

Прикарпатський національний університет імені Василя
Стефаника

Факультет математики та інформатики

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА МАГІСТЕРСЬКА РОБОТА

Тема: «Система аналізу електричних сигналів серця»

Виконав: студент 2 курсу

гр. КНМ - 21

ОР магістр

Воробій В.Д.

Спеціальність: 122
Комп'ютерні науки

Науковий керівник: к.т.н., доц
Превисокова Н.В.

Рецензент: к.т.н., доц
Ровінський В.А.

Івано-Франківськ, 2022

АНОТАЦІЯ

Дипломна робота: 77 с., 44 рис., 15 табл., 5 додатків, 27 джерел.

Реалізовано систему аналізу електричних сигналів серця. Розроблено мікроконтролерну систему, що слугує для зчитування сигналу з електродів прикріплених до тіла людини та передачі сигналу по протоколу UART. Проведено аналіз та порівняння різних цифрових фільтрів для мінімізації ефекту “коливання осі” в ЕКГ. Система, представлена в даній роботі, може бути застосована у сфері кардіології для полегшення діагностики серцево-судинних захворювань у пацієнтів.

Ключові слова: електрокардіограма, цифрова обробка сигналів, перетворення Фур’є, цифрові фільтри, Вейвлет перетворення.

ABSTRACT

Diploma: 77 p., 44 fig, 15 tabl., 5 appendixes, 27 sources.

Implemented system of ECG signals analysis. Implemented microcontroller system used for reading of signals from electrodes, attached to the human body, and transfer of it by UART protocol. Performed analysis and comparison of digital filters for removal of baseline wandering in ECG. The system demonstrated in this paper can be applied in the field of cardiology to simplify the diagnosis of cardiovascular diseases of patients.

Electrocardiogram, Digital signal processing, Wavelet transform, Fourier analysis, Digital filters.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1. Аналіз ЕКГ	9
1.2. Огляд сучасних засобів та систем діагностики серця	12
РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ СИГНАЛІВ ТА ЇХ ОБРОБКИ	15
2.1. Дискретизація сигналу	15
2.2. Спектральний аналіз сигналу	16
2.3. Методи проведення спектрального аналізу ЕКГ	16
2.3.1. Аналіз перетворення Фур'є та його реалізація	16
2.3.2. Віконне перетворення Фур'є	21
2.3.3. Вейвлети	22
3. Цифрові фільтри	30
РОЗДІЛ 3 РЕАЛІЗАЦІЯ АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ АНАЛІЗУ ЕЛЕКТРИЧНИХ СИГНАЛІВ СЕРЦЯ	40
3.1. Структурна схема пристрою	40
3.2. Аналіз аналогового інтерфейсу AD8232 та вбудованих у ньому фільтрів та підсилювачів	41
3.3. Аналіз джерел випромінювання, що впливають на електрокардіограф	46
3.4. Аналіз АЦП і його основних параметрів	47
3.5. Аналіз протоколу UART і його застосування для передачі сигналу на комп'ютер	49
3.5. Принципова електрична схема системи	51
РОЗДІЛ 4 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ АНАЛІЗУ ЕЛЕКТРИЧНИХ СИГНАЛІВ СЕРЦЯ	53

4.1. Розробка програми для мікроконтролера	53
4.2. Розробка програми для зчитування сигналу на комп'ютері	55
4.3. Програмна реалізація обрахунку частоти серцевих скорочень на основі ЕКГ за допомогою вейвлетів	57
4.4. Програмна реалізація порівняння цифрових фільтрів для мінімізації ефекту “коливання осі” в ЕКГ	59
ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	63
ДОДАТОК А Роздрук програми для мікроконтролера ESP-32	66
ДОДАТОК Б Роздрук програми для зчитування сигналу на комп'ютері	68
ДОДАТОК В. Програма для мінімізації ефекту коливання осі в ЕКГ	71
ДОДАТОК Г. Реалізація перетворення Фур'є мовою програмування Python	75
ДОДАТОК Д. Роздрук програми для визначення ритму серця	76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ЕКГ - електрокардіограма

АЦП - аналогово-цифровий перетворювач

АЧХ - амплітудно-частотна характеристика

ВСТУП

Захворювання серцево-судинної системи – це хвороби кровоносної системи та серцевого м'яза, що є основною причиною смертності у світі. В Україні, згідно з статистикою лікарів Запорізької обласної клінічної лікарні, відсоток серцево-судинних захворювань становить 63,3% у структурі смертності населення. До переліку серцево-судинних захворювань відносяться інсульт, інфаркт, гіпертонія, хвороба периферичних артерій, ревматична вада серця, вроджена вада серця та серцева недостатність.

Наявність хвороби серця професійний лікар-кардіолог визначає спираючись на огляд та розмову з пацієнтом, проте для підтвердження діагнозу застосовуються спеціальні діагностичні дослідження [1].

Електрокардіограма є найбільш поширеним, простим та разом з цим дуже інформативним методом дослідження серця. ЕКГ – це запис електричної діяльності серця, його аналіз дозволяє виявити різноманітні патології серця.

Проте серед експертів в галузі кардіології існують різні підходи щодо трактування результатів ЕКГ, також достатньо високим є відсоток помилок, пов'язаних з хибною інтерпретацією ділянок сигналів ЕКГ [1], що містять дефекти запису. Отже, існує потреба у створенні нових методах обробки ЕКГ.

Об'єкт дослідження - набір кардіограм пацієнтів.

Мета роботи - створення системи аналізу електричних сигналів. Система складається з двох підсистем. Перша підсистема - мікроконтролерна система, що зчитує електричний сигнал з електродів, прикріплених до тіла людини, здійснює його підсилення та очищення його від шумів і передає сигнал на комп'ютер. Друга підсистема - програмна реалізація, що здійснює цифрову обробку і аналіз отриманого сигналу.

Аналізу підлягають частоти QRS комплексу ЕКГ для обрахунку варіабельності серцевого ритму та пульсу людини, що є досить часомістким процесом для виконання людиною. На основі значень цих параметрів лікар-кардіолог з легкістю може визначити наявність аритмії та брадикардії у пацієнта.

Ефект "коливання осі" (baseline wandering) – це одна з найпоширеніших низькочастотних завад, що виникає при русі і диханні пацієнта під час проведення вимірювання [11] і для мінімізації цього ефекту може бути застосований високочастотний фільтр. Проведено аналіз та порівняння різних цифрових фільтрів для мінімізації ефекту “коливання осі” в ЕКГ. Мінімізація цього ефекту спрощує діагностику сигналу для лікаря-кардіолога.

Завдання роботи - проаналізувати сигнал ЕКГ та існуючі засоби для його виміру, проаналізувати методи цифрової обробки сигналів, розробити пристрій для зчитування сигналу з електродів, прикріплених до тіла людини, та передачі сигналу по інтерфейсу UART, проаналізувати джерела випромінювання що впливають на результати ЕКГ, розробити програмну реалізацію для порівняння цифрових фільтрів для мінімізації ефекту коливання осі в ЕКГ.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз ЕКГ

Електрокардіографія – це найпоширеніший метод дослідження серця, що наразі застосовується в кардіології. Приклавши кілька електродів до тіла людини, можна отримати залежність напруги від часу, яка відображає електричну активність серця у певний момент часу. При нормі ця залежність повинна мати форму, еквівалентну рисунку нижче [2].

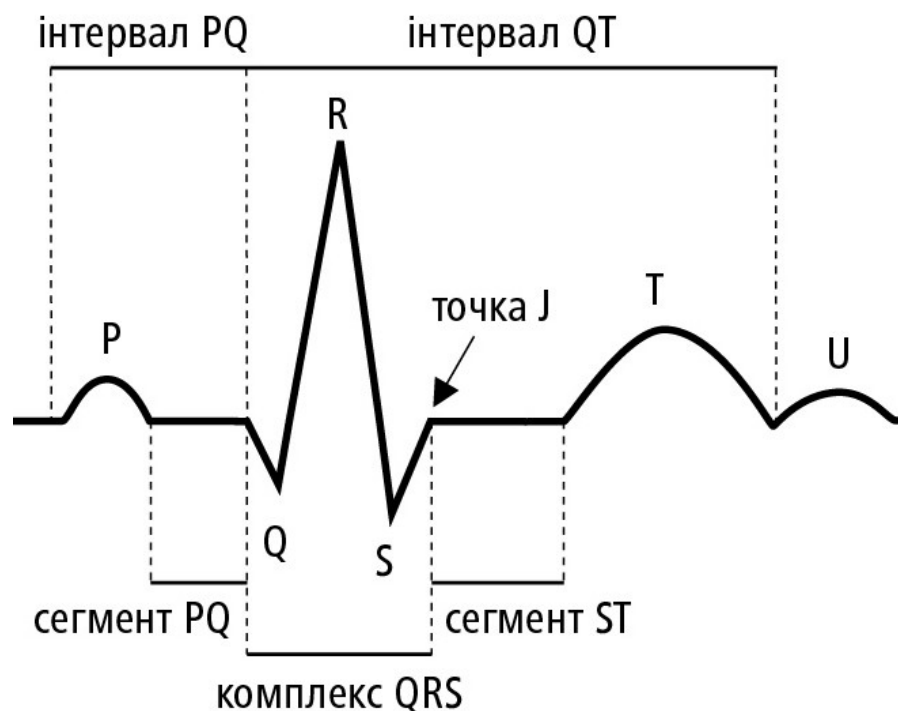


Рисунок 1.1. Нормальна форма ЕКГ

Здебільшого ЕКГ застосовується для функціонального дослідження серця. Також вона допомагає лікарю досліджувати порушення серцевої провідності та ритму, ішемічну хворобу серця, інфаркт міокарда і ряд інших серцевих захворювань [2].

На ЕКГ серцевий цикл розділений на зубці та інтервали, кожен з них відображає певну фазу розповсюдження хвилі збудження у міокарді:

1. Зубець Р – хвиля деполяризації передсердь

2. Інтервал P-Q – хвиля деполяризації до атріовентрикулярного вузла
3. Комплекс QRS
 - 3.1. Q — перше негативне відхилення від ізоелектричної лінії
 - 3.2. R — перше позитивне відхилення
 - 3.3. S — негативне відхилення після R зубця
4. Сегмент S-T - відображає початкову фазу реполяризації міокарда шлуночків
5. Зубець T – відповідає реполяризації шлуночків
6. Хвиля U - з'являється у разі електролітичних порушень

Відведення електрокардіограми - це проекція вектора Ейнтговена на деяку вісь. Вектор Ейнтговена - це інтегральний електричний вектор серця.

Типи відведень:

1. Стандартні
2. Стандартні посилені
3. Грудні
4. Додаткові

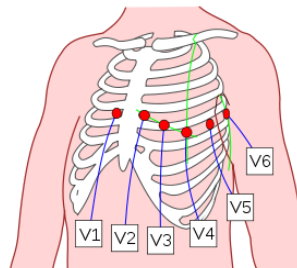


Рисунок 1.2. Схема розташування електродів

Таблиця 1.1.

Стандартні відведення		
Відведення	Увімкнення до “-” полюса гальванометра	Увімкнення до “+” полюса гальванометра
1	права рука	ліва рука
2	права рука	ліва нога
3	ліва рука	ліва нога

Таблиця 1.2.

Стандартні посилені відведення		
Відведення	Увімкнення до “-” полюса гальванометра	Увімкнення до “+” полюса гальванометра
aVR	ліва нога та ліва рука	права рука
aVL	права нога та права рука	ліва рука
aVF	права та ліва рука	ліва нога

Таблиця 1.3.

Грудні відведення		
Відведення	Увімкнення до “-” полюса гальванометра	Увімкнення до “+” полюса гальванометра
V_1	негативний об'єднаний електрод Вільсона	4 міжребер'я (праворуч)
V_2	негативний об'єднаний електрод Вільсона	4 міжребер'я (ліворуч)
V_3	негативний об'єднаний електрод Вільсона	рівень 4 міжребер'я між V_2 та V_4
V_4	негативний об'єднаний електрод Вільсона	5 міжребер'я по лівій середньо-ключичній лінії
V_5	негативний об'єднаний електрод Вільсона	5 міжребер'я по лівій передній пахвовій лінії
V_6	негативний об'єднаний електрод Вільсона	5 міжребер'я по лівій середній пахвовій лінії

1.2. Огляд сучасних засобів та систем діагностики серця

На ринку є дуже багато систем для проведення діагностики серцево-судинних захворювань різного цінового діапазону, енергоефективності, точності тощо. Нижче проаналізовано кілька найпоширеніших варіантів.

1. Електрокардіограф Неасо 100G

Це 12-канальний портативний пристрій, що користується попитом серед терапевтів. Надає можливість попереднього перегляду ЕКГ і його друку на термопапері після виміру. Також присутній слот під SD карту, на яку можна записати до 100 результатів досліджень [12]. Ціна цього пристрою сягає 23100 гривень.

Недоліком цього пристрою є висока ціна.



Рисунок 1.3. Вигляд пристрою Неасо 100G [12]

2. Пристрій ЕКГ РІМА GM-10

Цей пристрій призначений для слідкування за порушеннями серцевого ритму. Сигнал серця під час вимірювання відображається на кольоровому

дисплеї. Окрім того є можливість передавати сигнал по інтерфейсу Bluetooth. Ціна пристрою сягає близько 3000 гривень.

Недоліком цього пристрою є відсутність системи фільтрів для усунення завад.



Рисунок 1.4. Вигляд пристрою PIMA GM-10

3. Цифровий електрокардіограф “Біомед” іЕ 3

Це трьохканальний кардіограф, який оснащений дисплеєм високої роздільної здатності, системою фільтрів, термопринтером тощо. Термопринтер забезпечує одночасний друк трьох відведень ЕКГ і значення пульсу в певний момент часу [13].

Також пристрій дозволяє передавати сигнал по безпроводним інтерфейсам 3G та WiFi. Формат даних, що відображає сигнал може бути обраним користувачем з-поміж таких варіантів як: ECG, XML, JPEG, DICOM та PDF. Також є можливість зберігати результати на SD карту [13]. Вбудований акумулятор забезпечує автономну роботу пристрою на 3 години. Ціна цього пристрою сягає близько 24000 гривень.

Недоліком цього пристрою є висока ціна.



Рисунок 1.5. Вигляд пристрою “Біомед” iE 3 [13]

Висновок: пристрої наведені в цьому підрозділі мають надто високу ціну, при цьому в деяких з них відсутня система фільтрів для усунення шумів. Саме тому існує необхідність в мікроконтролерній системі, що має нижчу ціну і при цьому містить вбудовану систему фільтрів для усунення завад.

РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ СИГНАЛІВ ТА ЇХ ОБРОБКИ

2.1. Дискретизація сигналу

ЕКГ – це неперервна функція напруги від часу. Оскільки сучасні комп'ютери здатні обробляти тільки дискретні сигнали, сигнал ЕКГ перетворюють в цифрову форму шляхом виміру значення сигналу в рівні проміжки часу. Цей процес називається *дискретизацією*. Переважно, за дискретизацію сигналу відповідає аналогово-цифровий перетворювач (АЦП) [9]. Процедура дискретизації проілюстрована на рисунку нижче.

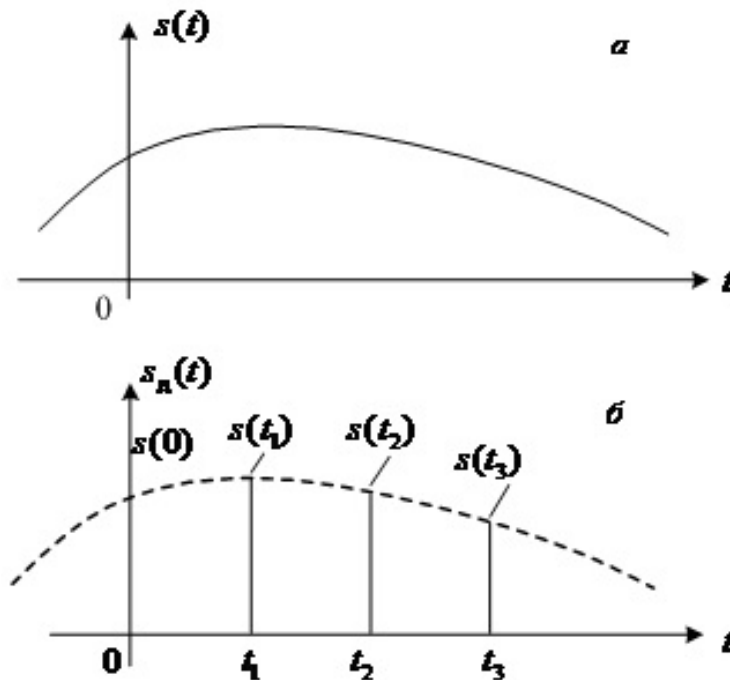


Рисунок 2.1. Дискретизація аналогового сигналу

Значення $s(t(i))$ дискретизованого сигналу називають відліками, а різниця між послідовними відліками ($s(t(i + 1)) - s(t(i))$) називають інтервалом дискретизації. При збільшенні інтервалу дискретизації пропускна здатність каналу зв'язку збільшується, але одночасно збільшується похибка відтворення початкового сигналу $s(t)$ [9].

ЕКГ відноситься до класу сигналів що утворюються у джерелах інформації, вони мають порівняно повільні зміни в часі миттєвих значень та довгу тривалість.

Спектральний аналіз цих сигналів показує, що спектральний склад концентрується в діапазоні частот від 0 до W_b Гц, де W_b - це максимальна частота у сигналі. Гармоніки з частотами вищими від W_b мають дуже низьку амплітуду, відповідно ними можна знехтувати. Ця властивість дозволяє обрати інтервал дискретизації, що дозволить з великою точністю відновити первинний сигнал [9]. Згідно з теоремою відліків Віттекера — Найквіста — Котельникова — Шеннона, якщо сигнал $f(t)$ має спектр частотою M , то його можна відновити без втрат за дискретними відліками, що взяті з частотою дискретизації $f = 2 * M$. При цьому функція $f(t)$ повинна бути неперервна на своїй області визначення [14].

2.2. Спектральний аналіз сигналу

Спектральний аналіз сигналу потрібний для того щоб з'ясувати які частоти присутні в сигналі та їх амплітуду. Математично це реалізується шляхом представлення сигналу $f(t)$ як ряд коефіцієнтів. Коефіцієнт це сума добутків сигналу $x(t)$ на функцію $\psi_n(t)$, $n \in Z$.

$$c_n = \int x(t) \psi_n(t) dt$$

Цей добуток рахує “ступінь схожості” двох функцій. Чим значення більше, тим функції більш схожі. Реалізація функції $\psi_n(t)$ залежить від методу спектрального аналізу.

2.3. Методи проведення спектрального аналізу ЕКГ

2.3.1. Аналіз перетворення Фур'є та його реалізація

Перетворення Фур'є це найпоширеніше інтегральне перетворення у науці та інженерії, що дозволяє розрахувати спектр частот для сигналів змінних у часі. Воно бере свій початок із рядів Фур'є, що є способом представлення довільної функції сумою простіших функцій. Точність відтворення функції визначається кількістю функцій, що враховується при розрахунку. Переважно в якості найпростіших

функцій застосовують тригонометричні функції синуса та косинуса. Ряд Фур'є в цьому випадку називається тригонометричним і задається наступною формулою:

$$\frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)]$$

де $\frac{a_0}{2}$ - це нульова гармоніка (середнє арифметичне сигналу $f(x)$), a_n, b_n - коефіцієнти ряду Фур'є.

Перетворення Фур'є є по суті продовженням рядів Фу'є, для функцій що мають період, який може наближатись до нескінченності [18].

Перетворення Фур'є є комплексною функцією, оскільки вона приймає частоту і повертає комплексне число.

$$F(w) = \int f(t) e^{-iwt} dt$$

Модуль комплексного числа відображає амплітуду синусоїди з цією частотою. Геометричною інтерпретацією модуля комплексного числа є відстань до точки $(0, 0)$.

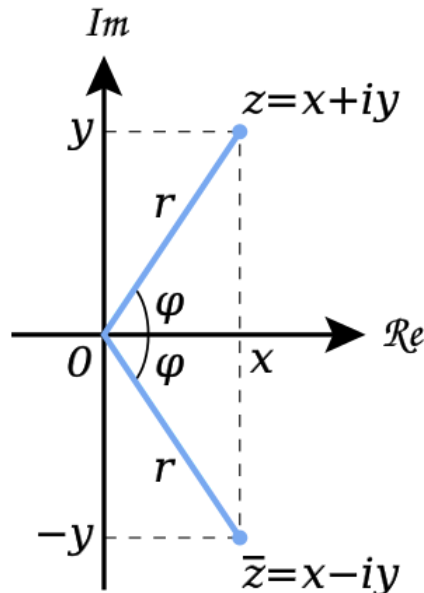


Рисунок 2.2. Геометрична інтерпретація модуля комплексного числа [18]

Відповідно модуль комплексного числа $z = x + iy$ рахується за наступною формулою:

$$|z| = \sqrt{\text{Re}(z)^2 + \text{Im}(z)^2} = \sqrt{x^2 + y^2}$$

Аргумент комплексного числа відображає зсув фази. Геометричною інтерпретацією аргументу комплексного числа є кут між віссю x та вектором $z = x + iy$.

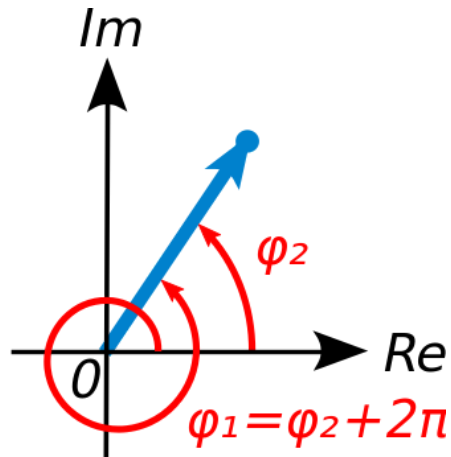


Рисунок 2.3. Геометрична інтерпретація аргументу комплексного числа [18]

Обернене перетворення Фур'є дозволяє з результату перетворення Фур'є отримати функцію для якої це перетворення Фур'є було здійснено. Математично воно задається виразом:

$$\frac{1}{2\pi} \int F(w) e^{iwt} dw = f(t)$$

де $F(w)$ перетворення Фур'є функції $f(t)$

Оскільки обчислення цих інтегралів є неможливим на комп'ютері, то існує видозміна перетворення Фур'є - дискретне перетворення Фур'є, що реалізується на комп'ютері.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i\pi}{N}kn} = \sum_{n=0}^{N-1} x_n \left[\cos\left(\frac{2\pi}{N}kn\right) - i \sin\left(\frac{2\pi}{N}kn\right) \right]$$

Проте ця реалізація занадто повільна, оскільки складність алгоритму зростає квадратично від розміру вхідного вектора - $O(n^2)$ [19].

Алгоритм Кулі — Тьюкі - це найбільш поширена реалізація дискретного перетворення Фур'є з складністю $O(n \log(n))$. Як і усі інші алгоритми перетворення

Фур'є, цей алгоритм широко використовується у приладах з цифровою обробкою сигналів, комп'ютерному моделюванні природних процесів тощо [20].

Алгоритм спочатку обчислює дискретне перетворення Фур'є для парних елементів $x_{2m} = x_0, x_2, \dots, x_{n-2}$ та непарних $x_{2m+1} = x_1, x_3, \dots, x_{n-1}$. Потім об'єднює ці результати для отримання дискретного перетворення Фур'є усєї послідовності. Обчислення виконується рекурсивно.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}$$

Варто зазначити що алгоритм працює тільки за умови якщо розмір вхідного вектора є степенем двійки ($N = 2^m$, m – ціле число) [20].

Нижче наведена ілюстрація роботи алгоритму для $N = 8$

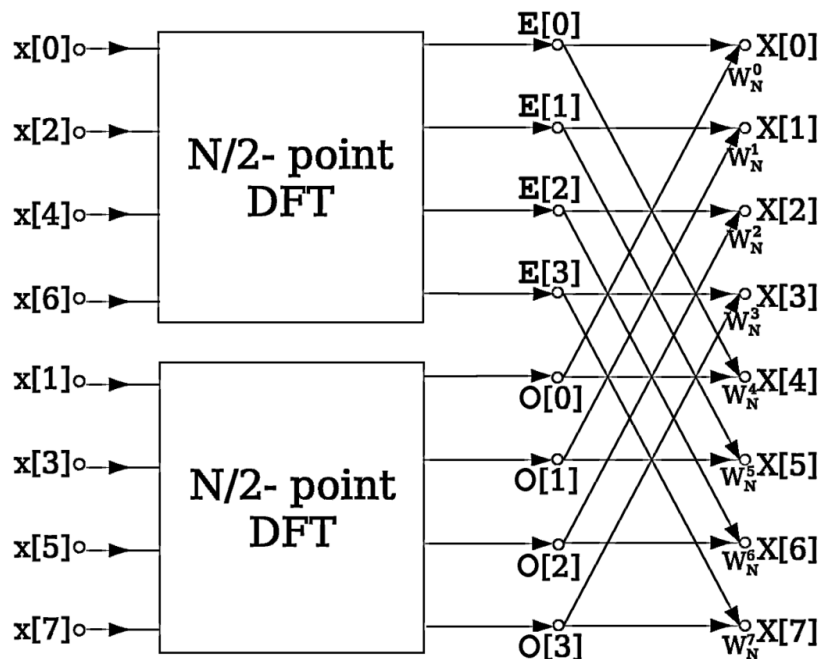


Рисунок 2.4. Ілюстрація роботи алгоритму Кулі — Тьюкі [20]

Реалізація перетворення Фур'є мовою програмування Python

Програма зчитує сигнал з файлу у форматі MAT, обраховує перетворення Фур'є за допомогою функції `fft` з пакету `Scipy` і відображає вхідний сигнал і його перетворення за допомогою функцій з пакету `Matplotlib`.

Результат виконання програми наведений нижче:

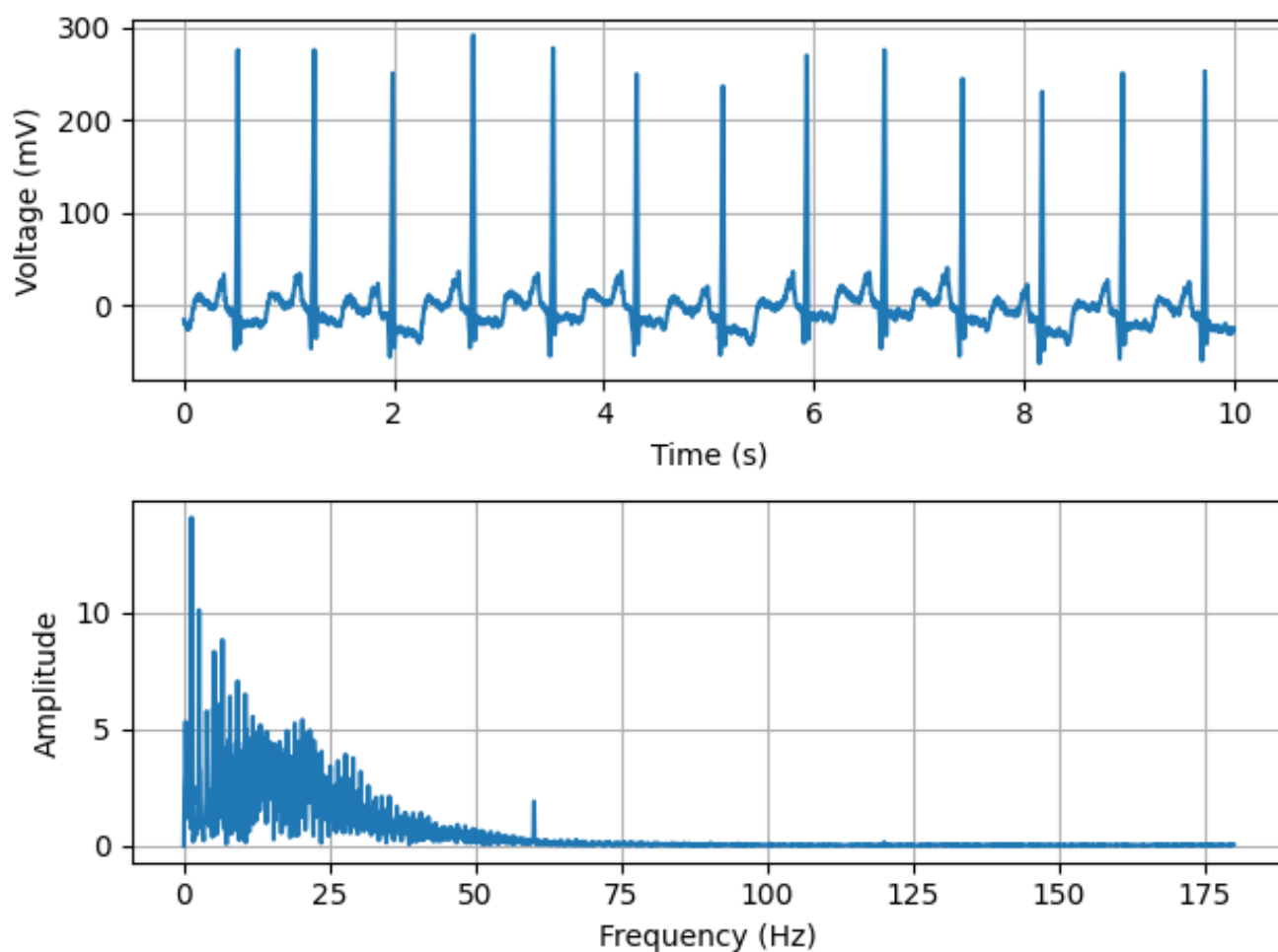


Рисунок 2.5. ЕКГ сигнал і його перетворення Фур'є

Недоліком класичного перетворення Фур'є є те, що воно не дає уявлення чи сигнал містить конкретну частоту в певний момент часу [19]. Для сигналів, в яких частоти не змінюються з часом цей недолік не є суттєвим. Проте, оскільки ЕКГ складається з частот які змінюються з часом, цей недолік є значним. Відповідно застосовувати його для аналізу частотного спектру сигналу ЕКГ, на мою думку, не є доцільним.

2.3.2. Віконне перетворення Фур'є

Щоб вирішити вищезгадану проблему, може бути застосоване віконне перетворення Фур'є. Віконне перетворення Фур'є – це видозміна класичного перетворення Фур'є, що використовується для отримання синусоїдної частоти, що є змінною в часі. На практиці, процедура обрахунку віконного перетворення Фур'є зводиться до розділення вхідного сигналу на сегменти і обрахунку перетворення Фур'є над кожним з них. Математично воно задається наступним чином.

$$F(m, w) = \sum_{n=-\infty}^{\infty} f[n] w[n - m] e^{-jwn}$$

де w – деяка віконна функція, f - вхідний сигнал, m - розмір вікна. сигнал

Недоліком віконного перетворення Фур'є є те що його важко інвертувати через велику надлишковість. Також, це перетворення не може виявити сигнал із різким збільшенням інтенсивності і подальшим різким поверненням до початкової величини [21]. Наприклад, сигнал імпульсу.

Якщо розмір вікна є занадто малим то перетворення погано виявляє низькочастотні гармоніки, оскільки згідно з принципом невизначеності функція не може бути одночасно сконцентрованою в фізичному просторі і просторі Фур'є.

Також, оскільки результатом віконного перетворення Фур'є є не спектр вхідного сигналу, а спектр добутку сигналу і віконної функції, то в результаті виникає ефект розтікання спектру сигналу.

Для мінімізації впливу цього ефекту застосовують більш гладку віконну функцію. Спектр цієї функції повинен мати більш широку головну частку і при цьому низький рівень бічних часток [21].

Типи віконних функцій:

1. Прямокутне вікно

$$w(n) = \begin{cases} 1, & n \in [0, N - 1] \\ 0, & n \notin [0, N - 1] \end{cases}$$

2. Вікно Хеннінга

$$w(n) = 0.5 \left(1 - \cos\left(\frac{2\pi n}{N - 1}\right) \right)$$

3. Вікно Хеммінга

$$w(n) = 0.53836 - 0.46164 \cos\left(\frac{2\pi n}{N - 1}\right)$$

4. Вікно Блекмана

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N - 1}\right) + a_2 \cos\left(\frac{4\pi n}{N - 1}\right)$$

$$a_0 = \frac{1 - \alpha}{2}; \quad a_1 = \frac{1}{2}; \quad a_2 = \frac{\alpha}{2}$$

5. Вікно Хайзера

$$w(n) = \frac{|I_0\left(\beta \sqrt{1 - \left(\frac{2n - N + 1}{N - 1}\right)^2}\right)|}{|I_0(\beta)|}$$

2.3.3. Вейвлети

Кращим рішенням є використання вейвлетів. Вейвлет - це хвилеподібне коливання з амплітудою що починається з нуля, збільшується або зменшується, і потім повертається до нуля один або більше разів. При цьому це коливання повинно бути локалізоване за часом та частотою [22].

Локальність за часом означає що енергія функції сконцентрована на обмеженому інтервалі. Енергія E_s неперервної функції $f(t)$ задається наступною формулою:

$$E_s = \int_{-\infty}^{\infty} |f(t)|^2 dt$$

Енергія E_s дискретизованої функції $f(t)$ задається наступною формулою:

$$E_s = \sum_{-\infty}^{\infty} |f(t)|^2$$

Локальність за частотою означає, що перетворення Фур'є вейвлету локалізоване.

Функція $\psi(x)$ є вейвлетом, якщо вона задовольняє двом умовам:

$$1. \int_{-\infty}^{\infty} \psi(x) dx = 0$$

$$2. \int_{-\infty}^{\infty} \frac{|\widehat{\psi}(w)|^2}{|w|} dw \equiv C_{\psi} < \infty$$

Друга умова є необхідною щоб можна було відтворити функцію з декомпозиції вейвлетів. [6]

Константу C_{ψ} називають прийнятною вейвлет константою і математично вона визначається як:

$$C_{\psi} = \int_{-\infty}^{\infty} \frac{\overline{\widehat{\psi}(w)} \widehat{\psi}(w)}{|w|} dw$$

Якщо значення цієї константи лежить в діапазоні від нуля до нескінченності, то вейвлет $\psi(w)$ називають прийнятним [6]. При цьому функція повинна проходити через точку (0,0).

Широко застосовані вейвлети:

1. Вейвлет Рікера

$$\psi(t) = \frac{2}{\sqrt{3\sigma\pi}^{1/4}} \left(1 - \left(\frac{t}{\sigma}\right)^2\right) e^{-\frac{t^2}{2\sigma^2}}$$

Цей вейвлет є другою похідною функції Гауса, $f(x) = \exp(-x^2)$ і він широко використовується в обчислювальній електродинаміці [23].

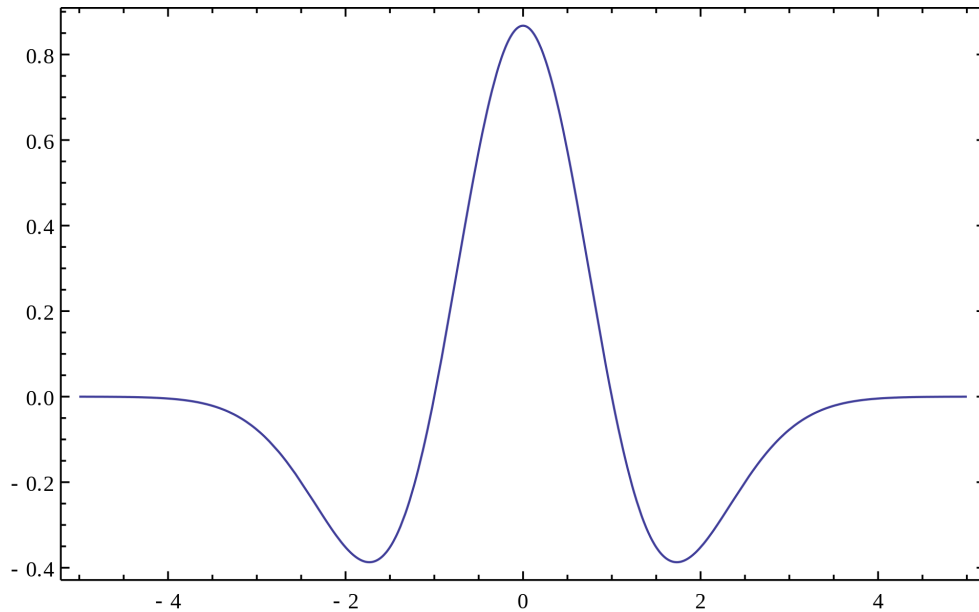


Рисунок 2.6. Вейвлет Рікера

2. Вейвлет Хаара

Це один із перших вейвлетів, що базується на ортогональній системі функцій. Цей вейвлет є ортогональним і добре локалізований у просторі, проте не є гладким. Перевага цього вейвлету полягає в тому що він швидко обчислюється, проте оскільки він не гладкий то менш корисний для аналізу неперевних функцій [24].

$$\psi(x) = \begin{cases} 1, & 0 \leq x < 1/2 \\ -1, & 1/2 \leq x < 1 \\ 0, & x \notin [0, 1) \end{cases}$$

На практиці цей вейвлет застосовується для компресії сигналів і зображень з плавними переходами.

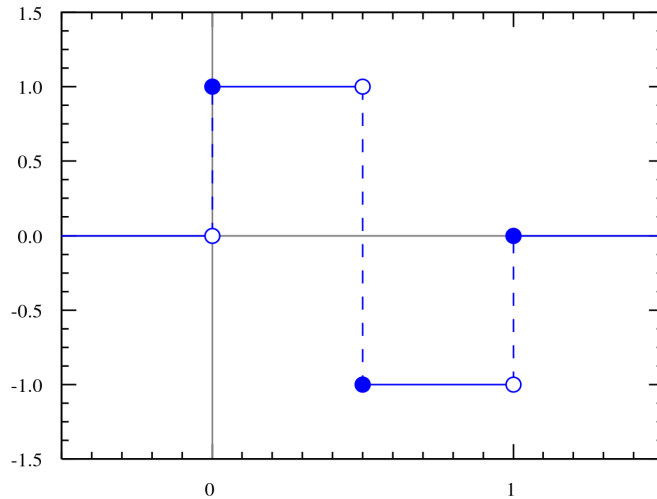


Рисунок 2.7. Вейвлет Хаара

3. Вейвлети Добеші

Вейвлети Добеші це сімейство ортогональних вейвлетів, що характеризується максимальною чисельністю зникаючих моментів для деякого носія функції. При N зникаючих моментів, кількістю коефіцієнтів рівна $2N$. Застосовують дві схеми іменування, DN з використанням кількості відводів та dbA з кількістю зникаючих моментів. Відповідно D4 і db2 є еквівалентними вейвлетами [25].

Серед 2^{N-1} рішень алгебраїчних рівнянь для умови ортогональності та моменту, обирають рішення при якому масштабуючий фільтр має екстримальну фазу.

Вони широко застосовуються для вирішення багатьох задач, наприклад, самоподібні властивості сигналу, сигнальні розриви, фрактальні проблеми тощо.

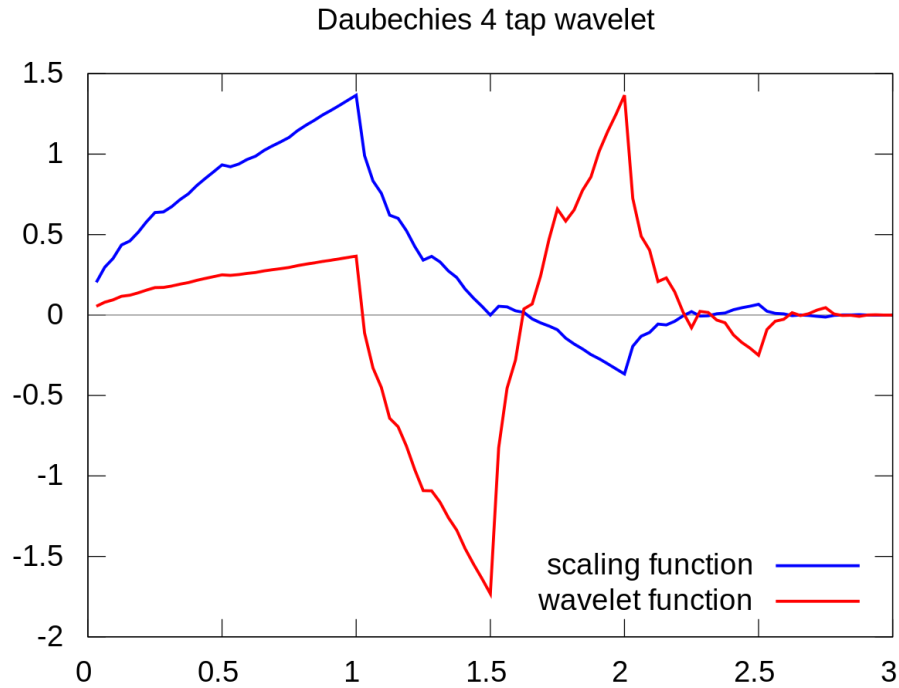


Рисунок 2.8. Вейвлет Добеші з 4 відводами

4. Вейвлет Мейера

Вейвлет Мейера це ортогональний вейвлет, що застосовується в фракталах випадкових полів, адаптивних фільтрах тощо. Функція диференційна в будь якій точці та визначена в частотній області відносно функції $\nu(x)$.

$$\Psi(\omega) := \begin{cases} \frac{1}{\sqrt{2\pi}} \sin\left(\frac{\pi}{2} \nu\left(\frac{3|\omega|}{2\pi} - 1\right)\right) e^{j\omega/2} & \text{if } 2\pi/3 < |\omega| < 4\pi/3, \\ \frac{1}{\sqrt{2\pi}} \cos\left(\frac{\pi}{2} \nu\left(\frac{3|\omega|}{4\pi} - 1\right)\right) e^{j\omega/2} & \text{if } 4\pi/3 < |\omega| < 8\pi/3, \\ 0 & \text{otherwise,} \end{cases}$$

$$\nu(x) := \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } 0 < x < 1, \\ 1 & \text{if } x > 1. \end{cases}$$

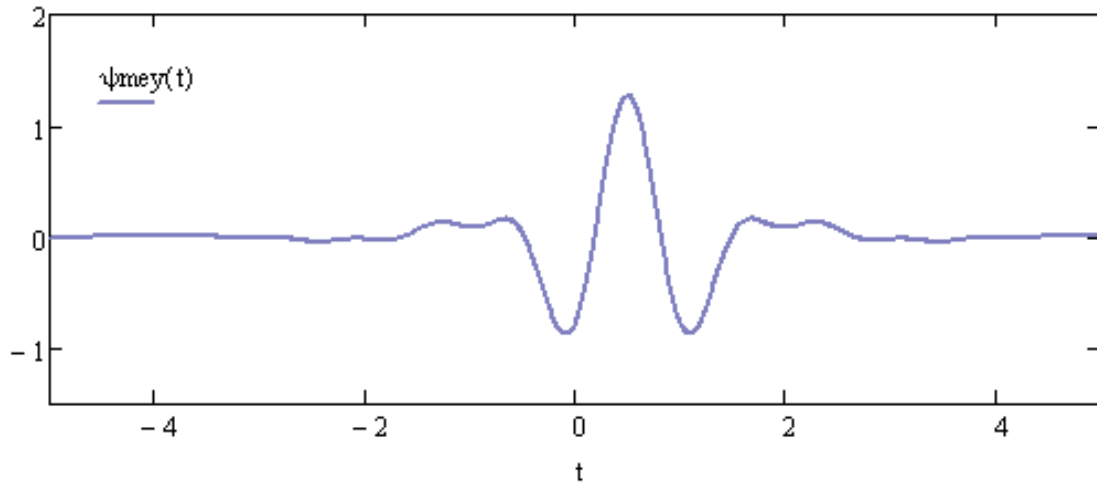


Рисунок 2.9. Вейвлет Майера

5. Вейвлет Морлета

Вейвлет Морлета це вейвлет що складається з комплексних експонент помножених на вікно Гауса.

$$\psi_{\sigma}(t) = c_{\sigma} \pi^{-1/4} e^{-0.5 t^2} (e^{i\sigma t} - k_{\sigma})$$

$$\text{де } k_{\sigma} = e^{-0.5 \sigma^2}, c_{\sigma} = (1 + e^{-\sigma^2} - 2e^{-0.75 \sigma^2})^{-0.5}$$

Цей вейвлет застосовується в музиці для точного визначення частоти короткочасного коливаючого сигналу. Також цей вейвлет застосовується для визначення ритму серцебиття в ЕКГ.

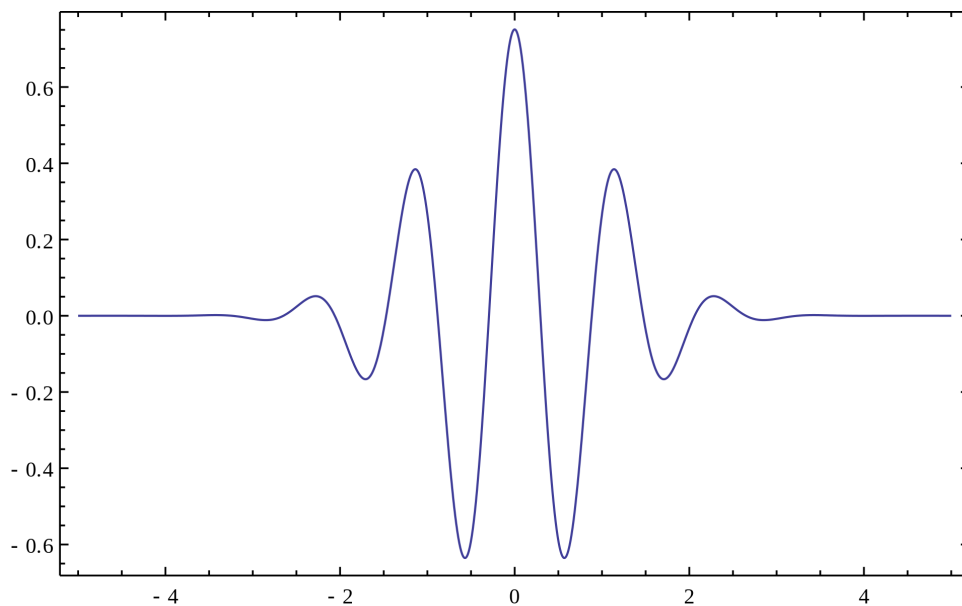


Рисунок 2.10. Вейвлет Морлета

Сімейство вейвлетів це множина функцій, одержаних шляхом зсуву і розширення (або звуження) початкового вейвлету.

Математично це задається наступним виразом:

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right)$$

де b - це зсув або центр $\Psi_{a,b}$ і a - це масштаб.

Коефіцієнт a звужує або розширює сигнал. У випадку коли значення цього коефіцієнту є малим, то вейвлет дасть детальну інформацію про функцію на обмеженому інтервалі. Коли значення цього коефіцієнту є достатньо великим, то вейвлет дасть інформацію про функцію на всій області її визначення.

Неперервне вейвлет перетворення функції $f(x)$ задається наступним математичним виразом:

$$Wf(a,b) = \int_{-\infty}^{\infty} f(x) \Psi_{a,b}(x) dx$$

Обернене неперервне вейвлет перетворення задається наступним виразом:

$$f(x) = \frac{1}{C_{\Psi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{|a|^{3/2}} Wf(a,b) \Psi_{a,b}(x) da db$$

Найпоширенішим застосуванням неперервного вейвлет перетворення є стискання зображень. У порівнянні зі звичними методами стискання зображень це перетворення стискає зображення з найменшою втратою якості вхідного зображення. Також його застосовують для вирішення диференційних рівнянь, аналізу ЕКГ, проектування фільтрів тощо.

Оскільки обрахунок цих інтегралів є неможливим на комп'ютері, то існує видозміна неперервного вейвлет перетворення - дискретне вейвлет перетворення, що легко реалізується на комп'ютері.

Основна відмінність між неперервним та дискретним вейвлет перетвореннями полягає в тому що неперервне вейвлет перетворює рахується для усіх комбінацій коефіцієнтів a і b , кількість комбінацій при цьому прямує до нескінченності. В той

час як дискретне вейвлет перетворення рахується для скінченної множини коефіцієнтів a і b .

Ефективний алгоритм для обчислення дискретного вейвлет перетворення називається швидким вейвлет перетворенням і його асимптотична складність рівна $O(n)$.

Як зазначено раніше складність алгоритму швидкого перетворення Фур'є рівна $O(n \log(n))$, отже швидкість цього алгоритму є кращою, що є безумовно його перевагою. Алгоритм є рекурсивним і приймає на вхід сигнал $s^{(j)}$.

$$s_n^{(k)} = \frac{1}{2} \sum_{m=-N}^N a_m s_{2n+m}^{(k+1)}$$

$$d_n^{(k)} = \frac{1}{2} \sum_{m=-N}^N b_m s_{2n+m}^{(k+1)}$$

для $k = J - 1, J - 2, \dots, M$ і для всіх цілих n .

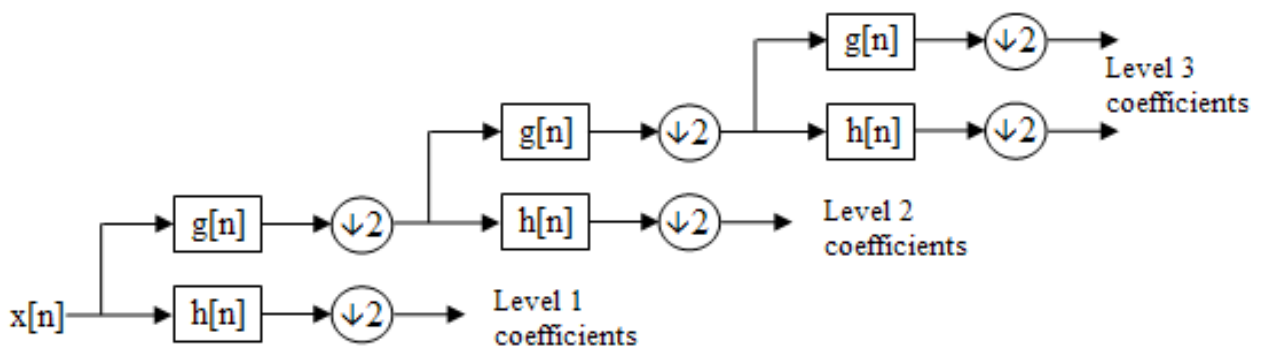


Рисунок 2.11. Ілюстрація роботи алгоритму швидкого вейвлет перетворення

На кожному кроці алгоритму, відбувається декомпозиція сигналу на низькі та високі частоти. Відповідно розмір вхідного вектору повинен бути кратним 2^n , де n - це кількість рівнів.

Наприклад, алгоритм 3 рівня розіб'є сигнал, що складається з 32 значень та частота якого лежить в діапазоні між 0 та f_n Гц, на наступні частоти:

Таблиця 2.1

Рівень	Діапазон частот	Кількість значень
3	Від 0 до $f_n / 8$ Гц	4
2	Від $f_n / 4$ до $f_n / 2$ Гц	8
1	Від $f_n / 2$ до f_n Гц	16

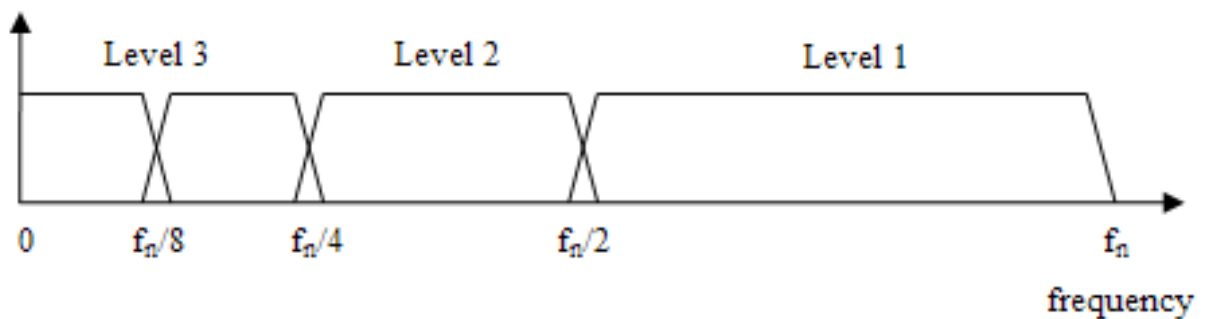


Рисунок 2.12. Ілюстрація частотного розподілу дискретного вейвлет перетворення

Алгоритм оберненого вейвлет перетворення приймає на вхід вектор $s^{(M)}$ і вектори $d^{(k)}$, $k = M, \dots, J - 1$.

$$s_n^{(k+1)} = \sum_{k=-N}^N a_k s_{2n-k}^{(k)} + \sum_{k=-N}^N b_k d_{2n-k}^{(k)}$$

для $k = J - 1, J - 2, \dots, M$ і для всіх цілих n .

3. Цифрові фільтри

Цифровий фільтр - це фільтр, що здійснює обробку цифрового сигналу з метою придушення або відокремлення певних частотних складових сигналу. У порівнянні з аналоговими фільтрами, цифрові фільтри мають наступні переваги:

1. Передавальна функція не залежить від температурних дрейф характеристик електронних елементів, тому фільтри є більш стабільними

2. Фільтри є програмованими, конфігурацію фільтру можна змінити без зміни електричної схеми
3. Фільтри легко проєктуються, реалізуються і тестуються на звичайному комп'ютері

Недоліком цифрових фільтрів є важкість роботи у реальному часі, оскільки обчислення мають завершитись протягом періоду дискретизації.

Порядок фільтру це кількість необхідних вхідних значень для обрахунку вихідного значення.

На приклад:

Фільтр нульового порядку: $y_n = a_0 x_n$

Фільтр першого порядку: $y_n = a_0 x_n + a_1 x_{n-1}$

Фільтр другого порядку: $y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2}$

Коефіцієнти a_0, a_1, \dots, a_n - коефіцієнти цифрового фільтру.

Цифрові фільтри поділяються на два види - фільтри із скінченною імпульсною характеристикою та фільтри з нескінченною імпульсною характеристикою.

Характерною особливістю фільтрів зі скінченною імпульсною характеристикою (англ. FIR-filter) є обмеженість за часом її імпульсної характеристики. Цей фільтр називають також нерекурсивним, оскільки відсутній зворотній зв'язок. Перевагою цього фільтру є те що він стійкий і його фазо-частотна характеристика може бути лінійною.

У випадку фільтру з нескінченною імпульсною характеристикою (англ. IIR-filter), розрахунок вихідної величини залежить від попередніх значень сигналу і від N попередніх значень вихідного сигналу. Тобто утворює зворотній зв'язок. Перевагою фільтру з нескінченною імпульсною характеристикою у порівнянні з фільтром з скінченною імпульсною характеристикою є краща швидкодія, проте цей

фільтр має нелінійну фазо-частотну характеристику [29]. Нелінійна фазо-частотна характеристика спотворює форму вхідного сигналу.

Ефект "коливання осі" (baseline wandering) – це одна з найпоширеніших низькочастотних завад, що виникає при русі і диханні пацієнта під час проведення вимірювання. Діапазон її частот сягає від 0.1 до 0.4 Гц. Для усунення цієї завади може бути застосований цифровий фільтр високих частот з частотою зрізу близько 0.5 Гц.

Фільтр Бесселя – цифровий фільтр з нескінченною імпульсною характеристикою, якому притаманна максимально лінійна фазо-частотна характеристика [29]. Недоліком цього фільтру є те, що перехід амплітудно-частотної характеристики [30] від дозволених частот до заборонених відбувається повільніше, ніж у інших фільтрів цього ж типу, що є не критичним у цьому випадку.

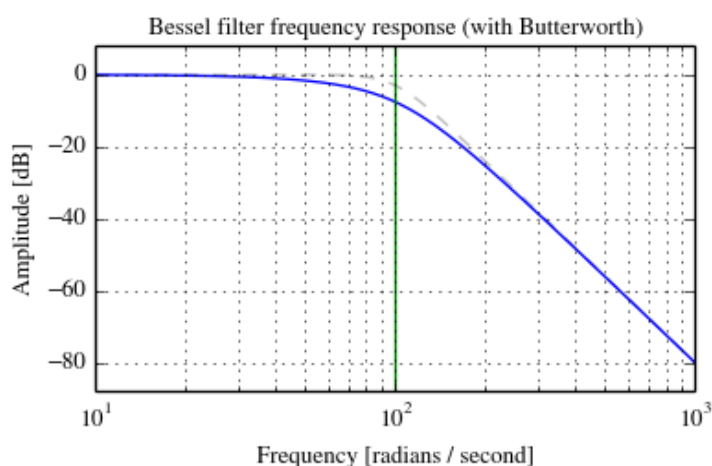


Рисунок 2.12. Амплітудно-частотна характеристика фільтру Бесселя

Передавальна функція фільтру Бесселя задається наступним виразом:

$$H(s) = \frac{\theta_n(0)}{\theta_n(s/w_0)}$$

де θ_n - обернений багаточлен Бесселя, w_0 - частота зрізу.

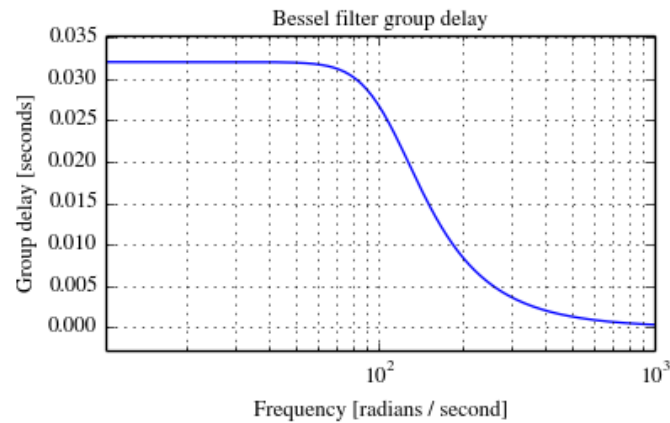


Рисунок 2.13. Фазо-частотна характеристика фільтру Бесселя

На рисунку наведена фазо-частотна характеристика фільтру Бесселя. Функція є практично лінійною біля частоти зрізу.

Фільтр Баттерворда - це цифровий фільтр з нескінченною імпульсною характеристикою, амплітудно-частотна характеристика [30] якого є максимально гладкою на частотах смуги пропускання, яка зменшується майже до нуля за межами смуги пропускання.

АЧХ фільтру Баттерворта n -го порядку може бути отримана із передаточної функції $H(s)$:

$$G^2(\omega) = |H(j\omega)|^2 = \frac{G_0^2}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

де n - порядок фільтру, ω_c - частота зрізу, G_0 - коефіцієнт підсилення по постійній складовій.

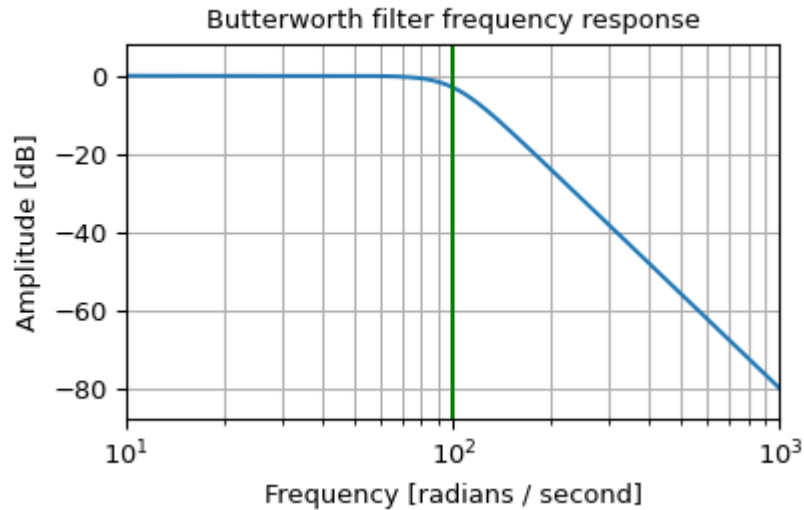


Рисунок 2.14. Амплітудно-частотна характеристика фільтру Баттерворда

На рисунку наведена АЧХ фільтру Баттерворда, функція лінійно спадає після частоти зрізу. У порівнянні з фільтром Бесселя, фазо-частотна характеристика [29] не є лінійною.

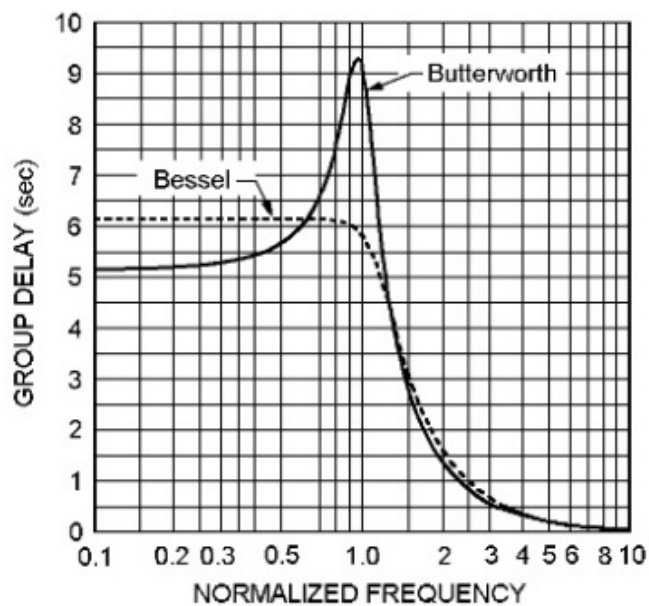


Рисунок 2.15. Порівняння фазо-частотної характеристики фільтру Бесселя і фільтру Баттерворта

На рисунку зображено фазо-частотну характеристику фільтру Бесселя та фільтру Баттерворта. Графік фазо-частотної характеристики фільтру Бесселя є більш лінійним на частоті зрізу.

Фільтр Чебишова - це цифровий фільтр з нескінченною імпульсною характеристикою, амплітудно-частотна характеристика [30] якого має крутий спад та суттєві пульсації на частотах смуг пропускання та придушення. Цей фільтр застосовується коли потрібно забезпечити хороше придушення частот в певному діапазоні, при цьому гладкість амплітудно-частотної характеристики [30] на частотах смуг придушення та пропускання не є важливою.

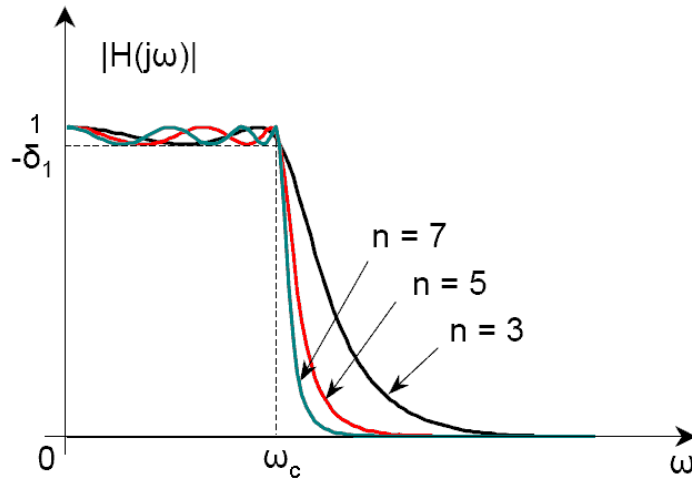


Рисунок 2.16. Амплітудно-частотні характеристики фільтру Чебишова

На рисунку зображено графіки амплітудно-частотної характеристики фільтру Чебишова в залежності від порядку n . Чим порядок цього фільтру вищий, тим спад АЧХ біля частоти зрізу ω_c є більш крутим.

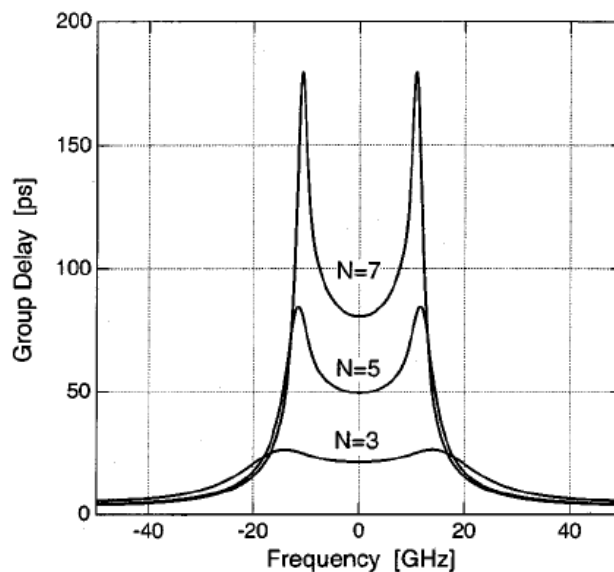


Рисунок 2.17. Фазо-частотні характеристики фільтру Чебишова

На рисунку зображено графіки фазо-частотних характеристик фільтру Чебишова в залежності від порядку n .

Передавальна функція фільтру Чебишова n -го порядку задається виразом:

$$G_n(w) = |H_n(jw)| = \frac{1}{\sqrt{1 + \epsilon^2 T_n^2\left(\frac{w}{w_0}\right)}}$$

де ϵ - показник пульсацій, w_0 - частота зрізу, $T_n(x)$ - багаточлен Чебишова n -го порядку.

Еліптичний фільтр - це цифровий фільтр з нескінченною імпульсною характеристикою, амплітудно-частотна характеристика якого має крутий спад та суттєві пульсації на частотах смуг пропускання та придушення. Величина кожної з них незалежна одна від одної.

Іншою особливістю цього фільтру є те що за рахунок більшого спаду його амплітудно-частотної характеристики, можна досягти кращого розділення частот.

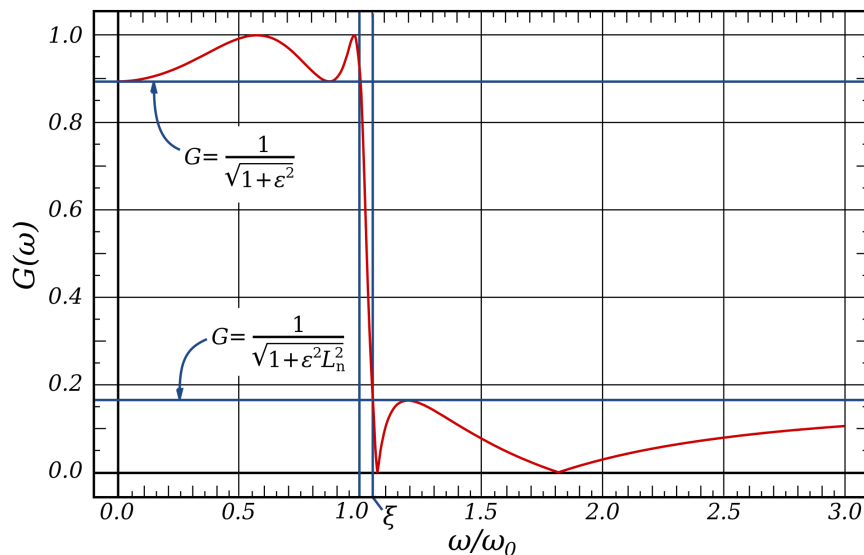


Рисунок 2.18. Амплітудно-частотна характеристика еліптичного фільтру

В смузі пропускання еліптична функція змінює значення від нуля до одиниці.

АЧХ в смузі пропускання, таким чином, змінює значення з 1 до $1 / \sqrt{1 + \epsilon^2}$.

В смузі придушення функція змінює значення від 0 до $1 / \sqrt{1 + \epsilon^2 L_n^2}$.

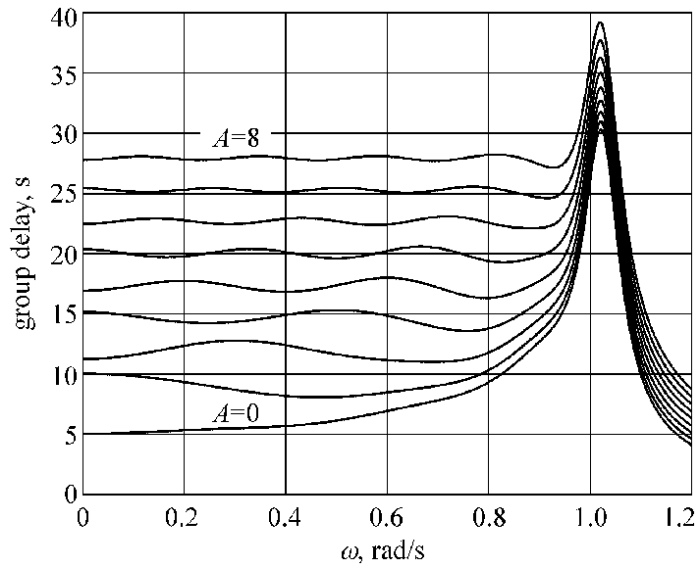


Рисунок 2.19. Фазо-частотні характеристики еліптичного фільтру

На рисунку зображено фазо-частотні характеристики еліптичного фільтру, в залежності від порядку A .

Передавальна функція еліптичного фільтру задається виразом:

$$G_n(\omega) = \frac{1}{\sqrt{1 + \epsilon^2 R_n^2(\xi, \omega/\omega_0)}}$$

де R_n - раціональна еліптична функція n -го порядку, ω_0 - частота зрізу, ϵ - показник пульсацій, ξ - показник селективності.

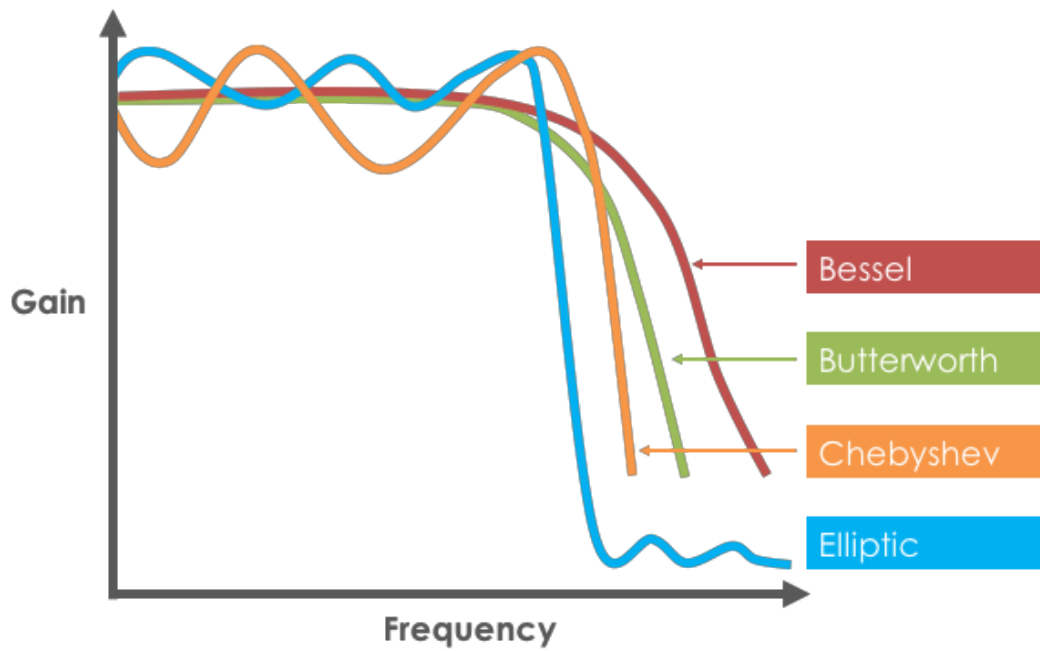


Рисунок 2.20. Графік амплітудно-частотних характеристик фільтрів Бесселя, Баттерворта, Чебишева та еліптичного фільтру [28]

Ефективність фільтрів в цьому випадку можна оцінити за двома параметрами:

1. В ідеалі цифровий фільтр повинен придушити частоти менші за 0.5 Гц, але амплітуда частот більших за 0.5 Гц повинна залишитись незмінною.

Обрахувати цей параметр можна наступним чином:

1. Здійснити перетворення Фур'є сигналу без шуму $f(w)$ та після застосованого фільтру $g(w)$
2. Обрахувати

$$\sum_{n=0.5}^{f_d} |f(n) - g(n)| \quad (2.1)$$

де f_d - частота дискретизації. Чим значення менше тим фільтр більш ефективний

2. Амплітуда частот в межах від 0 до 0.5 Гц повинна зменшитись. Чим середнє арифметичне значення амплітуд частот в цьому діапазоні менше тим фільтр більш ефективний.

$$\frac{\sum_{n=0.5}^{f_d} f(n)}{f_d - 0.5} \quad (2.2)$$

де f_d - частота дискретизації. Чим значення менше тим фільтр більш ефективний

РОЗДІЛ 3 РЕАЛІЗАЦІЯ АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ АНАЛІЗУ ЕЛЕКТРИЧНИХ СИГНАЛІВ СЕРЦЯ

3.1. Структурна схема пристрою

Структурна схема наводить головні функціональні компоненти пристрою та зв'язок між ними.

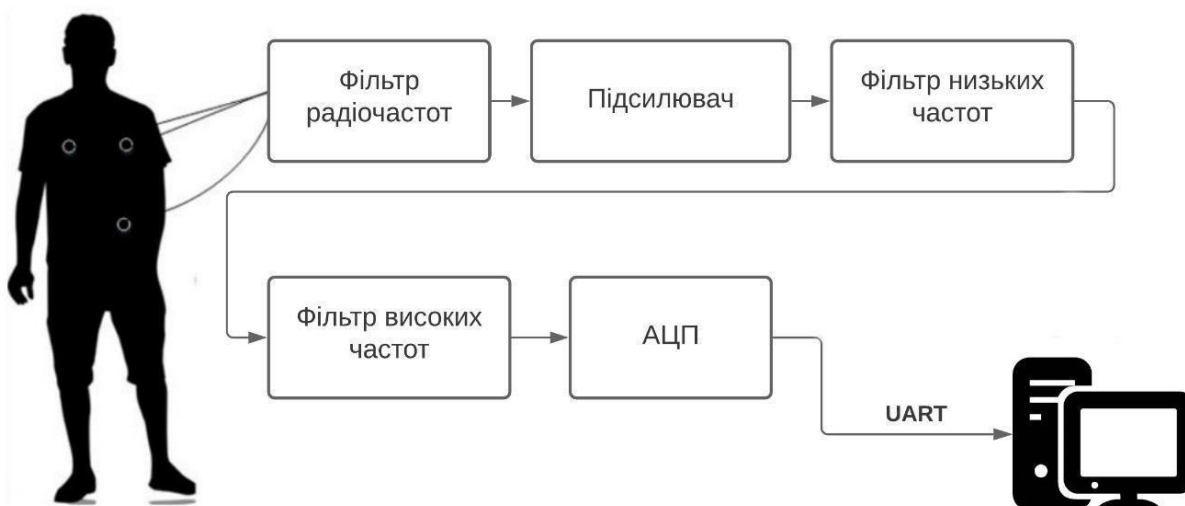


Рисунок 3.1. Структурна схема пристрою

Над сигналом, що надходить з електродів, прикладених до тіла людини, застосовується радіочастотний фільтр для усунення радіочастот. Надалі сигнал підсилюється, оскільки рівень сигналу, що надходить з тіла людини є дуже низьким. Застосовуються фільтри низьких і високих частот для виділення корисного сигналу з-поміж шуму. Сигнал проходить через аналого-цифровий перетворювач і після цього він передається по UART протоколу на комп'ютер.

3.2. Аналіз аналогового інтерфейсу AD8232 та вбудованих у ньому фільтрів та підсилювачів

Оскільки рівень сигнал, що надходить з тіла людини, є дуже низьким, то потрібна схема що підсилює цей сигнал. В якості такої схеми я застосував аналоговий інтерфейс AD8232 – розробка компанії Analog Devices.

AD8232 – інтегральний блок перетворення, призначений для біологічних сигналів. Основними його компонентами є фільтр радіочастот та інструментальний та операційний підсилювачі. Додатково вбудована система швидкого відновлення схеми зі стану насичення. Незадіяні виходи операційних підсилювачів можна використати для додаткового підсилення та фільтрування аналогового сигналу.

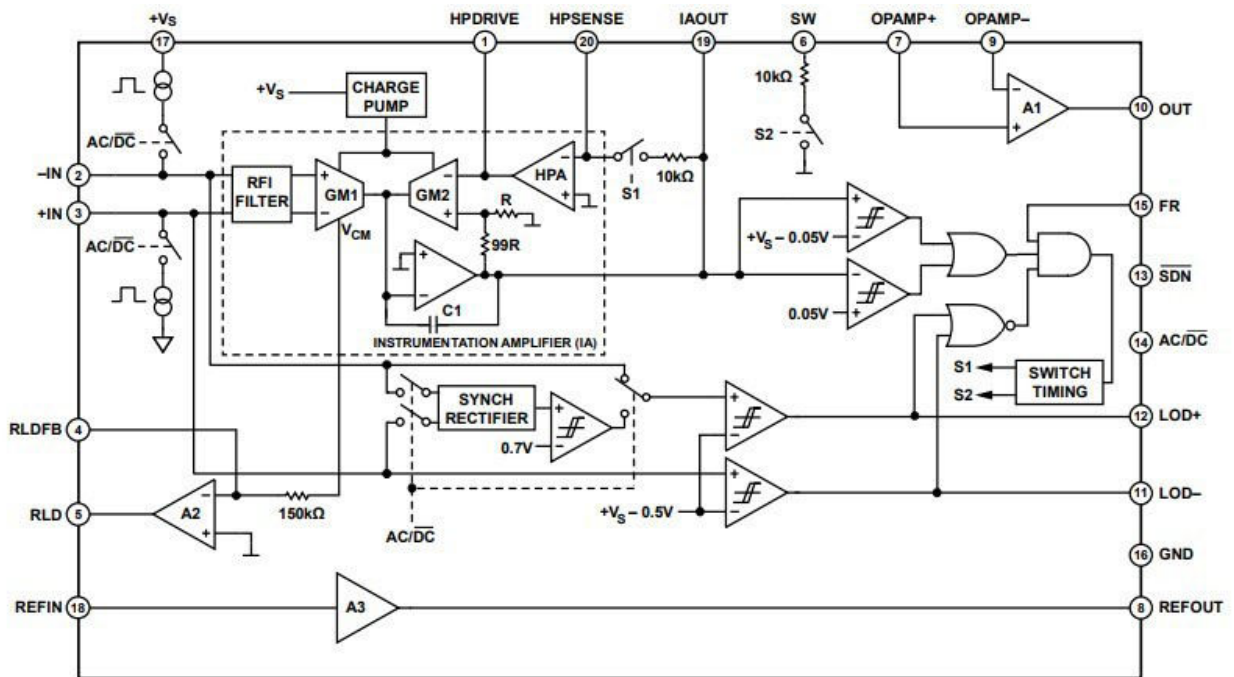


Рисунок 3.2. Функціональна діаграма AD8232 [26]

Для мінімізації впливу радіочастотного випаромінювання AD8232 на двох входах вхідного інструментального підсилювача містить конденсатор ємністю близько 15 пФ і резистор опором 10 кОм. Таким чином формується RC-фільтр низьких частот. Частота зрізу при цьому є рівною:

$$f_c = \frac{1}{2\pi RC} \approx 1.05 \text{ МГц}$$

Варто зазначити, що частота зрізу за рахунок додаткового зовнішнього опору є нижчою, відповідно вплив радіочастотного випромінювання ще більш мінімізований.

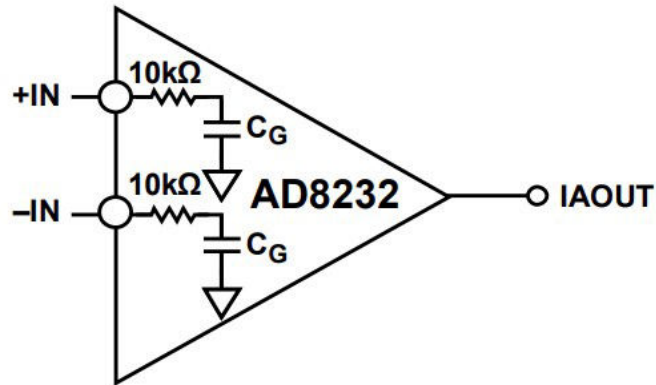


Рисунок 3.3. Радіочастотний фільтр у AD8232 [26]

Вхідний інструментальний підсилювач використовується в якості фільтру та підсилювача. Коефіцієнт його підсилення є константою.

$$K_1 = 100 \frac{\text{В}}{\text{В}}$$

Для того щоб цей підсилювач фільтрував надто малі значення напруги на вході утворено RC-коло між виводами HPSENSE, HPDRIVE і виходом інструментального підсилювача [26].

Це наведено на рисунку нижче.

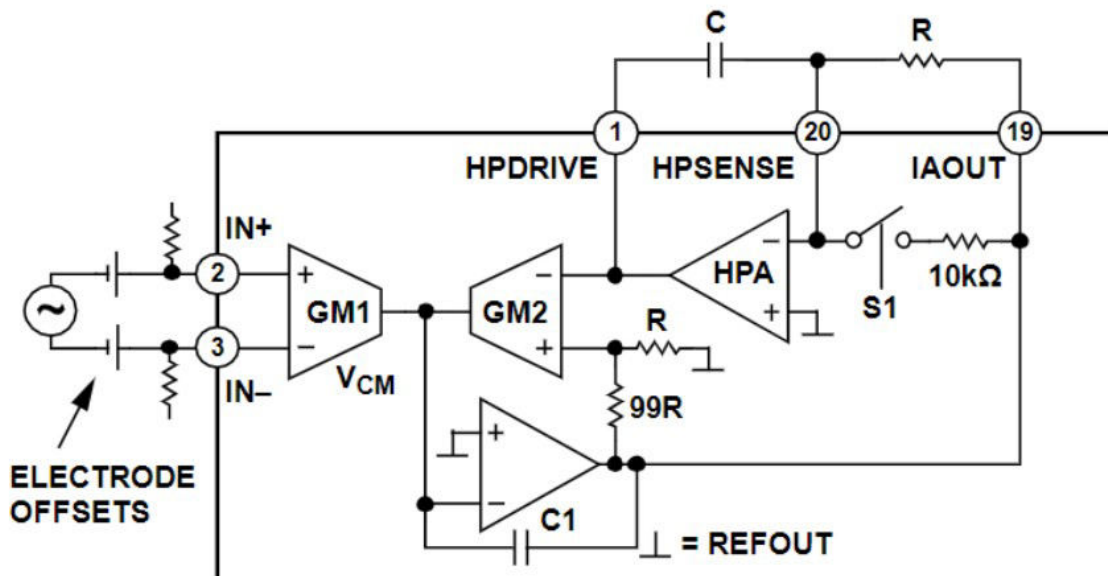


Рисунок 3.4. Схема фільтру високих частот першого порядку [26]

RC-коло реалізує інтегратор, що фільтрує занадто малі значення напруги та підтримує коефіцієнт підсилення сталим. Також він мінімізує ефект повільно наростаючих сигналів. Частота зрізу цього фільтру задається наступною формулою.

$$f_c = \frac{K_1}{2\pi RC} = \frac{100}{2\pi RC}$$

Як раніше зазначалось, мікросхема дозволяє застосувати незадіяні виходи операційних підсилювачів для додаткового підсилення та фільтрування сигналу. У фільтрах вищих порядків нахил амплітудно-частотної характеристики з більш різкими фронтами. Недоліком фільтрів вищих порядків є вищий рівень фазових спотворень, які виникають за рахунок наявності реактивного опору ємності [27].

Фільтр другого порядку реалізується досить просто. Достатньо додати ще одну RC-ланку до виходу підсилювача. Приклад такої реалізації наведений нижче.

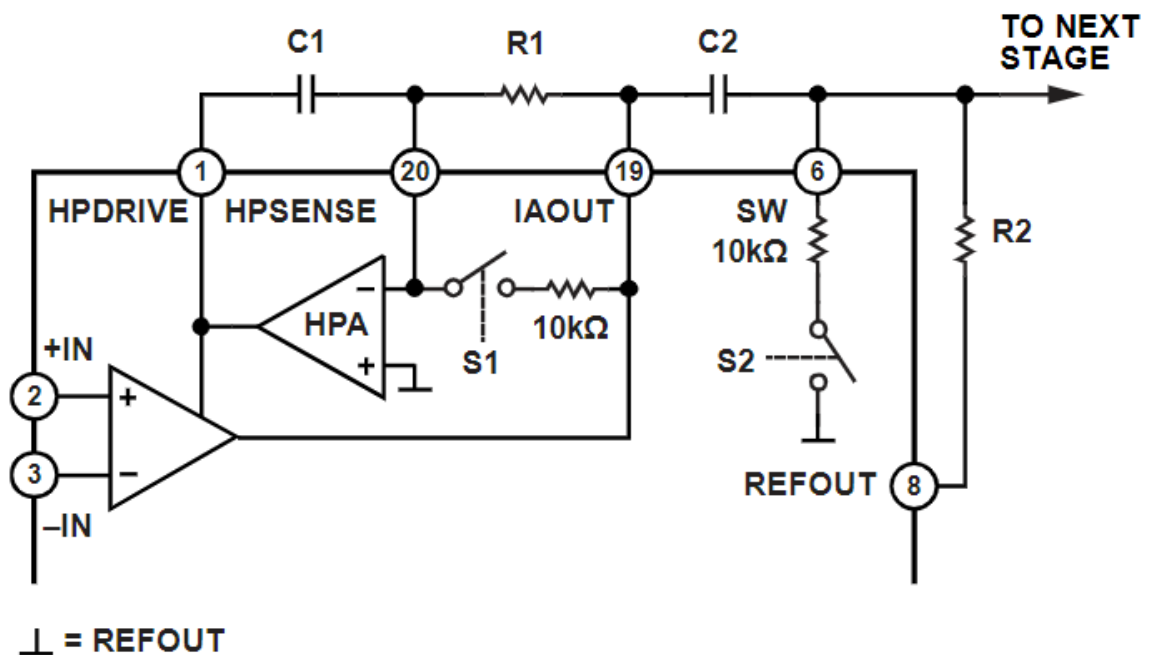


Рисунок 3.5. Реалізація фільтру високих частот другого порядку [26]

Незадіяні виходи операційного підсилювача застосовані для реалізації фільтру Саллена-Кі – активний електронний фільтр низьких частот другого порядку

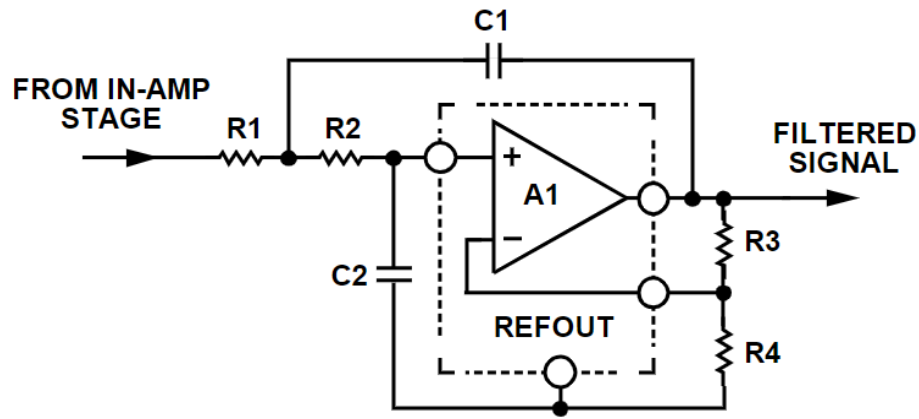


Рисунок 3.6. Схема фільтру Саллена-Кі [26]

Частота зрізу, коефіцієнт підсилення і добротність задаються наступними виразами.

$$f_c = \frac{1}{2\pi\sqrt{R_1 C_1 R_2 C_2}}$$

$$K_2 = 1 + \frac{R_3}{R_4}$$

$$Q = \frac{\sqrt{R_1 C_1 R_2 C_2}}{R_1 C_1 + R_2 C_2 + R_1 C_1 (1 - K_2)}$$

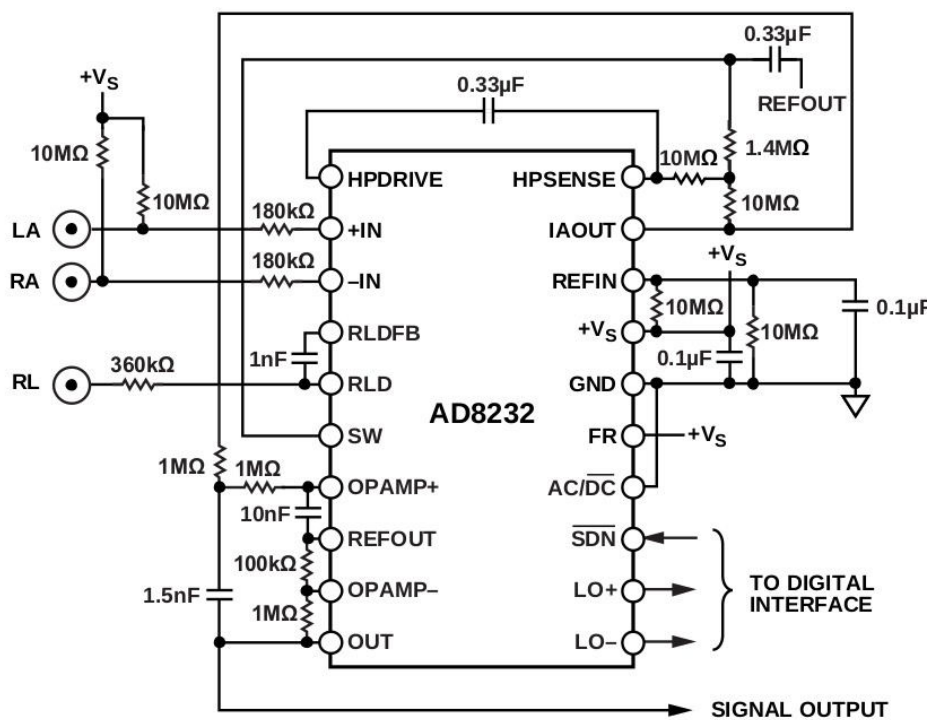


Рисунок 3.7. Схема AD8232 [26]

У проєкті AD8232 сконфігурований фільтром низьких частот другого порядку з частотою зрізу 40 Гц та фільтром високих частот другого порядку з частотою зрізу 0.5 Гц. Окрім того, сигнал підсилюється у 11 разів коли проходить через операційний підсилювач [26]. Тож, загальний коефіцієнт підсилення згідно з формулою для багатокаскадного підсилювача:

$$M = K_1 \cdot K_2 = 1000 \frac{B}{B}$$

де M - загальний коефіцієнт підсилення багатокаскадного підсилювача,

K_1 - коефіцієнт підсилення першого підсилювача,

K_2 - коефіцієнт підсилення другого підсилювача

На рисунку нижче наведена амплітудно-частотна характеристика AD8232.

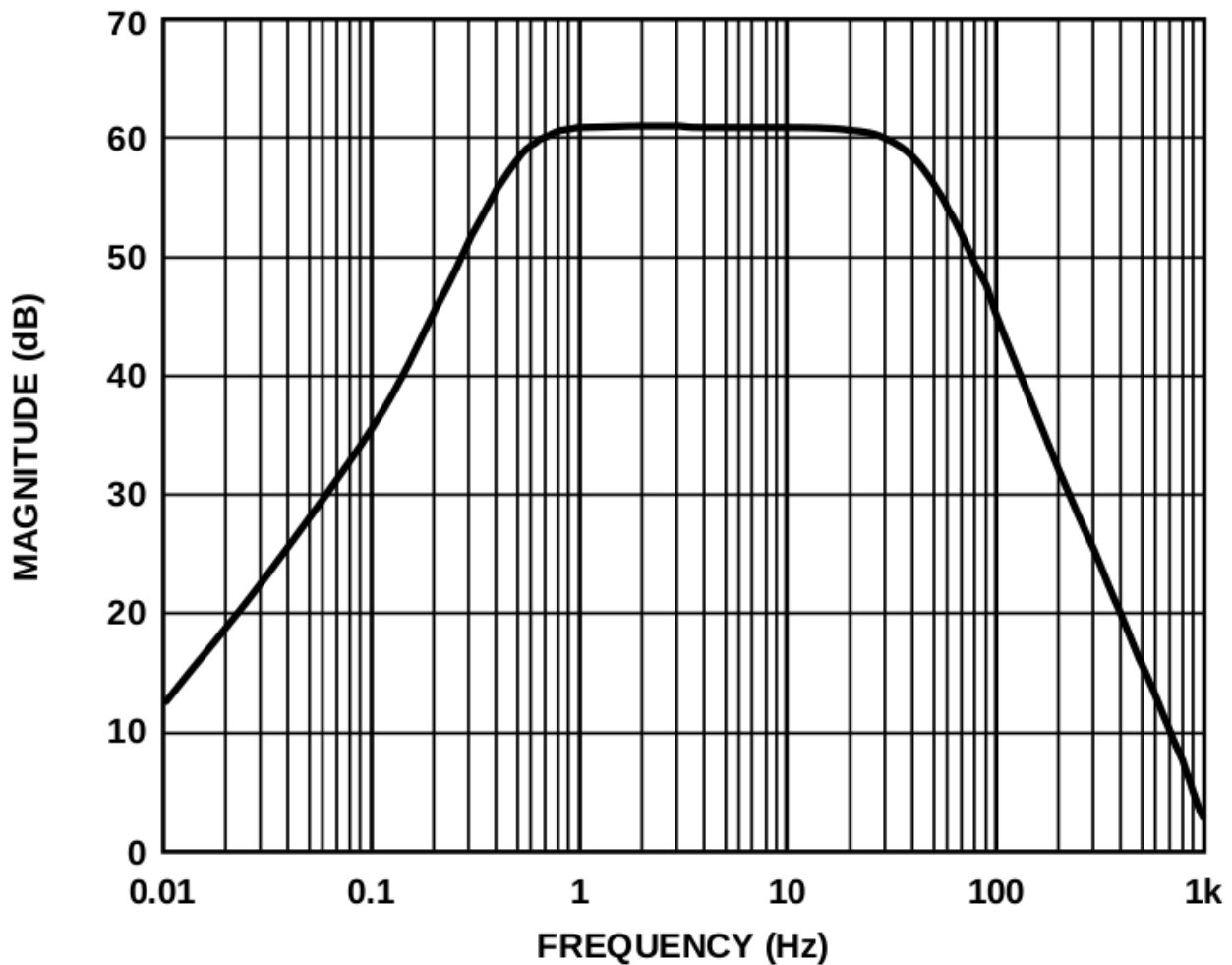


Рисунок 3.8. Амплітудно-частотна характеристика AD8232 [26]

3.3. Аналіз джерел випромінювання, що впливають на електрокардіограф

Нижче наведена таблиця діапазонів радіочастот [27].

Таблиця 3.1. Поділ діапазонів радіочастот

Номер	Назва діапазону	Діапазон довжини хвилі	Діапазон частот	Приклади застосування
1	Екстремально довгі хвилі	100000-10000 км	3-30 Гц	Комунікація з субмаринами
2	Супердовгі хвилі	10000-1000 км	30-300 Гц	Комунікація з субмаринами
3	Ультра низькі хвилі	1000-100 км	300-3000 Гц	Комунікація між субмаринами, підземний зв'язок
4	Наддовгі хвилі	100-10 км	3-30 кГц	Навігація, геофізика, безпроводні монітори частоти серцевого ритму
5	Довгі хвилі	10-1 км	30-300 кГц	RFID, аматорське радіо
6	Середні хвилі	1000-100 м	300-3000 кГц	Лавинний детектор, середньо-смугові трансляції

Найбільший вплив на ЕКГ мають лінії електропередачі.

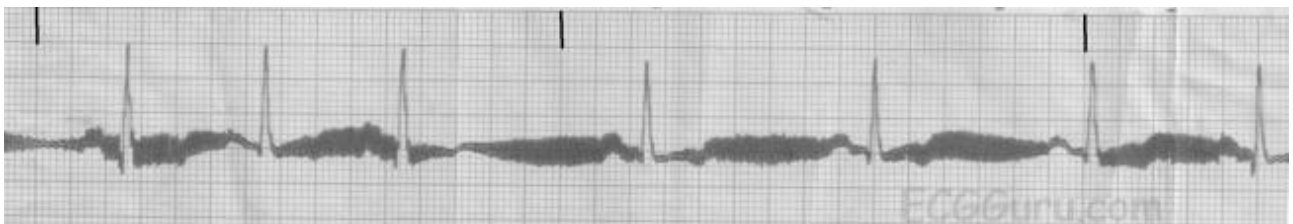


Рисунок 3.9. Вплив ліній електропередачі на ЕКГ

Частота ліній електропередач рівна 50 або 60 Гц. Оскільки аналоговий інтерфейс сконфігурований фільтром низьких частот другого порядку з частотою зрізу 40 Гц, то впливом ліній електропередачі на результат ЕКГ можна знехтувати.

3.4. Аналіз АЦП і його основних параметрів

Аналогово-цифровий перетворювач (АЦП) застосовується в системі для перетворення вхідного аналогового сигналу з датчика серцевих сигналів у цифровий, для подальшого його збереження в пам'ять мікроконтролера [15].

Основним параметром АЦП є розрядність, що характеризується кількість значень, які перетворювач може передати на вихід і вимірюється в бітах. До прикладу, АЦП, здатний видати 128 значень (0 - 127), має розрядність 7 бітів, оскільки $2^7 = 128$ [15].

Також, цей параметр може бути виражений у вольтах, у цьому випадку розрядність дорівнює напрузі, що відповідає максимально можливому вихідному коду, який ділиться на кількість вихідних значень. На приклад, якщо діапазон напруг вхідного сигналу від 0 до 10 В і розрядність АЦП 12 бітів (4096 рівнів квантування), то в результаті розрядність за напругою $= (10 - 0) / 4096 = 2.44$ мВ.

АЦП в системі має розрядність 12 бітів з максимальною напругою 2.5 В [1]. Максимальна частота дискретизації рівна 2 МГц в режимі DIG, 200 кГц в режимі RTC [3].

Тип АЦП — SAR (Successive Approximation ADC), принцип його роботи полягає в перетворенні вхідного аналогового сигналу в цифровий, використовуючи бінарний пошук по всіх рівнях квантування [16].

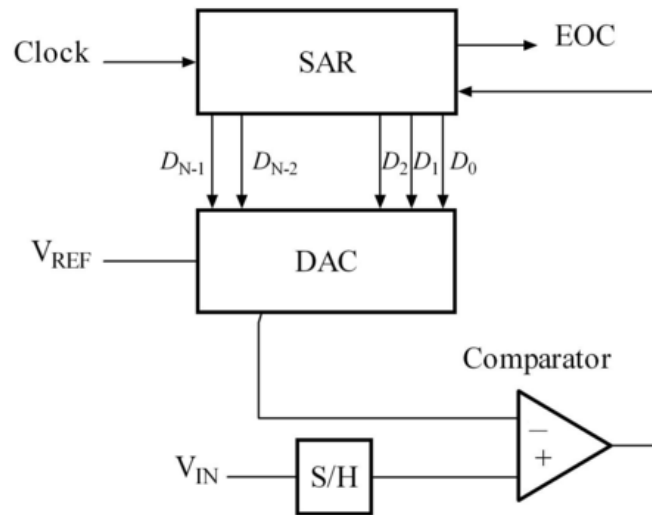


Рисунок 3.10. Функціональна схема Successive Approximation ADC

Його перевагами є висока точність, низьке споживання енергії та низький час затримки, що є суттєвим для медичних пристроїв. Недоліком цього типу АЦП є висока ціна [16].

3.5. Аналіз протоколу UART і його застосування для передачі сигналу на комп'ютер

UART (Universal Asynchronous Receiver/Transmitter) – це один з видів асинхронного приймача-передавача периферійних пристроїв, що передає дані між послідовною та паралельною формами. Переважно це окрема мікросхема чи її частина, що застосовується для з'єднання через периферійний чи комп'ютерний порт [5].

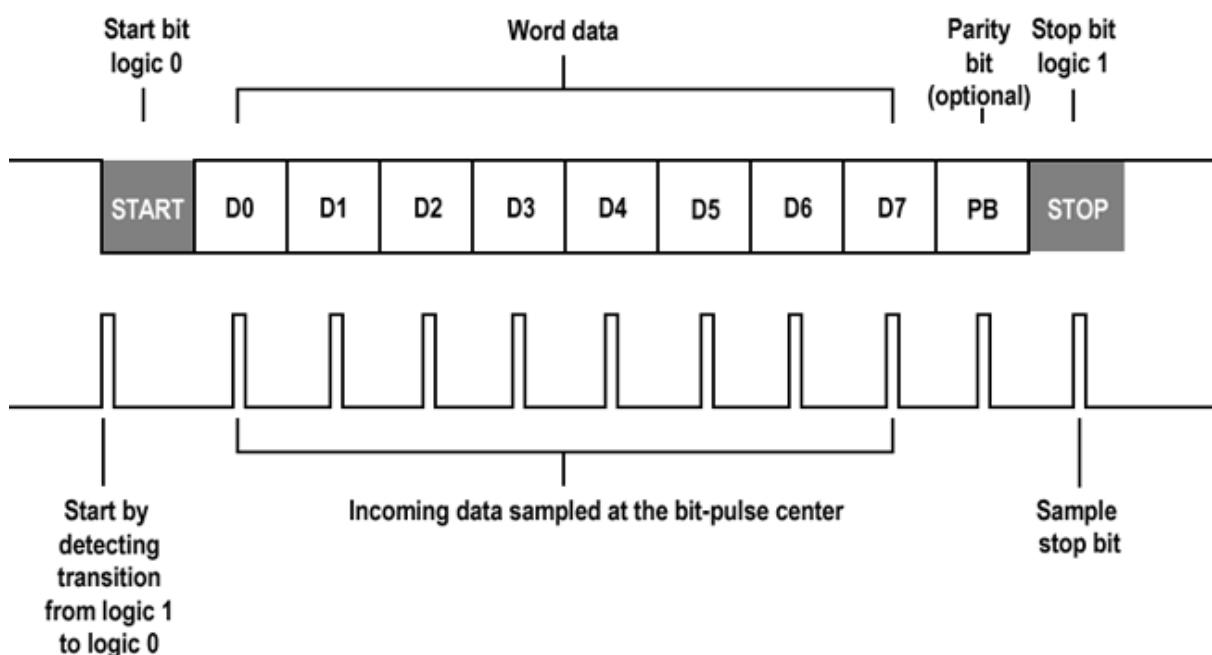


Рисунок 3.11. Процес передачі даних по інтерфейсу UART

В системі цей інтерфейс використовується разом з USB-UART адаптером для передачі сигналу з мікроконтролера на комп'ютер. Нижче проаналізований принцип роботи цього інтерфейсу для асинхронної передачі даних.

UART інтерфейс під час асинхронної передачі посилає стартовий біт, потім 5-8 бітів даних, перший — найменш значимий, потім не обов'язковий біт парності, і потім стопові біти. Стартовий біт надсилається в оберненій полярності до незайнятого стану ліній зв'язку. Стоповий біт відповідає незайнятому стану лінії та

робить паузу перед наступним сегментом даних [5]. Це називається асинхронною старт-стоповою передачею.

Біт парності при непарній перевірці перевіряє чи кількість одиниць між стартовим і стоповим бітами є непарною, при парній перевірці – чи парною відповідно. Непарна перевірка рахується більш надійною, оскільки може засвідчити що хоча б одна одиниця передалася, що дозволяє UART інтерфейсам пересинхронізуватись [5].

Система містить три UART інтерфейси типу RS232 [17] та RS485.

3.5. Принципова електрична схема системи

Принципова електрична схема – це схема що дає повне уявлення про структуру пристрою та його роботу. На рисунку 3.12 наведено принципову електричну схему пристрою для зчитування сигналу з електродів, прикріплених до тіла людини, та передачі сигналу по інтерфейсу UART. Для розробки цієї схеми я використав програму EasyEDA.

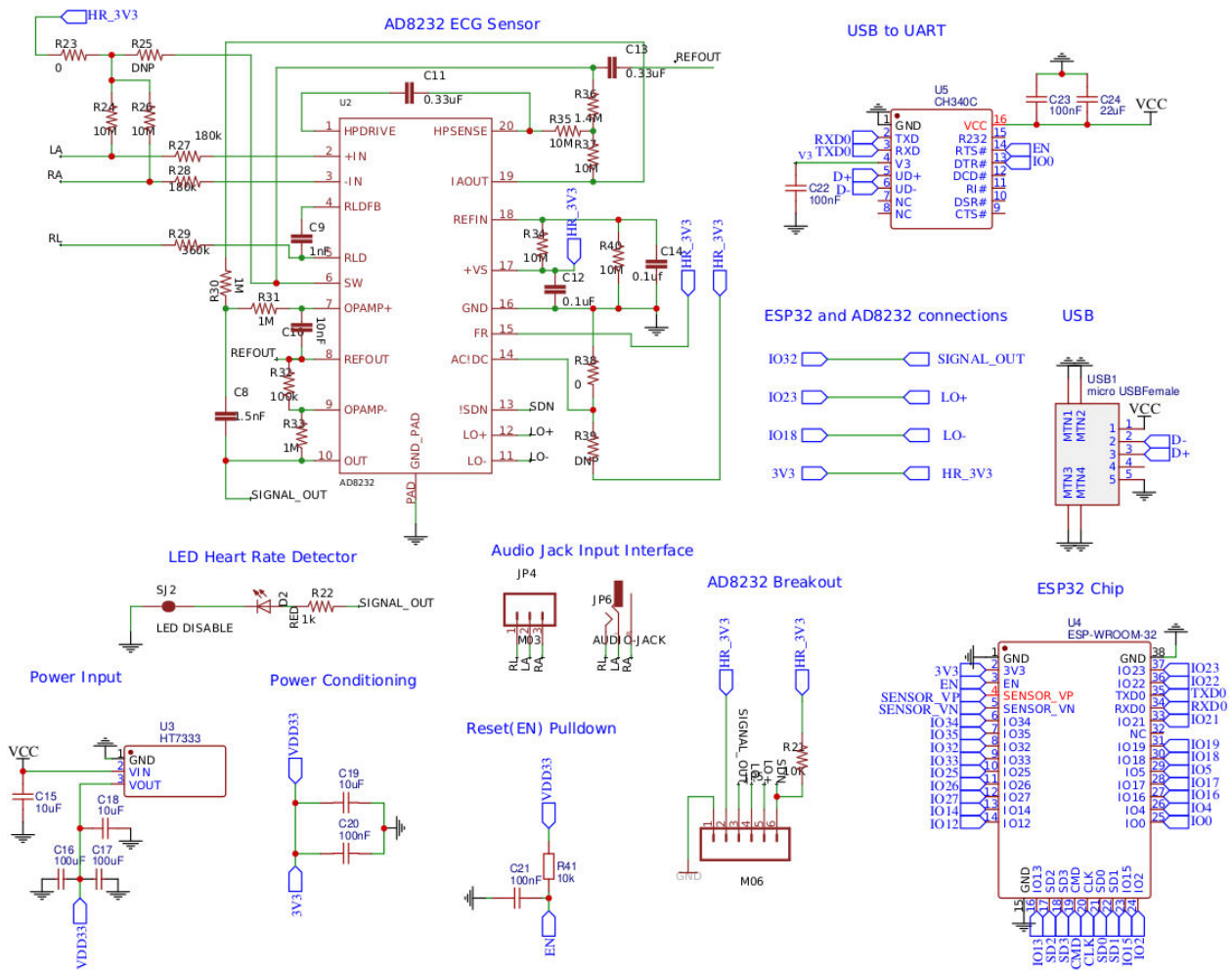


Рисунок 3.12. Принципова електрична схема пристрою

3.6. Економічне обґрунтування

Загальна вартість цього пристрою 11.77 доларів. У ціну входить плата для розробки ESP-32-WROVER, що містить систему на кристалі ESP32, USB-UART конвертор, стабілізатор живлення тощо. Ціна цієї плати - 7 доларів. Також у ціну входить аналоговий інтерфейс SparkFun AD8232, що наразі коштує 4.77 доларів. Вартістю провідників та електродів можна знехтувати. На ринку фігурують схожі пристрої починаючи з 3000 гривень. При цьому, деякі з них бракують функціоналом передачі сигналу на комп'ютер, деякі не містять вбудованої системи фільтрів тощо.

Я вважаю, що розроблена система є економічно вигідною і ретельно протестувавши, її можна використовувати не тільки у навчальних цілях, але й в комерційних.

РОЗДІЛ 4 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ АНАЛІЗУ ЕЛЕКТРИЧНИХ СИГНАЛІВ СЕРЦЯ

4.1. Розробка програми для мікроконтролера

Програма для мікроконтролера відповідає за зчитування сигналу з АЦП і за його передачу по інтерфейсу UART. Програма написана мовою програмування C з допомогою бібліотеки ESP-IDF. ESP-IDF надає зручний програмний інтерфейс для взаємодії з периферійними пристроями.

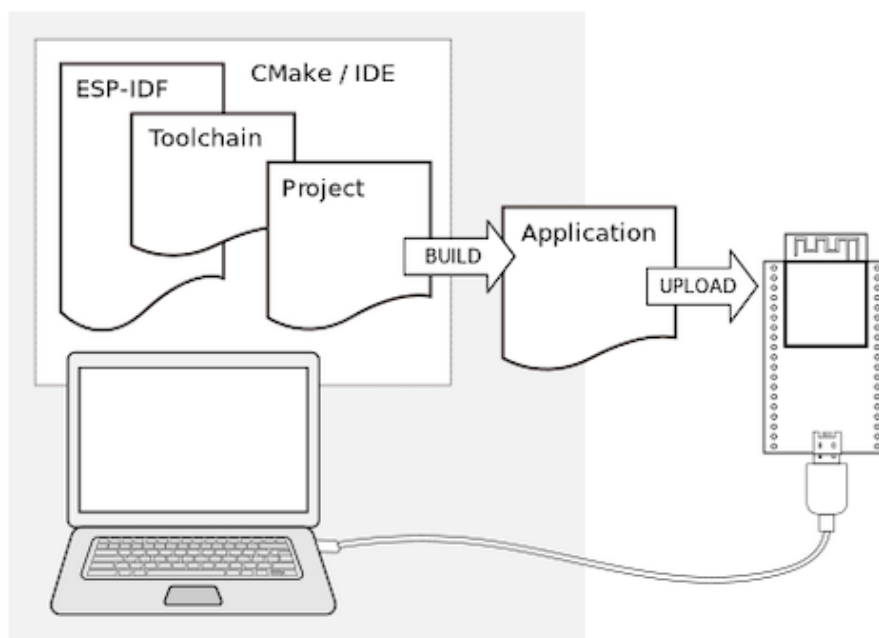


Рисунок 4.1. Процес розробки з використанням бібліотеки ESP-IDF

Система на кристалі містить два вбудованих АЦП, кожен з яких підтримує два режими зчитування:

1. Режим одноразового зчитування (Single read mode)
2. Режим неперервного зчитування (Continuous read mode)

Недоліком одноразового зчитування є низька максимальна частота дискретизації та вище навантаження на процесор, тому я обрав режим неперервного зчитування [4].

В цьому режимі за зчитування з АЦП відповідає DMA (Direct memory access) контролер [4]. Контролер зчитує з АЦП в матрицю з певною частотою дискретизації. Як-тільки рядок матриці заповнений, він робить переривання і процесор обробляє рядок матриці. Це проілюстровано на рисунку нижче.

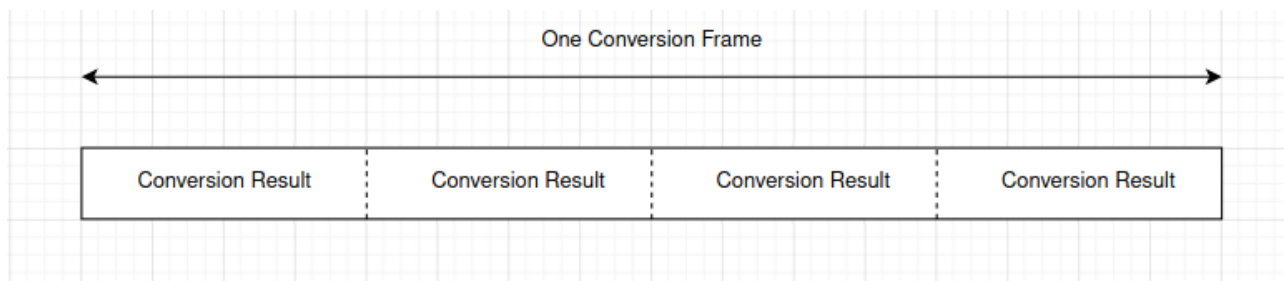


Рисунок 4.2. Зчитування з АЦП в режимі неперервного зчитування

В цьому випадку процесор передає рядок по UART інтерфейсу. Кількість колонок в матриці не має бути малою оскільки в такому випадку процесор буде часто перериватись. І також не занадто великою, тому що об'єм оперативної пам'яті є обмеженим.

Нехай:

Частота дискретизації (Гц) - f , кількість рядків у матриці $M - n$, кількість колонок у матриці $M - m$, швидкість зв'язку по каналі даних b .

Як раніше зазначено, розрядність АЦП – 12 бітів. Отже максимальне значення що АЦП може передати є $2^{12} - 1 = 4095 - 4$ байти інформації (оскільки кожен символ в кодуванні ASCII займає один байт). Окрім того, символ роздільник теж займає один байт.

Тож час для передачі рядку по UART рівний:

$$t(n) = \frac{5 \cdot 8 \cdot n}{b} = \frac{40 \cdot n}{b} \text{сек}$$

Час для зчитування з АЦП всієї матриці:

$$g(n, m) = \frac{n \cdot m}{f} \text{сек}$$

Необхідна умова

$$t(m) < g(n - 1, m) \Rightarrow \frac{40 \cdot m}{b} < \frac{(n-1) \cdot m}{f} \Rightarrow \frac{40 \cdot f}{b} < (n - 1)$$

Швидкість зв'язку по каналі даних (baud rate) в системі 115200 бод [18].

На АЦП надходить сигнал з діапазоном частот від 1 Гц до приблизно 150 Гц, отже за теоремою Котельникова мінімальна частота дискретизації повинна бути мінімум у два рази вища. Задана частота дискретизації у програмі - 1 кГц.

$$\frac{40 \cdot 1000}{115200} < n - 1 \Rightarrow n > 1.35$$

Отже матриця повинна мати хоча б два рядки.

Повний код програми для мікроконтролера наведений у додатку А.

4.2. Розробка програми для зчитування сигналу на комп'ютері

Програма для зчитування сигналу на комп'ютері написана на Java і виконує кілька паралельних процесів:

1. Процес зчитування сигналу з СОМ порту і запис значень сигналу в оперативну пам'ять
2. Процес, що зчитує отримані значення з оперативної пам'яті і відображає графік з останніх N значень
3. Процес, що виконується наприкінці завершення програми для запису отриманого сигналу в файл для подальшого аналізу

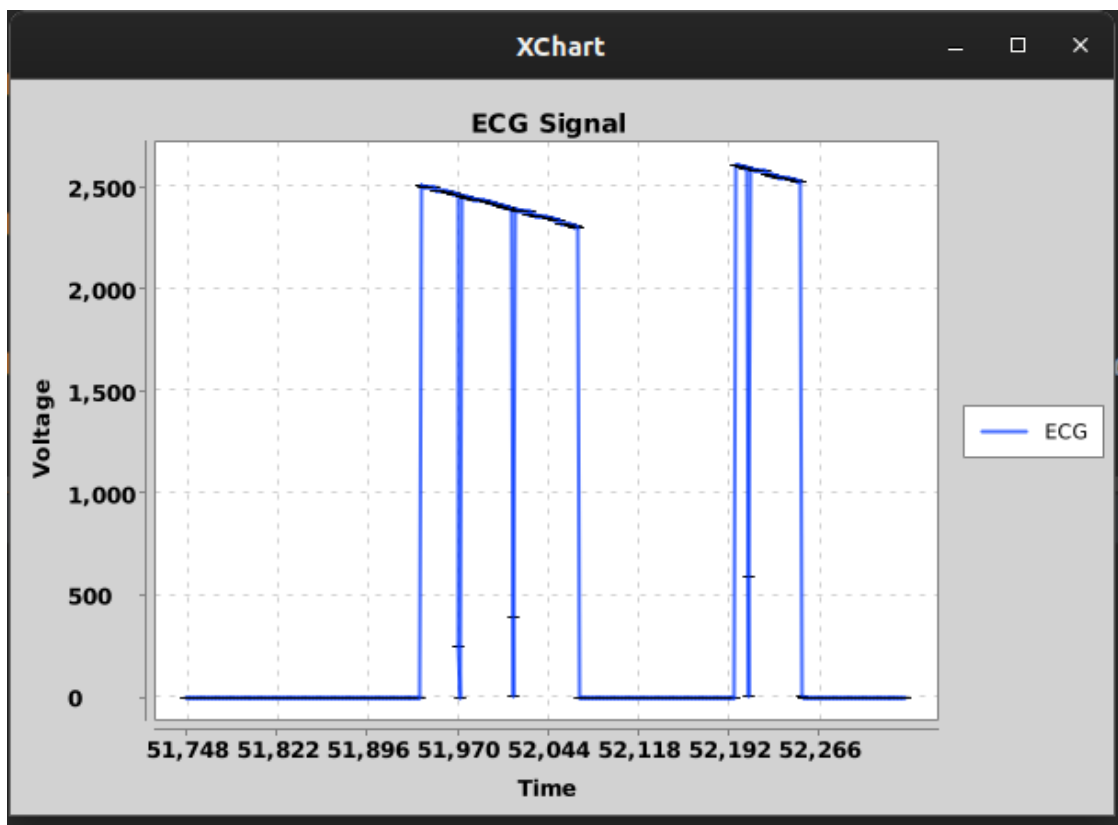


Рисунок 4.3. Скріншот програми для зчитування сигналу на комп'ютері

Для користування програмою достатньо підключити мікроконтролерну систему по USB до комп'ютера і процес зчитування та відображення сигналу автоматично розпочнеться. Після завершення програми (під час натискання на кнопку закриття програми, розташовану в верхньому правому куті) розпочинається запис сигналу в файл. Файл зберігається в home директорії на комп'ютері з назвою `ecg_{date}.txt`, де `date` - дата і час відкриття програми. Формат файлу `txt`. Кожен рядок файлу містить дискретизоване значення сигналу в діапазоні між 0 до 4095.

Зчитування з COM порту реалізоване з допомогою бібліотеки `JSerialComm`, відображення графіку - з допомогою бібліотеки `Xchart`, що базується на `Swing`. Зчитування сигналу і відображення графіку відбуваються в різних потоках виконання. Сигнал зберігається в динамічний масив. Оскільки сигнал паралельно зчитується і записується з різних потоків, то потрібна синхронізація між потоками. В якості механізму синхронізації я застосував ключове слово `synchronized`.

Повний код програми наведений у додатку Б.

4.3. Програмна реалізація обчислення частоти серцевих скорочень на основі ЕКГ за допомогою вейвлетів

Варіабельність серцевого ритму - це явище, що проявляється в зміні інтервалу між сусідніми серцевими циклами. Частота серцевих скорочень це величина обернена до варіабельності серцевого ритму [7].

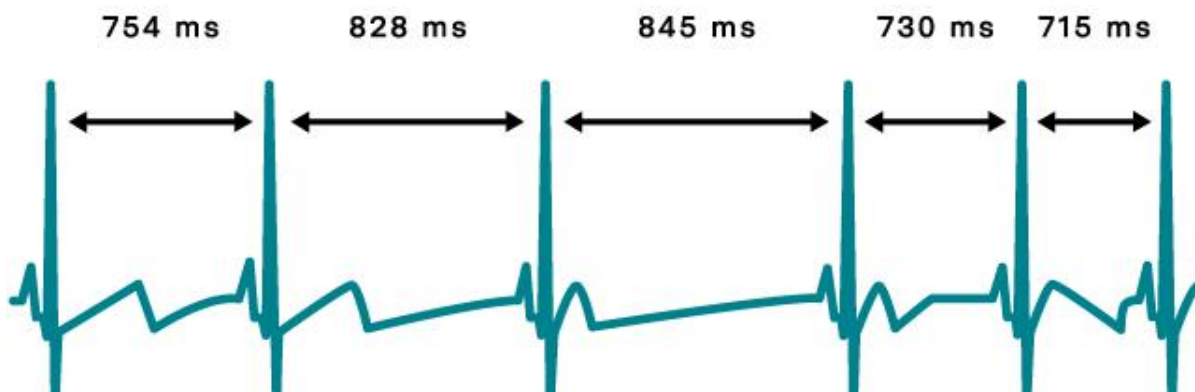


Рисунок 4.4. Варіабельність серцевого ритму

Варіабельність серцевого ритму рахують між сусідніми R піками. Частота R-R лежить в діапазоні між 0 та 0.5 Гц. За допомогою вейвлетів можна зробити частотну декомпозицію сигналу і виділити R піки для спрощення їх знаходження для подальшого обчислення варіабельності. Здійснено дискретне вейвлет перетворення за допомогою вейвлету Добеші 4 рівня над дискретизованим ЕКГ сигналом з частотою дискретизації 360 Гц.

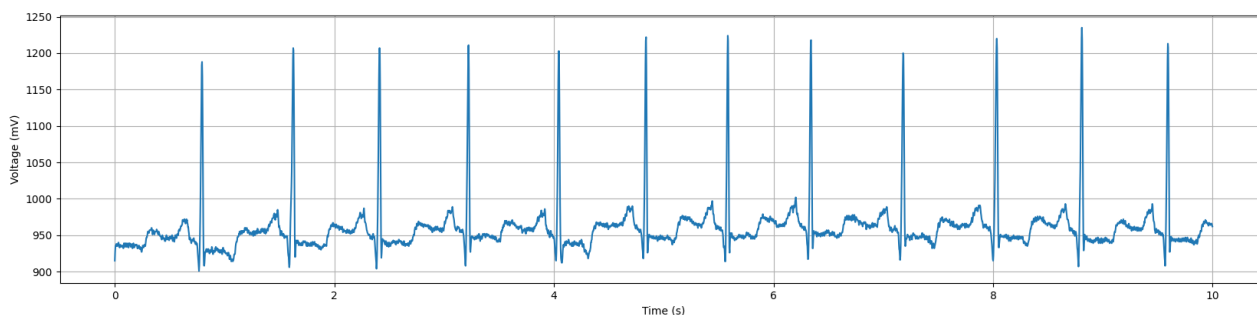


Рисунок 4.5. Сигнал ЕКГ

В результаті декомпозиції сигналу отримано 4 коефіцієнти:

Таблиця 4.1.

Коефіцієнт	Діапазон частот
D_4	Від 0 до $f_n / 16$ Гц
D_3	Від $f_n / 16$ до $f_n / 8$ Гц
D_2	Від $f_n / 4$ до $f_n / 2$ Гц
D_1	Від $f_n / 2$ до f_n Гц

Здійснено обернене вейвлет перетворення, з коефіцієнтами D_4, Z, Z, Z .

Z - це вектор розміру $f_n \cdot t$, заповнений нулями. Таким чином отримано сигнал у діапазоні частот між 0 до $f_n / 16$ Гц.

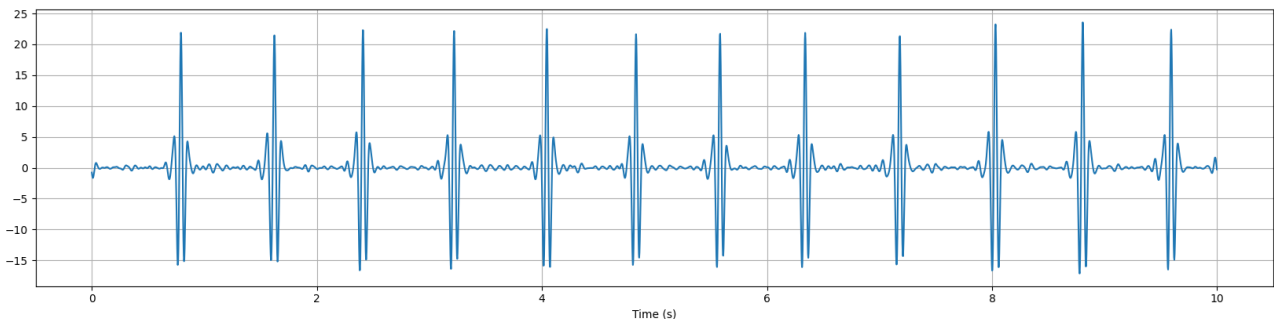


Рисунок 4.6. Оброблений сигнал ЕКГ

Після цього застосував алгоритм, що знаходить усі локальні максимуми. Користуючись вектором локальних максимумів, обрахував частоту серцевих скорочень між усіма R-R піками.

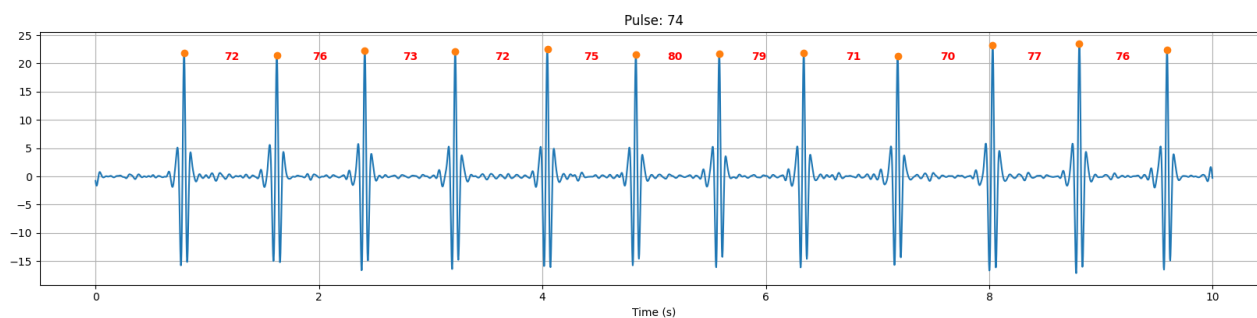


Рисунок 4.7. Оброблений сигнал ЕКГ з відображенням частоти серцевих скорочень між усіма R-R піками

Код програми наведений у додатку Д.

4.4. Програмна реалізація порівняння цифрових фільтрів для мінімізації ефекту “коливання осі” в ЕКГ

Здійснено програмну реалізацію на основі досліджень і формул із пункту 2.3.

Програма написана мовою програмування Python з використанням таких пакетів як:

1. NumPy - бібліотека для векторних обчислень
2. SciPy - бібліотека для математичних обчислень
3. Matplotlib - бібліотека для візуалізації

Програма відображає графік початкового сигналу та графік сигналів після застосування фільтрів, проаналізованих у пункті 2.3, та результат перетворення Фур’є в діапазоні між 0 - 3 Гц.

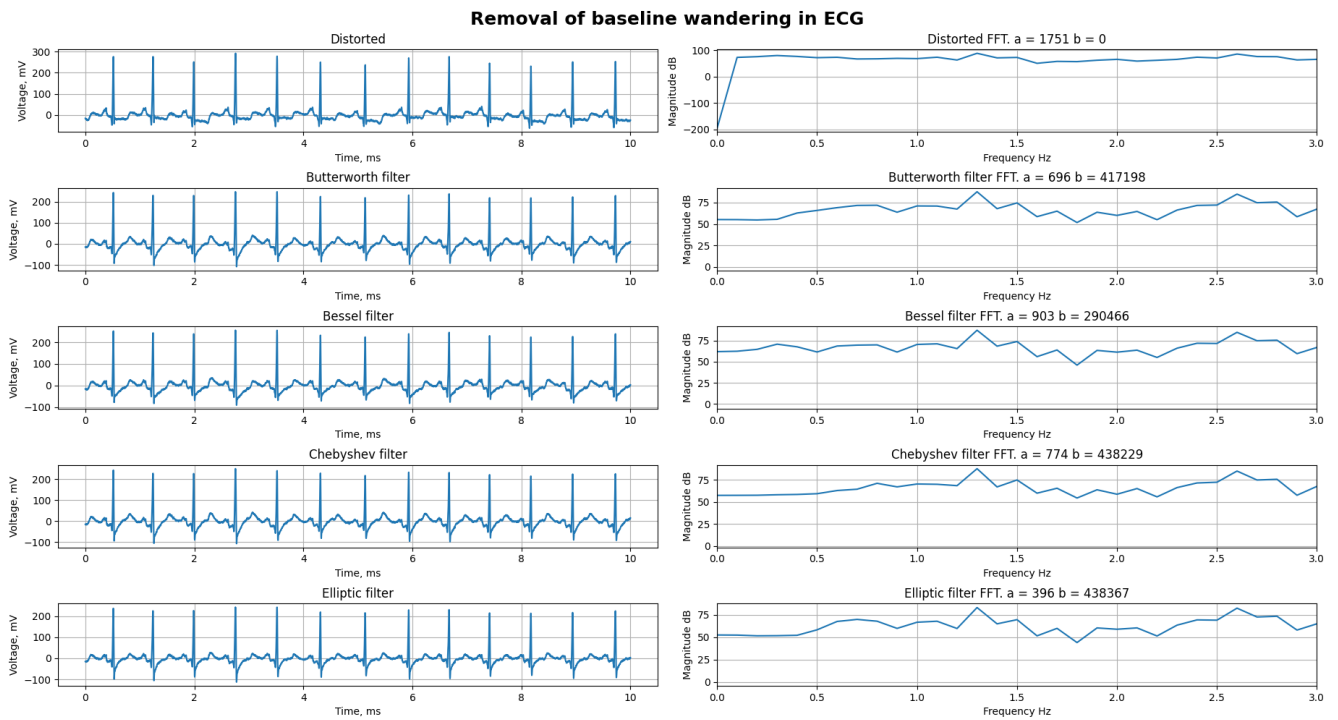


Рисунок 4.6. Програмна реалізація для мінімізації ефекту “коливання осі” в ЕКГ

На рисунку наведений результат виконання програми. Інтерфейс відображає 4 сигнали зліва та їх відповідні перетворення Фур’є в діапазоні між 0 - 3 Гц.

1. Початковий сигнал
2. Сигнал після застосованого фільтру Батерворта
3. Сигнал після застосованого фільтру Бесселя
4. Сигнал після застосованого фільтру Чебишева
5. Сигнал після застосованого еліптичного фільтру

Додатково, для кожного сигналу відображаються коефіцієнти a та b , де a - значення обраховане згідно з формулою 2.1, b - значення обраховане згідно з формулою 2.2.

Результати експерименту:

Таблиця 4.2

Номер	Фільтр Батерворда		Фільтр Бесселя		Фільтр Чебишева		Еліптичний фільтр	
	a	b	a	b	a	b	a	b
1	696	417198	903	290466	774	438229	396	438367
2	632	667790	1148	460959	620	705252	329	667746
3	3435	985507	2477	688727	710	1024045	1651	1014071
4	1284	1020117	2298	693553	227	1059035	122	1040981
5	609	566517	2359	443196	1190	599273	949	556709
6	2467	433784	1706	327487	1662	459997	74	417844
7	700	839465	1071	585732	317	873925	1083	866445

Висновки: в результаті експерименту з'ясовано що фільтр Бесселя суттєво мінімізує ефект “коливання осі” в ЕКГ, при цьому він найменше вплинув на частоти вищі 0.5 Гц, відповідно він практично не спотворює ЕКГ. Інші фільтри в деяких випадках набагато краще придушують небажані частоти, але при цьому вони більше спотворюють сигнал ніж фільтр Бесселя. Це пояснюється тим, що фільтру Бесселя притаманна лінійна фазо-частотною характеристикою, в той час як решта фільтрів мають нелінійну фазо-частотну характеристику. Я вважаю, що фільтр Бесселя є найкращим вибором з-поміж інших фільтрів для мінімізації ефекту коливання осі, оскільки він найменше спотворює форму ЕКГ, що важливо для діагностування серцево-судинних захворювань.

Роздрук програми наведено у Додатку Г.

ВИСНОВКИ

У ході виконання цієї роботи розглянуто основні аспекти фізіології серця, а саме будова серця, серцеві скорочення тощо, та їх зв'язок з сигналом електрокардіограми (формування комплексів, зубців, інтервалів серцевих скорочень тощо). Зроблено огляд сучасних засобів та систем діагностики серця, проаналізовано їх переваги та недоліки.

Проаналізовано методи цифрової обробки сигналів - перетворення Фур'є, віконне перетворення Фур'є та вейвлет перетворення. Розглянуто їх переваги та недоліки, а також алгоритми їх програмної реалізації. Проаналізовано цифрові фільтри та їх переваги та недоліки перед апаратними фільтрами.

Розроблено мікроконтролерну систему, що зчитує електричний сигнал з електродів, прикріплених до тіла людини, здійснює його підсилення та очищення його від шумів і передає сигнал на комп'ютер. Проаналізовано вбудовані в неї підсилювачі та апаратні фільтри. Проаналізовано принцип роботи протоколу UART. Проаналізовано принцип роботи аналогово-цифрового перетворювача та алгоритм роботи АЦП SAR типу, що міститься в мікросхемі AD8232. Оцінено економічну вигідність пристрою.

Розроблено програму для мікроконтролера та програму для комп'ютера для зчитування сигналу в режимі реального часу. Здійснено порівняння цифрових фільтрів для мінімізації ефекту "коливання осі" в ЕКГ за допомогою програмної реалізації. Реалізовано програму для обрахунку варіабельності серцевого ритму та серцевого пульсу за допомогою вейвлетів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Graham Jackson (2009). Heart Health. Class Publishing
2. Electrocardiography - Wikipedia [Електронний ресурс] – URL: en.wikipedia.org/wiki/Electrocardiography / (дата звернення 11.05.2022)
3. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение [Текст] : пер. с англ. / Б. Скляр. – 2-е изд., испр. – М. : Издательский дом «Вильямс», 2003. – 1104 с
4. Analog to Digital Converter (ADC) - EspressIf [Електронний ресурс] - URL: <https://docs.espressif.com/projects/esp-idf/en/v4.4.2/esp32/api-reference/peripherals/adc.html>
5. UART - Wikipedia [Електронний ресурс] – URL: <https://uk.wikipedia.org/wiki/UART> / (дата звернення 12.05.2022)
6. Introduction to Wavelets analysis - Université de Montréal [Електронний ресурс] – URL: <http://www.iro.umontreal.ca/~mignotte/IFT6150/Articles/WaveletIntroduction.pdf> / (дата звернення 12.07.2022)
7. R-R Interval - ScienceDirect [Електронний ресурс] - URL: <http://www.iro.umontreal.ca/~mignotte/IFT6150/Articles/WaveletIntroduction.pdf> / (дата звернення 19.07.2022)
8. Energy of signal - Wikipedia [Електронний ресурс] - URL: [https://en.wikipedia.org/wiki/Energy_\(signal_processing\)](https://en.wikipedia.org/wiki/Energy_(signal_processing)) / (дата звернення 19.09.2022)
9. Основні поняття теорії сигналів - Основні поняття теорії сигналів [Електронний ресурс] - URL: <https://org2.knuba.edu.ua/mod/book/tool/print/index.php?id=23040&chapterid=126> / (дата звернення 01.10.2022)
10. Heart Rate - Wikipedia [Електронний ресурс] - URL: https://en.wikipedia.org/wiki/Heart_rate / (дата звернення 01.06.2022)

11. Baseline wandering - ScienceDirect [Электронный ресурс] - URL:
<https://www.sciencedirect.com/topics/computer-science/baseline-wander>
12. Неако 100G - Мед Магазин [Электронный ресурс] - URL:
https://med-magazin.ua/item_n2665.htm
13. Біомед ІОЗ - Медлан [Электронный ресурс] - URL:
<https://medlan.pro/kardiologiya/elektrokardiografy/ie-3>
14. Теорема Котельникова-Шеннона [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem
15. Analog to digital converter - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Analog-to-digital_converter
16. Successive-approximation ADC - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Successive-approximation_ADC
17. ESP-32 Technical reference manual - Espressif [Электронный ресурс] - URL:
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
18. Fourier transform - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Fourier_transform
19. Discrete Fourier Transform - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Discrete_Fourier_transform
20. Fast Fourier Transform - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Fast_Fourier_transform
21. Short-time Fourier Transform - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Short-time_Fourier_transform
22. Wavelets - Wikipedia [Электронный ресурс] - URL:
<https://en.wikipedia.org/wiki/Wavelet>
23. Ricker Wavelet - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Ricker_wavelet
24. Haar Wavelet - Wikipedia [Электронный ресурс] - URL:
https://en.wikipedia.org/wiki/Haar_wavelet

25. Daubechies Wavelet - Wikipedia [Электронный ресурс] - URL:

https://en.wikipedia.org/wiki/Daubechies_wavelet

26. AD8232 Data Sheet - Analog Devices [Электронный ресурс] - URL:

<https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>

27. Радіочастотний спектр - Wikipedia [Электронный ресурс] - URL:

https://en.wikipedia.org/wiki/Radio_spectrum

28. Filter topology Faceoff - Billey Technologies [Электронный ресурс] - URL:

<https://blog.bliley.com/filter-typology-face-off-a-closer-look-at-the-top-4-filter-types>

29. Phase Response - Wikipedia [Электронный ресурс] - URL:

https://en.wikipedia.org/wiki/Phase_response

30. Frequency Response - Wikipedia [Электронный ресурс] - URL:

https://en.wikipedia.org/wiki/Frequency_response

ДОДАТОК А Роздрук програми для мікроконтролера ESP-32

```

#include <string.h>
#include <stdio.h>
#include "sdkconfig.h"
#include "esp_log.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/semphr.h"
#include "driver/adc.h"
#include "esp32/rom/miniz.h"

#define TIMES 256
#define GET_UNIT(x) ((x>>3) & 0x1)

#define ADC_RESULT_BYTE 2
#define ADC_CONV_LIMIT_EN 1
#define ADC_CONV_MODE ADC_CONV_SINGLE_UNIT_1
#define ADC_OUTPUT_TYPE ADC_DIGI_OUTPUT_FORMAT_TYPE1

// Розмір буферу
#define MAX_STORE_BUF_SIZE 1024
#define CONV_LIMIT_NUM 250

// Частота дискретизації (Гц)
#define SAMPLE_FREQ_HZ 1000

#define ADC1_CHANNEL_MASK BIT(7)
#define ADC2_CHANNEL_MASK 0
#define ADC1_CHANNEL ADC1_CHANNEL_4

// Функція що ініціалізує АЦП в режимі DMA
// Режим DMA краще підходить для виміру високочастотних сигналів
static void configure_adc() {
    adc_digi_init_config_t adc_dma_config = {
        .max_store_buf_size = MAX_STORE_BUF_SIZE,
        .conv_num_each_intr = TIMES,
        .adc1_chan_mask = ADC1_CHANNEL_MASK,
        .adc2_chan_mask = ADC2_CHANNEL_MASK,
    };
    ESP_ERROR_CHECK(adc_digi_initialize(&adc_dma_config));

```

```

adc_digi_configuration_t digital_controller_config = {
.conv_limit_en = ADC_CONV_LIMIT_EN,
.conv_limit_num = CONV_LIMIT_NUM,
.sample_freq_hz = SAMPLE_FREQ_HZ,
.conv_mode = ADC_CONV_MODE,
.format = ADC_OUTPUT_TYPE,
.pattern_num = 1
};

adc_digi_pattern_config_t adc_pattern[SOC_ADC_PATT_LEN_MAX] = {0};
uint8_t unit = GET_UNIT(ADC1_CHANNEL);
uint8_t ch = ADC1_CHANNEL & 0x7;
adc_pattern[0].atten = ADC_ATTEN_DB_11;
adc_pattern[0].channel = ch;
adc_pattern[0].unit = unit;
adc_pattern[0].bit_width = SOC_ADC_DIGI_MAX_BITWIDTH;

digital_controller_config.adc_pattern = adc_pattern;
ESP_ERROR_CHECK(adc_digi_controller_configure(&digital_controller_config));
}

void app_main(void)
{
esp_err_t ret;
uint32_t ret_num = 0;
uint8_t result[TIMES] = {0};
memset(result, 0xcc, TIMES);
configure_adc();
adc_digi_start();

while(1) {
ret = adc_digi_read_bytes(result, TIMES, &ret_num, ADC_MAX_DELAY);
if (ret == ESP_OK || ret == ESP_ERR_INVALID_STATE) {
for (int i = 0; i < ret_num; i += ADC_RESULT_BYTE) {
adc_digi_output_data_t *p = (void*)&result[i];
printf("%d\n", p->type1.data);
}
// Задержка 1 мс
vTaskDelay(1);
}
}

adc_digi_stop();
ret = adc_digi_deinitialize();
}

```

ДОДАТОК Б Роздрук програми для зчитування сигналу на комп'ютері

```

import com.fazecast.jSerialComm.SerialPort;
import com.fazecast.jSerialComm.SerialPortDataListener;
import com.fazecast.jSerialComm.SerialPortEvent;
import org.knowm.xchart.QuickChart;
import org.knowm.xchart.SwingWrapper;
import org.knowm.xchart.XYChart;

import java.io.*;
import java.nio.charset.StandardCharsets;
import java.time.Instant;
import java.util.*;
import java.util.stream.Stream;

public class Main {
    private static final int BAUD_RATE = 115200;
    private static final String COM_PORT = "/dev/ttyUSB0";
    private static final String SERIES_NAME = "ECG";
    private static final String Y_TITLE = "Voltage";
    private static final String X_TITLE = "Time";
    private static final String CHART_TITLE = "ECG Signal";
    private static final int SIZE = 100;

    private static XYChart chart;
    private static SwingWrapper<XYChart> sw;
    private static final List<Integer> ecgDataList = Collections.synchronizedList(new ArrayList<>());
    private static final List<Integer> timeDataList = Collections.synchronizedList(new ArrayList<>());
    private static final List<Integer> errorBarsDataList = Collections.synchronizedList(new ArrayList<>());
    private static final List<Integer> wholeEcgSignal = new ArrayList<>();
    private static int count = 1;
    private static final Instant START = Instant.now();

    public static void main(String[] args) {

        var serialPort = SerialPort.getCommPort(COM_PORT);
        serialPort.setBaudRate(BAUD_RATE);

        serialPort.openPort();

        Runtime.getRuntime().addShutdownHook(new Thread(Main::writeSignalToFile));
    }
}

```

```
serialPort.addDataListener(new SerialPortDataListener() {
```

```
    @Override
```

```
    public int getListeningEvents() {
        return SerialPort.LISTENING_EVENT_DATA_RECEIVED;
    }
}
```

```
    @Override
```

```
    public void serialEvent(SerialPortEvent serialPortEvent) {
        var values = new String(serialPortEvent.getReceivedData()).trim().split("\\s+");
        for (var value : values) {
            System.out.println("Received data = " + value);
        }
        Stream.of(values).map(s -> {
            try {
                return Integer.parseInt(s);
            } catch (NumberFormatException formatException) {
                return null;
            }
        })
        .filter(Objects::nonNull)
        .forEach(v -> {
            wholeEcgSignal.add(v);
            ecgDataList.add(v);
            timeDataList.add(count++);
            errorBarsDataList.add(0);
        });
    }
});
```

```
// Create Chart
```

```
chart = QuickChart.getChart(CHART_TITLE, X_TITLE, Y_TITLE, SERIES_NAME, new double[]{0}, new double[]{0});
```

```
// Show it
```

```
sw = new SwingWrapper<>(chart);
```

```
sw.displayChart();
```

```
TimerTask chartUpdaterTask = new TimerTask() {
```

```
    @Override
```

```
    public void run() {
        if (timeDataList.size() >= SIZE) {
            javax.swing.SwingUtilities.invokeLater(() -> {
```

```
        chart.updateXYSeries(SERIES_NAME, timeDataList, ecgDataList, errorBarsDataList);
        sw.repaintChart();
        timeDataList.clear();
        ecgDataList.clear();
        errorBarsDataList.clear();
    });
}
}
};
Timer timer = new Timer();
timer.scheduleAtFixedRate(chartUpdaterTask, 0, 250);
}

private static void writeSignalToFile() {
    try {
        try (var stream = new BufferedOutputStream(new FileOutputStream("ecg_" + START + ".txt"))) {
            var N = wholeEcgSignal.size();
            for (var i = 0; i < N; i++) {
                if (i > 0) {
                    stream.write("\n");
                }
                stream.write(String.valueOf(wholeEcgSignal.get(i)).getBytes(StandardCharsets.UTF_8));
            }
        }
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

ДОДАТОК В. Програма для мінімізації ефекту коливання осі в ЕКГ

```
import matplotlib
import numpy as np
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from numpy import mean
from scipy import signal, io
import matplotlib.pyplot as plt
from scipy.fft import fft
import tkinter as tk

sampling_frequency = 360
cutoff_frequency = 0.5

root = tk.Tk()

def normal_cutoff_freq(fs, cutoff):
    nyq = 0.5 * fs # Nyquist frequency
    return cutoff / nyq # Normalized cut-off frequency

# mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/1 NSR/100m.mat')
# mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/1 NSR/105m (0).mat')
# mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/1 NSR/106m (0).mat')
# mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/1 NSR/106m (2).mat')
# mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/1 NSR/108m (13).mat')
# mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/1 NSR/228m (5).mat')
mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/7 PVC/123m (1).mat')

ecg = mat['val'][0]
```

```

ecg = ecg - mean(ecg)

N = len(ecg)
T = 1.0 / sampling_frequency

x = np.linspace(0.0, N * T, N)

# Computation of Fourier transform of distorted signal and filtered signals by Butterworth and Bessel filters

yf_distorted = fft(ecg)

xf = np.linspace(0.0, 1.0 / (2.0 * T), N // 2) # Frequency vector

order = 7

filters = [
    (signal.butter(order, cutoff_frequency, btype='highpass', analog=False, output="sos", fs=sampling_frequency), 'Butterworth filter'),
    (signal.bessel(order, cutoff_frequency, btype='highpass', analog=False, norm='mag', output="sos", fs=sampling_frequency), 'Bessel filter'),
    (signal.cheby2(order, 50, cutoff_frequency, btype='highpass', analog=False, output="sos", fs=sampling_frequency), 'Chebyshev filter'),
    (signal.ellip(order, 5, 40, cutoff_frequency, btype='highpass', analog=False, output="sos", fs=sampling_frequency), 'Elliptic filter'),
]

# Visualization

fig, axes = plt.subplots((len(filters) + 1), 2)

fig.suptitle('Removal of baseline wandering in ECG', fontsize = 18, fontweight='bold')

def show_signal_with_fft(x, y, xf, yf, axes, pos, name):
    sum_filtered_amplitudes = 0

```



```

b = 0

count = 0

for i in range(len(xf)):

    f = xf[i]

    if f <= cutoff_frequency:

        sum_filtered_amplitudes += yf[i]

        count += 1

    else:

        b += abs(yf[i] - yf_distorted[i])

a = int(abs(sum_filtered_amplitudes / count))

b = int(b)

axs[pos][0].plot(x, y)

axs[pos][0].set_ylabel("Voltage, mV")

axs[pos][0].set_xlabel("Time, ms")

axs[pos][0].set_title(name)

axs[pos][0].grid()

axs[pos][1].plot(xf, 20 * np.log10(np.abs(yf[0: N // 2])))

axs[pos][1].set_ylabel("Magnitude dB")

axs[pos][1].set_xlabel("Frequency Hz")

axs[pos][1].set_xlim([0, 3])

axs[pos][1].set_title(name + " FFT. a = " + str(a) + " b = " + str(b))

axs[pos][1].grid()

# Distorted signal

# and corresponding Fourier transform, limited to 0 – 3 Hz

show_signal_with_fft(x, ecg, xf, yf_distorted, axs, 0, 'Distorted')

i = 1

```

```
for filter in filters:
```

```
    filtered = signal.sosfilt(filter[0], ecg)
```

```
    show_signal_with_fft(x, filtered, xf, fft(filtered), axs, i, filter[1])
```

```
    i += 1
```

```
fig.set_size_inches(18.5, 10.5, forward=True)
```

```
#fig.canvas.manager.full_screen_toggle() # toggle fullscreen mode
```

```
fig.tight_layout()
```

```
plt.show()
```

```
bar = FigureCanvasTkAgg(fig, root)
```

```
bar.get_tk_widget().pack(side=tk.LEFT, fill=tk.BOTH)
```

```
matplotlib.pyplot.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=None)
```

```
root.mainloop()
```

ДОДАТОК Г. Реалізація перетворення Фур'є мовою програмування Python

```
import numpy as np
from scipy.fft import fft, fftfreq
from scipy import mean
import matplotlib.pyplot as plt
import scipy.io

file_name = '/home/vitaliivorobii/Desktop/master/MLII/1 NSR/100m.mat'
signal_vector = scipy.io.loadmat(file_name)['val'][0]
signal_vector = signal_vector - mean(signal_vector)

frequency = 360
num_points = len(signal_vector)
T = 1.0 / frequency

x = np.linspace(0.0, num_points * T, num_points, endpoint=False)

yf = fft(signal_vector)
xf = fftfreq(num_points, T)[:num_points//2]

figure, axes = plt.subplots(2, 1)

axes[0].plot(x, signal_vector)
axes[0].set_ylabel("Voltage (mV)")
axes[0].set_xlabel("Time (s)")
axes[0].grid()

axes[1].plot(xf, 2.0 / num_points * np.abs(yf[0:num_points // 2]))
axes[1].set_ylabel("Amplitude")
axes[1].set_xlabel("Frequency (Hz)")
axes[1].grid()
figure.tight_layout()
plt.show()
```

ДОДАТОК Д. Роздрук програми для визначення ритму серця

```
import tkinter as tk

import matplotlib.pyplot as plt

import numpy as np

import pywt

from scipy import io

from scipy.signal import find_peaks

sampling_frequency = 360

root = tk.Tk()

mat = io.loadmat('/home/vitaliivorobii/Desktop/master/MLII/1 NSR/100m_1.mat')

ecg = mat['val'][0]

N = len(ecg)

T = 1 / sampling_frequency

figure, axes = plt.subplots(2, 1)

x = np.linspace(0.0, N * T, N, endpoint=False)

axes[0].plot(x, ecg)

axes[0].set_ylabel("Voltage (mV)")

axes[0].set_xlabel("Time (s)")

axes[0].grid()

[(cA4, cD4), (cA3, cD3), (cA2, cD2), (cA1, cD1)] = pywt.swt(ecg, 'db4', level=4, norm=True)

# cD4 = np.zeros(N)

cA4 = np.zeros(N)

cA3 = np.zeros(N)

cD3 = np.zeros(N)
```

```

cA2 = np.zeros(N)
cD2 = np.zeros(N)
cA1 = np.zeros(N)
cD1 = np.zeros(N)

filtered = pywt.iswt([(cA4, cD4), (cA3, cD3), (cA2, cD2), (cA1, cD1)], 'db4')
peaks, _ = find_peaks(filtered, height=20)
sum = 0

for i in range(1, len(peaks)):
    print(((peaks[i] - peaks[i - 1]) / sampling_frequency))
    pulse = int(60 / ((peaks[i] - peaks[i - 1]) / sampling_frequency))
    sum += pulse
    axes[1].text(x[peaks[i]] - 0.4, max(filtered) - 3, s=pulse, weight="bold", color="r", horizontalalignment='center')

avg_pulse = sum / (len(peaks) - 1)

axes[1].plot(x, filtered)
axes[1].plot(x[peaks], filtered[peaks], "o")
axes[1].set_xlabel("Time (s)")
axes[1].set_title("Pulse: " + str(int(avg_pulse)))
axes[1].grid()
figure.tight_layout()
plt.show()

```