תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תכנות מעבדי DSP

**P4- Heartbeat Detection**

**מגישים:**

**שמעון קריכלי 212141428**

**יוסי ממו 312422603**

**בהנחיית:**

מר קרוין יצחק

הוגש בתאריך: 20.6.2023

תוכן עניינים

[2 דרישות הפרויקט 3](#_Toc138109484)

[3 דיאגרמת בלוקים של התוכנה 3](#_Toc138109485)

[4 תרשים זרימה של המערכת 4](#_Toc138109486)

[5 שיקולי תכנון 6](#_Toc138109487)

[5.1 שיקולי התכנון עבור המסנן 6](#_Toc138109488)

[5.2 שיקולי התכנון עבור התוכנית 9](#_Toc138109489)

[5.2.1 תצורות 9](#_Toc138109490)

[5.2.2 פונקציות התוכנית 12](#_Toc138109491)

[6 תוצאות ההרצה 18](#_Toc138109492)

[7 Execution graph, Task and CPU load 20](#_Toc138109493)

[7.1 Execution graph 20](#_Toc138109494)

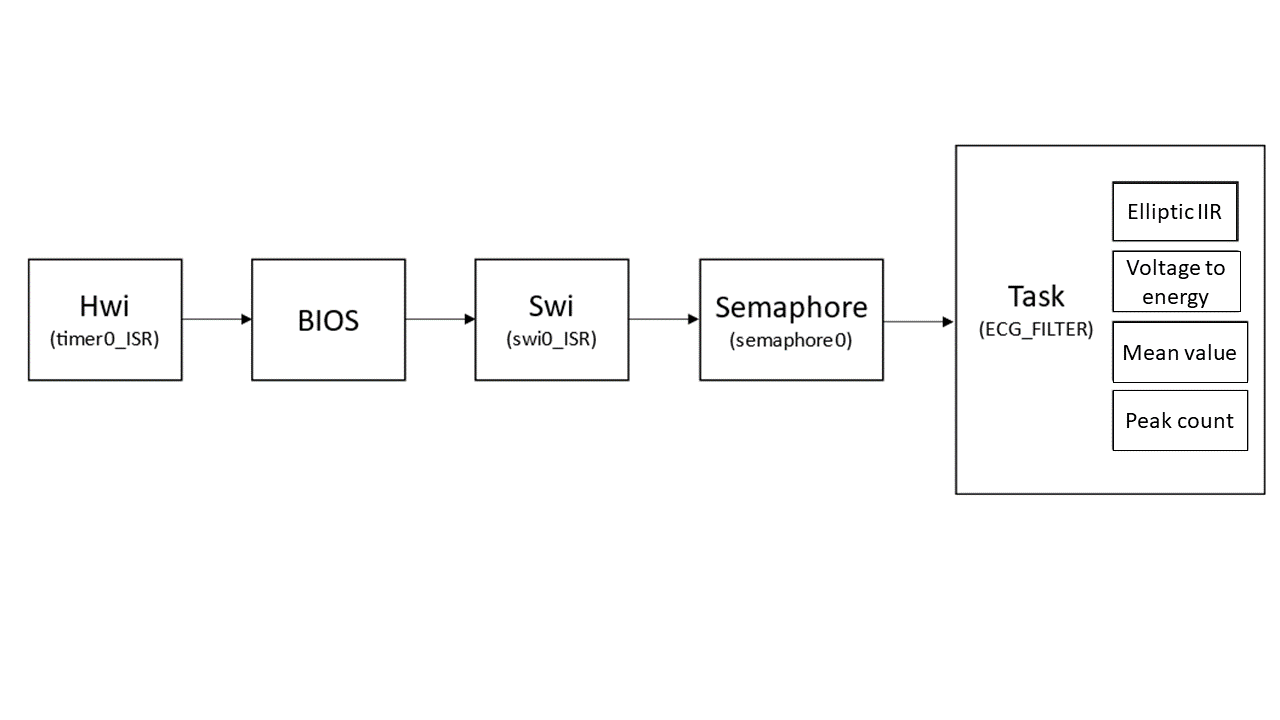
[8 בעיות הנדסיות ופתרונן 21](#_Toc138109495)

# דרישות הפרויקט

* השמת אות דופק נתון במערך בגודל 1788 ודגימתו בתדר דגימה של HZ800
* שימוש ב- Timer אשר דוגם את האות בתדר הדגימה שנקבע ובאופן רציף וציקלי
* שימוש בתוכנת MATLAB לתכנון שני מסננים LP ו-HP על מנת ליצור BP לא סימטרי המתאים לסינון התדרים הלא רצויים ושיפור הזיהוי של הדופק. בהמלצת המרצה נבחר מסנן מסוג IIR
* זיהוי מרווח הזמן בין הדפקים הראשיים (בין S1 ל- S1) ובין דופק ראשי לדופק משני (בין S1 ל- S2) וחישוב קצב פעימות הלב על סמך מרווח הזמן בין הדפקים הראשיים
* הצגה על מסך ה- console את קצב פעימות הלב, מרווחי הזמן בין הדפקים והאינדקסים שלהם במערך הדגימות  
  בפורמט bpm .
* הצגת הגרפים של האות בשלבים השונים של העיבוד במישור הזמן ובמישור התדר
* הצגת Execution graph, CPU load, Task load

# דיאגרמת בלוקים של התוכנה

דיאגרמת הבלוקים בדגש על ארכיטקטורת multi-threading בזמן אמת:



Task0

איור מס. 1 דיאגרמת הבלוקים של התוכנה

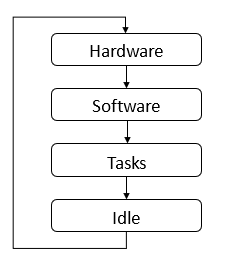
# תרשים זרימה של המערכת

ה- Priority של ה- threads במערכת RTOS הינה בסדר הבא:



איור מס. 2 ה- priority של ה-threads

וסדר ביצוע הפעולות הינו:



איור מס. 3 סדר ביצוע ה-threads בכל הרצה

תרשים הזרימה של התוכנית:

A picture containing text, diagram, screenshot, font

Description automatically generatedA picture containing text, number, screenshot, receipt

Description automatically generatedA picture containing text, font, diagram, circle

Description automatically generatedA picture containing text, diagram, font, screenshot

Description automatically generated

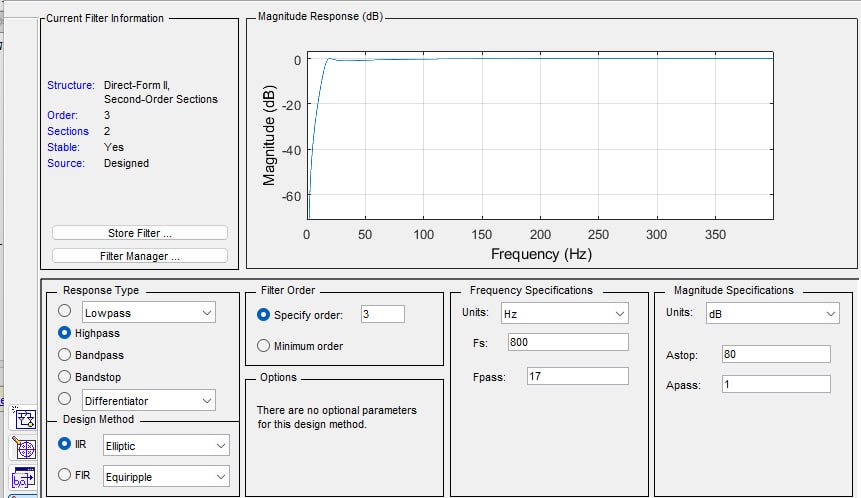
איור מס. 4 תרשים הזרימה של התוכנית

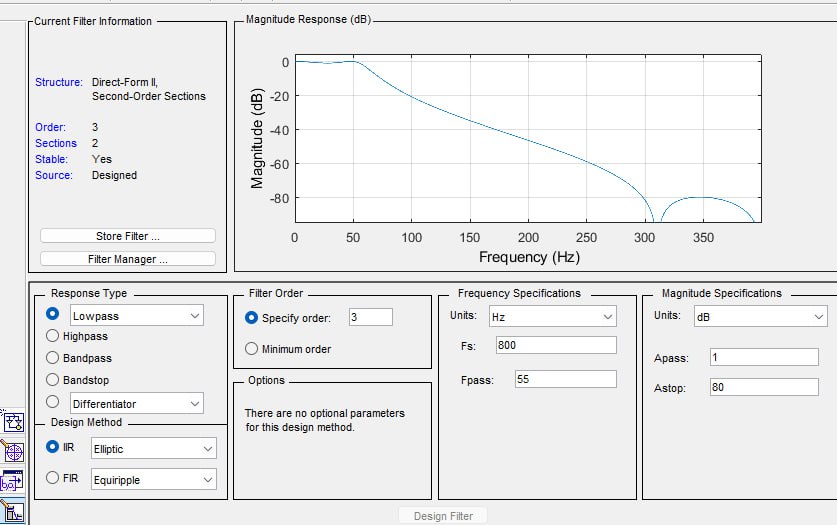
# שיקולי תכנון

מטרת הפרויקט היא לחשב את קצב פעימות הלב של אדם ולהציג אותו בפורמט bpm בהינתן אות הדופק לב. ראשית, ביצענו את תכנון המסננים באמצעות אפליקציית Filter Designer ב- MATLAB. בחרנו בשני המסננים מסוג IIR אליפטי וLPF מסדר =3 וHPF מסדר =3, מתוך השיקולים שיוסברו בהמשך. עבור מימוש התוכנית, נעזרנו בתוכנית מתוך מעבדה מספר 6 בקורס (IIR Filter) וקובץ תבנית שהתקבל מהמרצה.

## שיקולי התכנון עבור המסנן

להלן תצוגת המסנן הנבחר כפי שמופיעה ב- Filter Designer ב-MATLAB:





איור מס. 5 תכנון המסנן ב- Filter Designer ב- MATLAB

תכנון המסנן בוצע במספר שלבים:

1. על מנת להבין אילו תדרים עלינו לסנן, בשלב הראשון הסתכלנו על אות במישור התדר ובחנו את התדרים הרצויים.
2. תחילה בוצעה עבודה עם מסנן יחיד מסוג LPF.
3. היות ומסנן מסוג זה מסנן תדרים נמוכים אבל את הגבוהים מעביר, עברה ההפרעה הסינוסואידלית של האות בתדר נמוך ולכן נצטרך לתכנן מסנן שונה.
4. לאחר התייעצות עם המרצה, על מנת לזהות באופן מיטבי את התדרים המתאימים עבור S1 ו- S2 עם מינימום רעשים מסנן BPF לא סימטרי הצליח לפתור את הבעיה, הרכבתו התבצע באמצעות שילוב מסנן LPF ו-HPF
5. על מנת להבין אילו תדרים עלינו לסנן, בשלב הראשון הסתכלנו על אות במישור התדר המרתו לFFT ובחנו את התדרים הרצויים. באמצעות ניסוי וטעיה מצאנו שהמסנן אמור LPF בעל תדר העברה של 55 וHPF בעל תדר העברה של 17.
6. מסנן HPF מסנן את הרעשים בתדרים הנמוכים ומסנן ה-LPF מסנן את התדרים הגבוהים.
7. קבענו את סוג המסנן להיות IIR מכיוון שהוא יעיל יותר מבחינה חישובית. ניתן ליישם אותו באמצעות מספר קטן של מקדמים לעומת ה-FIR מה שהופך אותו לחסכוני מבחינת חומרה ומשאבים חישוביים.
8. בחרנו את מסנן ה- IIR להיות אליפטי מכמה סיבות:
9. ראינו שהוא מעניק לנו Transition band חד יותר לעומת הסוגים האחרים.
10. ראינו שיש לו יכולת גבוהה של הנחתה ב- Stop band.
11. ראינו שעבורו אנחנו מקבלות סינון טוב עם מעט מקדמים.
12. בחרנו את סדר המסנן להיות 3 על מנת לקבל רמת סינון מספקת בכמות מינימלית של חישובים. תחילה ביצענו במערכת השמה של מסנן עם דרגה שתיים וראינו שהתוצאות אינן טובות. לאחר מכן ביצענו תכנון חוזר של המסנן אך הפעם עם שלוש דרגות וראינו שתוצאות השתפרו מאוד.

להלן המפרט של המסנן:

Filter type: IIR

Response: Lowpass

Number of sections: 2

Filter order: 3

Design method: elliptic

Passband frequency: 55Hz

Cutoff frequency (-3dB point): 60Hz

Half power frequency (-6dB point): 66Hz

Stopband frequency: 300Hz

Transition band width: 245Hz

Passband ripple: 1dB

Attenuation Rate:

Filter type: IIR

Response: High-pass

Number of sections: 2

Filter order: 3

Design method: elliptic

Passband frequency: 17Hz

Cutoff frequency (-3dB point): 15.5Hz

Half power frequency (-6dB point): 14.1Hz

Passband ripple: 1dB

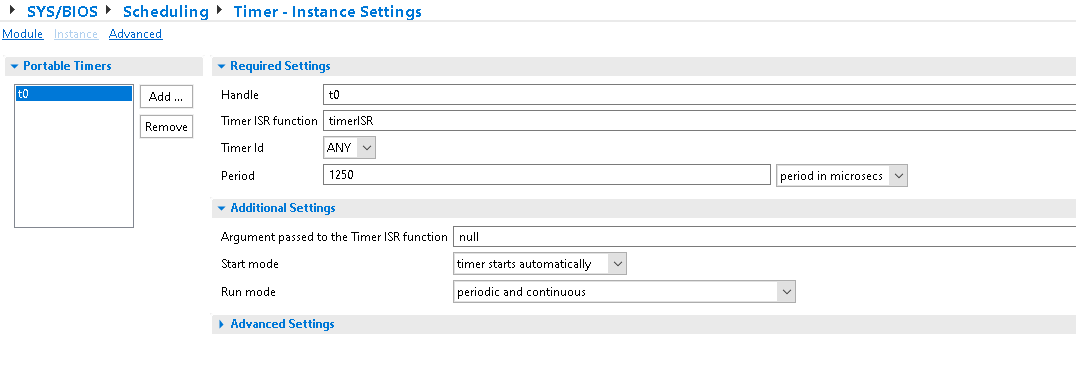
Attenuation Rate:

## שיקולי התכנון עבור התוכנית

### תצורות

#### Timer

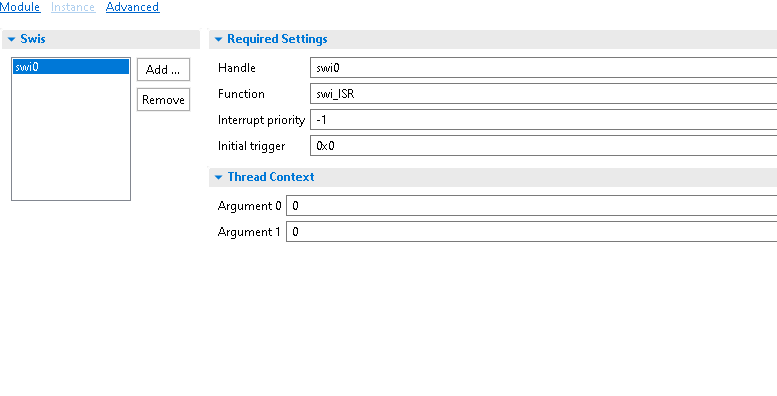
timer0 הינו מונה אשר ניתן להשתמש בו ליצירת פסיקות תקופתיות. המונה מופעל בצורה אוטומטית , ומוגדר כך שסופר בצורה מחזורית כלומר כאשר מסיים מחזור גל שלם יחזור לנקודת ההתחלה ויספור מחדש.  
טיימר t0 סופר לפי מיקרו שניות ולכן זמן המחזור נקבע לפי תדר בדגימה של הגל (800 הרץ), ולכן על מנת למצוא את הזמן מחזור בוצע חישוב פשוט של 1 חלקי תדר הדגימה. בפרויקט זה הטיימר משמש לדגימת האות, בכל פעם שהוא מופעל נבצע post לפסיקת תוכנה.



איור מס. 7 תצורת ה-Timer

#### Swi

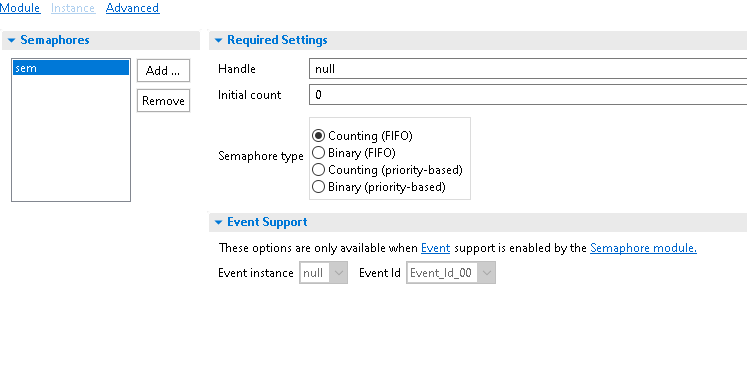
swi0 הינו פסיקת תוכנה עבור המעבד. משתמשים בו על מנת לעבור בין משימות, כך שמשימה בעלת עדיפות גבוהה יותר (מה שבתוך שגרת הפסיקה של swi0) מקדימה משימה בעלת עדיפות נמוכה יותר. במקרה שלנו, הקריאה ל- swi0 מתרחשת בתוך שגרת הפסיקה של timer0, כלומר בכל 1250 מיקרו-שניות. בשגרת הפסיקה של swi0 אנו קוראים לtask0 ומבצעים את כלל הפעולות על האות.



איור מס. 8 תצורת ה- Swi

#### Semaphore

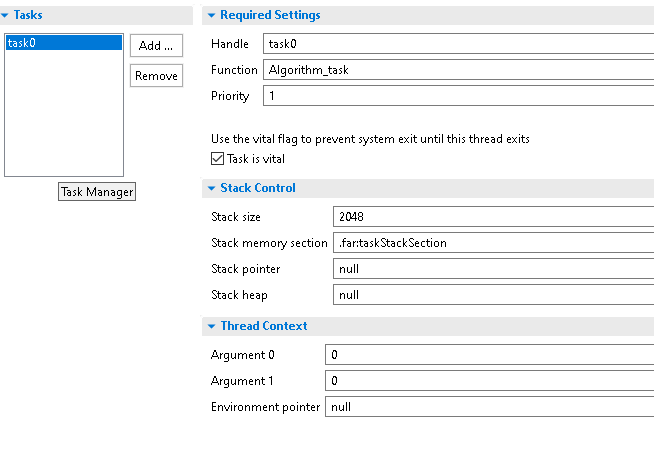
semaphore0 משמש לסנכרון תהליכים מרובים. הוא מאפשר ל- thread אחד לשלוט מתי threads אחרים יכולים לגשת למשאב משותף, תוך שהם ממשיכים בעבודתם בזמן ההמתנה. בכך הוא מבטיח שרק thread מסוים ישתמש במשאב משותף בכל רגע נתון. במקרה זה, ה- semaphore מתזמן את הקריאה ל- TASK: בסוף שגרת הפסיקה של swi0 ה- semaphore מאותת על משאב זמין (באמצעות הפקודה semaphore\_post) בעוד שבפונקציה TASK ה-semaphore מאותת על המתנה למשאב (באמצעות הפקודה semaphore\_pend).



איור מס. 9 תצורת ה- Semaphore

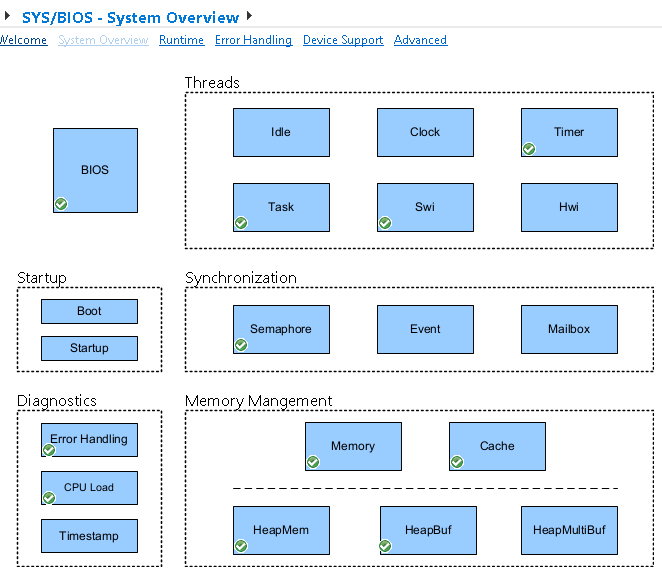
#### Task

זה יחידת העבודה שמבוצעת על ידי התוכנית. ניתן להשתמש בו כדי לארגן ולבצע פעולות בסדר מסוים או במרווחי זמן ספציפיים. במקרה שלנו הוא מעין הפונקציה הראשית בתוך לולאה אינסופית שקוראת לכל הפעולות ואלגוריתמים לעבוד על האות הדגום, בין הפעולות: energy calc, mean value, peak\_counts.



איור מס. 10 תצורת ה- Task

#### System Overview



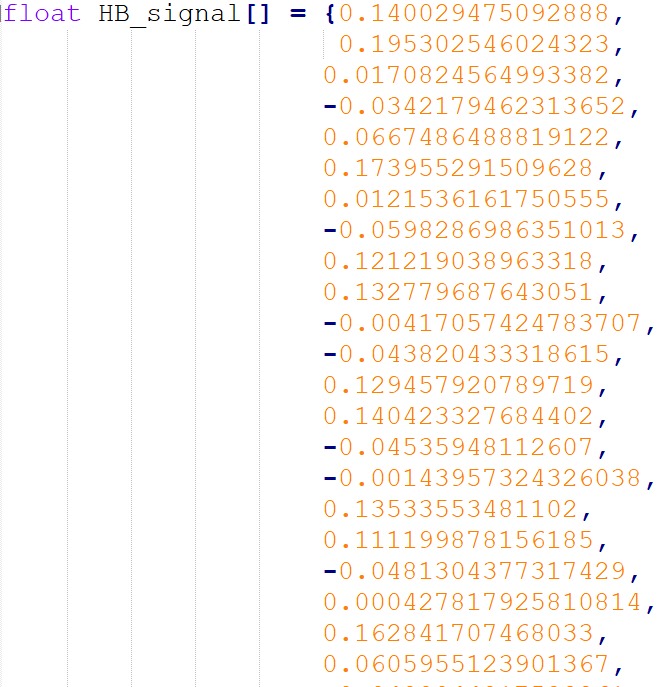
איור מס. 11 System Overview

### התוכנית

#### signal

המערך נקרא HB\_signal והוא מכיל 1788 ערכים המייצגים את האותות החשמליים של הלב לאורך זמן. קובץ זה ניתן לנו מהמרצה ואין אנו מימשנו אותו, מלבד השמתו בתוכנית. בתוכנית הראשית קיים מערך של האות אשר מקודם בכל פעם שטיימר מסיים לספור זמן מחזור שלם.

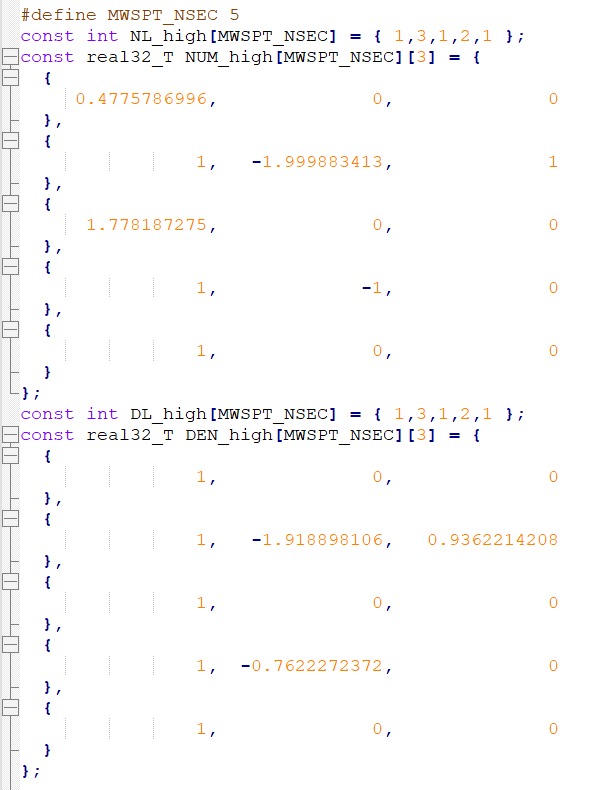
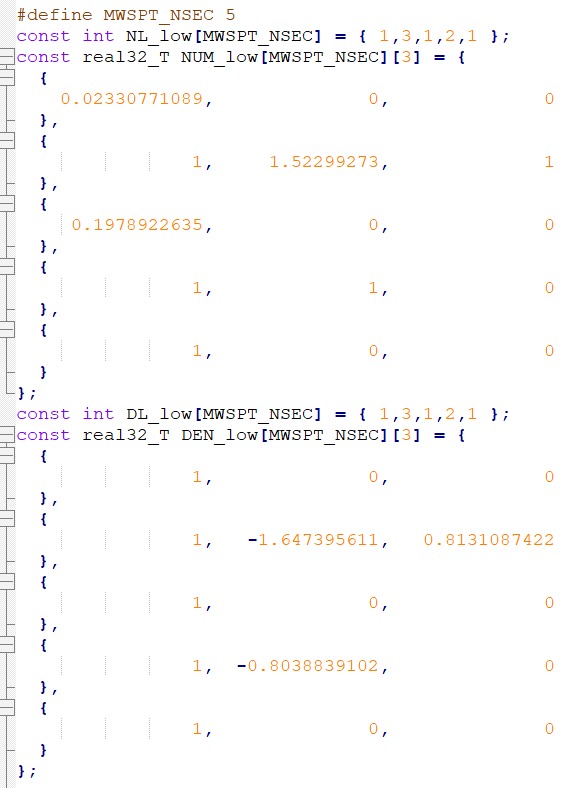
להלן חלק מקטע הקוד:



איור מס. 12 HB\_signal

#### Coefficients

אלה שני קבצי header בהם מגדירים מערכים דו מימדיים המכילים את המקדמים של המסננים IIR שתיכננו באמצעות תוכנת MATLAB. אלו הם הערכים המספריים המגדירים את המאפיינים של המסנן בהתאם לפרמטרים שהכנסנו בממשק ה-Filter Designer ב- MATLAB.

להלן קטע הקוד:

איור מס. 13 coefficients

#### timerISR ()

A close-up of a computer code

Description automatically generated with low confidence

איור מס. 14 שגרת הפסיקה של ה- Timer

#### swi\_ISR ()

בשגרת פסיקה זו מתבצעת דגימת ערך מהמערך HB\_signal ושמירת הערך בתוך משתנה בשם sample. בסוף הפסיקה מתבצעת קריאה ל-semaphore לאיתות על משאבים זמינים.

A picture containing text, screenshot, font, line

Description automatically generated

איור מס. 15 שגרת הפסיקה של Swi

#### signal filtration-task

את האות שקיבלנו אנו מעבירים דרך מספר בלוקים:



A screenshot of a computer

Description automatically generatedלהלן האות המקורי:

FFT:

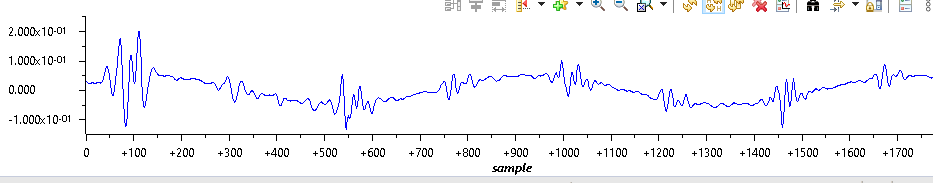
A screenshot of a computer

Description automatically generated

ניתן לראות את מה שהוביל לאינטואיציה: הפרעה קרוב לתדר DC – HPF

הפרעה בתדר גבוה – LPF

1. A picture containing text, line, diagram, screenshot

   Description automatically generatedתחילה העברנו את האות דרך LPF שתיכננו על מנת לסנן את הרעשים הגבוהים ולהלן התוצאה:

1. לאחר מכן, העברנו את האות שהתקבל דרך HPF כדי לסנן את ההפרעה הסינוסואידלית:

A picture containing text, line, plot, screenshot

Description automatically generatedA screenshot of a computer

Description automatically generated

1. A screenshot of a computer

   Description automatically generated with medium confidenceאחר כך על מנת לראות את דפיקות הלב בתצורת אנרגיה ולא כאות אקוסטי העלינו את האות בריבוע:
2. לבסוף ביצענו מיצוע בזמן אמת לאות באמצעות באפר בגודל של 40 דגימות שמצאנו שהוא הטוב ביותר מכיוון שהוא המינימלי ביותר שאינו מפספס עליות צמודות באות האנרגיה, הבאפר מקבל את ערכי האות ועושה ממוצע ביניהם עבור כל דגימה:

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing diagram, sketch, technical drawing, drawing

Description automatically generatedאת הבאפר תיכננו כך שלאחר שהוא יתמלא, הוא יבצע overlap והדגימות הבאות ייכנסו בתחילתו

1. לאחר כל המניפולציות על האות, התחלנו לנתח אותו על מנת למצוא את ההפרשים s1-s2 & s1-s1.

זאת עשינו על ידי לקיחת אינטרוול של 60 על מנת שבעת הריצה לא נדרוך על אותו peak

על מנת למצוא פיקים, השתמשנו באלגוריתם Threshold פשוט כאשר הערך המינימלי של הפיק הגבוה ביותר הוא:

A picture containing text, diagram, sketch, black and white

Description automatically generatedעל מנת לקבל ערך זה עברנו על אות האנרגיה וחיפשנו את הנקודה הרלוונטית.

האלגוריתם פעל כך שכל פעם שמצאנו מקסימום מקומי קפצנו קדימה על מנת למצוא את המינימום המקומי הבא.

ובנוסף כל פעם שמצאנו מקסימום מקומי חיפשנו את המקסימום הבא כדי לדעת את קצב הדפיקות.

את ההפרשים בין s1-s2 הדפסנו בקונסול יחד עם קצב הדפיקות שחושב.

A picture containing text, diagram, sketch, origami

Description automatically generated

# תוצאות ההרצה

לבסוף התוצאה שקיבלנו היא שקצב הלב הוא 34BPM וההפרש בין s1-s2 הוא ~220 אינדקסים כלומר:

האות המקורי נראה כך:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedLPF שקיבלנו במטלב נראה כך ובחרנו שתי נקודות לשם השוואה בין המסנן האידיאלי למסנן המעשי שקיבלנו

לאחר העברת האות המקורי דרך המסנן קיבלנו התמרת FFT הזאת והדגשנו את הנקודות הרלוונטיותA picture containing text, line, diagram, screenshot

Description automatically generated:

A picture containing text, line, diagram, plot

Description automatically generated

בתדר 50Hz האות המקורי הוא בעל אמפליטודה של 0.97 ואילו לאחר מעבר דרך הפילטר הוא באמפליטודה של 0.6376 כלומר בוצע פה ניחות מעשית אפילו שלא היה אמור להיות כאן ניחות לפי הפילטר של המטלב.

על פי המטלב הניחות בתדר 50 אמור להיות בערך -0.06db ואילו מעשית קיבלנו -1db, ברור לנו שזאת תוצאה לא מדוייקת בכלל אבל חשוב לציין שהאות שלנו רץ כל הזמן ויכול להיות שאם היינו בודקים בזמן אחר היינו מקבלים תוצאה שונה.

וכמובן המדידות נעשו באמצעות העין ולא מדוייקות.

בנוסף, מכיוון שלא היה לנו גישה תמידית לתוכנה נאלצנו לחשוב על דרים יצירתיות להשלמת הפערים ולכן השתמשנו באתר <https://apps.automeris.io/wpd/> על מנת למצוא נקודות על הגרף ללא יכולת להשתמש בCCS.

A screenshot of a computer

Description automatically generatedHPF שקיבלנו במטלב נראה כך ובחרנו שתי נקודות לשם השוואה בין המסנן האידיאלי למסנן המעשי שקיבלנו

לאחר העברת האות המקורי דרך המסנן קיבלנו התמרת FFT הזאת והדגשנו את הנקודות הרלוונטיות

A picture containing text, line, plot, screenshot

Description automatically generated

A picture containing text, line, screenshot, plot

Description automatically generated

בתדר 250Hz האות המקורי הוא בעל אמפליטודה של 0.5 וגם לאחר הסינון קיבלנו 0.3447.

על פי המטלב הניחות בתדר 250 אמור להיות בערך -0.01db ואילו מעשית קיבלנו -1db, ברור לנו שזאת תוצאה לא מדוייקת בכלל אבל חשוב לציין שהאות שלנו רץ כל הזמן ויכול להיות שאם היינו בודקים בזמן אחר היינו מקבלים תוצאה שונה.

וכמובן המדידות נעשו באמצעות העין ולא מדוייקות.

בנוסף, מכיוון שלא היה לנו גישה תמידית לתוכנה נאלצנו לחשוב על דרים יצירתיות להשלמת הפערים ולכן השתמשנו באתר <https://apps.automeris.io/wpd/> על מנת למצוא נקודות על הגרף ללא יכולת להשתמש בCCS.

# Execution graph, Task and CPU load

## Execution graph

גרף הביצוע מציג באופן חזותי את סדר ביצוע הפקודות על ידי מחשב ומראה כמה זמן לוקח כל חלק בתוכנית. משתמשים בו על מנת למצוא צווארי בקבוק ולספק תובנות מדוע פעולות מסוימות נמשכות יותר זמן מאחרות לשם ייעול ביצועי התוכנה.

להלן גרף הביצוע של התוכנית שלנו:

A picture containing text, screenshot, line, font

Description automatically generated

ניתן לראות שאנו רוב הזמן בidle ואכן לפי הסיכום של התוכנה רואים שאני באמת idle כמעט 100% מהתכנית.

A screenshot of a graph

Description automatically generated with low confidenceנתון זה עומד בדרישה של Real time של זמן idle של יותר מ 80% על מנת שיהיה אפשר להכניס שיפורים עתידיים.

Task load:

A screenshot of a graph

Description automatically generated with medium confidence

ניתן לראות גם כאן בגרף הtask load את היעילות של התוכנית.

# בעיות הנדסיות ופתרונן

|  |  |
| --- | --- |
| בעיה | פתרון |
| חשבנו בתחילה לקחת רק LPF ולהתעלם מהרעש הסינוסואידלי אך הבנו שזה רעיון לא טוב ויפריע לחישובים הבאים | שימוש בשני מסננים: HPF וLPF כדי לייצר BP לא סימטרי |
| מציאת הגודל המתאים לבאפר של המיצוע | ניסוי וטעיה תוך הנחה שהגודל צריך לכסות כמה שיותר דגימות אך לבצע כמה שפחות חישובים |
| במציאת הפיקים חשבנו בתחילה לבצע חיפוש פשוט של שני פיקים אך לא חשבנו על הפיקים הנמוכים יותר ולא לקחנו אותם בחשבון מה שגרם לתוצאה לא הגיונית | חשבנו על אלגוריתם חדש שלוקח בחשבון גם את הפיקים הנמוכים. |