

Велосипедний бортовий комп'ютер та система безпеки на базі ESP32

Стан проблеми. Зростання популярності індивідуального транспорту вимагає розробки інтегрованих систем, що підвищують безпеку та інформативність поїздок. Існуючі рішення часто є вузькоспеціалізованими або мають обмежені можливості логування. Для створення комплексної, функціональної та універсальної системи, що поєднує світлову сигналізацію, автоматичний стоп-сигнал та точний трекінг, необхідне потужне та гнучке апаратне рішення.

Проект має підвищену актуальність завдяки можливості використання не лише на звичайних велосипедах, а й на електротранспорті (електровелосипеди, електросамокати). Це досягається завдяки універсальності базового принципу вимірювання швидкості через датчик Холла для будь-якого обертового колеса[3].

Постановка задачі. Розробити та реалізувати інтелектуальну бортову систему на базі високопродуктивного мікроконтролера ESP32[1,5,6]. Система повинна забезпечувати точний вимір швидкості та пройденої відстані за допомогою датчика Холла[3]. Необхідно реалізувати ручне керування світловими сигналами поворотів та переднім ліхтарем за допомогою фізичних кнопок. Важливою функцією є автоматична активація стоп-сигналу шляхом аналізу зменшення швидкості, що вираховується через зміну кількості обертів колеса за певний відрізок часу, без використання додаткового акселерометра. Усі сирі дані мають бездротово передаватися в реальному часі на мобільний додаток через протокол Bluetooth Low Energy (BLE)[2,4] для обчислень, візуалізації та логування. Кінцевою метою є досягнення високої швидкодії та енергоефективності, необхідних для автономної роботи від павербанку.

Розв'язання задачі. Центральним елементом є мікроконтролер ESP32, який використовується завдяки його 32-бітній архітектурі, великому обсягу SRAM та вбудованій підтримці BLE[1,5]. Входи системи включають один ключовий датчик – Датчик Холла (для швидкості та сповільнення, підключення через зовнішнє переривання)[3] та фізичні кнопки для ручного керування світлом і поворотами. Виходи керують світлодіодами та ліхтарем через транзисторні ключі, забезпечуючи надійну роботу від павербанку. Прошивка ESP32 (Тонкий Клієнт)[1] обробляє переривання від датчика Холла, керує логікою світлових сигналів (Toggle-логіка для кнопок, неблокуюче мигання), працює як BLE Server та активно передає сирі дані через механізм Notify[2]. Логіка автоматичного стоп-сигналу реалізується програмно на ESP32 або в

мобільному додатку шляхом постійного порівняння поточної швидкості (V_2) із попередньою швидкістю (V_1), розрахованими на основі часу між обертами колеса. Якщо швидкість зменшується різкіше, ніж заданий поріг, активується стоп-сигнал, що забезпечує надійну активацію при гальмуванні без використання додаткового акселерометра. Мобільний додаток (Товстий Клієнт) працює як BLE Client, приймає сирі дані та виконує всі необхідні обчислення (швидкість, відстань), забезпечуючи візуалізацію (Dashboard) і логування поїздок у пам'яті телефону[2,4].

Система легко адаптується до електросамокатів та електровелосипедів, оскільки вимір швидкості та сповільнення базується на універсальному методі: кількість обертів колеса, зафіксована датчиком Холла[3]. Для адаптації потрібна лише зміна калібрувальної константи (довжини кола колеса) у програмному забезпеченні мобільного додатка.

Для перевірки коректності роботи програми та оцінки швидкодії розроблено план верифікації, який включає наступні етапи:

Таблиця 1.

Етапи тестування коректності роботи програми

Назва етапу тестування	Оцінюваний Параметр	Очікуваний Результат
Тест 1 “Зв’язок”	Стабільність BLE-з’єднання	Безперервна передача даних протягом усього циклу поїздки
Тест 2 “Керування”	Час реакції на натискання кнопок	Миттєве вмикання/вимикання світла та поворотів (до 50 мс)
Тест 3 “Автоматика”	Точність спрацювання стоп-сигналу	Визначення мінімального порогу сповільнення для активації
Тест 4 “Швидкодія”	Частота оновлення швидкості в додатку	Оновлення показника на смартфоні не рідше ніж 4 рази на секунду

Інноваційні функції системи включають інтелектуальний алгоритм прогнозування потреби у воді на основі історичних даних, автодіагностику сенсорів, режим енергозбереження (deep sleep) між циклами поливу та збереження даних у хмарі через ThingSpeak або Firebase для аналізу та оптимізації режимів поливу. Система також підтримує масштабування, що дозволяє додавати нові вазони або сенсори без значної модифікації апаратної частини [4,5,6].

Висновки. Реалізація інтелектуальної системи на базі ESP32 з бездротовим зв'язком BLE забезпечує надійність, гнучкість та можливість детального аналізу поїздок. Застосування єдиного датчика для швидкості

та стоп-сигналу спрощує апаратну частину, а прийнята архітектура дозволяє мінімізувати навантаження на мікроконтролер та гарантує високу швидкодію критично важливих функцій безпеки.

Література.

- [1] Monk, S. Programming the ESP32 with Arduino. McGraw-Hill Education, 2021. ISBN: 9781260463943..
- [2] Aksoy, K. Beginning Bluetooth Low Energy. Apress, 2017. DOI: 10.1007/978-1-4842-2300-4.
- [3] Fraden, J. Handbook of Modern Sensors: Physics, Designs, and Applications. Springer, 2015. DOI: 10.1007/978-3-319-19307-1.
- [4] Bianchi, D. et al. A Study of the Feasibility of Co-located App Attacks against BLE. arXiv:1808.03778 [cs.CR], 2018
- [5] Anufriev, A. et al. Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things. ResearchGate, 2017.
- [6] Nasibov, Y. et al. Getting Started with Arduino and ESP32: A Beginner's Guide with Wokwi. ResearchGate, 2024