המצלמה: זיהוי החיישן הפעיל והגדרת החשיפה לערך ספציפי.
3. לכידת מסגרת: המתנה לקבוצה קוהרנטית של פריימים ממצלמת RealSense. אחזור מסגרות הצבע והעומק.
4. תמונת צבע לפני עיבוד: המרת את מסגרת הצבע למערך NumPy. החלת תיקון גמא כדי לשפר את בהירות התמונה והניגודיות. השוואת ההיסטוגרמה בנפרד בכל ערוץ צבע על מנת לשפר את האיכות החזותית.
5. חישוב מרחק: קבלת הממדים של תמונת הצבע המעובדת. קביעת הקואורדינטות המרכזיות של התמונה. השגת ערך המרחק במרכז באמצעות מסגרת העומק.
6. מידע על מרחק שכבת-על: הוספת טקסט לתמונה הצבעונית המעובדת, המציין את המרחק למרכז במטרים. הצגת טקסט המרחק במיקום מוגדר.
7. שמירת תמונה מעובדת: שמירת את תמונת הצבע הסופית עם מידע על המרחק בנתיב קובץ מוגדר (פורמט JPEG). הדפסת הודעה המאשרת את שמירת התמונה בהצלחה.
8. ניקוי וסיום: עצירת הזרמת מצלמת RealSense עבור שחרור משאבים. ציון את הנתיב לשמירת התמונה המעובדת וקריאה לפונקציה capture_color_image_with_distance בעבור ביצוע תהליך לכידת התמונה ושמירתו.

- אלגוריתם חישוב המרחק של הפיקסל עליו נמצא העכבר בזמן אמת בתמונה:

(ממומש על ידי הסקריפט: detect_distance_in_meters.py)

1. חידוש הגדרות המצלמה: יצירת מופע של המחלקה DepthCamera מהמודול realsense_depth. הגדרת נקודת עניין ראשונית (ברירת מחדל: (400, 300)).
2. יצירת אירוע עכבר: הגדרת פונקציית (show_distance) callback כדי לעדכן את נקודת המשתנה הגלובלית בהתבסס על לחיצות עכבר על התמונה המוצגת. שיוך פונקציית ההתקשרות חזרה לחלון "Color frame" באמצעות פונקציית setMouseCallback של OpenCV.
3. לולאה ראשית - אינטראקציה בזמן אמת: לולאה אינסופית לאינטראקציה בזמן אמת. צילום פריים ממצלמת RealSense בשיטת get_frame.
הצגת התמונה בחלון בשם "Color frame". הצגת עיגול אדום בנקודת העניין שצוינה (מיקום במעבר העכבר).
4. חישוב והצגת מרחק: חילוץ מסגרת העומק מהמסגרת שנלכדה. אחזור ערך העומק (במילימטרים) בנקודה שצוינה והמרתו למטרים. הצגת מידע המרחק בטקסט ירוק על מסגרת הצבע, בסמוך לנקודה.
5. אינטראקציה עם המשתמש: בדיקה באופן מתמיד אם יש אירוע של לחיצה על מקשים (cv2.waitKey(1)). אם מקש 'Esc' (27) נלחץ, יציאה מהלולאה האינסופית.

6. סיום: סגירת את כל חלונות OpenCV. סיום תהליכי מצלמת RealSense.

- <u>זיהוי אובייקטים במרחק מוגדר והתראה בעת הזיהוי:</u>

(ממומש על ידי הסקריפט: gesture_recognition_alert.py)

1. אתחול RealSense Pipeline: קביעת תצורה של זרמי עומק וצבע עם פרמטרים ספציפיים.
2. הגדרת מרחק המטרה לזיהוי אובייקט.
3. ביצוע לולאה ראשית: לולאה אינסופית לעיבוד מסגרת בזמן אמת.
4. המתנה לזוג מסגרות קוהרנטי (עומק וצבע). אחזור מסגרות העומק והצבע ממצלמת RealSense.
5. המרת מסגרת: המרת מסגרת העומק למערך numpy להדמיה וכן המרת מסגרת הצבע למערך numpy להמשך עיבוד.
6. חישוב מרחק אובייקט: חילוץ ערך העומק במרכז התמונה הצבעונית. בדיקה אם האובייקט נמצא במרחק הרצוי.
7. תצוגת התראת קרבה: אם האובייקט נמצא במרחק היעד: הצגת "התראת קרבה" על גבי המסך ואת המרחק באמצע התמונה הצבעונית.
8. אינטראקציה עם המשתמש: שבירת הלולאה כאשר מקש 'q' נלחץ.
9. ניקוי וסיום: עצירת RealSense pipeline על מנת לשחרר משאבים. סגירת כל חלונות OpenCV.

<u>אלגוריתמי הבדיקה של חיישן הLIDAR:</u>

- <u>אלגוריתם יצירת הדמיה בזמן אמת של נתוני סריקת LIDAR.</u>

(ממומש על ידי הסקריפט 2D_scan.py)

1. אתחול Pygame ותצוגה: הגדרת את סביבת Pygame הגדרת התצוגה לרזולוציה הנדרשת. צור משטח (LCD) Pygame עם רקע שחור. עדכן את התצוגה כדי להציג את המצב ההתחלתי.
2. הגדרת תצורת RPLidar: הגדרת שם הפורט של RPLidar ואתחול האובייקט RPLidar עם הפורט שצוין.
3. הגדר נתונים להדמיה: הגדרת משתנה לשינוי קנה מידה של נתוני lidar. יצירת משטח Pygame עם נקודות lidar המיוצגות כפיקסלים לבנים. עדכון התצוגה על מנת להמחיש נתוני Lidar בזמן אמת.
4. לולאה ראשית: חזרה ברציפות על סריקות לידר. עדכון רשימת scan_data עם זוגות מרחק-זווית מכל סריקה. יצירת ויזואליציה של נתוני lidar על משטח Pygame באמצעות קואורדינטות קרטזיות. ביצוע בדיקה האם יש אירועי Pygame. הפסקת הסקריפט נעשית אם מקש Escape נלחץ, ותהיה יציאה מהלולאה.
5. ניקוי וסיום: ביציאה מהלולאה: עצירת ה-lidar והמנוע. ניתוק מכשיר הלידר. יציאה מסביבת Pygame.

- <u>אלגוריתם שימוש בנתוני LIDAR עבור זיהוי אובייקטים קרובים והתרעה למשתמש:</u>

(ממומש על ידי הסקריפט lidar_distance.py)

1. הגדרת Lidar: אתחול את ה-RPLidar עם שם הפורט שנבחר.
2. פונקציית קריאת נתונים של Lidar: חזרה ברציפות על סריקות lidar. סינון מדידות לא חוקיות או לא אמינות בהתבסס על טווח מרחק. אחסון צמדי זווית-מרחק חוקיים ברשימת הנתונים. הפסקת הלולאה לאחר הסריקה המלאה הראשונה.
3. פונקציית בדיקת מרחק: מעבר איטרטיבי על נתוני הלידר ובדיקה אם מרחק כלשהו נמצא מתחת לסף שצוין. החזרת True אם אובייקט קרוב מדי (על פי הסף שנקבע). אחרת, החזר False.
4. פונקציית הפעלת התראה: הדפסת הודעה המציינת שאובייקט קרוב מדי.
5. לולאה ראשית: אתחול מונה למעקב אחר אובייקטים שזוהו. קריאה ברציפות את נתוני הלידר ובדיקת מרחקים (באמצעות הפונקציות שצוינו קודם לכן). אם חפץ קרוב מדי (מרחק מתחת לסף), מתבצעת הפעלת ההתרעה והדפסה של ערך המונה. לבסוף מתבצעת הגדלת ערך המונה עבור כל אובייקט שזוהה.
6. ניקוי וסיום: עצירת הלידר והמנוע שלו.

ניתוח תוצאות

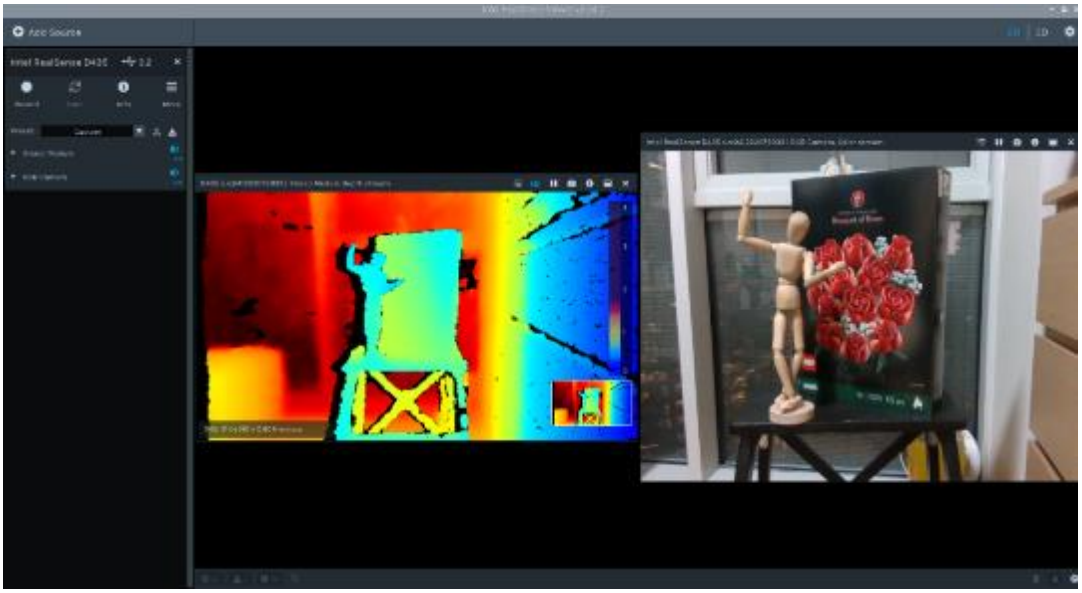
תוצרים ובדיקות- מצלמת Intel realsense d435

- על מנת לראות את תוצר המצלמה הגולמי ומפת המרחקים שהמצלמה מציגה נפתח את הטרמינל ונריץ את הפקודה:

```
realsense-viewer
```

- בשלב זה תעלה תוכנת intel realsense viewer אשר פותחת את המצלמה ומציגה תמונה ב2D ו3D וכמובן מציגה depth image. ניתן לראות שהכל עובד כראוי, הרזולוציה והתמונות טובות, המרחקים הגיוניים.

המסך: מצד ימין מופיע העצם המצולם- קופסת קרטון ולצידה דמות עץ). מצד שמאל מופיעה מפת המרחקים לפי צבעים והמקרא בצידה הימני. כפי שניתן לראות, דמות העץ ממוקמת לפני קופסת הקרטון וקופסת הקרטון ממוקמת בהטיה מרחקית מהמצלמה ועל כן גרדיאנט הצבעים ניכר שכן יש הבדלי מרחקים בין קצוות הקופסה.



איור 12- תוכנת intel realsense viewer: מימין- מה שרואה המצלמה, משמאל- מפת מרחקים כך שבכחול העצמים שנמצאים במרחק הקרוב ביותר ובאדום העצמים שנמצאים במרחק הרחוק ביותר

בנוסף, מצורפים הסקריפטים שכתבנו בpython שנמצאים בתיקייה הבאה:

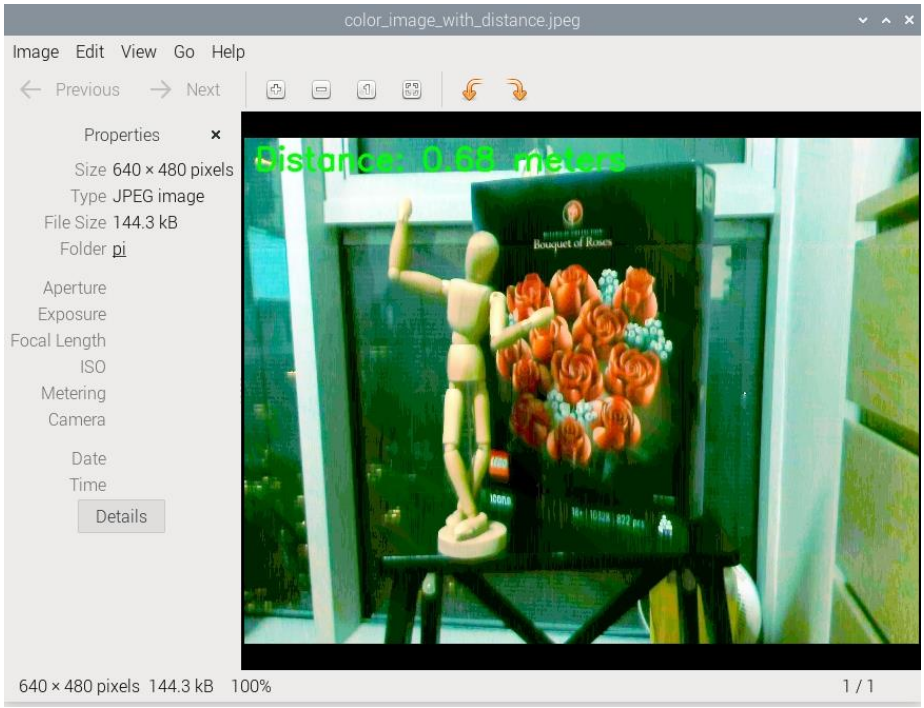
```
home/pi/camera_test
```

- תחילה, על מנת לגשת ולעשות שימוש בסקריפטים נסגור את תוכנת intel realsense viewer שכן היא עושה שימוש בלעדי במצלמה כאשר היא פתוחה.
- נשתמש בפקודות הבאות

```
Cd camera_test
Python3 <ScriptName>.py
```

- Calculate_distance_center.py:

סקריפט Python זה משתמש במצלמת Intel RealSense D435, OpenCV ו- RealSense SDK כדי ללכוד תמונה צבעונית תוך מתן מידע מרחק למרכז התמונה. הוא מכוון את החשיפה, מחיל תיקון גמא והשוואת היסטוגרמה, מחשב מרחק, מציג אותו על גבי התמונה ושומר את התוצאה הסופית. סקריפט זה מתאים ליישומים הדורשים נתונים חזותיים ומדידות מרחק משופרים. התוכנית שומרת את התמונה ב home\pi עם הצגה של המרחק של הפיקסל הממוקם במרכז התמונה.



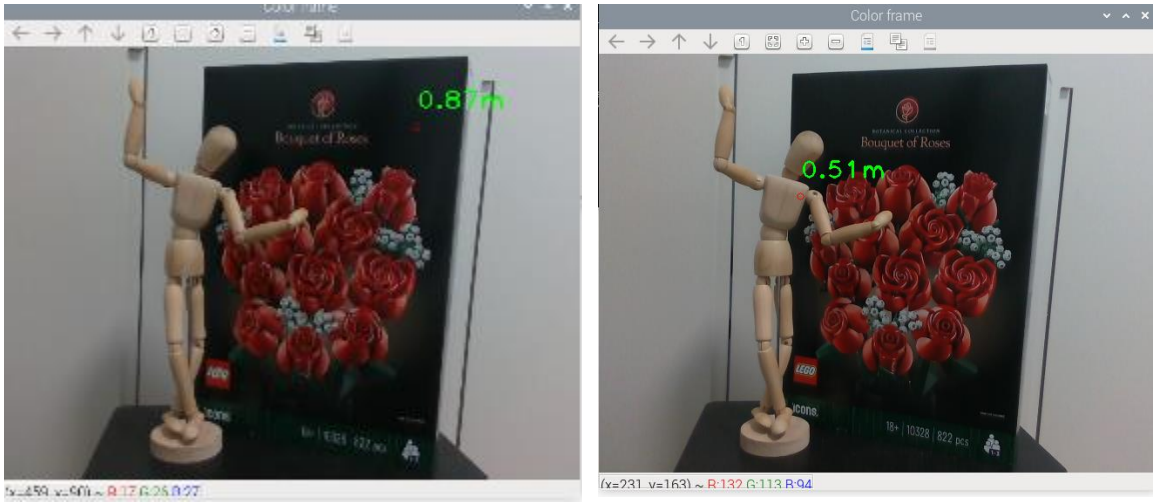
איור 13 - תוצר התוכנית Calculate_distance_center.py- מרחק הנקודה שממוקמת בפיקסל המרכזי

- detect_distance_in_meters.py:

סקריפט Python זה משתמש במצלמת Intel RealSense D435 ובמודול 'realsense_depth' כדי לספק מידע עומק בזמן אמת עבור נקודה שנבחרה על ידי המשתמש במסגרת הצבע. הוא מאתחל את מצלמת RealSense, פותח חלון אשר מציג את התמונה שהמצלמה מצלמת בזמן אמת, יוצר אירוע עכבר לבחירת נקודה, ומעדכן באופן רציף את מידע העומק המוצג וממיר את הערכים ממילימטרים למטרים עבור נוחות המשתמש. הנקודה שנבחרה (שהעכבר ממוקם בה בתצורת hover- ריחוף) מסומנת בעיגול אדום, והמרחק לנקודה זו מוצג בירוק על מסגרת הצבע. משתמשים יכולים לחקור באופן דינמי מדידות עומק במיקומים שונים, מה שהופך אותו לכלי בעל ערך לניתוח עומק אינטראקטיבי והבנת המאפיינים המרחביים של סצנה.

סקריפט זה מייבא את הסקריפט realsense_depth.py (שנמצא באותה תקייה) ששם מוגדר קלאס עם מתודות המשמשות אותנו. נעזרנו בבניית טסט זה במדריך הבא:

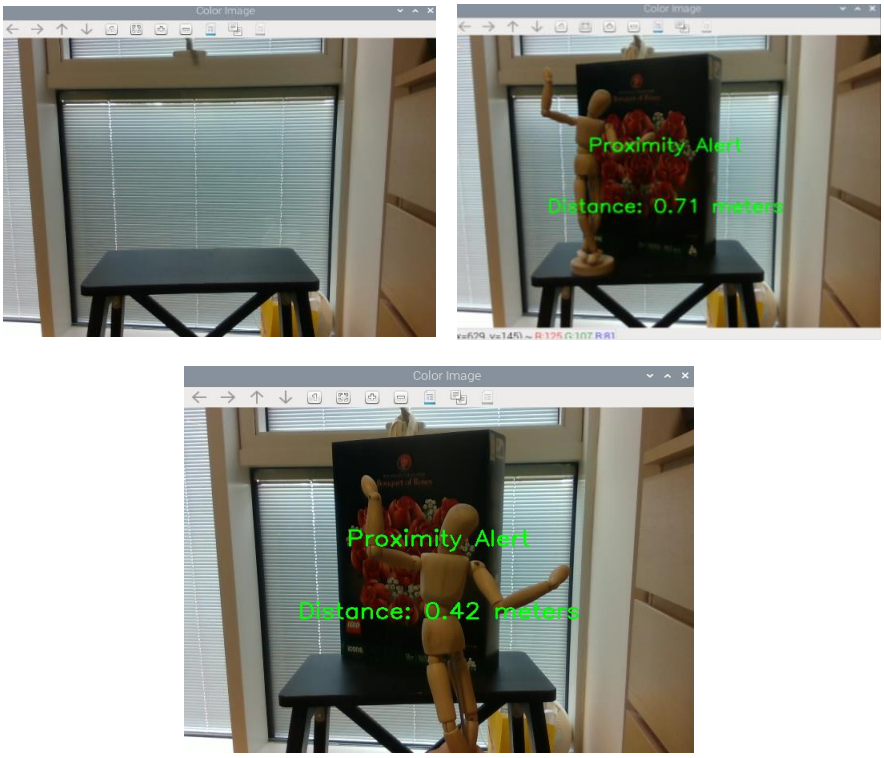
https://pysource.com/2021/03/11/distance-detection-with-depth-camera-intel-realsense-d435i/



איור 14- תוצר התוכנית detect_distance_in_meters.py- מציג מרחק של הנקודה שעליה העכבר מצביע

- gesture_recognition_alert.py :

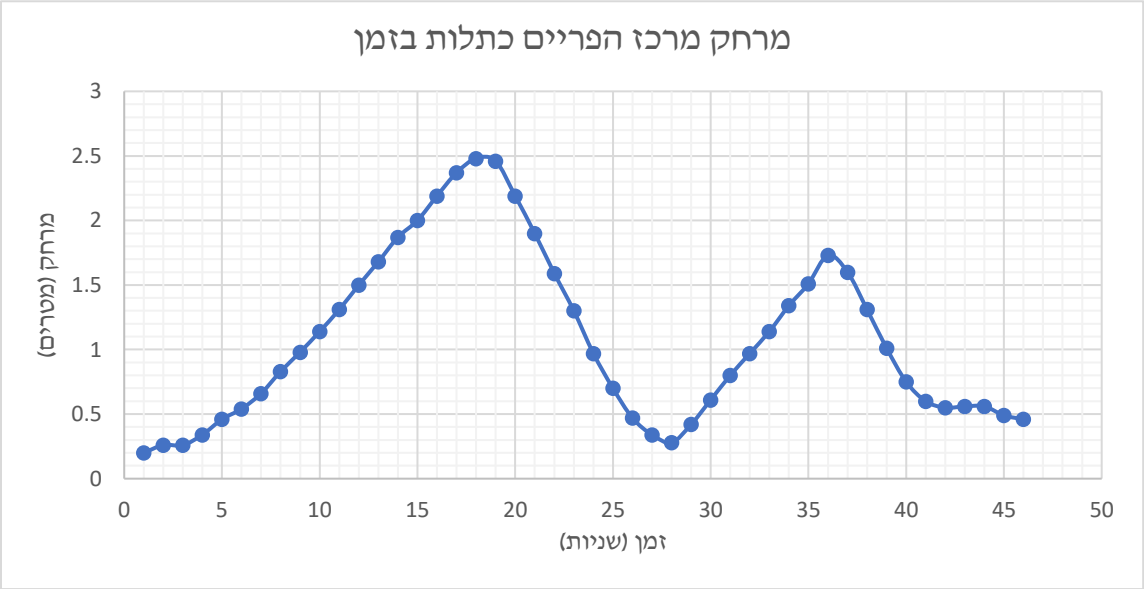
סקריפט Python זה משתמש במצלמת Intel RealSense D435 על מנת לזהות אובייקטים במרחק מוגדר. תחילה, בעת הרצת הסקריפט נפתח חלון ובו מוצגת התמונה אותה מצלמת המצלמה בזמן אמת. כאשר אובייקט נמצא בטווח המוגדר (0.75 מטר בעבור הדוגמה המוצגת), הוא מציג הודעת "התראת קרבה" ("Proximity Alert") יחד עם המרחק לאובייקט באמצע המסך. ספריית OpenCV משמשת לעיבוד תמונה ועיבוד טקסט, ומאפשרת לסקריפט להציג את המסר האינפורמטיבי על עדכון המצלמה בזמן אמת. משתמשים יכולים להתאים את מרחק היעד ואת האלמנטים החזותיים כגון גודל גופן וצבע כך שיתאימו להעדפותיהם. הסקריפט תוכנן להיות אינטואיטיבי וניתן להתאמה עבור יישומים שונים הדורשים זיהוי קרבה והתראות משתמשים.



איור 15- תוצר התוכנית gesture_recognition_alert.py- התראה כאשר העצם במרכז התמונה קרוב מ0.75 מטרים

- Center_depth_print.py

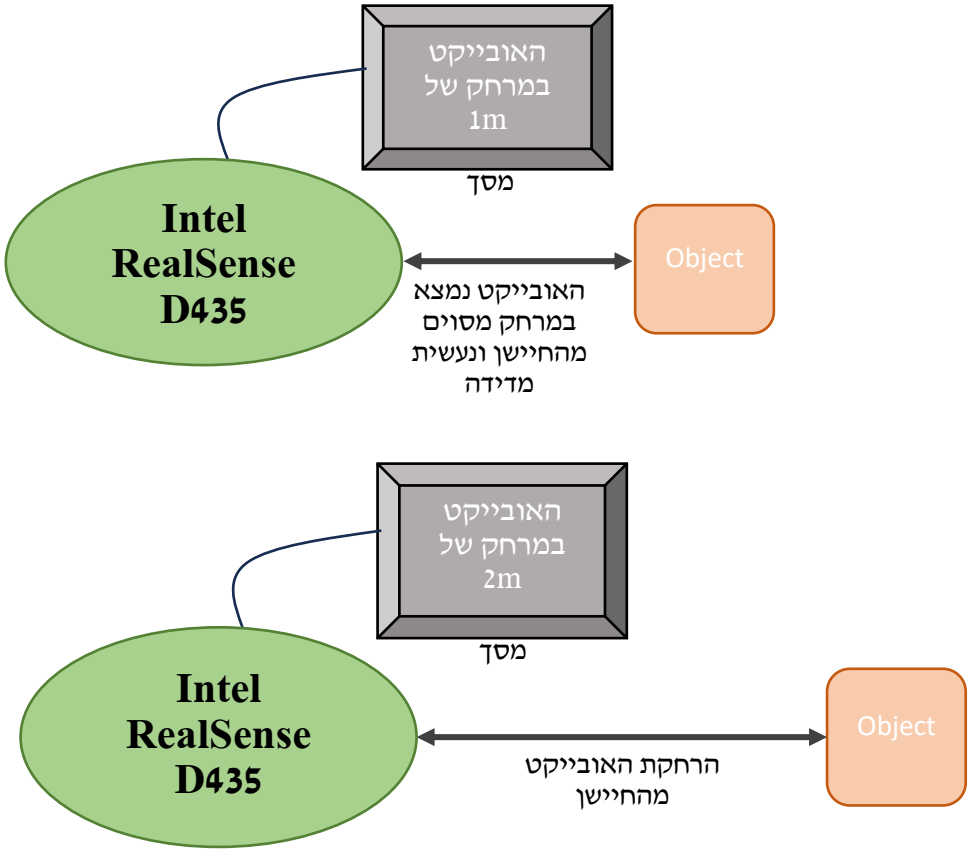
סקריפט Python זה משתמש במצלמת Intel RealSense D435 כדי לספק מידע עומק בזמן אמת עבור הפיקסל האמצעי שהמצלמה מצלמת. הסקריפט מדפיס לטרמינל בכל שניה (מסגרת הזמן ניתנת לשינוי) את המרחק במטרים שבו נמצא העצם שמצולם במרכז הפריים (Frame). נציג את התוצאות עבור ניסוי שנערך בגרף הבא:



גרף 1- תוצר התוכנית Center_depth_print.py - מרחק מרכז הפריים כתלות בזמן מול אובייקט שמתקרב ומתרחק מהמצלמה

ניתן לראות כי העצם החל קרוב למצלמה ובמהלך הזמן הלך והתרחק, לאחר 17 שניות העצם התקרב שוב, לאחר 28 שניות העצם התרחק מהמצלמה ולבסוף בנקודת ה36 שניות הוא התקרב עד תום הניסוי.

תוצאות הניסוי המלאות (ערכים מדויקים) נמצאים בטבלה בנספח 3.



איור 16- הדגמה של ניסוי תיעוד המרחקים כתלות בזמן תוך שימוש בתוכנית Center_depth_print.py

תוצרים ובדיקות עבור חיישן הRPLidar

בתחילת כל שימוש בחיישן נצטרך להפעיל את הסביבה הוירטואלית. הפקודות לכך מופיעות בנספח 4.

- ניתן לראות את הסקריפטים שנכתבו בPython בתיקיה `lidar_test` : נשתמש בפקודה הבאה על מנת לגשת בסקריפטים בתיקיה :

```
Cd /home/pi/lidar_test
```

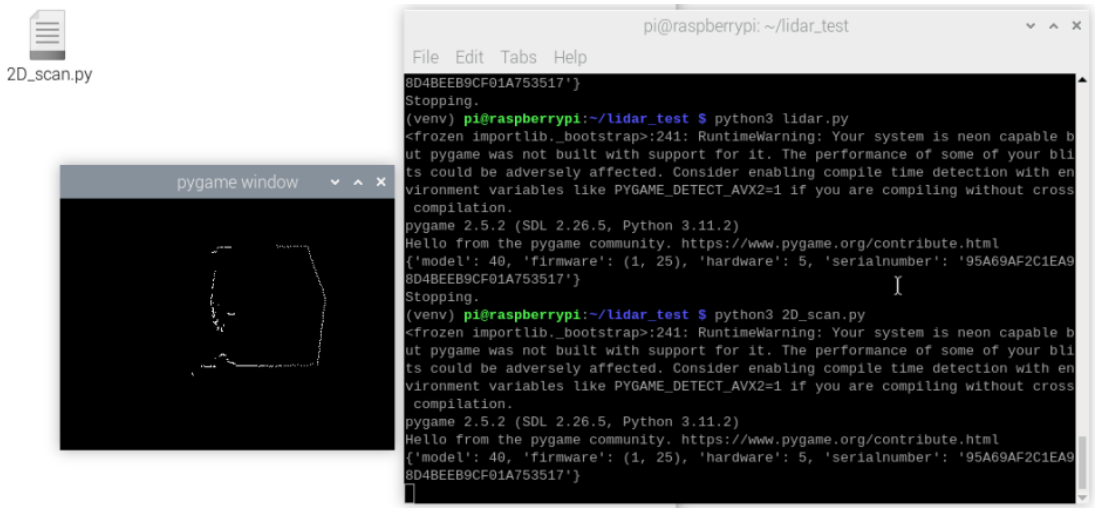
- נריץ את הסקריפטים באמצעות הפקודה :

```
Python3 <ScriptName>.py
```

- 2D_scan.py :

סקריפט זה יוצר הדמיה בזמן אמת של נתוני סריקת LIDAR בחלון pygame, ומאפשר למשתמשים לנטר את הסביבה הסובבת באמצעות טכנולוגיית LIDAR.
משתמש ב-Python ובספריות שונות כדי להגדיר ולשלוט בחיישן LIDAR למדידת מרחק.
הסקריפט נועד לרוץ על צג עם ממדים 320x240 פיקסלים ומגדיר את תצוגת ה-pygame בהתאם.
במידה ונרצה לשנות את גודל המסך ואת הגודל של סריקת החדר על גבי המסך זה ניתן לשינוי בסקריפט את הפרמטר "SCALING_FACTOR" .
הפונקציה העיקרית של הסקריפט, בשם `process_data`, מעבדת נתוני סריקת LIDAR ומציגה אותם בחלון ה-pygame. הוא מחשב את הקואורדינטות הקרטזיות של כל נקודת נתונים של LIDAR בהתבסס על הזווית והמרחק שלה, ואז ממפה את הנקודות הללו על המסך. התצוגה מתעדכנת בזמן אמת עם קבלת נתוני סריקת LIDAR חדשים.
הסקריפט משתמש בספריית RPLidar כדי לתקשר עם חיישן LIDAR ולאחזר נתוני סריקה. הוא מגדיר את חיישן LIDAR ומאתחל משתנים, כגון `max_distance`, המשמש לשינוי קנה המידה של הנתונים כך שיתאימו לתצוגה. הסקריפט מדפיס מידע על חיישן LIDAR ועל נתוני הסריקה, ומעדכן את התצוגה בזמן אמת עם המידע המעובד.

נראה תוצרים על שני חדרים שונים : הדמיה עליונה- חדר ראשון, הדמיה תחתונה- חדר שני.



איור 17- הדמיית שני חדרים שונים ממבט על באמצעות LPLIDAR

- lidar_distance.py :

סקריפט Python זה משתמש בספריית rplidar כדי להתממשק עם חיישן LIDAR ולנטר את הסביבה הסובבת. הפונקציונליות העיקרית של הסקריפט מוגדרת בפונקציות. הפונקציה read_lidar_data לוכדת נתוני LIDAR, מסננת מדידות שנופלות מחוץ לטווח מרחק מוגדר. הפונקציה check_distance מעריכה אם אובייקט כלשהו קרוב מדי על סמך מרחק סף מוגדר מראש. אם אובייקט מזוהה בטווח שצוין, הפונקציה trigger_alarm נקראת, המציינת מצב שעלול להיות מסוכן.

בלולאה הראשית, הסקריפט עוקב ברציפות אחר נתוני LIDAR, בודק אובייקטים קרובים. אם אובייקט מזוהה בתוך מרחק הסף שהוגדר, ישנה התראה, ומונה מוגדל. הלולאה נקטעת אם המשתמש לוחץ על Ctrl+C, ומבוצעות הליכי ניקוי כדי לעצור את חיישן LIDAR, לעצור את המנוע שלו ולהתנתק ממנו.

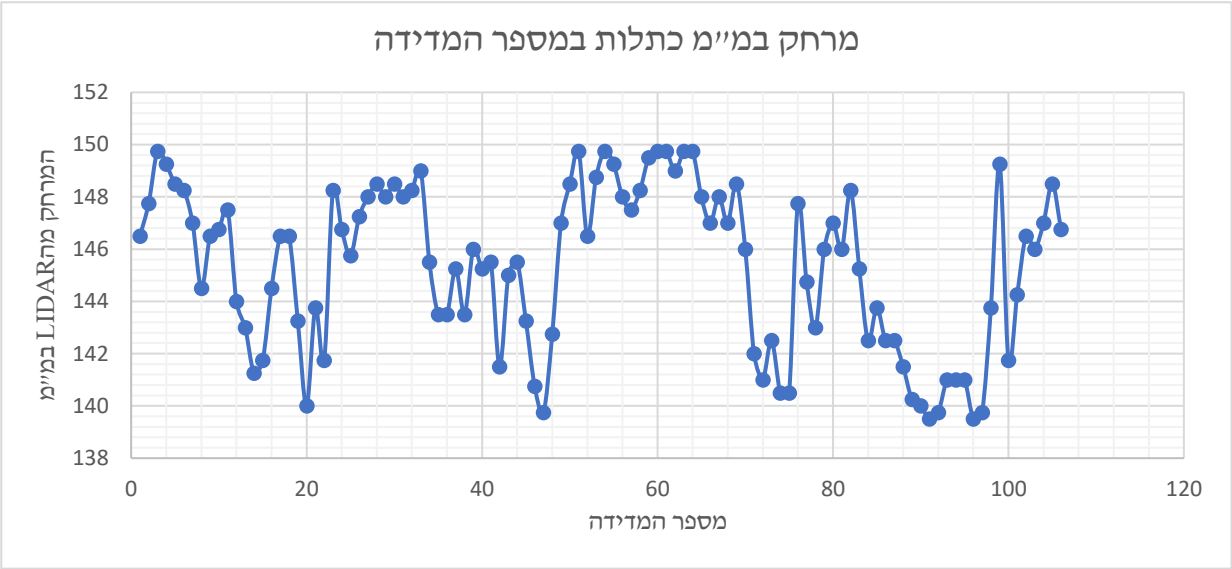
סקריפט זה הינו דוגמה בסיסית לשימוש בנתוני LIDAR על מנת לזהות אובייקטים קרובים ולהתריע למשתמש, מה שהופך אותו למתאים ליישומים כמו חישת קרבה או הימנעות מהתנגשות. ניתן לבצע התאמות למרחק הסף ולפרמטרים אחרים כדי להתאים אישית את הסקריפט בהתבסס על דרישות ספציפיות.

עבור ההתאמה שנעשתה לבדיקות שלנו :

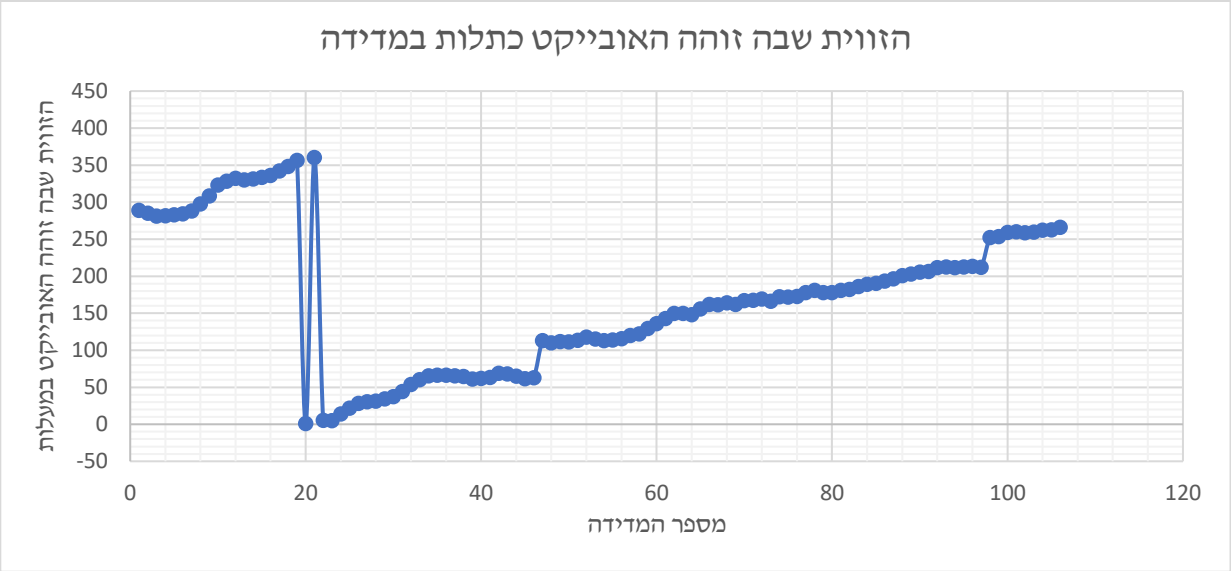
הסקריפט מתריע אם קיים אובייקט שנמצא קרוב מ 150 מ״מ . הוא מתריע באמצעות הדפסה למסך של ״object is too close! Trigger alarmed״ . הסקריפט בנוסף מבצע ספירה של כמות הפעמים שהאירוע מתרחש. צילום מסך של האזהרה נמצא בנספח 5.

עבור סקריפט זה נעשתה התאמה נוספת לניסוי והיא מופיעה בסקריפט: lidar_distance_w_values.py

התאמה זו בעצם מדפיסה את הזווית שבו אותר האובייקט שגרם להתראה וכן גם את המרחק המזוהה בעת ההתראה. ניתן לראות את התוצאות בגרפים הבאים:



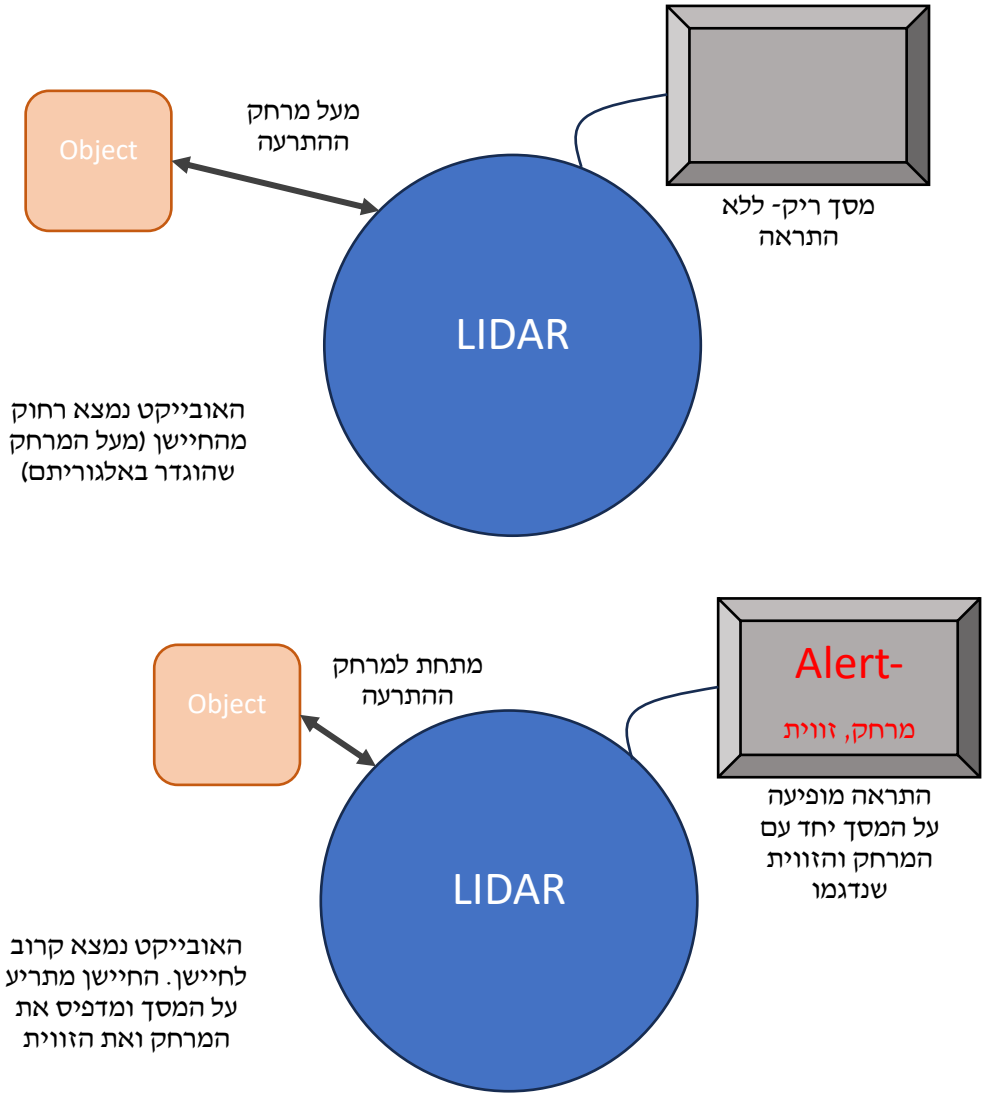
גרף 2- תוצר התוכנית lidar_distance.py- הצגת המרחקים הנדגמים בעת קירוב אובייקט לחיישן מתחת למרחק של 15 ס״מ (150 מ״מ)



גרף 3- תוצר התוכנית lidar_distance.py- הצגת הזווית שבה נמצא האובייקט בעת דגימה שנעשית כאשר האובייקט מתקרב לחיישן במרחק מתחת ל15 ס״מ (150 מ״מ)

ניתן לראות כי האובייקט התקרב למרחקים שונים אך המערכת התריעה ונכתבה מדידה בכל פעם שהאובייקט היה קרוב לחיישן במרחק קצר מ150 מ״מ (15 ס״מ). בנוסף, האובייקט הוזז מסביב לחיישן כסיבוב אחד (כ-360 מעלות) על מנת להראות את תיעוד הזווית בה זוהה ולכן ניתן לראות בגרף את זווית הימצאות האובייקט בעת המדידות.

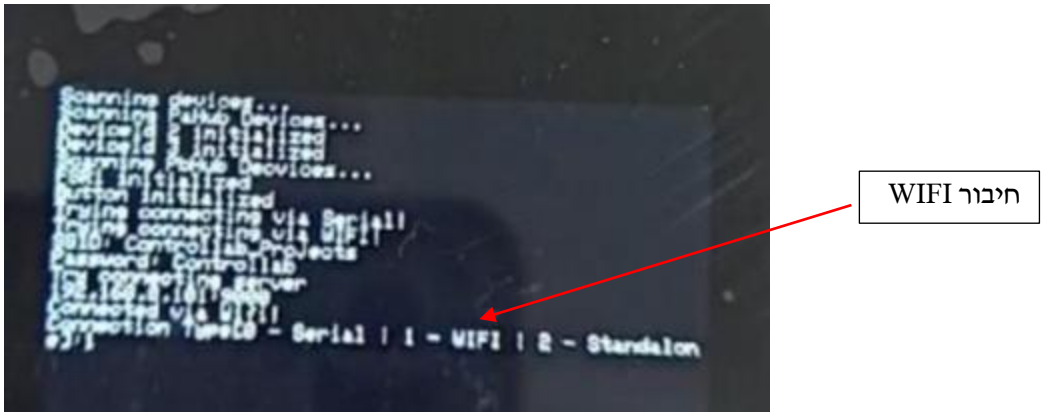
ערכים מדויקים למבדק זה מופיעים בנספח 6.



איור 18- הדגמה של ניסוי תיעוד המרחקים וזווית הימצאות האובייקט כתלות בדגימה שנעשית כאשר האובייקט קרוב לחיישן במרחק שקטן מ150 מ״מ. שימוש בתוכנית lidar_distance.py

חיבור עם M5stack Arduino

בעבור החיבור עם M5stack, החיבור שמבוצע בין שתי התשתיות הוא בתצורת WIFI. ניתן לראות כי החיבור הצליח.



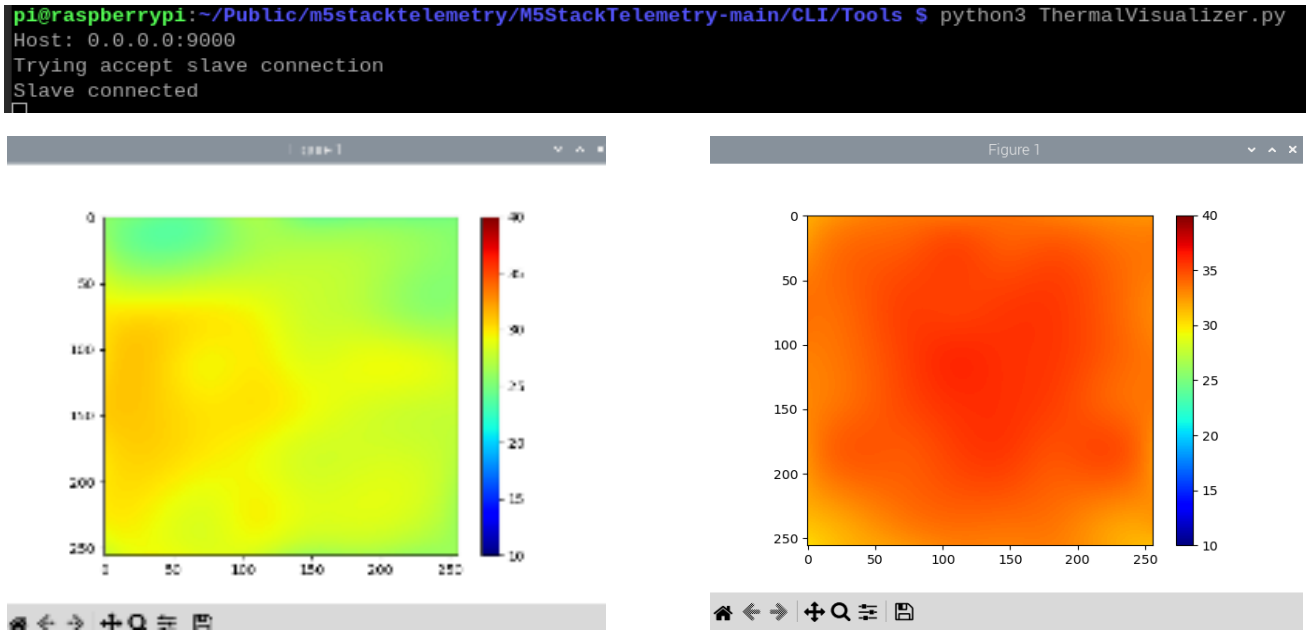
איור 19- חיווי של ביצוע חיבור WIFI עם הArduino

איור 20- חיווי על הצלחת החיבור בין הRaspberry Pi לArduino על גבי צג הArduino

כמו שניתן לראות בRaspberry PI משמש כמאסטר, ונעשית סריקה של המכשיר וחיבור בתצורת הWIFI. על גבי התצוגה ב Raspberry PI אפשר לראות את המוצאים של הסנסורים של ה M5stack- צילום מסך של תוצאה זו ופירוט מופיע בנספח 7.

הרצת הסקריפט ThermalVisualizer.py :

בעבור מצב שבו חיישן הטמפרטורה לא סרק אובייקט בטמפרטורה גבוהה (שמאל) לעומת סריקת אובייקט בטמפרטורה גבוהה (ימין). השינויים נעשו בזמן אמת.



איור 21- תוצר בדיקת חיישן הטמפרטורה והצגת התוצר בRaspberry Pi : מימין- אובייקט חם שנמצא על החיישן ועל כן ניתן לראות גוון אחיד אדום על גבי המסך, משמאל- הסרת האוייבקט החם מהחיישן והתקררות החיישן- ועל כן הצגת צבע יותר ירוק

סיכום, מסקנות והצעות להמשך

בפרויקט זה נדרשנו לבנות תשתית חומרתית ותוכנתית מבוססת מחשב Raspberry Pi שתהווה תשתית לפרויקטים עתידיים בהנדסת חשמל בשילוב עם פרוייקטים של פיזיותרפיה. במהלך העבודה על הפרויקט נדרשנו לקבוע את החיישנים איתם הכי מתאים לעבוד ושהחיבוריות שלהם לRaspberry Pi תהיה מיטבית. לאחר מחקר זה ביצענו רכש של כלל הסנסורים ולאחר שאלו הגיעו לידינו ביצענו את עבודת ההתקנה והבדיקות בתצורה של עבודה והתעמקות בכל חיישן בנפרד. למדנו לבצע חיבור בין הRaspberry Pi לבין החיישנים השונים, יצרנו תשתית התקנה עבור כל אחד מהחיישנים וביצענו מבדקים שונים על כל אחד מהחיישנים אשר מראים את הפונקציונליות שלטעמנו תבוא לידי ביטוי בצורה השכיחה ביותר בעת השימוש העתידי בתשתית. התשתית מסוגלת להכיל את החיבורים למצלמה, לLIDAR, למסך הOLED, למסך בחיבור הHDMI, למקלדת, לעכבר, לM5stack בחיבור WIFI בבת אחת. על כן התשתית יכולה לעמוד בזכות עצמה וניתן לעשות בה שימוש בפרויקטים עתידיים לצרכים שונים שיכולים לפתח שימוש בכל החיישנים יחד ושימוש בכל אחד בנפרד.

עבור המצלמה, הראנו את תצורת השימוש בתוכנה של Real-sense אשר מציגה את התמונה לצד מפת המרחקים שנתונה על ידי סקאלה של צבעים, מבדק שמראה את המרחק של הנקודה האמצעית שקולטת המצלמה וכן מבדק שמתריע שנקודה זו קרובה ממרחק מסוים.

עבור חיישן הLIDAR הראנו את תצורת הסריקה שלו והצגת התוצר החזותי של המרחב אותו הוא סורק בתמונה "עילית" והראנו מבדק שבו החיישן מתריע על התקרבות של עצם מתחת למרחק מסוים לבחירתנו.

עבור מסך הOLED ניתן לראות את תצורת החיבור שלו ולהדגים תצורות הצגה שונות בו.

עבור החיבור עם הM5stack אמנם לא הייתה הצלחה בביצוע החיבור הסיריאלי כפי שציפינו אך ביצענו חיבור בתצורת WIFI ובעת החיבור התקבלו המוצאים של הסנסורים של הM5stack על גבי הRaspberry Pi .

המטרות שהוצבו מראש כללו שני חיישנים נוספים – חיישן מיקרופון וחיישן רמקול. תוך העבודה בפרויקט ומתוקף הנסיבות המציאותיות בתקופה זו, נאלצנו לוותר על חיבור שני החיישנים הללו.

בנוסף, היה רצון ליצור בדיקה שמשלבת שני חיישנים יחד בפעולה משותפת, אך בשלב זה לא נעשתה בדיקה שכזו ולכן אנחנו מעוניינים להמליץ על ביצוע עתידי של פרויקט תשתית המשך שיעסוק בשילוב החיישנים אחד עם השני, יצירת מבדקים מתקדמים שמשלבים יחד כמה סוגים של חיישנים, מערכת התראה ועיבוד מידע מתקדמת ובין היתר גם בניית תצורה פיזית שבה יוחזקו החיישנים יחד וכך יהיה ניתן לקחת את התשתית כמקשה אחת לשימוש בפרויקטים העתידיים.

תיעוד הפרויקט

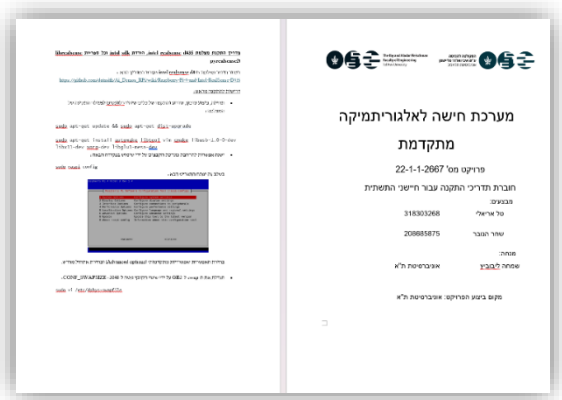
מסמך הגדרת חיישנים:

המסמך מכיל את כלל האפשרויות המתאימות שמצאנו עבור כל רכיב חומרתי שנרצה לעשות בו שימוש בפרויקט. המסמך יכיל על כל אפשרות: מידע (מפרטים), מידע על שיטות החיבור ופרוטוקולי תקשורת, עלות וקישור לרכש, יתרונות וחסרונות והמבדקים השונים אשר אפשר לבצע עבור כל סוג של סנסור.

איור 22 - מסמך הגדרת החיישנים

מסמך התקנות של כלל החיישנים:

חוברת זו מכילה את התדריכים המפורטים להתקנת כל אחד מהרכיבים וחיבורו לRaspberry Pi. היא כוללת הפניות לאתרים רלוונטיים שבהם נעזרנו על מנת לשמור על תוקף תוכן החוברת וכן הדרכות מפורטות אשר עומדות בעד עצמן עבור כל חיישן שנעשה עליו מבדק בעבודה זו, הספריות שנעשה שימוש בהן והסביבות הווירטואליות שנדרשה התקנתן בעת השימוש בחיישן



איור 23 - מסמך התקנות של החיישנים

Git-hub: מכיל את כלל הקבצים של הפרויקט, את ספר הפרויקט, מצגת, פוסטר וסקריפטים.
https://github.com/ShaharHanno/TAU-RP-Sensing/

מקורות

1. Raspberry Pi - Official Website - [Link](https://www.raspberrypi.com/)
2. Distance Detection with Depth Camera (Intel RealSense D435i) - [Link](https://pysource.com/2021/03/11/distance-detection-with-depth-camera-intel-realsense-d435i/)
3. Connecting a Fan to a Raspberry Pi - [Link](https://www.freva.com/connecting-a-fan-to-a-raspberry-pi/)
4. Slamtec RPLIDAR on Pi (PDF) - [Link](https://cdn-learn.adafruit.com/downloads/pdf/slamtec-rplidar-on-pi.pdf)
5. Raspberry Pi 4 Model B Specifications - [Link](https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/)
6. Intel RealSense Depth Camera D435 - [Link](https://www.intelrealsense.com/depth-camera-d435/)
7. Interfacing SSD1306 0.91 Inch OLED I2C Display with Arduino - [Link](https://electropeak.com/learn/interfacing-ssd1306-0-91-inch-oled-i2c-display-with-arduino/)
8. Raspberry Pi Pico: OLED Display (SSD1306) (YouTube) - [Link](https://www.youtube.com/watch?v=lRTQ0NsXMuw)
9. Using an I2C OLED Display Module with the Raspberry Pi - Raspberry Pi Spy - [Link](https://www.raspberrypi-spy.co.uk/?p=13185)
10. Enable I2C Interface on the Raspberry Pi - Raspberry Pi Spy - [Link](https://www.raspberrypi-spy.co.uk/?p=2080)
11. Raspberry Pi 4 and Intel RealSense D435 - Datasith/Ai_Demos_RPi Wiki - [Link](https://github.com/datasith/Ai_Demos_RPi/wiki/Raspberry-Pi-4-and-Intel-RealSense-D435)
12. Raspberry Pi 4 Install Realsense SDK make -j1 Issues - IntelRealSense/librealsense GitHub - [Link](https://github.com/IntelRealSense/librealsense/issues/3062)
13. Import Error SDK 2.24.0 Python3.7 on Raspberry Pi 4 Raspbian 10 Buster - IntelRealSense/librealsense GitHub - [Link](https://github.com/IntelRealSense/librealsense/issues/4375)
14. Roboticia RPLidar - [Link](https://github.com/Roboticia/RPLidar)
15. Raspberry Pi Documentation - [Link](https://www.raspberrypi.com/documentation/)
16. Raspberry Pi 4 Model B - [Link](https://www.raspberrypi.com/products/raspberry-pi-4-model-b/)

נספחים

1. נספח 1 : פרק רקע תיאורטי, תחת הנושא של Raspberry PI, לוח GPIO, מידע נוסף וחיבורים נוספים :

מידע נוסף:

כמו גם התקני קלט ופלט פשוטים, ניתן להשתמש בפיני GPIO עם מגוון פונק' חלופיות, חלקם זמינים על כל הפינים וחלקם על פינים ספציפיים.

- PWM (pulse-width modulation)
  - Software PWM available on all pins
  - Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19
- SPI (Serial Peripheral Interface)
  - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
  - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C
  - Data: (GPIO2); Clock (GPIO3)
  - EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
- Serial
  - TX (GPIO14); RX (GPIO15)

חיבורים נוספים

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)

- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient
- * A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

2. נספח 2 : פרק רקע תיאורטי, תחת הנושא של Raspberry PI, התקנת מערכת ההפעלה Raspian OS:
   - בחירה ב Raspberry Pi os 32bit
   - הורדה מהלינק : https://www.raspberrypi.com/software/
   - צריבת מערכת ההפעלה על גבי micro SD card.
   - חיבור הכרטיס ל Raspberry Pi .
   - כעת ניתן להשתמש במכשיר באופן תקין.

3. נספח 3 : טבלת תוצאות ניסוי מדידת מרחק הפיקסל המרכזי :

Time (s)	Depth [m]
1	0.2
2	0.26
3	0.26
4	0.34
5	0.46
6	0.54
7	0.66
8	0.83
9	0.98
10	1.14
11	1.31
12	1.5
13	1.68
14	1.87
15	2
16	2.19
17	2.37
18	2.48
19	2.46
20	2.19
21	1.9
22	1.59

23	1.3
24	0.97
25	0.7
26	0.47
27	0.34
28	0.28
29	0.42
30	0.61
31	0.8
32	0.97
33	1.14
34	1.34
35	1.51
36	1.73
37	1.6
38	1.31
39	1.01
40	0.75
41	0.6
42	0.55
43	0.56
44	0.56
45	0.49
46	0.46

טבלה 2- תוצאות ניסוי מדידת מרחק הפיקבל המרכזי

4. נספח 4 : הערה לגבי בדיקות הLIDAR :

בתחילת כל שימוש בחיישן :

נפעיל את הסביבה הוירטואלית ע״י הפקודה :

```
source venv/bin/activate
```

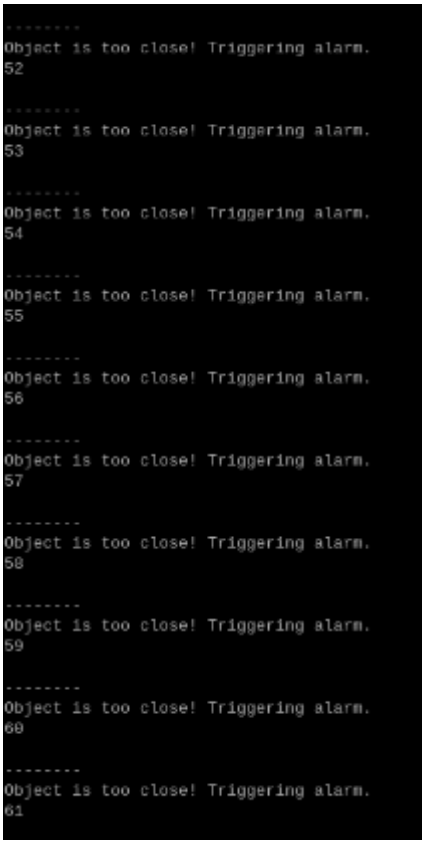
בתוך הסביבה הוירטואלית נוריד את החבילות ע״י הפקודות הבאות : (מתבצע בשימוש הראשוני בלבד)

```
pip install numpy pygame
pip install rplidar
pip3 install rplidar-roboticia
```

בחרנו את מודל פייתון זה מRoboticia מכיוון שמצאנו כי הממשק שלו עם LIDAR שלנו טוב.

ניתן לראות פירוט של המודל בגיט הבא : https://github.com/Roboticia/RPLidar

5. נספח 5 : צילום מסך של הודעת האזהרה של הLIDAR בעבור המבדק :



איור 24-הודעת האזהרה של הLIDAR בעת מבדק מיקום עצם במרחק שקטן מ150 מ"מ

6. נספח 6 : נספח תיעוד (ערכים מדויקים) של מדידת המרחקים והזוויות של ה :LIDAR

num of measure	angle[deg]	Distance [mm]
1	289.078125	146.5
2	285.046875	147.75
3	281.28125	149.75
4	281.46875	149.25
5	282.765625	148.5
6	284.328125	148.25
7	288.1875	147
8	297.390625	144.5
9	308.421875	146.5
10	323.140625	146.75
11	328.03125	147.5
12	332.328125	144
13	329.859375	143
14	331.265625	141.25
15	333.453125	141.75

16	335.90625	144.5
17	341.984375	146.5
18	348.203125	146.5
19	356.375	143.25
20	1.140625	140
21	360.234375	143.75
22	5.21875	141.75
23	4.640625	148.25
24	13.921875	146.75
25	21.75	145.75
26	28.015625	147.25
27	30.25	148
28	31.125	148.5
29	34.03125	148
30	37.140625	148.5
31	44.0625	148
32	53.84375	148.25
33	60.09375	149
34	65.484375	145.5
35	66.296875	143.5
36	66.265625	143.5
37	65.375	145.25
38	64.671875	143.5
39	60.875	146
40	61.953125	145.25
41	63.234375	145.5
42	68.859375	141.5
43	67.859375	145
44	64.75	145.5
45	61.40625	143.25
46	62.625	140.75
47	113.09375	139.75
48	110.0625	142.75
49	111.796875	147
50	111.390625	148.5
51	113.390625	149.75
52	117.6875	146.5
53	115.28125	148.75
54	113	149.75
55	113.828125	149.25
56	115.71875	148
57	119.796875	147.5
58	122.0625	148.25
59	129.21875	149.5
60	135.78125	149.75
61	142.765625	149.75
62	149.5625	149

63	149.671875	149.75
64	147.828125	149.75
65	155.65625	148
66	161.703125	147
67	161.40625	148
68	163.953125	147
69	161.734375	148.5
70	166.90625	146
71	167.359375	142
72	169	141
73	166.0625	142.5
74	172.21875	140.5
75	171.875	140.5
76	172.484375	147.75
77	177.734375	144.75
78	180.640625	143
79	177.875	146
80	177.671875	147
81	180.84375	146
82	182.28125	148.25
83	185.796875	145.25
84	189.078125	142.5
85	190.109375	143.75
86	193.359375	142.5
87	196.53125	142.5
88	200.5	141.5
89	202.78125	140.25
90	205.5625	140
91	206.109375	139.5
92	211.609375	139.75
93	212.28125	141
94	211.5	141
95	212.453125	141
96	213.28125	139.5
97	211.78125	139.75
98	252.265625	143.75
99	253.265625	149.25
100	258.84375	141.75
101	259.75	144.25
102	258.453125	146.5
103	259.625	146
104	262.03125	147
105	262.390625	148.5
106	266.1875	146.75

טבלה 3- מדידת המרחקים והזוויות של הLIDAR

7. נספח 7 : צילום מסך של חיבור הRaspberry Pi עם ה :M5stack

ניתן לראות ערכים שנקלטים על ידי הסנסורים בarduino ומוצגים בצג הRaspberry Pi.

```
File  Edit  Tabs  Help
 [35.2 36.0 35.8 36.0 36.2 36.0 36.0 36.8]
 [35.0 35.8 35.8 36.2 36.2 36.5 36.5 36.5]
 [35.2 36.0 36.2 36.8 36.5 36.5 36.2 36.8]
 [35.2 36.2 36.0 36.8 36.8 36.5 36.5 37.0]
 [35.5 35.8 36.2 36.5 36.5 36.5 36.5 36.2]
 [35.2 35.5 36.0 35.8 35.8 36.5 36.2 36.5]
 [35.0 35.0 35.0 34.8 35.8 35.8 35.8 35.8]]
ToF distances[mm] 8x8:
[[46 51 54 58 66 72 66 59]
 [41 46 47 55 60 64 61 60]
 [38 40 44 46 52 62 61 63]
 [32 37 35 41 47 55 58 65]
 [30 33 35 37 44 45 49 55]
 [26 31 31 36 37 40 43 49]
 [25 27 29 33 33 37 39 44]
 [21 23 28 29 31 36 37 39]]
FSR value: 0
Imu(Gyro=GyroData(x=  -0, y=  -0, z=  -0), Accel=AccelData(x=   0, y=   0, z=  10))
AMG88xx ° 8x8:
[[34.8 35.2 35.5 35.5 35.8 35.8 35.8 36.2]
 [35.2 36.0 36.0 36.0 36.2 36.2 36.0 36.2]
 [35.2 35.8 36.2 36.2 36.5 36.2 36.5 36.0]
 [35.5 35.8 36.2 36.8 36.2 36.5 36.5 36.8]
 [35.2 35.8 35.8 36.5 36.5 36.8 36.2 36.8]
 [35.0 36.0 36.2 36.8 36.8 36.8 36.5 36.5]
 [35.5 35.8 36.2 36.0 36.2 36.0 35.8 36.2]
 [35.0 35.2 35.2 35.5 35.0 35.5 35.8 36.0]]
ToF distances[mm] 8x8:
[[  44   51   57   61   65   69   65   55]
 [  41   46 1267   56   59   64   61   59]
 [  38   42   44   46   52   62   62   64]
 [  33   38   37   43   46   55   57   65]
 [  29   35   36   37   44   47   50   55]
 [  26   33   31   36   38   42   42   48]
 [  25   25   29   32   34   38   39   44]
 [  20   23   26   29   30   35   34   38]]
FSR value: 0
Imu(Gyro=GyroData(x=   0, y=  -0, z=  -0), Accel=AccelData(x=   0, y=   0, z=  10))
AMG88xx ° 8x8:
[[35.0 35.2 36.0 35.5 35.2 35.2 35.5 36.2]
 [35.0 35.2 36.0 36.0 36.0 36.0 36.0 36.5]
 [35.0 36.0 36.0 36.0 36.5 36.8 36.5 36.0]
 [35.0 36.0 36.0 36.8 36.2 36.2 36.0 36.8]
 [35.2 36.2 36.0 36.8 36.8 36.5 36.5 36.5]
 [35.5 35.8 36.2 36.5 36.5 36.5 36.8 36.5]
 [35.2 35.2 36.0 36.5 35.8 35.5 35.8 36.2]
 [35.0 35.2 34.8 35.5 35.5 35.5 35.8 36.0]]
ToF distances[mm] 8x8:
[[  44   51   57   61   65   69   65   55]
 [  41   46 1267   56   59   64   61   59]
 [  38   42   44   46   52   62   62   64]
 [  33   38   37   43   46   55   57   65]
 [  29   35   36   37   44   47   50   55]
 [  26   33   31   36   38   42   42   48]
 [  25   25   29   32   34   38   39   44]
 [  20   23   26   29   30   35   34   38]]
```

איור 25- קבלת טלמטריה מArduino

TOF ניתן לראות מד מרחק: מטריצה שמציגה מרחק מהחיישן במ״מ (64 פיקסלים שמוצגים במטריצה 8X8 וכל איבר בה מייצג את הערך המתאים)

AMG מצלמה תרמית- מציגה טמפרטורה במעלות צלזיוס (64 פיקסלים שמוצגים במטריצה)

FSR חיישן לחץ- מודפסים ביטים שמייצגים את הלחץ המופעל על החיישן

IMU (GYRO) – מיקום בשלושה צירים XYZ

ACC – מד תאוצה בשלושה צירים XYZ