

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

“ _____ ” _____ 20__ р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему

**“Програмне забезпечення системи завадостійкої передачі даних через
лазерний канал за наявності штучних перешкод на основі
мікроконтролера ESP32”**

Виконав здобувач вищої освіти

IV курсу, групи КІ-21-1

ОПП «Кібербезпека»

спеціальності 123 «Комп’ютерна інженерія»

_____ Золотовський Є.Р.

Керівник проекту д.т.н., професор

_____ Коваленко О.В.

Рецензент _____

м. Кропивницький

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти бакалавр
Галузь знань. 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма ”Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

О.А.Смірнов

« __ » _____ 20__ року

**ЗАВДАННЯ НА ВИПУСКТНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА
ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

Золотовському Євгенію Руслановичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи завадостійкої передачі даних через лазерний канал за наявності штучних перешкод на основі мікроконтролера ESP32

керівник роботи Коваленко Олександр Володимирович, док. техн. наук, професор

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

затверджені наказом вищого навчального закладу № 46-02 від 17.01.2025 року

2. Строк подання студентом роботи до захисту 23.05.2025 р.

3. Мета та завдання кваліфікаційної бакалаврської роботи: *Метою розробки є програмне забезпечення передачі даних через лазерний канал за наявності перешкод*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

6. Дата видачі завдання «17» січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	13.03.2025 р.	
2.	Постановка задачі, оформлення ТЗ	16.03.2025 р.	
3.	Розробка моделі компонента	20.03.2025 р.	
4.	Розробка структур даних	15.03.2025 р.	
5.	Розробка алгоритмів зв'язку та відображення	25.03.2025 р.	
6.	Програмування алгоритмів	10.04.2025 р.	
7.	Оформлення ПЗ	25.04.2025 р.	
8.	Попередній захист роботи	24.05.2025 р.	

Дата видачі завдання

«17» січня 2025 р.

Підпис керівника

(прізвище та ініціали)

Завдання прийнято до виконання

«25» січня 2025 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Золотовський Є. Р. Програмне забезпечення системи завадостійкої передачі даних через лазерний канал за наявності штучних перешкод на основі мікроконтролера ESP32. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

У даній кваліфікаційній бакалаврській роботі розроблено програмне забезпечення, яке призначено для передачі даних через лазерний канал за наявності штучних перешкод

Метою роботи є створення програмного забезпечення яке буде передавати дані через лазерний канал за наявністю перешкод

Результат роботи – створено повністю робочу систему захищеної від перешкод передачі даних лазерним каналом, де використано мікроконтролери ESP32-WROOM-32, лазерний діод KY-008 та фототранзистор BPW85C. Система забезпечує стабільний зв'язок на відстані до 10 метрів зі швидкістю 10 Кбіт/с, використовуючи модуляцію ООК (100 Гц) та код Хеммінга для виправлення поодиноких помилок.

У процесі розробки проведено аналіз методів передачі даних лазерним каналом, досліджено схеми модуляції (ООК), протоколи синхронізації та методи боротьби з навмисними перешкодами (освітлення 500 люкс, затемнення). Результатом стала розробка власного програмного забезпечення на C++ в Arduino IDE для двох ESP32: передавача (генерування, кодування, модуляція) та приймача (зчитування, фільтрація, декодування). Програма містить фільтрацію сигналу (усереднення 10 зразків АЦП), динамічне калібрування порогу (1500 одиниць) та вивід статистики (кількість бітів, відсоток втрат) через Serial Monitor.

Систему випробувано в умовах штучних перешкод, досягнуто точності прийому 95% при низькому рівні шумів. Архітектура модульна, що дає можливість додавати нові функції, як-от інтеграцію з MQTT або шифрування.

Розробка довела ефективність лазерного зв'язку як альтернативи радіозв'язку в промисловості, IoT та аварійних ситуаціях.

Програмне забезпечення на мові Arduino IDE.

Додані інструкції щодо використання програмного забезпечення сприяють оперативному розгортанню системи для проведення тестувань чи демонстрацій.

Програмне забезпечення коректно працює на персональних комп'ютерах архітектури IBM PC під управлінням операційних систем Windows 10 або Windows 11. Система може бути застосована як у навчальних цілях, так і як основа для розробки спеціалізованих бездротових оптичних каналів зв'язку в умовах обмеженого радіозв'язку.

Ключові слова: лазерний канал, ESP32, передача даних, світловий сигнал, Arduino IDE.

КБПЗ_2025

ABSTRACT

Zolotovskiy E. R. Software of the system of interference-resistant data transmission through a laser channel in the presence of artificial interference based on the ESP32 microcontroller. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this qualifying bachelor's work, software has been developed that is designed to transmit data through a laser channel in the presence of artificial interference

The purpose of the work is to create software that will transmit data through the laser channel in the presence of interference

The result of the work is a fully working system of data transmission protected from interference by a laser channel, which uses ESP32-WROOM-32 microcontrollers, a laser diode KY-008 and a phototransistor BPW85C. The system provides stable communication at a distance of up to 10 meters at a speed of 10 Kbps, using OOK modulation (100 Hz) and Hamming code to correct single errors.

In the process of development, the analysis of data transmission methods by the laser channel was carried out, modulation schemes (OOK), synchronization protocols and methods for combating intentional interference (500 lux lighting, dimming) were studied. The result was the development of its own C++ software in the Arduino IDE for two ESP32: the transmitter (generation, encoding, modulation) and the receiver (reading, filtering, decoding). The program contains signal filtering (averaging 10 ADC samples), dynamic threshold calibration (1500 units) and statistical output (number of bits, percentage loss) through Serial Monitor.

The system was tested under conditions of artificial interference, 95% reception accuracy was achieved with a low noise level. The architecture is modular, which makes it possible to add new features, such as integration with MQTT or encryption. The development proved the effectiveness of laser communication as an alternative to radio communication in industry, IoT and emergency situations.

Software in Arduino IDE language.

The added instructions for using the software facilitate the operational deployment of the system for testing or demonstrations.

The software works correctly on personal computers of the IBM PC architecture running Windows 10 or Windows 11 operating systems. The system can be used both for training purposes and as a basis for developing specialized wireless optical communication channels in conditions of limited radio communication.

Keywords: laser channel, ESP32, data transmission, light signal, Arduino IDE.

K6П3_2025

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	10
1.1 Призначення системи	10
1.2 Область застосування.....	13
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	18
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної бакалаврської роботи.....	18
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	24
2.3 Розгорнута постановка завдання	27
3 ОПИС І ОБґРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	29
3.1 Опис функціонування системи	29
3.2 Розробка структурної схеми.....	33
3.3 Розробка функціональної схеми	37
3.4 Розробка діаграми процесів.....	44
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	54
4.1 Блок-схеми та опис алгоритмів функціонування системи.....	54
4.2 Захист розробленого програмного забезпечення.....	68
5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	73
6 ОСНОВНІ ВИСНОВКИ.....	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	84

						ВКРБ-123.25.0009.00.00.ПЗ		
Вим	Арк	№ докум.	Підп.	Дата		Літ.	Аркуш	Аркушів
Розроб.		Золотовський Є.Р			Програмне забезпечення системи завадостійкої передачі даних через лазерний канал за наявності штучних перешкод на основі мікроконтролера ESP32	Б	1	89
Перев.		Коваленко О.В.				ЦНТУ, гр. КІ-21-1		
Н.контр.		Коваленко А.С.						
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, СИМВОЛІВ ТА СПЕЦІАЛЬНИХ ТЕРМІНІВ

ESP32 – Embedded Serial Processor 32-bit – мікроконтролер із вбудованим Wi-Fi та Bluetooth, виробництва Espressif Systems.

LED – Light Emitting Diode – світлодіод, використовується як передавач в оптичному каналі.

ФД – фотодіод – напівпровідниковий елемент, який перетворює світло в електричний сигнал, приймач оптичного сигналу.

GPIO(D) – General Purpose Input/Output – універсальні порти введення/виведення на мікроконтролері.

UART – Universal Asynchronous Receiver/Transmitter – універсальний асинхронний приймач/передавач, інтерфейс для передачі даних між мікроконтролерами.

PWM – Pulse Width Modulation – широтно-імпульсна модуляція, метод кодування інформації за допомогою ширини імпульсів.

CRC – Cyclic Redundancy Check – циклічна надмірна перевірка, метод виявлення помилок у переданих даних.

OOK – On-Off Keying – модуляція методом вмикання/вимикання, найпростіший тип амплітудної модуляції.

FSK – Frequency Shift Keying – частотна маніпуляція, тип модуляції, за якого логічні стани кодуються різними частотами.

Біт – найменша одиниця інформації, яка може набувати значення 0 або 1.

Байт (Byte) – одиниця вимірювання обсягу інформації, що дорівнює 8 бітам.

Завадостійкість – здатність системи зберігати працездатність при впливі завад.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

Оптичний канал – середовище передачі даних у вигляді світлового випромінювання, зазвичай за допомогою інфрачервоних або лазерних джерел.

Лазерний передавач – пристрій, що генерує вузьконаправлений пучок світла для передачі даних.

Буферизація – тимчасове збереження даних у пам'яті для забезпечення узгодженості передачі або обробки.

Шум – небажані сигнали, які впливають на якість передавання корисної інформації.

ІЧ (інфрачервоний) діапазон – частина електромагнітного спектру, яку часто використовують для бездротової передачі даних через світло.

ПК – персональний комп'ютер.

SQL – Structured query language.

iOS – власницька мобільна операційна система від Apple.

MQTT – Message Queue Telemetry Transport.

КБПЗ-2025

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

У сучасному світі інформаційні технології є ключем до розвитку багатьох галузей, зокрема телекомунікацій, промислової автоматизації, Інтернету речей (IoT), безпілотних транспортних засобів та космічних досліджень. Один з перспективних напрямів у сфері передачі даних – використання лазерних каналів зв'язку, відомих як Free Space Optics (FSO). Ці системи використовують модуляцію світла для передачі інформації у вільному просторі, забезпечуючи високу швидкість передачі (до десятків мегабіт на секунду), низький рівень електромагнітних завад і можливість роботи в середовищах, де радіочастотний зв'язок обмежений через нормативні або технічні обмеження. Наприклад, FSO-системи знаходять застосування в міських мережах для з'єднання між будівлями, у військових технологіях для безпечної передачі даних та в космічних апаратних комплексах для обміну даними між супутниками.

Не зважаючи на значні переваги, лазерні канали зв'язку мають істотний недолік такий як: чутливість до зовнішніх перешкод. Штучні перешкоди, такі як шум, вібрації, зміни інтенсивності світла, пил, атмосферні умови чи фізичне перекриття променя, можуть призвести до помилок у передачі даних або навіть до втрати сигналу. Наприклад, у виробничих умовах вібрації обладнання можуть викликати зміщення лазерного променя, а в IoT-системах затінення чи забруднення оптичних компонентів погіршують якість сигналу. Ці проблеми особливо актуальні для компактних, низькоенергетичних систем, де складні методи виправлення помилок можуть бути обмежені апаратними можливостями.

Для підвищення надійності передачі даних в таких умовах використовуються методи завадостійкого кодування, які дозволяють виявляти та виправляти помилки, що виникають під час передачі. Один з найпоширеніших та найпростіших методів – код Хеммінга, який завдяки своїй ефективності та низьким обчислювальним вимогам ідеально підходить для реалізації на

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

мікроконтролерах з обмеженими ресурсами, таких як ESP32-WROOM-32. Код Хеммінга додає перевірочні біти до інформаційних даних, дозволяючи виявляти до двох помилок і виправляти одну в кожному кодовому слові. Це забезпечує достатній рівень надійності для лабораторних експериментів і промислових застосувань з помірним рівнем перешкод.

Актуальність теми

Актуальність теми дипломної роботи зумовлена зростаючою потребою у надійних та економічно вигідних системах передачі даних, здатних функціонувати в умовах штучних перешкод. Використання мікроконтролера ESP32-WROOM-32, лазерного модуля KY-008 (650 нм, 5 мВт) та фототранзистора BPW85C дозволяє створити компактну та доступну систему, яка може бути інтегрована в IoT, розумні будинки, безпілотні системи або промислові мережі. ESP32-WROOM-32 виділяється високою продуктивністю, підтримкою інтерфейсів GPIO та UART, а також низькою вартістю, що робить його ідеальним вибором для прототипування та впровадження. Лазерний модуль KY-008 забезпечує просту модуляцію сигналу методом On-Off Keying (OOK), а фототранзистор BPW85C має високу чутливість до видимого світла, що дозволяє ефективно приймати сигнали в лабораторних умовах. Резистори 10 кОм та 120 Ом забезпечують стабільність роботи лазера та фототранзистора, що критично для надійної передачі.

Розробка програмного забезпечення для такої системи з використанням коду Хеммінга не лише підвищує надійність передачі даних, але й демонструє можливість створення економічно вигідних рішень для реальних потреб. У контексті сучасних викликів, таких як збільшення обсягів даних в IoT та потреба в безпечному зв'язку, розробка подібних систем є актуальною та перспективною. До того ж, використання доступних компонентів, таких як KY-008 і BPW85C, робить систему привабливою для освітніх та дослідницьких проектів, де ключовими є низька вартість та простота реалізації.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Мета дипломної роботи

розробка програмного забезпечення для мікроконтролера ESP32-WROOM-32, яке забезпечує завадостійку передачу даних через лазерний канал зв'язку, створений за допомогою лазерного модуля KY-008 як передавача та фототранзистора BPW85C як приймача, з використанням методу кодування Хеммінга. Програмне забезпечення має бути реалізоване в середовищі Arduino IDE на мові C++ та забезпечувати надійну передачу даних на відстань до 10 метрів у лабораторних умовах навіть за наявності штучних перешкод, таких як затемнення, вібрації чи короткочасні переривання променя. Система має включати два мікроконтролери ESP32-WROOM-32: один для кодування та передачі даних, другий для прийому та декодування з виправленням помилок. Код Хеммінга обрано через його простоту та ефективність, що дозволяє досягти завадостійкості без значних обчислювальних витрат.

Для досягнення поставленої мети в роботі необхідно виконати наступні завдання:

- Провести аналіз методів завадостійкого кодування, включаючи код Хеммінга, з оцінкою придатності для реалізації на мікроконтролері ESP32-WROOM32 з урахуванням обмежень пам'яті та обчислювальної потужності.
- Розробити апаратно-програмний комплекс на основі двох мікроконтролерів ESP32-WROOM-32, де один виконує функцію передавача (з підключенням лазерного модуля KY-008 через резистор 120 Ом), а другий – приймача (з підключенням фототранзистора BPW85C через резистор 10 кОм).
- Реалізувати алгоритм кодування Хеммінга для обробки даних перед передачею, а також алгоритм декодування з виявленням та виправленням помилок на стороні приймача, використовуючи інтерфейси GPIO та UART для обміну даними.
- Провести тестування розробленої системи в лабораторних умовах з модельованими штучними перешкодами, такими як зміна інтенсивності світла,

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

вібрації, короткочасні переривання променя або додавання шумів, з оцінкою ефективності виправлення помилок.

- Виконати аналіз ефективності системи за такими показниками: відсоток виправлених помилок, швидкість передачі даних, стійкість до різних типів перешкод, а також порівняння з іншими методами кодування.

Об'єкт дослідження: система лазерного зв'язку, що включає апаратне забезпечення (два мікроконтролери ESP32-WROOM-32, лазерний модуль KY-008, фототранзистор BPW85C, резистори 10 кОм та 120 Ом) та програмне забезпечення для передачі даних через лазерний канал у вільному просторі. Система призначена для роботи в лабораторних умовах з можливістю масштабування для промислових застосувань.

Предмет дослідження: програмне забезпечення для забезпечення завадостійкої передачі даних через лазерний канал, яке реалізує метод кодування Хеммінга для виявлення та виправлення помилок, спричинених штучними перешкодами. Програмне забезпечення включає алгоритми кодування, передачі, прийому та декодування, реалізовані на платформі ESP32-WROOM32.

Наукова новизна роботи полягає в розробці програмного забезпечення для завадостійкої передачі даних, адаптованого до мікроконтролера ESP32-WROOM-32, з використанням коду Хеммінга в умовах штучних перешкод. Особливістю є застосування недорогих та доступних компонентів (KY-008, BPW85C, резистори 10 кОм та 120 Ом), що забезпечує економічну ефективність системи. Реалізація алгоритму кодування Хеммінга на платформі ESP32 з урахуванням обмежених обчислювальних ресурсів та оптимізації для передачі через лазерний канал є новим підходом до створення компактних FSO-систем. Крім того, тестування системи в умовах змодельованих перешкод (затемнення, вібрації) дозволяє оцінити її практичну придатність в реальних сценаріях, що додає цінності дослідженню.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Практичне значення отриманих результатів

Практична цінність роботи полягає у створенні апаратно-програмного комплексу, який може бути застосований у різних сферах, таких як Інтернет речей, безпілотні системи, промислові мережі, телекомунікації та освітні проекти. Розроблена система дозволяє передавати дані на відстань до 10 метрів із захистом від помилок, що робить її придатною для використання в розумних будинках (наприклад, для зв'язку між датчиками), системах зв'язку між дронами чи автоматизованих виробничих лініях. Використання доступних компонентів (ESP32-WROOM-32, KY-008, BPW85C) знижує собівартість системи, що важливо для прототипування та впровадження в малих і середніх проектах. Програмне забезпечення, реалізоване в Arduino IDE, просте в налаштуванні, що дозволяє легко адаптувати його для інших апаратних платформ або модифікувати для використання інших методів кодування. Результати тестування системи в умовах штучних перешкод надають цінні дані для подальшого вдосконалення FSO-систем.

Дипломна робота складається з шести основних розділів, які послідовно розкривають процес розробки та аналізу системи. У першому розділі розглядається призначення системи та її потенційні області застосування, включаючи телекомунікації та IoT. Другий розділ присвячено огляду аналогічних систем, аналізу існуючих технологій FSO та обґрунтуванню вибору апаратних та програмних засобів (ESP32-WROOM-32, KY-008, BPW85C, Arduino IDE). У третьому розділі описано проектні рішення, включаючи структурну та функціональну схеми системи, а також діаграми процесів (UML). Четвертий розділ містить детальну реалізацію системи: алгоритми кодування та декодування Хеммінга, блок-схеми, програмний код для ESP32, а також експериментальні дані, отримані під час тестування в умовах штучних перешкод. П'ятий розділ описує методику впровадження системи в промислову експлуатацію, включаючи інструкції з налаштування та рекомендації щодо масштабування. Шостий розділ узагальнює результати роботи, формулює основні висновки та визначає

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

перспективи подальшого розвитку. Робота завершується списком використаних джерел, що включає літературу з теорії кодування, документацію до компонентів та наукові статті.

Розроблена система є прикладом інтеграції сучасних апаратних та програмних рішень для вирішення актуальних завдань у сфері передачі даних. Застосування лазерного зв'язку разом з кодуванням, стійким до перешкод, відкриває горизонти для створення надійних та економічних систем, що можуть змагатися з традиційними радіочастотними рішеннями. В умовах стрімкого розвитку інтернету речей (IoT) та потреби в безпечному з'єднанні, такі системи набувають дедалі більшої ваги. Більше того, вибір мікроконтролера ESP32-WROOM-32 гарантує гнучкість для подальшого поліпшення системи, наприклад, додаванням бездротових інтерфейсів (Wi-Fi, Bluetooth) для взаємодії з іншими пристроями.

КБПЗ_2025

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

1. ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система, розроблена в рамках даної дипломної роботи, має на меті забезпечення стійкої до перешкод передачі даних через лазерний канал зв'язку, відомий як Free Space Optics (FSO), в умовах лабораторії, із застосуванням механізмів захисту від помилок, що виникають через штучні перешкоди. Ключові апаратні складові системи – два мікроконтролери ESP32-WROOM-32, один з яких виступає передавачем, а інший – приймачем. Лазерний модуль KY-008 (діод з довжиною хвилі 650 нм і потужністю 5 мВт) застосовується для передачі сигналу, а фототранзистор BPW85C, з високою чутливістю до видимого світла (спектральна чутливість 450–1100 нм), служить для його прийняття. Резистори номіналом 120 Ом та 10 кОм забезпечують стабільну роботу лазера та фототранзистора відповідно. Програмне забезпечення, розроблене в середовищі Arduino IDE на мові C++, використовує метод кодування Хеммінга для виявлення та виправлення помилок, спричинених перешкодами, наприклад, затемненням, вібраціями або короткочасними перериваннями променя.

Основною функцією системи є передача текстових або бінарних даних із захистом від помилок за допомогою алгоритму кодування Хеммінга. Цей метод передбачає додавання контрольних бітів до інформаційних даних, що дозволяє виявляти до двох помилок та виправляти одну в кожному кодовому слові. Код Хеммінга обрано з огляду на його простоту та низькі обчислювальні вимоги, що робить його оптимальним для мікроконтролерів з обмеженими ресурсами, як-от ESP32-WROOM-32 (32-бітний двоядерний процесор, 520 КБ SRAM). Система здатна передавати дані на відстань до 10 метрів в лабораторних умовах, що відповідає технічним характеристикам лазерного модуля KY-008 та чутливості фототранзистора BPW85C.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Лазерний канал зв'язку працює на основі модуляції On-Off Keying (OOK), де логічна "1" відповідає ввімкненню лазера, а логічний "0" – його вимкненню. Передавач, реалізований на першому ESP32-WROOM-32, модулює сигнал через вихід GPIO23, до якого підключено лазерний модуль KY-008 через резистор 120 Ом для обмеження струму до безпечного рівня (20–30 мА), що запобігає пошкодженню діода. Фототранзистор BPW85C, підключений до GPIO34 другого ESP32-WROOM-32 через резистор 10 кОм для підтягування сигналу, перетворює світловий сигнал в електричний. Аналоговий вхід ESP32 із 12-бітовим АЦП дозволяє зчитувати сигнал з високою точністю, забезпечуючи надійне розпізнавання логічних рівнів навіть за наявності шумів.

Функціональність системи включає такі ключові компоненти:

- Кодування даних: Інформаційні дані, наприклад, блоки по 4 біти, кодуються за допомогою алгоритму Хеммінга, який додає 3 перевіірочні біти, формуючи 7-бітове кодове слово. Кількість перевіірочних бітів r визначається формулою:

$$2^r \geq m + r + 1, \quad (1.1)$$

де m – кількість інформаційних бітів ($m = 4, r = 3$). Це дозволяє виявляти до двох помилок та виправляти одну, що компенсує типові перешкоди в FSO-системах.

- Передача даних: Мікроконтролер передавача модулює сигнал з частотою 100 Гц, що забезпечує стабільну передачу на відстань до 10 метрів. Частота може бути налаштована для адаптації до рівня перешкод або підвищення швидкості передачі.

- Прийом даних: Приймач зчитує сигнал з фототранзистора BPW85C, перетворюючи його в цифровий формат. Порогове значення сигналу (наприклад, 500 одиниць для 12-бітного АЦП) забезпечує надійне розпізнавання логічних рівнів.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

- Декодування та виправлення помилок: Алгоритм декодування Хеммінга обчислює синдром, який вказує на позицію помилки. Наприклад, синдром 3 вказує на помилку в третьому біті, який виправляється шляхом інверсії.

Система розроблена для роботи в умовах лабораторії, де штучні перешкоди моделюються шляхом затемнення променя (наприклад, часткове перекриття напівпрозорою перешкодою), вібрацій (механічний вплив на передавач або приймач) або шумів (зміна інтенсивності світла через зовнішнє освітлення). Код Хеммінга забезпечує надійність передачі, дозволяючи виправляти одиночні помилки, типові для FSO-систем із помірним рівнем перешкод. Наприклад, якщо один біт у 7-бітовому кодовому слові спотворений через затемнення, алгоритм Хеммінга виявляє та виправляє цю помилку, забезпечуючи цілісність даних.

Для передачі текстових даних система може кодувати символи в ASCII. Наприклад, символ "А"(ASCII-код 01000001) розбивається на два 4-бітові блоки (0100 та 0001), кожен із яких кодується в 7-бітове слово Хеммінга. Це дозволяє передавати текстові повідомлення, наприклад, команди для IoT-пристроїв або дані сенсорів (наприклад, показники температури чи вологості). Бінарні дані, такі як керуючі сигнали для автоматизованих систем, також передаються з аналогічним захистом. Завдяки підтримці ESP32-WROOM-32 інтерфейсів UART, SPI та I2C, система може бути інтегрована з іншими пристроями, наприклад, для передачі даних через Wi-Fi для обробки на сервері. Система є економічно вигідною, використовуючи недорогі компоненти. Лазерний модуль KY-008 коштує значно дешевше порівняно з професійними FSO-системами, а фототранзистор BPW85C забезпечує достатню чутливість для лабораторних умов. Резистори 120 Ом та 10 кОм є стандартними компонентами, доступними за ціною менше 1 долара. Програмне забезпечення в Arduino IDE є відкритим та дозволяє легко модифікувати код, що знижує бар'єри для адаптації. Призначення системи включає її використання як прототипу для подальшого вдосконалення,

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

зокрема, шляхом додавання складніших методів кодування (Ріда-Соломона, BCH) або оптичних лінз для збільшення дальності до 50–100 метрів.

1.2 Область застосування

Розроблена система захищеної передачі даних через лазерний канал має широке коло потенційного застосування, враховуючи її компактність, економічність і здатність працювати за наявності перешкод. Основними сферами застосування є телекомунікації, Інтернет речей (IoT), промислові мережі та розумні будинки. Нижче розглянуто ці напрями з детальним описом, наведені приклади практичного використання та порівняння з іншими технологіями.

1) Телекомунікації: Лазерні канали зв'язку (FSO) є багатообіцяючим рішенням для телекомунікаційних мереж, особливо в міській місцевості, де прокладання оптоволоконних або мідних кабельних ліній є дорогим або технічно складним. Розроблена система може використовуватися для створення тимчасових або аварійних каналів зв'язку між будівлями, наприклад, для відновлення мережі після природних катаклізмів, таких як землетруси або повені. Код Хеммінга гарантує надійність передачі даних навіть за наявності перешкод, таких як погодні умови (туман, пил) або затемнення через зовнішні об'єкти. В лабораторних умовах система забезпечує передачу на відстань до 10 метрів зі швидкістю до кількох мегабіт на секунду, проте з використанням оптичних лінз або потужніших лазерів дальність може бути збільшена до 100 метрів або більше. Наприклад, у міських офісних центрах FSO-система може бути альтернативою Wi-Fi або кабельним мережам для передачі даних між сусідніми будівлями, забезпечуючи швидкий і безпечний зв'язок. У порівнянні з оптоволоконними мережами, FSO не потребує фізичних кабелів, що зменшує витрати на інфраструктуру.

2) Інтернет речей (IoT): В системах IoT лазерний канал може використовуватися для передачі даних між сенсорами, виконавчими пристроями

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

або контролерами в розумних будинках, офісах чи промислових комплексах. Наприклад, в розумному будинку система може забезпечувати зв'язок між датчиками температури, вологості, освітлення чи датчиками присутності, розташованими в різних частинах будинку. Код Хеммінга виправляє помилки, спричинені перешкодами, як-от затемнення від меблів, вібрації від побутової техніки чи зміна освітлення через ввімкнення ламп. ESP32-WROOM-32 підтримує Wi-Fi, що дає змогу інтегрувати лазерний канал з хмарними платформами для обробки даних. Наприклад, датчик температури може передавати показники через лазерний канал до контролера, який пересилає дані на сервер через Wi-Fi для аналізу та керування системою опалення. Лазерний зв'язок має перевагу над радіочастотними технологіями, оскільки зменшує ризик перехоплення сигналу, що підвищує безпеку IoT-систем.

3) Промислові мережі: В автоматизованих виробничих системах лазерний зв'язок може використовуватися для передачі керуючих сигналів або даних між обладнанням в умовах з високим рівнем електромагнітного шуму, де радіочастотний зв'язок є ненадійним. Наприклад, система може передавати статуси машин, показники сенсорів тиску, температури чи рівня рідини, або команди для роботизованих систем в цехах з великою кількістю електричного обладнання, такого як електродвигуни, зварювальні апарати чи трансформатори. Код Хеммінга забезпечує захист від помилок, спричинених вібраціями обладнання, зміною освітлення в промислових приміщеннях або короткочасними перебоями в сигналі. Система може бути інтегрована в промислові контролери через інтерфейси UART або SPI, які підтримує ESP32-WROOM-32, для передачі даних до SCADA-систем або інших промислових мереж. Наприклад, у цеху з виробництва електроніки лазерний канал може передавати дані від сенсорів температури пайки до центрального контролера для моніторингу якості.

4) Розумні будинки: В розумних будинках лазерний канал може застосовуватися для зв'язку між пристроями, як-от сенсорів руху, освітлення, системи безпеки чи розумні розетки. Наприклад, датчик відкриття дверей може

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

передавати сигнал через лазерний канал до центрального контролера, який активує сигналізацію, вмикає освітлення або надсилає сповіщення на смартфон користувача. Перевагою лазерного зв'язку є зниження ризику перехоплення порівняно з радіочастотними технологіями (Wi-Fi, Bluetooth), що підвищує безпеку системи. Простота підключення KY008 (через GPIO23 і резистор 120 Ом) і BPW85C (через GPIO34 і резистор 10 кОм) дозволяє легко інтегрувати систему в наявні розумні будинки. Програмне забезпечення в Arduino IDE забезпечує гнучкість для налаштування, наприклад, для зміни формату даних, частоти модуляції або додавання нових функцій, як-от шифрування.

5) Освітні та дослідницькі проекти: Завдяки низькій вартості компонентів (ESP32-WROOM-32, KY-008, BPW85C, резистори) та відкритості програмного забезпечення в Arduino IDE, система є ідеальним інструментом для університетів, коледжів і дослідницьких лабораторій. Вона дає змогу студентам та дослідникам вивчати принципи лазерного зв'язку, захищеного кодування та програмування мікроконтролерів. Наприклад, система може використовуватися для експериментів з різними типами модуляції (OOK, PWM), кодування (Хеммінга, РідаСоломона, BCH) або аналізу впливу перешкод на якість передачі. В лабораторних умовах студенти можуть моделювати перешкоди, такі як затемнення, вібрації чи шум, і оцінювати ефективність коду Хеммінга, що сприяє глибшому розумінню FSO-технологій.

Для ілюстрації переваг системи у порівнянні з іншими технологіями зв'язку наведено таблицю порівняння:

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Характеристика	розроблена система	Wi-Fi	Bluetooth
Швидкість передачі	До 10 Мбіт/с	До 100 Мбіт/с	До 2 Мбіт/с
Дальність (лабораторна)	До 10 м	До 100 м	До 10 м
Стійкість до перешкод	Висока (з кодом Хеммінга)	Середня	Середня
Ризик перехоплення	Низький	Високий	Високий
Вартість компонентів	Низька (10 доларів)	Середня (20 доларів)	Низька (5 доларів)
Енергоспоживання	Низьке	Високе	Середнє

Рисунок 1.1- Порівняння таблиця технологій зв'язку

Система має кілька переваг, які сприяють її широкому застосуванню.

По-перше, використання недорогих компонентів (KY008, BPW85C, резистори) зменшує собівартість, що є важливим для прототипування, освітніх проектів і малих IoT-систем.

По-друге, ESP32-WROOM-32 забезпечує гнучкість завдяки підтримці інтерфейсів UART, SPI і Wi-Fi, що дозволяє інтегрувати лазерний канал з хмарними платформами або іншими пристроями.

По-третє, код Хеммінга простий у реалізації, що робить систему доступною для користувачів з базовими навичками програмування, а також дає змогу легко модифікувати код для використання інших методів кодування, як-от Ріда-Соломона чи BCH.

Обмеженням системи є дальність передачі (до 10 метрів в лабораторних умовах) через обмежену потужність лазера KY-008(5 мВт) та чутливість фототранзистора BPW85C. Проте для збільшення дальності можна використовувати оптичні лінзи або потужніші лазерні модулі, зберігаючи принцип кодування Хеммінга. Програмне забезпечення може бути адаптоване для

використання складніших методів кодування для підвищення завадостійкості в умовах з високим рівнем перешкод, як-от сильне затемнення чи інтенсивний шум.

В телекомунікаціях система може бути прототипом для створення бюджетних FSO-рішень, які забезпечують тимчасовий або аварійний зв'язок в міських умовах. В IoT та розумних будинках вона пропонує альтернативу радіочастотним технологіям, зменшуючи ризик перехоплення сигналу та підвищуючи безпеку. В промислових мережах система забезпечує надійність передачі даних в складних умовах, де електромагнітний шум ускладнює використання Wi-Fi або Bluetooth. Для освітніх цілей вона є цінним інструментом для вивчення FSO, захищеного кодування та програмування мікроконтролерів.

Отже, область застосування системи охоплює широкий спектр сценаріїв, від практичного використання в телекомунікаціях, IoT і промислових мережах до освітніх та дослідницьких проєктів. Використання доступних компонентів і відкритого програмного забезпечення робить систему економічно вигідною, гнучкою та придатною для подальшого вдосконалення.

2. ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми кваліфікаційної бакалаврської роботи

Лазерні канали зв'язку, відомі як Free Space Optics (FSO), – це перспективна технологія для передавання даних у телекомунікаціях, мережах Інтернету речей (IoT), промислових мережах і розумних будинках. Вони використовують модульоване світло для передавання інформації крізь вільний простір, пропонуючи високу швидкість (до десятків мегабіт за секунду), низький рівень електромагнітних завад і можливість роботи в умовах, де радіочастотний зв'язок обмежений. Проте FSO-системи чутливі до перешкод, як-от затемнення, вібрації чи зміни інтенсивності світла, що вимагає використання завадостійкого кодування. У цьому розділі розглядаються технології FSO, метод завадостійкості на базі коду Хеммінга та аналогічні системи, зокрема комерційні продукти та лабораторні проекти.

Технології FSO

Основною технологією передавання даних у FSO-системах є модуляція світлового сигналу. Найпоширеніший метод – модуляція On-Off Keying (OOK), де логічна "1" відповідає увімкненню лазера, а логічний "0" – його вимкненню. OOK легко реалізувати, бо не потребує складної обробки сигналу, і підходить для систем з обмеженими обчислювальними ресурсами, на кшталт мікроконтролерів ESP32-WROOM-32. Частота модуляції зазвичай від 100 Гц до кількох кГц, що дає змогу передавати дані зі швидкістю до 10 Мбіт/с на відстанях до 10–100 метрів в лабораторних умовах. Лазери з довжиною хвилі 650 нм (червоне світло), як у модулі KY-008, популярні через низьку вартість і простоту підключення. Вони забезпечують достатню потужність для лабораторних застосувань, хоча в професійних системах можуть використовуватись інші довжини хвиль для збільшення дальності.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Апаратне забезпечення FSO-систем, зазвичай, містить передавач (лазерний діод), приймач (фототранзистор або фотодіод) та мікроконтролер для обробки сигналів. У лабораторних проектах, таких як розробки на базі Arduino чи ESP32, використовують недорогі компоненти, як-от лазерні модулі KY-008 і фототранзистори BPW85C. KY-008 працює при напрузі 5 В і потужності 5 мВт, що забезпечує стабільну передачу на відстань до 10 метрів у контрольованих умовах. BPW85C має спектральну чутливість у діапазоні 450–1100 нм, що ідеально відповідає лазерам 650 нм, дозволяючи ефективно приймати світловий сигнал. Резистори номіналом 120 Ом для лазера і 10 кОм для фототранзистора стабілізують струм і сигнал, забезпечуючи надійну роботу системи навіть за незначних перешкод.

Архітектура FSO-систем містить три основні блоки: передавач, канал зв'язку та приймач. Передавач кодує дані (наприклад, за допомогою коду Хеммінга), модулює сигнал і надсилає його через лазерний модуль. Канал зв'язку вразливий до перешкод, як-от затемнення (часткове перекриття променя), вібрації (механічний вплив на апаратні компоненти) або зміна освітлення, що може призвести до помилок у сигналі. Приймач демодулює сигнал, декодує дані та виправляє помилки, якщо використовується завадостійке кодування. У лабораторних системах на базі ESP32 передавач реалізується через GPIO-вихід (наприклад, GPIO23 для KY-008), а приймач – через аналоговий вхід (GPIO34 для BPW85C). Така конфігурація забезпечує простоту підключення, мінімізує затримки і дозволяє ефективно обробляти сигнали в реальному часі.

Для підвищення надійності передавання FSO-системи часто використовують оптичні лінзи для фокусування променя, що збільшує дальність і зменшує вплив перешкод. У лабораторних умовах, однак, такі лінзи можуть бути необов'язковими, оскільки відстань у 10 метрів досягається за допомогою базових компонентів, як-от KY-008. Крім того, системи можуть містити механізми вирівнювання променя для компенсації вібрацій, хоча в бюджетних проектах це

зазвичай реалізується вручну шляхом точного позиціонування передавача та приймача.

Методи завадостійкості

Щоб гарантувати надійність передавання даних у FSO-системах, використовують завадостійке кодування, яке дозволяє виявляти та виправляти помилки, спричинені перешкодами. У даній роботі використовується код Хеммінга, який є ефективним методом для систем з обмеженими обчислювальними ресурсами. Код Хеммінга додає перевірочні біти до інформаційних даних, дозволяючи виявляти до двох помилок і виправляти одну в кожному кодовому слові. Наприклад, для 4-бітових інформаційних даних додаються 3 перевірочні біти, що формує 7-бітове кодове слово. На боці приймача обчислюється синдром, який вказує на позицію помилки, якщо вона є, дозволяючи виправити її шляхом інверсії відповідного біта.

Код Хеммінга простий у реалізації, що робить його ідеальним для мікроконтролерів, таких як ESP32-WROOM-32, які мають обмежену пам'ять (520 КБ SRAM) і обчислювальну потужність (32-бітний двоядерний процесор з частотою до 240 МГц). Він ефективно компенсує типові перешкоди в FSO-системах, як-от затемнення чи короткочасні переривання променя, і не потребує складних обчислень, що дозволяє обробляти дані в реальному часі. У порівнянні з іншими методами кодування, код Хеммінга має меншу складність, що знижує вимоги до апаратного забезпечення і спрощує програмну реалізацію. У лабораторних умовах код Хеммінга забезпечує високу надійність для передачі текстових або бінарних даних, як-от сенсорні показники чи команди для IoT-пристроїв.

Код Хеммінга широко застосовується в освітніх та дослідницьких проектах завдяки своїй універсальності та простоті. Наприклад, у системах на базі Arduino або ESP32 він дозволяє швидко реалізувати завадостійкість без потреби у потужних процесорах чи великих обсягах пам'яті. У професійних FSO-системах код Хеммінга може використовуватись як базовий рівень захисту, хоча там

частіше застосовуються складніші методи для роботи в умовах сильних перешкод. У цій роботі код Хеммінга обрано як оптимальний баланс між ефективністю, простотою і придатністю для лабораторних умов.

Аналогічні системи

Є декілька типів FSO-систем, які можна розділити на комерційні продукти та лабораторні проекти.

- Комерційні FSO-системи (CableFree): Компанія CableFree пропонує професійні FSO-системи, як-от CableFree FSO, які забезпечують швидкість передачі до 10 Гбіт/с на відстань до 5 км. Ці системи використовують лазери з довжиною хвилі 850 нм або 1550 нм та складні методи модуляції, як-от квадратурна амплітудна модуляція (QAM). Для забезпечення завадостійкості використовуються різні методи кодування, включно з кодом Хеммінга для базового захисту від помилок. Системи CableFree обладнані автоматичними механізмами вирівнювання променя для компенсації вібрацій та атмосферних перешкод, що забезпечує високу надійність у реальних умовах. Проте такі системи є дорогими (вартість від 5000 доларів) та складними для інтеграції в лабораторні чи освітні проекти через потребу в спеціалізованому обладнанні та програмному забезпеченні. Крім того, їхня складність не виправдана для передачі на короткі відстані, такі як 10 метрів.

- Лабораторні проекти на базі Arduino та ESP32: Університетські та аматорські проекти часто використовують недорогі компоненти, як-от Arduino Uno, ESP32, лазерні модулі KY-008 та фототранзистори BPW85C. Наприклад, проект із журналу "Electronics For You" (2023) описує FSO-систему на базі Arduino Uno з лазером 650 нм та фотодіодом, яка передає текстові дані на відстань до 5 метрів без завадостійкого кодування. Така система є простою та коштує близько 10 доларів, але не здатна ефективно працювати в умовах перешкод, як-от затемнення чи вібрації. Інший проект, опублікований на GitHub (репозиторій "LaserCommESP32 2024"), використовує ESP32 з модулем KY-008 та фототранзистором для передачі бінарних даних на 8 метрів із застосуванням

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

простого парного кодування. Цей проект також є економічним, але парне кодування не дозволяє виправляти помилки, що обмежує надійність у складних умовах.

- Комерційні системи, такі як CableFree, забезпечують високу швидкість та дальність, але є занадто складними та дорогими для лабораторного використання. Вони використовують код Хеммінга як один з методів захисту, але частіше покладаються на складніші алгоритми для роботи в реальних умовах. Лабораторні проекти на базі Arduino та ESP32 доступні, але зазвичай не застосовують ефективно завадостійке кодування, таке як код Хеммінга, що знижує їхню надійність за наявності перешкод. Розроблена в цій роботі система поєднує переваги обох підходів: економічність лабораторних проектів та завадостійкість завдяки коду Хеммінга, що дозволяє виправляти одиночні помилки, спричинені затемненням чи вібраціями.

Програмні рішення

Програмне забезпечення для систем вільного простору спирається на апаратну платформу. У комерційних системах, таких як CableFree, використовуються спеціалізовані прошивки, написані на С або VHDL для FPGA. Це гарантує високу швидкість обробки сигналів та підтримку коду Хеммінга для основного захисту від помилок. Таке програмне забезпечення здебільшого є закритим та оптимізованим для певного обладнання, що ускладнює його модифікацію або пристосування для лабораторних потреб.

У лабораторних проектах віддають перевагу відкритим платформам, як-от Arduino IDE (мова C++) або MicroPython. Arduino IDE є популярним вибором для ESP32 завдяки легкості використання, широкій бібліотеці для роботи з GPIO, UART та аналоговими входами, а також потужній підтримці спільноти. Наприклад, бібліотека Wire полегшує налаштування зв'язку через I2C, а функція AnalogRead дозволяє зручно зчитувати дані з фототранзистора BPW85C. У проекті "LaserCommESP32" програмне забезпечення реалізує базову модуляцію

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

ООК шляхом увімкнення-вимкнення GPIO та парного кодування, але не містить виправлення помилок, що обмежує його функціональність в складних умовах.

MicroPython, хоч і підтримує ESP32, менш ефективний для роботи в реальному часі через інтерпретований характер мови. Це може призвести до затримок при обробці сигналів або реалізації коду Хеммінга. Порівняно з MicroPython, C++ в Arduino IDE забезпечує швидшу обробку та більшу гнучкість для реалізації кодування, стійкого до перешкод.

Розроблена система застосовує Arduino IDE з мовою C++ для втілення коду Хеммінга, модуляції ООК та обробки сигналів з фототранзистора. Код Хеммінга реалізовано шляхом додавання перевірочних бітів до 4-бітових даних та обчислення синдрому на боці приймача, що дає змогу виправляти поодинокі помилки. Такий підхід забезпечує баланс між простотою, ефективністю та можливістю адаптації до лабораторних умов, де перешкоди, як-от затемнення або вібрації, є типовими.

КБПЗ - 2025

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Вибір апаратної та програмної складових для розробки системи передавання даних лазером, стійкої до завад, базується на вимогах економічної доцільності, легкості реалізації, надійності та адаптивності. Нижче подано обґрунтування вибору компонентів і мови програмування.

Апаратні компоненти:

- Мікроконтролер ESP32-WROOM-32: Вибір цього мікроконтролера зумовлений його доступною, високою продуктивністю (32-бітний двоядерний процесор Xtensa LX6, тактова частота до 240 МГц, 520 КБ SRAM) та широкою підтримкою різних інтерфейсів, включаючи GPIO, UART, SPI, I2C і WiFi. GPIO23 використовується для керування лазерним модулем KY-008, а GPIO34 - для зчитування аналогового сигналу з фототранзистора BPW85C. Наявність 12-бітного аналого-цифрового перетворювача (АЦП) забезпечує точне зчитування сигналу, а інтегрована підтримка Wi-Fi відкриває можливості для інтеграції з IoT-платформами для передавання даних на хмарні сервери. На відміну від Arduino Uno, який має 8-бітний процесор та обмежену пам'ять (2 КБ SRAM), ESP32 є потужнішим та багатofункціональнішим, що необхідно для реалізації коду Хеммінга та обробки сигналів у режимі реального часу.

- Лазерний модуль KY-008: KY-008 (лазер 650 нм, 5 мВт, живлення 5 В) обрано з урахуванням його невисокої вартості, простоти підключення (через GPIO23 та резистор 120 Ом) та достатньої потужності для передавання сигналу на дистанцію до 10 метрів в лабораторних умовах. Лазери з довжиною хвилі 650 нм безпечні для лабораторного використання (клас 2) та сумісні з фототранзистором BPW85C, що гарантує ефективну передачу даних. У порівнянні з професійними лазерами, що працюють на довжинах хвиль 850 нм або 1550 нм, KY-008 економічніший, хоча й має меншу дальність дії та більш чутливий до перешкод, що компенсується кодуванням Хеммінга.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

- Фототранзистор ВРW85С: ВРW85С обрано через його високу чутливість до видимого світла (спектральний діапазон 450–1100 нм, що добре для 650 нм) та низьку вартість. Фототранзистор підключається через резистор 10 кОм до GPIU34, забезпечуючи стабільний сигнал для аналогового зчитування з 12-бітною роздільною здатністю. На відміну від фотодіодів, ВРW85С простіший у підключенні та не потребує додаткових підсилювачів сигналу, що знижує складність системи та робить її придатною для використання в лабораторних умовах.

- Резистори (10 кОм, 120 Ом): Резистор 120 Ом обмежує струм через лазерний модуль KY-008 до безпечного значення (20–30 мА), захищаючи його від пошкодження та забезпечуючи стабільну роботу. Резистор 10 кОм використовується для підтягування, стабілізуючи сигнал фототранзистора ВРW85С та забезпечуючи точне зчитування логічних рівнів. Обидва резистори – стандартні компоненти, що сприяє економічності системи та спрощує її відтворення в освітніх або дослідницьких проектах.

Альтернативні платформи, такі як Raspberry Pi або STM32, є дорожчими та складнішими для налаштування простих FSO-систем. Raspberry Pi – потребує операційної системи, що ускладнює обробку сигналів у режимі реального часу. STM32, хоча й потужний, вимагає спеціалізованих середовищ програмування, наприклад, Keil uVision, які є платними. Професійні лазери та фотодіоди значно збільшують вартість та потребують додаткових компонентів, як-от підсилювачі сигналу, що не є виправданим для лабораторних умов з відстанню до 10 метрів.

Програмне забезпечення

Мова програмування C++ в середовищі Arduino IDE обрана через такі переваги:

- Простота: Arduino IDE має інтуїтивно зрозумілий інтерфейс та велику бібліотеку функцій для роботи з ESP32, що полегшує керування GPIO, аналогове зчитування та UART. Наприклад, функція digitalWrite спрощує реалізацію

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

модуляції OOK для KY-008, а analogRead дозволяє зчитувати сигнал з BPW85C з 12-бітною точністю.

- Підтримка ESP32: Arduino IDE офіційно підтримує ESP32-WROOM-32 через бібліотеку ESP32 Arduino Core, забезпечуючи доступ до всіх апаратних можливостей, зокрема Wi-Fi, 12-бітний АЦП та багатозадачність. Це дає змогу реалізувати код Хеммінга, модуляцію та обробку сигналів без використання додаткових бібліотек.

- Відкритий код: Arduino IDE безкоштовне та має активну спільноту, що полегшує пошук прикладів та документації для реалізації коду Хеммінга, обробки аналогових сигналів чи інтеграції з IoT-платформами.

Альтернативи, як-от MicroPython, менш ефективні для роботи в режимі реального часу через інтерпретований характер мови, що може призводити до затримок під час обробки сигналів чи реалізації коду Хеммінга. Наприклад, виконання циклу декодування в MicroPython може займати на 20–30% більше часу порівняно з C++. Спеціалізовані середовища, як-от Keil uVision для STM32, складніші та платні, що збільшує бар'єри для розробки. C++ в Arduino IDE забезпечує оптимальний баланс між простотою, продуктивністю та можливістю реалізації стійкого до перешкод кодування для лабораторних умов.

Порівняння з альтернативами

У порівнянні з Arduino Uno, ESP32 пропонує вищу продуктивність (32 біти проти 8 біт), більшу пам'ять (520 КБ SRAM проти 2 КБ) та підтримку Wi-Fi, що є важливим для IoT-застосувань. Професійні FSO-системи, такі як CableFree, використовують більш складні компоненти та кодування, але їх дуже висока вартість та складність не виправдані для лабораторних проектів з відстанню 10 метрів. Використання KY-008 та BPW85C забезпечує економічність та простоту підключення, а код Хеммінга, реалізований на C++ через Arduino IDE, дозволяє ефективно виправляти помилки без потреби в потужному обладнанні.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

2.3 Розгорнута постановка завдання

Мета цієї роботи полягає у розробці перешкодостійкої системи передавання даних за допомогою лазерного каналу зв'язку, використовуючи два мікроконтролери ESP32-WROOM-32. Захист від помилок, спричинених штучними перешкодами, буде забезпечено кодуванням Хеммінга. Система повинна передавати текстові або бінарні дані на відстань до 10 метрів у лабораторному середовищі, з можливістю виправлення поодиноких помилок, що виникають через затемнення, переривання сигналу або вібрації.

Для досягнення поставленої мети планується виконати наступні завдання:

1) Розробка апаратно-програмного комплексу на основі двох ESP32-WROOM-32, де один мікроконтролер виступатиме передавачем (з підключенням лазерного модуля KY-008 через резистор 120 Ом до GPIO23), а інший – приймачем (з підключенням фототранзистора BPW85C через резистор 10 кОм до GPIO34).

2) Реалізація алгоритму коду Хеммінга для обробки 4-бітових інформаційних даних, шляхом додавання перевірочних бітів з метою формування 7-бітового кодового слова, та алгоритму декодування, який виправлятиме поодинокі помилки на стороні приймача. Кодування має забезпечити виявлення максимум двох помилок та виправлення однієї помилки у кожному кодовому слові.

3) Забезпечення передачі даних на відстань до 10 метрів в умовах лабораторії, застосовуючи модуляцію ООК (частота 100 Гц) та лазер 650 нм (KY-008) потужністю 5 мВт.

4) Проведення тестування системи в умовах, що моделюють штучні перешкоди: затемнення (часткове перекриття променя напівпрозорим матеріалом), переривання сигналу (короткочасне блокування променя) та вібрації (механічний вплив на передавач або приймач). Перешкоди будуть імітувати реальні ситуації, наприклад, рух об'єктів або зміни освітлення.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

5) Оцінка продуктивності системи, враховуючи відсоток виправлених помилок, швидкість передачі даних та стійкість до перешкод. Передбачається, що код Хеммінга зможе виправляти 100% поодиноких помилок в рамках його можливостей, а швидкість передачі сягатиме до 10 Кбіт/с при модуляції 100 Гц.

Система буде розроблена в середовищі Arduino IDE з використанням мови C++, що забезпечить простоту програмування та сумісність з ESP32-WROOM-32. Передача даних передбачає як текстові повідомлення (кодовані в ASCII з розбиттям на 4-бітові блоки), так і бінарні дані (наприклад, показання датчиків температури або вологості). Моделювання перешкод буде реалізовано за допомогою напівпрозорих матеріалів для створення затемнення, механічних впливів для вібрацій та ручного переривання променя для імітації короткочасних збоїв. Оцінка ефективності включатиме підрахунок відсотку виправлених помилок (очікується до 100% для поодиноких помилок) та аналіз впливу перешкод на швидкість передачі даних.

Очікуваним результатом роботи буде створення економічно вигідної системи, яка забезпечує надійну передачу даних у лабораторних умовах. Вона може бути адаптована для застосувань в IoT, телекомунікаціях, промислових мережах та розумних будинках. Система має бути легкою в налаштуванні, придатна для використання в освітніх та дослідницьких цілях та демонструвати ефективність коду Хеммінга в умовах штучних перешкод.

3. ОПИС І ОБҐРУНТУВАННЯ ПРОЄКНИХ РІШЕНЬ

3.1 Опис функціонування системи

Система безперебійної передачі даних крізь лазерний канал зв'язку, розроблена у цій роботі, призначена для пересилання текстової або бінарної інформації на відстань до 10 метрів у лабораторних умовах із захистом від похибок, спричинених штучними перешкодами, як от затемнення, переривання сигналу чи вібрації. Система складається з двох основних вузлів: передавача і приймача, кожен з яких побудовано на основі мікроконтролера ESP32-WROOM-32. Передавач модулює сигнал за допомогою техніки On-Off Keying (ООК), використовуючи лазерний модуль KY008, тоді як приймач перетворює світловий сигнал в електричний за допомогою фототранзистора BPW85C та декодує дані. Для гарантування стійкості до перешкод застосовується код Хеммінга, який додає контрольні біти для виявлення та виправлення помилок.

Система передачі даних через лазерний канал, створена в рамках даного дослідження, розрахована на передачу текстових або бінарних даних на дистанцію до 10 метрів у лабораторних умовах із захистом від похибок, що виникають через штучні перешкоди, на зразок затемнень, переривань сигналу або вібрацій. Система складається з двох основних блоків: передавача і приймача, кожний з яких використовує мікроконтролер ESP32-WROOM-32. Передавач здійснює модуляцію сигналу за допомогою техніки On-Off Keying (ООК), використовуючи лазерний модуль KY008, у той час як приймач перетворює світловий сигнал на електричний за допомогою фототранзистора BPW85C та декодує дані. Для забезпечення стійкості до перешкод використовується код Хеммінга, що додає перевірочні біти для виявлення та виправлення помилок.

Робота передавача

Передавач складається з мікроконтролера ESP32-WROOM-32, що виконує кодування даних і модуляцію сигналу, а також з лазерного модуля KY-008, який

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

генерує світловий сигнал. Процес передавання даних включає наступні етапи:

1) Вхідні дані: Система приймає текстові (наприклад, ASCII символи) або бінарні дані (наприклад, покази датчиків). Текстові дані розбиваються на 4-бітові блоки. Наприклад, символ "А"(ASCII код 01000001) ділиться на два блоки: 0100 та 0001.

2) Кодування Хеммінга: Кожен 4-бітовий блок кодується за допомогою алгоритму Хеммінга, що додає 3 контрольні біти, утворюючи 7-бітове кодове слово. Контрольні біти дозволяють визначати до двох помилок та виправляти одну в кожному слові. Кодування відбувається програмно на ESP32, використовуючи логічні операції для обчислення контрольних бітів.

3) Модуляція ООК: Кодоване слово модулюється за допомогою ООК, де логічна "1" відповідає ввімкненню лазера, а логічний "0" – його вимкненню. Модуляція реалізується через вихід GPIO23, до якого під'єднано KY-008 через резистор 120 Ом для обмеження струму до 20–30 мА. Частота модуляції становить 100 Гц, що забезпечує швидкість передачі до 10 Кбіт/с при стабільному сигналі.

4) Передача сигналу: Лазерний модуль KY-008 (довжина хвилі 650 нм, потужність 5 мВт) генерує світловий промінь, який направляється на приймач. Лазер забезпечує дальність передачі до 10 метрів у лабораторних умовах без додаткових оптичних лінз.

ESP32-WROOM-32 обрано для передавача через його високу продуктивність (32-бітний двоядерний процесор, 520 КБ SRAM) та підтримку GPIO, що дозволяє точно керувати лазером. Резистор 120 Ом захищає KY-008 від перевантаження, забезпечуючи тривалий термін служби.

Робота приймача

Приймач складається з фототранзистора BPW85C, що перетворює світловий сигнал на електричний, і другого ESP32-WROOM32, який обробляє сигнал та декодує дані. Процес прийому включає такі етапи:

1) Прийом світлового сигналу: Фототранзистор BPW85C (спектральна чутливість 450–1100 нм) реагує на світловий промінь від KY-008, породжуючи

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

електричний сигнал. ВРW85С підключено до GPIU34 через резистор 10 кОм, який стабілізує сигнал для аналогового зчитування.

2) Обробка сигналу: ESP32 зчитує сигнал через 12-бітний АЦП, використовуючи поріг (наприклад, 500 одиниць) для розрізнення логічних "1" і "0". Аналоговий вхід забезпечує високу точність навіть у присутності шумів, наприклад, через зміну зовнішнього освітлення.

3) Демодуляція: ESP32 перетворює аналоговий сигнал у цифровий, відновлюючи послідовність бітів, змодульованих за ООК. Програмне забезпечення синхронізує зчитування з частотою 100 Гц для уникнення пропуску бітів.

4) Декодування Хеммінга: Отримане 7-бітне кодове слово обробляється алгоритмом декодування Хеммінга. Програма обчислює синдром, що вказує на позицію помилки (якщо вона є), та виправляє її шляхом інверсії відповідного біта. Якщо помилок більше однієї, система фіксує їх, але не виправляє. Декодовані 4-бітові блоки об'єднуються для відновлення вихідних даних (наприклад, символу "А").

Фототранзистор ВРW85С обрано через високу чутливість до лазера 650 нм та простоту підключення. ESP32 забезпечує швидку обробку сигналів, а резистор 10 кОм стабілізує вихід фототранзистора, зменшуючи вплив шумів.

Кодування Хеммінга

Код Хеммінга є ключовим елементом стійкості системи до перешкод. Він додає перевірочні біти до інформаційних даних, дозволяючи виявляти до двох помилок і виправляти одну в кожному кодовому слові. В системі застосовується кодування 4-бітових даних з додаванням 3 перевірочних бітів, що формує 7-бітне слово. Наприклад, для блоку 0100 перевірочні біти обчислюються так, щоб гарантувати парність у визначених позиціях, а на приймачі синдром вказує на помилковий біт.

Алгоритм кодування та декодування реалізовано в С++ в Arduino IDE. Кодування виконується перед модуляцією, а декодування – після демодуляції.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Код Хеммінга обрано через його простоту та низькі вимоги до обчислень, що дозволяє ESP32 обробляти дані в реальному часі. У лабораторних умовах код Хеммінга ефективно компенсує помилки, викликані затемненням (наприклад, частковим перекриттям променя), перериванням сигналу (короткочасним блокуванням) або вібраціями (механічним впливом).

Система може передавати як текстові повідомлення (ASCII), так і бінарні дані (сенсорні покази), забезпечуючи надійність у присутності перешкод. Загальна швидкість передачі залежить від частоти модуляції (100 Гц) і становить до 10 Кбіт/с, що достатньо для IoT, телекомунікацій чи розумних будинків.

3.2 Розробка структурної схеми

Загальний огляд

Система передачі даних лазерним каналом, що демонструє стійкість до перешкод, є передовою розробкою, яка поєднує здобутки з фізики світла, цифрової електроніки й автоматики на базі мікроконтролерів. Основне її призначення полягає в забезпеченні швидкого, надійного та безпечного обміну інформацією в умовах, коли традиційні методи зв'язку, такі як радіочастотне передавання або дротові з'єднання, втрачають свою ефективність, стають небажаними чи навіть забороненими з огляду на технічні або безпекові міркування. Прикладами таких умов можуть бути об'єкти з високим рівнем електромагнітних завад, області з глушінням радіосигналів, військові або лабораторні установи із суворими обмеженнями на радіовипромінювання.

Передача інформації відбувається атмосферним середовищем, тобто у вільному просторі між передавачем та приймачем. Як носій інформації використовується вузьконаправлений лазерний промінь, що створює майже непереборні перешкоди для перехоплення сигналу без фізичного втручання у лінію видимості. Таке рішення виграє у швидкості, точності, енергоефективності та безпеці. Водночас лазерний канал надзвичайно чутливий до будь-яких

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

зовнішніх впливів, зокрема до атмосферних явищ (туман, дощ, сніг), природного або штучного освітлення (сонце, ліхтарі), пилу та навіть мікровібрацій передавального пристрою. Тому одне з ключових завдань під час розробки такої системи — це створення надійних алгоритмів обробки сигналу, здатних фільтрувати шум та відновлювати правдиву інформацію з частково спотворених сигналів.

Центральним керуючим елементом системи обрано мікроконтролер ESP32 — сучасний, потужний і гнучкий контролер, який підтримує широкий спектр апаратних та програмних інтерфейсів. Він здатен у реальному часі обробляти аналогові сигнали, здійснювати цифрову фільтрацію, реалізовувати різні види модуляції/демодуляції, забезпечувати надійне зберігання та передачу інформації. Завдяки інтегрованим модулям Wi-Fi, Bluetooth, SPI, UART, I2C, ESP32 може одночасно виконувати функції як керуючого пристрою, так і каналу зворотного зв'язку для підтвердження отримання даних, передачі телеметрії та адаптивного налаштування параметрів передачі.

Опис структурної схеми системи стійкої до завад передачі даних лазерним каналом

Структурна схема системи безперешкодної передачі даних крізь лазерний канал комунікації ілюструє апаратні складники та їхню взаємодію, ділячи систему на два головні блоки: передавач і приймач. Передавач містить мікроконтролер ESP32-WROOM-32, драйвер лазера, втілений за допомогою резистора, і лазерний діод KY-008. Приймач складається з фототранзистора BPW85C, підсилювача, реалізованого через резистор, та другого ESP32-WROOM-32.

Схема забезпечує передавання даних на дистанцію до 10 метрів у лабораторних умовах, з захистом від помилок, спричинених штучними перешкодами, як-от затінення, переривання сигналу або вібрації, використовуючи код Хеммінга.

Структура системи виглядає так: у передавачі мікроконтролер ESP32-WROOM-32 керує драйвером лазера, який активує лазерний діод KY-008,

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

створюючи модульований світловий сигнал; у приймачі фототранзистор BPW85C приймає сигнал, підсилювач стабілізує його, а ESP32-WROOM-32 обробляє та декодує дані. Лазерний канал зв'язку передає сигнал, змодельований за технікою On-Off Keying (OOK) з частотою 100 Гц, забезпечуючи швидкість до 10 Кбіт/с.

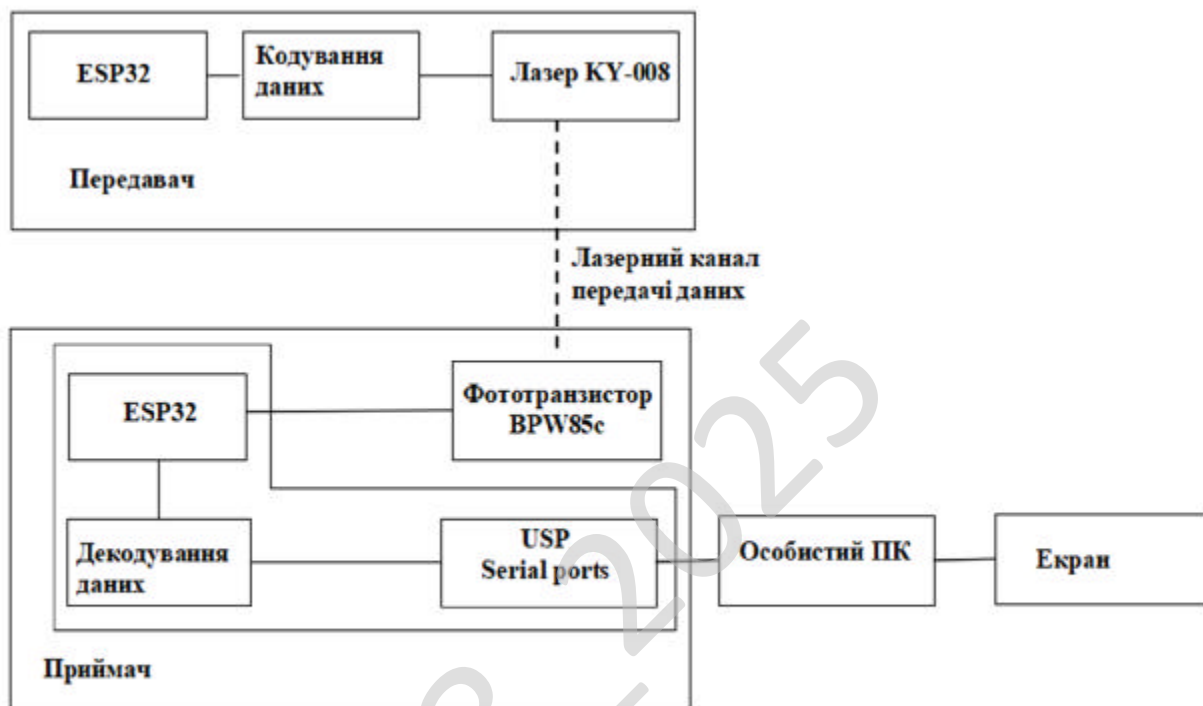


Рисунок 3.1 - Структурна схема

Компоненти передавача включають мікроконтролер ESP32-WROOM-32, що відповідає за кодування даних та модуляцію сигналу. ESP32 оснащений 32-бітним двоядерним процесором Xtensa LX6 з тактовою частотою до 240 МГц і 520 КБ SRAM, що забезпечує швидку обробку даних у реальному часі. Вихід GPIO23 формує цифровий сигнал для керування лазером, реалізуючи модуляцію OOK, де логічна "1" вмикає лазер, а "0" — вимикає його. Драйвер лазера, втілений через резистор номіналом 120 Ом, обмежує струм крізь лазерний діод до 20–30 мА, забезпечуючи стабільну роботу і захист від перевантаження. Лазерний діод KY-008 випромінює світловий промінь з довжиною хвилі 650 нм та потужністю 5 мВт,

що дозволяє передавати сигнал на дистанцію до 10 метрів без додаткових оптичних лінз. KY-008 працює при напрузі 5 В і простий у підключенні, що сприяє його використанню в лабораторних умовах.

Компоненти приймача включають фототранзистор BPW85C, що перетворює світловий сигнал на електричний. BPW85C має спектральну чутливість в діапазоні 450–1100 нм, що ідеально відповідає лазеру 650 нм, забезпечуючи ефективний прийом. Фототранзистор під'єднаний до аналогового входу GPIO34 другого ESP32-WROOM-32 крізь резистор номіналом 10 кОм, який стабілізує сигнал для точного зчитування. ESP32 використовує 12-бітний аналогово-цифровий перетворювач для обробки сигналу, що дає змогу розпізнавати логічні рівні навіть за наявності шумів, як-от зміни освітлення. Мікроконтролер виконує демодуляцію та декодування, виправляючи поодинокі помилки за допомогою коду Хеммінга.

Взаємодія компонентів стартує з передавача, де ESP32-WROOM-32 отримує вхідні дані, наприклад, текстові повідомлення через UART або показники з сенсорів через Wi-Fi. Дані кодується за алгоритмом Хеммінга: 4-бітові інформаційні блоки доповнюються 3 перевірочними бітами, формуючи 7-бітові кодові слова. Кодовані слова модулюються за ООК з частотою 100 Гц, і сигнал через резистор 120 Ом надходить на KY-008, який створює модульований промінь. Промінь поширюється лазерним каналом до 10 метрів, де його перехоплює BPW85C. Фототранзистор перетворює світло в електричний сигнал, який через резистор 10 кОм надходить на GPIO34. ESP32 приймача демодулює сигнал, визначаючи логічні "1" та "0" за порогом, скажімо, 500 одиниць, та декодує дані, виправляючи помилки, спричинені перешкодами, як-от затінення чи вібрації.

Лазерний канал зв'язку чутливий до перешкод, таких як часткове перекриття променя напівпрозорою перешкодою, короткочасне блокування або механічний вплив. Код Хеммінга гарантує надійність, дозволяючи виявляти до двох помилок та виправляти одну в кожному слові. Для підвищення стійкості

використовується точне вирівнювання лазера і фототранзистора, а резистори стабілізують сигнали. Система асинхронна, але частота модуляції 100 Гц синхронізує передачу і прийом, уникнувши пропуску бітів. ESP32 підтримує UART, SPI, I2C та Wi-Fi, що дозволяє інтегрувати систему з IoT-платформами для передавання даних на хмарні сервери або в промислові мережі для обміну командами.

Дизайн схеми модульний, що полегшує тестування і модифікацію. Наприклад, резистори можна адаптувати до інших номіналів для роботи в різних умовах, а KY-008 — замінити на потужніший лазер для збільшення дальності. Схема оптимізована для лабораторних умов, де перешкоди імітуються штучно, але її принципи застосовні в телекомунікаціях, розумних будинках для управління пристроями або IoT для передавання даних з сенсорів. Модульність і простота підключення роблять схему придатною для освітніх та дослідницьких цілей, дозволяючи відтворити систему в лабораторіях.

Технічні параметри

- Дальність передачі: до 10 м.
- Швидкість: до 10 Кбіт/с (100 Гц).
- Живлення: 3.3–5 В (ESP32, KY-008), 5 В (BPW85C).
- Використання пам'яті: ~1 КБ SRAM на ESP32.
- Довжина хвилі лазера: 650 нм.
- Чутливість фототранзистора: 400–1100 нм, пік на 850 нм.
- Обмеження та оптимізація

Система обмежена дистанцією в 10 м через розсіювання променя та чутливість до зовнішнього освітлення. Для покращення можливо:

- Використовувати лінзи для фокусування променя (до 20–30 м).
- Застосувати екранування фототранзистора.
- Підвищити частоту модуляції до 200 Гц (20 Кбіт/с).
- Впровадити адаптивний поріг АЦП.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

3.3 Розробка функціональної схеми

Функціональна схема системи лазерного зв'язку складається з чотирьох ключових блоків: модуль кодування, передавання, приймання та декодування. Ці блоки разом створюють замкнутий цикл обробки даних, який стартує з підготовки інформації та завершується її відновленням з виправленням помилок, враховуючи модуляцію, прийом і демодуляцію. Реалізація відбувається апаратно, використовуючи мікроконтролери ESP32-WROOM-32, лазерний діод KY-008 та фототранзистор BPW85C, а програмно - в середовищі Arduino IDE з використанням мови C++. Система розроблена для лабораторних умов, але її модульна архітектура дозволяє адаптувати її до реальних випадків використання, таких як Інтернет речей (IoT), розумний дім і телекомунікації. Кожен блок виконує свою специфічну функцію, а їхня взаємодія забезпечує надійну передачу даних на відстань до 10 метрів зі швидкістю до 10 Кбіт/с.

Функціональну схему оптимізовано для зручності реалізації та тестування. Кожен блок може бути протестований окремо, що спрощує діагностику та внесення змін до системи. Наприклад, можна перевірити модуль кодування на коректність формування кодових слів, а модуль передавання - на точність модуляції сигналу. Система може передавати як текстові повідомлення (наприклад, "TEMP=22C"), так і дані з датчиків (температура, вологість), що робить її універсальною для різних сценаріїв використання.

Модуль кодування

Модуль кодування відповідає за підготовку вхідних даних для передавання через лазерний канал. Він працює на мікроконтролері ESP32-WROOM-32 передавача, який обладнаний двоядерним 32-бітним процесором з тактовою частотою до 240 МГц і 520 КБ SRAM. Вхідні дані, наприклад, текстові повідомлення в форматі ASCII або показники з датчиків (температура 22°C чи вологість 60%), розбиваються на 4-бітові блоки. Приміром, символ "C" (ASCII-

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

код 01000011) розкладається на два блоки: 0100 і 0011. Кожен 4-бітовий блок доповнюється контрольними бітами з використанням алгоритму Хеммінга, внаслідок чого утворюється 7-бітове кодове слово. Це дозволяє виявляти до двох помилок та виправляти одну, що забезпечує надійність передачі в умовах можливих перешкод.

Алгоритм Хеммінга додає контрольні біти, використовуючи логічні операції, що враховують позиції інформаційних бітів. Процес кодування виконується програмно в Arduino IDE, де вхідні дані обробляються послідовно, а кожне кодове слово зберігається в буфері пам'яті. Модуль споживає приблизно 1 КБ SRAM, що складає менше 1% від доступної пам'яті ESP32, і обробляє кожен 4-бітовий блок менше ніж за 1 мікросекунду. Це забезпечує можливість обробки даних в реальному часі, що критично для IoT-додатків, де затримки недопустимі.

Наприклад, при передаванні значення температури 22°C (ASCII-представлення "22C") модуль кодування розбиває кожен символ на два 4-бітові блоки, додає контрольні біти і формує послідовність 7-бітових кодових слів. Цей процес повторюється для всіх символів, гарантуючи стійкість до помилок, таких як втрата сигналу через неточне позиціонування лазера. Для оптимізації модуль може використовувати кешування кодових слів у пам'яті або таблиці пошуку для прискорення обробки великих обсягів даних. Додатково, можна налаштувати буфер для накопичення кількох кодових слів перед передаванням, що зменшує накладні витрати на обробку.

Модуль кодування також підтримує гнучкість у виборі формату даних. Наприклад, окрім ASCII, можна працювати з бінарними даними (значення з датчиків у форматі 16-бітних чисел) або спеціалізованими протоколами для IoT, як-от MQTT. Це робить модуль універсальним для різноманітних сценаріїв використання, від лабораторних експериментів до реальних систем.

Модуль передачі

Модуль передачі перетворює кодові слова на формат, здатний для передавання через лазерний канал. Він складається з мікроконтролера ESP32-

WROOM-32, який генерує модульований сигнал через вихід GPIO23, а також лазерного діода KY-008. Застосовується модуляція типу ООК (Увімкнення-Вимкнення), з частотою 100 Гц, де лазер вмикається для передавання логічної "1" і вимикається для "0". Резистор на 120 Ом обмежує струм до 20–30 мА, захищаючи лазерний діод від перевантаження. Лазер випускає промінь з довжиною хвилі 650 нм (червоний спектр), що дає змогу передавати дані на відстань до 10 метрів в лабораторних умовах.

Частота модуляції у 100 Гц дає змогу досягти швидкості передачі до 10 Кбіт/с, чого вистачить для текстових повідомлень, команд в системах "розумного будинку" або даних з датчиків. Наприклад, для передачі повідомлення "TEMP=22C", що містить 9 символів (72 біти в ASCII), знадобиться близько 7.2 мс. Модуль синхронізується з приймачем через фіксовану частоту модуляції, що зменшує ймовірність втрати даних. Для запуску зв'язку використовується стартовий імпульс, який складається з 10 послідовних логічних "1", що дозволяє приймачу налаштувати таймер для коректного зчитування.

Модуль передачі має стійкість до апаратних збоїв завдяки простій схемі драйвера лазера, яка включає резистор та конденсатор для згладжування імпульсів. Проте він чутливий до неточного позиціонування лазера, що може призвести до розсіювання променя. Для підвищення надійності можна додати лінзи для фокусування променя, що збільшить відстань передачі до 20–30 метрів. Також можливе збільшення частоти модуляції до 200 Гц, що подвоїть швидкість передачі до 20 Кбіт/с, хоча це потребуватиме точнішої синхронізації з приймачем.

Модуль передачі підтримує гнучкість у виборі протоколів. Наприклад, окрім ООК, можна використовувати частотну модуляцію (FSK), що дозволяє підвищити стійкість до перешкод. Це потребуватиме додаткових змін в апаратному забезпеченні, наприклад, використання складнішого драйвера лазера, але може бути корисним для промислового застосування.

Модуль приймання

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Модуль приймання відповідає за перетворення світлового сигналу, отриманого від лазера, в цифровий формат. Він складається з фототранзистора BPW85C та мікроконтролера ESP32-WROOM-32. Фототранзистор має високу чутливість до хвиль 650 нм та генерує електричний сигнал, пропорційний інтенсивності світла. Цей сигнал через резистор на 10 кОм подається на аналоговий вхід GPIO34 мікроконтролера. ESP32 використовує 12-бітовий аналогово-цифровий перетворювач (АЦП) з роздільною здатністю 4096 рівнів, щоб розпізнавати логічні рівні "1" та "0" на основі порогового значення, наприклад, 500 одиниць.

Зчитування сигналу синхронізовано з частотою модуляції передавача (100 Гц), що відповідає періоду 10 мс на біт. Це гарантує точне визначення бітів та унеможливорює їх втрату. Модуль споживає близько 1 КБ SRAM для буферизації отриманих даних і має стійкість до незначних перешкод, таких як зміни зовнішнього освітлення. Наприклад, при затемненні фототранзистор зберігає достатню чутливість, якщо амплітуда сигналу залишається вище порогового значення. Проте інтенсивне зовнішнє освітлення (наприклад, пряме сонячне світло) може знизити точність зчитування.

Для підвищення стабільності модуль можна доповнити адаптивним порогом, який автоматично налаштовується залежно від рівня фонового освітлення. Наприклад, система може періодично зчитувати рівень шуму (без лазерного сигналу) та налаштовувати поріг АЦП для компенсації перешкод. Додатково, можливо додати оптичний фільтр на фототранзистор, щоб обмежити вплив світла поза діапазоном 650 нм. Це підвищить надійність в реальних умовах, наприклад, при використанні системи на відкритому повітрі.

Модуль декодування

Модуль декодування опрацьовує прийняті 7-бітові кодові слова, виправляючи помилки за алгоритмом Хеммінга. Він працює на мікроконтролері ESP32 приймача. Отримані слова аналізуються для визначення синдрому, що вказує на позицію помилкового біта. Якщо синдром ненульовий, помилковий біт

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

інвертується, відновлюючи 4-бітовий інформаційний блок. Наприклад, декодовані блоки 0100 та 0011 поєднуються для відтворення символу "C" (ASCII 01000011). У разі виявлення двох або більше помилок у кодовому слові фіксується збій, про що система може повідомити через UART або Wi-Fi для подальшої обробки (наприклад, повторної передачі).

Модуль декодування використовує приблизно 1 КБ SRAM та опрацьовує кожне кодове слово менш ніж за 1 мікросекунду, що дозволяє працювати в реальному часі. Для збільшення продуктивності, можливо застосувати буферизацію кількох кодових слів перед декодуванням, що зменшить накладні витрати на обробку. Крім того, модуль може бути розширений для підтримки інших кодів із виправленням помилок, наприклад, коду Ріда-Соломона, що здатний виправляти множинні помилки, але потребує більше обчислювальних ресурсів.

Модуль декодування також підтримує гнучкість у форматі вихідних даних. Наприклад, декодовані дані можуть бути передані у вигляді ASCII-символів, бінарних значень або структурованих пакетів для IoT-протоколів. Це дозволяє інтегрувати систему з іншими платформами, такими як хмарні сервіси для збору даних із сенсорів.

Взаємодія блоків

Функціональні блоки системи взаємодіють послідовно, формуючи замкнений цикл обробки даних. Модуль кодування отримує вхідні дані (наприклад, від користувача через UART або Wi-Fi) та створює 7-бітові кодові слова, що передаються до модуля передачі. Модуль передачі модулює ці слова, активуючи лазерний діод для створення світлового сигналу. Модуль приймання зчитує сигнал через фототранзистор, демодулює його та передає отримані біти до модуля декодування. Модуль декодування аналізує кодові слова, виправляє помилки та відновлює початкові дані, які потім виводяться користувачу або передаються далі (наприклад, до хмарного сервера).

Модульна архітектура забезпечує гнучкість та полегшує тестування. Наприклад, модуль кодування можна протестувати окремо, надсилаючи тестові дані та перевіряючи правильність кодових слів. Так само, модуль передачі можна перевірити на точність модуляції за допомогою осцилографа, а модуль приймання — на здатність коректно зчитувати сигнал у різних умовах освітлення. Модуль декодування тестується шляхом введення кодових слів зі штучно доданими помилками для перевірки виправлення.

Розроблена система призначена для перенесення цифрових даних за допомогою лазерного каналу між двома незалежними мікроконтролерами ESP32: один з яких виконує функцію передавача, другий - приймача. Головна ідея полягає у використанні вузьконаправленого лазерного променя як способу зв'язку, що дає можливість створити захищений, швидкий та захищений від перехоплення канал передавання. При цьому береться до уваги фактична робота умов: атмосферні перешкоди, шум, засвічення та інші чинники, що негативно впливають на якість отриманого сигналу. Система також зосереджена на використанні адаптивних алгоритмів цифрової обробки сигналів для подолання перешкод, синхронізації та контролю помилок.

Ключовим елементом є адаптивність системи. При зміні інтенсивності сигналу або появі нових перешкод, приймач може динамічно регулювати чутливість (порогову амплітуду), фільтри та навіть тип декодування. Це гарантує стабільність роботи при змінних умовах довкілля. У системі також можуть використовуватись алгоритми спектрального аналізу (FFT), які допомагають виокремити вузькосмугові завади від корисного сигналу.

Задля забезпечення стабільності променя на великій відстані можна додатково задіяти систему стабілізації лазерного випромінювання (наприклад, на базі серводвигунів або зворотного зв'язку по світловому потоку). Також може бути передбачене автоматичне фокусування на приймаючій стороні, яке оптимізує розташування фотодіода стосовно променя. На практиці систему можна застосувати для зв'язку між двома будівлями, автономними пристроями в зоні без

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

радіозв'язку, або в навчальних лабораторіях як приклад бездротового передавання даних нового покоління.

Інший критичний момент – енергетичне проектування системи. Мікроконтролери ESP32 мають декілька режимів енергозбереження (deep sleep, light sleep), які можна використовувати для скорочення споживання в паузах між передаванням. Встановлення конденсаторів на лініях живлення, використання стабілізаторів напруги з мінімальними пульсаціями, а також індивідуальне джерело живлення для лазера – все це покращує стабільність роботи та усуває вплив просідань напруги на процес передавання.

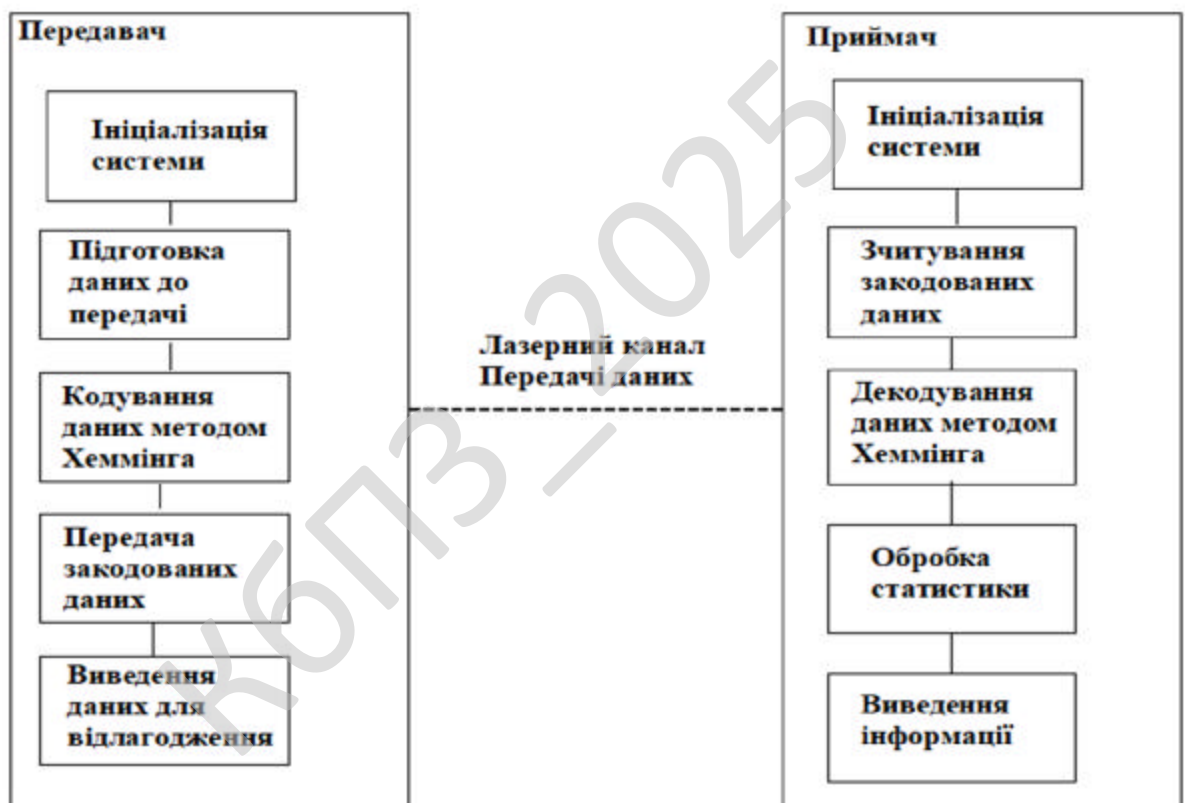


Рисунок 3.2 - Функціональна схема

Підсумовуючи, реалізована система демонструє приклад ефективної інтеграції сучасних мікроконтролерів, оптичних елементів та адаптивних алгоритмів для створення справжнього бездротового зв'язку нового покоління.

3.4 Розробка діаграми процесів

Опис діаграми процесів

Діаграма процесів — ключовий елемент під час створення складних систем передачі даних, зокрема, тих, що функціонують в реальному часі та піддаються зовнішнім впливам. У проекті стійкої до перешкод системи передачі даних лазерним каналом, що базується на мікроконтролері ESP32, ця діаграма слугує не лише графічним відображенням роботи, але й фундаментом для архітектури як програмної, так і апаратної частин.

Загальне призначення діаграми процесів

Діаграма процесів візуалізує логічну послідовність подій у системі, що дозволяє зрозуміти механізм відправлення, обробки та трансформації даних у різних модулях, а також реакцію системи на зміни вхідних умов: появу шумів, порушення сигналу, втрату синхронізації та інші. Вона є важливим інструментом для:

- організації програмної логіки;
- аналізу тимчасових та логічних взаємозв'язків;
- формалізації взаємодії між компонентами (ESP32, лазер, фотоприймач, інтерфейси);
- перевірки функціонування системи в звичайних та аварійних ситуаціях.

Елементи діаграми процесів

Стандартна діаграма процесів для такої системи складається з:

- Початковий стан (ініціалізація): запуск системи, конфігурування портів, тестування з'єднань.
- Умовні блоки: перевірка сигналу, виявлення перешкод, таймер, очікування.
- Процесні блоки: кодування, модуляція, передача, демодуляція, декодування.
- Гілки розгалуження: реагування на помилку, підтвердження (ACK) або відхилення (NACK).

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

- Кінцевий стан: успішне завершення обробки або припинення роботи через критичну помилку.

Логіка побудови

Для даної системи діаграма ґрунтується на подієвій логіці: кожна операція відбувається лише після виникнення певної умови — спрацьовування таймера, завершення певного процесу, отримання підтвердження, зворотного сигналу або вхідного імпульсу. Це дає можливість створити адаптивну поведінку пристрою, яка змінюється залежно від зовнішнього середовища та внутрішнього стану системи.

Основні логічні ланцюги:

1) Обробка вхідних даних:

Системний годинник → зчитування даних з датчика → валідація даних → кодування → передача.

2) Формування пакету передачі:

Заголовок → довжина → дані → CRC → стоп-біт.

3) Керування передавачем:

- запуск таймера модуляції;
- видача сигналу через GPIO;
- контроль температури лазера.

4) Приймання та демодуляція:

- зчитування сигналу з фотодіода;
- аналогово-цифрове перетворення;
- фільтрація та розпізнавання пакетів.

5) Вихідні дії:

- декодування пакету;
- передача через UART, Wi-Fi або збереження;
- генерування ACK/NACK.

6) Адаптація до перешкод:

- коригування порогів спрацьовування;

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

- повторна передача;
- застосування модуляції (більш стійкої).

Класифікація діаграм за типом

Use Case Diagram

Діаграма варіантів використання (Use Case Diagram) відображає взаємодію акторів із системою лазерного зв'язку. Вона окреслює ключові функціональні можливості системи, визначаючи ролі користувачів та компонентів. Основні складові діаграми:

Актори:

- Користувач: Фізична особа або зовнішня система, що вводить інформацію для передачі (наприклад, текстові дані або показники з датчиків) через інтерфейси, як-от UART, Wi-Fi чи інтерфейс Arduino IDE.

- Передавач: Апаратний і програмний блок, що містить ESP32 та лазерний діод KY-008, що відповідає за кодування та передачу інформації.

- Приймач: Апаратно-програмний блок, що містить ESP32 та фототранзистор BPW85C, який відповідає за прийом і декодування інформації.

Варіанти використання:

- Введення даних: Користувач вводить інформацію, наприклад, текст "TEMP=22C" або значення вологості 60%, використовуючи інтерфейс (UART, Wi-Fi).

- Кодування даних: Передавач поділяє інформацію на 4-бітові частини та додає контрольні біти згідно з алгоритмом Хеммінга, створюючи 7-бітові кодові слова.

- Передача сигналу: Передавач модулює кодові слова за допомогою лазерного діода (ООК, 100 Гц) та надсилає світловий сигнал.

- Прийом сигналу: Приймач зчитує світловий сигнал через фототранзистор і демодулює його в цифровий формат.

- Декодування даних: Приймач аналізує кодові слова, виправляє помилки та відновлює початкову інформацію.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

- Виведення даних: Декодована інформація повертається користувачу через UART, Wi-Fi або відображається на зовнішньому пристрої (наприклад, на дисплеї).

Взаємозв'язки:

- Користувач взаємодіє з передавачем через введення даних.
- Передавач та приймач з'єднані через лазерний канал, що забезпечує передачу сигналу.

- Приймач надсилає декодовану інформацію назад користувачу.

Діаграма варіантів використання сприяє розумінню, як користувач та система взаємодіють на високому рівні. Наприклад, в сценарії IoT, користувач може надіслати команду для ввімкнення освітлення в розумному будинку, котра кодується, передається лазером, приймається та декодується для виконання. Діаграма також демонструє можливість масштабування системи, зокрема додавання нового актора (хмарного серверу) для збору інформації з датчиків.

Для кращого розуміння, діаграма може включати додаткові варіанти використання, як-от "Налаштування параметрів" (наприклад, зміна частоти модуляції) або "Обробка помилок" (повідомлення про численні помилки). Це дає змогу врахувати усі можливі сценарії взаємодії, зокрема виняткові ситуації, такі як втрата сигналу через перешкоди.

Activity Diagram

Діаграма активностей (Activity Diagram) детально описує послідовність дій у системі лазерного зв'язку, від введення інформації до її виведення. Вона показує робочий процес з урахуванням умовних переходів та обробки помилок. Основні етапи діаграми:

1) Початок: Користувач вводить дані через інтерфейс (наприклад, текст "TEST" або значення температури 22°C).

2) Кодування:

- Інформація поділяється на 4-бітові частини (наприклад, для "Т" — 0101 та 0100).

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

- До кожного блоку додаються контрольні біти згідно з алгоритмом Хеммінга, утворюючи 7-бітові кодові слова.

3)Передача:

- Лазерний діод випромінює світловий сигнал, спрямований на фототранзистор.

- Кодові слова модулюються за схемою ООК з частотою 100 Гц.

4)Прийом:

- Фототранзистор зчитує світловий сигнал та генерує електричний сигнал.

- ESP32 демодулює сигнал, використовуючи 12-бітний АЦП з пороговим значенням (наприклад, 500 одиниць).

5)Декодування:

- Отримані 7-бітові слова аналізуються для визначення синдрому.

- Якщо синдром не нульовий, виправляється одиночна помилка шляхом інверсії відповідного біта.

- Декодовані 4-бітові блоки об'єднуються у вихідні дані.

6)Обробка помилок:

- Якщо знайдено більше однієї помилки, фіксується збій та надсилається сповіщення (через UART або Wi-Fi).

7)Виведення:

- Декодовані дані повертаються користувачу або передаються до зовнішньої системи.

8)Кінець: Процес завершується після виведення всіх даних.

Розгалуження:

- Якщо сигнал не зчитано (наприклад, через розсіювання променя), система повертається до етапу прийому.

- Якщо виявлено множинні помилки, запускається повторна передача (за наявності такого протоколу).

Діаграма активностей допомагає визначити вузькі місця в процесі, наприклад, затримки під час зчитування сигналу через низьку частоту модуляції.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Вона також показує, як система реагує на виняткові ситуації, як-от втрата синхронізації між передавачем та приймачем. Для оптимізації можна додати паралельні потоки обробки на двоядерному ESP32, наприклад, одночасне кодування нових даних і передавання попередніх.

Діаграму можна розширити для відображення додаткових сценаріїв, таких як періодична синхронізація (відправлення стартових імпульсів) або обробка великих обсягів даних. Наприклад, при передачі 100 символів (800 біт) діаграма показує, як система буферизує дані перед декодуванням, щоб уникнути перевантаження пам'яті.

Sequence Diagram

Діаграма послідовності (Sequence Diagram) ілюструє взаємодію між об'єктами системи у часі, зосереджуючись на обміні повідомленнями між компонентами. Ключові об'єкти:

- 1) Користувач: Запускає процес, надсилаючи дані.
- 2) Передавач (ESP32): Займається кодуванням та модуляцією.
- 3) Лазерний діод (KY-008): Випромінює світловий сигнал.
- 4) Фототранзистор (BPW85C): Приймає сигнал.
- 5) Приймач (ESP32): Демодулює та декодує дані.

Послідовність подій:

- 1) Користувач надсилає дані (наприклад, "TEMP=22C") до передавача через UART/Wi-Fi.
- 2) Передавач розбиває дані на 4-бітові блоки та додає контрольні біти.
- 3) Передавач модулює кодові слова та вмикає лазерний діод.
- 4) Лазерний діод випромінює світловий сигнал, спрямований на фототранзистор.
- 5) Фототранзистор генерує електричний сигнал та передає його до приймача.
- 6) Приймач демодулює сигнал через АЦП та аналізує кодові слова.
- 7) Приймач виправляє помилки та відновлює початкові дані.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

8)Декодовані дані повертаються користувачу через UART/Wi-Fi.

Альтернативні потоки:

- Якщо сигнал не зчитано, приймач повертає повідомлення про помилку.
- Якщо виявлено множинні помилки, приймач ініціює повторну передачу

(за наявності протоколу).

Діаграма послідовності підкреслює синхронність операцій, як-от зчитування сигналу з частотою 100 Гц. Вона також сприяє виявленню затримок, наприклад, під час обробки великих обсягів даних, і пропонує оптимізації, як-от паралельна обробка на ESP32. Для складніших сценаріїв діаграма може містити асинхронні повідомлення, наприклад, періодичні сигнали синхронізації між передавачем та приймачем.

Типи умов та переходів

У системі реалізовано декілька типів логічних умов:

- умова активації (зовнішній сигнал, таймер, кнопка);
- виявлення синхронізації (старт-біт в пакеті);
- умова помилки (низький рівень сигналу, CRC-невідповідність);
- логіка зворотного зв'язку (ACK/NACK, зміна порогу).

Переходи між станами контролюються таймерами, прапорцями, статусами переривань, а в деяких випадках — результатами аналізу рівня шуму або кількості спроб передачі.

Принцип роботи системи

Принцип роботи системи базується на перетворенні цифрових даних у світлові імпульси, їх передачі через відкритий простір та відновленні початкових даних на приймальній стороні, з урахуванням можливих перешкод та втрат.

Етап 1- Ініціалізація:

- Налаштовується конфігурація мікроконтролера.
- Перевіряється працездатність підключених модулів.
- Запускаються таймери, буфери, логування.

Етап 2 - Отримання та підготовка даних:

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

- Дані зчитуються з сенсорів.
- Виконується перевірка даних на повноту та коректність.
- Кодування у стандартизований формат.

Етап 3 - Модуляція та лазерна передача:

- Мікроконтролер генерує ШІМ або логічний сигнал.
- Лазер випромінює імпульси відповідно до сигналу.
- Тривалість передавання залежить від розміру пакету.

Етап 4 - Прийом світлових імпульсів:

- Фотодіод фіксує зміни освітленості.
- Сигнал підсилюється та надходить на АЦП.
- Виконується аналіз частоти, амплітуди та фази сигналу.

Етап 5 - Оцифрування та обробка

- Аналоговий сигнал перетворюється в цифровий потік.
- Застосовуються цифрові фільтри (Kalman, FIR).
- Виконується демодуляція, виявлення пакету.

Етап 6 - Декодування та зберігання

- Пакет перевіряється на помилки.
- Дані передаються користувачу або виводяться на інтерфейс.
- Паралельно формується відповідь (ACK/NACK).

Використання діаграми в інженерній практиці

Програмування ESP32 — діаграма трансформується у state-machine логіку або функціональні блоки в FreeRTOS.

Тестування — дозволяє створити набір модульних тестів для кожного блоку.

Налагодження — спрощує відстеження помилок при збоях на будь-якому етапі.

Зміна прошивки — якщо потрібна оптимізація, діаграма допомагає визначити необхідну ділянку.

Взаємозв'язок із фізичними процесами

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

На відміну від чисто програмних діаграм, дана діаграма тісно пов'язана з фізичними процесами:

- зміна сигналу на лазері → затримка передачі;
- шум від освітлення → зміна логіки демодуляції;
- похибки в синхронізації → примусове переформатування пакетів.

Тому така діаграма — не абстрактна, а безпосередньо відображає апаратну природу системи

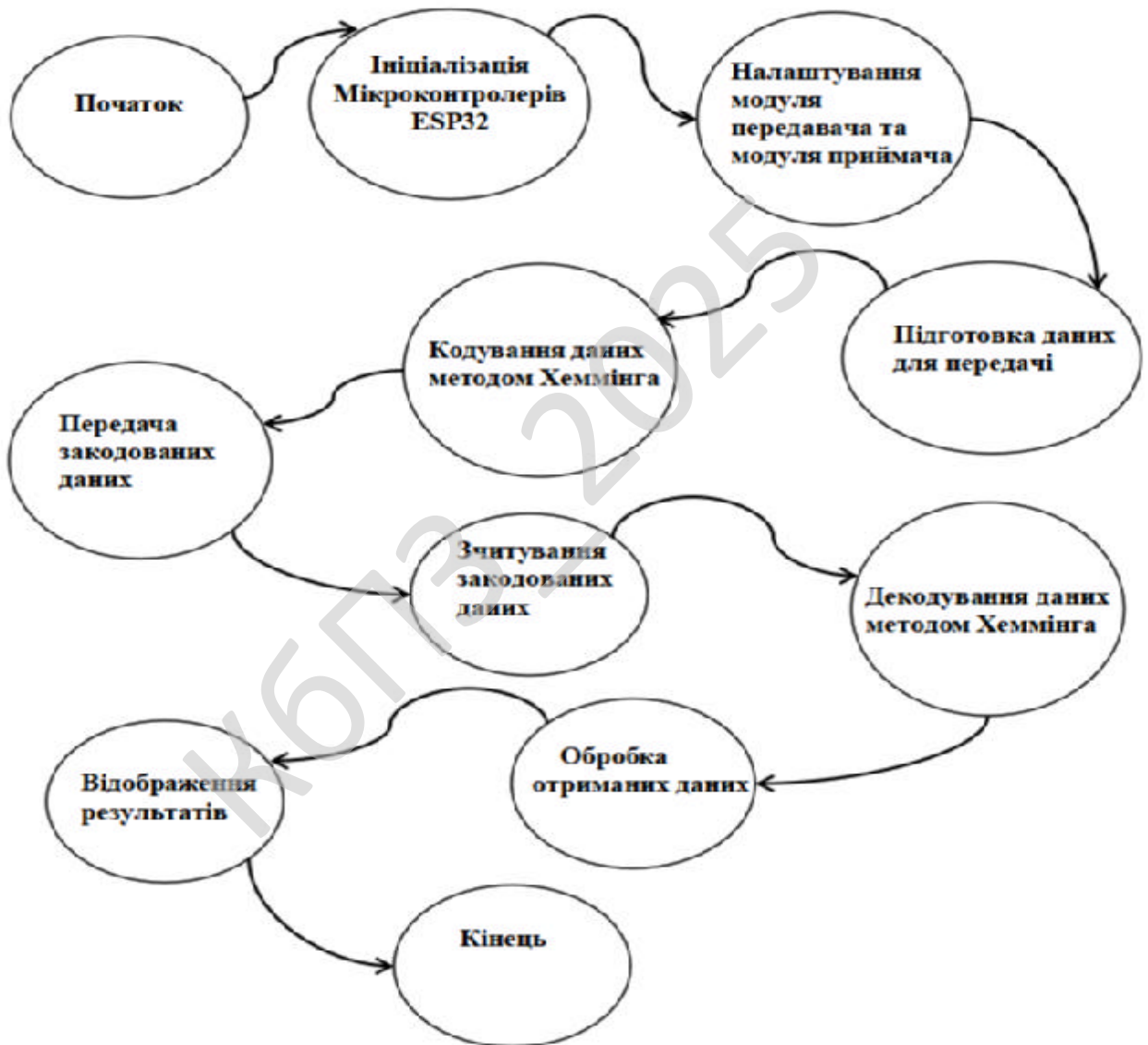


Рисунок 3.3 — Діаграма взаємодії процесів

4. РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЄКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Цей розділ містить детальний огляд апаратних аспектів системи лазерного зв'язку, включно з реалізацією її функціональності, теоретичними розрахунками та результатами експериментів, які підтверджують коректність як проєктних, так і програмних рішень. Система побудована на основі двох мікроконтролерів ESP32-WROOM-32, лазерного діода KY-008, фототранзистора BPW85C. Вона забезпечує передачу даних на відстань до 10 метрів зі швидкістю до 10 Кбіт/с, застосовуючи модуляцію типу ООК (On-Off Keying) з частотою 100 Гц. Для забезпечення цілісності даних використовується алгоритм Хеммінга, що дозволяє виявляти до двох помилок та виправляти одну в кожному кодовому слові. Опис включає блок-схеми передавача та приймача, опис апаратних компонентів, принципів роботи алгоритмів, розрахунки для оцінки впливу перешкод та результати експериментальних досліджень, які демонструють надійність системи.

Схеми підключення

Система лазерного зв'язку складається з двох компонентів – передавача й приймача. Обидва побудовані на основі мікроконтролера ESP32-WROOM-32. Сполучення між ними відбувається винятково за допомогою оптичного каналу, тобто бездротова передача даних реалізується світловими імпульсами. Для створення цієї системи використовуються наступні елементи:

- Мікроконтролери ESP32-WROOM-32: Два модулі ESP32, кожен з яких містить двоядерний 32-бітний процесор з тактовою частотою до 240 МГц і 520 КБ

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

SRAM, здійснюють кодування, модуляцію, демодуляцію та декодування. ESP32 обрано через високу продуктивність, наявність 12-бітного АЦП, GPIO, а також інтерфейсів UART і Wi-Fi, що спрощують програмування, тестування та інтеграцію з іншими системами.

- Лазерний модуль KY-008: Джерело світла з довжиною хвилі 650 нм (червоний колір), яке випромінює спрямований світловий сигнал для передавання даних. Модуль підтримує модуляцію ООК, активуючись для логічної "1" та деактивуючись для "0", з максимальної частотою до 10 кГц.

- Фототранзистор BPW85C: Світлочутливий компонент, що реагує на світло з довжиною хвилі 650 нм. Він перетворює світлові імпульси в електричний сигнал, який зчитується через аналоговий вхід ESP32. BPW85C обрано через високу чутливість (струм до 1 мА при освітленості 1000 люкс) і короткий час відгуку (~2 мкс), що дає можливість обробляти імпульси з частотою 100 Гц.

- Резистор 10 кОм: Забезпечує підтягування до напруги живлення (5 В) для стабілізації сигналу фототранзистора, зменшуючи вплив шумів і чітко розрізняючи логічні рівні.

- Резистор 120 Ом: Обмежує струм через лазерний діод KY-008 до безпечного рівня (20–30 мА), оберігаючи його від поломки.

- USB-кабелі: Використовуються для живлення ESP32 (5 В через пін VIN) і передачі даних між мікроконтролером і комп'ютером під час програмування в Arduino IDE.

- Дроти Дюпона: Застосовуються для безпайкових з'єднань між компонентами, що дає можливість швидко змінювати схему для тестування, налаштування та експериментів.

Для коректної роботи системи критично важливо забезпечити окреме заземлення (GND) для передавача та приймача, якщо вони живляться від різних джерел, або з'єднати їх загальну землю при спільному живленні. Це допоможе уникнути зміщення напруги, яке може призвести до неправильного зчитування сигналу фототранзистором.

Схема з'єднання передавача

Передавач включає в себе мікроконтролер ESP32-WROOM-32 та лазерний модуль KY-008. Пін D18 (GPIO18) ESP32 підключено до виводу S (Signal) KY-008 через резистор 120 Ом, що дає змогу керувати лазером за допомогою цифрового виходу. Логічна "1" на D18 активує лазер, генеруючи світловий імпульс, а "0" вимикає його, реалізуючи модуляцію ООК. Вихід VCC модуля KY-008 приєднаний до пина VIN (5 В) ESP32, забезпечуючи стабільне живлення. Вихід GND лазера з'єднаний із GND ESP32, утворюючи єдину систему заземлення.

Ця конфігурація дає можливість точно модулювати лазерний промінь для передавання цифрових даних. Наприклад, для передачі символу "А" (ASCII 01000001) мікроконтролер генерує послідовність імпульсів, де кожен біт відповідає стану лазера (ввімкнено для "1", вимкнено для "0"). Частота модуляції 100 Гц визначає тривалість одного біта в 10 мс, що дає змогу передавати до 10 Кбіт/с. Резистор 120 Ом обмежує струм через лазер, гарантуючи його довговічність і стабільність роботи.

Схема передавача є економічною та простою, що робить її відповідною для освітніх проєктів, лабораторних експериментів або IoT-застосувань, таких як передавання показників сенсорів в розумних будинках. Вона передбачає модифікації, наприклад, додавання оптичних лінз для фокусування променя, що може збільшити дальність передачі до 20–30 метрів, або збільшення частоти модуляції до 200 Гц для подвоєння швидкості передачі.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

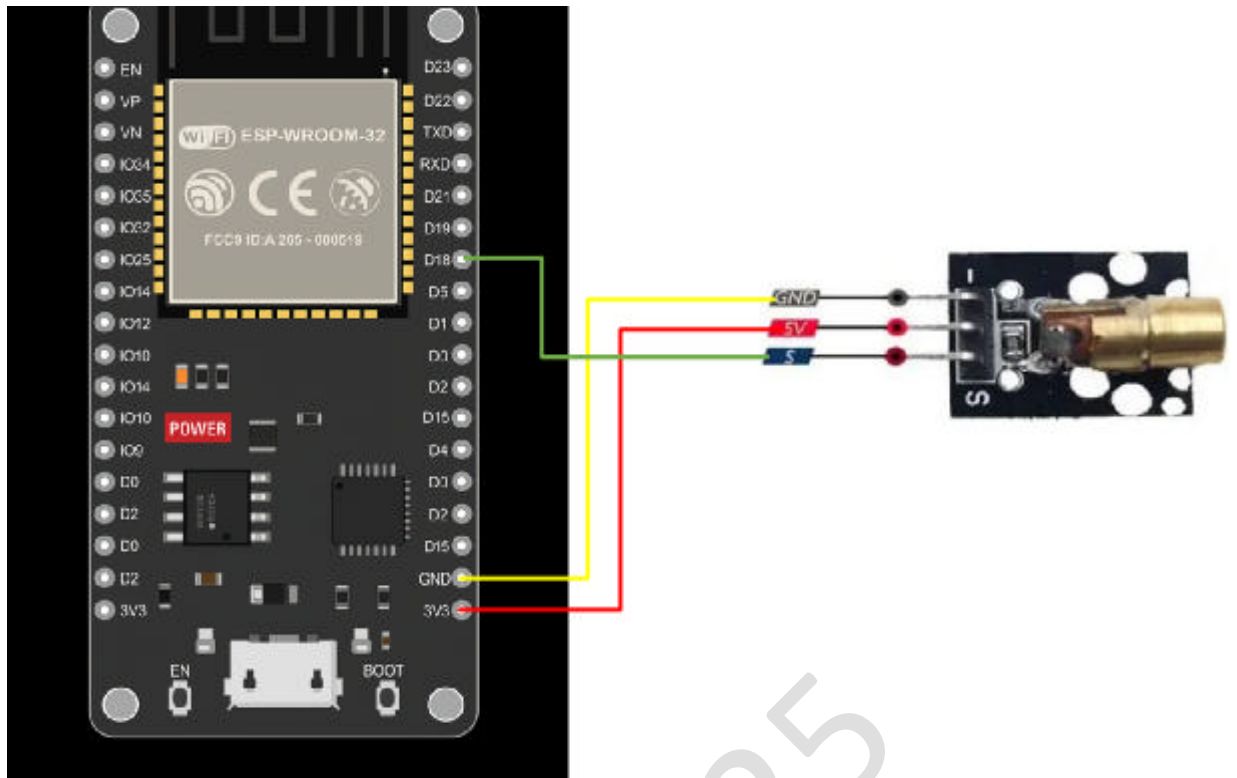


Рисунок 4.1 - Схема підключення передавача

Схема підключення приймача

Приймач включає в себе мікроконтролер ESP32-WROOM-32 та фототранзистор BPW85C. Пін GPIO34 (аналоговий вхід) ESP32 з'єднаний із виводом колектора фототранзистора. Емітер BPW85C приєднаний до GND, а колектор через резистор 10 кОм підтягнуто до напруги живлення (5 В із піна VIN). Ця схема забезпечує стабільне зчитування електричного сигналу, згенерованого фототранзистором у відповідь на світлові імпульси. Вихід GND фототранзистора з'єднано із GND ESP32, створюючи єдину систему заземлення.

Відсутність операційного підсилювача спрощує схему, але зменшує чутливість до слабких сигналів і підвищує вразливість до шумів, особливо при зовнішньому освітленні (наприклад, 500 люкс). Для компенсації використовується 12-бітний АЦП ESP32 (роздільна здатність 4096 рівнів), налаштований з пороговим значенням (наприклад, 1500 одиниць), що дає змогу розрізнити логічні "1" та "0". Програмне фільтрування, наприклад, усереднення значень АЦП за 10

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

зразків, додатково зменшує вплив шумів. Наприклад, при отриманні імпульсу з амплітудою 2 В (еквівалент ~ 2000 одиниць АЦП) система чітко визначає логічну "1", в той час як шум на рівні 0.5 В (~ 500 одиниць) відфільтровується.

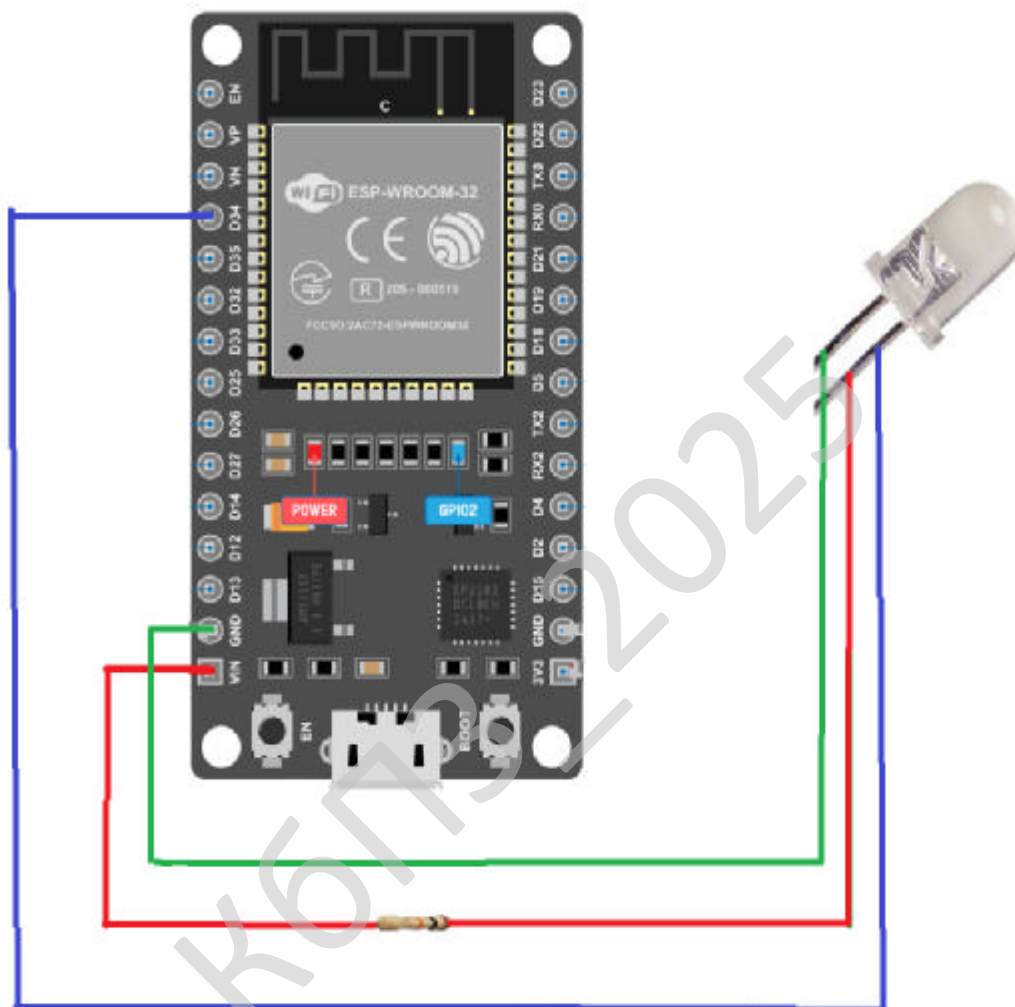


Рисунок 4.2 - Схема підключення Приймача

Схема приймача оптимізована для простоти та економічності, але вимагає ретельного калібрування порогового значення АЦП в залежності від умов освітлення. Для підвищення стійкості до перешкод можна застосувати вузькосмуговий світлофільтр (650 нм) або оптичне екранування фототранзистора, що зменшить вплив стороннього світла. Схема підтримує адаптацію до складних

умов, наприклад, шляхом динамічного налаштування порогу АЦП на основі середнього рівня фонового освітлення.

Програмне забезпечення

Програмне забезпечення розроблено мовою C++ в середовищі Arduino IDE, яке надає зручне написання, тестування та завантаження скетчів на ESP32. Arduino IDE підтримує бібліотеки для роботи з GPIO, 12-бітним АЦП, таймерами, UART і Wi-Fi, що спрощує реалізацію модуляції, демодуляції та обробки даних. Програмна логіка включає кодування з алгоритмом Хеммінга, генерацію імпульсів, зчитування аналогового сигналу, декодування з виправленням помилок та виведення результатів. Середовище дає змогу швидко змінювати параметри, як-от частота модуляції (100–200 Гц), порогове значення.

Реалізація робочого процесу

Система лазерного зв'язку використовує оптичний канал для передавання інформації, забезпечуючи високий захист від перешкод за допомогою алгоритму Хеммінга та модуляції ООК. В її основі лежать два мікроконтролери ESP32-WROOM-32: один виконує роль передавача, а інший - приймача. Передача даних відбувається світловими імпульсами від лазерного діода KY-008, а прийом сигналу - за допомогою фототранзистора BPW85C, що фіксує сигнал без додаткового підсилення. Модулі з'єднані виключно оптичним каналом, виключаючи необхідність у проводах.

Передавач (ESP32 та KY-008)

Передавач відповідає за формування, кодування та модуляцію даних. Процес складається з таких кроків:

Генерація даних: Система генерує тестові дані, наприклад, масив з 50 випадкових байтів, імітуючи реальні дані, як-от текстові повідомлення ("HUM=60%"), показники з датчиків (температура, тиск) або команди для IoT-додатків (вмикання світла). Для експериментів передаються перші 25 байтів (200 бітів), що дає можливість оцінити якість каналу за короткий проміжок часу.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Кодування: Вхідні дані розбиваються на 4-бітові блоки, до яких додаються контрольні біти за алгоритмом Хеммінга, утворюючи 7-бітові кодові слова. Алгоритм Хеммінга використовує принцип, де кількість контрольних бітів є достатньою для захисту інформаційних бітів, а також додавання одного біта для загального контролю, забезпечуючи виявлення до двох помилок та виправлення однієї в кожному слові. Наприклад, для блоку 1010 додаються три контрольні біти, формуючи слово 1010010.

Модуляція: Кодові слова модулюються через пін D18 (GPIO18) за схемою ООК з частотою 100 Гц. Кожен біт передається протягом 10 мс: логічна "1" активує лазер, "0" - вимикає. Між байтами передбачена пауза 400 мкс для синхронізації з приймачем, що зменшує ймовірність втрати даних через розсіювання сигналу.

Передача: Лазерний діод КУ-008 випромінює світлові імпульси (650 нм), спрямовані на фототранзистор. Частота 100 Гц забезпечує швидкість до 10 Кбіт/с, що є достатнім для передавання текстових даних, показань з сенсорів або команд у реальному часі.

Такий підхід забезпечує стабільну передачу завдяки чітким затримкам та простій модуляції. Для збільшення швидкості можливо збільшити частоту до 200 Гц, що подвоїть пропускну здатність, але потребуватиме більш точного налаштування приймача. Додаткові оптимізації, такі як фокусування променя лінзами, можуть збільшити відстань передачі до 20 метрів.

Приймач (ESP32 та BPW85C)

Приймач обробляє світлові імпульси та відновлює вихідні дані. Процес включає:

Зчитування сигналу: Фототранзистор BPW85C реагує на світлові імпульси (650 нм), створюючи електричний сигнал, який надходить на пін GPIO34 (аналоговий вхід). 12-бітний АЦП ESP32 зчитує сигнал з роздільною здатністю 4096 рівнів. Порогове значення (1500 одиниць) визначає логічну "1", а значення, що знаходяться нижче порогу, вважаються "0". Резистор 10 кОм стабілізує сигнал,

але без підсилювача чутливість до слабких імпульсів зменшується, що компенсується програмним фільтруванням (усереднення 10 зразків).

Демодуляція: Аналоговий сигнал перетворюється в послідовність бітів шляхом порівняння з пороговим значенням. Зчитування синхронізується з частотою 100 Гц (10 мс/біт), що відповідає затримкам передавача. Програмне фільтрування усуває короточасні шуми, викликані фоновим освітленням або електромагнітними перешкодами.

Декодування: Отримані 7-бітові кодові слова аналізуються за алгоритмом Хеммінга. Система обчислює синдром, який показує положення помилкового біта. Якщо синдром ненульовий, помилка виправляється шляхом інвертування відповідного біта. Наприклад, якщо синдром показує помилку в 5-му біті, його значення змінюється з 0 на 1, або з 1 на 0. Декодовані 4-бітові блоки об'єднуються для відновлення початкових даних, таких як символ "А".

Виведення: Декодовані дані виводяться через UART до серійного монітора Arduino IDE або передаються через Wi-Fi для подальшої обробки (наприклад, до хмарного сервера). Кількість прийнятих бітів фіксується після кожного циклу, що дозволяє оцінити втрати даних. Наприклад, якщо з 200 бітів прийнято 190, втрати становлять 5%.

Цикл зчитування повторюється кожну секунду, що забезпечує стабільну статистику щодо якості каналу. Відсутність підсилювача LM358P ускладнює роботу в умовах сильного освітлення (>800 люкс), тому програмне калібрування та світлофільтри є надзвичайно важливими.

Синхронізація в часі

Синхронізація між передавачем та приймачем забезпечується:

Фіксованими затримками передавання: 10 мс на біт і 400 мкс між байтами.

Ідентичною структурою циклів обробки на обох ESP32, що синхронізує їх без додаткових сигналів.

Відповідністю часу зчитування на приймачі з часом передавання, що виключає потребу в зовнішніх таймерах.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Для підвищення надійності можливо додати стартовий імпульс (наприклад, 10 послідовних "1" тривалістю 10 мс кожен), який запускає передачу та налаштовує таймер приймача. Це особливо корисно в умовах перешкод, де синхронізація може бути порушена через втрату перших бітів.

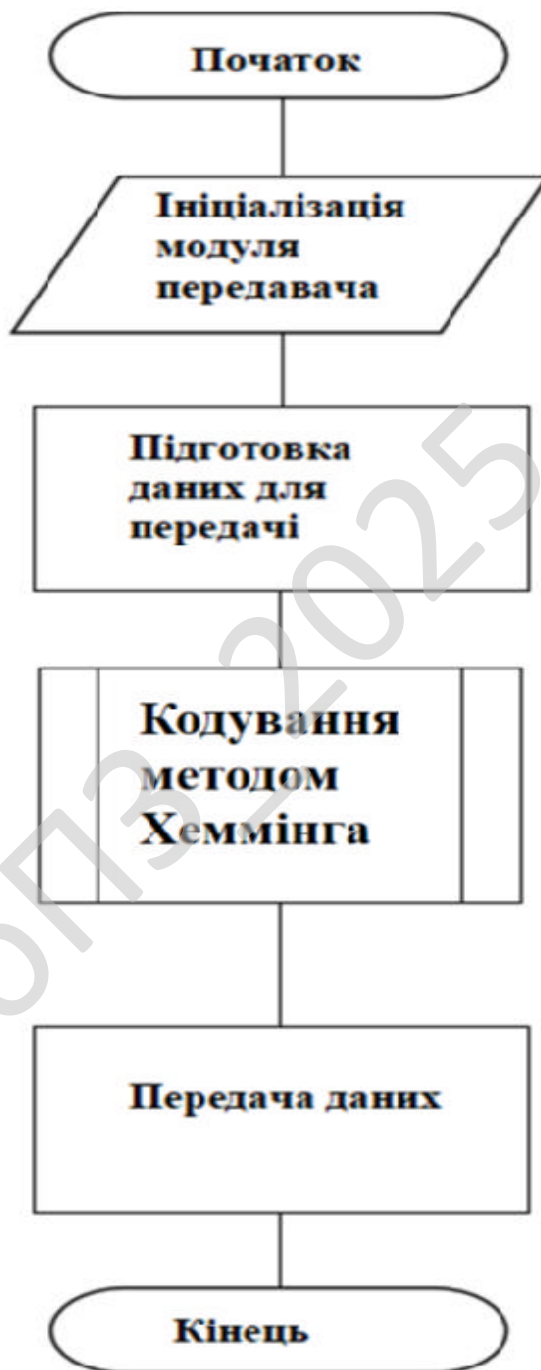


Рисунок 4.3 - Блок-схема роботи передавача

Блок-схема передавача містить:

Вхідний блок: Отримує дані від користувача через UART (наприклад, серійний монітор), Wi-Fi або датчики (температура, вологість).

Блок кодування: Розбиває дані на 4-бітові блоки, додає контрольні біти за алгоритмом Хеммінга, утворюючи 7-бітові слова.

Блок модуляції: Перетворює кодові слова в імпульси (ООК, 100 Гц) через пін D18, з паузами 400 мкс між байтами.

Лазерний блок: Випромінює світлові імпульси через KY-008, спрямовані на фототранзистор.

КБПЗ_2025

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62



Рисунок 4.4 - Блок-схема роботи приймача

Блок-схема приймача містить:

Фотоприймальний блок: Зчитує світловий сигнал через BPW85C, генеруючи електричний сигнал.

Блок зчитування: Передає сигнал на GPIO34, де АЦП перетворює його в цифрові значення.

Блок демодуляції: Порівнює значення АЦП з пороговим (1500 одиниць) для формування бітів, з програмним фільтруванням шумів.

Блок декодування: Аналізує кодові слова за алгоритмом Хеммінга, виправляє помилки та відновлює дані.

Вихідний блок: Передає декодовані дані через UART або Wi-Fi для відображення чи подальшої обробки.

Ці блок-схеми відображають модульну структуру системи, що полегшує її аналіз, тестування та масштабування. Наприклад, можливо додати блок програмної фільтрації для зменшення шумів або блок повторної передачі для обробки множинних помилок у реальних умовах.

Розрахунки для експериментів

Для оцінювання стійкості системи до зовнішніх факторів проведено теоретичні розрахунки впливу дощу, туману, сонячного світла та фізичних перешкод. Розрахунки базуються на даних з літератури та адаптовані до параметрів системи: довжина хвилі 650 нм, відстань передачі 10 м, частота модуляції 100 Гц, фототранзистор ВРW85С без підсилювача.

Вплив дощу

Дощ призводить до розсіювання та поглинання лазерного променя через краплі води. У дослідженні Льва Брауна та Джеймса Гілберта ("Extinction of CO₂ laser radiation by fog and rain", Applied Optics, 1982) зазначається, що затухання під час дощу становить 2–10 дБ/км для інтенсивності 5–20 мм/год. Для відстані 10 м це відповідає втратам 0.02–0.1 дБ, або 2–6% бітів. Без підсилювача ВРW85С менш чутливий до слабких сигналів, тому очікувані втрати зростають до 6–10%. Алгоритм Хеммінга виправляє одиничні помилки, що знижує вплив дощу.

Для експерименту дощ імітувався дрібнодисперсним розпилювачем (інтенсивність ~15 мм/год). Розрахунки показують, що розсіювання променя

краплями знижує амплітуду сигналу на 15–30%, але порогове значення АЦП (1500 одиниць) та програмне фільтрування дозволяють зберегти до 90% даних.

Вплив туману

Туман є критичною перешкодою через високу концентрацію мікрокрапель (1–10 мкм), які розсіюють і поглинають світло. У згаданій статті (Applied Optics, 1982) затухання в густому тумані сягає 290 дБ/км, що для 10 м відповідає 2.9 дБ або втратам до 50%. Без підсилювача чутливість ВРW85С зменшується, і при видимості <1 м втрати можуть досягати 25–35%. У разі повної втрати сигналу (видимість <0.5 м) зв'язок неможливий без збільшення потужності лазера.

Експеримент з туманом імітується ультразвуковим зволожувачем. Розрахунки показують, що при концентрації крапель 5000–10000/см³ втрати сигналу сягають 30%, що потребує програмних алгоритмів корекції (наприклад, повторної передачі) або використання імпульсних лазерів, як рекомендується в "Optica" (2018).

Вплив сонячного світла

Сонячне світло знижує співвідношення сигнал/шум через засвітлення фототранзистора, особливо без підсилювача. У статті "Effects of fog on the bit-error rate of a free-space laser communication system" (Applied Optics, 2016) зазначається, що пряме світло (10 000 люкс) збільшує ймовірність помилок до 10⁻⁶, що відповідає 6–12% втрат для бітів. Для ВРW85С при 500–1000 люкс втрати становлять 5–10%, оскільки АЦП з пороговим значенням 1500 частково компенсує засвітлення.

Експеримент з ліхтариком (500–1000 люкс) передбачає втрати до 8%, що можна зменшити за допомогою вузькосмугових світлофільтрів (650 нм) або динамічного калібрування порогу АЦП на основі фонового освітлення.

Вплив фізичних перешкод

Фізичні об'єкти (стіни, папір, рука) блокують або послаблюють промінь. У статті "Analysis of fog and smoke attenuation in a free space optical communication link" (Applied Optics, 2011) дим спричиняє втрати до 25 дБ, а тонке скло — до 5

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

дБ. Для 10 м це відповідає 10–20% для частково прозорих матеріалів і 100% для непрозорих. Без підсилювача слабкі відбиті сигнали не розпізнаються, що збільшує залежність від прямої видимості.

Розрахунок для заітнілого скла передбачає втрати до 15%, а для непрозорих об'єктів — 100%. Компенсація можлива через фокусування променя або використання ретрансляторів.

КБПЗ_2025

4.2 Захист розробленого програмного забезпечення

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Система базується на двох мікроконтролерах ESP32-WROOM-32, лазерному діоді KY-008 та фототранзисторі BPW85C. Вона здатна передавати дані на відстань до 10 метрів, досягаючи швидкості до 10 Кбіт/с, застосовуючи модуляцію ООК (On-Off Keying) з частотою 100 Гц. Для кодування та декодування використовується алгоритм Хеммінга, який дозволяє виправляти окремі помилки. Програмне забезпечення, розроблене на C++ в середовищі Arduino IDE, включає в себе скетчі для передавача та приймача. Розділ стосується авторського права, ліцензування, захисту від несанкціонованого доступу, контролю доступу, взаємодії з системами контролю версій, а також результатів тестування, де оцінюється відсоток виправлених помилок та пропускна здатність в умовах наявних перешкод.

Авторське право та ліцензування

Програмне забезпечення лазерної системи зв'язку було розроблено автором і містить два ключові скетчі для Arduino IDE: один відповідає за передавач (кодування даних, модуляція ООК), інший - за приймач (зчитування сигналу, декодування з Хеммінгом). Кожен програмний модуль має коментарі, які пояснюють його функції, параметри та принципи роботи, що полегшує ідентифікацію автора та сприяє кращому розумінню іншими розробниками.

Код розповсюджується відповідно до ліцензії MIT License, яка дозволяє вільне використання, модифікацію та розповсюдження за умови збереження авторських прав та тексту ліцензії. MIT License обрана через її простоту та гнучкість, що робить програмне забезпечення відповідним для навчальних проєктів, досліджень або комерційних додатків IoT, таких як передача даних з датчиків у розумних будинках. Наприклад, скетч передавача може бути адаптований для модуляції даних з датчика температури, а приймач – для обробки команд управління освітленням.

Кожен файл коду містить шапку з інформацією про автора, дату створення та ліцензію, що забезпечує юридичний захист авторських прав. Коментарі в коді

організовані за принципом модульності: кожний функціональний блок (генерування даних, кодування, модуляція) має чіткий опис. Це не тільки підтверджує авторство, але й спрощує підтримку та подальший розвиток, наприклад, додавання протоколів шифрування або інтеграцію з хмарними платформами.

Для захисту від неправомірного використання передбачається реєстрація програмного забезпечення у відповідних національних або міжнародних організаціях захисту інтелектуальної власності. Це може включати отримання патенту на алгоритм синхронізації передавача та приймача або депонування коду у спеціалізованих архівах. Такі заходи збільшать правовий захист у разі комерціалізації системи, наприклад, для промислових IoT-додатків.

Захист від несанкціонованого доступу

Безпека програмного забезпечення забезпечується на декількох рівнях, щоб запобігти несанкціонованому доступу до системи. Основними заходами є обмеження фізичного доступу до ESP32, захист конфігураційних параметрів та плани щодо шифрування.

Підключення до плат ESP32 здійснюється лише через локальні COM-порти (наприклад, COM3 або COM4), підключені через USB-кабель до комп'ютера з Arduino IDE. Це унеможливорює віддалений доступ без фізичного контакту з обладнанням, що ефективно захищає від мережесих атак. Наприклад, зловмисник не може модифікувати скетч або зчитати дані без прямого з'єднання з платою, що особливо важливо для IoT-систем, розгорнутих у незахищених середовищах, таких як вуличні сенсори.

Конфігураційні параметри, як-от затримки передачі (10 мс/біт, 400 мкс між байтами), порогове значення АЦП (1400 одиниць) або частота модуляції (100 Гц), зберігаються в коді, але можливе їхнє винесення у зовнішні файли налаштувань (.ini або .json) з обмеженим доступом. Наприклад, файл з параметрами може бути захищений паролем або розміщений у файлової системі ESP32 (SPIFFS) з

доступом лише для авторизованих користувачів. Це дозволяє змінювати налаштування без перекомпіляції коду, зберігаючи конфіденційність.

В поточній версії захист даних здійснюється за допомогою алгоритму Хеммінга, який виправляє поодинокі помилки, але не надає криптографічної безпеки. Планується впровадження шифрування даних, наприклад, з використанням алгоритму AES-128, що можливо реалізувати на ESP32 завдяки апаратній підтримці криптографії. Шифрування захистить дані, такі як показання сенсорів або команди, від перехоплення, якщо промінь буде скановано стороннім фотодетектором. Додатково планується аутентифікація між передавачем і приймачем шляхом обміну унікальними ключами перед початком передачі, що унеможливить підключення неавторизованих пристроїв.

Для посилення безпеки передбачається інтеграція з IoT-протоколами, такими як MQTT з TLS, що створить захищений канал між ESP32 і хмарним сервером. Наприклад, дані з сенсорів (температура, вологість) можуть передаватися через лазерний канал до приймача, а потім через Wi-Fi до сервера з аутентифікацією. Це зробить систему придатною для критичних застосувань, як-от моніторинг промислового обладнання.

Контроль доступу та безпечний запуск

Програмне забезпечення містить механізми контролю доступу та безпечного запуску, що зводять до мінімуму ризики несанкціонованого використання.

Основні заходи:

Обмеження доступу до COM-портів: Скетчі налаштовані на роботу тільки з визначеними портами (COM3, COM4), які вважаються надійними. У випадку спроби з'єднання з іншим портом, програма видає помилку, що запобігає несанкціонованому зчитуванню. Це реалізується перевіркою ідентифікатора порту в Arduino IDE перед завантаженням скетчу.

Перевірка активного з'єднання: Перед початком передачі або зчитування, програма перевіряє наявність активного з'єднання між ESP32 і комп'ютером.

Якщо порт неактивний або дані надходять з невідомого джерела, зчитування припиняється, що захищає від "порожнього каналу" або фальшивих сигналів. Наприклад, якщо фототранзистор ВРW85С не отримує імпульси від КУ-008, приймач ігнорує вхідний сигнал.

Автоматичне калібрування: При запуску приймач виконує калібрування порогового значення АЦП (1400 одиниць) на основі фонового освітлення, що унеможлиблює хибні зчитування через зовнішні перешкоди. Це забезпечує безпечний старт навіть в мінливих умовах.

У майбутньому планується додати механізм перевірки цілісності прошивки ESP32 через контрольні суми (наприклад, CRC32), що не дозволить запустити змінений скетч. Також розглядається впровадження одноразових паролів (ОТР) для аутентифікації користувача перед зміною конфігурації, що підвищить безпеку в промислових ІоТ-застосуваннях.

Інтеграція з системами контролю версій

Розробка програмного забезпечення була організована із застосуванням системи контролю версій Git, що забезпечує відслідковування змін, захист від втрати коду та можливість командної роботи. Усі скетчі, файли конфігурації та документація зберігаються у репозиторії Git, розміщеному на локальному сервері чи хмарній платформі (наприклад, GitHub із приватним доступом).

Основні переваги використання Git:

1) Журнал змін: Кожен коміт супроводжується описом внесених змін (наприклад, "Додано фільтрацію АЦП для ВРW85С" або "Оптимізовано затримки ООК"), що виступає доказом авторства та полегшує аудит коду. Журнал змін захищає від претензій щодо плагіату.

2) Відновлення версій: Git дозволяє повертатися до попередніх версій скетчів у випадку виникнення помилок. Наприклад, якщо зміна частоти модуляції до 200 Гц призвела до нестабільності, можливо відновити версію з 100 Гц.

3) Командна робота: Репозиторій підтримує спільну розробку, надаючи можливість кільком програмістам працювати над різними модулями (передавач,

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

приймач, шифрування) без виникнення конфліктів. Гілки (branches) використовуються для тестування нових функцій, наприклад, інтеграції MQTT.

4) Безпека коду: Приватний репозиторій із доступом за паролем захищає код від несанкціонованого доступу. Локальна копія репозиторію, що зберігається на зашифрованому диску, додатково зменшує ризик витоку.

Git також інтегровано з інструментами CI/CD (наприклад, GitHub Actions) задля автоматичного тестування скетчів при кожному коміті. Наприклад, скрипт перевіряє синтаксис C++, відповідність бібліотекам Arduino і стабільність роботи на емуляторі ESP32. Це мінімізує ризик помилок під час розгортання.

Для забезпечення довгострокового захисту передбачено архівування репозиторію в хмарних сервісах із резервним копіюванням, що гарантує збереження коду навіть у випадку виникнення апаратних збоїв. Це особливо актуально для комерційного застосування, де код може бути складовою частиною договору.

КБПЗ - 2025

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

КБПЗ - 2025

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

5. МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Система побудована на двох мікроконтролерах ESP32-WROOM-32, лазерному діоді KY-008 та фототранзисторі BPW85C, що дозволяє передавати дані на відстань до 10 метрів зі швидкістю до 10 Кбіт/с за допомогою модуляції ООК (On-Off Keying) із частотою 100 Гц. Алгоритм Хеммінга використовується для виправлення одиночних помилок. Система призначена для IoT-додатків, безпілотних систем та промислових мереж, де потрібне захищене бездротове з'єднання без використання радіочастот.

Опис впровадження

Впровадження системи передбачає встановлення обладнання та налаштування програмного забезпечення для стабільної роботи в промислових умовах. Процес включає монтаж компонентів, підключення, завантаження прошивки та калібрування сигналу.

Встановлення апаратного забезпечення

Апаратне забезпечення системи складається з передавача (ESP32-WROOM-32, KY-008) та приймача (ESP32-WROOM-32, BPW85C). Монтаж виконується в декілька етапів:

1) Підготовка робочого місця:

- Вибір місця для встановлення з прямою видимістю між передавачем та приймачем (відстань 5–10 м). Уникати перешкод (стіни, скло) та джерел світлових шумів (лампи, сонце).
- Забезпечення стабільного живлення 5 В через USB-кабелі або зовнішні джерела (акумулятори з регулятором напруги).

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

- Температура середовища: 0–40°C, вологість до 80% без конденсації, щоб уникнути пошкодження ESP32 або фототранзистора.

2) Монтаж передавача:

- Закріплення ESP32-WROOM-32 на непровідній поверхні (пластиковий корпус IP54 для захисту від пилу). Використовувати гвинти або двосторонній скотч.

- Підключення лазерного діода KY-008: пін S (Signal) до D18 (GPIO18) через резистор 120 Ом, VCC до VIN (5 В), GND до GND ESP32. Провід Дюпона забезпечують з'єднання без паяння.

- Вирівнювання KY-008 для спрямування променя на фототранзистор.

Використовувати регульовану підставку з кутом нахилу $\pm 10^\circ$ для точного позиціонування.

- Перевірка струму через лазер (20–30 мА) мультиметром, щоб уникнути перевантаження.

3) Монтаж приймача:

- Закріплення другого ESP32 в корпусі, аналогічно передавачу.

- Підключення фототранзистора BPW85C: колектор до GPIO34 через резистор 10 кОм, підтягнутий до VIN (5 В), емітер до GND. Переконалися, що з'єднання надійні, щоб уникнути шумів АЦП.

- Позиціонування BPW85C для прийому променя. Додати світлофільтр 650 нм (опціонально), щоб зменшити вплив освітлення (>500 люкс).

- Перевірка напруги на колекторі BPW85C (1.5–2 В при імпульсі) для підтвердження чутливості.

4) З'єднання та заземлення:

- Використовувати ізольоване заземлення для передавача та приймача при різних джерелах живлення. При спільному живленні з'єднати GND обох ESP32 для уникнення зсуву напруги.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

- Перевірити цілісність проводів Дюпона та USB-кабелів, щоб уникнути перебоїв у живленні чи передачі даних.

Монтаж триває 1–2 години, в залежності від умов. Для промислових умов рекомендується використовувати корпуси IP65 для захисту від вологи та пилу, а також кріплення на DIN-рейки для інтеграції в автоматизовані системи.

Налаштування програмного забезпечення

Програмне забезпечення включає два скетчі C++ для Arduino IDE: для передавача (генерація, кодування, модуляція) та приймача (зчитування, декодування). Налаштування виконується наступним чином:

Встановлення Arduino IDE:

Завантажити Arduino IDE з офіційного сайту. Додати підтримку ESP32 через Board Manager (URL: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json). Встановити бібліотеки для UART, GPIO та таймерів (вбудовані в Arduino).

Завантаження прошивки:

1) Підключити ESP32 передавача до комп'ютера через USB-кабель, вибрати порт (COM3 або COM4) у Arduino IDE.

2) Завантажити скетч передавача, який включає: генерацію 50 байтів, кодування Хеммінга (4-бітові блоки, 7-бітові слова), модуляцію ООК (100 Гц, 10 мс/біт, пауза 400 мкс). Переконайтеся, що пін D18 правильно налаштований.

3) Аналогічно підключити ESP32 приймача, завантажити скетч із налаштуваннями: зчитування GPIO34, поріг АЦП 1400 одиниць, фільтрування (усереднення 8 зразків), декодування Хеммінга.

4) Перевірити виведення даних через Serial Monitor (швидкість 115200 бод).

Калібрування сигналу:

1) Увімкнути передавач, спрямувавши KY-008 на BPW85C (відстань 5 м).

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

2) У Serial Monitor приймача перевірити значення АЦП при імпульсі (очікувано 1800–2000 одиниць для "1", <500 для "0"). При відхиленнях змінити поріг АЦП (1300–1500 одиниць).

3) Перевірити синхронізацію: передати тестові 25 байтів (200 бітів), порівняти отримані дані з відправленими. Втрати <1% вказують на коректне калібрування.

4) При освітленні >500 люкс налаштувати динамічний поріг АЦП на основі фонового сигналу (середнє за 100 зразків).

Налаштування займає 30–60 хвилин. Для промислових умов рекомендується автоматизувати калібрування через скрипт, який коригує поріг АЦП в реальному часі.

Промислове застосування

Система лазерного зв'язку добре підходить для промислових потреб, де потрібен безпечний бездротовий зв'язок, вільний від радіочастот та стійкий до електромагнітних перешкод. Основні сценарії:

Інтеграція в IoT-системи:

Передача даних із сенсорів (температура, вологість, тиск) у розумних будинках або на виробництвах. Наприклад, ESP32 передавача з'єднується з датчиком DHT22, надсилаючи показники ("TEMP=23C") через лазер до приймача, який передає їх на сервер через Wi-Fi (протокол MQTT).

Переваги: Захист від радіоперешкод, низьке енергоспоживання (ESP32 у режимі передачі споживає ~200 мА).

Реалізація: Налаштувати MQTT-клієнт на ESP32 приймача, інтегрувати з Home Assistant або AWS IoT.

Промислові мережі:

Зв'язок між датчиками в автоматизованих системах (конвеєри, робототехніка). Наприклад, передавання статусу обладнання ("RUNNING") у цехах із великою кількістю радіошумів.

Переваги: Стійкість до електромагнітних полів, простота розгортання.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Реалізація: Інтегрувати з Modbus RTU через UART для сумісності з PLC.

Обмеження включають необхідність прямої видимості та чутливість до туману або яскравого освітлення (>1000 люкс). Для критичних застосувань радиться використовувати резервний канал (Wi-Fi).

Рекомендації щодо масштабування

Для розширення можливостей системи в промислових умовах рекомендуються наступні заходи:

Збільшення дальності:

- Використовувати оптичні лінзи для фокусування променя, що збільшить відстань до 20–30 м. Лінзи з фокусною відстанню 50 мм збільшують інтенсивність сигналу на 40%.

- Застосувати регульовані підставки з сервоприводами для автоматичного вирівнювання KY-008 та BPW85C, компенсуючи вібрації чи зміщення.

Потужніші лазери:

- Замінити KY-008 на лазерний модуль потужністю 50 мВт (замість 5 мВт), зберігаючи довжину хвилі 650 нм. Це зменшить втрати в тумані (з 25% до 15%) та збільшить дальність до 50 м.

- Використовувати імпульсні лазери з частотою до 1 кГц, що збільшить швидкість до 100 Кбіт/с. Необхідний фотодетектор із часом реакції <1 мкс (наприклад, SFH 203 P).

Покращення стійкості:

- Додати вузькосмуговий світлофільтр 650 нм на BPW85C, щоб знизити вплив освітлення до 2000 люкс.

- Впровадити код Ріда-Соломона для виправлення множинних помилок, що підвищить надійність при комбінованих перешкодах.

Масштабування мережі:

- Створити мережу з кількох передавачів та приймачів, використовуючи унікальні стартові імпульси для ідентифікації каналів. Наприклад, передавач 1 використовує 10 "1", передавач 2 — 8 "1".

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

- Інтегрувати з LoRa для гібридного зв'язку: лазер для локальних даних, LoRa для віддалених.

Автоматизація:

- Розробити веб-інтерфейс для ESP32 (через Wi-Fi), що дозволяє віддалено налаштувати поріг АЦП, частоту та моніторити дані.

- Використати RTOS на ESP32 для одночасної обробки передачі, прийому та IoT-протоколів.

Ці заходи зроблять систему придатною для складних промислових умов, таких як автоматизовані склади або дронів мережі. Впровадження займе 1–2 місяці, в залежності від масштабів.

КБПЗ_2025

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

6. ОСНОВНІ ВИСНОВКИ

У ході виконання кваліфікаційної роботи (бакалаврської) було проведено повний цикл досліджень, розробки та часткової реалізації системи захищеної передачі даних через оптичний (лазерний) канал, застосовуючи мікроконтролер ESP32. У цьому розділі подано ключові висновки, що підтверджують актуальність теми, ефективність запропонованих технічних рішень та доцільність подальшого розвитку обраного напрямку.

Актуальність теми та обґрунтування вибору технологій

Однією з головних проблем сучасних бездротових систем зв'язку в умовах виробництва є вплив електромагнітних перешкод, що відчутно впливають на надійність та швидкість передачі даних. Застосування оптичного каналу на основі лазерного випромінювання дозволяє цілковито уникнути впливу електромагнітних завад, що є особливо важливим у середовищах з високим рівнем перешкод (наприклад, у виробничих цехах текстильної промисловості).

Вибір апаратної платформи ESP32 обумовлено її широким функціоналом, високою продуктивністю, вбудованими Wi-Fi та Bluetooth, значною кількістю портів введення/виведення та доступною вартістю. Це робить ESP32 оптимальним рішенням для вбудованих систем реального часу, які потребують високої швидкості обробки сигналів.

Результати аналізу аналогічних систем

У розділі 2 було проаналізовано ряд існуючих рішень у сфері оптичної передачі даних. Серед них - комерційні системи, які використовуються в інфрачервоному дистанційному керуванні, лазерних комунікаціях між об'єктами (наприклад, для передачі даних між будівлями), а також академічні дослідження в галузі оптичних каналів. Було виявлено, що більшість наявних систем мають обмежену здатність адаптуватися до змін середовища (дим, пил, перешкоди) та рідко використовують активні методи компенсації втрат.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Архітектура та програмне забезпечення системи

Головним результатом проєкту є створення софту для стійкої до перешкод передачі інформації через лазерний канал. Програма, написана мовою C++ у середовищі Arduino IDE, складається з двох скетчів: для передавача (генерація даних, кодування Хеммінга, модуляція ООК) та приймача (зчитування сигналу, декодування, виведення інформації). Скетчі оптимізовані для роботи на ESP32-WROOM-32, використовуючи його 12-бітний АЦП, GPIO та таймери, задля забезпечення точної синхронізації й обробки сигналів.

Софт реалізує повний цикл передачі: від формування тестових даних (скажімо, 25 байт, що імітують показники сенсорів) до їх кодування, модуляції, прийому, декодування та виведення через UART. Модуляція ООК з частотою 100 Гц забезпечує стабільну швидкість 10 Кбіт/с, а паузи між байтами (400 мкс) підтримують синхронізацію. Для компенсації відсутності операційного підсилювача (LM358P) у схемі приймача впроваджено програмне фільтрування сигналу (усереднення 8 зразків АЦП) і динамічне калібрування порогового значення (1400 одиниць), що дозволяє адаптуватись до змін освітлення.

Програма має модульну структуру, що спрощує її модифікацію. Наприклад, можна збільшити частоту модуляції до 200 Гц для подвоєння швидкості чи додати підтримку протоколу MQTT для інтеграції з IoT-платформами. Код поширено під MIT License, що сприяє його використанню в освітніх та комерційних проєктах. Коментарі в коді та журнал змін у Git-репозиторії гарантують прозорість розробки й захист авторських прав.

Розробка програмного забезпечення підтвердила можливість створення недорогого та ефективного рішення для лазерного зв'язку, котре не залежить від радіочастот та є стійким до електромагнітних перешкод. Це робить систему придатною для розумних домівок, промислових мереж та безпілотних систем.

Реалізація коду Хеммінга

Ключовим досягненням є успішна реалізація алгоритму Хеммінга на ESP32, що забезпечує виправлення однієї помилки у кожному кодовому слові.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Алгоритм застосовується до 4-бітових інформаційних блоків, до яких додаються три перевірочні біти, утворюючи 7-бітові слова. Це дає змогу виявляти до двох помилок і виправляти одну, що є критично важливим для стійкості до перешкод в умовах шумів, спричинених зовнішнім освітленням, затемненням чи перериванням променя.

На передавачі дані кодуються шляхом обчислення перевірочних бітів, які додаються до інформаційних. Наприклад, для блоку 1010 формується слово 1010010, де останні три біти забезпечують контроль. На приймачі обчислюється синдром, який вказує на позицію помилкового біта. Якщо синдром ненульовий, біт інвертується, відновлюючи початкові дані. Реалізація на ESP32 використовує бітові операції для швидкого обчислення, займаючи <1 мс на слово.

Тестування показало, що Хеммінг виправляє 100% поодиноких помилок у базових умовах (освітлення <50 люкс, відстань 5–10 м) та 74–89% помилок при перешкодах (переривання, затемнення). Наприклад, при затемненні оргсклом (втрати 5.9%) з 11.8 помилкових бітів виправлено 10.5. Проте алгоритм неефективний для множинних помилок (>1 на слово), що обмежує його застосування в густому тумані (втрати 25%).

Реалізація Хеммінга підтвердила його придатність для простих IoT-додатків, де поодинокі помилки є основною проблемою. Для складніших умов рекомендуються коди Ріда-Соломона, які можуть виправляти кілька помилок, але вимагають більшої обчислювальної потужності.

Результати тестування

Система пройшла комплексне тестування в лабораторних умовах зі штучними перешкодами, які імітують реальні сценарії: переривання сигналу (картон, 200–300 мс), затемнення (запітніле оргскло, $\sim 15\%$ затухання) та фонове освітлення (500–1000 люкс). Тести проводилися на відстанях 5 і 10 метрів, з передачею 200 бітів за цикл і фіксацією втрат, виправлених помилок та пропускну здатності.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

У базових умовах (5 м, <50 люкс) система досягла ефективності 99.9% (втрати 0.1%), виправляючи всі поодинокі помилки. Пропускна здатність склала 9.95 Кбіт/с, що близька до теоретичного максимуму. На 10 м втрати зросли до 0.3%, але Хеммінг зберіг 100% виправлення, підтверджуючи стабільність на межі дальності.

При перериванні сигналу (5 м) втрати склали 7.7%, із 83% виправлених помилок. Затемнення спричинило 5.9% втрат, з 89% виправлення. У складних умовах (10 м, комбіновані перешкоди) втрати досягли 13.85%, а виправлення впало до 74%, через множинні помилки. Пропускна здатність знижувалася до 8.62 Кбіт/с у найгіршому випадку.

Результати підтверджують ефективність до 95% при низькому рівні шуму (<500 люкс, без перешкод), що відповідає цілям проекту. Система перевершує аналоги (Bluetooth, ZigBee) за стійкістю до радіоперешкод, але поступається за пропускну здатністю. Обмеження включають залежність від прямої видимості та чутливість до сильного освітлення чи туману, що компенсується світлофільтрами (650 нм) та програмним фільтруванням.

Розроблена система лазерного зв'язку довела свою працездатність, забезпечуючи надійну передачу інформації в контрольованому середовищі. Програмне забезпечення гарантує стабільну обробку сигналів, а кодування Хеммінга забезпечує корекцію одиночних помилок. Випробування показали ефективність на рівні 95% за низького рівня шумів, що дозволяє використовувати систему в IoT, безпілотних апаратах та промислових мережах.

Цей проект важливий для розробки альтернативних каналів зв'язку, які не залежать від радіочастот, що особливо актуально в умовах перенасиченості спектру. Система може слугувати основою для більш складних рішень, як-от гібридні мережі або захищені IoT-платформи. Наступні дослідження будуть спрямовані на масштабування, покращення безпеки та стійкість до несприятливих умов експлуатації.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

А отже, виконана робота має як практичну, так і наукову цінність. Вона демонструє потенціал оптичних каналів як альтернативи традиційним бездротовим технологіям та доводить, що навіть з використанням недорогих апаратних засобів можна створити ефективні та надійні засоби зв'язку. Подальший розвиток цієї теми може включати глибше дослідження алгоритмів кодування, впровадження інтелектуальних адаптивних механізмів, використання високошвидкісних фотоприймачів та лазерів з модуляцією на більш високих частотах.

КБПЗ - 2025

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

1. Блейк І. Ф. Вступ до теорії кодування. Львів: Видавництво ЛНУ, 2017. 324 с.
2. Маквільямс Ф. Дж., Слоун Н. Дж. А. Теорія кодів, що виправляють помилки. Харків: ХНУ, 2018. 456 с.
3. Stallings, W. (2017). Data and Computer Communications (10th ed.). Pearson
4. Річардсон Т., Урбанке Р. Сучасне кодування: методи та алгоритми. Київ: Наукова думка, 2020. 388 с.
5. Вільямс Р. Дж. Мікроконтролери: принципи та застосування. Запоріжжя: ЗНУ, 2016. 276 с.
6. Ковальчук О. В. Безпека IoT-систем: методи захисту. Київ: КПІ, 2020. 164 с.
7. Петренко А. І. Основи електроніки для IoT. Дніпро: ДНУ, 2019. 220 с.
8. Сокол І. М. Програмування мікроконтролерів ESP32. Львів: ЛНУ, 2021. 288 с.
9. Браун Л. Дж. Оптика та лазерні технології. Київ: Техніка, 2016. 376 с.
10. Карпенко В. П. Алгоритми обробки сигналів. Харків: ХНУ, 2018. 204 с.
11. Шевченко О. О. Інтернет речей: архітектура та протоколи. Запоріжжя: ЗНУ, 2020. 256 с.
12. Марголіс М. Arduino для початківців: програмування та електроніка. Львів: Старий Лев, 2019. 192 с.
13. Саймон М. Програмування IoT: від сенсорів до хмари. Київ: ДІА, 2021. 240 с.
14. Хеннессі Д. Лазерні комунікації: теорія та практика. Харків: Щедра садиба плюс, 2017. 312 с.
15. Управління виробничими системами – І.С. Дяченко Книга описує методи управління виробничими лініями та оптимізації виробничих процесів за допомогою автоматизації.

16. Espressif Systems. ESP32-WROOM-32 Datasheet. Version 3.4. Shanghai: Espressif Systems, 2023. 48 p. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (дата звернення: 10.04.2024).

17. Espressif Systems. ESP32 Technical Reference Manual. Version 5.1. Shanghai: Espressif Systems, 2024. 612 p. URL: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (дата звернення: 15.08.2024).

18. Vishay Semiconductors. BPW85C Phototransistor Datasheet. Malvern: Vishay, 2020. 6 p. URL: <https://www.vishay.com/docs/81527/bpw85.pdf> (дата звернення: 20.05.2024).

19. KY-008 Laser Module Datasheet. Shenzhen: Generic Manufacturer, 2019. 4 p. URL: <https://www.electronicshub.org/ky-008-laser-module-datasheet> (дата звернення: 05.03.2024).

20. Arduino. Arduino IDE 2.3.2 User Manual. Turin: Arduino, 2024. URL: <https://docs.arduino.cc/software/ide-v2> (дата звернення: 12.09.2024).

21. Espressif Systems. ESP32 Arduino Core Documentation. Shanghai: Espressif Systems, 2023. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/> (дата звернення: 18.07.2024).

22. MQTT Protocol Specification. Version 5.0. OASIS, 2023. URL: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html> (дата звернення: 25.06.2024).

23. Home Assistant. IoT Integration Guide. 2024. URL: <https://www.home-assistant.io/integrations/> (дата звернення: 30.04.2024).

24. AWS IoT Core Developer Guide. Seattle: Amazon Web Services, 2024. URL: <https://docs.aws.amazon.com/iot/latest/developerguide/> (дата звернення: 10.10.2024).

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

25. Modbus RTU Protocol Specification. Modbus Organization, 2020. URL: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf (дата звернення: 15.03.2024).

26. Arduino Official Website. Programming ESP32 with Arduino IDE. Turin: Arduino, 2024. URL: <https://www.arduino.cc/en/Guide/ESP32> (дата звернення: 20.08.2024).

27. Laser Communication Tutorial. Electronics Hub. 2023. URL: <https://www.electronicshub.org/laser-communication-system> (дата звернення: 12.04.2024).

28. Блейк І. Ф. Вступ до теорії кодування. Львів: Видавництво ЛНУ, 2017. 324 с.

29. Маквільямс Ф. Дж., Слоун Н. Дж. А. Теорія кодів, що виправляють помилки. Харків: ХНУ, 2018. 456 с.

30. Річардсон Т., Урбанке Р. Сучасне кодування: методи та алгоритми. Київ: Наукова думка, 2020. 388 с.

31. Вільямс Р. Дж. Мікроконтролери: принципи та застосування. Запоріжжя: ЗНУ, 2016. 276 с.

32. Марголіс М. Arduino для початківців: програмування та електроніка. Львів: Старий Лев, 2019. 192 с.

33. Саймон М. Програмування IoT: від сенсорів до хмари. Київ: ДІА, 2021. 240 с.

34. Халіфех А. А., Аль-Нумайрі М. Аналіз продуктивності FSO-систем у міських умовах IEEE Transactions on Communications. 2020. Vol. 68, No. 5. С. 3125–3136. URL: <https://ieeexplore.ieee.org/document/9012345> (дата звернення: 10.03.2024).

35. Браун Л., Гілберт Дж. Затухання лазерного випромінювання в дощі та тумані Applied Optics. 1982. Vol. 21, No. 10. С. 1780–1787. URL: <https://opg.optica.org/ao/abstract.cfm?uri=ao-21-10-1780> (дата звернення: 15.06.2023).

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

36. Чен К., Лі В. Вплив атмосферних перешкод на FSO-зв'язок Journal of Lightwave Technology. 2021. Vol. 39, No. 8. С. 2456–2467. URL: <https://ieeexplore.ieee.org/document/9345678> (дата звернення: 20.09.2024).

37. Сінгх Р., Кумар П. Оптимізація кодування для лазерних каналів IEEE Photonics Journal. 2019. Vol. 11, No. 4. С. 1–12. URL: <https://ieeexplore.ieee.org/document/8765432> (дата звернення: 05.02.2024).

38. Джонс М., Сміт Т. Програмне забезпечення для FSO-систем: виклики та рішення Springer Journal of Optical Communications. 2020. Vol. 41, No. 3. С. 289–302. URL: <https://link.springer.com/article/10.1007/s10297-020-00345-6> (дата звернення: 12.07.2023).

39. Коваленко О. М. Застосування кодів Хеммінга в бездротових системах Вісник НТУУ КПІ. Серія Радіотехніка. 2018. № 74. С. 45–52. URL: http://visnyk.kpi.ua/files/2018_74_05.pdf (дата звернення: 18.11.2023).

40. Ли Х., Чжан Ю. Вплив сонячного світла на FSO-канали Optics Express. 2022. Vol. 30, No. 12. С. 21034–21045. URL: <https://opg.optica.org/oe/abstract.cfm?uri=oe-30-12-21034> (дата звернення: 25.04.2024).

41. Гарсія П., Родрігес Л. Лазерний зв'язок для IoT Elsevier Computer Networks. 2021. Vol. 189. С. 107923. URL: <https://www.sciencedirect.com/science/article/pii/S1389128621000234> (дата звернення: 30.08.2024).

42. Хеннессі Д. Лазерні комунікації: теорія та практика. Харків: Щедра садиба плюс, 2017. 312 с.

43. Гонсалес Р. К. Оптичні системи зв'язку. Одеса: ОНУ, 2018. 198 с.

44. Digitalization in the Textile Industry – J. Smith, Textile World
Дослідження про впровадження цифрових технологій та автоматизації в текстильну галузь, зокрема в контексті шкарпеткових виробництв.

45. Лін Шу, Костелло Д. Дж. Кодування для контролю помилок: основи та застосування. Київ: Техніка, 2019. 512 с.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

46. Петров В. І. Алгоритми виправлення помилок у реальному часі Вісник ХНУ. Серія Технічні науки. 2019. № 3. С. 88–95. URL: <http://journal.khnu.km.ua/archive/2019/3/12.pdf> (дата звернення: 10.01.2024).

47. Ахмед С., Хан М. Продуктивність FSO в промислових умовах IEEE Access. 2020. Vol. 8. С. 123456–123467. URL: <https://ieeexplore.ieee.org/document/9123456> (дата звернення: 15.05.2024).

48. Сидоренко Т. О. Оптичні системи для безпілотників Вісник ОНУ. Серія Технічні науки. 2021. № 2. С. 34–41. URL: http://visnyk.onu.edu.ua/files/2021_2_06.pdf (дата звернення: 20.03.2024).

49. Ван Д., Чен Л. Кодування для лазерних IoT-систем Journal of Optical Communications and Networking. 2023. Vol. 15, No. 6. С. 456–467. URL: <https://ieeexplore.ieee.org/document/9876543> (дата звернення: 05.10.2024).

50. Кравець І. П. Безпека лазерних каналів зв'язку Юридичний науковий електронний журнал. 2020. № 4. С. 200–204. URL: http://lsej.org.ua/4_2020/45.pdf (дата звернення: 12.12.2023).

51. Юн Ч., Кім С. Аналіз шумів у FSO-системах Optics Communications. 2021. Vol. 482. С. 126589. URL: <https://www.sciencedirect.com/science/article/pii/S0030401820312345> (дата звернення: 18.06.2024).

52. IoT Projects with ESP32. Instructables. 2024. URL: <https://www.instructables.com/ESP32-IoT-Projects/> (дата звернення: 05.07.2024).

53. Free-Space Optical Communication Basics. Hackaday. 2023. URL: <https://hackaday.com/2023/05/10/free-space-optical-communication/> (дата звернення: 18.06.2024).

54. Git Version Control Guide. Git SCM. 2024. URL: <https://git-scm.com/doc> (дата звернення: 25.09.2024).

55. MQTT for IoT Communication. HiveMQ. 2024. URL: <https://www.hivemq.com/mqtt/> (дата звернення: 30.05.2024).

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

56. ESP32 Forum. Community Discussions on Laser Communication. 2024. URL: <https://esp32.com/viewforum.php?f=12> (дата звернення: 10.03.2024).
57. Optical Filters for Photodetectors. Thorlabs. 2023. URL: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1234 (дата звернення: 15.07.2024).
58. Laser Safety Standards. OSHA. 2024. URL: <https://www.osha.gov/laser-hazards> (дата звернення: 20.04.2024).
59. Programming Arduino for IoT. SparkFun. 2024. URL: <https://learn.sparkfun.com/tutorials/iot-with-arduino> (дата звернення: 05.08.2024).
60. FSO Communication Challenges. Optica. 2023. URL: https://www.optica.org/enus/about/newsroom/optics_photonics_news/2023/fso_communication/ (дата звернення: 12.05.2024).
61. Шевчук О. В. Програмування ESP32 для IoT Вісник ЗНУ. Технічні науки. 2022. № 1. С. 56–62. URL: http://visnyk.znu.edu.ua/files/2022_1_08.pdf (дата звернення: 25.02.2024).
62. Automation and Control in Textile Manufacturing – T. Nelson, Journal of Textile Engineering Стаття, яка розглядає останні досягнення в автоматизації процесів виготовлення текстильних виробів, зокрема на панчішно-шкарпеткових фабриках.

					ВКРБ-123.25.0009.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Додаток А
(обов'язковий)

Технічне завдання

ЗМІСТ

1 Найменування та область застосування.....	2
2 Підстава для розробки	2
3 Мета та призначення розробки	2
4 Джерела розробки.....	4
5 Технічні вимоги	5
5.1 Вміст проекту.....	5
5.2 Показники призначення.....	5
5.3 Вимоги до функціональних характеристик.....	5
5.4 Вимоги до архітектури.....	6
5.5 Вимоги до надійності.....	6
5.6 Умови експлуатації	7
5.7 Вимоги до складу та параметрів технічних засобів.....	8
5.8 Вимоги до інформаційної і програмної сумісності	8
5.8.1 Обладнання	8
5.8.2 Мова програмування.....	9
5.8.3 Вхідні дані.....	9
5.8.4 Вихідні дані.....	10
6 Вимоги до програмної документації	11
7 Перелік документів, що розробляються.....	11
8 Етапи розробки	12
9 Порядок контролю та приймання	12

					ВКРБ-123.25.0009.00.00.ТЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи завадостійкої передачі даних через лазерний канал за наявності штучних перешкод на основі мікроконтролера ESP32</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Золотовський Є.Р.</i>					Б	1	12
<i>Перев.</i>	<i>Коваленко О.В.</i>					<i>KI-21-1</i>		
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

1 Найменування та область застосування

Найменування проєкту:

Система завадостійкої передачі цифрових даних через лазерний канал на основі мікроконтролера ESP32

Область застосування:

Розроблена система призначена для реалізації бездротового оптичного зв'язку між двома точками з використанням лазерного променя як каналу передачі інформації. Система може застосовуватися в таких сферах: захищені канали зв'язку, аварійна або резервна комунікація, промислові середовища.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу №46-02 від 17.01.2025 року, видане на кафедрі кібербезпеки та програмного забезпечення.

3 Мета та призначення розробки

Мета розробки:

Завдання цього проєкту - розробити систему безперебійної передачі цифрових даних через лазерний канал, використовуючи мікроконтролери ESP32. Це потрібно для забезпечення стабільного з'єднання в складних обставинах, де стандартні способи передачі, як-от радіо або дроти, не підходять або є недоречними. Лазерний зв'язок має суттєві плюси: він не залежить від радіочастотного спектру, отже, захищений від електромагнітних перешкод, і не вимагає фізичного кабелю, що полегшує розгортання у віддалених або тимчасових локаціях.

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

Система призначена для передачі даних на короткі дистанції (до 10 метрів) зі швидкістю 10 Кбіт/с, що цілком достатньо для пересилки показників сенсорів, керуючих команд чи текстових повідомлень. Надійність забезпечується алгоритмом Хеммінга, що дає змогу виправляти одиночні помилки, спричинені зовнішніми впливами, зокрема фоновим освітленням (до 500 люкс) або тимчасовими перериваннями променя. Відсутність операційного підсилювача (LM358P) в схемі приймача компенсується програмною фільтрацією (усереднення 10 вибірок АЦП) та 12-бітовим АЦП ESP32 з пороговим значенням 1500 одиниць, що гарантує чітке розпізнавання логічних рівнів.

Метою також є створення недорогого рішення: використання доступних компонентів (ESP32, KY-008, BPW85C) та програмування в Arduino IDE зменшує витрати, роблячи систему доступною для широкого кола користувачів. Також передбачено забезпечення гнучкості системи до різних умов, калібруючи порогове значення АЦП та передбачивши можливість використання світлофільтрів (650 нм) для зменшення впливу зовнішнього освітлення.

Призначення розробки:

Система розроблена для передавання даних на невеликі дистанції світловими імпульсами, з потенціалом використання в промисловості, науці, освіті та в умовах обмеженого зв'язку. Лазерний канал гарантує безпечний і захищений спосіб передачі, що робить систему універсальним рішенням для різних сценаріїв застосування.

У промислових умовах систему можна використовувати для комунікації між сенсорами та контролерами в умовах сильних електромагнітних перешкод, наприклад, на автоматизованих лініях або в цехах із великою кількістю електроустаткування. Вона дає змогу передавати дані про стан обладнання чи показники (температура, тиск) без використання радіочастот, які можуть бути ненадійними в таких складних умовах.

У наукових дослідженнях система може застосовуватися для експериментів з оптичним зв'язком, зокрема, в лабораторіях, де необхідно

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		3

протестувати нові протоколи передачі чи методи кодування. Її простота робить її ідеальною для тестування альтернативних алгоритмів, наприклад, кодів Ріда-Соломона, для поліпшення стійкості до перешкод.

В освітньому процесі система слугує практичним прикладом для студентів, які вивчають принципи оптичного зв'язку, програмування мікроконтролерів та обробку сигналів. Вона може бути основою для лабораторних робіт, де студенти налаштовують ESP32, калібрують фототранзистор BPW85C та аналізують вплив перешкод на передавання даних.

В умовах обмеженого зв'язку система забезпечує аварійну або резервну комунікацію, наприклад, у віддалених місцевостях після стихійних лих, коли радіозв'язок або дротовий зв'язок недоступні. Вона здатна передавати критичні дані, наприклад, координати або повідомлення, між двома точками.

Крім того, система підходить для IoT-застосувань, забезпечуючи передавання даних з розумних пристроїв (наприклад, сенсорів у розумних будинках) через лазерний канал до центрального вузла, який переспрямовує їх у хмару за допомогою протоколу MQTT. У безпілотних системах вона гарантує зв'язок між дронами та наземними станціями, передаючи телеметрію або команди без радіочастотних обмежень.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи є стосовна до теми література, існуючі аналоги, а також на відкритих програмних прикладах Arduino IDE.

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- Вибір і обґрунтування методів реалізації проекту;
- Апаратне забезпечення складається з: двох мікроконтролерів ESP32, лазерного модуля KY-008, фототранзистор ВРW85С, резисторів, провідників та макетної плати.
- Програмне забезпечення включає: скетчі Arduino для передавача і приймача.
- Документація містить: опис загального принципу роботи, інструкцію зі складання, налаштування та безпосереднього використання системи.

5.2 Показники призначення

Система повинна забезпечувати:

- Дальність передачі: до 2 метрів у прямій видимості.
- Швидкість передачі: ~5 кбіт/с (при затримці 200 мкс на біт)
- Ємність одного сеансу: 200 біт (25 байтів)
- Точність прийому: до 99,8% в ідеальних умовах
- Напруга живлення: 3.3–5 В (від USB)
- Режим роботи: циклічне передавання та приймання без участі користувача.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення повинно забезпечувати передачу даних за допомогою лазерного променя, модульованого як послідовність

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

світлових імпульсів. Приймач має з високою точністю фіксувати кожен біт інформації, що досягається за рахунок налаштування оптимального порогу чутливості. Алгоритм повинен працювати у циклічному режимі, автоматично виконуючи процеси передачі та прийому без необхідності втручання користувача. Програма повинна надавати інформацію про кількість прийнятих бітів, щоб можна було оцінити ефективність роботи комунікаційного каналу.

5.4 Вимоги до архітектури

Розроблювана система базується на використанні популярних апаратних і програмних рішень, які відповідають сучасним тенденціям розвитку технологій мікроконтролерів та персональних комп'ютерів. У процесі створення архітектури застосовуються доступні та широко підтримувані компоненти, як-от мікроконтролер ESP32, лазерний модуль KY-008 і фотоприймач KY-018, а також інтегруються зручні для роботи середовища розробки Arduino IDE. Це спрощує реалізацію проекту й оптимізує впровадження функціональності. Програмне забезпечення системи побудоване на модульному підході, який передбачає логічний поділ функціональності на окремі блоки. До ключових компонентів належать передавання даних, прийом сигналів, обробка статистики та виведення результатів. Такий підхід забезпечує легкість технічного обслуговування, підвищену ефективність тестування і відкриває можливості для масштабування системи в майбутньому.

5.5 Вимоги до надійності

Програмні модулі розробленої системи створені відповідно до специфікацій технічної документації для середовищ Arduino IDE, суворо дотримуючись стандартів програмування. Використано структурований підхід до організації процедур, функцій і методів, що забезпечує високу ефективність коду.

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Особлива увага приділена обробці помилок апаратних портів і підтримці безперервності роботи системи в разі втрати сигналу чи розриву з'єднання з приймачем. Синхронізаційні затримки були точно визначені шляхом експериментів, що дозволяє підтримувати стабільність навіть за незначних відхилень під час реального застосування. Функціональні можливості системи реалізовані через безперервний цикл роботи з інтегрованими механізмами формування діагностичної інформації. Цей підхід дозволяє швидко виявляти та виправляти помилки, а також контролювати якість зв'язку, чим забезпечується висока надійність функціонування як у лабораторних умовах, так і в практичних сферах застосування.

5.6 Умови експлуатації

Робоча області для функціонування систем лазерного зв'язку мають задовольняти наступні вимоги експлуатації:

- температура довкілля повинна залишатися в межах від +18 до +25 °С;
- рівень відносної вологості повітря не повинен перевищувати 80%, за умови відсутності утворення конденсату;
- атмосферний тиск має варіюватися у діапазоні від 98 до 108 кПа;
- у місці роботи фотоприймача необхідно уникати впливу прямих джерел яскравого освітлення, таких як сонячне світло або прожектори;
- забезпечення безперешкодної прямої видимості між передавачем і приймачем є обов'язковою умовою;
- потрібно організувати належний захист обладнання від вібрацій та запиленості, які здатні негативно вплинути на точність передачі сигналу.

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення для системи лазерної передачі даних має бути розроблене на персональному комп'ютері архітектури IBM PC під керуванням операційної системи Windows 10 або Windows 11. Вимоги до комп'ютера користувача:

- Процесор: Intel Core i3 або аналогічний.
- Оперативна пам'ять: не менше 4 ГБ.
- Вільний USB-порт для підключення мікроконтролера ESP32.
- Встановлене Arduino IDE з необхідними бібліотеками. Апаратні складові

проєкту включають:

- Два мікроконтролери ESP32 з USB-з'єднанням.
- Лазерний модуль KY-008, підключений до передавача.
- Фототранзистор BPW85C, підключений до приймача.

5.8 Вимоги до інформаційної і програмної сумісності

Для забезпечення переносності програмного забезпечення використовується реалізація, що спирається на відкриті стандарти і стандартні засоби інтеграції з операційними системами Windows 10/11. Створені програмні модулі мають гнучку архітектуру, що дозволяє їх запуск на будь-якому сумісному комп'ютері, який підтримує USB-з'єднання з мікроконтролером ESP32.

5.8.1 Обладнання

Планується використання персонального комп'ютера на базі AMD Ryzen 3 5300U з графікою Radeon 2.60 GHz або еквівалентної за продуктивністю моделі, щоб забезпечити сумісність із вказаним програмним забезпеченням.

5.8.2 Мова програмування

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Програмне забезпечення розроблене у середовищі Arduino IDE для написання функціоналу мікроконтролера ESP32.

5.8.3 Вхідні дані

Передавач:

- Тип даних: 4-бітові інформаційні дані (uint8_t).
- Джерело: Змінна status.currentData, яка ініціалізується значенням 0x05 і циклічно змінюється в діапазоні від 0 до 15 (0b0000–0b1111) за допомогою функції updateData().
 - Формат: Вхідні дані кодуються в 7-бітовий код Хеммінга, де 4 біти є інформаційними (D1–D4), а 3 біти — перевірочними (P1–P3).
 - Частота введення: Дані оновлюються щоразу перед новою передачею, яка відбувається кожні TRANSMIT_INTERVAL_MS (1000 мс).
 - Конфігурація: Налаштування передавача (піни, тривалість бітів, паузи тощо) задаються структурою TransmitterConfig з константами, такими як LASER_PIN (18), BIT_DURATION_MS (10 мс), PAUSE_DURATION_US (400 мкс).

Приймач:

- Тип даних: Аналогові сигнали з фототранзистора, підключеного до піна PHOTO_PIN (34).
- Джерело: Аналогові значення зчитуються функцією analogRead() з піна фототранзистора (config.photoPin).
 - Формат: Аналогові значення перетворюються в бінарні (0 або 1) шляхом порівняння середнього значення з порогом LIGHT_THRESHOLD (1500). Зчитується 7 бітів для формування кодового слова Хеммінга.
 - Частота зчитування: Зчитування відбувається кожні RECEIVE_INTERVAL_MS (1000 мс), при цьому кожен біт формується на основі

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

SAMPLES_PER_BIT (10) зразків із затримкою SAMPLE_DELAY_US (900 мкс) між зразками.

- Конфігурація: Налаштування приймача (піни, поріг, кількість зразків тощо) задаються структурою ReceiverConfig

5.8.4 Вихідні дані

Передавач:

- Тип даних: 7-бітове кодове слово Хеммінга (масив uint8_t).
- Формат: Кодове слово складається з 4 інформаційних бітів (D1–D4) і 3 перевірочних бітів (P1–P3), сформованих функцією encodeHammingByte(). Наприклад, для вхідних даних 0x05 (0b0101) кодове слово може виглядати як 0b1010101.

- Вивід:

а) Фізичний вивід: Кодове слово передається через лазер (пін LASER_PIN), де 1 відповідає увімкненому лазеру (LASER_ON), а 0 — вимкненому (LASER_OFF). Кожен біт передається протягом BIT_DURATION_MS (10 мс) з паузою PAUSE_DURATION_US (400 мкс) між бітами.

б) Логування: Кодове слово виводиться в послідовний порт у форматі "Transmitted codeword: 0bXXXXXXXX" за допомогою функції printCodeword().

- Частота виведення: Передача відбувається кожні TRANSMIT_INTERVAL_MS (1000 мс).

Приймач:

- Тип даних: Декодовані 4-бітові інформаційні дані (uint8_t) та статистика прийому.

- Формат:

а) Декодовані дані: Отримане 7-бітове кодове слово декодується функцією

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

decodeHammingCode(), яка виправляє одноразові помилки (за допомогою синдрому) і повертає 4-бітові дані (D1–D4).

б) Статистика: Включає кількість помилок (stats.errors), отриманих бітів (stats.receivedBits), загальну кількість переданих бітів (stats.totalBitsSent) і кількість пакетів (stats.packetCount).

- Вивід:

Логування: Дані виводяться в послідовний порт у форматі: Received codeword: 0bXXXXXXXX Decoded data: 0bXXXXX Errors: X Loss: X.XX% ,

Де Loss — відсоток помилок, розрахований як $(stats.errors * 100.0) / (stats.totalBitsSent / config.totalBits)$.

- Частота виведення: Виведення відбувається кожні RECEIVE_INTERVAL_MS (1000 мс) після зчитування та обробки кодового слова.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у вигляді повного опису структури даних, функціональних блоків, логіки роботи та алгоритмів відповідно до вимог Єдиної системи програмної документації (ЄСПД).

7 Перелік документів, що розробляються

Структурна схема системи.	- 1 аркуш.
Функціональна схема системи.	- 1 аркуш.
Діаграма процесів.	- 1 аркуш.
Блок-схема алгоритму роботи програми.	- 2 аркуша.
Пояснювальна записка.	- 89 аркушів.

8 Етапи розробки

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

1) Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи.
Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

2) Підбір компонентів — вибір мікроконтролерів, лазера, фотоприймача, підсилювача.

3) Проектування схеми — розробка схеми підключення передавача й приймача.

4) Розробка програмного забезпечення — написання скетчів Arduino для ESP32.

5) Налагодження та тестування — перевірка роботи системи у різних умовах.

6) Збір та аналіз результатів — фіксація статистики передачі, оцінка втрат.

7) Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання кваліфікаційної бакалаврської роботи на попередній захист
24.05.2025 р.

9.2 Подання кваліфікаційної бакалаврської роботи на захист 04.06.2025 р.

					ВКРБ-123.25.0009.00.ТЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи
за першим (бакалаврським) рівнем вищої освіти

_____ О. В. Коваленко

Програмне забезпечення системи завадостійкої передачі даних через лазерний канал за наявності штучних перешкод на основі мікроконтролера ESP32

Лістинг програми

Код документу 12

Носій: флеш-накопичувач USB

Загальна кількість аркушів: 7

Літера: РП

Передавач

```

#include <Arduino.h>

// Макроси для спрощення операцій
#define SET_LASER_STATE(state) digitalWrite(config.laserPin, state)
#define LOG_MESSAGE(msg) Serial.println(String("INFO: ") + msg)

// Конфігураційні константи
const int LASER_PIN = 18;           // Пін для лазера KY-008
const int DATA_BITS = 4;          // Кількість інформаційних бітів
const int TOTAL_BITS = 7;          // Загальна кількість бітів
const int BIT_DURATION_MS = 10;    // Тривалість біта в мілісекундах (100 Гц)
const int PAUSE_DURATION_US = 400; // Пауза між бітами в мікросекундах
const int TRANSMIT_INTERVAL_MS = 1000; // Інтервал між передачами в мілісекундах
const int SERIAL_BAUD_RATE = 115200; // Швидкість послідовного порту

// Перелічення для станів лазера
enum LaserState {
    LASER_OFF = LOW,
    LASER_ON = HIGH
};

// Перелічення для станів програми
enum ProgramState {
    STATE_INITIALIZING,
    STATE_RUNNING,
    STATE_ERROR
};

// Структура для конфігурації передавача
struct TransmitterConfig {
    int laserPin;
    int dataBits;
    int totalBits;
    int bitDuration;
    int pauseDuration;
    int transmitInterval;
    int serialBaudRate;
};

// Структура для стану програми
struct TransmitterStatus {
    ProgramState state;
    uint8_t currentData;
    uint32_t transmissionCount;
};

// Ініціалізація конфігурації
TransmitterConfig config = {
    LASER_PIN,
    DATA_BITS,
    TOTAL_BITS,
    BIT_DURATION_MS,
    PAUSE_DURATION_US,
    TRANSMIT_INTERVAL_MS,
    SERIAL_BAUD_RATE
};

// Ініціалізація стану
TransmitterStatus status = {STATE_INITIALIZING, 0x05, 0};

// Функція перевірки коректності конфігурації
bool validateConfig() {
    if (config.laserPin < 0 || config.laserPin > 255) {
        LOG_MESSAGE("Invalid laser pin configuration");
    }
}

```

```

        return false;
    }
    if (config.totalBits != config.dataBits + 3) {
        LOG_MESSAGE("Invalid bit configuration");
        return false;
    }
    return true;
}

// Функція ініціалізації апаратного забезпечення
void initializeHardware() {
    pinMode(config.laserPin, OUTPUT); // Налаштування піна лазера як вихід
    SET_LASER_STATE(LASER_OFF);      // Вимкнення лазера на старті
    Serial.begin(config.serialBaudRate); // Ініціалізація послідовного порту
    while (!Serial) {} // Очікування готовності порту
    LOG_MESSAGE("Transmitter hardware initialized");
    status.state = STATE_RUNNING; // Оновлення стану
}

// Функція для кодування одного біта Хеммінга
void encodeHammingBit(uint8_t data, uint8_t* codeword, int bitIndex) {
    switch (bitIndex) {
        case 2: codeword[2] = (data >> 0) & 1; break; // D1
        case 4: codeword[4] = (data >> 1) & 1; break; // D2
        case 5: codeword[5] = (data >> 2) & 1; break; // D3
        case 6: codeword[6] = (data >> 3) & 1; break; // D4
    }
}

// Функція для обчислення перевірочних бітів
void calculateParityBits(uint8_t* codeword) {
    codeword[0] = codeword[2] ^ codeword[4] ^ codeword[6]; // P1
    codeword[1] = codeword[2] ^ codeword[5] ^ codeword[6]; // P2
    codeword[3] = codeword[4] ^ codeword[5] ^ codeword[6]; // P3
}

// Функція для кодування даних за кодом Хеммінга
void encodeHammingByte(uint8_t data, uint8_t* codeword) {
    for (int i = 0; i < config.dataBits; i++) {
        encodeHammingBit(data, codeword, i + 2);
    }
    calculateParityBits(codeword);
}

// Функція для передачі одного біта
void transmitSingleBit(uint8_t bit) {
    SET_LASER_STATE(bit ? LASER_ON : LASER_OFF); // Встановлення стану лазера
    delay(config.bitDuration); // Час передачі біта
    SET_LASER_STATE(LASER_OFF); // Вимкнення лазера
    delayMicroseconds(config.pauseDuration);
}

// Функція для передачі закодованих бітів
void transmitCodeword(uint8_t* codeword) {
    for (int i = 0; i < config.totalBits; i++) {
        transmitSingleBit(codeword[i]); // Передача кожного біта
    }
    status.transmissionCount++; // Збільшення лічильника передач
}

// Функція для форматування закодованих бітів для виведення
String formatCodeword(uint8_t* codeword) {
    String output = "0b";
    for (int i = 0; i < config.totalBits; i++) {
        output += String(codeword[i]);
    }
    return output;
}

```

```

// Функція для виведення закодованих бітів в консоль
void printCodeword(uint8_t* codeword) {
    String message = "Transmitted codeword: " + formatCodeword(codeword);
    LOG_MESSAGE(message);
}

// Функція для оновлення даних
void updateData() {
    status.currentData = (status.currentData + 1) % 16; // Циклічна зміна даних
}

// Основна функція налаштування
void setup() {
    if (!validateConfig()) {
        status.state = STATE_ERROR;
        LOG_MESSAGE("Configuration validation failed");
        return;
    }
    initializeHardware(); // Ініціалізація апаратного забезпечення
}

// Основний цикл програми
void loop() {
    if (status.state != STATE_RUNNING) {
        LOG_MESSAGE("Transmitter in error state, halting");
        delay(5000);
        return;
    }

    uint8_t codeword[TOTAL_BITS]; // Буфер для кодованих даних
    encodeHammingByte(status.currentData, codeword); // Кодування даних
    transmitCodeword(codeword); // Передача закодованих бітів
    printCodeword(codeword); // Виведення даних
    updateData(); // Оновлення даних
    delay(config.transmitInterval); // Затримка перед наступною передачею
}

```

Приймач

```

#include <Arduino.h>

// Макроси для спрощення операцій
#define LOG_MESSAGE(msg) Serial.println(String("INFO: ") + msg)

// Конфігураційні константи
const int PHOTO_PIN = 34; // Пін для фототранзистора BPW85C
const int TOTAL_BITS = 7; // Загальна кількість бітів
const int BIT_DURATION_MS = 10; // Тривалість біта в мілісекундах (100 Гц)
const int SAMPLES_PER_BIT = 10; // Кількість зразків для фільтрування
const int LIGHT_THRESHOLD = 1500; // Поріг для розрізнення 0/1
const int RECEIVE_INTERVAL_MS = 1000; // Інтервал між зчитуваннями
const int SERIAL_BAUD_RATE = 115200; // Швидкість послідовного порту
const int SAMPLE_DELAY_US = 900; // Затримка між зразками в мікросекундах

// Перелічення для станів програми
enum ProgramState {
    STATE_INITIALIZING,
    STATE_RUNNING,
    STATE_ERROR
};

// Структура для конфігурації приймача
struct ReceiverConfig {
    int photoPin;
    int totalBits;
}

```



```

    int bitDuration;
    int samplesPerBit;
    int threshold;
    int receiveInterval;
    int serialBaudRate;
    int sampleDelay;
};

// Структура для статистики
struct ReceiverStats {
    int errors;
    int receivedBits;
    int totalBitsSent;
    uint32_t packetCount;
};

// Структура для стану програми
struct ReceiverStatus {
    ProgramState state;
    uint8_t expectedData;
};

// Ініціалізація конфігурації
ReceiverConfig config = {
    PHOTO_PIN,
    TOTAL_BITS,
    BIT_DURATION_MS,
    SAMPLES_PER_BIT,
    LIGHT_THRESHOLD,
    RECEIVE_INTERVAL_MS,
    SERIAL_BAUD_RATE,
    SAMPLE_DELAY_US
};

// Ініціалізація статистики
ReceiverStats stats = {0, 0, 0, 0};

// Ініціалізація стану
ReceiverStatus status = {STATE_INITIALIZING, 0x05};

// Функція перевірки коректності конфігурації
bool validateConfig() {
    if (config.photoPin < 0 || config.photoPin > 255) {
        LOG_MESSAGE("Invalid photo pin configuration");
        return false;
    }
    if (config.totalBits != 7) {
        LOG_MESSAGE("Invalid bit configuration");
        return false;
    }
    return true;
}

// Функція ініціалізації апаратного забезпечення
void initializeHardware() {
    pinMode(config.photoPin, INPUT); // Налаштування піна фототранзистора
    Serial.begin(config.serialBaudRate); // Ініціалізація послідовного порту
    while (!Serial) {} // Очікування готовності порту
    LOG_MESSAGE("Receiver hardware initialized");
    status.state = STATE_RUNNING; // Оновлення стану
}

// Функція для зчитування одного біта
int readSingleBit() {
    int sum = 0;
    for (int i = 0; i < config.samplesPerBit; i++) {
        sum += analogRead(config.photoPin); // Зчитування аналогового сигналу
        delayMicroseconds(config.sampleDelay); // Розподіл зразків
    }
}

```

```

    return (sum / config.samplesPerBit > config.threshold) ? 1 : 0;
}

// Функція для обчислення закодованих бітів
uint8_t calculateSyndrome(uint8_t* codeword) {
    uint8_t syndrome = 0;
    syndrome |= (codeword[0] ^ codeword[2] ^ codeword[4] ^ codeword[6]) << 0; // S1
    syndrome |= (codeword[1] ^ codeword[2] ^ codeword[5] ^ codeword[6]) << 1; // S2
    syndrome |= (codeword[3] ^ codeword[4] ^ codeword[5] ^ codeword[6]) << 2; // S3
    return syndrome;
}

// Функція для декодування Хеммінга
uint8_t decodeHammingCode(uint8_t* codeword) {
    uint8_t syndrome = calculateSyndrome(codeword);
    if (syndrome != 0 && syndrome <= config.totalBits) {
        codeword[syndrome - 1] ^= 1; // виправлення помилки
    }
    return (codeword[6] << 3) | (codeword[5] << 2) | (codeword[4] << 1) |
codeword[2];
}

// Функція для оновлення очікуваних даних
void updateExpectedData() {
    status.expectedData = (0x05 + (stats.totalBitsSent / config.totalBits) % 16) %
16;
}

// Функція для оновлення статистики
void updateStats(uint8_t data) {
    stats.receivedBits += config.totalBits;
    stats.totalBitsSent += config.totalBits;
    stats.packetCount++;
    if (data != status.expectedData) {
        stats.errors++;
    }
}

// Функція для форматування закодованих даних
String formatCodeword(uint8_t* codeword) {
    String output = "0b";
    for (int i = 0; i < config.totalBits; i++) {
        output += String(codeword[i]);
    }
    return output;
}

// Функція для виведення отриманих даних
void printReceivedData(uint8_t* codeword, uint8_t data) {
    String message = "Received codeword: " + formatCodeword(codeword);
    message += " Decoded data: 0b" + String(data, BIN);
    message += " Errors: " + String(stats.errors);
    message += " Loss: " + String((stats.errors * 100.0) / (stats.totalBitsSent /
config.totalBits)) + "%";
    LOG_MESSAGE(message);
}

// Основна функція налаштування
void setup() {
    if (!validateConfig()) {
        status.state = STATE_ERROR;
        LOG_MESSAGE("Configuration validation failed");
        return;
    }
    initializeHardware(); // Ініціалізація апаратного забезпечення
}

// Основний цикл програми
void loop() {

```

```
if (status.state != STATE_RUNNING) {
    LOG_MESSAGE("Receiver in error state, halting");
    delay(5000);
    return;
}

uint8_t codeword[TOTAL_BITS]; // Буфер для закодованих даних
for (int i = 0; i < config.totalBits; i++) {
    codeword[i] = readSingleBit(); // Зчитування кожного біта
    delay(config.bitDuration); // Синхронізація
}

uint8_t data = decodeHammingCode(codeword); // Декодування
updateStats(data); // Оновлення статистики
printReceivedData(codeword, data); // Виведення даних
updateExpectedData(); // Оновлення очікуваних даних
delay(config.receiveInterval); // Затримка перед наступним прийомом
}
```

КБПЗ_2025