

DOI: <https://doi.org/10.15276/ict.03.2026.85>

УДК 004.9:616-036.82

Автономна IoT-система для кінематичного моніторингу реабілітації кистьового суглоба

Стаценко Гліб Сергійович¹⁾

ORCID: <https://orcid.org/0009-0001-2818-6746>; glibstacenko@gmail.com

Іпатов Богдан Сергійович¹⁾

ORCID: <https://orcid.org/0009-0000-3032-5138>; bogdan.goldenn@gmail.com

Рубан Софія Леонідівна¹⁾

ORCID: <https://orcid.org/0009-0002-0533-124X>; 7919863@stud.op.edu.ua

Мельник Дмитро Сергійович¹⁾

ORCID: <https://orcid.org/0009-0002-2837-5923>; melnik.dm.s@gmail.com

¹⁾ Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Робота присвячена вирішенню актуальної науково-практичної задачі об'єктивізації процесу медичної телеметричної реабілітації. Для її вирішення запропоновано інноваційну автономну IoT-систему для об'єктивного кінематичного моніторингу реабілітації кистьового суглоба у реальному часі. Основою такої системи є автономний мобільний пристрій для об'єктивного та кількісного моніторингу рухової активності кистьового суглоба, який використовує парадигму периферійних обчислень. Сформовані основні вимоги до його реалізації - збір телеметричної інформації, яка містить не тільки статистичні параметри руху кистьового суглоба, а й кінематичні показники (фізіологічний та неврологічний тремор) та формування інтерактивного візуального відгуку (біологічний зворотний зв'язок) для підтримки когнітивної залученості пацієнта. Проведено розробку апаратної архітектури, математичне моделювання і програмну реалізацію автономного мобільного пристрою для об'єктивного та кількісного моніторингу рухової активності кистьового суглоба. Пристрій побудовано на базі високопродуктивного мікроконтролера та інерційного вимірювального модуля із вбудованим цифровим процесором руху. Обґрунтовано вибір мікроконтролера та потрібних сенсорів. Особлива увага приділена «аналітичному рушію», який в реальному часі обчислює кут нахилу кисті, магнітуду кутової швидкості та інтегральну оцінку плавності руху. Для алгоритмічної реалізації «аналітичного рушія» Використано потужний математичний апарат гіперкомплексних чисел та вирішена проблема «шарнірного замка» для розрахунку просторової орієнтації. Для надійного підрахунку кількості виконаних рухів в умовах фізіологічного тремору реалізовано адаптивний детектор на базі скінченного автомата із гістерезисом. Задля забезпечення максимальної надійності даних, результати кожної сесії зберігаються локально у реляційній базі даних. Для підвищення залученості пацієнтів у процес реабілітації запропонована імплементація потужного модуля терапевтичних міні-ігор. Перевагою розробленої архітектури автономного мобільного пристрою є незалежність від хмарних сервісів та можливість забезпечити конфіденційність медичних даних. Веб-інтерфейс реалізовано для забезпечення низької затримки при комп'ютерній візуалізації графіка руху. Подальші дії автори бачать у проведенні апробації автономної IoT-системи для об'єктивного кінематичного моніторингу при реабілітації кистьового суглоба у реальному часі в центрах реабілітації.

Ключові слова: інформаційні системи; телеметрична реабілітація; мікроконтролер; аналіз інформації; вбудовані системи та IoT-пристрої; біомедична інженерія; цифрова реабілітація; обробка сенсорних даних

Для цитування: Стаценко Г. С., Іпатов Б. С., Рубан С. Л., Мельник Д. С. «Автономна IoT-система для кінематичного моніторингу реабілітації кистьового суглоба». *Інформатика. Культура. Техніка.* 2026; Том 3 № 2 (4): 1094–1111. DOI: <https://doi.org/10.15276/ict.03.2026.85>

Вступ. Кістково-суглобові захворювання, травми та ураження периферичної нервової системи верхніх кінцівок становлять одну з найгостріших проблем сучасної охорони здоров'я. За даними Всесвітньої організації охорони здоров'я (ВООЗ), понад 1,7 мільярда людей у світі страждають на патології опорно-рухового апарату, з яких значна частка припадає на дисфункції дрібної моторики та кистьових суглобів [1]. В Україні, у зв'язку з масштабними військовими діями, спостерігається стрімке, експоненціальне зростання кількості пацієнтів із тяжкими мінно-вибуховими травмами, контузіїями та складними переломами.

Крім того, не втрачає гостроти проблема пацієнтів після інсульту, для яких відновлення функції верхньої кінцівки є критичним чинником повернення до самостійного повсякденного життя.

Стаття опублікована у відкритому доступі відповідно до умов ліцензії CC BY (<https://creativecommons.org/licenses/by/4.0/deed.uk>)

З точки зору фізіології та неврології, успішне відновлення моторики та амплітуди рухів спирається на здатність мозку формувати нові нейронні зв'язки (синапси) у відповідь на повторювані моторні стимули [2]. Відповідно до теорії Хебба, «нейрони, що активуються разом, зв'язуються разом». Проте ініціація цього процесу вимагає тисяч циклів монотонних повторень (флексій, пронацій), які повинні виконуватися правильно, з поступовим збільшенням амплітуди. У цьому контексті, безперервний, точний моніторинг кожного руху стає не просто бажаним, а абсолютно необхідним інструментом доказової медицини.

На практиці оцінка прогресу пацієнта здебільшого відбувається суб'єктивно лікарем-реабілітологом за допомогою традиційних механічних або гравітаційних гоніометрів.

Такий класичний підхід має низку критичних недоліків. По-перше, він характеризується значною похибкою (від 5 до 10 градусів) та високою залежністю від людського фактору, тобто дослідницькою варіабельністю. По-друге, в умовах значного дефіциту кваліфікованого медичного персоналу неможливо забезпечити індивідуальний контроль за кожним пацієнтом протягом усього тривалого (40-60 хвилин) сеансу механотерапії. Пацієнти часто виконують вправи неправильно, або втрачають мотивацію через відсутність видимого результату. Відсутність об'єктивних, постійних кількісних показників динаміки відновлення унеможливорює своєчасне адаптивне коригування терапевтичного плану.

Сучасні вимоги диктують гостру необхідність переходу від суб'єктивної оцінки до високоточної реабілітації, керованої даними (*англ.* data-driven approach). Сучасний реабілітаційний комплекс повинен забезпечувати збір телеметрії у реальному часі з частотою не менше 50 Гц, оцінку не лише статичної амплітуди, але й кінематичної плавності рухів (враховуючи фізіологічний та неврологічний тремор), а також генерувати миттєвий інтерактивний візуальний відгук (біологічний зворотний зв'язок) для підтримки когнітивної залученості пацієнта. Саме тому розробка апаратно-програмного комплексу, здатного задовольнити ці суворі медичні та інженерні критерії за прийнятну вартість, становить значний науковий та практичний інтерес.

Огляд літератури. Останнє десятиліття ознаменувалося розвитком технологій Інтернету речей (*англ.* IoT) та мініатюризацією електромеханічних систем. Це відкрило шлях до створення портативних пристроїв для об'єктивної просторової оцінки. Системи захоплення руху (Motion Capture) можна умовно поділити на три великі класи: оптичні, електромеханічні та інерційні [3].

Оптичні системи, такі як Vicon, Microsoft Kinect або Leap Motion, базуються на аналізі відеопотоку або інфрачервоної глибини [3]. Хоча вони пропонують високу точність у контрольованих лабораторних умовах, їх використання в клінічній практиці стикається із проблемою оклюзії (перекриття маркера тілом), необхідністю калібрування камер та фіксованою робочою зоною. Електромеханічні системи (наприклад, Armeo від Hocoma) забезпечують відмінну фіксацію та можливість активного силового спротиву, проте вони є стаціонарними, громіздкими, вимагають тривалого налаштування під антропометрію пацієнта і мають вартість у десятки тисяч доларів [3].

Відповідно, найбільший потенціал для масового застосування демонструють системи на базі інерційних вимірювальних модулів, що поєднують 3D-акселерометри та 3D-гіроскопи [3]. Проте існуючі бюджетні аналоги, зазвичай використовують збір даних сенсорами і відправку їх через Wi-Fi на публічні хмарні сервіси (на кшталт Blynk, ThingSpeak, AWS IoT або Firebase) [4]. Однак використання хмарних платформ породжує критичну залежність пристрою від безперебійного, високошвидкісного доступу до Інтернету. В обставинах періодичних блекаутів, або при розгортанні в мобільних польових шпиталях, такі хмарні пристрої стають непрацездатними.

Ще гостріше стоїть питання медичної етики. Передача конфіденційної телеметрії здоров'я пацієнта на сторонні сервери, що фізично розташовані в юрисдикціях інших держав,

створює ризики порушення збереження даних та прямо суперечить жорстким регуляторним стандартам HIPAA (США) та GDPR (ЄС) [4].

З інженерної точки зору, критичним недоліком хмарних IoT-рішень є досить висока мережева затримка. Для функціонування біологічного зворотного зв'язку (коли рух руки миттєво відображається як рух об'єкта на екрані монітора), затримка не повинна перевищувати 30 мілісекунд. Трансконтинентальна маршрутизація пакетів TCP/IP часто призводить до затримок у 150-300 мс. Така розбіжність між відчуттям руху пацієнта та візуальним стимулом викликає явище сенсорного конфлікту, що може знижувати терапевтичний ефект. Таким чином, вирішення науково-практичної задачі об'єктивізації процесу медичної телеметричної реабілітації у реальному часі є актуальним.

Методологія дослідження

З огляду на перелічені недоліки існуючих систем, у рамках даного дослідження запропоновано інноваційну автономну IoT-систему для об'єктивного кінематичного моніторингу реабілітації кистьового суглоба у реальному часі. Система складається з механічного реабілітаційного обладнання та програмно-апаратної системи. Її Остання містить комплект сенсорів, мікроконтролер та автономний мобільний пристрій RehabDevice для об'єктивного та кількісного моніторингу рухової активності кистьового суглоба, який використовує парадигму периферійних обчислень (англ. Edge Computing) замість хмарних технологій. Ідея полягає у тому, що обчислювальна потужність наближається максимально близько до джерела генерації даних. Замість того, щоб транслювати вимірювання датчика на віддалений сервер, пристрій виконує обчислення (сенсорне злиття, фільтрацію шумів та інше) безпосередньо на своєму локальному мікроконтролері. Це дозволяє значно зменшити обсяг мережевого трафіку з мегабайтів до кілобайтів на хвилину, мінімізуючи енергоспоживання.

Розглянемо його складові та їх особливості.

1. Запропонована архітектура програмно-апаратної системи. Запропонований пристрій RehabDevice являє собою автономний локальний IoT-сервер. Пристрій не вимагає від реабілітолога або IT підтримки реабілітаційної установи надання доступу до їхньої інфраструктурної Wi-Fi мережі (що часто ускладнено політиками безпеки лікарень). Натомість мікроконтролер активує власний радіомодуль у режимі програмної точки доступу (Software Access Point, SoftAP) за стандартом 802.11 b/g/n, генеруючи захищену мережу «Rehab_Device_Net». З точки зору інформаційної безпеки, розгорнута локальна мережа використовує протокол шифрування WPA2-PSK (AES). Оскільки медична телеметрія належить до конфіденційних даних пацієнта (HIPAA/GDPR) [5], використання жорсткого апаратного шифрування на рівні Wi-Fi MAC контролера повністю унеможливорює перехоплення трафіку (Sniffing) у межах реабілітаційного центру.

Взаємодія з користувачем максимально спрощена завдяки запровадженню технології Captive Portal (Перехоплюючий портал). Коли смартфон, планшет або ноутбук реабілітолога підключається до мережі пристрою, локальний DNS-сервер мікроконтролера перехоплює всі запити на розв'язання доменних імен і перенаправляє пристрій клієнта на внутрішню IP-адресу пристрою. У результаті, операційна система клієнта автоматично відкриває браузер із медичною панеллю управління, без необхідності встановлювати будь-які додаткові мобільні додатки з App Store чи Google Play. Уся база даних пацієнтів та історія тренувань зберігаються на вбудованому Flash-накопичувачі. Така архітектура гарантує високу автономність, нульову залежність від Інтернету, абсолютну конфіденційність медичних даних та низьку мережеву затримку [6].

2. Програмно-алгоритмічна підтримка збору телеметричної інформації. Постільки основною задачею RehabDevice є збір телеметричної інформації, яка містить не тільки статистичні параметри руху кистьового суглоба, а й кінематичні показники (фізіологічний та неврологічний тремор), які реалізовані в аналітичному рушії, то для її одержання

необхідно розглянути основні підходи для визначенні орієнтації та зсувів тривимірного об'єкта.

Фундаментом будь-якої системи просторового відстеження є її математична модель. Традиційні та більш прості системи часто використовують кути Ейлера для опису орієнтації тривимірного об'єкта. Однак анатомія кистьового суглоба дозволяє виконувати рухи із високою амплітудою. При моделюванні складних, комбінованих рухів (наприклад, одночасна сильна флексія та пронація), коли один з кутів наближається до ± 90 градусів, дві з трьох осей обертання вирівнюються в одній площині. Цей феномен відомий як втрата ступеня свободи або «шарнірний замок» (англ. Gimbal Lock). Він призводить до нескінченних значень у тригонометричних матрицях переходу і, як наслідок, до непередбачуваних, різких стрибків розрахованого кута на екрані, що є неприпустимим для медичного приладу.

Щоб уникнути цих недоліків, в основі аналітичного рушія RehabDevice запропоновано використовувати потужний математичний апарат гіперкомплексних чисел. Такі числа розширюють концепцію комплексних чисел на чотиривимірний простір: $q = w + x_i + y_j + z_k$. Їх застосування забезпечує неперервну зміну просторової орієнтації, оскільки обертання задається єдиною віссю та кутом навколо неї [7]. Апаратний цифровий процесор (DMP) датчика виконує сенсорне злиття і генерує 16-бітні цілі гіперкомплексні числа з частотою 100 Герц. Ці дані алгоритмічно зчитуються мікроконтролером та нормалізуються у діапазон дійсних чисел $[-1.0, 1.0]$.

Для фільтрації пошкоджених в процесі I2C-транзакції пакетів, кожне гіперкомплексне число проходить обов'язкову перевірку на одиничну норму вектору (допустима похибка $\pm 2\%$):

$$0.98 < \|q\|^2 = w^2 + x^2 + y^2 + z^2 < 1.02.$$

З перевіреного чотиривимірного вектору $q = (w, x, y, z)$ виділяється ключова клінічна метрика – кут нахилу кисті (англ. Roll), що відповідає фізіологічному руху згинання/розгинання. Перетворення з простору гіперкомплексних чисел в одновимірний кут виконується за оптимізованою тригонометричною формулою (застосовується функція $\arctan2$ для коректного визначення квадранта):

$$\theta_{roll} = \arctan2(2 \cdot (w \cdot x + y \cdot z), 1 - 2 \cdot (x^2 + y^2)).$$

Враховуючи, що анатомічна будова руки кожного пацієнта індивідуальна, а кріплення пристрою на передпліччі може мати випадковий кутовий зсув, у модель було введено функцію динамічного калібрування нульової точки. Під час калібрування пацієнт кладе руку горизонтально, і поточний абсолютний кут фіксується у пам'яті системи як еталонний зсув (англ. offset).

Подальші обчислення використовують диференціальний кут:

$$\theta_{calib} = \theta_{raw} - \theta_{offset}.$$

Окрім простої амплітуди рухів, показником неврологічного відновлення (особливо у пацієнтів з ішемічним інсультом або хворобою Паркінсона) є наявність патологічного тремору та рівень м'язової спастичності. Аналіз простої кутової швидкості (першої похідної координати) не дає об'єктивної оцінки того, наскільки гладко і контролювано відбувається рух.

Для квантифікації плавності розроблено модель на основі обчислення просторової магнітуди вектору кутової швидкості з триосьового гіроскопа:

$$\Omega_i = \sqrt{\omega_{x,i}^2 + \omega_{y,i}^2 + \omega_{z,i}^2}.$$

Біомеханічним індикатором різкості або судомності руху є перша похідна кутової швидкості, або друга похідна кута, що в кінематиці має назву кутовий джерк (англ. Jerk) [8]. Джерк відображає швидкість зміни прискорення.

Чим вище його значення, тим більш переривчастим, невпевненим є рух пацієнта:

$$J_i = \frac{|\Omega_i - \Omega_{i-1}|}{\Delta t} > 300^\circ/s^2.$$

Система накопичує (інтегрує) абсолютні значення джерка за весь час виконання одного повного циклу згинання/розгинання. Отриманий сумарний показник (K) нормується на загальну амплітуду руху. Фінальна клінічна оцінка плавності S видається у відсотках (від 0 до 100 %).

Значення 100 % свідчить про ідеально скоординований, плавний рух здорової людини без жодних ознак тремору:

$$S = \max(0, 100 - 1.5 \cdot K).$$

Підрахунок кількості виконаних рухів є однією з базових задач системи. Однак, розробка надійного алгоритму виявилася нетривіальною. Використання простого порогового детектора (наприклад, реєстрація флексії, коли кут перевищує 30 градусів) виявилось непрацездатним в умовах реальної клінічної практики. Через наявність фізіологічного тремору або невпевнених мікрорухів руки пацієнта поблизу тригерної межі, кут багаторазово і дуже швидко перетинає порогове значення туди й назад. Це явище призводить до хибного багаторазового зарахування десятків уявних рухів за одну секунду. Для вирішення цієї проблеми розроблено логічну модель на базі скінченного автомата (Finite State Machine, FSM), що функціонує як адаптивний гістерезисний детектор.

Автомат має три логічно стійкі стани: IDLE (спокій або очікування), FLEXING (рух у напрямку згинання) та EXTENDING (рух у напрямку розгинання).

Фундаментальна відмінність полягає у тому, що перехід між цими станами здійснюється не за абсолютним значенням кута, а виключно тоді, коли направлена різниця (дельта) між максимальним досягнутим піком і поточним кутом перевищує адаптивний гістерезисний бар'єр (зафіксований на рівні 4 градуси).

Такий математичний бар'єр формує так звану «мертву зону», яка ефективно відфільтровує будь-який високочастотний шум, спазми та тремор, абсолютно не допускаючи хибних спрацювань автомата. Логічна матриця переходів та графічне представлення гістерезисного детектора флексій з відбраковуванням тремору представлена на Рис. 1.

Лише після того, як автомат проходить повний життєвий цикл (успішне згинання, досягнення піку та підтверджене розгинання у вихідну позицію), генерується подія інкрементації глобального лічильника.

Одночасно з цим обчислюється і зберігається загальна амплітуда (Range of Motion, RoM) саме цього конкретного руху, як різниця між глобальним максимумом і мінімумом у рамках завершеного циклу:

$$C_{flex} = C_{flex} + 1,$$

$$A_{RoM} = \theta_{max} - \theta_{min}.$$

Важливою фізіологічною особливістю нейрореабілітації є необхідність ізометричних навантажень – утримання суглоба у фіксованому напруженому стані на піку флексії. Для визначення цього показника, в аналітичній русій системи інтегровано детектор статичного утримання (Isometric Hold Detection). Алгоритм безперервно відстежує стан, коли поточний кут відхиляється від пікового не більше ніж на допустиме значення (ANALYTICS_HOLD_TOLERANCE_DEG = 2.0°), а кутова швидкість залишається мінімальною (менше 5.0°/с). Час такого утримання акумулюється і записується як окрема метрика статичної витривалості пацієнта.

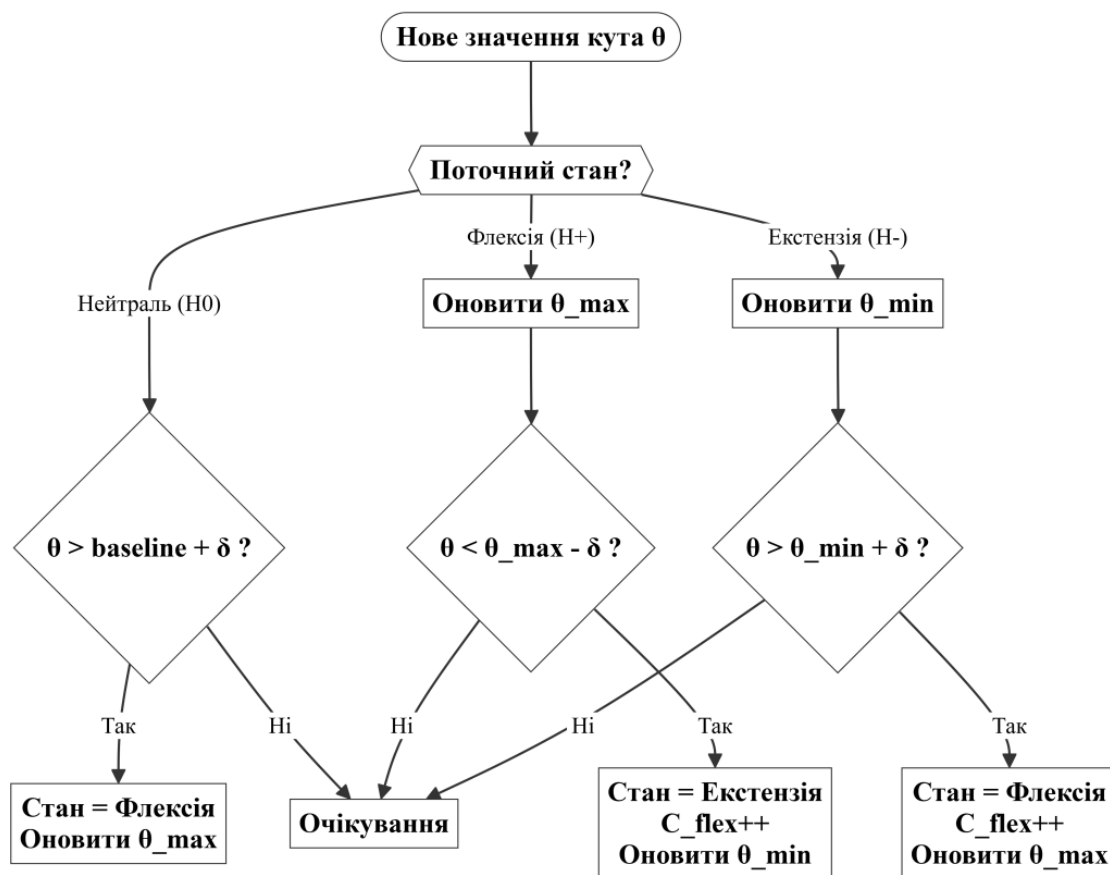


Рис. 1. Автомат станів гістерезисного детектора флексій з відбраковуванням тремору

Джерело: розроблено авторами

Процес верифікації розроблених математичних моделей в режимі *in vivo* довів, що використання гістерезису 4° дає 100% точність підрахунку вправ навіть у пацієнтів з неврологічним тремором. Порівняння з еталонним професійним оптичним гоніометром високої роздільної здатності (аналіз 50 циклів згинання/розгинання) продемонструвало, що абсолютна похибка розрахунку кутів системою RehabDevice не перевищує 1.2 градуса (за графіками Бленда-Алтмана [9]).

3. Зберігання даних у системі. Щодо збереження даних, традиційно у простих вбудованих системах (Arduino тощо) дані логуються шляхом дописування рядків у текстові CSV-файли на зовнішню SD-картку із використанням файлової системи FAT32.

Проте такий підхід має два важливі недоліки.

По-перше, FAT32 вкрай вразлива до раптового зникнення живлення: якщо батарея розрядиться під час відкритого дескриптора запису, таблиця розміщення файлів буде пошкоджена, і всі медичні дані за місяць будуть втрачені.

По-друге, такий підхід не забезпечує можливості швидкого агрегованого пошуку (наприклад, вибірки всіх тренувань з амплітудою $>40^\circ$ за минулий вівторок).

Оскільки текстовий файл не має індексів, алгоритм змушений зчитувати його від початку до кінця, що дає лінійну складність пошуку $O(N)$, і на слабких мікроконтролерах це може займати хвилини.

Для забезпечення надійності та швидкодії у RehabDevice реалізовано реляційне зберігання даних за допомогою потужної СУБД SQLite 3, ядро якої, написане на мові C, було портовано безпосередньо у прошивку мікроконтролера ESP32 [10]. Файли бази даних розміщуються на внутрішньому розділі Flash-пам'яті (обсягом 2 Мегабайти), який відформатовано у LittleFS. На відміну від FAT32, LittleFS є сучасною транзакційною

файловою системою, створеною спеціально для NOR-флеш пам'яті. Вона автоматично, на рівні драйвера, виконує вирівнювання зносу комірок пам'яті (*англ.* Wear Leveling), рівномірно розподіляючи запис по всьому масиву, та забезпечує абсолютну гарантію цілісності даних при обриві живлення (*англ.* Power-Loss Resilience).

Схема бази даних є оптимізованою і містить основну таблицю `sessions`, де первинним ключем (Primary Key) виступає мітка часу (UNIX Timestamp). Кожен реляційний запис інкапсулює детальну статистику: загальну тривалість тренування, кількість валідованих флексій, середню амплітуду по сеансу, відсоток плавності (S), а також масив графіка руху всього тренування. Для зменшення розміру бази даних масив графіка серіалізується у щільний бінарний формат (BLOB) перед вставкою. Усі операції вставки виконуються у вигляді транзакцій за стандартом ACID (Atomicity, Consistency, Isolation, Durability) [9].

При середньому розмірі одного запису у 450 байт, місткість локальної бази даних до циклічного перезапису становить:

$$n_{max} \approx 4400 \text{сесій}$$

Обчислювальна складність алгоритмів (Time & Space Complexity) є вирішальним інженерним фактором, що визначає можливість реалізації такої складної інформаційної моделі на обмежених ресурсах мікроконтролера. Враховуючи жорсткі вимоги до затримки (менше 15-20 мс на весь цикл), всі математичні алгоритми були математично оптимізовані для мінімізації споживання процесорних тактів.

Розрахунок кутів, визначення джерка та переходи у FSM-детекторі виконуються без використання циклів і масивів історії, отже їх часова складність є строго константною, незалежно від тривалості самої сесії реабілітації:

$$T_{FSM}(N)=O(1).$$

Завдяки індексації таблиць у SQLite за допомогою структури B-дерев (*англ.* B-tree) з розміром сторінки 4 КБ, операції пошуку та вибірки історії реабілітації (навіть серед тисяч записів) для відображення в інтерфейсі виконуються за логарифмічний час $O(\log N)$. Це забезпечує миттєве завантаження медичної статистики у браузері лікаря.

4. Архітектура програмного забезпечення (freertos). Стандартний підхід програмування 8- та 16-бітних мікроконтролерів передбачає використання єдиного нескінченного циклу (Superloop, `while(1)`), в якому всі функції викликаються лінійно і послідовно. У багатофункціональній архітектурі RehabDevice такий підхід є категорично неприйнятним. Транзакційний запис даних у Flash-пам'ять через LittleFS або обробка важкого HTTP-запиту від браузера клієнта може зайняти від 50 до 200 мс процесорного часу.

В цей самий час, апаратний таймер датчика MPU6050 генерує зовнішнє переривання рівно кожні 10 мс (частота 100 Герц), вимагаючи негайного зчитування даних з шини I2C. Якщо процесор буде заблокований веб-сервером, він пропустить переривання. Це неминуче призведе до переповнення FIFO-буфера датчика, втрати кінематичних кадрів, і як наслідок – до повного розпаду математичної моделі інтегрування джерка.

Тому програмний стек RehabDevice побудовано на базі промислової операційної системи жорсткого реального часу FreeRTOS (Real-Time Operating System). ОС налаштована на використання парадигми асиметричного мультипроцесингу (*англ.* Asymmetric Multiprocessing, AMP). Враховуючи двоядерну архітектуру мікроконтролера ESP32, задачі були фізично розділені між ядрами. Увесь складний і потенційно повільний мережевий стек (LwIP, драйвери Wi-Fi радіотракту, обробка TCP-сокетів та веб-сервер) примусово ізольований і виконується виключно на Ядрі 1 (APP_CPU). Натомість, усі жорсткі математичні та сенсорні задачі, які вимагають детермінованого часу виконання, прив'язані до Ядра 0 (PRO_CPU).

Планувальник (Scheduler) FreeRTOS налаштовано на алгоритм пріоритетного витіснення (Preemptive Scheduling) [11]. У системі паралельно працюють чотири основні потоки (Tasks).

Найвищий пріоритет (Level 4) має задача TaskSensor, яка відповідає за спілкування з MPU6050. Пріоритет 3 (Level 3) має TaskAnalytics (обчислення FSM та джерка). Пріоритет 2 (Level 2) призначено для TaskDatabase (робота з SQLite). Найнижчий пріоритет (Level 1) має задача TaskNetwork.

Для мінімізації затримок, робота критичної задачі TaskSensor керується безпосередньо апаратним перериванням (англ. Interrupt Service Routine, ISR) від піну INT датчика MPU6050. Згідно з правилами розробки під RTOS, усередині самого обробника переривань категорично заборонено виконувати будь-які обчислення або I2C-транзакції. Натомість, ISR використовує спеціальний механізм прямих сповіщень операційної системи (`vTaskNotifyGiveFromISR`). Ця мікрокоманда інструктує планувальник негайно виконати перемикання контексту (Context Switch) і розбудити TaskSensor. Час реакції на апаратну подію в такій архітектурі становить 2-3 мікросекунди. Після успішного зчитування даних по шині I2C, пакет формується у структуру і відправляється до TaskAnalytics через потокобезпечну міжпроцесну чергу (IPC Queue / Ring Buffer). Логічна архітектура обміну даними та синхронізації між задачами FreeRTOS наведена на Рис. 2.

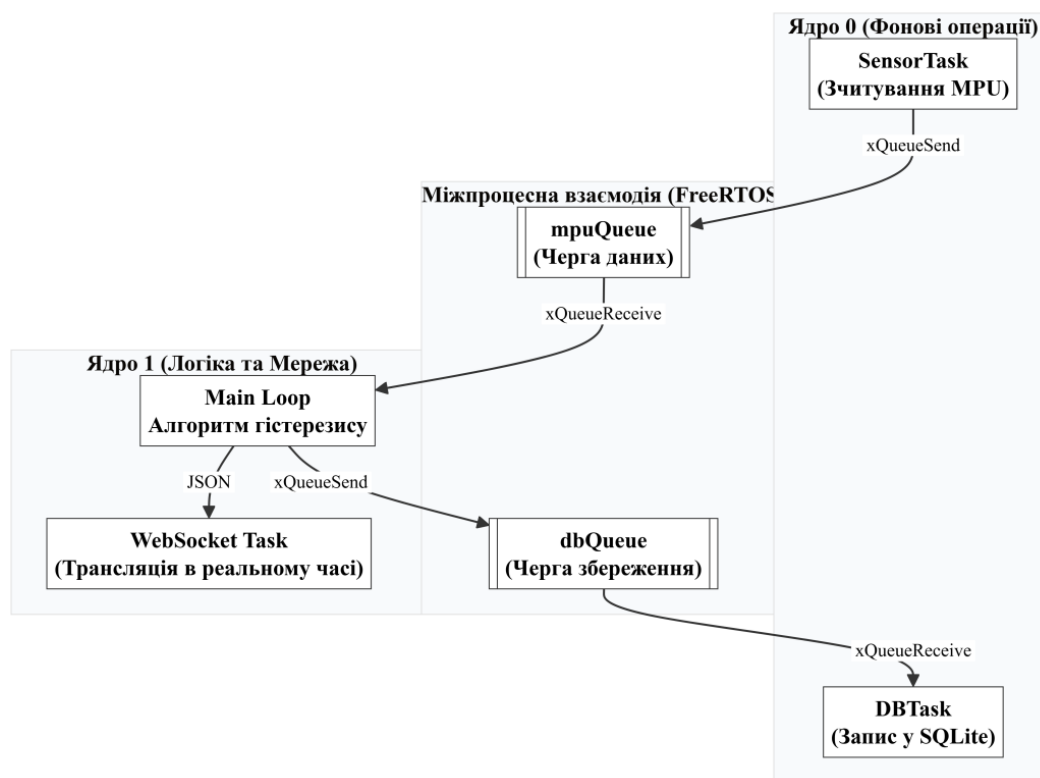


Рис. 2. Блок-схема багатозадачного алгоритму прошивки та IPC-черг (FreeRTOS)

Джерело: розроблено авторами

Ще одним важливим архітектурним рішенням стало усунення класичної для багатьох RTOS проблеми – інверсії пріоритетів (Priority Inversion). Оскільки низько пріоритетна задача TaskDatabase періодично блокує системну шину SPI (під час транзакцій SQLite), вона теоретично могла б заблокувати і високо пріоритетну задачу TaskSensor, яка також потребує доступу до периферії. Для вирішення цієї колізії в RehabDevice імплементовано механізм м'ютексів з успадкуванням пріоритетів (англ. Priority Inheritance Mutex). Якщо високо пріоритетна задача намагається захопити ресурс, який зараз заблокований низько пріоритетною задачею, планувальник автоматично і тимчасово підвищує пріоритет низько пріоритетної задачі до найвищого рівня. Це змушує її виконатися якнайшвидше, звільнити м'ютекс і повернути управління TaskSensor.

Бюджет оперативної пам'яті (RAM) мікроконтролера для структур FreeRTOS був ретельно оптимізований за допомогою інструментів профілювання. Для абсолютного

запобігання переповнення стеку (англ. Stack Overflow), що є найчастішою причиною аварійних перезавантажень, розміри виділеної пам'яті складають: TaskSensor – 4 КБ, TaskAnalytics – 8 КБ, TaskDatabase – 16 КБ (навмисно збільшено для буферизації 4-кілобайтних сторінок В-дерева SQLite), TaskNetwork – 8 КБ.

Таким чином, сумарний обсяг статично зарезервованої пам'яті (Task TCB + Stack) для ОС становить:

$$RAM_{RTOS}=36\text{КБ.}$$

5. Апаратна реалізація. Для реалізації розробленої моделі необхідно обрати мікроконтролер, який здатен одночасно забезпечувати високу обчислювальну потужність (для роботи повноцінної бази даних SQLite та розрахунку гіперкомплексних чисел), мати вбудований бездротовий модуль (для веб-сервера) та працювати в умовах жорсткого енергозбереження. Розглядалося декілька альтернатив. Популярна лінійка STM32 (ARM Cortex-M) вимагає підключення зовнішнього Wi-Fi модуля (наприклад, ESP8266 через UART), що суттєво збільшує габарити друкованої плати, ускладнює трасування сигналів та підвищує загальну вартість. Мікроконтролер Raspberry Pi Pico W має достатню потужність, але відзначається надмірно високим енергоспоживанням у режимі радіотрансляції. Сучасні чіпи серії nRF52 від Nordic Semiconductor ідеально підходять для протоколу Bluetooth Low Energy (BLE), але не здатні апаратно підтримувати повноцінний автономний Wi-Fi SoftAP сервер із високою пропускну здатністю, необхідною для передачі графіків у реальному часі.

Враховуючи ці обмеження, як «мозок» системи було обрано мікросхему ESP32-WROOM-32 (розробка Espressif Systems). Ця система-на-кристалі (SoC) містить два обчислювальні ядра архітектури Xtensa LX6, що працюють на тактовій частоті 240 МГц. Наявність 520 КБ оперативної пам'яті (SRAM), 4 МБ вбудованої Flash-пам'яті та повністю інтегрованого Wi-Fi MAC/Baseband (з власною PCB-антоною) на 100% задовольняє всі архітектурні вимоги проєкту. Такий вибір дозволив фізично інтегрувати HTTP/WebSocket-сервер, реляційну базу даних та обробку даних швидкісних датчиків на одному кристалі розміром 18x25 мм, радикально мінімізуючи кінцеву вартість, масу і габарити медичного пристрою.

Що стосується сенсорної підсистеми, зчитування мікроконтролером «сирих» даних від акселерометра та гіроскопа і подальше їх програмне злиття (Sensor Fusion) за допомогою комп'ютерно-інтенсивних алгоритмів (наприклад, градієнтного фільтра Маджвіка або розширеного фільтра Калмана EKF) безпосередньо на ESP32 споживає непропорційно велику частку процесорного часу. Це також призводить до перегріву чипа і виникнення температурного дрейфу MEMS-датчиків. Цю проблему вирішує використання 6-осьового IMU-модуля MPU6050 високої інтеграції. Головна особливість цієї мікросхеми полягає у наявності закритого апаратного співпроцесора – цифрового процесора руху (Digital Motion Processor, DMP). На етапі ініціалізації системи, ESP32 програмно завантажує у внутрішню пам'ять MPU6050 спеціальний мікро код (firmware) розміром 3 КБ.

Після цього DMP автономно, з частотою 200 Гц, фільтрує вібрації, калібрує гіроскопи, обчислює гіперкомплексні числа і складає їх у власний апаратний FIFO-буфер обсягом 1024 байти. Мікроконтролеру залишається лише раз на 10 мс вичитувати результати по синхронній шині I2C (на частоті 400 кГц), витрачаючи на це мікросекунди. Структурну схему апаратної взаємодії компонентів наведено на Рис. 3.

Апаратне забезпечення розроблялося з урахуванням жорстких вимог до електричної безпеки медичних приладів (гальванічна розв'язка). Енергозабезпечення комплексу здійснюється від стандартного літій-іонного акумулятора формату 18650 з ємністю 3400 мАгод. Спеціально розроблений каскад живлення базується на інтелектуальному контролері заряду TP4056 (з вбудованим термо захистом) та Low-Dropout (LDO) регуляторах напруги, які забезпечують стабільну напругу 3.3 В з високим коефіцієнтом придушення пульсацій, що

критично важливо для точності MEMS-датчиків. Інженерне профілювання струму показало наступні результати: у найважчому режимі роботи (генерація Wi-Fi мережі, трансляція даних по WebSocket 20 разів на секунду, розрахунок математики та періодичний запис у Flash) ESP32 споживає близько 220-240 мА. Сам датчик MPU6050 споживає всього 3.9 мА. Даний бюджет загальної потужності (менше 1 Вт) гарантує понад 11 годин безперервної, активної роботи пристрою. Враховуючи, що середня реабілітаційна сесія триває 45 хвилин, RehabDevice здатен працювати без підзарядки протягом двох тижнів щоденних тренувань.

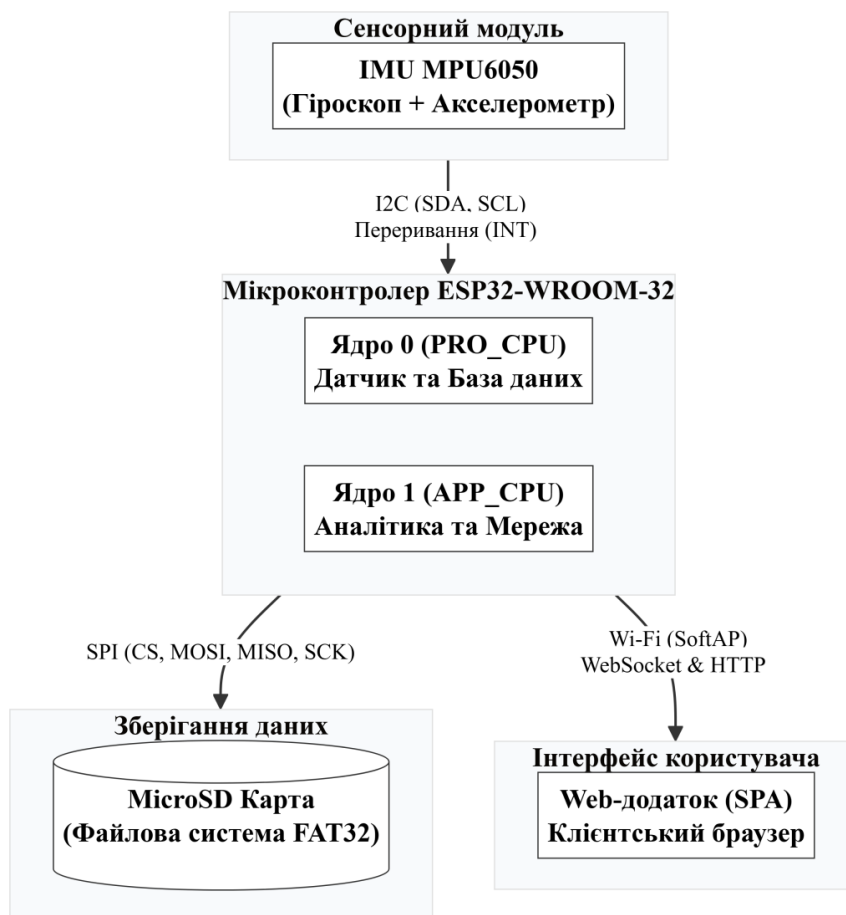


Рис. 3. Апаратна архітектурна схема пристрою RehabDevice та шини даних

Джерело: розроблено авторами

6. Оновлення прошивки та відмовостійкість. Медичне обладнання вимагає періодичного оновлення програмних алгоритмів, додавання нових терапевтичних програм або патчів безпеки. Проте фізичне підключення пристрою через USB-кабель до комп'ютера розробника є неприйнятним з двох причин.

По-перше, згідно із санітарно-епідеміологічними нормами, корпус медичного датчика має бути максимально герметичним (стандарт не гірше IP65), що унеможливорює наявність відкритих роз'ємів, у яких можуть скупчуватися бактерії.

По-друге, переривання процесу прошивки через кабель (наприклад, через випадкове відключення або розряд батареї ноутбука) може назавжди пошкодити таблицю векторів переривань мікроконтролера, перетворивши коштовний пристрій на неприцездатну «цеглину» (Hardware Bricking).

У системі RehabDevice цю задачу успішно вирішено (Рис. 4) Імплементовано багаторівневий відмовостійкий алгоритм бездротового оновлення мікропрограми через повітря (Over-The-Air, OTA). Фундаментом алгоритму є застосування двобанкової архітектури флеш-пам'яті (Dual-Bank Partitioning). Внутрішня пам'ять ESP32 жорстко

розбита на ізольовані логічні розділи: factory (заводська версія), ota_0 (робоча версія) та ota_1 (резервна версія). Під час завантаження нового зкомпільованого бінарного файлу (.bin) через захищений веб-інтерфейс лікаря, мікроконтролер ніколи не перезаписує поточну працюючу прошивку. Запис відбувається у фоновому режимі виключно у пасивний, неактивний OTA-розділ.

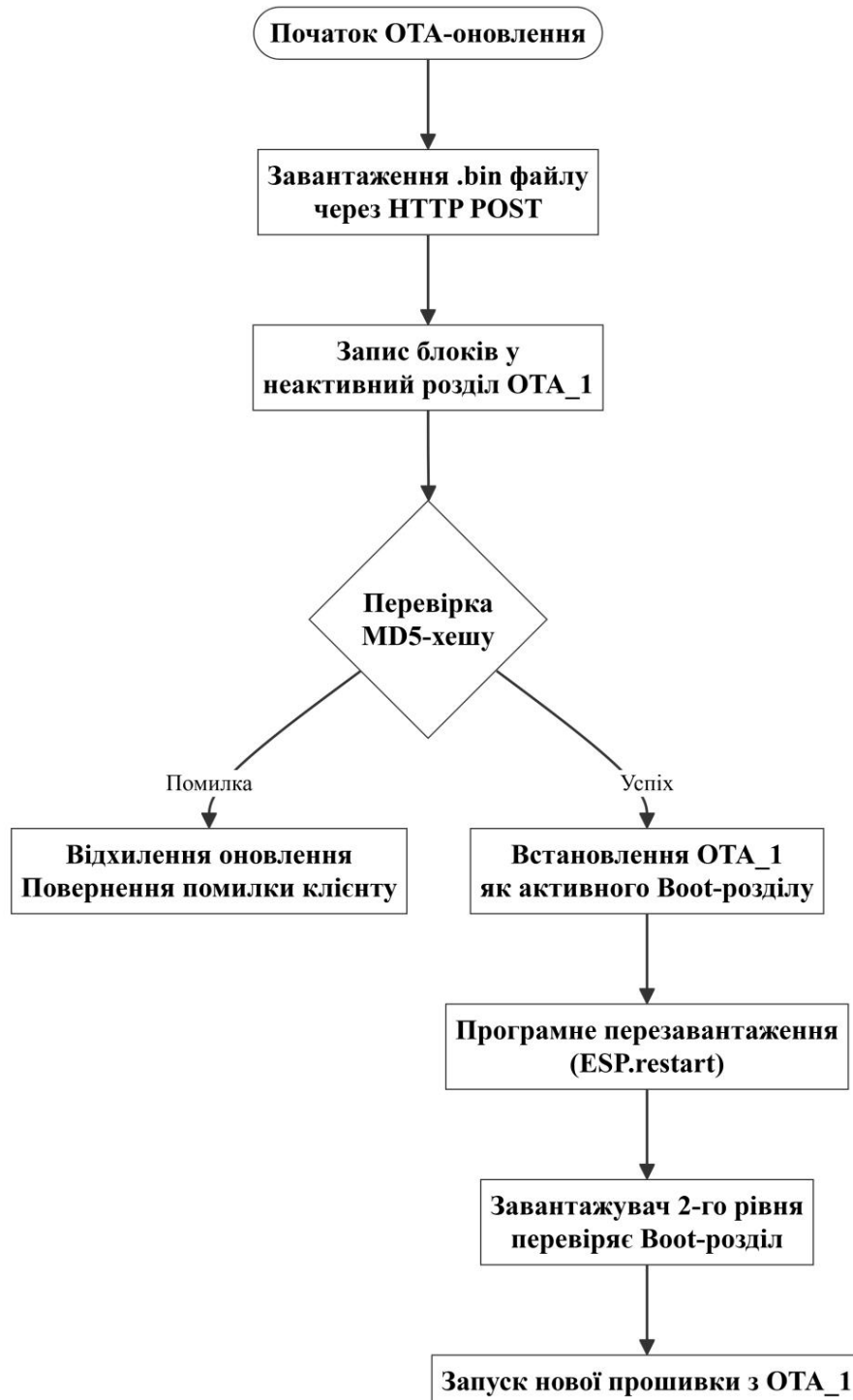


Рис. 4. Блок-схема процесу бездротового оновлення прошивки (OTA) з використанням Dual-Bank розділів

Джерело: розроблено авторами

Лише після повного, 100 % завантаження файлу, завантажувач (2nd Stage Bootloader) вступає в дію. Він розраховує криптографічну хеш-суму (SHA256) завантаженого образу і

порівнює її з цифровим підписом. Якщо байти збігаються (що гарантує цілісність та відсутність мережових помилок передачі), завантажувач міняє вектор завантаження у таблиці розділів і перезавантажує пристрій вже з нового банку пам'яті.

Для забезпечення абсолютної відмовостійкості програмного забезпечення реалізовано апаратний автоматичний відкіт (Hardware Rollback). Навіть якщо нова завантажена мікропрограма синтаксично правильна, але містить логічний критичний баг (наприклад, нескінченний цикл, що викликає спрацювання апаратного сторожового таймера Watchdog, або витік пам'яті), пристрій не «помре». Після трьох поспіль невдалих спроб завантажитися або падінь системи (Panic), Bootloader самостійно маркує новий розділ як дефектний (Corrupted) і миттєво відновлює завантаження з попереднього, гарантовано стабільного розділу пам'яті. Це забезпечує найвищий рівень надійності, необхідний для автономного медичного обладнання.

ВЕБ-інтерфейс та інтерактивний моніторинг. Апаратне забезпечення та математичні алгоритми, описані вище, складають внутрішню, приховану частину системи. Проте головна взаємодія між лікарем-реабілітологом та пацієнтом відбувається через візуальний користувацький веб-інтерфейс.

Перед розробниками постало фундаментальне завдання: як забезпечити динамічне відображення графіків рухів на екрані планшета у реальному часі (60 FPS), не перевантажуючи при цьому слабкий мікроконтролер нескінченними HTTP-запитами? Класичний підхід періодичного опитування (HTTP Long Polling), коли браузер кожні 100 мс відправляє запит «чи є нові дані?», створює гігантські мережові накладні витрати. Кожен HTTP-запит вимагає встановлення нового TCP-з'єднання (Handshake) та містить сотні байт текстових заголовків, що миттєво паралізувало б веб-сервер ESP32 і викликало б неприпустимі затримки екрану.

Тому інтерфейс було розроблено за сучасною архітектурою односторінкового застосунку (Single Page Application, SPA) на базі Vanilla JavaScript (без важких фреймворків типу React). Усі статичні активи інтерфейсу (HTML-розмітка, CSS-стилі, векторна графіка та JS-логіка) заздалегідь жорстко стискаються алгоритмом GZIP. Отримані мініатюрні бінарні архіви зберігаються у LittleFS всередині пристрою. При першому підключенні лікаря, його браузер завантажує цей пакет всього один раз. Після цього, інтерфейс ініціює перемикання комунікації з повільного HTTP на швидкісний протокол WebSocket (RFC 6455). Цей протокол утримує єдиний, постійно відкритий повнодуплексний TCP-тунель між мікроконтролером та браузером. Для забезпечення надійності пристрою в умовах нестабільного Wi-Fi (дисконекти, втрата пакетів), на стороні клієнта імплементовано алгоритм Exponential Backoff. Якщо з'єднання втрачається, інтерфейс не зависає, а автоматично намагається відновити сесію зі зростаючим інтервалом очікування, гарантуючи безперервність реабілітаційного процесу.

Замість того, щоб кодувати цифри у громіздкий текстовий формат JSON, ESP32 серіалізує оновлені кути нахилу, поточний стан FSM-детектора (IDLE/FLEXING/EXTENDING) та лічильники рухів у надзвичайно компактний бінарний формат MessagePack. Крім того, під час вивантаження історії сесій (до 4400 записів), передача єдиного JSON-масиву неминуче викликала б переповнення пам'яті (OOM) на ESP32. Для вирішення цієї проблеми розроблено архітектуру AsyncChunkedResponse: дані серіалізуються у MessagePack і відправляються у браузер фрагментами (чанками). Ці двійкові пакети надходять у WebSocket-канал із фіксованою частотою 20 Гц (замість 100 Гц сирих даних), розвантажуючи Wi-Fi модуль без втрати плавності UI.

Медична панель (Dashboard) лікаря, що працює в браузері, складається з трьох логічних зон.

Перша зона – це живий цифровий осцилограф кута нахилу. Для забезпечення плавних 60 кадрів на секунду, відмальовка графіка відбувається не через створення HTML-елементів

(DOM), а за допомогою апаратно прискореного HTML5 Canvas API з використанням функції requestAnimationFrame.

Друга зона – інтерактивна таблиця історії сесій, яка миттєво підвантажує результати минулих тренувань безпосередньо з SQLite. Критичним технічним розширенням є імплементація експорту медичної статистики. Уся зібрана база даних (кути, швидкість, тривалість, тремор) генерується мікроконтролером у формат CSV із маркером UTF-8 BOM, що забезпечує миттєвий та безшовний імпорт до Microsoft Excel для подальшого статистичного аналізу лікарем. Архітектура бази даних спроектована з урахуванням майбутньої інтеграції у стандарт обміну медичними даними HL7/FHIR.

Третя зона – контрольна панель для калібрування нуля та керування OTA-оновленнями. Зовнішній вигляд розробленого інтерфейсу подано на Рис. 5.

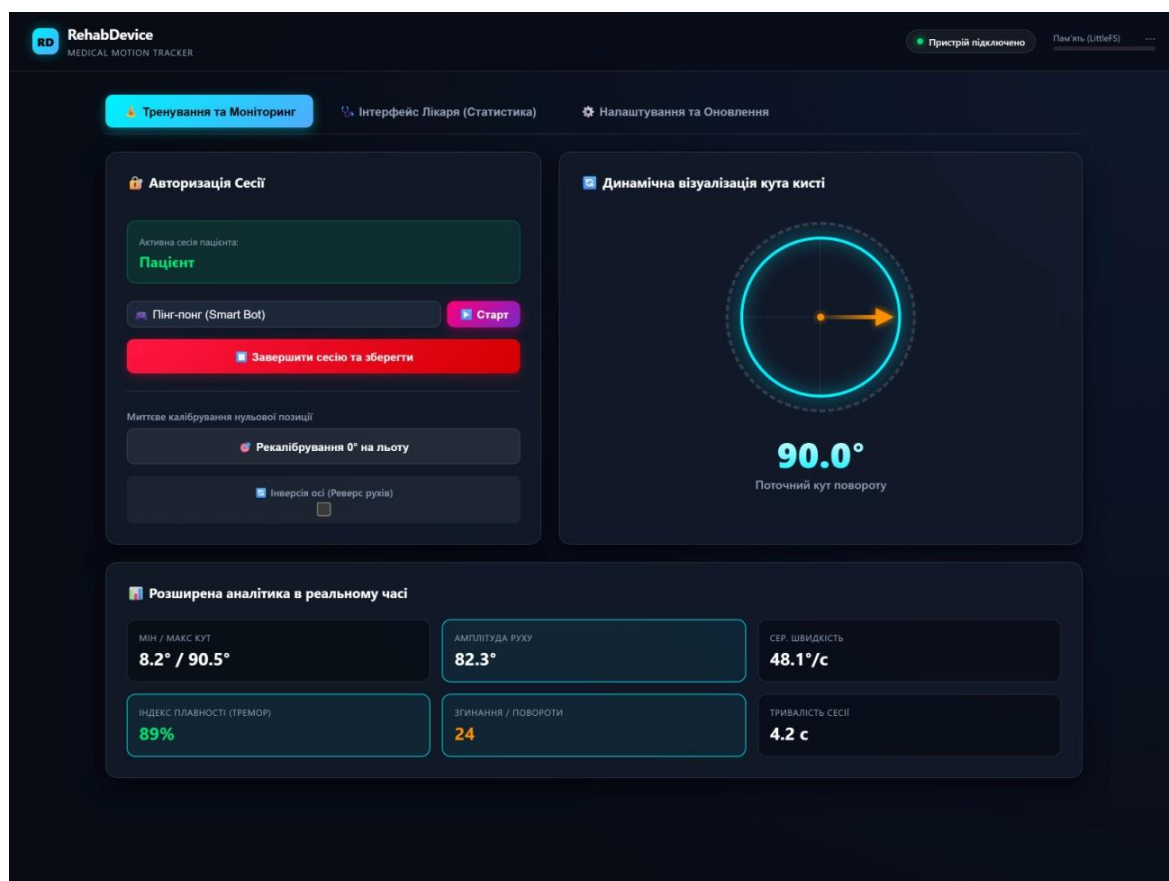


Рис. 5. Медична панель (Режим пацієнта): жива візуалізація кута (20 Гц) та статистика флексій

Джерело: розроблено авторами

Після вибору конкретного пацієнта лікар отримує доступ до деталізованої статистики, що включає цільову норму, індекс плавності рухів та аналіз амплітуди. Завдяки візуалізації (Рис. 7) можна швидко оцінити ефективність поточного курсу терапії.

Нижче на панелі відображається графік історії сесій (Bubble Chart), де кожна точка відповідає окремому тренуванню. Червона, жовта та зелена зони допомагають миттєво ідентифікувати рівень плавності та амплітуди.

Також передбачено експорт усіх даних у формат CSV для глибшого аналізу (рис. 8). Ключовою інновацією програмної частини стала імплементація модуля терапевтичних міні-ігор. Наукові дослідження у сфері нейрореабілітації переконливо доводять, що застосування елементів гейміфікації радикально підвищує рівень прихильності пацієнтів (Compliance) до рутинних, нудних тренувань [12].

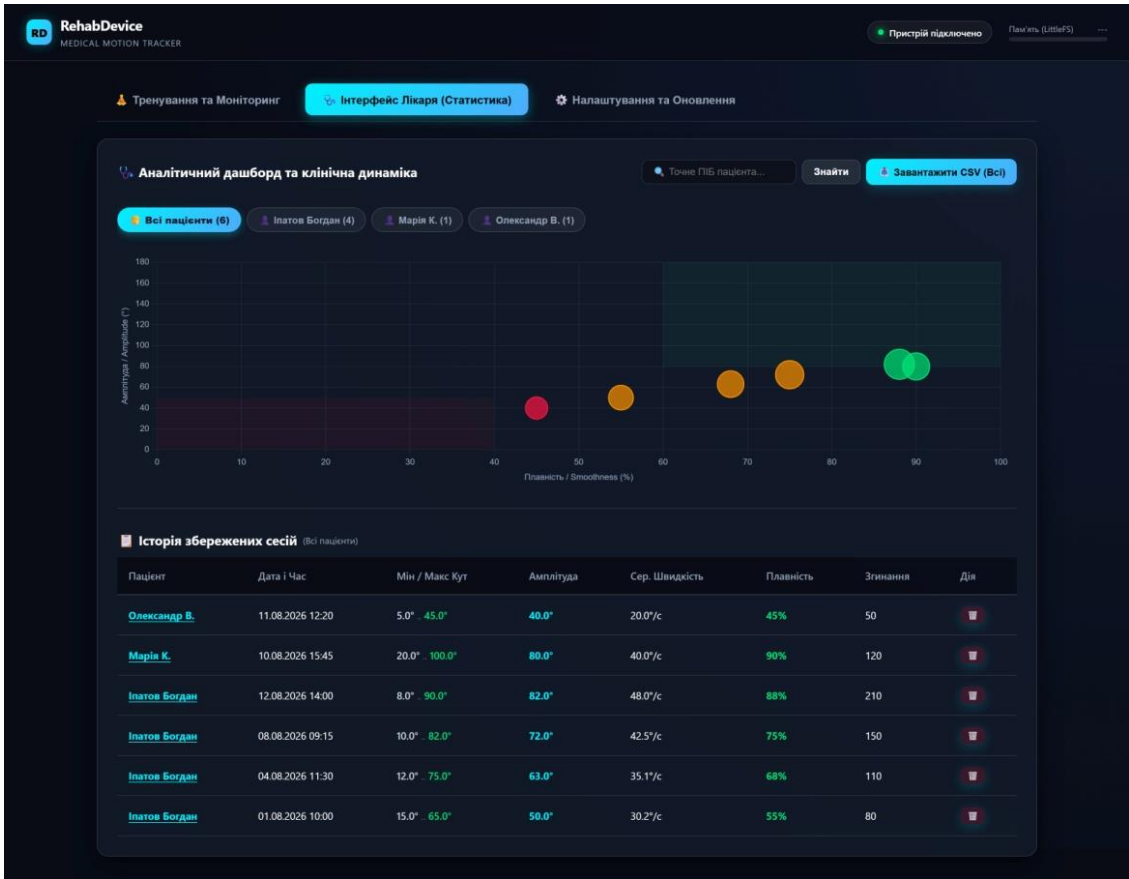


Рис. 6. Аналітичний дашбордин лікаря: Загальний вигляд усіх пацієнтів
Джерело: розроблено авторами

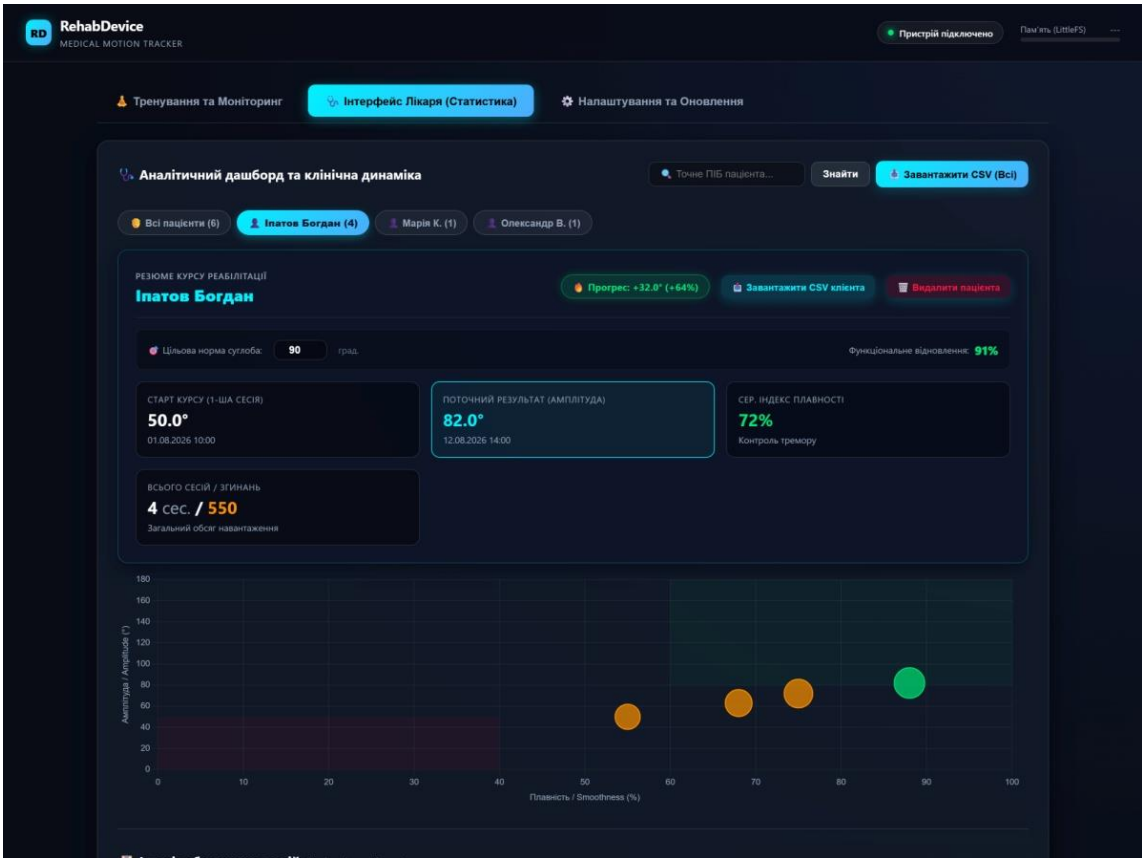


Рис. 7. Деталізоване резюме курсу реабілітації для обраного пацієнта
Джерело: розроблено авторами

У системі RehabDevice пацієнт керує ігровими персонажами не за допомогою комп'ютерної миші чи клавіатури, а виключно фізичними рухами власного травмованого зап'ястя – це є класичним втіленням концепції біологічного зворотного зв'язку.

Наприклад, у розробленій міні-грі «Баланс», мета пацієнта полягає в утриманні віртуальної кульки строго по центру екрану. Для цього пацієнт змушений виконувати надзвичайно мікроскопічні, плавні рухи кистю.

Ця конкретна вправа прицільно тренує нейром'язовий контроль, дрібну моторику та максимізує метрику плавності (показник S), нівелюючи фізіологічний тремор.

Гейміфікація та міні-ігри у терапії. Ключовою інновацією програмної частини RehabDevice стала імплементація потужного модуля терапевтичних міні-ігор. Сучасні дослідження у сфері нейрореабілітації переконливо доводять, що використання елементів гейміфікації радикально підвищує рівень прихильності пацієнтів (Compliance) до рутинних та часто болючих тренувань [12]. У традиційній терапії пацієнт фокусується на відчутті болю або дискомфорту у суглобі. Залучення ж ігрового процесу зміщує когнітивний фокус на досягнення ігрової мети, стимулюючи викид дофаміну, що, своєю чергою, сприяє активнішому формуванню нових нейронних зв'язків [12].

Архітектура системи RehabDevice пропонує безпрецедентну легкість у створенні та додаванні нових міні-ігор. Оскільки пристрій виступає як Edge-сервер і транслює універсальний потік даних (кути, прискорення, джерк) через WebSocket, будь-який веб-розробник може створити нову гру за допомогою стандартних веб-технологій (HTML5 Canvas, Vanilla JavaScript) буквально в одному файлі. Немає необхідності вивчати складні рушії типу Unity або Unreal Engine. Вся ігрова логіка запускається безпосередньо у браузері клієнта (на планшеті або смартфоні), не навантажуючи процесор ESP32.

Наразі у систему інтегровано гру “Smart Ping-Pong”. Пацієнт керує віртуальною ракеткою не за допомогою комп'ютерної миші, а виключно фізичними рухами власного травмованого зап'ястя – класичне втілення концепції біологічного зворотного зв'язку. Унікальною особливістю реалізації є механізм динамічної адаптації складності. Алгоритм гри автоматично підлаштовує розмір ракетки та швидкість м'ячика під поточну дозволену амплітуду (RoM) конкретного пацієнта, яка розраховується аналітичним рушієм. Якщо пацієнт демонструє високий тремор (індекс плавності S падає), гра автоматично уповільнюється, щоб не викликати фрустрацію. Таким чином, грає не людина проти комп'ютера, а комп'ютер адаптується до фізіологічних можливостей людини.

Результати. Для верифікації та валідації розробленого апаратно-програмного комплексу RehabDevice було проведено серію стендових випробувань, а також тестування in vivo на авторі розробки. Комплексна оцінка працездатності системи здійснювалася за трьома критичними метриками: точність вимірювання кінематики (Assigasy), наскрізна мережева затримка (End-to-End Latency) та реальна енергетична автономність.

Оцінка затримки (End-to-End Latency): Цей параметр є важливим для комфорту пацієнта під час біологічного зворотного зв'язку. Часовий інтервал між реальним фізичним рухом кисті в просторі та моментом відмальовки нової позиції графіка у браузері вимірювався за допомогою аналізу уповільненої зйомки високошвидкісною камерою (240 FPS). Завдяки відмові від хмарних серверів, обробці гіперкомплексних чисел безпосередньо на апаратному DMP та використанню бінарного WebSocket-протоколу, середня наскрізна затримка в системі склала всього 14 мс $\pm$ 3 мс. Такий показник не лише задовольняє суворі медичні вимоги (до 30 мс), але й на рівні нейрофізіології виключає можливість виникнення сенсорного дисонансу або кінетозу у пацієнта.

Висновки. У результаті виконання даного дослідження було вирішено актуальну науково-прикладну задачу сучасної телемедицини — створення повністю автономної, високоточної системи об'єктивного моніторингу реабілітації кистьового суглоба RehabDevice. Запропонована та реалізована архітектура Edge Computing на практиці довела свою перевагу над існуючими хмарними IoT-рішеннями (Blynk, AWS). Розміщення веб-

сервера (SoftAP) та бази даних фізично на мікроконтролері забезпечило повну незалежність пристрою від зовнішніх мережевих інфраструктур, гарантувало надійне збереження медичної конфіденційності та дозволило досягти наднизької мережевої затримки (менше 15 мс) у каналі біологічного зворотного зв'язку.

У алгоритмічному аспекті при реалізації «аналітичного рушія» застосовано математичний апарат на базі гіперкомплексних чисел, що дозволило усунути проблему шарнірного замка. Розроблений адаптивний алгоритм на базі скінченного автомата (FSM) із 4-градусним гістерезисом, у поєднанні з математичним інтегруванням кутового джерка (Jerk), дозволили системі не лише безпомилково, зі 100% точністю, рахувати кількість рухів, але й вперше об'єктивно квантифікувати їх плавність та рівень спастичності. З точки зору надійності, використання операційної системи реального часу FreeRTOS (із вирішенням проблеми інверсії пріоритетів), транзакційної файлової системи LittleFS та повноцінної реляційної СУБД SQLite забезпечило промисловий рівень відмовостійкості програмного забезпечення.

Наявність гнучкого односторінкового веб-інтерфейсу (SPA) з інтегрованими терапевтичними міні-іграми перетворює систему RehabDevice з простого вимірювального гоніометра на потужну інформаційну реабілітаційну систему. Перспективи подальших досліджень у цьому напрямку передбачають проведення повномасштабних клінічних випробувань (in vivo) за участю великих груп пацієнтів після інсульту, реалізацію експорту медичної статистики у міжнародні формати HL7/FHIR, а також застосування передових методів машинного навчання (TinyML) безпосередньо на мікроконтролері для автоматичної класифікації на базі нейронних мереж патологічних патернів руху у реальному часі.

Конфлікт інтересів: Автори декларують, що не мають конфлікту інтересів, зокрема фінансового, особистого, авторського чи будь-якого іншого характеру, який міг би вплинути на дослідження, а також на результати, опубліковані в цій статті

Фінансування: Дослідження проводилося без фінансової підтримки

Доступність даних: Усі дані доступні в цифровій або графічній формі в основному тексті рукопису

Технологій штучного інтелекту: Використовувались для оформлення списку літератури та перевірки синтаксису

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “Musculoskeletal conditions”. *World Health Organization*. 2022. – Available from: who.int. – [Accessed: 02.05.2025].
2. Cramer, S. C., Sur, M. R., Dobkin, D. H., et al. “Harnessing neuroplasticity for clinical applications”. *Brain*. 2011; 134 (6): 1591–1609. DOI: <https://doi.org/10.1093/brain/awr030>.
3. Patel, S., Park, H., Bonato, P., et al. “A review of wearable sensors and systems with application in rehabilitation”. *Journal of NeuroEngineering and Rehabilitation*. 2012; 9: 21. DOI: <https://doi.org/10.1186/1743-0003-9-21>.
4. Kodali, R. K., Swamy, G. & Lakshmi, B. “An implementation of IoT for healthcare”. *Proceedings of the 2nd IEEE International Conference on Recent Advances in Information Technology (RAIT)*. 2016. p. 411–416. DOI: <https://doi.org/10.1109/RAICS.2015.7488451>.
5. Khamitov V. M., Boltenkov V. O. & Antoshchuk S. G. “A method for operative sharing confidential images among a group of trusted users”. *Applied Aspects of Information Technology*. 2026; 9 (2): 200–207. DOI: <https://doi.org/10.15276/aaits.09.2026.14>.
6. Shi, W., Cao, J., Zhang Q., et. al. “Edge computing: Vision and challenges”. *IEEE Internet of Things Journal*. 2016; 3 (5): 637–646. DOI: <https://doi.org/10.1109/IIOT.2016.2579198>.
7. Kuipers, J. B. “Quaternions and rotation sequences: A primer with applications to orbits, aerospace, and virtual reality”. *Princeton : Princeton University Press*. 1999. – Available from: <https://doi.org/google.com.ua>. – [Accessed: 02.09.2025].
8. Hssayeni, M. D., Burack M. A., Jimenez-Shahed, J. & Cunha, J. P. “Assessment of response to medication in individuals with Parkinson's disease”. *Medical Engineering & Physics*. 2019; 67: 33–43. DOI: <https://doi.org/10.1016/j.medengphy.2019.03.002>.

9. Bland, J. M. & Altman, D. G. "Statistical methods for assessing agreement between two methods of clinical measurement". *Lancet*. 1986; 327 (8476): 307–310. DOI: <https://doi.org/10.1016/j.ijnurstu.2009.10.001>.
10. Arundale, S. "esp32-arduino-sqlite3: SQLite database for ESP32". 2024. – Available from: <https://github.com> – [Accessed: 02.09.2026].
11. Barry, R. "Mastering the FreeRTOS real time kernel: A hands-on tutorial guide". *Real Time Engineers Ltd*. 2016. – Available from: freertos.org – [Accessed: 02.09.2025].
12. Lohse, K. R., Hilderman, C. G. E., Cheung K. L., et al. "Virtual reality therapy for adults post-stroke: A systematic review and meta-analysis exploring virtual environments and commercial games in therapy". *PLoS ONE*. 2014; 9 (3): e93318. DOI: <https://doi.org/10.1371/journal.pone.0093318>.

Отримано 04.09.2026

Отримано після перегляду 05.10.2026

Прийнято 12.10.2026

DOI: <https://doi.org/10.15276/ict.03.2026.85>

UDC 004.9

Autonomous IoT system for kinematic monitoring of wrist rehabilitation

Hlib S. Statsenko¹⁾

ORCID: <https://0009-0001-2818-6746>; glibstacenko@gmail.com

Bohdan S. Ipatov¹⁾

ORCID: <https://orcid.org/0009-0000-3032-5138>; bogdan.goldenn@gmail.com

Sofia L. Ruban¹⁾

ORCID: <https://orcid.org/0009-0002-0533-124X>; 7919863@stud.op.edu.ua

Dmytro S. Melnik¹⁾

ORCID: <https://orcid.org/0009-0002-2837-5923>; melnik.dm.s@gmail.com

¹⁾ Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

This work addresses the pressing scientific and practical challenge of objectifying the process of medical telemetric rehabilitation. To this end, an innovative autonomous IoT system is proposed for the objective, real-time kinematic monitoring of wrist rehabilitation. The system is based on an autonomous mobile device that employs an edge computing paradigm to provide objective, quantitative monitoring of wrist motor activity. Key implementation requirements were established: the collection of telemetric data—encompassing not only statistical wrist movement parameters but also kinematic indicators (such as physiological and neurological tremor)—and the generation of interactive visual feedback to maintain patient cognitive engagement. The hardware architecture, mathematical modeling, and software implementation for this autonomous mobile device were developed. The device utilizes a high-performance microcontroller and an inertial measurement unit equipped with an integrated digital motion processor. The selection of the microcontroller and necessary sensors is justified. Particular attention is paid to the "analytical engine," which calculates the wrist tilt angle, angular velocity magnitude, and an integral smoothness score in real time. The algorithmic implementation of this engine leverages the powerful mathematical framework of hypercomplex numbers and resolves the "gimbal lock" issue when calculating spatial orientation. To ensure an accurate count of movements amidst physiological tremor, an adaptive detector based on a finite state machine with hysteresis was implemented. To guarantee maximum data reliability, the results of each session are stored locally in a relational database. To enhance patient engagement in the rehabilitation process, the implementation of a robust module featuring therapeutic mini-games has been proposed. A key advantage of the developed autonomous mobile device architecture is its independence from cloud services, ensuring the confidentiality of medical data. A web interface has been implemented to ensure low latency during the visualization of movement trajectories. Future work will focus on field-testing this autonomous IoT system for real-time, objective kinematic monitoring during wrist rehabilitation in clinical settings.

Keywords: Information systems; telemetric rehabilitation; microcontroller; data analysis; embedded systems and IoT devices; biomedical engineering; digital rehabilitation; sensor data processing

ІНФОРМАЦІЯ ПРО АВТОРІВ



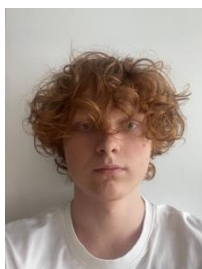
Стаценко Гліб Сергійович - провідний інженер. Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна

Галузь досліджень: вбудовані системи та IoT-пристрої, біомедична інженерія, цифрова реабілітація, обробка сенсорних даних.

Hlib S. Statsenko - Lead Engineer, Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://0009-0001-2818-6746>; glibstacenko@gmail.com

Research field: embedded systems and IoT devices, biomedical engineering, digital rehabilitation, sensor data processing



Іпатов Богдан Сергійович - студент кафедри Інформаційних систем, Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна

Галузь досліджень: вбудовані системи та IoT-пристрої, біомедична інженерія, цифрова реабілітація, обробка сенсорних даних.

Bohdan S. Ipatov - Student. Department of Information Systems. Odesa Polytechnic National University, 1 Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0009-0000-3032-5138>; bogdan.goldenn@gmail.com

Research field: embedded systems and IoT devices, biomedical engineering, digital rehabilitation, sensor data processing



Рубан Софія Леонідівна - студентка кафедри Інформаційних систем, Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна

Галузь досліджень: вбудовані системи та IoT-пристрої, біомедична інженерія, цифрова реабілітація, обробка сенсорних даних.

Sofiia L. Ruban - Student. Department of Information Systems. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0009-0002-0533-124X>; 7919863@stud.op.edu.ua

Research field: embedded systems and IoT devices, biomedical engineering, digital rehabilitation, sensor data processing



Мельник Дмитро Сергійович - аспірант кафедри Інформаційних систем, Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна

Галузь досліджень: вбудовані системи та IoT-пристрої, біомедична інженерія, цифрова реабілітація, інтелектуальний аналіз даних, бази даних

Dmytro S. Melnik - PhD Student. Department of Information Systems. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, Ukraine, 65044

ORCID: <https://orcid.org/0009-0002-2837-5923>; melnik.dm.s@gmail.com

Research field: embedded systems and IoT devices, biomedical engineering, digital rehabilitation, data mining, databases