

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інформаційна безпека та електронні комунікації
(повне найменування факультету)

Кафедра інформаційних технологій електронних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему: ДОСЛІДЖЕННЯ МЕТОДИК РОЗРАХУНКУ, СТВОРЕННЯ
АЛГОРИТМУ РОЗРАХУНКУ МОНІТОРИНГУ КАЛОРІЙ ТА РОЗРОБКА
ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ ЕЛЕКТРИЧНОГО
ВЕЛОСИПЕДУ

Виконав: студент 2 курсу, групи БК-512м

Спеціальності 172 Телекомунікації та
(код і найменування спеціальності)
радіотехніка

Освітня програма (спеціалізація) Інтелектуальні технології мікросистемної
радіоелектронної техніки

АВРАМЕНКО Павло

(ПРИЗВИЩЕ Ім'я)

Керівник ФАРАФОНОВ Олексій

(ПРИЗВИЩЕ Ім'я)

Рецензент КОРОТУН Андрій

(ПРИЗВИЩЕ Ім'я)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Інформаційна безпека та електронні комунікації

Кафедра Інформаційні технології електронних засобів

Ступінь вищої освіти третьій (магістр)

Спеціальність 172 «Телекомунікації та радіотехніка»

(код і найменування)

Освітня програма (спеціалізація) інтелектуальні технології мікросистемної радіоелектронної техніки

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

В.о. Завідувача кафедри ІТЕЗ,

к.т.н. О.Ю. МАЛІЙ

« 18 » серпня 2023 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА СТУДЕНТА

АВРАМЕНКО Павло Валерійович

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) «Дослідження методик розрахунку, створення алгоритму розрахунку моніторингу калорій та розробка програмно-апаратного комплексу для електричного велосипеда»

керівник проєкту (роботи): к.т.н., доцент, ФАРАФОНОВ Олексій Юрійович

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від «21» листопада 2023 року №448

2. Строк подання студентом проєкту (роботи) «27» грудня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне

завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): огляд технології; компоненти та схема пристрою; андроїд

застосунок; науково-дослідний розділ.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових

креслень, кількість слайдів, плакатів) —

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1 Теоретичні відомості та постановка завдання	ФАРАФОНОВ Олексій Доцент	18.08.2023	25.08.2023
2 Огляд предметної області	ФАРАФОНОВ Олексій Доцент	25.08.2023	08.09.2023
3 Методика підрахунку калорій	ФАРАФОНОВ Олексій Доцент	08.09. 2023	29.09.2023
4 Апаратне забезпечення	ФАРАФОНОВ Олексій Доцент	29.09. 2023	06.10.2023
5 Розробка програмного забезпечення	ФАРАФОНОВ Олексій. Доцент	06.10. 2023	17.11.2023
6 Перевірка розробленого програмно-апаратного комплексу	ФАРАФОНОВ Олексій Доцент	17.11. 2023	08.12.2023
Нормоконтролер	ПОСПЕСВА Ірина Старший викладач	08.12. 2023	15.12.2023

7. Дата видачі завдання « 18 » серпня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1 Постановка завдань по роботі	1 тиждень	Розділ 1
2 Пошук наукових робіт та аналогів по темі роботи	2 тижні	Розділ 2
3 Виконання науково-дослідної частини	4–5 тижні	Розділ 3
4 Розробка алгоритму підрахунку спалених калорій	6 тиждень	Пункт 3.5
5 Розробка апаратного забезпечення	7 тижні	Розділ 4
6 Дослідження протоколу обміну даних	8–9 тиждень	Пункт 5.2
7 Розробка програмного забезпечення мікроконтролеру	10–11 тиждень	Розділ 5
8 Розробка андроїд застосунку	12–13 тиждень	Розділ 5
9 Оформлення пояснювальної записки та документів	14–15 тиждень	-
10 Нормоконтроль та рецензування	16–17 тижні	-
11 Захист роботи.	18 тиждень	-

Студент

_____ Павло АВРАМЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Олексій ФАРАФОНОВ
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра: 69 с., 6 табл., 24 рис., 2 дод., 21 джерело.

ЕЛЕКТРИЧНИЙ ВЕЛОСИПЕД, РОЗРАХУНОК СПАЛЕНИХ КАЛОРИЙ, АНДРОЇД ЗАСТОСУНОК ДЛЯ ЕЛЕКТРОВЕЛОСИПЕДУ, ПРОГРАМНИЙ ВЕЛОКОМП'ЮТЕР, КТ-LCD.

Об'єкт дослідження – енергозатрати людиною під час подорожі на електричному велосипеді, протокол обміну даними між контролером велосипеда та LCD дисплеєм.

Ціль роботи – розробка апаратно-програмного комплексу з розширеним функціоналом, що замінить штатний дисплей електричного велосипеда. У розширеному функціоналі приділяється особлива увага підрахунку спалених калорій.

Проведені наступні дослідження:

– дослідження наукових робіт та вивчення вихідного коду онлайн калькуляторів для отримання уточненої формули розрахунку спалених калорій при їзді на електричному велосипеді;

– дослідження протоколу обміну даними між штатним дисплеєм типу «КТ-LCD5» та контролером типу «Kunteng КТ» електричного велосипеда. Для дешифрування пропрієтарного протоколу використовувалися мікроконтролери типу «АТmega 328р», «ESP8266» під'єднані до ліній даних дисплею з виводом інформації на ПК і наступним аналізом отриманих даних при впливі на електричний велосипед.

Наукова новизна досліджень полягає у створенні формули методом кореляційно-регресійного аналізу для розрахунку метаболічного еквіваленту для велосипедиста в залежності від швидкості руху велосипеда. Це дозволяє робити досить точний підрахунок спалених калорій при катанні на велосипеді в залежності від швидкості руху, статі, віку, маси тіла та зросту велосипедиста.

Також розроблений алгоритм підрахунку спалених калорій при їзді на електричному велосипеді з системою допомоги педальним асистентом.

В результаті досліджень протоколу обміну даними між дисплеєм та контролером електричного велосипеда створена бібліотека на мові C++ для простої взаємодії з контролером велосипеда, яка дає можливість налаштувати контролер та приймати інформацію про стан велосипеда у реальному часі. У відкритому доступі аналогічні бібліотеки відсутні.

В результаті роботи створений апаратно-програмний комплекс, що включає мікроконтролерний пристрій малих габаритів з дротовим підключенням до контролера електричного велосипеда та бездротовим підключенням до андроїд смартфона. Розроблений андроїд застосунок, який відображає поточний стан електричного велосипеда (швидкість, пройдену відстань, час, електричне споживання, спалені калорії, тощо) та налаштує характеристики контролера мотор-колеса. Для зменшення ціни пристрою обраний готовий мікроконтролерний модуль типу ESP-01s та стандартний пластиковий корпус. Друкована плата розроблена в САД системі EasyEDA editor, що дає змогу негайного початку виробництва на потужностях JLCPCB (Китай).

Виконана успішна перевірка та порівняння розробленого програмно-апаратного комплексу з існуючими фітнес додатками.

Розроблений програмно-апаратний комплекс зацікавить як існуючих користувачів електричних велосипедів, так і людей, що піклуються про своє здоров'я та підраховують калорії. Наслідком заохочення людей до користування електричними велосипедами з частковою відмовою від ДВЗ з поєднанням із «зеленими» технологіями є збереження довкілля, економія палива, поліпшення здоров'я. Ця тема пов'язана зі статтею [1] «Використання відновлюваних джерел енергії для захисту довкілля».

Результати дослідження спалення калорій при катанні на електричному велосипеді рекомендується використовувати при розробці фітнес-додатків та

велосипедних комп'ютерів, у сфері охорони здоров'я для розробки харчових раціонів та карт фізичних навантажень.

Розроблений апаратно-програмний комплекс рекомендується встановлювати на електричні велосипеди, що підтримують дисплеї КТ-LCD.

Пропонується продовження досліджень у напрямку уточнення алгоритму та формули підрахунку метаболічного еквіваленту. Окрім врахованих параметрів пропонується враховувати підйоми та спуски під час подорожі, каданс при обертанні педалей. Також необхідні додаткові практичні дослідження з виміром метаболічного еквіваленту при різній швидкості їзди та різним кадансом на спусках та підйомах траси.

ЗМІСТ

	с.
Реферат.....	2
Скорочення та умовні позначки	7
Вступ	8
1 Теоретичні відомості та постановка завдання	9
2 Огляд предметної області	12
2.1 Загальний огляд об'єкта дослідження	12
2.2 Огляд існуючих рішень	13
3 Методика підрахунку калорій	18
3.1 Калорія	18
3.2 Типи витрат калорій	18
3.3 Методи розрахунку базових витрат калорій.....	19
3.4 Методика розрахунку додаткових витрат калорій при їзді на велосипеді	22
3.5 Розробка алгоритму підрахунку витрат калорій при подорожі на електричному велосипеді.....	27
4 Апаратне забезпечення.....	29
4.1 Модуль з мікроконтролером.....	29
4.2 Блок живлення.....	30
4.3 Блок-схема апаратного забезпечення	31
4.4 Корпус пристрою	32
4.5 Схема та друкована плата пристрою	33
5 Розробка програмного забезпечення	36
5.1 Середа розробки та мови програмування.....	36
5.2 Протокол КТ-LCD	36
5.3 Розробка бібліотеки КТ-LCD на мові C++.....	40
5.4 Допоміжні бібліотеки	42
5.5 Обмін даними по бездротовому з'єднанню	44

5.6 Інтерфейс користувача	44
5.7 Сценарії JavaScript. Взаємодія з користувачем та мікроконтролером ESP8266	46
5.8 Android застосунок «КТ-LCD».....	47
6 Перевірка розробленого програмно-апаратного комплексу	50
6.1 Обладнання.....	50
6.2 Дорожні умови	50
6.3 Критерії успішності перевірки	51
6.4 Результати перевірки та порівняння з існуючими рішеннями.....	51
Висновки	54
Рекомендації	56
Перелік джерел посилання.....	57
Додаток А Алгоритм підрахунку затрачених калорій	61
Додаток Б Лістинг класу КТ-LCD на мові C++.....	62

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

У цьому звіті вжито такі позначки та скорочення:

C++ – мова програмування;

CAD система – система автоматизованого проєктування (САП або САПР);

IDE (integrated development environment) – комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм;

JSON (JavaScript Object Notation) – текстовий формат обміну даними між комп'ютерами;

MET (metabolic equivalent of task) – метаболічний еквівалент;

PAS (Pedaling Assisted System) – система допоміжного педалювання електричного велосипеда;

Web Socket – протокол, що призначений для обміну інформацією між браузером та вебсервером в режимі реального часу;

БВК – базові витрати калорій людиною у спокої;

ДВК – додаткові витрати калорій людиною під час фізичного навантаження;

СВК – сумарні витрати калорій людиною під час фізичного навантаження.

ВСТУП

В рамках сучасних технологій та практичних застосувань, особлива увага приділяється питанням енергоефективності та здорового способу життя. Дослідження, що присвячені аспектам подорожування на електричних велосипедах, зокрема споживанню енергії під час таких подорожей, виходять за межі стандартних підходів.

Об'єктом дослідження стала методика підрахунку енергозатрат особи під час подорожей на електричному велосипеді. Ця тема має важливе значення для подальшого розвитку фітнес-технологій та електротранспорту. У зв'язку з ростом популярності здорового способу життя можна відзначити збільшення інтересу громадськості до визначення власного рівня фізичної активності. На даний час не існує зручних засобів підрахунку енергозатрат під час подорожей на електричному велосипеді. Популярні мобільні фітнес-застосунки не враховують внесок електромотору при визначенні затраченої енергії.

Метою даної роботи є розробка апаратно-програмного комплексу з розширеним функціоналом, спрямованим на заміну штатного дисплею електричного велосипеда. Основний акцент робиться на розширенні функціоналу з урахуванням підрахунку спалених калорій, що є важливим для більшої точності та контролю під час їзди.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

Підрахунок затрачених калорій є важливою частиною багатьох фітнес-та харчових програм, оскільки допомагає людям керувати своїм ваговим статусом та забезпечує належний рівень енергії для підтримки активного способу життя. Основні аспекти цього підрахунку включають у себе:

- базові витрати калорій, що є кількістю енергії, яку організм витрачає у стані спокою для підтримки базових фізіологічних функцій, таких як дихання, температурна регуляція та робота органів. БВК визначається такими факторами як вага, зріст, вік і стать;

- фізична активність, що являє затрати енергії, які залежать від рівня фізичної активності. Вона може бути розділена на кілька категорій, таких як легка, середня і інтенсивна активність. Також важливо враховувати тривалість тренувань.

Необхідність підрахунку затрачених калорій полягає в допомозі людям у досягненні різних фітнес-цілей, таких як збереження або зниження ваги, набір маси, забезпечення ефективності тренувань.

Багато сучасних фітнес-додатків і трекерів забезпечують зручний спосіб підрахунку затрачених калорій на основі різноманітних параметрів та інформації про фізичну активність. Однак, важливо також пам'ятати, що точність цих розрахунків може варіюватися, і їх слід використовувати як загальні орієнтири, а не абсолютні значення.

Сучасні фітнес-трекери та велокомп'ютери справляються з задачами підрахунку затрачених калорій при їзді на класичному велосипеді. Але на даний час немає жодних зручних засобів та методик вимірювання затрачених калорій при їзді на електровелосипеді з педальним асистентом, який зменшує навантаження на велосипедиста.

Для вирішення проблеми підрахунку калорій необхідно розібратися в базових принципах роботи педального асистенту (далі система PAS), визначити показники для розрахунку та розробити рівняння.

Система PAS є технологічним рішенням, що полегшує педалювання на велосипеді. Основні принципи її роботи включають:

- детектування швидкості обертання педалей за допомогою датчика холу;
- визначення рівня підтримки з врахуванням швидкості обертання педалей та користувацьких налаштувань;
- активація електричного двигуна при детектуванні руху педалей.

Ці принципи сприяють покращенню ефективності педалювання, роблять велосипеди більш доступними для різних груп користувачів і зменшують фізичне навантаження під час подорожі на велосипеді.

Для отримання даних, таких як швидкість руху, активація PAS системи, потужність електродвигуна від контролеру електровелосипеду. необхідно визначити та обрати способи взаємодії з системою.

В Україні розповсюдженні контролери мотор-коліс «Kunteng» та сумісні комп'ютери для електричних велосипедів типу KT-LCD. Найпростіший варіант отримання інформації про стан велосипеду є пряме підключення до існуючих датчиків або паралельне застосування додаткових датчиків, що ускладнює та подорожчує систему. Також розглядається варіант підключення до штатної системи обміну даними велокомп'ютеру та отримання необхідних показників без додаткового обладнання. Для користувача бажана недорога, проста у встановленні та експлуатації система. Отже, необхідно розглянути можливість взаємодії з велосипедним контролером по цифровій шині даних. Для відображення інформації про стан велосипеду та затрачені калорії можна застосувати дисплей з кнопками керування, що буде встановлюватися на рулі. Такий підхід також приведе до суттєвого подорожчання вартості системи. Тому слід обрати альтернативу взаємодії з користувачем та розглянути у якості дисплею смартфон користувача.

Виходячи з вищезазначеного, задача проєкту полягає в розробці апаратно-програмного комплексу, який буде зберігати та обмінюватися даними з велосипедним контролером, візуалізувати отримані та розраховані

дані на дисплеї смартфона користувача. Для з'єднання та передачі інформації на смартфон необхідно застосувати стандартний безпроводний протокол зв'язку.

Проміжними етапами роботи є:

- вибір апаратного обладнання;
- реверс-інжиніринг протоколу KT-LCD;
- розробка методики підрахунку затрачених калорій з використанням системи PAS;
- перевірка розробленої методики на практиці;
- розробка програмного забезпечення для обраного обладнання та смартфона;
- перевірка та наладка отриманого програмно-апаратного комплексу.

2 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Загальний огляд об'єкта дослідження

Моніторинг фізичної активності людини та відстеження її рухової активності протягом дня є важливим елементом сучасних систем здоров'я і фітнесу. За допомогою сучасних технологій, таких як фітнес-трекери та мобільні застосунки, люди можуть відстежувати свою фізичну активність, аналізувати її та приймати інформовані рішення про свій життєвий стиль.

Регулярна фізична активність поліпшує кардіоваскулярну функцію, зміцнює м'язи та кістки, підтримує здорову вагу та покращує настрій.

Сучасні технології, включаючи мобільні застосунки та пристрої, надають зручні інструменти для відстеження фізичної діяльності, зокрема використовуються сенсори для вимірювання рухової активності та алгоритми для обробки отриманих даних.

Важливою функцією систем моніторингу фізичної активності є підрахунок спалених калорій під час тренувань, що дозволяє користувачам оцінювати ефективність тренувань та планувати наступні заняття з урахуванням калорійного балансу.

Незважаючи на переваги мобільних застосунків для моніторингу фізичної активності, важливо враховувати недоліки, такі як точність вимірювань, приватність даних та людський фактор.

Під час їзди на електричному велосипеді з системою педального асистування електричний двигун приймає на себе частину навантаження, допомагаючи велосипедисту їхати швидше та на більші відстані. При чому класичні мобільні застосунки, фітнес-трекери, велосипедні комп'ютери не в змозі відрізнити частки енергії, затрачені велосипедистом чи електродвигуном, без придбання дорогих датчиків для виміру сили педалювання. Вартість таких датчиків перевищує вартість електрокомплекту велосипеда.

Розгляд моніторингу фізичної активності на електричному велосипеді може сприяти створенню унікальних програмних продуктів, які враховують аспекти педального асистування та покращують функціонал інших застосунків, що використовуються для моніторингу активності та здоров'я. Можливість вимірювати калорії під час їзди заохочує людей використовувати електричні велосипеди замість бензинових автомобілів, що, в свою чергу, захищає довколишнє середовище. Використання відновлюваних джерел енергії під час заряджання батареї електричного велосипеда зменшує викиди двоокису вуглецю та інших шкідливих речовин в атмосферу, про що опублікована стаття в [1].

2.2 Огляд існуючих рішень

Перед розробкою нового апаратно-програмного продукту, зокрема мобільного застосунку для вимірювання спалених калорій під час катання на електричному велосипеді з системою PAS, важливо виконати аналіз існуючих рішень.

Перед початком розробки важливо з'ясувати, які аналогічні рішення вже існують на ринку та як вони працюють. Цей аналіз дозволяє уникнути повторення помилок, які можуть бути вже виявлені в інших програмах та використовувати найкращі практики у власній розробці.

Детальний огляд існуючих застосунків також може допомогти виявити можливості для нових інновацій та вдосконалень у власному проєкті. Аналіз показує, які функції та особливості є найбільш цінними для користувачів, і допомагає зосередитись на їх задоволеності.

Враховання потреб та очікувань користувачів є ключовим етапом при аналізі застосунків, тому оцінка вже існуючих рішень дозволить краще зрозуміти, як покращити їх для задоволення потреб користувачів у програмному продукті, який відслідковує витрачені калорії під час катання на електричному велосипеді з системою PAS.

2.2.1 Застосунок SuperCycle

SuperCycle – це програма-велокomp'ютер, яка відстежує та картографує поїздки на велосипеді та відображає у реальному часі дані датчиків GPS, такі як швидкість, відстань, висота, спалені калорії, каденс і потужність. Рейтинг на Google Play 4,7 та більше 100 тис. завантажень. Інтерфейс SuperCycle відображений на рисунку 1.

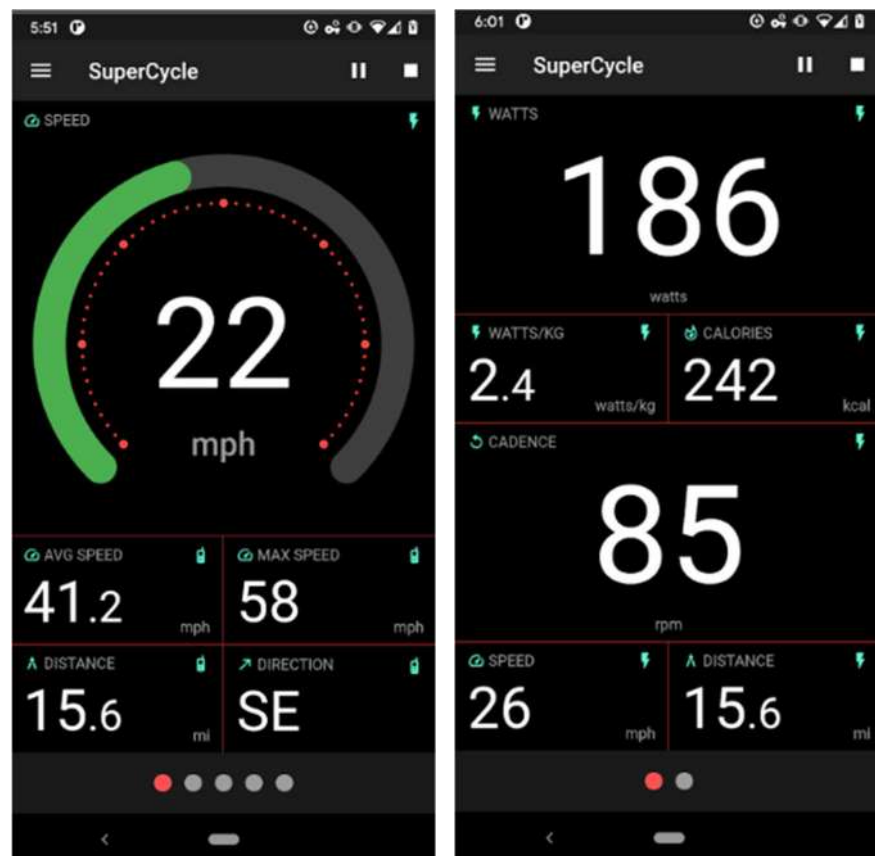


Рисунок 1 – Інтерфейс застосунку SuperCycle

Програма (за ствердженням розробників) може виміряти витрачену енергію з великою точністю за допомогою встановлення вимірювача потужності на велосипед.

Вимірювач потужності являє собою шатуни з вбудованими датчиками навантаження, акселерометрами та гіроскопами. Зовнішній вигляд шатунів показаний на рисунку 2.



Рисунок 2 – Шатуни Shimano Dura-Ace R9100

Суттєвим недоліком такого рішення є ціна датчика навантаження, що складає більше 400 американських доларів.

Розгляд відгуків до застосунку в більшості носить позитивний характер. Деякі користувачі звертають увагу на неточність показань спідометру, підраховані спалені калорії не відповідають дійсності. Іноді припиняється підрахунок одометром. Автори заявляють, що на основі GPS даних розрахунок калорій може бути дуже не точний.

2.2.2 Застосунок Zepp life

Zepp Life популярний додаток для фітнес-пристроїв Xiaomi. Zepp Life відстежує активність, аналізує якість сну та оцінює прогрес тренувань, в тому числі на велосипеді. Оскільки додаток багатофункціональний, розглядаються лише функції моніторингу під час тренування на велосипеді. Зовнішній вигляд інтерфейсу додатку відображено на рисунку 3.

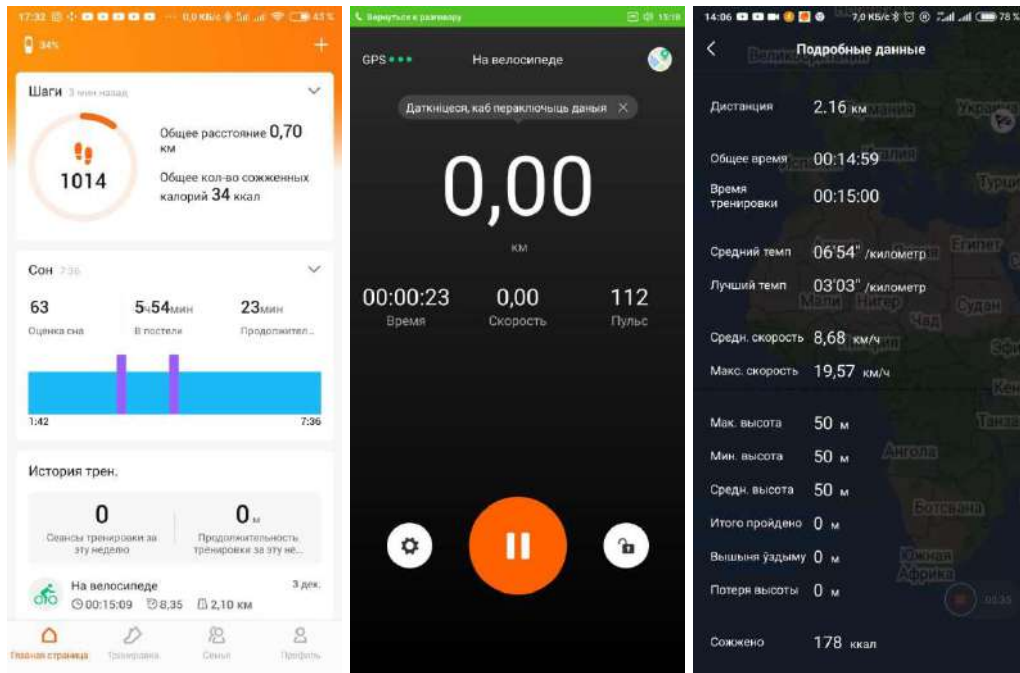


Рисунок 3 – Интерфейс застосунку Zepp Life

Основна перевага Zepp Life полягає у розповсюженості та популярності фітнес-пристроїв Xiaomi в Україні. Також до переваг відноситься картографування тренувальних поїздок, зберігання історії тренувань, зручний інтерфейс.

Користувачі додатку Zepp Life відмічають неточність у підрахунку затрачених калорій, проблеми синхронізації з браслетом, зникнення частини даних тренування при зникненні GPS сигналу.

При розробці пристрою необхідно порівняти показники з Zepp Life під час подорожі електричним велосипедом.

2.2.3 Дисплей Kunteng KT-LCD

LCD дисплей купується окремо та застосовується на електричному велосипеді як бортовий комп'ютер для управління та налаштування контролеру мотор-колеса. Існує сім різновидів LCD дисплеїв, які позначаються різною цифрою наприкінці назви та мають різну форму. За функціоналом усі дисплеї майже однакові. LCD дисплей відображає рівень заряду батареї, швидкість, час поїздки, відстань та контролює систему PAS,

круїз контроль, режим пішоходу, максимальну швидкість. Зовнішній вигляд дисплею LCD5 наведений на рисунку 4.



Рисунок 4 – Дисплей LCD5

До переваг LCD дисплею можна віднести наявність екрану та кнопок управління, зручне кріплення на руль велосипеда, автоматичне вимкнення контролеру при простої, добра видимість у сонячну погоду.

До недоліків LCD дисплею можна віднести незрозуміле меню налаштувань, відсутність лічильника розходу електроенергії та калорій.

При розробці програмно-апаратного комплексу керування електричним велосипедом слід врахувати зазначені переваги і недоліки.

3 МЕТОДИКА ПІДРАХУНКУ КАЛОРІЙ

3.1 Калорія

Калорія [2] – позасистемна одиниця кількості тепла і термодинамічного потенціалу (внутрішньої енергії, ентальпії, вільної енергії), що дорівнює 4,1868 джоуля. Одна кілокалорія дорівнює тисячі калорій. Раніше калорією вважали кількістю тепла, яке витрачається на нагрівання 1 г води на 1 °С.

Калорійність [2] – енергетична цінність харчових продуктів або раціонів харчування, кількість тепла, що його виділяють харчові чи кормові речовини, коли вони окислюються в організмі людини або тварини.

Для експериментального визначення калорійності харчового продукту його спалюють. Теплова енергія, що виділяється під час експерименту підігріває воду. Вимірявши різницю температури води не складно підрахувати кількість затрачених калорій на підігрів. Отримана величина і буде калорійністю.

3.2 Типи витрат калорій

Під час життєдіяльності людини наявні два типи витрат калорій, а саме основний (базовий) та додатковий.

Базові витрати калорій [3] (англ. Basal Metabolic Rate) – це кількість калорій, які організм витрачає на підтримку основних його функцій під час спокою при температурному комфорті. До базового обміну відноситься дихання, кровообіг, робота органів та регуляція температури тіла. Базова витрата калорій залежить від кількох факторів, таких як вік, стать, вага та склад тіла.

Додаткові витрати калорій виникають під час фізичної активності, такої як ходьба, розумова діяльність, підйоми по сходах, спорт та ін. Додаткові витрати калорій варіюються у кожної особи й залежать від рівня активності, тривалості фізичних навантажень та фізичних характеристик людини. Розрізнення між основними та додатковими витратами калорій є ключовим

для ефективного контролю харчування та фізичних навантажень. Визначення й моніторинг цих витрат сприяють кращому розумінню й регулюванню загального енергетичного балансу організму. Ці знання важливі для підтримання здорової ваги, вибору правильного харчування і підтримки загального здоров'я.

3.3 Методи розрахунку базових витрат калорій

Базові витрати калорій (далі БВК) вимірюються прямою або непрямую калориметрією [3]. Метод прямої калориметрії визначає кількість тепла, виділеного організмом за певний проміжок часу. Метод непрямой калориметрії базується на розрахунку даних кількості спожитого кисню і виділеного вуглекислого газу.

Підрахунок БВК необхідний для встановлення калорійного балансу потреби організму та розробці харчового раціону. Існують декілька методів підрахунку БВК на основі параметрів організму, таких як зріст, вага, вік, стать. Ці методи дають орієнтовні значення через індивідуальні особливості людей.

Формули для підрахунку, що набули відомості, наступні:

- формула Гарріса-Бенедикта;
- формула Оуена;
- формула Міффліна-Сен-Жеора.

3.3.1 Формула Гарріса-Бенедикта

Формула Гарріса-Бенедикта – це метод, який оцінює БВК у людини на основі параметрів її тіла. Формула Гарріса-Бенедикта включає наступні вхідні дані: стать, вага, зріст, вік.

Оригінальні рівняння опубліковані у 1918–1919 роках [4], [5]. Вони переглянуті у 1984 році Розою і Шізгалом [6] та в 1990 році Міффліном і Сент-Жеором [7] для поліпшення точності.

Рівняння Гарріса-Бенедикта для чоловіків відображено у формулі (1).

$$\text{БВК}_q = 13,397 \cdot m + 4,799 \cdot h - 5,677 \cdot a + 88,363, \quad (1)$$

де БВК – базові витрати калорій, ккал/добу;

m – вага, кг;

h – зріст, см;

a – вік, роки.

Рівняння розрахунку базових витрат калорій у ккал/добу Гарріса–Бенедикта для жінок наведено у формулі (2).

$$\text{БВК}_ж = 9,247 \cdot m + 3,098 \cdot h - 4,33 \cdot a + 447,593 \quad (2)$$

Наведені рівняння не враховують склад тіла. М'язи та жир витрачають різну кількість калорій. Тому людина зі спортивною фігурою та людина з ожирінням з однаковим зростом/вагою фактично матимуть різний БВК.

3.3.2 Формула Оуена

Формули Оуена – це спрощений метод, який оцінює БВК у людини на основі ваги тіла. У 1986 році опублікована формула для жінок [8]. У дослідженні приймали участь 44 жінки віком від 18 до 65 років, масою тіла від 43 до 171 кг. У 1987 році опублікована формула для чоловіків [9]. У дослідженні приймали участь 60 чоловіків віком від 18 до 82 років, масою тіла від 60 до 171 кг.

Рівняння Оуена для чоловіків наведено у формулі (3).

$$\text{БВК}_q = 879 + 10,2 \cdot m, \quad (3)$$

де БВК – базові витрати калорій, ккал/добу;

m – вага, кг.

Рівняння розрахунку базових витрат калорій у ккал/добу Оуена для жінок наведено у формулі (4).

$$\text{БВК}_{\text{ж}} = 795 + 7,18 \cdot m, \quad (4)$$

У різних дослідженнях формули Оуена оцінені як ненадійні. Розрахункові результати часто відхиляються від фактичних замірів витрат калорій.

3.3.3 Формула Міффіна-Сен-Жеора

Формула Міффіна-Сен-Жеора – один з найпоширеніших методів підрахунку БВК. Метод розроблений Гаррі Міффіном та Рудольфом Сен-Жеором у 1990 році.

Рівняння Міффіна-Сен-Жеора для чоловіків наведено у формулі (5).

$$\text{БВК}_{\text{ч}} = 10 \cdot m + 6,25 \cdot h - 5 \cdot a + 5, \quad (5)$$

де БВК – базові витрати калорій, ккал/добу;

m – вага, кг;

h – зріст, см;

a – вік, роки.

Рівняння розрахунку базових витрат калорій у ккал/добу Міффіна-Сен-Жеора для жінок наведено у формулі (6).

$$\text{БВК}_{\text{ж}} = 10 \cdot m + 6,25 \cdot h - 5 \cdot a - 161 \quad (6)$$

Формула Міффіна-Сен-Жеора розроблена на основі значних наукових досліджень та застосовується у медицині для оцінки базової енергетичної потреби пацієнтів. Формула є стандартизованим методом, що застосовується професіоналами з харчування та дієтології.

3.3.4 Вибір формули для розрахунку БВК

Серед розглянутих формул неточними є формули (3) та (4). Формули (1) та (2) застарілі порівняно з формулами (5) та (6). Отже, формули

Міффіна-Сен-Жеора є найпоширенішими і найточнішими методами розрахунку базової витрати калорій.

На основі вищезазначеного вирішено використовувати формули Міффіна-Сен-Жеора для розрахунку базових витрати калорій.

Перевагами обраного методу розрахунку базової витрати калорій є точність розрахунків. Формула Міффіна-Сен-Жеора враховує стать, вагу, зріст та вік, що робить її більш індивідуалізованою та точною у порівнянні із розглянутими формулами.

3.4 Методика розрахунку додаткових витрат калорій при їзді на велосипеді

Результати дослідження витрати калорій та показників велосипедистів під час руху [10] при катанні на електричному велосипеді з педальним асистентом у режимах «NO», «ECO» та «POW» наведені на рисунку 5.

	NO		ECO		POW	
	Mean	SD	Mean	SD	Mean	SD
Intensity (MET)	6.1*	1.6	5.7	1.2	5.2*	1.4
Total energy consumption (kcal)	108.1*†	18.4	94.3†‡	14.9	77.5*‡	20.0
HR (bpm)	123.8*†	23.2	116.2†	22.4	112.4*	22.9
HR (%HR _{max})	73.9*†	14.5	69.3†	13.5	67.1*	14.1
Cycling speed (km·h ⁻¹)	19.6*†	2.4	21.1†‡	2.2	23.4*‡	1.7
Cycling time (min:s)	13:38*†	1:49	12:45†‡	1:26	11:33*‡	0:58
Power output (W)	118.2*†	30.9	101.8†‡	24.8	94.2*‡	29.2
PR (rpm)	55.0*†	5.6	51.5†	6.1	49.1*	7.2
RPE (range, 6–20)	13.1*†	1.6	10.8†‡	1.0	8.3*‡	1.8
CS (range, 1–5)	1.7*†	0.7	1.1†	0.3	1.0*	0.2

* Significant difference between NO and POW.

† Significant difference between NO and ECO.

‡ Significant difference between ECO and POW.

Physiological variables measured during cycling tests in NO, ECO, and POW mode (mean and SD).

Рисунок 5

Згідно дослідження [10] затрати калорій чоловіків при вимкненому моторі і русі від педалей за 13 хв. 38 с. склали 108,1 ккал, що еквівалентно 475 ккал/год. Отже, розроблена методика для підрахунку витрачених калорій під час їзди на електричному велосипеді має давати схожі результати.

Згідно досліджень [11] витрати енергії за хвилину під час їзди на електричному велосипеді становили від 4,9 до 6,5 ккал/хв. (від 294 до 390 ккал/год.). В таблиці 1 наведені результати десяти досліджень [11] показників MET еквіваленту при їзді на електричному велосипеді та при ходьбі.

Таблиця 1 – Показники MET еквіваленту при їзді на електричному велосипеді та при ходьбі

Дослідження [11]	E-bike HA	E-bike LA	E-bike NA	CB	Walking
Bernsten, 2017	-	6,9	-	8,4	-
Gojanovic, 2011	6,1	7,3	-	8,2	6,5
Hansen, 2017	6,0	6,6	-	6,4	-
Langford, 2017	5,1	-	-	5,8	4,5
La Salle, 2017	-	-	8,3	-	8,5
Louis, 2012	4,2	5,6	6,5	-	-
Peterman, 2016	4,9	-	-	-	-
Simons, 2009	5,2	5,7	6,1		-
Sperlich, 2012	5,2	-	-	7,1	-
Theurel, 2011	5,6	-	-	5,9	-
Примітка 1. HA – максимальна допомога PAS, LA – мінімальна допомога PAS; NA – без допомоги; CB – звичайний велосипед. При дослідженнях використовувалися методи з вимірюванням затраченого кисню, виділеного діоксиду вуглецю та замірами серцевого ритму.					

MET еквівалент можна перерахувати в затрати енергії у ккал/добу за формулою (7).

$$E = \text{БВК} \cdot \text{MET}, \quad (7)$$

Аналіз даних досліджень, наведених в таблиці 1, дозволяє зробити наступні висновки:

- MET еквівалент при їзді на звичайному велосипеді та на електричному велосипеді з вимкненою системою PAS схожі за значеннями;
- при вмиканні системи PAS, MET еквівалент зменшується;
- при збільшенні допомоги системою PAS, MET еквівалент зменшується;
- при їзді на велосипеді з будь-яким рівнем підтримки системою PAS, MET еквівалент більший, ніж при ходьбі;
- дослідження не враховують підйоми, спуски та каданс обертання педалей, що унеможливує їх врахування при підрахунку спалених калорій та потребує додаткових досліджень.

Отже, витрати калорій на електричному велосипеді більші ніж при ходьбі за однаковий проміжок часу.

Розглянуто декілька онлайн калькуляторів для підрахунку калорій при їзді на велосипеді. Виявилось, що більшість з них не враховують дані велосипедиста, такі як стать, вік та зріст. Переглянуто вихідний код онлайн калькулятора «livelong» [12]. При розрахунку витрат калорій при їзді на звичайному велосипеді у калькуляторі застосована формула (8).

$$E = 0,3 \cdot m \cdot V, \quad (8)$$

де E – загальні витрати калорій, ккал/добу;

m – вага велосипедиста, кг;

V – швидкість велосипеда, км/год.

Для більш точного підрахунку витрат калорій за допомогою кореляційно-регресійного аналізу визначена залежність MET еквіваленту від швидкості велосипеда з використанням даних Американської ради з фізичних вправ [13]. Результати аналізу наведені на рисунку 6.

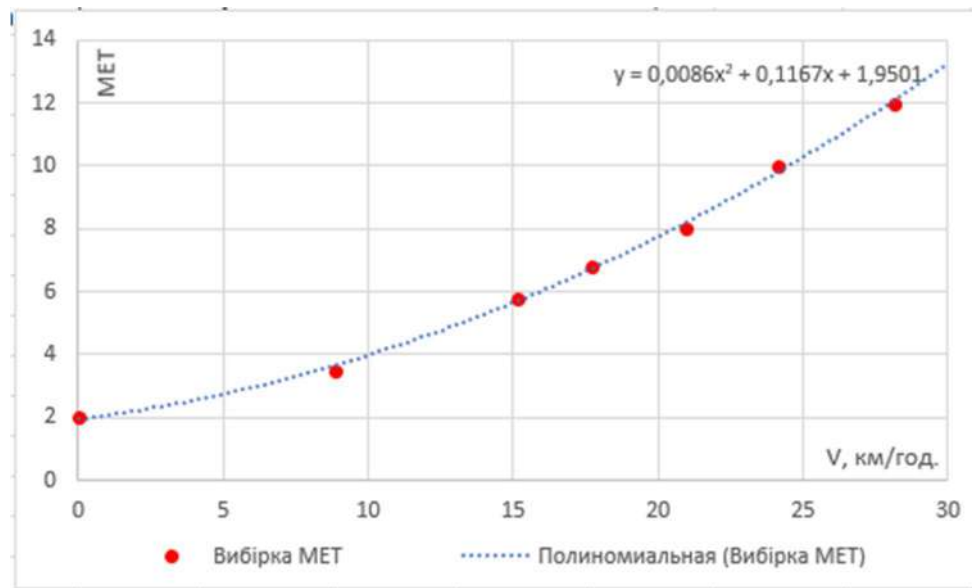


Рисунок 6 – Результати кореляційно-регресійного аналізу визначення залежності MET еквіваленту від швидкості велосипеда

Отримана формула для підрахунку MET еквіваленту при катанні на велосипеді представлена нижче:

$$\text{MET} = 0,0086 \cdot V^2 + 0,1167 \cdot V + 1,95, \quad (9)$$

де MET – метаболічний еквівалент;

V – швидкість велосипеда, км/год.

Кількість спалених калорій при подорожі велосипедом визначається за формулою (7) із використанням розрахованого MET за формулою (9) та БВК, розрахованими за формулою (5) для чоловіків або (6) для жінок.

Для перевірки виведеної формули (9) розрахуємо витрати калорій за параметрами велосипедистів, як у дослідженні [10], що відображені на рисунку 5.

Вихідні дані розрахунку:

- очікуваний результат 475 ккал/год.;
- стать чоловіча;
- вік 52,2 р.;

- зріст 173,3 см;
- вага 73,6 кг;
- швидкість руху 19,6 км/год.

Розрахунок за формулами (5), (9), (7):

$$\text{БВК}_q = (10 \cdot 73,6 + 6,25 \cdot 173,3 - 5 \cdot 52,2 + 5) = 1563 \text{ (ккал/добу.)}$$

$$\text{MET} = 0,0086 \cdot 19,6^2 + 0,1167 \cdot 19,6 + 1,95 = 7,54$$

$$E = (1563 \cdot 7,54)/24 = 491 \text{ (ккал/год.)}$$

Отже, для заданих параметрів результат співпав з очікуваним з різницею у 3,37 %. Різниця пояснюється фізіологічними особливостями різних людей, що брали участь в дослідженнях та похибками вимірювань.

Порівняння розрахованих витрачених калорій при різних швидкостях по формулам (5), (9) та (7) з розрахунком на онлайн калькуляторі «livelong» [12] зображено на рисунку 7.

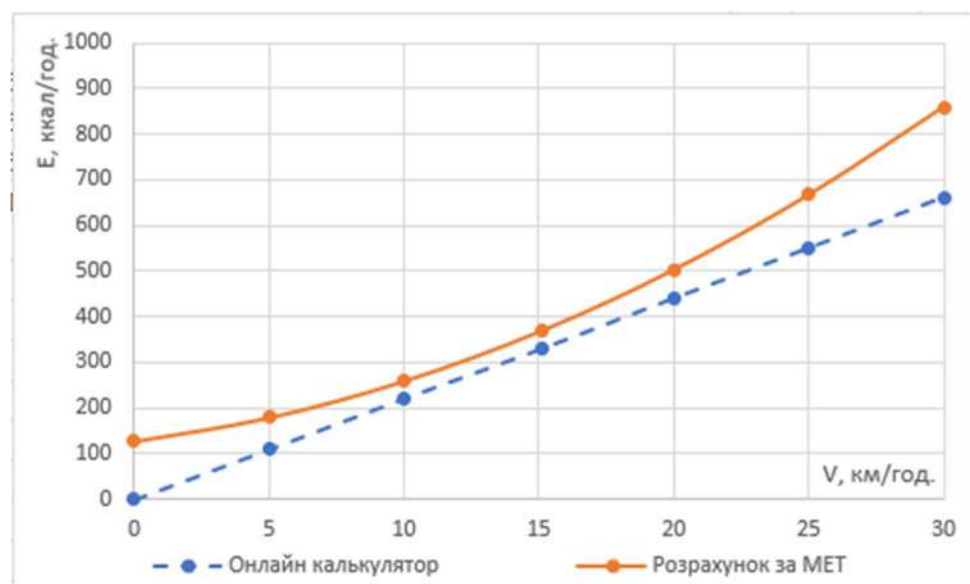


Рисунок 7

Виходячи з порівняння (див. рисунок 7) можна зробити висновок, що онлайн-калькулятор розраховує витрати калорій за спрощеною лінійною формулою, яка не враховує базові витрати на малих швидкостях та опір вітру на великих. Невеликий приріст витрачених калорій на середніх швидкостях пояснюється врахуванням більшої кількості вхідних параметрів при розрахунку за MET.

Отже, досить точно підрахувати спалені калорії при подорожі велосипедом можна за допомогою формул (5), (6), (7) із використанням розрахованого MET за формулою (9).

3.5 Розробка алгоритму підрахунку витрат калорій при подорожі на електричному велосипеді

Для розробки алгоритму необхідно визначити вхідні дані, що використовуються у формулах (5), (6), (7), (9) та вибрати додаткові параметри, такі як відліки вимірювань, різні рівні допомоги системи PAS.

Вхідні дані:

- поточна швидкість;
- поточний час;
- параметри велосипедиста (вік, зріст, вага, стать);
- поточний рівень допомоги системою PAS у діапазоні (0-1);
- формули для обчислення калорій (5), (6), (7), (9).

Розроблений алгоритм підрахунку витрат калорій зображений на рисунку А.1 у додатку А.

На рисунку 8 представлена функція на мові C++, що повертає миттєве значення підрахованих кілокалорій на годину та має вхідні параметри велосипедиста, швидкості руху та відсотку допомоги системи PAS.

```
float calculateCalories(const User &user, float velocity, float pasLevel)
{
    // базові витрати калорій людиною у спокої за добу розраховуються один раз:
    static const float BMR = (10.0 * user.weight + 6.25 * user.height - 5.0 *
user.age + 5.0 * !user.sex - 161.0 * user.sex);
    // Метаболічний еквівалент:
    float MET = 0.0086 * velocity * velocity + 0.1167 * velocity + 1.95;
    //ДВК за добу з урахуванням PAS:
    float DVK = BMR * (MET-1) * pasLevel;
    // Спалені калорії на добу:
    float energy = BMR + DVK;
    return energy / 24; // на годину
}
```

Рисунок 8

Отже, у розділі були розглянуті існуючі та запропоновані нові формули розрахунку затрат енергії велосипедистом під час руху, запропонований алгоритм та код на мові C++ підрахунку затрачених калорій під час руху на електричному велосипеді з системою педального асистента.

4 АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Модуль з мікроконтролером

Для розробки обраний модуль типу ESP-01S на базі мікроконтролера ESP8266 як такий, що є доступним, має низьку ціну (близько 2\$), наявні необхідні інтерфейси UART, WIFI, має малі габарити (14x24 мм), низьке електроспоживання та підтримується сучасними IDE. Наявність вбудованої антени та флеш пам'яті на модулі спрощує розробку. Зовнішній вигляд модуля ESP-01S в масштабі 2:1 зображений на рисунку 9.

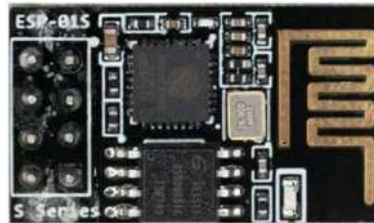


Рисунок 9

Основні характеристики модуля ESP-01S наведені в таблиці 2.

Таблиця 2 – Основні характеристики модуля ESP-01S

Стандарти WIFI	802.11 b / g / n
Вбудований стек	TCP / IP
Обсяг FLASH, кБ	1024
Напруга живлення, В	3,3
Струм живлення	60-215мА
Інтерфейси GPIO, UART	два GPIO, UART,
Мікроконтролер	ESP8266, 32 біт
Тактова частота, МГц	80 або 160
Оперативна пам'ять, кБ	80
Робоча температура, °С	від мінус 40 до плюс 125
Габаритні розміри (Д×Ш×В), мм	24,2×14,2×10

Тестування та налагодження програмного забезпечення відбувалося на модулі NODEMCU, який відрізняється від ESP-01S наявністю вбудованого програматора з USB інтерфейсом та іншою розводкою друкованої плати. Суттєвих відмінностей модулі не мають.

4.2 Блок живлення

Для живлення модуля мікроконтролера необхідна постійна напруга 3,3 В з силою струму до 0,3А. Вхідна напруга блоку живлення має бути вищою за напругу повністю зарядженого акумулятора велосипеда 54,6 В. Блок живлення має бути з високим ККД для зберігання заряду акумулятора.

Заданим критеріям відповідає DC-DC перетворювач, побудований на спеціалізованій мікросхемі типу LM2596HVS-3.3 з фіксованою вихідною напругою 3,3 В, вхідною напругою від 4,5 В до 60 В та має ККД до 79 %. Застосування обраної мікросхеми спрощує електричну схему та зменшує кількість додаткових радіоелементів. Підключення LM2596HVS-3.3, рекомендоване виробником [15], наведено на рисунку 10.

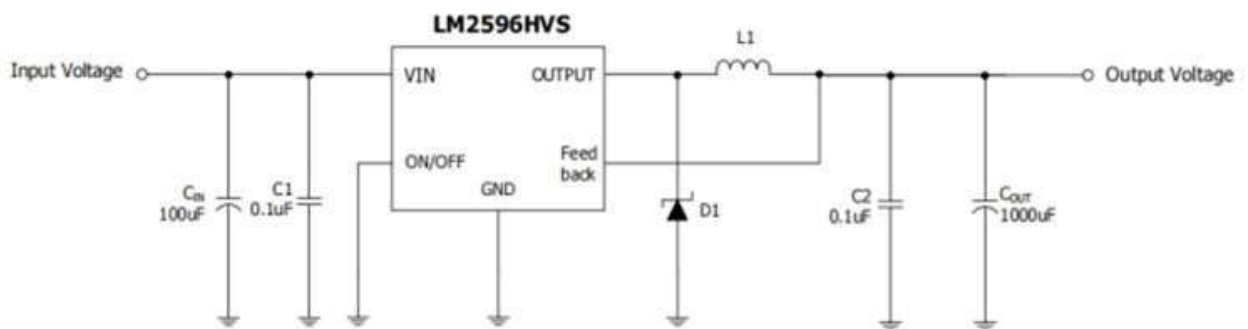


Рисунок 10

Для тестування можна застосувати готовий модуль, вартість якого складає близько 1\$. Зовнішній вигляд модуля DC-DC перетворювача зображений на рисунку 11.

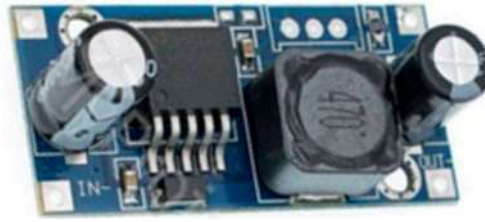


Рисунок 11

4.3 Блок-схема апаратного забезпечення

На рисунку 12 представлена блок-схема з'єднань апаратного забезпечення.

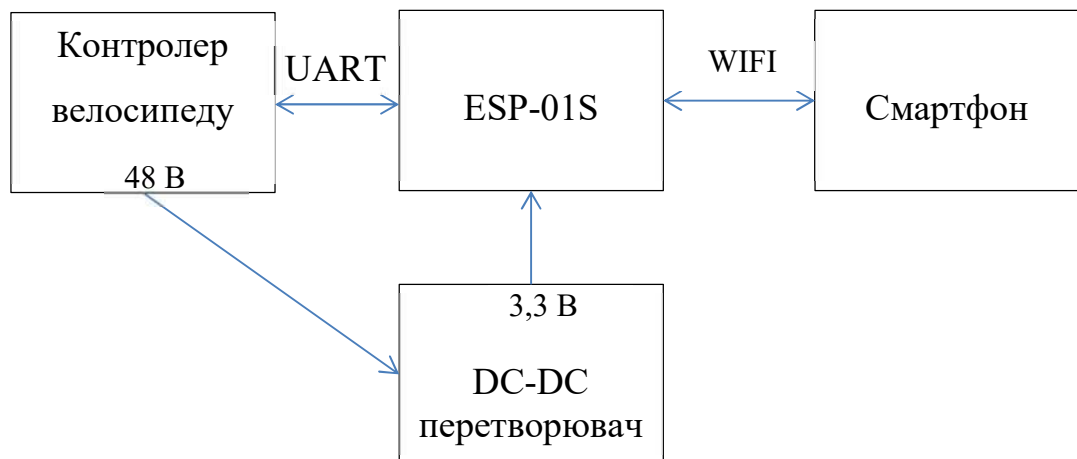


Рисунок 12

На рисунку 13 зображений стандартний роз'єм для підключення дисплею до контролеру електричного велосипеда та призначення контактів.

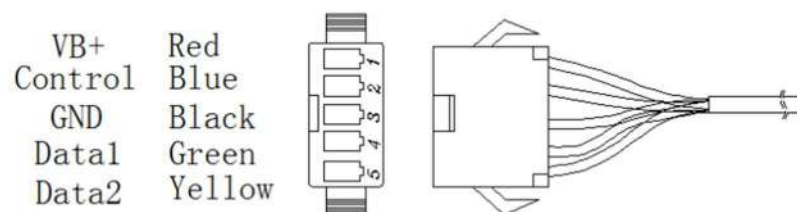


Рисунок 13

Замість дисплею підключається мікроконтролерний модуль. Контакт «TX» мікроконтролеру підключається до контакту «Data1», контакт «RX» через резистивний дільник (для зменшення 5 В до 3.3 В) – до «Data2», контакт «GND» – до «GND». DC-DC перетворювач приєднується до контактів «VB+» та «GND». Контакти ключа запалювання підключаються до «Control» та «VB+».

4.4 Корпус пристрою

Для зменшення собівартості готового пристрою необхідно знизити вартість виготовлення корпусу. Варіант розроблення прес-форм для виготовлення корпусу не розглядається через його велику вартість. Залишається варіант виготовлення корпусу на 3D принтері, що займає багато часу і має вигляд кустарного виробництва, або обрати існуючий корпус.

Обрано існуючий пластиковий корпус з розмірами 37,6x18,4x53,4 мм вартістю вісім гривень. Зовнішній вигляд корпусу наведено на рисунку 14.



Рисунок 14

Внутрішній розмір корпусу для встановлення друкованої плати становить 11,2 x 34 x 50 мм. Кріпити корпус до рами велосипеда пропонується за допомогою пластикових стяжок за сам корпус та дроти, що від нього відходять.

4.5 Схема та друкована плата пристрою

Для дизайну друкованої плати була розроблена схема у спеціалізованій онлайн CAD системі EasyEDA editor [18], яка дозволяє обрати компоненти, в автоматичному та ручному режимах спроектувати друковану плату та маски до них. Вигляд схеми пристрою в EasyEDA editor зображено на рисунку 15.

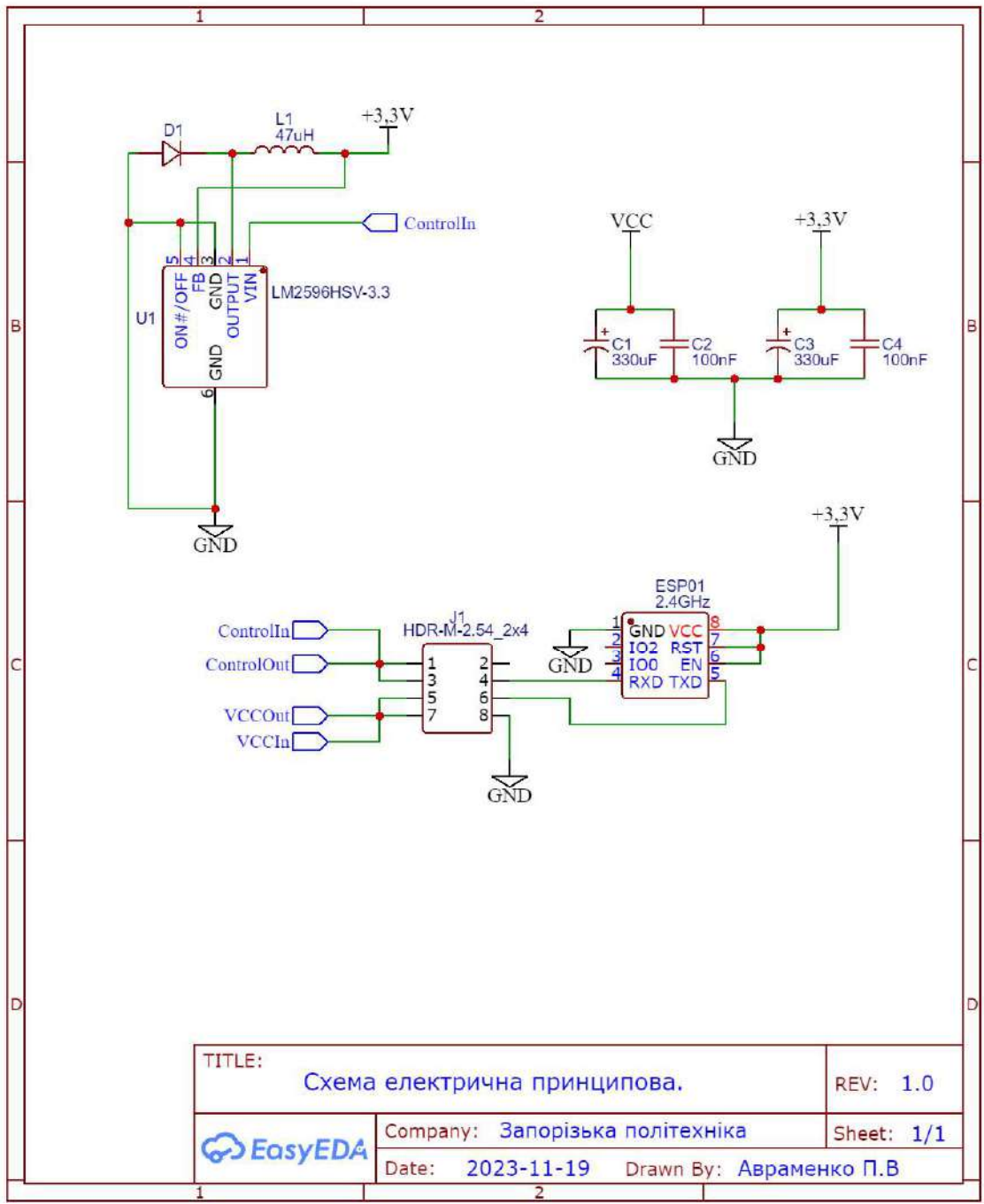


Рисунок 15

3D модель зовнішнього вигляду розробленої друкованої плати представлений на рисунку 16.

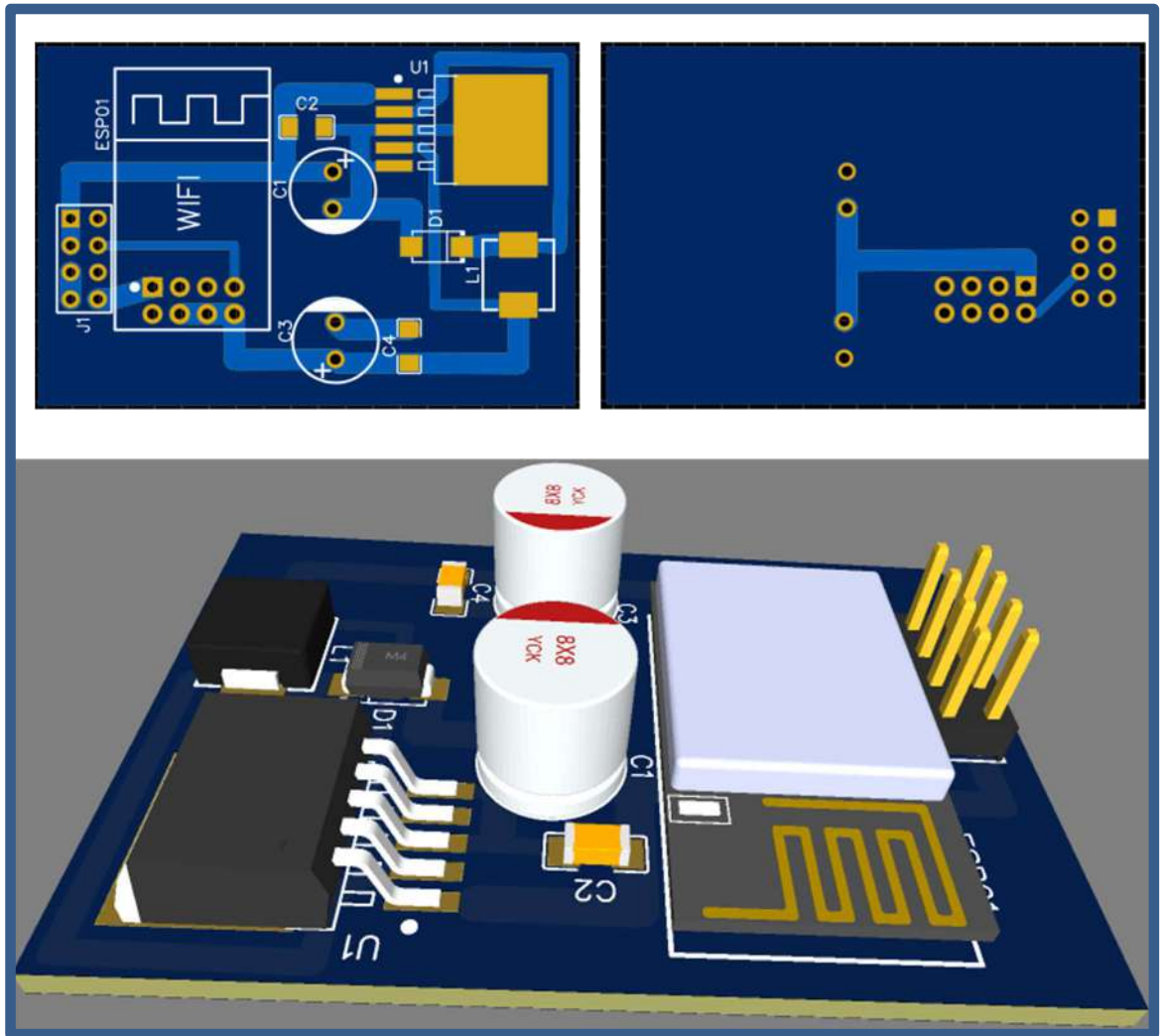
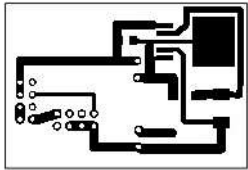
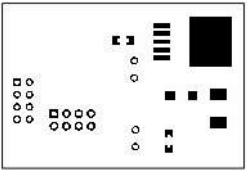
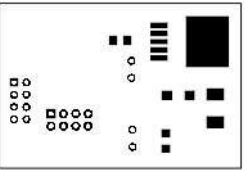
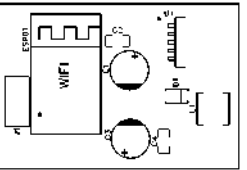
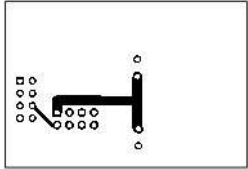
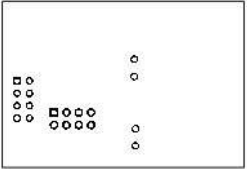
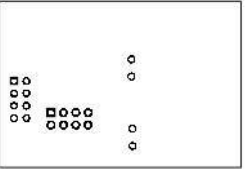


Рисунок 16

Вигляд розводки друкованої плати та масок, необхідних для виготовлення плати, отриманих в EasyEDA editor, у масштабі 1:1 зведений до таблиці 3.

Таблиця 3 – Макет друкованої плати

Шар	Розводка плати	Маска плати	Маска для паяльної пасти	Шовкографія
Верхній				
Нижній				Відсутня

Отримані файли в САД системі EasyEDA editor збережені у хмарному сховищі і можуть бути використані для замовлення готових друкованих плат з установленими компонентами у JLCPCB. Таким чином, виробництво друкованих плат може починатися негайно.

5 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Середа розробки та мови програмування

Для розробки програмного забезпечення мікроконтролера обрана об'єктно-орієнтована мова програмування C++.

Для розробки інтерфейсу користувача обрана html розмітка зі стилями css. Для взаємодії та оновлення інтерфейсу обраний JavaScript. Конвертація html в мобільний додаток виконана на онлайн платформі «WEBINTOAPP».

У якості єдиного IDE для розробки на всіх мовах програмування обрано «Visual Studio Code», яка є сучасним стандартом розробки. Для компілювання та завантаження програми до мікроконтролеру обране розширення PlatformIO, що засновано в Україні і являє собою потужний кросплатформний редактор вихідного коду.

5.2 Протокол KT-LCD

Протокол KT-LCD не задокументовано у відкритих джерелах. Інформація щодо його роботи знайдена на форумах, де ентузіасти діляться своїми дослідженнями. Знайдено дві теми [16], [17] у яких обговорюється складові протоколу. Інформація не повна та невпорядкована. Відомо, що KT-LCD базується на інтерфейсі UART, який працює на швидкості 9600 бод.

Знята осцилограма ліній даних між контролером велосипеда та дисплеєм за допомогою двоканального осцилографа типу ISDS205A. Результати осцилографування наведені на рисунку 17.

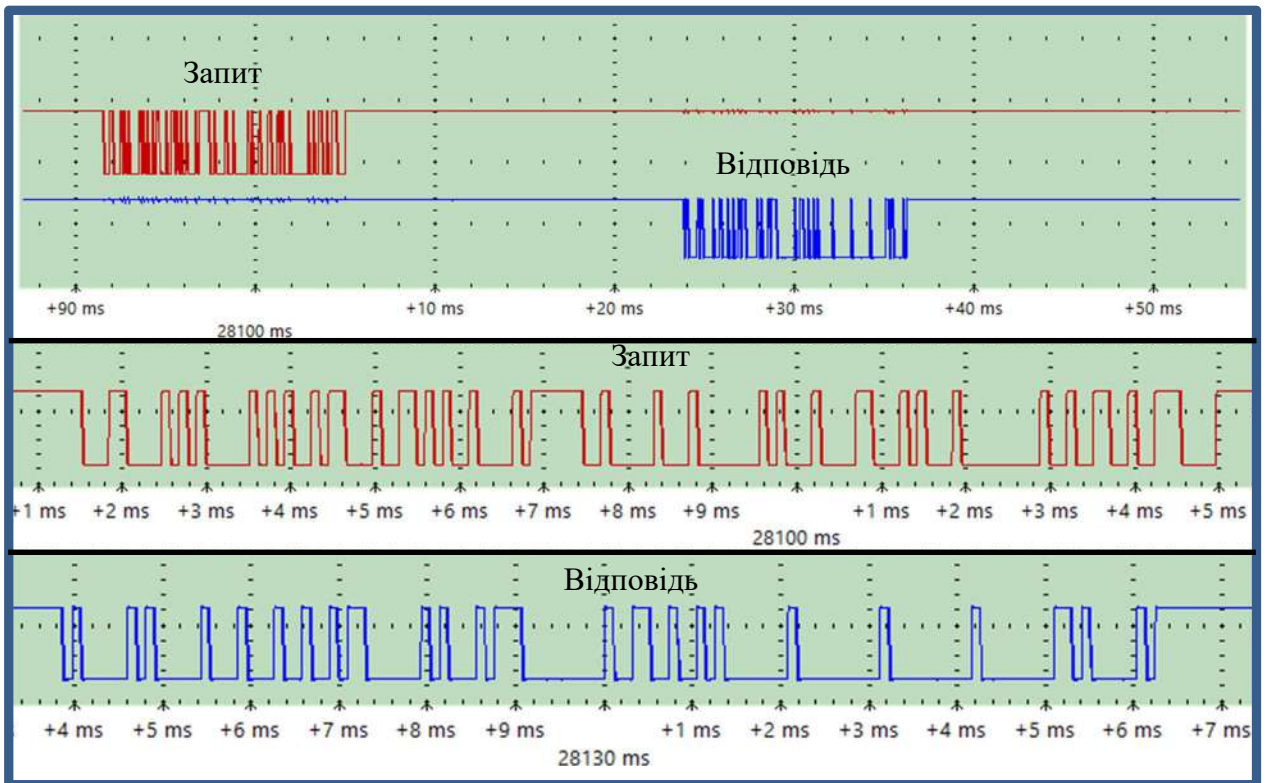


Рисунок 17 – Осцилограма запиту до контролера велосипеда та його відповідь дисплею під час поїздки

Виявилося, що з частотою 10 Гц дисплей посилає до контролера велосипеда масив із 13 байт, в якому містяться поточні налаштування, такі як розмір колеса, ліміт швидкості, ліміт потужності, напруга вимкнення, налаштування поведінки контролера від елементів керування, контрольна сума тощо. При отриманні такого масиву контролер надсилає до дисплея свій поточний стан з параметрами, такими як період обертання колеса, потужність, температура мотора, стан органів керування, рівень заряду батареї, контрольну суму.

Проведено дослідження протоколу, для чого була під'єднана плата Arduino pro nano до ліній контролеру велосипеда Data1 та Data2 з під'єднаним штатним LCD-дисплеєм. Зчитані дані, які передаються по лініях. Приклад отриманих даних від контролера зведено до таблиці 4, де кожна строчка

відповідає одному отриманому масиву типу `uint8_t` в десятковій системі числення.

Таблиця 4

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
65	16	36	14	145	0	106	2	2	0	128	65
65	16	36	0	232	0	30	2	1	0	128	65
65	16	36	0	232	0	30	2	1	0	128	65
65	16	36	0	233	0	31	2	1	0	128	65
65	16	36	0	233	0	31	2	1	0	128	65
65	16	36	0	232	0	30	2	1	0	128	65
65	16	36	0	232	0	30	2	1	0	128	65
65	8	36	0	225	0	226	17	1	0	1	62
65	8	36	0	217	0	133	17	1	0	1	97

Дослідним шляхом при аналізі змін у байтах масиву під час впливу на органи керування велосипеду та зіставлення наявної інформації по протоколу КТ-LCD встановлені відповідності байтів таким впливам.

Отже, протокол КТ-LCD є двонапрямний. У якості майстра виступає дисплей, на запит якого відповідає контролер.

Заповнення масиву запиту до контролеру велосипеду виконує розроблений код C++, зображений на рисунку 18.

```

BYTE[0] = parameter.P5_BatteryMonitorModes;
BYTE[1] = (parameter.pasLevel & 0x07);
bitWrite(BYTE[1], 6, parameter.walkMode);
bitWrite(BYTE[1], 7, parameter.frontLight);
BYTE[2] = ((parameter.maxSpeedLimit - 10) << 3 & 0xF8) |
          (SizeHEX >> 2 & 0x07);
BYTE[3] = parameter.P1_MotorFeature;
BYTE[4] = (SizeHEX << 6 & 0xC0) |
          (parameter.P2_WheelSetSpeedPulse & 0x07) |
          (parameter.P3_PasControlMode << 3 & 0x08) |
          (parameter.P4_ThrottleActiveMode << 4 & 0x10);
bitWrite(BYTE[4], 5, bitRead(parameter.maxSpeedLimit - 10, 5));
BYTE[6] = (parameter.C1_PasSensorSensitivity << 3 & 0x38) |
          (parameter.C2_MotorPhase & 0x07);
BYTE[7] = (parameter.C5_ControllerMaxCurrent & 0x0F) |
          (parameter.C14_PasAdjustment << 5 & 0x60);
BYTE[8] = ((parameter.C4_ThrottleFunction + 1) << 5 & 0xE0) |
          (parameter.C12_ControllerLowestVoltage & 0x07) |
          (parameter.cruise << 4 & 0x10);
BYTE[9] = parameter.C4_SpeedLimitValueOfThrottle & 0x7F;
BYTE[10] = (parameter.C13_ControllerAbsBraking << 2 & 0x1C);
BYTE[11] = parameter.C4_PercentageValueOfTheFirstGearSpeed & 0x3F;
BYTE[12] = 0x0E;
BYTE[5] = BYTE[0] ^ BYTE[1] ^ BYTE[2] ^ BYTE[3] ^ BYTE[4] ^ BYTE[6] ^
          BYTE[7] ^ BYTE[8] ^ BYTE[9] ^ BYTE[10] ^ BYTE[11] ^ BYTE[12] ^ 2;

```

Примітка 2. Назви параметрів відповідають назвам, наведеним в інструкції по експлуатації дисплею KT-LCD [14]. Змінна «SizeHEX» зберігає код розміру колеса.

Рисунок 18 – Заповнення масиву запиту до контролеру велосипеда

Отриманий масив від контролеру велосипеда розкладається по змінним кодом C++, що зображений на рисунку 18.

```

batteryLevel = BYTE[1];
ratedVoltage = BYTE[2];
speedPeriod = (BYTE[3] << 8) | BYTE[4];
error = BYTE[5];
throttle = !bitRead(BYTE[7], 0);
cruise = bitRead(BYTE[7], 3);
assistant = bitRead(BYTE[7], 4);
brake = bitRead(BYTE[7], 5);
power = BYTE[8] * 8;
temperature = (int8_t)BYTE[9] + 15;

```

Рисунок 19 – Обробка отриманого масиву від контролера

5.3 Розробка бібліотеки KT-LCD на мові C++

Після визначення параметрів протоколу була створена бібліотека (клас) для зручної взаємодії з контролером велосипеда.

Клас призначений для отримання та передачі інформації по протоколу KT-LCD до контролеру мотор-колеса електротранспорту. Лістинг класу наведено в додатку Б. Клас може використовуватися з багатьма видами електротранспорту, такими як велосипеди, самокати, моноколеса, що підтримують протокол KT-LCD.

Суть класу в кодуванні інформації та налаштувань, які зрозумілі людині в байтові масиви, які відповідають протоколу KT-LCD та передаються по UART.

При створенні об'єкту класу створюються:

- публічна структура `ReceiveDataStruct` `receiveData` у стеку, що зберігає отримані параметри стану від контролеру мотор-колеса. Структура має публічний метод `uint8_t batteryPercent()`, що перетворює коди контролера у зрозумілі людині відсотки заряду батареї;

– публічна структура `SettingsStruct settings` у стеку, що зберігає налаштування користувача. Назви параметрів структури відповідають назвам параметрів інструкції на дисплей KT-LCD [14]. Структура має публічні методи `float onePeriodDistance()` та `uint8_t wheelRimDiameterHEX()`, що повертають дистанцію в метрах за один оберт колеса та код діаметра колеса у відповідності до протоколу KT-LCD;

– масив `transferBuffer[]` на кучі, що зберігає дані з користувацькими налаштуваннями для передачі в контролер мотор-колеса;

– масив `receiveBuffer[]` на кучі, що зберігає дані, отримані у відповідь з контролеру мотор-колеса.

Клас KT-LCD має два конструктори:

– `KTLCD_Display(SettingsStruct *settings_variable_to_remove)`.

Конструктор приймає структуру з налаштуваннями контролеру;

– `KTLCD_Display()`. Конструктор за замовчуванням. Налаштування необхідно задати пізніше у структурі `SettingsStruct settings`.

Деструктор видаляє з кучі створені масиви `transferBuffer[]` та `receiveBuffer[]`.

Публічні методи класу KT-LCD:

– `void sendSettingsToController()` призначений для відправки налаштувань на контролер мотор-колеса. Після спрацювання методу контролер надсилає відповідь з поточним станом.

– `bool dataAvailable()` повертає `true` при отриманні даних з контролеру мотор-колеса та оновлює публічну структуру `ReceiveDataStruct receiveData`, з якої можна зчитувати поточний стан контролера.

Приватні методи класу KT-LCD:

– `bool getReceivedBufferFromController()` викликає приватний метод `SerialRead(receiveBuffer, RECEIVE_BUFFER_SIZE)` і при успішності його виконання викликає приватний метод `getDataFromReceiveBuffer(receiveBuffer, RECEIVE_BUFFER_SIZE)`;

- `void fillTransferBuffer()` заповнює масив `transferBuffer[]` відповідно до налаштувань `SettingsStruct settings`;
- `uint8_t XOR(uint8_t *buffer, size_t bufferSize, uint8_t placeXor)` обчислює контрольну суму вхідного масиву пропускаючи елемент масиву з номером `placeXor`;
- `bool SerialRead(uint8_t *buffer, size_t bufferSize)` зчитує масив з буферу UART. Перевіряє, чи всі елементи масиву одночасно не дорівнюють нулю (виникає при відсутності живлення контролера мотор-колеса) та викликає метод `uint8_t XOR(uint8_t *buffer, size_t bufferSize, uint8_t placeXor)`;
- `void getDataFromReceiveBuffer(uint8_t *buffer, size_t bufferSize = RECEIVE_BUFFER_SIZE)` заповнює структуру `ReceiveDataStruct receiveData` на основі масиву, отриманого від контролера.
- `void sendBufferToController(uint8_t *buffer, size_t bufferSize = TRANSFER_BUFFER_SIZE)` відправляє масив з налаштуваннями по UART до контролеру мотор-колеса.

Для роботи класу необхідно виконати попереднє налаштування UART командою `Serial.begin(9600)`.

5.4 Допоміжні бібліотеки

Для зручного підрахунку та зберігання отриманих параметрів від контролеру велосипеда розроблені бібліотеки «TimeBasedCounter», «AverageValue», «SaveTemplateDataToLFS».

5.4.1 Бібліотека «TimeBasedCounter»

Бібліотека призначена для підрахунку пройденої відстані, витраченого заряду батареї, витрачених калорій велосипедистом. При створенні екземпляру класу бібліотека запам'ятовує поточний час. Регулярно викликається функція (не рідше ніж раз на 6 секунд) `void measure(float currentValue)`, яка отримує поточну швидкість / потужність / калорійність і на основі різниці часу з попереднім викликом робить підрахунки.

Функція `float getResult()` повертає значення пройденого шляху / витраченого заряду / витрачених калорій з моменту створення екземпляру класу або з моменту останнього виклику функції встановлення початкової величини `void reset(float startValue = 0)`.

5.4.2 Бібліотека «AverageValue»

Бібліотека призначена для підрахунку середнього арифметичного та максимального значення вхідної величини. Застосовується для підрахунку середнього та максимального значення швидкості. Для роботи бібліотеки з заданою періодичністю має викликатися функція `void measure(float currentValue)` для передачі поточної швидкості руху. Бібліотека зберігає у змінних `float resultAverage, maxValue` середню та максимально виміряну швидкість руху.

5.4.3 Бібліотека «SaveTemplateDataToLFS»

Бібліотека призначена для зручного зберігання інформації будь-якого типу у файловій системі флеш накопичувача, вбудованого в модуль мікроконтролера ESP8266. Для використання бібліотеки необхідно створити екземпляр шаблонного класу, передавши йому тип даних, що будуть зберігатися, розташування та ім'я файлу, в якому будуть зберігатися дані. Для ініціалізації бібліотеки необхідно викликати конструктор.

Приклад

```
int a;
```

```
SaveTemplateDataToLFS<int> DATAint(a, "/int.bin").
```

Після виконання цієї команди змінна «а» прийме значення, що було раніше збережене.

Для збереження нових даних виконується команда `writeToFile`.

Приклад

```
DATAint.writeToFile(a).
```

Замість типу даних «int» можна застосувати будь-який інший, наприклад структуру з багатьма змінними та масивами даних.

Бібліотека застосовується для зберігання даних одометрів, лічильників енергії, налаштувань велосипеда.

5.5 Обмін даними по бездротовому з'єднанню

Для обміну даними використовується мережа WIFI. Мікроконтролер запускає вебсервер та створює точку доступу, до якої автоматично під'єднується смартфон. Мікроконтролерний модуль у своїй пам'яті містить вебсторінки інтерфейсів користувача, що робить можливим перегляд інформації та введення налаштувань за допомогою будь-якого WIFI пристрою з встановленим браузером, що підтримує html5.

Спроба обмінюватися даними за допомогою POST запитів дала незадовільні результати через затримки при втраті пакетів даних. Замість POST запитів застосований Web Socket, що надав змогу оновлювати дані про стан велосипеда через WIFI з'єднання з частотою 10 гц. Для встановлення з'єднання через Web Socket на мікроконтролері застосована стороння бібліотека «WebSockets.h», що має ліцензію «free software». Застосовані алгоритми відновлення з'єднання Web Socket при збоях передачі.

Формат даних, що передаються між мікроконтролером та смартфоном JSON. Цей формат базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних, використовується переважно для передавання структурованої інформації через мережу.

5.6 Інтерфейс користувача

Інтерфейс користувача створений у програмі CorelDRAW як векторне зображення SVG. Кожному інтерактивному елементу надана зрозуміла назва для подальшої роботи з цим елементом у JavaScript. Інтерфейс налаштувань створений як розмітка html5 із застосуванням css стилів і має вигляд таблиці з параметрами. Зовнішній вигляд інтерфейсів наведений на рисунку 20.



Рисунок 20 – Скріншоти інтерфейсу розробленого застосунка

Головний інтерфейс складається з чотирьох основних полів:

- годинник/секундомір;
- спідометр з вбудованими одометрами, лампами стану велосипеду;
- кнопки швидких налаштувань;
- таблиці показників.

Лампи спідометра загоряються при ввімкненні світла, виникненні несправності, включенні круїз контролю, перевищенні температури мотору, натисканні на гальма та при розряді акумулятору. Лампи світла і несправності інтерактивні. При натисканні на лампу світла контролеру велосипеду

подається команда ввімкнення фари. При горінні лампи несправності натискання на неї відкриє вікно з інформацією про несправність.

Одометри також є інтерактивними для можливості змінити загальний пробіг велосипеда та для обнулення добового одометру.

Інтерфейс налаштувань містить опис налаштувань з рекомендаціями та поля для введення даних. Налаштування поділені на три розділи :

- параметри «Р»;
- параметри «С»;
- параметри велосипедиста.

Параметри «Р» та «С» відповідають параметрам, наданим в інструкції користувача на ЖК дисплей типу KT-LCD [14] та зберігають у собі такі дані, як інформацію про фізичні дані мотор-колеса, типи датчиків, поведінку контролеру, алгоритми запуску мотору, обмеження швидкості та потужності. Параметри велосипедиста, такі як вага, зріст, вік, необхідні для правильного підрахунку витрачених калорій.

Збереження налаштувань відбувається при натисканні кнопки «Зберегти», яка знаходиться в кінці сторінки налаштувань. Після збереження налаштувань у флеш пам'яті мікроконтролера ESP8266, відкривається вікно з підтвердженням, що налаштування збережено.

5.7 Сценарії JavaScript. Взаємодія з користувачем та мікроконтролером ESP8266

JavaScript (далі JS) в проєкті використовується для створення веб- та мобільного застосунку для надання їм інтерактивності, передачі та отримання даних з мікроконтролера ESP8266.

В головному інтерфейсі використовуються наступні файли сценаріїв JS:
– variables.js. Містить структуру зі змінними, величини яких отримуються від мікроконтролера ESP8266, та константу часу оновлення відображення інтерфейсу;

- `buttons.js`. Сценарій оброблює події при натисканні кнопок користувачем;
- `table.js`. Сценарій кожні 100 мс читає дані з `variables.js` та заповнює ними таблицю з інформацією про поточний стан велосипеда;
- `corelBugs.js`. Сценарій виправляє неточності відображення інтерфейсу через часткову несумісність `svg` графіки в `corelDraw` та браузері;
- `alert.js`. Сценарій відображає короткі повідомлення для підтвердження дій користувача, таких як скидання лічильників. Також сценарій відображає інформацію про помилки в контролері чи колесі велосипеда при натисканні на лампу «Check»;
- `scoreboard.js`. Сценарій кожні 100 мс читає дані з `variables.js` та оновлює стан спідометра з усіма його лампами та елементами;
- `stopwatch.js`. Сценарій відповідає за відображення поточного часу, а при натисканні кнопок біля нього поточний час замінюється секундоміром. Сценарій зберігає час запуску і стан секундоміру у пам'яті `localStorage`, через що відлік часу відображається вірно навіть після закриття і повторного відкриття сторінки;
- `gps.js`. Сценарій визначає висоту та швидкість по даним GPS телефону та виводить їх на спідометр. Швидкість відображається як додаткова стрілка на спідометрі.
- `webSocket.js`. Сценарій встановлює з'єднання з мікроконтролером ESP8266, містить функцію передачі та рекурсивну функцію прийому даних.

В інтерфейсі налаштувань використовується файли сценарію `webSocket.js` та вбудований сценарій в `html` файл для отримання поточних та відправки нових налаштувань до мікроконтролера ESP8266.

5.8 Android застосунок «KT-LCD»

Для побудови застосунку Android з розширенням `*.apk` був використаний онлайн сервіс «App Maker» на платформі «WEBINTOAPP» [19].

Інтерфейс сервісу простий, інтуїтивно зрозумілий, відображений на рисунку 21.

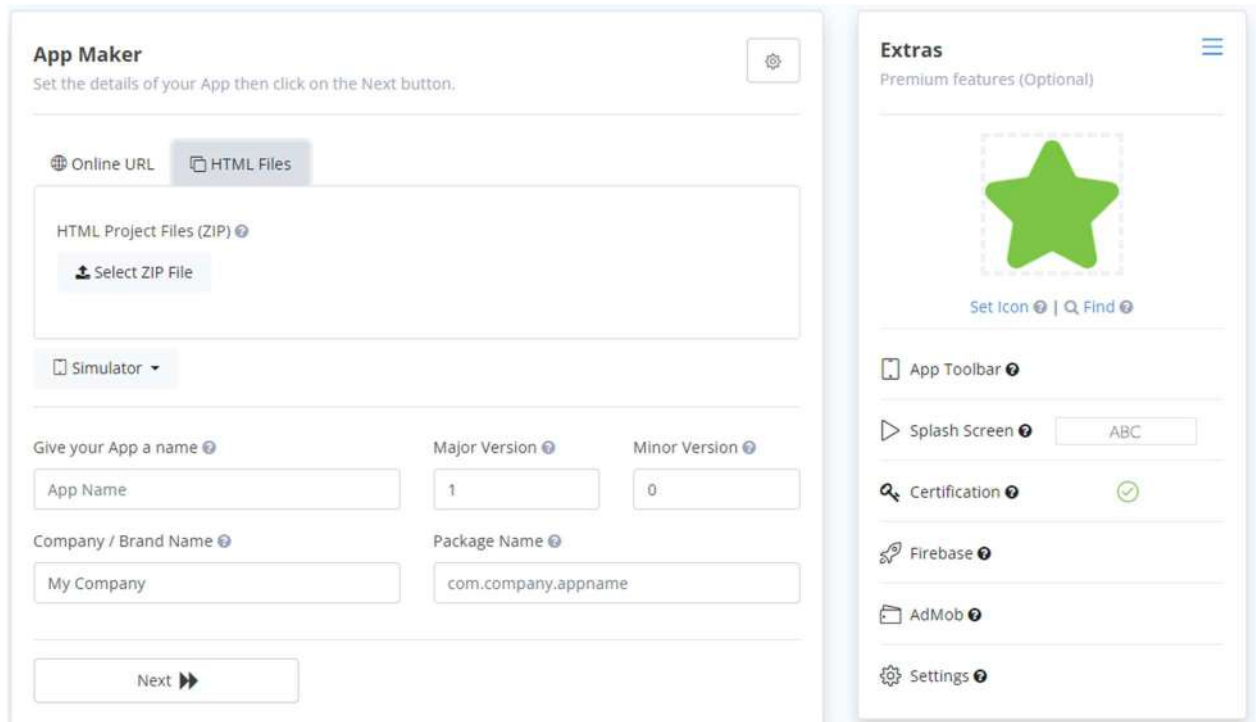


Рисунок 21

Після безкоштовної реєстрації необхідно до конструктору застосунків завантажити архів у форматі zip з файлами html, js, css, іконку застосунку, ввести назву та обрати необхідні налаштування, такі як доступ до GPS приймача, утримання дисплею постійно увімкненим, заборона зміни орієнтації дисплею. Менше ніж за хвилину конструктор робить готовий застосунок та розміщує його на окремій вебсторінці [20]. Для завантаження Android застосунку пропонується відсканувати qr-код, зображений на рисунку 22.



Рисунок 22 – Qr-код для завантаження застосунку «KT-LCD» для Android пристроїв

Сервіс «App Maker» має premium функцію, що дозволяє будувати застосунки для платформи IOS, що в перспективі може дати більше охоплення користувачів.

Вихідний код застосунку та програми мікроконтролеру ESP8266 завантажено у репозиторій github [21]. Розроблений вихідний код займає близько 80 сторінок A4 друкованого тексту. Для переходу на сторінку репозиторію github пропонується відсканувати qr-код, зображений на рисунку 23.



Рисунок 23 – Qr-код для переходу до репозиторію з вихідним кодом застосунку та програми мікроконтролера ESP8266

6 ПЕРЕВІРКА РОЗРОБЛЕНОГО ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

Після розробки і налагодження програмно-апаратного комплексу КТ-LCD його необхідно перевірити в реальних умовах поїздки та порівняти отримані результати з існуючими рішеннями.

6.1 Обладнання

Для перевірки програмно-апаратного комплексу КТ-LCD використовувався дорожній велосипед з алюмінієвою рамою, на який встановлено мотор-колесо потужністю 350 Вт та система PAS. Апаратна частина складалася з плати для наладки типу NODEMCU з вбудованим мікроконтролером типу ESP8266. Джерело живлення встановлювалося окреме з акумуляторним живленням та вихідною напругою 5В. Для порівняння результатів заїздів на андроїд смартфон були встановлені додатки КТ-LCD, SuperCycle, Google Fit та Zepp Life. З додатком Zepp Life використовувався фітнес браслет Xiaomi типу Mi Band 6.

6.2 Дорожні умови

Погода хмарна, температура повітря плюс 5 °С, вологий асфальт з невеликими нерівностями. Коло тренувань протяжністю 3,8 км. Швидкість та набір висоти вказано на рисунку 24.



Рисунок 24 – Швидкість та набір висоти під час заїзду за даними GPS

6.3 Критерії успішності перевірки

Критерії успішності перевірки наведені в таблиці 5.

Таблиця 5

Параметр	Критерії успішності
1. Швидкість	Різниця показів GPS та КТ-LCD не більша за 2 км/год. при постійній швидкості
2. Середня швидкість	Різниця показів GPS та КТ-LCD не більша за 2 км/год. при постійній швидкості
3. Відстань	Різниця показів GPS та КТ-LCD не більша за 5 % при сталій швидкості
4. Час поїздки	Час поїздки повинен рахуватися з моменту включення акумулятора до моменту закінчення поїздки
5. Показник потужності мотор-колеса	Максимальна виміряна потужність має відповідати номінальній потужності контролеру
6. Кількість спалених калорій	Кількість спалених калорій має корелюватися з дослідженнями [10] та [11]
7. Еквівалент спалення жиру	Має бути в 7,716179176 разів менший за кількість спалених калорій
8. Збереження показників	Показники одометрів та спалених калорій мають відображатися після повторного увімкнення системи
9. Швидкість наростання спалених калорій	Швидкість наростання спалених калорій має зменшуватися при збільшенні рівня PAS

6.4 Результати перевірки та порівняння з існуючими рішеннями

Перевірка програмно-апаратного комплексу КТ-LCD проводилася згідно з 6.3 та при умовах, зазначених в 6.2. Всі параметри, вказані в таблиці 5, відповідають критеріям успішності. Перевірка пройдена успішно.

Усереднені результати порівняння при п'ятих поїздках та при умовах, зазначених в п.6.2 з існуючими рішеннями наведені в таблиці 6.

Таблиця 6

Параметр	Результати			
	KT-LCD *	Zepp Life	SuperCycle**	Google Fit
Відстань, км	3,973	3,81	3,8	3,86
Час поїздки, хв.	18:48	18:24	18:09	18:16
Середня швидкість, км/год.	14,8	12,46	12,6	12,69
Максимальна швидкість, км/год.	38,3	34,58	35,4	31,76
Кількість спаленої енергії, ккал	122	158	75	150
Примітка 3. * – під час перевірки система допомоги PAS вимкнена. Різниця у часі поїздки викликана почерговим запуском тренування на одному андроїд телефоні. ** – показники спалених калорій сильно різняться між однаковими заїздами.				

Розрахуємо витрати калорій. Параметри велосипедиста: вік 36 років, зріст 184 см, вага 84 кг, час поїздки 18,5 хвилин. Згідно даних Американської ради з фізичних вправ [13], метаболічний еквівалент MET становить від 4, при повільному катанні на велосипеді, до 5,8, при швидкості 15 км / год.

Підрахуємо базові витрати калорій за формулою (5):

$$E_{\text{ч}} = 10 \cdot 84 + 6,25 \cdot 184 - 5 \cdot 36 + 5 = 1815 \text{ (ккал/добу)},$$

Підрахуємо кількість спалених калорій за формулою (7) для часу п'ять хвилин. Для MET 4,0:

$$E = (1815 \cdot 4) / 24 / 60 \cdot 18,5 = 93,3 \text{ (ккал)}$$

Для MET 5,8:

$$E = (1815 \cdot 5,8) / 24 / 60 \cdot 18,5 = 135,2 \text{ (ккал)}$$

Отже, підрахунок спалених калорій (122 ккал, див. Таблицю 6) застосунком КТ-LCD відповідає розрахункам за метаболічним еквівалентом. Застосунки Zepp Life, Google Fit завищують показники спалених калорій, SuperCycle, - надає менші показники.

ВИСНОВКИ

В ході роботи проведені дослідження, в результаті яких створена формула методом кореляційно-регресійного аналізу для розрахунку метаболічного еквіваленту для велосипедиста в залежності від швидкості руху велосипеда. Це дозволяє враховувати стать, вік та зріст велосипедиста та дає результат на основі швидкості пересування велосипеда. Розроблений алгоритм підрахунку спалених калорій при їзді на електричному велосипеді з використанням системи допомоги педального асистента.

Отримана формула та алгоритм перевірені на практиці. Результати порівняні з розрахунками втрачених калорій через MET еквівалент за класичними формулами.

Розроблений програмний код, що враховує результати досліджень спалених калорій, та фізичний мікроконтролерний пристрій для встановлення на електричний велосипед. Для виводу інформації та взаємодії з користувачем розроблений андроїд додаток, що обмінюється інформацією з фізичним мікроконтролерним пристроєм через бездротову мережу.

Проведені тренувальні заїзди на електричному велосипеді для перевірки правильності роботи комплексу.

Проведено дослідження пропрієтарного протоколу обміну даними між контролером мотор-колеса електричного велосипеда та LCD дисплею. В результаті дослідження створена C++ бібліотека для простої взаємодії з контролером електричного велосипеда, яка може бути застосована з іншими видами електротранспорту.

Результати дослідження підрахунку спалених калорій при катанні на електричному велосипеді рекомендується використовувати при розробці фітнес-додатків та велосипедних комп'ютерів, у сфері охорони здоров'я для розробки харчових раціонів та карти фізичних навантажень.

Розроблений апаратно-програмний комплекс зацікавить як існуючих користувачів електричних велосипедів, так і людей, що піклуються про своє

здоров'я та підраховують калорії. Наслідком заохочення людей до користування електричними велосипедами з частковою відмовою від ДВЗ з поєднанням з «зеленими» технологіями є збереження довкілля, економія палива, поліпшення здоров'я.

Розроблений апаратно-програмний комплекс рекомендується встановлювати на електричні велосипеди, що підтримують дисплеї КТ-LCD.

Пропонується продовження досліджень у напрямку уточнення алгоритму та формули підрахунку метаболічного еквіваленту. Окрім врахованих параметрів пропонується враховувати підйоми та спуски під час подорожі, каданс при обертанні педалей. Також необхідні додаткові практичні дослідження з виміром метаболічного еквіваленту при різній швидкості їзди та різним кадансом на спусках та підйомах траси.

РЕКОМЕНДАЦІЇ

Необхідність проведення подальших досліджень за звітною тематикою актуальна, оскільки у суспільстві популярна тематика здоров'я, зростає популярність фітнес браслетів та спортивних застосунків, через подорожчання палива люди все частіше пересідають на електричний транспорт.

Рекомендується продовження досліджень у напрямку уточнення алгоритму та формула підрахунку метаболічного еквіваленту. Окрім врахованих параметрів пропонується враховувати підйоми та спуски під час подорожі, каденс при обертанні педалей. Також необхідні додаткові практичні дослідження з виміром метаболічного еквіваленту при різній швидкості їзди з різним кадансом на спусках та підйомах. Після знаходження залежностей рекомендується доповнити алгоритм розрахунку спалених калорій знайденими коефіцієнтами. Величину спусків та підйомів траси можна вимірювати барометричним датчиком.

Результати дослідження спалення калорій при катанні на електричному велосипеді рекомендується використовувати при розробці фітнес-додатків та велосипедних комп'ютерів, у сфері охорони здоров'я для розробки харчових раціонів та планів тренувань.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Тиждень науки-2023. Факультет будівництва, архітектури та дизайну. Тези доповідей науково-технічної конференції, УДК 620.9:504.06, Використання відновлюваних джерел енергії для захисту довкілля, Якімцов Ю.В. Авраменко П.В., Запоріжжя, 24-28 квітня 2023 р. [Електронний ресурс] / Редкол. : В. Шаломєєв (відпов. ред.) Електрон. дані. – Запоріжжя : НУ «Запорізька політехніка», 2023. – 263 с. – 1 електрон. опт. диск (DVDROM); 12 см. – Назва з тит. екрана. ISBN 978-617-529-405-5
2. Мала гірнича енциклопедія. У 3 т. Т.1. / за ред. докт. техн. наук В.С. Білецького. – Донецьк : Донбас, 2004. – 640 с. – ISBN 966-7804-14-3.
3. Фізіологія з основами анатомії людини: Підруч. для студ. вищ. навч. закладів / Л.М. Малоштан, О.К. Рядних, Г.П. Жегунова та ін.; За ред. Л.М. Малоштан. – Х.: Вид-во НФаУ: Золоті сто-рінки, 2003. – 432 с. ISBN 966–615–193–6 ISBN 966–8032–85–3
4. Harris JA, Benedict FG, A Biometric Study of Human Basal Metabolism. Proceedings of the National Academy of Sciences of the United States of America. 1918. Vol. 4, No 12, P. 370-373.
5. J. Arthur Harris and Francis G. Benedict. A Biometric Study of Basal Metabolism in Man, Washington, DC: Carnegie Institution. 1918. 280 p.
6. Roza AM, Shizgal HM, The Harris Benedict equation reevaluated: resting energy requirements and the body cell mas. The American journal of clinical nutrition. 1984. Vol. 40, No 1, P. 168-182.

7. Mifflin MD, St Jeor ST, Hill LA, Scott BJ, Daugherty SA, Koh YO A new predictive equation for resting energy expenditure in healthy individuals. *The American journal of clinical nutrition*. 1990. Vol. 51, No 2, P. 241-247
8. Owen, O.E., Kavle, E., Owen, R.S., Polansky, M., Caprio, S., Mozzoli, M.A., Kendrick, Z.V., Bushman, M.C., Boden, G. A Reappraisal of caloric requirements in healthy women. *Am. J. Clin. Nutr.* 1986. Vol. 44, No 1, P. 1-19
9. Owen, O.E.; Holup, J.L.; D'Alessio, D.A.; Craig, E.S.; Polansky, M.; Smalley, K.J.; Kavle, E.C.; Bushman, M.C.; Owen, L.R.; Mozzoli, M.A. A reappraisal of the caloric requirements of men. *Am. J. Clin. Nutr.* 1987. Vol. 46, No 6, P. 875-885.
10. A New Mode for Meeting Physical Activity Guidelines? : *Medicine & Science in Sports & Exercise* 41(11) : p 2097-2102, November 2009 : Research Center Physical Activity, Work and Health, TNO-VU/VUmc, VU University Medical Center, Amsterdam, THE NETHERLANDS. : [Электронный ресурс] : Режим доступа : https://journals.lww.com/acsm-msse/fulltext/2009/11000/electrically_assisted_cycling__a_new_mode_for.18.aspx
11. Bourne, J.E., Sauchelli, S., Perry, R. et al. Health benefits of electrically-assisted cycling: a systematic review. *Int J Behav Nutr Phys Act* 15, 116 2018 : [Электронный ресурс] : Режим доступа : <https://doi.org/10.1186/s12966-018-0751-8>
12. Калькулятор калорий при езде на велосипеде. [Электронный ресурс] : Режим доступа : <https://livelong.pro/kalkulyator-kalorij-pri-ezde-na-velosipede/>

13. Britannica® Group ProCon.org. MET Values for 800+ Activities. [Електронний ресурс] : Режим доступу : <https://golf.procon.org/met-values-for-800-activities/>
14. Xebike [Електронний ресурс] : Xdocs : KT-LCD6 electric bicycle meter KT—LCD6 Product / Suzhou KUNTENG electronics CO LTD // User Manual. 2016. Режим доступу до інструкції : <https://xebike.com/wp-content/uploads/2020/01/kt-lcd6-ebike-special-meter.pdf>
15. Alldatasheet: LM2596HVS-3.3 Step-Down Switching Regulator / Datasheet/ TECH PUBLIC Electronics co LTD : [Електронний ресурс] : Режим доступу : <https://pdf1.alldatasheet.com/datasheet-pdf/view/1761605/TECHPUBLIC/LM2596HVS-3.3.html>
16. Форум S-LCD to S12S controller communication protocol hacked [Електронний ресурс] : Режим доступу : <https://endless-sphere.com/sphere/threads/s-lcd-to-s12s-controller-communication-protocol-hacked.73471/>
17. Форум. Протокол KT-LCD. [Електронний ресурс] : Режим доступу : <https://electrotransport.ru/index.php?topic=58478.0>
18. EasyEDA editor [Електронний ресурс] : Режим доступу : <https://easyeda.com/editor>
19. Webintoapp [Електронний ресурс] : Режим доступу : <https://www.webintoapp.com/>
20. Сторінка розробленого мобільного застосунку KT-LCD [Електронний ресурс] : Режим доступу : <https://www.webintoapp.com/store/188618>

21. Вихідний код проєкту KT-LCD [Електронний ресурс] : Режим доступу : <https://github.com/androidpasha/KT-LCD3.git>

ДОДАТОК А

Алгоритм підрахунку затрачених калорій

На рисунку А.1 наведений Алгоритм підрахунку затрачених калорій.

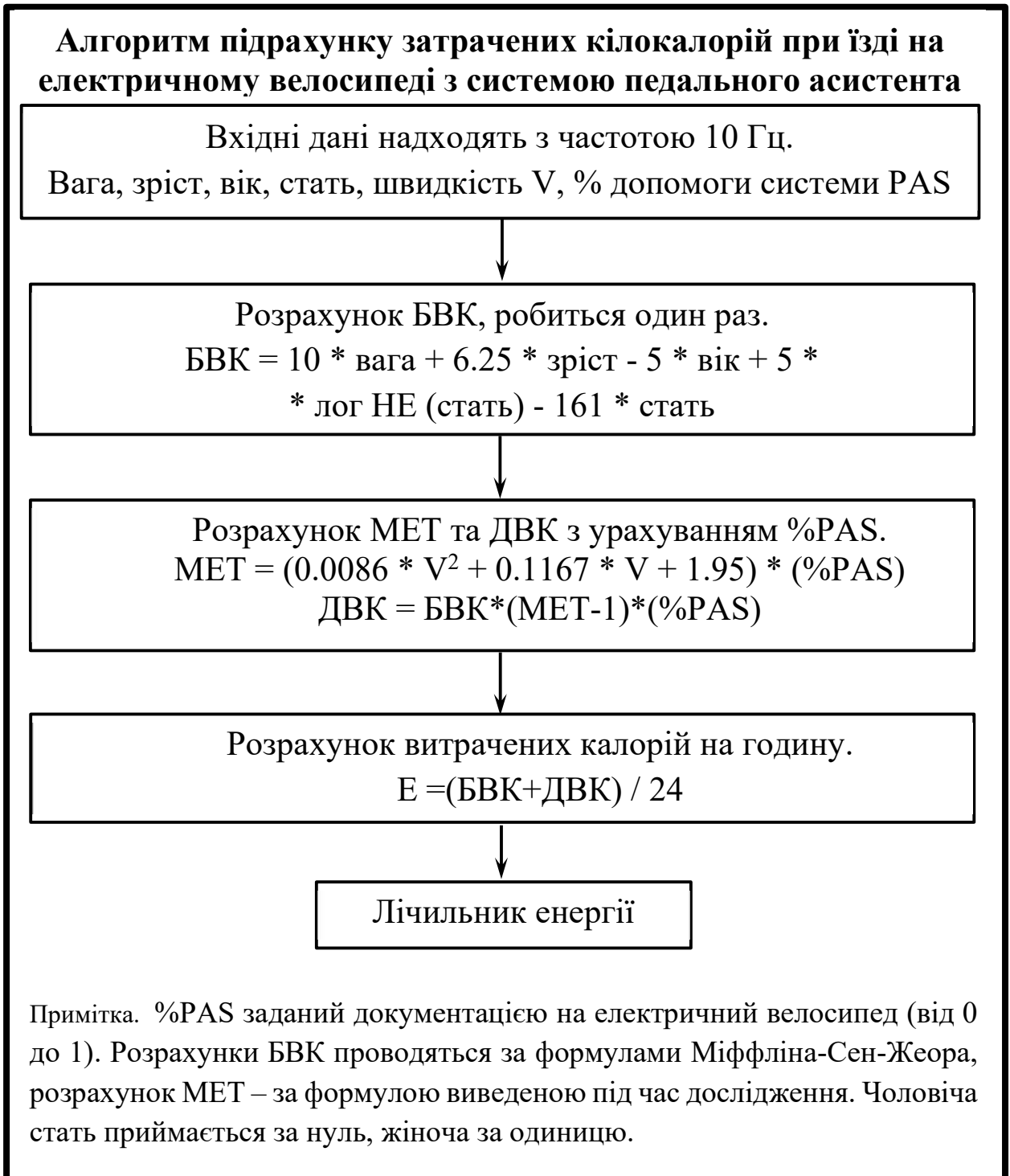


Рисунок А.1 - алгоритм підрахунку витрат калорій при їзді на електричному велосипеді

ДОДАТОК Б

Лістинг класу КТ-LCD на мові С++

```

#define RECEIVE_BUFFER_SIZE 12
#define TRANSFER_BUFFER_SIZE 13
#define SERIALTIMEOUT 50
#define MINSPEED 2

typedef struct __attribute__((packed, aligned(1))) SettingsStruct
{
    uint8_t pasLevel : 3; // Range:0-3
    bool cruise;
    bool walkMode;
    bool frontLight;
    uint8_t powerFactor;
    uint8_t maxSpeedLimit;
    uint16_t wheelRimDiameter; // 6-28 inch and 700C
    uint8_t P1_MotorFeature : 8; // Range : 1-255.
    uint8_t P2_WheelSetSpeedPulse : 3; // Rage:0-6
    bool P3_PasControlMode;
    bool P4_ThrottleActiveMode;
    uint8_t P5_BatteryMonitorModes : 6; // Range:0-40
    uint8_t C1_PasSensorSensitivity : 3; // Range:0-7
    uint8_t C2_MotorPhase : 3; // Range:0-7
    uint8_t C4_ThrottleFunction : 3; // Range:0-4
    uint8_t C4_SpeedLimitValueOfThrottle : 7; // Range:0-127
    uint8_t C4_PercentageValueOfTheFirstGearSpeed : 6; // Range:0-100
    uint8_t C5_ControllerMaxCurrent : 4; // Range:0-10
    uint8_t C12_ControllerLowestVoltage : 3; // Range:0-7
    uint8_t C13_ControllerAbsBraking : 3; // Range:0-5

```

```

uint8_t C14_PasAdjustment : 2;    // Range:1-3

float onePeriodDistance()
{
    static float result = 0;
    static uint16_t previousDiameter = 0;
    if (previousDiameter != wheelRimDiameter)
    {
        previousDiameter = wheelRimDiameter;
        float diametrInc = (float)wheelRimDiameter;
        if (wheelRimDiameter == 700)
            diametrInc = 27.5;
        result = PI * diametrInc * 0.0254f;
    }
    return result;
}

uint8_t wheelRimDiameterHEX()
{
    const uint16_t wheelSize[] = {5, 6, 8, 10, 12, 14, 16, 18, 20, 23, 24, 26, 28,
29, 700};
    const uint8_t wheelSizeHEX[] = {0x16, 0x12, 0x0A, 0x0E, 0x02, 0x06, 0x00,
0x04, 0x08, 0x0C, 0x10, 0x14, 0x1C, 0x1E, 0x18};
    for (uint8_t i = 0; i < sizeof(wheelSize) / sizeof(wheelSize[0]); i++)
        if (wheelRimDiameter == wheelSize[i])
            return wheelSizeHEX[i];
    return 0x10; // default 24`
}

} SettingsStruct;

typedef struct __attribute__((packed, aligned(1))) ReceiveDataStruct
{
    uint32_t speedPeriod;

```

```

float speed;
uint16_t power;
uint16_t temperature;
bool throttle;
bool cruise;
bool assistant;
byte brake;
uint8_t error;    //(0x20 or 0x25 or 0x28): "0info", 0x21: "6info", 0x22:
"1info", 0x23: "2info", 0x24: "3info", 0x26: "4info"
uint8_t ratedVoltage; // 24,36,48V
uint8_t batteryLevel; // 0: empty, 1: border flashing, 2: charging, 3: empty, 4: 1
bar, 8: 2 bars, 16: full
uint8_t batteryPercent()
{
    uint8_t byteToPercent[2][6] = {
        {0, 1, 3, 4, 8, 16}, // 0: empty, 1: border flashing, 2: charging, 3: empty, 4:
1 bar, 8: 2 bars, 16: full
        {0, 10, 25, 50, 75, 100}};
    for (int i = 0; i < 6; i++)
        if (batteryLevel == byteToPercent[0][i])
            return byteToPercent[1][i];
    return 0;
}
} ReceiveDataStruct;

class KTLCD_Display
{
public:
    KTLCD_Display(SettingsStruct *settings_variable_to_remove)
    {

```

```

Serial.setTimeout(SERIALTIMEOUT);
settings = *settings_variable_to_remove;
delete[] settings_variable_to_remove;
settings_variable_to_remove = nullptr;
}

KTLCD_Display()
{
Serial.setTimeout(SERIALTIMEOUT);
}

ReceiveDataStruct receiveData;
SettingsStruct settings;

~KTLCD_Display()
{
delete[] receiveBuffer;
delete[] transferBuffer;
}

void sendSettingsToController()
{
if ((receiveData.cruise == true) and receiveData.throttle == true)
settings.cruise = false;
fillTransferBuffer();
sendBufferToController(transferBuffer, TRANSFER_BUFFER_SIZE);
}

bool dataAvailable()
{

```

```

if ((bool)Serial.available() == true) // >= RECEIVE_BUFFER_SIZE)
{
    return getReceivedBufferFromController();
}
return false;
}

```

private:

```

uint8_t *receiveBuffer = new uint8_t[RECEIVE_BUFFER_SIZE];
uint8_t *transferBuffer = new uint8_t[TRANSFER_BUFFER_SIZE];

```

```

bool getReceivedBufferFromController()

```

```

{
    if (SerialRead(receiveBuffer, RECEIVE_BUFFER_SIZE) == true)
    {
        getDataFromReceiveBuffer(receiveBuffer, RECEIVE_BUFFER_SIZE);
        return true;
    }
    return false;
}

```

```

void fillTransferBuffer()

```

```

{

```

```

#define BYTE transferBuffer

```

```

#define parameter settings

```

```

uint8_t SizeHEX = parameter.wheelRimDiameterHEX();

```

```

BYTE[0] = parameter.P5_BatteryMonitorModes;

```

```

BYTE[1] = (parameter.pasLevel & 0x07);
bitWrite(BYTE[1], 6, parameter.walkMode);
bitWrite(BYTE[1], 7, parameter.frontLight);
BYTE[2] = ((parameter.maxSpeedLimit - 10) << 3 & 0xF8) |
          (SizeHEX >> 2 & 0x07);
BYTE[3] = parameter.P1_MotorFeature;
BYTE[4] = (SizeHEX << 6 & 0xC0) |
          (parameter.P2_WheelSetSpeedPulse & 0x07) |
          (parameter.P3_PasControlMode << 3 & 0x08) |
          (parameter.P4_ThrottleActiveMode << 4 & 0x10);
bitWrite(BYTE[4], 5, bitRead(parameter.maxSpeedLimit - 10, 5));
BYTE[6] = (parameter.C1_PasSensorSensitivity << 3 & 0x38) |
          (parameter.C2_MotorPhase & 0x07);
BYTE[7] = (parameter.C5_ControllerMaxCurrent & 0x0F) |
          (parameter.C14_PasAdjustment << 5 & 0x60);
BYTE[8] = ((parameter.C4_ThrottleFunction + 1) << 5 & 0xE0) |
          (parameter.C12_ControllerLowestVoltage & 0x07) |
          (parameter.cruise << 4 & 0x10);
BYTE[9] = parameter.C4_SpeedLimitValueOfThrottle & 0x7F;
BYTE[10] = (parameter.C13_ControllerAbsBraking << 2 & 0x1C);
BYTE[11] = parameter.C4_PercentageValueOfTheFirstGearSpeed & 0x3F;
BYTE[12] = 0x0E;
BYTE[5] = XOR(transferBuffer, TRANSFER_BUFFER_SIZE, 5) ^ 2;
#undef BYTE
#undef parameter
}

uint8_t XOR(uint8_t *buffer, size_t bufferSize, uint8_t placeXor)
{
    uint8_t XOR = 0;

```

```

for (uint8_t i = 0; i < bufferSize; i++)
{
    if (i == placeXor)
        continue;
    XOR ^= buffer[i];
}
return XOR;
}

```

```

bool SerialRead(uint8_t *buffer, size_t bufferSize)
{
    Serial.readBytes(buffer, bufferSize); // Отримує масив по UART
    uint16_t bufferSum = 0;
    for (uint8_t i = 0; i < bufferSize; i++)
        bufferSum += buffer[i];

    if (XOR(buffer, bufferSize - 1, 6) != buffer[6] or bufferSum == 0) // останній
байт буферу не розраховується у XOR, тому bufferSize - 1
        return false;
    return true;
}

```

```

void getDataFromReceiveBuffer(uint8_t *buffer, size_t bufferSize =
RECEIVE_BUFFER_SIZE)
{
#define BYTE buffer
    receiveData.batteryLevel = BYTE[1];
    receiveData.ratedVoltage = BYTE[2];
    receiveData.speedPeriod = (BYTE[3] << 8) | BYTE[4]; // BYTE[3] * 256 +
BYTE[4];

```

```

    receiveData.speed = settings.onePeriodDistance() * 3600 /
receiveData.speedPeriod;
    if (receiveData.speed < MINSPEED)
        receiveData.speed = 0;
    receiveData.error = BYTE[5];
    receiveData.throttle = !bitRead(BYTE[7], 0);
    receiveData.cruise = bitRead(BYTE[7], 3);
    receiveData.assistent = bitRead(BYTE[7], 4);
    receiveData.brake = bitRead(BYTE[7], 5);
    receiveData.power = BYTE[8] * settings.powerFactor;
    receiveData.temperature = (int8_t)BYTE[9] + 15;
#undef BYTE
}

void sendBufferToController(uint8_t *buffer, size_t bufferSize =
TRANSFER_BUFFER_SIZE)
{
    Serial.write(buffer, bufferSize); // Надсилає масив по UART
}
};

```