

Real-Time Voice Control of a Human Robotic Arm using Raspberry Pi

**שליטה קולית בזמן אמת על כף יד רובוטית
אנושית באמצעות Raspberry Pi**



**Student: Avi Digma
Semester B, 2024**

Facilitator: Dr. Yakov Damatov

רציונל

פרויקטים רבים בתחום הרובוטיקה עושים שימוש בסט פקודות קוליות המוגדרות מראש בעיקר כמודל של למידת מכונה.

בדרך כלל, פקודות קוליות ניתנות באנגלית. עם זאת, יש צורך משמעותי בתמיכה בפקודות קוליות בשפה הלאומית של המדינה בה מתוכנן להשתמש ברובוט.

מטרה

המטרה היא לפתח פלטפורמת מערכת שליטה קולית התומכת בעברית בזמן אמת לכף יד רובוטית אנושית.

שימוש ב-Raspberry Pi באמצעות ציוד היקפי ממשפחתו כגון מסך, מצלמה ומקלדת, יחד עם מיני מיקרופון USB (לא ממשפחת Raspberry Pi), יהפוך את המערכת לקומפקטית יותר וקלה יותר לשימוש, מבלי לפגוע בכוח העיבוד של המערכת.

שלבי פיתוח המערכת

- שלב ראשון: עבודה מול המערכת כפיילוט – בשלב זה נבחן את יכולות המערכת כאשר נחבר אליה מיקרופון Mini USB, מקלדת ועכבר חוטיים של מחשב נייד, וחיבור למסך מחשב באמצעות כבל HDMI - Micro HDMI.
- שלב שני: נבצע מינימיזציה למערכת – נחבר אליה ציוד ממשפחת Raspberry Pi: מקלדת + לוח מגע אלחוטיים.
- שלב שלישי: נבצע בדיקות סופיות של כלל המערכת לאחר כל החיבורים כולל מסך מגע ייעודי עבור Raspberry Pi.

מטלות הנדרשות לפיתוח הפלטפורמה

➤ Raspberry Pi 4:

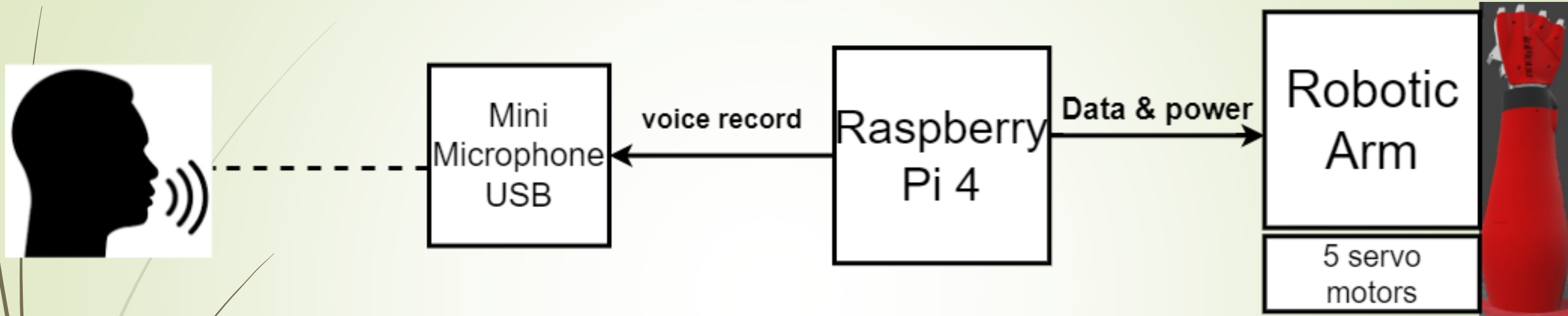
➤ התקנת תוכנות וספריות נדרשות.

➤ ממשק החומרה בדגש על המיקרופון והפעלתו ב-Raspberry.

➤ עבודה עם דרייברים (קול ותצוגה).

➤ פיתוח אלגוריתם ותוכנה

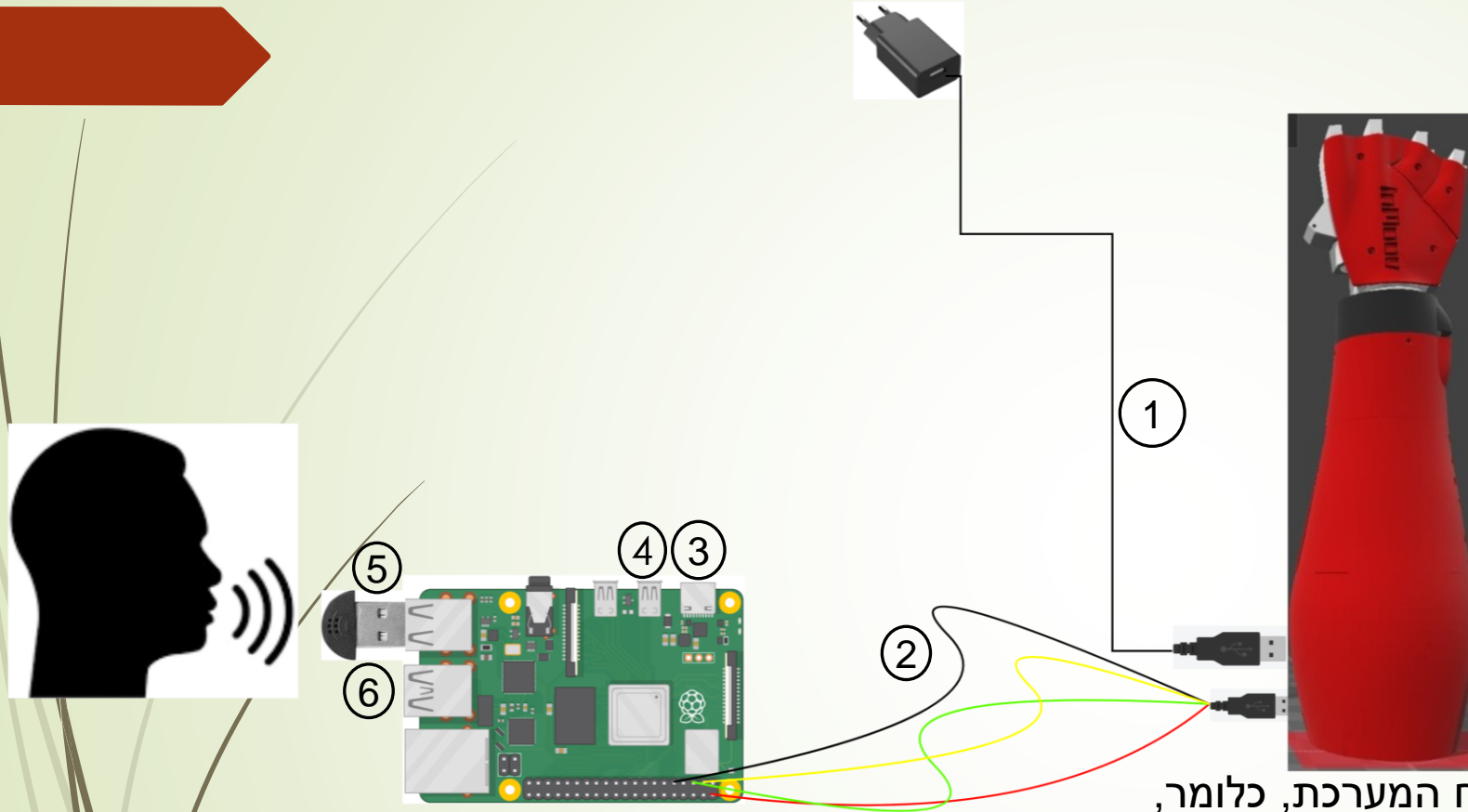
מבנה הפלטפורמה (מערכת לשליטה קולית בזמן אמת על כף יד רובוטית באמצעות Raspberry Pi)



איור מס' 1. דיאגרמת בלוקים כללית של המערכת

איור מס' 1 מציג את האינטראקציה בין חלקי המערכת השונים, מהאזנה לפקודת המשתמש ועד לתנועת האצבעות של היד הרובוטית הווירטואלית. המערכת כוללת מיקרו מחשב, Raspberry Pi, המקבל מידע קולי מהמיקרופון, מעבד מידע זה ולאחר מכן משדר את אותות הבקרה המתאימים למנועים הנמצאים בתוך הזרוע הרובוטית הווירטואלית. מנועים אלו מספקים את תנועת האצבעות של היד הרובוטית.

חיבור בין רכיבי המערכת (שלב ביניים של פיתוח המערכת)

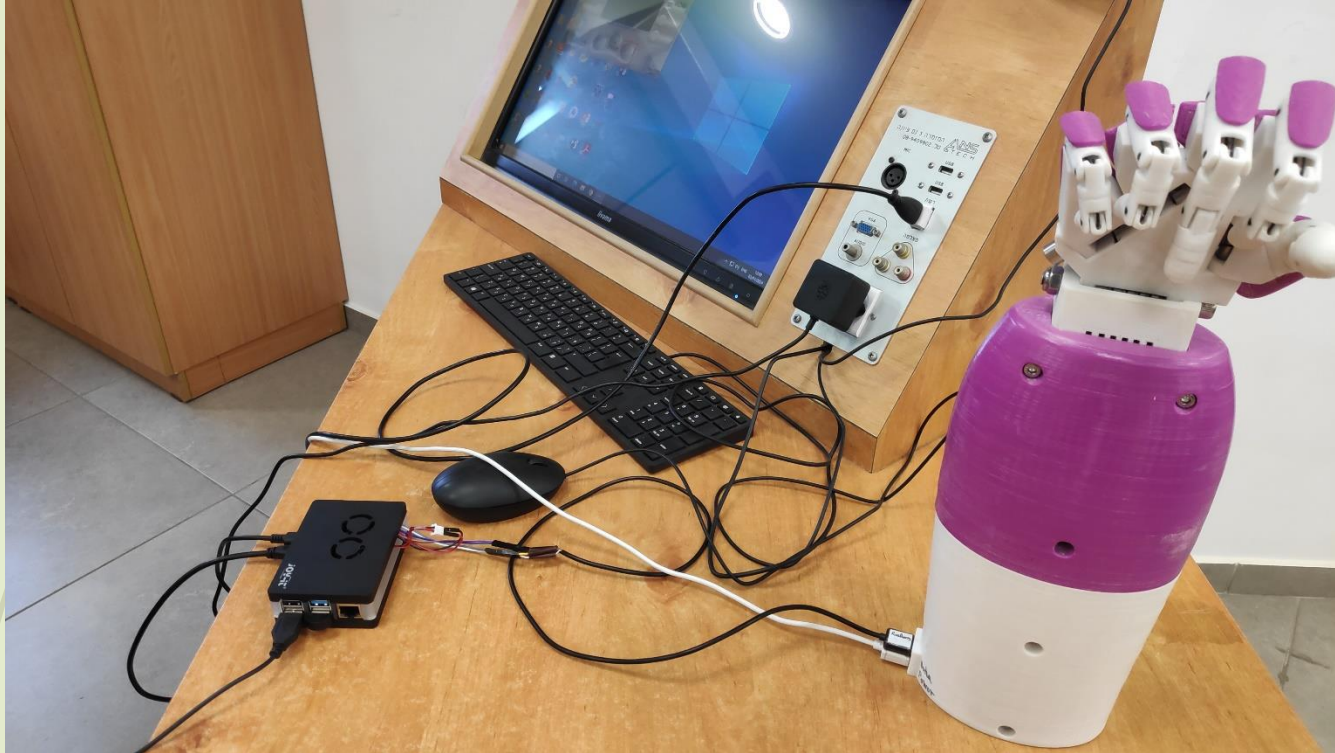


איור מס' 2. קישורים בין רכיבי המערכת

איור מס' 2 מציג את אחד משלבי הביניים של פיתוח המערכת, כלומר, השימוש במיקרו-מחשב Raspberry Pi במערכת במקום במחשב.

1. ספק כוח עבוד כף היד הרובוטית
2. כבל data&power המעביר אותות למנועי כף היד
3. ספק כוח עבור Raspberry Pi
4. יציאה לחיבור למסך HDMI-Micro HDMI
5. חיבור של מיקרופון USB Mini
6. חיבורי USB למקלדת ועכבר חוטיים של מחשב נייד

מבנה מערכת כפיילוט – Raspberry Pi (שלב ביניים של פיתוח המערכת)



כדי לבצע פיתוח, ניסוי, ובדיקה ראשונית של המערכת נעשה שימוש בצידוד: מקלדת ועכבר חוטיים של מחשב ניח, מיקרופון Mini USB ובנוסף חיברנו למסך חיצוני של מחשב ניח ע"י כבל HDMI-Micro HDMI.

איור מס' 3. חיבורי המערכת. גרסה 1.

משאבי מערכת שליטה קולית בזמן אמת על כף יד רובוטית אנושית באמצעות Raspberry Pi (שלב ביניים)

משאבי ה-Raspberry Pi4 - הסבר כללי :

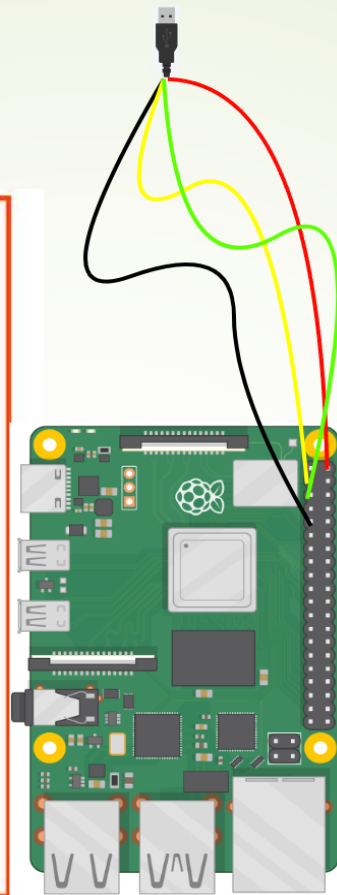
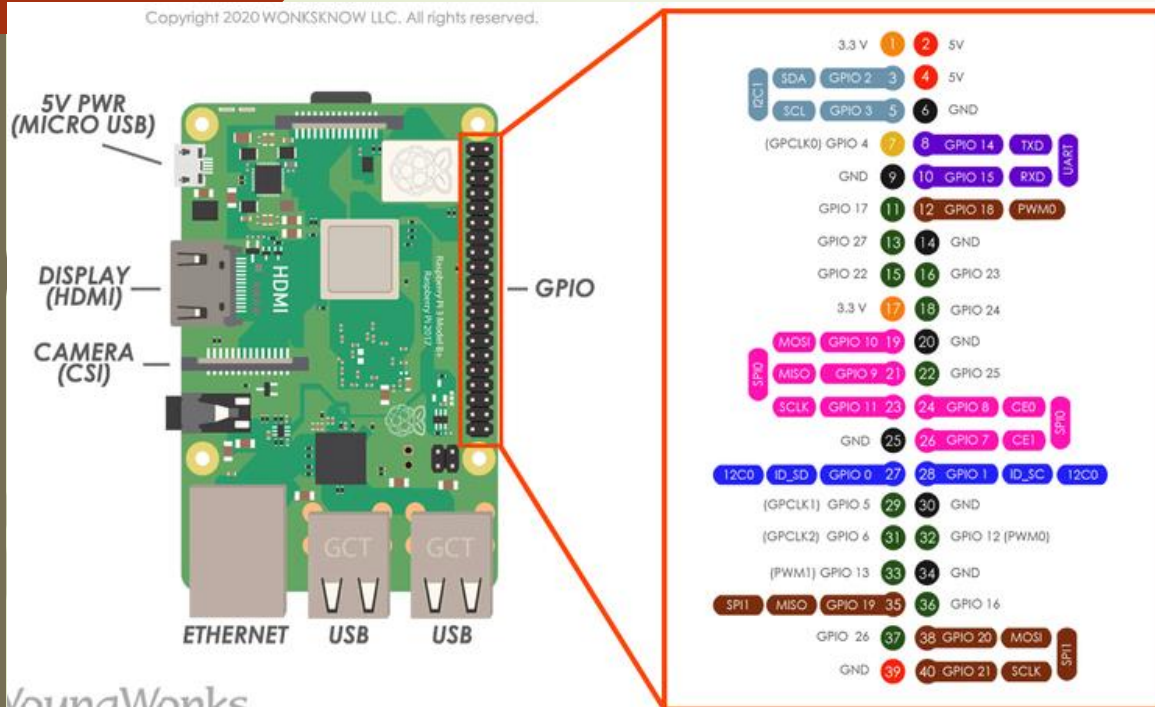
ה-Raspberry pi4 הוא מיני מחשב שכולו נמצא על לוח אחד, ובעל יכולת לשמש בנוסף כבקר. הוא מכיל את החיבורים הבאים:

1. מעבד @ 1.8GHz 64-bit SoC Cortex-A72 (ARM v8), Broadcom BCM2711, Quad core
2. זיכרון פנימי: 8 GB
3. USB-c 5V (חיבור לחשמל)
4. 2 יציאות micro-HDMI
5. 2 כניסות USB 2.0
6. 2 כניסות USB 3.0
7. כניסת Ethernet
8. יציאת Audio
9. כניסת microSD
10. 2-lane MIPI CSI (עבור מצלמת Raspberry)
11. 2-lane MIPI DSI (עבור חיבור למסך ייעודי)
12. PoE HAT (4 חיבורי פינים שמאפשרים את הפעלת המחשב באמצעות רשת Ethernet המופעלת במתח)
13. 40 Pin GPIO Header (קו של 40 חיבורי קלט/פלט המרחיבים את יכולות המחשב ומאפשרים שימוש בו כבקר)

רשימת רכיבי מערכת ומאפייני ה-Raspberry Pi:

- זיכרון עבודה: 512 GB SD Card
- מערכת הפעלה: Raspbian (מערכת הפעלה ייחודית עבור Raspberry Pi מבוססת Debian)
- מעטפת: כיסוי פלסטיק קשיח כולל מאוורר המתחבר ל-GPIO
- מיקרופון: USB Mini Microphone
- יד רובוטית: מכילה 5 מנועי Servo

חיווט Raspberry Pi עם היד הרובוטית



Power

5V (+) – אדום

GND (-) – שחור

Data

SDA – צהוב

SCL – ירוק

איור 5. סקירה של מבנה ה-GPIO

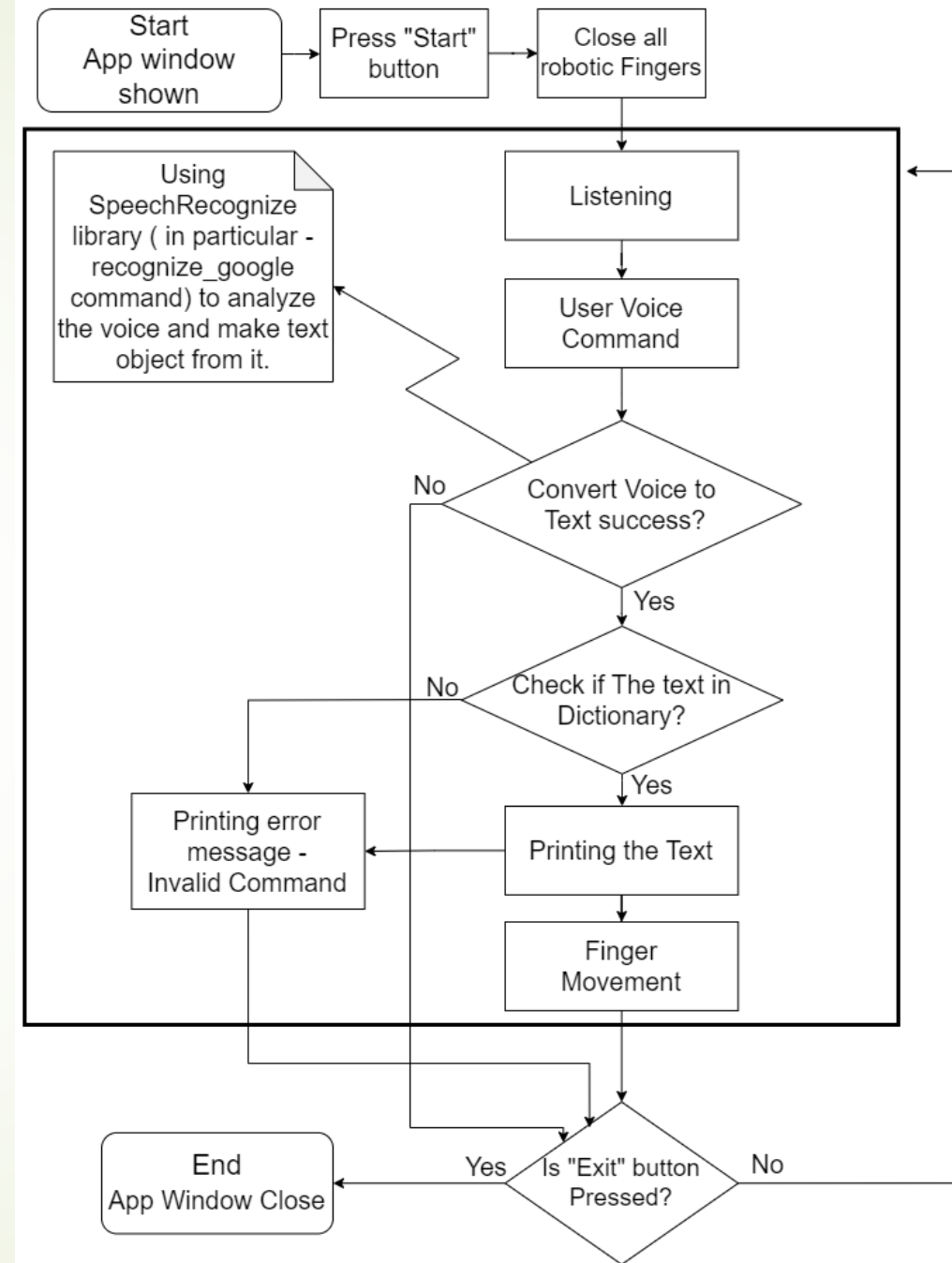
איור 4. חיווט כבל data&power ל-Raspberry Pi

באיורים 4 ו-5 ניתן לראות איך מתבצע חיווט כבל data&power אל Raspberry Pi4. הכבל בצידו האחד מתחבר אל ה-GPIO Bus כפי שמתואר בתמונה ומצידו האחר מחובר ב-USB אל כניסה מתאימה ביד הרובוטית. ניתן לחוות את המערכת בכמה אופנים: ישנם 2 כניסות 5V, 8 כניסות GND, ו-2 אופציות של חיבורי I2C (SDA ו-SCL). ניתן לראות בבירור את המבנה (איור 5) והחיווט (איור 4) בנספחים 1 ו-2 בסוף המצגת.

תרשים זרימה של אלגוריתם שליטה קולית על יד רובוטית באמצעות Raspberry Pi

תיאור האלגוריתם:

1. הופעת חלון האפליקציה
2. לחיצה על כפתור Start.
3. סגירת כל אצבעות הרובוט (אתחול).
4. כל עוד לא לחצנו על כפתור Exit:
 - 4.1 האזנה תוך התעלמות מרעשי רקע.
 - 4.2 המרת הקלט הקולי לטקסט
 - 4.3 אם ההמרה הצליחה:
 - 4.3.2 אם המילה/פקודה הוגדרה מראש:
 - 4.3.2.1 הדפסת הפקודה
 - 4.3.2.2 פנייה למנוע ספציפי לפי שם האצבע או לכל המנועים אם הפקודה הייתה כללית והפעלתם ע"פ מצבם העכשווי – כל זה ע"י פקודת מתן קלט של זווית למנוע (0-180).
 - 4.3.3 אחרת:
 - 4.3.3.1 הדפסת הודעת שגיאה: "פקודה לא חוקית"
5. שחרור החלון וסגירת האפליקציה.



בחירת ספריה להמרת קול לטקסט

במהלך פיתוח האפליקציה נדרשנו לקבל החלטה באיזו ספריית המרת קול לטקסט להשתמש. לצורך קבלת החלטה ביצענו ניסויים רבים עם כל אחת מהאפשרויות (google, whisper, sphinx...). המאפיינים שנבדקו הם: מהירות, איכות ניתוח, תמיכה בעברית ואפשרות שימוש ללא חיבור אינטרנט. הגענו למסקנה שעיבוד נתונים בעזרת ספריות שעושות שימוש במודלים לוקאליים, דורש כוח עיבוד גדול מאוד, המקשה על הפלטפורמה הזו. לבסוף הוחלט שעבור המטרות הנוכחיות שלנו נשתמש בספריית SpeechRecognize כאשר פקודת התרגום תהיה recognize_google. השימוש בגוגל כאמצעי לתרגום מביא את התוצאות הטובות ביותר, הוא מהיר, מדויק, תומך בעברית ויש צורך בחיבור לאינטרנט על מנת להשתמש בו.

הגדרת מיני מיקרופון (USB) למערכת ההפעלה של Raspberry Pi

על מנת שהאפליקציה תוכל לגשת למיקרופון ולהפעיל אותו היה צורך בסנכרון שלו עם מערכת ההפעלה Raspbian. נדרשנו לטפל ידנית בקבצי התצורה (קונפיגורציה) של התקני הקול ולשנות את הגדרות ברירת המחדל כך שעבור קלט קולי האופציה הראשונה תהיה המיקרופון. בנוסף לכך התקנו ספריה נוספת הנקראת sounddevice והוספנו אותה לקוד. זאת על מנת לתת חיזוק כך שהתוכנית תדע שהוא פונה להתקן קול (דרך ברירת המחדל שהגדרנו).

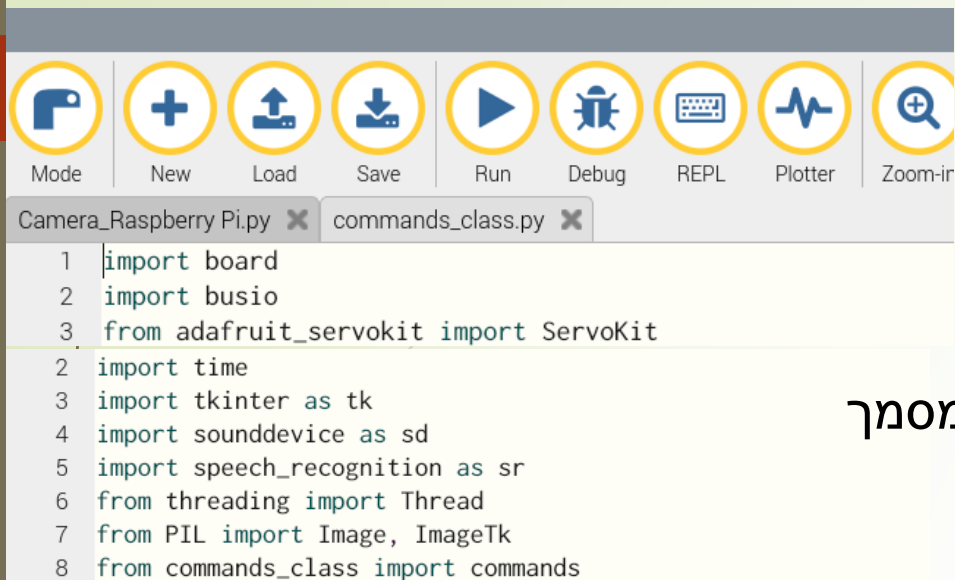
רעשי רקע

במהלך הניסויים שערכנו עם המיקרופון, התגלה שהוא מקליט לא מעט רעשי רקע המפריעים להמרת הקול לטקסט. לאחר בדיקה גילינו שלספריית SpeechRecognize קיימת פקודת התעלמות מרעשי רקע.

חלון אפליקציה ותמונות מתחלפות

לאפליקציית שליטה קולית נדרשת אינטראקציה (קצת יותר מבסיסית) עם המשתמש. לכן הוחלט שכדאי לשלב ממשק משתמש בסיסי (חלון אפליקציה שנבנה באמצעות ספריית tkinter) המאפשר למשתמש לדעת מתי התוכנית מאזינה לפקודות וכיצד נראה התרגום. כדי להפוך את האינטראקציה לידידותית יותר החלטנו לשלב תמונה של אוזן כאשר האפליקציה מאזינה, תמונת בצורת גלגל שיניים כאשר מתבצעת המרת קול לטקסט (convert voice to text), ובנוסף שורת טקסט המציגה את התרגום. השימוש בכלי זה התאפשר בעזרת ספריית Treads.

תוכנה – הפעלה וריצה



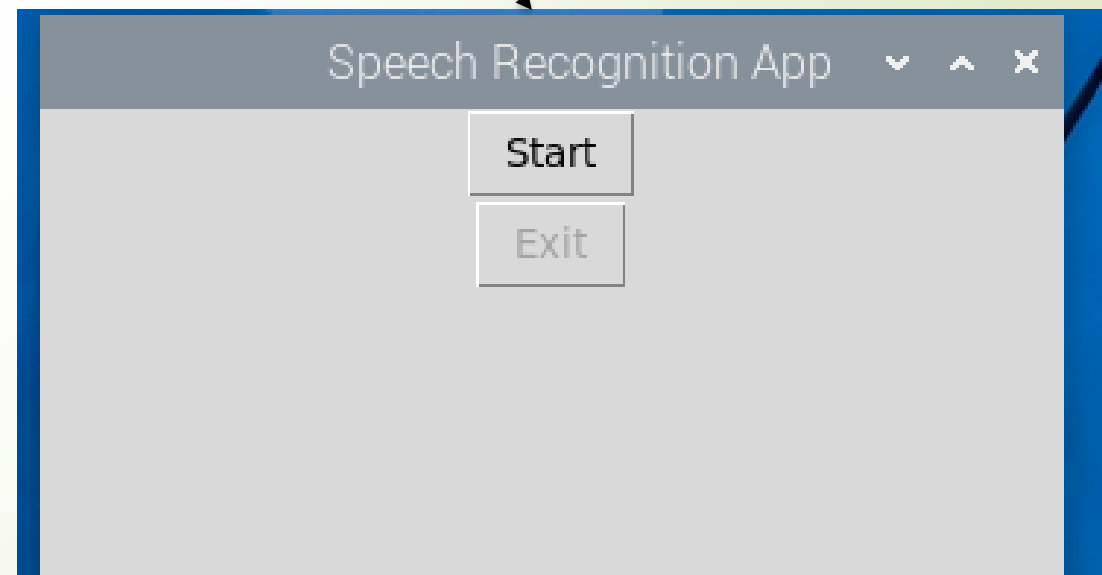
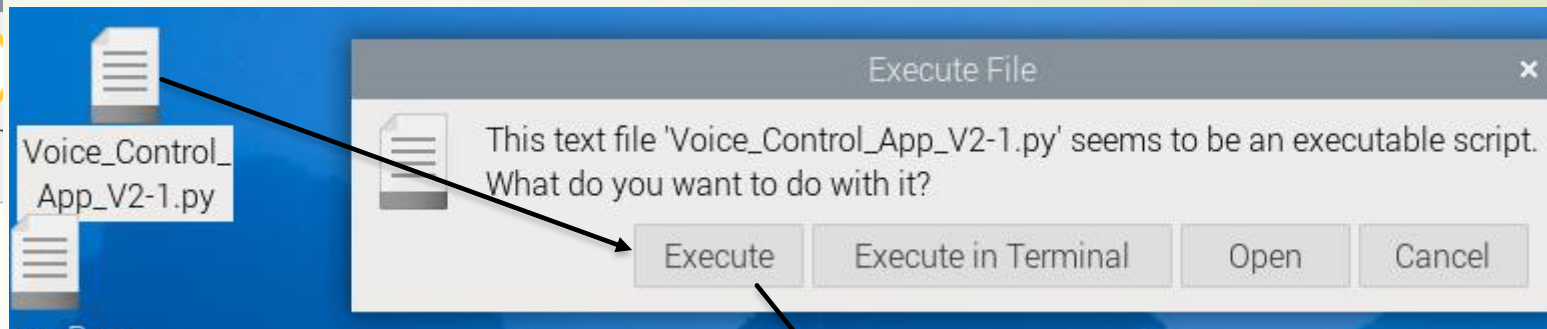
```
def my_listen(self):
    while self.is_listening:
        try:
            with self.microphone as source:
                self.recognizer.adjust_for_ambient_noise(source, duration=3)
                self.image_label.config(image=self.ear_image)
                self.master.after(0, lambda: self.output_label.config(text=f"{listeningText[::-1]}", font=("Helvetica", 20), fg="green"))
                self.master.update()
                audio_data = self.recognizer.listen(source, phrase_time_limit=3)

                self.image_label.config(image=self.gear_image)
                self.master.update()

                text = self.recognizer.recognize_google(audio_data, language="iw-IL")
                text2 = self.makemove(text)
                if(text2=="פקודה לא חוקית, נסה שוב"):
                    self.master.after(0, lambda: self.update_output_label(text2, "red"))
                else:
                    self.master.after(0, lambda: self.update_output_label(text2, "black"))
                time.sleep(1)

        except sr.UnknownValueError:
            print("Could not understand audio")
        except sr.RequestError as e:
            print(f"Error connecting to Google API: {e}")
        time.sleep(1)
    self.master.after(100, self.my_listen)
```

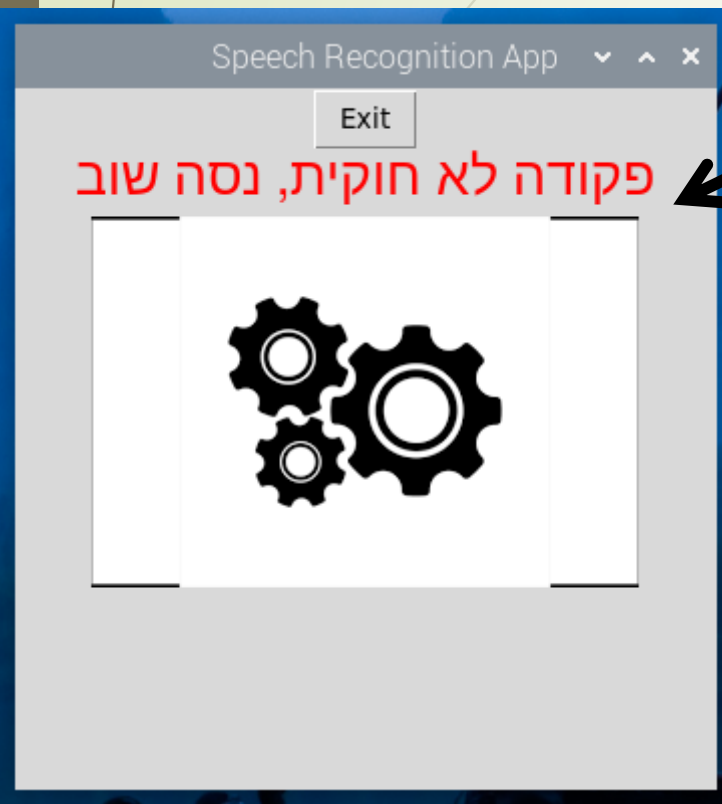
איור 7. הפעלת התוכנה – הרצה של מסמך הקוד כקובץ executable script.



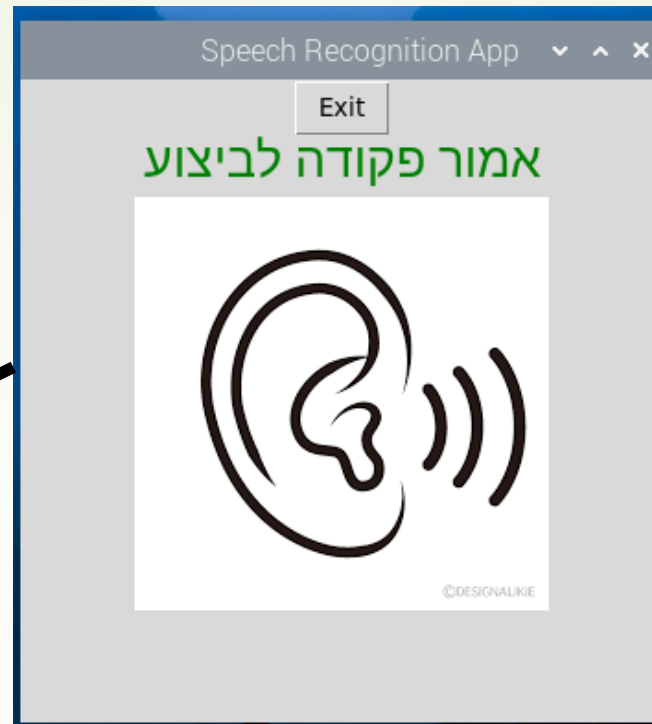
איור 9. חלון האפליקציה הראשי – לפני תחילת ביצוע פעולות.

איור 8. קטעי קוד המציגים את הספריות שבהן התוכנה עושה שימוש במהלך הריצה, וכן את הלולאה הראשית של התוכנה

אינטראקציה בין המשתמש לאפליקציה



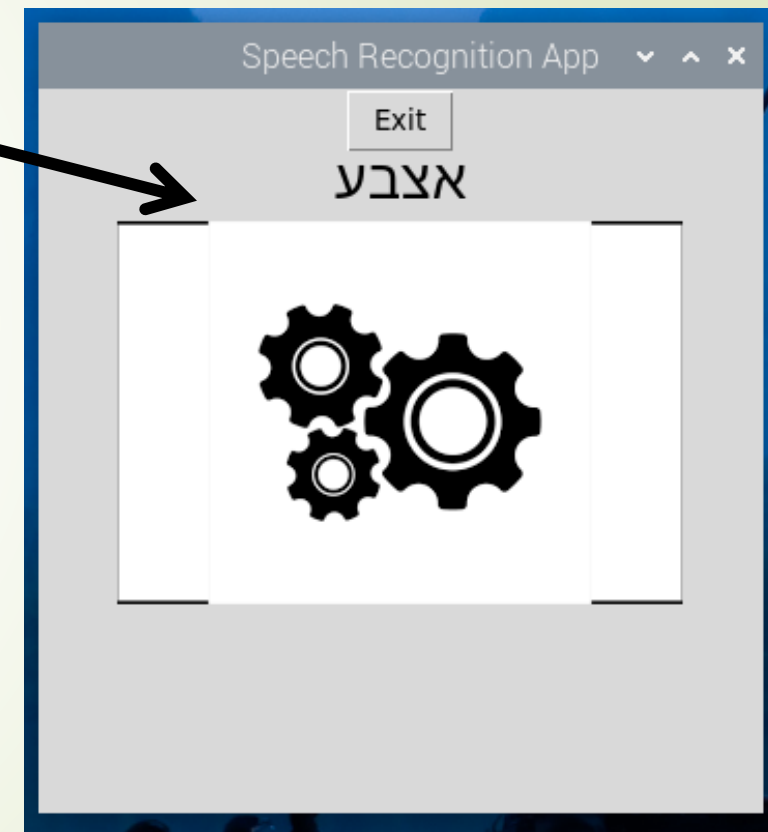
איור 11. האפליקציה מציגה למשתמש הודעת שגיאה מתאימה עבור פקודה שלא מוגדרת.



איור 10. האפליקציה מאזינה למשתמש ומחכה לפקודה ממנו.

הערה:

בכל שלב ניתן לצאת מהאפליקציה ע"י לחיצה על כפתור Exit.



איור 12. האפליקציה מציגה למשתמש את הפעולה החוקית שאמר.

שימוש במערכת

במערכת ניתן להשתמש בשתי דרכים – בחיבור למסך או ללא חיבור.

כאשר המערכת מחוברת למסך ניתן לשים לב לאינטראקציה עם האפליקציה:

1. כאשר האפליקציה מאזינה למשתמש היא מציגה הודעה מתאימה ותמונה של אוזן. היא מקשיבה למשך כ-3 שניות.
2. לאחר מכן מתבצע שלב ניתוח השמע והזזת האצבעות הרובוטיות. בשלב זה האפליקציה מציגה תמונה של גלגל שיניים וטקסט מעליה – אם המילה מוגדרת מראש כפקודה חוקית היא תודפס ותתבצע תזוזה בהתאם, אחרת תוצג הודעה למשתמש: "פקודה לא חוקית".
3. לאחר כמה שניות נחזור למצב בשלב 1.

כאשר מתבצע שימוש במערכת ללא מסך נצטרך לשים לב להשהיה בין הפעולות: כאשר האפליקציה עוברת למצב האזנה נוצרת השהייה של 2-3 שניות שבהן היא לא מאזינה אלא מכינה את עצמה לפעולה.

1. נפעיל את האפליקציה.
2. נמתין 2-3 שניות.
3. כעת האפליקציה מאזינה לנו במשך 3 שניות – נגיד את הפקודה שנרצה שהיד תבצע ונחכה שהיא תקרה – אם המילה מוגדרת מראש כפקודה חוקית תתבצע תזוזה בהתאם, אחרת לא יקרה כלום.
4. נחזור לשלב 2.

שלב הפיתוח השני – מינימיזציה של המערכת

- בשלב הפיתוח הזה נבצע חיבור לציוד ההיקפי ממשפחת Raspberry Pi – נרצה לבצע התאמה מיטבית ל-Raspberry Pi, נוחות וקומפקטיות.
 - לשם כך נחליף את הרכיבים הבאים: מקלדת, עכבר ומסך.
 - נשתמש במקלדת אלחוטית ומשטח מגע המובנה בה ומסך מגע ייעודי, המתחבר ישירות ללוח.
 - המטרה כאן היא ליצור סביבה אופטימלית עבור הפלטפורמה ובעיקר עבור המשתמש.
- בשקפים הבאים נתאר את תהליך המינימיזציה שהמערכת עברה בשלב הפיתוח השני.

מערכת שליטה קולית בזמן אמת על כף יד רובוטית אנושית באמצעות Raspberry Pi

כפי שניתן לראות באיורים, בגרסה זו של המערכת, חיברנו ל-Raspberry Pi:

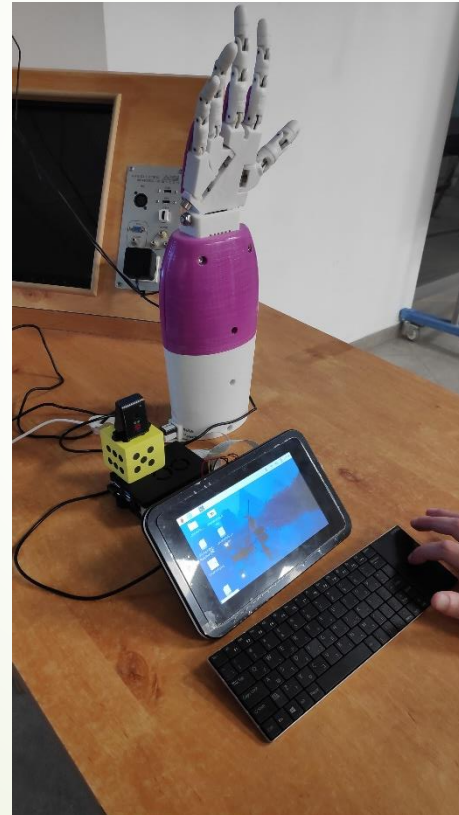
1. מיני מקלדת + משטח TouchPad אלחוטיים המתחברים עם Nano Wireless USB Adapter.
2. מיקרופון USB Mini Microphone בחיבור כניסת USB של Raspberry Pi.
3. מסך מגע ממשפחת Raspberry Pi בכניסת "2-lane MIPI DSI" ייעודית עבור מסך זה.



איור 13. חיבור המסך

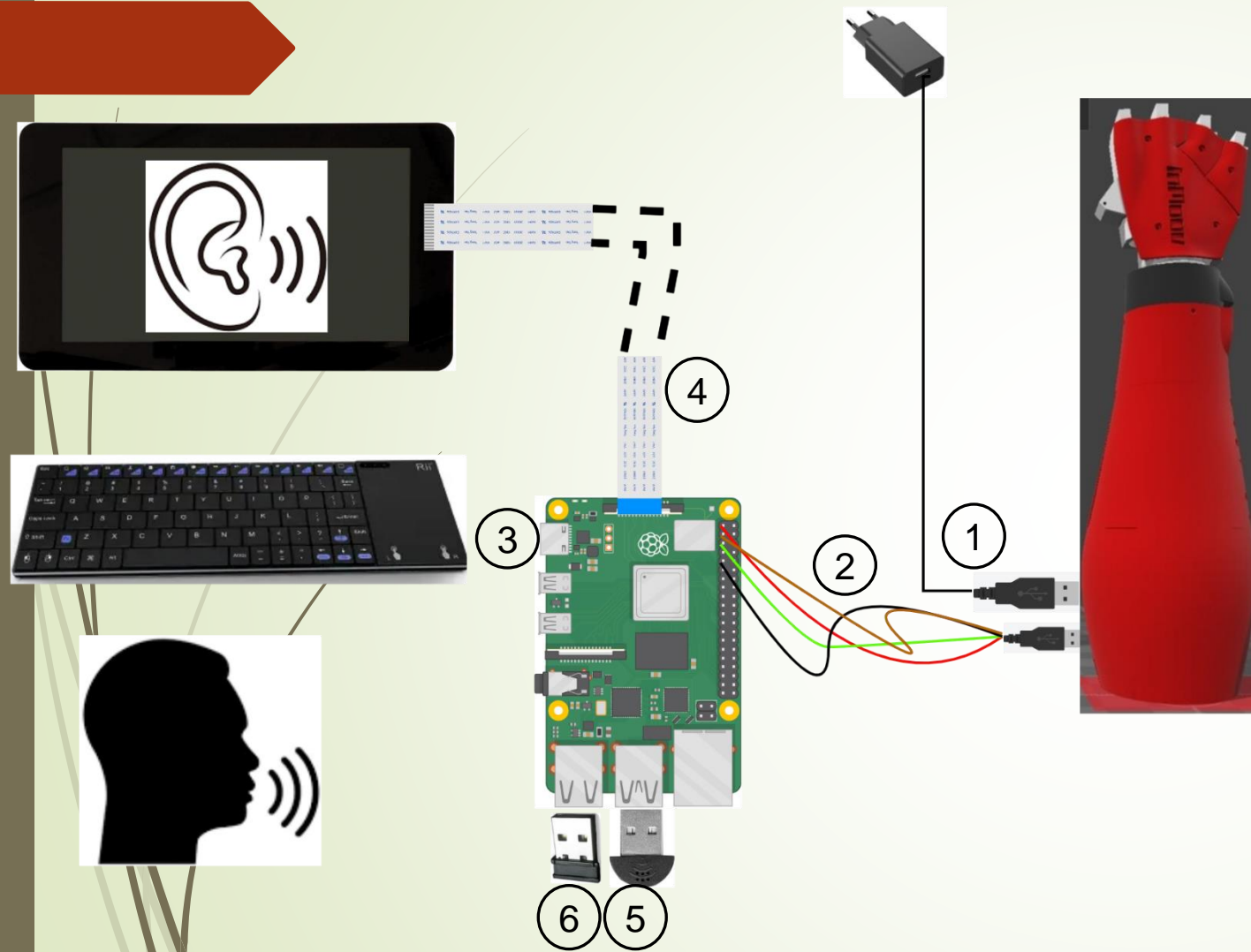


איור 14. חיבור מיני מיקרופון USB, מקלדת ומשטח מגע אלחוטיים (המשטח מובנה בצד המקלדת)



איור 15. המערכת החדשה בשלמותה

חיבורים בין רכיבי המערכת (שלב סופי)

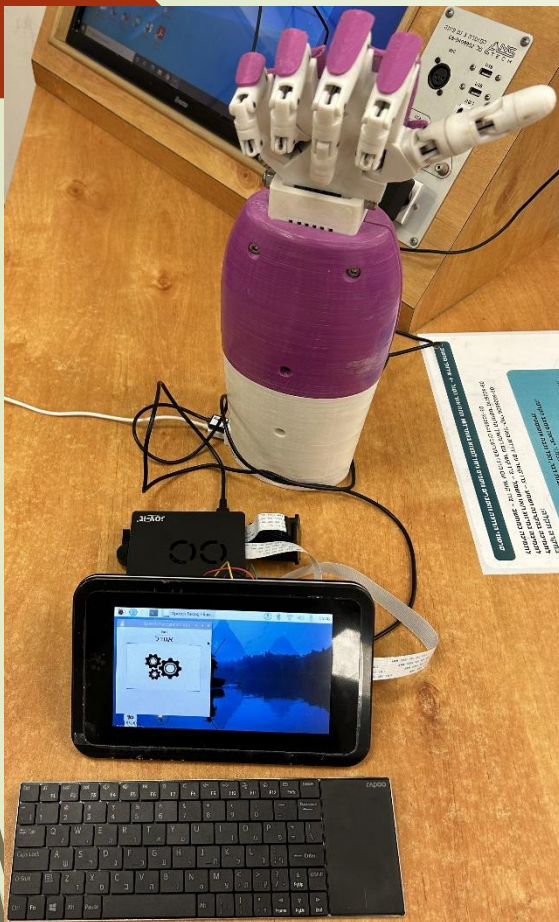


איור מס' 16 מציג את השלב הסופי של פיתוח המערכת, כלומר, השימוש במיקרו-מחשב Raspberry Pi במקום במחשב, שאליו חובר ציוד היקפי תואם ממשפחתו.

1. ספק כוח עבוד כף היד הרובוטית
2. כבל data&power המעביר אותות למנועי כף היד
3. ספק כוח עבור Raspberry Pi
4. קישור למסך מגע מותאם ל-Raspberry Pi בחיבור "2-lane MIPI DSI"
5. חיבור מיני מיקרופון USB
6. מתאם Nano Wireless USB Adapter למקלדת ומשטח מגע אלחוטיים

איור מס' 16. חיבורים בין רכיבי המערכת

תמונות – התפתחות המערכת



איור 18. תגובת היד למשתמש לאחר שאמר את המילה "אגודל" – אגודל מורם, גרסה סופית



איור 17. תגובת היד למשתמש לאחר שאמר את המילה "אגרוף" – יד סגורה לגמרי, שלב פיתוח ראשון (פיילוט)

באיורים 17-18 ניתן לראות את התפתחות המערכת מהשלב הראשון המשמש כפלטפורמה יציבה המריצה את התוכנה וניתן להשתמש בה כסט מוכן עבור ההרצה – בשלב זה המערכת כללה ציוד כמו מקלדת ועכבר חוטיים, מיני מיקרופון וחיבור למסך מחשב.

לאחר מכן התבצע מעבר לציוד היקפי ממשפחת Raspberry Pi – מקלדת אלחוטית עם משטח מגע מובנה, ומסך מגע המותאמים כולם לעבודה עם Raspberry Pi. ניתן לראות שבשתי המערכות השונות היד מגיבה כנדרש למשתמש.

סיכום

לסיכום, העבודה על פרויקט זה עסקה באופן עמוק בהבנה של הקשר שבין חומרה לתוכנה. במהלך פיתוח המערכת היה צורך בביצוע מטלות מורכבות:

- לימוד והבנה כיצד לעבוד עם קבצי קונפיגורציה של מערכת ההפעלה של Raspberry Pi כדי לבצע שדרוג כך שמיני מיקרופון הפך להיות התקן ברירת מחדל לקלט קולי.
- לימוד ושימוש בסט פקודות הטרמינל לעבודה עם התקני USB כמו: `lsusb`, `ls/dev/usb`, `aplay -l`, `arecord` המאפשרים גישה דרך הטרמינל להתקנים המחוברים לכניסות ה-USB כמו מקלדת, עכבר, מיני מיקרופון ומצלמה, ומאפשרים להפעיל את חלקם (כמו מיני מיקרופון או מצלמה).
- פיתוח חלון אפליקציה בשימוש ספריית פייתון `tkinter`.
- בחירת הספרייה המתאימה ביותר להמרת הקול לטקסט, לאחר ביצוע ניסויים עם כמה ספריות קול כמו: `google`, `whisper`, `sphinx`.

בסיום הפיתוח נוצרה מערכת לשליטה קולית על כף יד רובוטית הכוללת חומרה ותוכנה. למערכת מחוברים רכיבים ממשפחת Raspberry Pi שהפכו אותה לקומפקטית וייצוגית. על המערכת רצה אפליקציה חלונאית השולטת על כף היד הרובוטית באמצעות פקודות קוליות מהמשתמש.

מסקנות

- Raspberry Pi היא פלטפורמה המספקת תנאים אופטימליים לעבודה על פרויקט של חומרה-תוכנה, היכולות שלו מגוונות מאוד והוא מספק יעילות טובה ביחס לפרויקטים של שליטה קולית.
- שימוש בחומרה המספקת קלט עבור אפליקציה דורש לעיתים שינויים בקבצי קונפיגורציה והתאמתם לעבודה בסביבת הפיתוח.
- הפיתוח יצר מערכת מבוזרת בין ענן גוגל ל-Raspberry Pi (מבחינת משאבים – חישוביות וזיכרון): מצד אחד עושה שימוש בענן של גוגל לצורך המרת הקול לטקסט ומצד שני ביצוע שאר פעולות המערכת נתמכות במשאבי Raspberry Pi.

תוכניות עתידיות

- הפלטפורמה שנוצרה מהווה בסיס טוב לפרויקטים עתידיים כמו שילוב במערכות הדורשות משאבים נוספים כדי להרחיב את יכולותיהן.
- נרצה בעתיד לשלב אלגוריתם "חכם" יותר העובד עם מודל חכם שניצור ונאמן או עם מערכת AI כמו Chat GPT.
- נרצה בנוסף לשלב את הידע שרכשנו בנושא ה-UI על מנת ליצור אפליקציה אחת המשלבת את הפרויקט הנוכחי עם מערכת ראייה ממוחשבת.

ביבליוגרפיה

1. פורמים לעזרה:

• <https://stackoverflow.com/>: StackOverflow

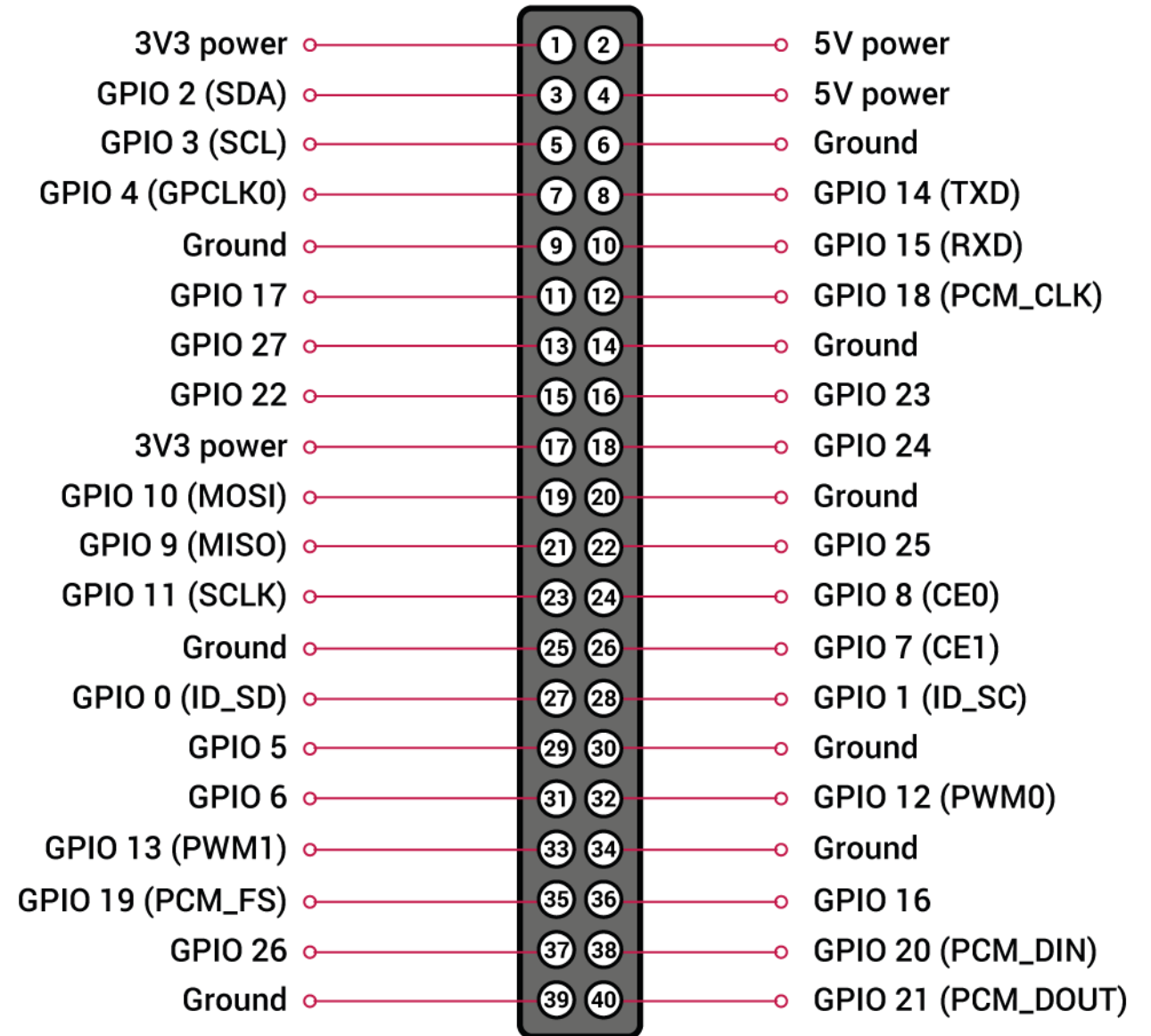
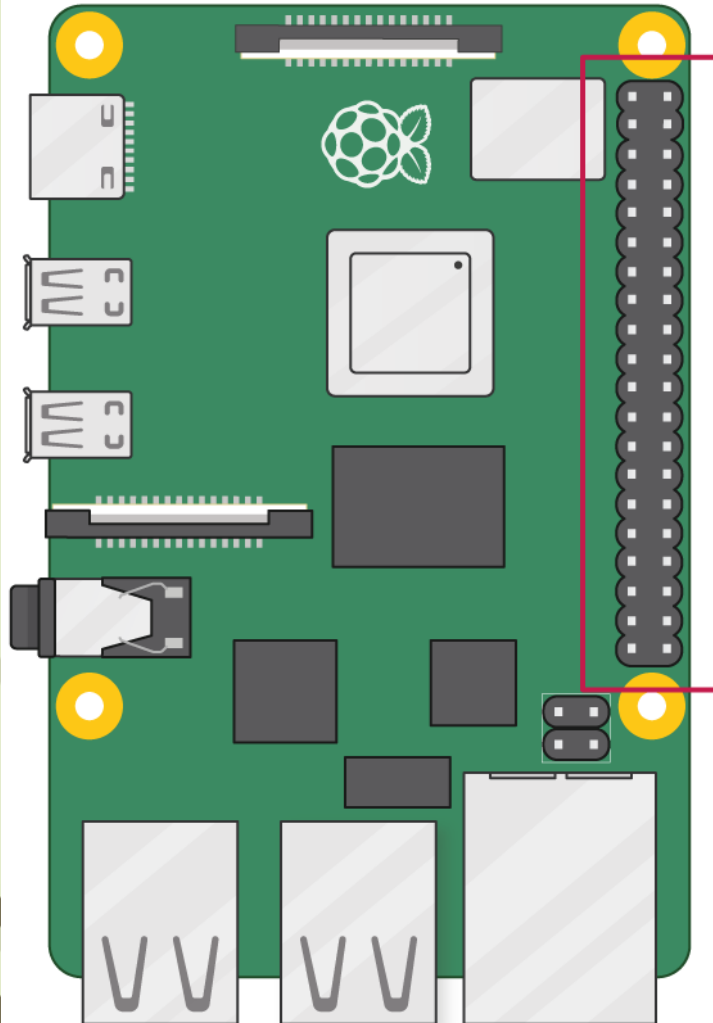
• <https://github.com/community>: GitHub Forum

• <https://forums.raspberrypi.com/>: Raspberry Pi Forum

2. הכנת הדיאגרמות: DrawIO <https://app.diagrams.net/>

3. התקנת מסך מגע: https://www.youtube.com/watch?v=SIUfAliSzJA&ab_channel=MakeUseOf

Raspberry Pi 4 GPIO Bus – 1 נפח



נספח 2 – דוגמה לחיווט כבל Data&Power ל-GPIO Bus

