

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет програмних систем та прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем**

До захисту допущено:

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Системне програмування та спеціалізовані комп'ютерні системи»

спеціальності 123 «Комп'ютерна інженерія»

**на тему: «Програмно-апаратна система моніторингу якості повітря на
основі плати розробника сімейства STM32»**

Виконав:

студент IV курсу, групи KB-22

Міндер Вадим Юрійович _____

Керівник:

доц. каф. СП і СКС, к.т.н

Марченко Олександр Іванович _____

Консультант з нормконтролю:

доц. каф. СП і СКС, к.т.н

Клятченко Ярослав Михайлович _____

Рецензент:

доц. каф. ПЗКС, к.т.н

Заболотня Тетяна Миколаївна _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет програмних систем та прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____Віталій РОМАНКЕВИЧ

«___»_____20__ р.

ЗАВДАННЯ
на дипломний проєкт студенту
Міндеру Вадиму Юрійовичу

1. Тема проєкту «Програмно-апаратна система моніторингу якості повітря на основі плати розробника сімейства STM32», керівник проєкту Марченко О.І., кандидат технічних наук, доцент кафедри системного програмування і спеціалізованих комп'ютерних систем, затверджені наказом по університету від «28» травня 2026р. № 1984-с

2. Термін подання студентом проєкту _____

3. Вихідні дані проєкту: див. Технічне завдання.

4. Зміст пояснювальної записки

- аналіз існуючих рішень;
- розробка апаратного забезпечення системи моніторингу;
- розробка програмного забезпечення системи моніторингу;
- тестування;

- керівництво користувача.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій, тощо)

- функціональна схема системи;
- структурна схема імпорту програмних модулів системи;
- схема алгоритму центрального вузла;
- схема алгоритму периферійного вузла.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., к.т.н., доцент каф. СП і СКС		

7. Дата видачі завдання: 30 жовтня 2025р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	13.11.2025	
2.	Розроблення та узгодження технічного завдання	17.11.2025	
3.	Аналіз існуючих рішень	03.12.2025	
4.	Підготовка матеріалів першого розділу дипломного проєкту	13.01.2026	
5.	Розробка апаратного забезпечення	20.01.2026	
6.	Підготовка матеріалів другого розділу дипломного проєкту	09.02.2026	
7.	Розробка програмного забезпечення	15.05.2026	

8.	Підготовка матеріалів третього розділу дипломного проєкту	20.05.2026	
10.	Підготовка графічної частини дипломного проєкту	23.05.2026	
9.	Підготовка матеріалів четвертого розділу дипломного проєкту	27.05.2026	
9.	Підготовка матеріалів п'ятого розділу дипломного проєкту	29.05.2026	
11.	Оформлення документації дипломного проєкту	30.05.2026	
12.	Попередній огляд матеріалів диплому на кафедрі	30.05.2026	

Студент

(підпис)

Вадим МІНДЕР

Керівник проєкту

(підпис)

Олександр МАРЧЕНКО

АНОТАЦІЯ

Бакалаврський дипломний проєкт включає пояснювальну записку (60 стор., 23 рис., 1 табл., список використаної літератури з 20 найменувань, 4 додатки).

Об'єкт розробки – програмно-апаратна система моніторингу якості повітря на основі плати розробника сімейства STM32, яка забезпечує збір, обробку, збереження та передавання даних про температуру, вологість і атмосферний тиск.

Розроблена система дозволяє: здійснювати вимірювання кліматичних параметрів за допомогою сенсора BME280; обмінюватися даними між вузлами за допомогою технології LoRa; передавати інформацію на сервер для подальшого відображення на вебсторінці; зберігати історію вимірювань на SD-карті; відображати дані на TFT-дисплеї; забезпечувати енергоефективну роботу периферійних вузлів шляхом використання режимів сну мікроконтролера. У процесі розробки були використані мікроконтролер STM32F411CEU6, сенсор BME280, LoRa-модуль XL1278-SMT, Wi-Fi-модуль ESP-01S, TFT-дисплей ST7735S, модуль SD та програмна бібліотека FatFs.

В ході розробки:

- проведено аналіз існуючих систем моніторингу якості повітря;
- сформульовано функціональні вимоги до системи;
- розроблено структуру програмно-апаратної системи;
- реалізовано програмне забезпечення для збору, обробки та передавання даних;
- реалізовано механізми збереження даних у постійній пам'яті;
- розроблено механізм бездротового обміну даними між вузлами;
- виконано тестування системи та оптимізацію енергоспоживання.

Розроблена система може бути використана для автономного моніторингу параметрів мікроклімату у житлових, комерційних та промислових приміщеннях.

Ключові слова:

ПРОГРАМНО-АПАРАТНА СИСТЕМА, STM32, МОНІТОРИНГ ЯКОСТІ
ПОВІТРЯ, LORA, BME280, FATFS, ВБУДОВАНА СИСТЕМА,
ЕНЕРГОЕФЕКТИВНІСТЬ, БЕЗДРОТОВА ПЕРЕДАЧА ДАНИХ.

ABSTRACT

The bachelor's diploma project includes an explanatory note (60 pages, 23 figures, 1 table, a list of used literature with 20 titles, 4 Appendices).

The object of development is a software and hardware air quality monitoring system based on the STM32 family development board, which provides the collection, processing, storage, and transmission of data on temperature, humidity, and atmospheric pressure.

The developed system allows to: measure climatic parameters using the BME280 sensor; exchange data between nodes using LoRa technology; transmit information to a server for further display on a webpage; store the measurement history on a SD card; display data on a TFT display; and ensure the energy-efficient operation of peripheral nodes by utilizing microcontroller sleep modes.

During the development process, the STM32F411CEU6 microcontroller, BME280 sensor, XL1278-SMT LoRa module, ESP-01S Wi-Fi module, ST7735S TFT display, SD module, and FatFs software library were used.

Development process included:

- An analysis of existing air quality monitoring systems was conducted;
- functional requirements for the system were formulated;
- the structure of the software and hardware system was developed;
- software for data collection, processing, and transmission was implemented;
- mechanisms for storing data in non-volatile memory were implemented;
- a mechanism for wireless data exchange between nodes was developed;
- system testing and energy consumption optimization were performed.

Keywords:

SOFTWARE AND HARDWARE SYSTEM, STM32, AIR QUALITY MONITORING, LORA, BME280, FATFS, EMBEDDED SYSTEM, ENERGY EFFICIENCY, WIRELESS DATA TRANSMISSION.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.468300.002 ТЗ	Програмно-апаратна система моніторингу показників периферійних датчиків на основі плати розробника сімейства STM32	2	A4	
			Технічне завдання			
	A4	ІАЛЦ.468300.003 ТП	Програмно-апаратна система моніторингу показників периферійних датчиків на основі плати розробника сімейства STM32	1	A4	
			Відомість технічного проекту			
	A4	ІАЛЦ.468300.004 ПЗ	Програмно-апаратна система моніторингу показників периферійних датчиків на основі плати розробника сімейства STM32	60	A4	
			Пояснювальна записка			

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.468300.005 Д1	Схема підключення центрального вузла.	1	A4	
			Схема функціональна			
	A4	ІАЛЦ.468300.006 Д2	Схема імпорту програмних модулів системи.	1	A4	
			Схема структурна			
	A4	ІАЛЦ.468300.007 Д3	Алгоритм роботи центрального вузла.	1	A4	
			Схема алгоритму			
	A4	ІАЛЦ.468300.008 Д4	Алгоритм роботи периферійного вузла.	1	A4	
			Схема алгоритму			
		Диск CD-ROM	Текст пояснювальної записки.			
			Графічний матеріал			

Змін.	Арк.	№ докум.	Підпис	Дата
-------	------	----------	--------	------

ІАЛЦ.468300.001 ОА

Арк.
2

ЗМІСТ

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОБОТИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1 Вимоги до модулю програмного забезпечення, що розробляється	3
5.2 Рекомендовані вимоги до апаратного забезпечення	3
5.3 Вимоги до мінімального програмного забезпечення	3
6. ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ.468300.002 ТЗ		
Зм.	Арк.	№ докум.	Підп.	Дата			
Розроб.		Міндер В.Ю. Марченко О.І.			Програмно-апаратна система моніторингу якості повітря на основі плати розробника сімейства STM32 Технічне завдання	Літ.	Аркуш
							Аркушів
Н. контр.		Клятченко Я.М.			КПІ ім. Ігоря Сікорського, ФПСІМ, КВ-22		
Затв.		Романкевич В.О.					

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування проєкту – “Програмно-апаратна система моніторингу якості повітря на основі плати розробника сімейства STM32”.

Область застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп’ютерних систем Національного технічного університету України “Київський Політехнічний Інститут імені Ігоря Сікорського”.

3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ

Метою дипломного проєкту є розробка універсальної програмно-апаратної системи моніторингу якості повітря на основі плати розробника сімейства STM32 із можливістю збору, обробки та візуалізації даних у реальному часі.

4. ДЖЕРЕЛА РОБОТИ

Джерелами інформації для розроблення є технічна література, публікації у періодичних виданнях та Інтернет ресурси.

					ІАЛЦ.468300.002 ТЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до модулю програмного забезпечення, що розробляється

- підтримка датчиків BME280;
- підтримка Wi-Fi-з'єднання центральним вузлом;
- підтримка LoRa-зв'язку між центральним та периферійними вузлами;
- можливість динамічного підключення та заміни периферійних вузлів під час роботи системи;
- можливість відображення показників датчиків на екрані центрального вузла;
- можливість перегляду показників датчиків та керування системою моніторингу за допомогою вебсайту;
- можливість підключення багатьох периферійних вузлів до центрального;
- підтримка SD-карти пам'яті, збереження в ній історії показників датчиків.

5.2 Рекомендовані вимоги до апаратного забезпечення

- процесор: AMD Ryzen 5 або подібний;
- оперативна пам'ять: 4 Гб;
- простір на диску: 1 Гб;
- плата для розробки на базі STM32F4xx;
- модулі датчиків з підтримуваними мікросхемами (BME280);
- модуль LoRa-антени SX1278 та модуль Wi-Fi ESP-01S;
- модуль екрану з контролерами екрану ST7735S;
- модуль SD-карти

5.3 Вимоги до мінімального програмного забезпечення

- операційна система Linux, Windows;
- набір інструментів CubeIde;

					ІАЛЦ.468300.002 ТЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		3

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту
1.	Вивчення літератури за тематикою проєкту	13.11.2025
2.	Розроблення та узгодження технічного завдання	17.11.2025
3.	Аналіз існуючих рішень	03.12.2025
4.	Підготовка матеріалів першого розділу дипломного проєкту	13.01.2026
5.	Розробка апаратного забезпечення	20.01.2026
6.	Підготовка матеріалів другого розділу дипломного проєкту	09.02.2026
7.	Розробка програмного забезпечення	15.05.2026
8.	Підготовка матеріалів третього розділу дипломного проєкту	20.05.2026
9.	Підготовка графічної частини дипломного проєкту	23.05.2026
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	27.05.2026
11.	Підготовка матеріалів п'ятого розділу дипломного проєкту	29.05.2026
12.	Оформлення документації дипломного проєкту	30.05.2026
13.	Попередній огляд матеріалів диплому на кафедрі	30.05.2026

ВІДОМІСТЬ ТЕХНІЧНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4	ІАЛЦ.468300.000	Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.468300.001 ОА	Опис альбому	2	
3	A4	ІАЛЦ.468300.002 ТЗ	Технічне завдання	4	
4	A4	ІАЛЦ.468300.003 ТП	Відомість технічного проєкту	1	
5	A4	ІАЛЦ.468300.004 ПЗ	Пояснювальна записка	60	
6	A4	ІАЛЦ.468300.005 Д1	Схема підключення центрального вузла. Схема функціональна	1	
7	A4	ІАЛЦ.468300.006 Д2	Взаємодія програмних модулів. Схема структурна	1	
8	A4	ІАЛЦ.468300.007 Д3	Алгоритм роботи центрального вузла. Схема алгоритму	1	
9	A4	ІАЛЦ.468300.008 Д4	Алгоритм роботи периферійного вузла. Схема алгоритму	1	

				ІАЛЦ.468300.003 ТП		
	ПІБ	Підп.	Дата			
Розробн.	Міндер В.Ю.			Програмно-апаратна система моніторингу якості повітря на основі плати розробника сімейства STM32 Відомість технічного проєкту	Лист	Листів
Керівн.	Марченко О.І.				1	1
Консульт.					КПІ ім. Ігоря Сікорського Каф. СП і СКС Гр. КВ-22	
Н/контр.	Клятченко Я.М.					
Зав.каф.	Романкевич В.О.					

Пояснювальна записка до дипломного проєкту
на тему: Програмно-апаратна система моніторингу якості повітря на
основі плати розробника сімейства STM32

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	4
ВСТУП.....	5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	6
1.1 Sencor SWS4900.....	6
1.2 Xiaomi Smart Temperature and Humidity Monitor 3.....	6
1.3 Ajax LifeQuality Jeweller.....	7
1.4 uutzinger Airquality.....	7
1.5 Обґрунтування теми дипломного проєкту та постановка задачі.....	8
2. РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ	10
2.1 Визначення складових компонентів системи моніторингу	10
2.1.1 Мікроконтролер	10
2.1.2 Датчики	11
2.1.3 Пам'ять.....	12
2.1.4 Дисплей	13
2.1.5 Бездротовий зв'язок.....	14
2.1.6 Система живлення.....	16
2.2 Розробка апаратної частини системи моніторингу.....	17
2.2.1 Апаратна частина центрального вузла	17
2.2.2 Апаратна частина периферійного вузла	18
3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ	20
3.1. Структурна схема програмних модулів системи моніторингу	20
3.2 Модуль датчика показників навколишнього середовища	21
3.2.1 I2C.....	21

					ІАЛЦ.468300.004 ПЗ			
Зм	Арк.	№ докум.	Підп.	Дата	Програмно-апаратна система моніторингу якості повітря на основі плати розробника сімейства STM32 Пояснювальна записка	Літ.	Аркуш	Аркушів
Розроб.	Міндер В.Ю.						1	60
	Марченко О.І.							
Н. контр.	Клятченко Я.М.					КПІ ім. Ігоря Сікорського, ФПСІМ, КВ-22		
Затв.	Романкевич В.О.							

3.2.2	Драйвер датчика BME280	22
3.3	Модуль бездротового зв'язку	23
3.3.1	SPI	23
3.3.2	Драйвер LoRa-модему XL1278-SMT	24
3.3.3	Протокол пакету та CRC перевірка.....	25
3.3.4	UART	25
3.3.5	Драйвер Wi-Fi-модуля ESP-01S	26
3.4	Модуль використання постійної пам'яті.....	26
3.4.1	Драйвер SD-карти	27
3.4.2	Алгоритм збереження даних по вузлах	27
3.5	Модуль дисплею	28
3.5.1	Графічна бібліотека	29
3.5.2	Драйвер дисплею	30
3.5.3	Графічний інтерфейс	31
3.6	Програмна реалізація центрального вузла	32
3.6.1	Алгоритм ініціалізації	32
3.6.2	Алгоритм кешування даних та керування опитуванням вузлів.....	33
3.7	Програмна реалізація периферійного вузла	35
3.7.1	Алгоритм збору та формування пакету даних.....	36
3.7.2	Режими енергозбереження.....	36
3.8	Серверна частина системи моніторингу	38
3.8.1	TCP-сервер для прийому даних.....	38
3.8.2	Алгоритм обробки та запису даних до БД	38
3.9	Програмна реалізація вебінтерфейсу	39
3.9.1	Структура вебсайту.....	39

3.9.2 Реалізація візуалізації даних та статистики	40
4. ТЕСТУВАННЯ	42
4.1 Тестування зчитування показників датчика	42
4.2 Тестування збору даних з периферійних вузлів.....	43
4.3 Тестування відображення показників	46
4.4 Тестування вебсайту	47
4.5 Тестування збереження та перегляду історії вимірювань.....	48
4.6 Тестування енергоефективності вузлів.....	50
5. КЕРІВНИЦТВО КОРИСТУВАЧА	51
5.1 Інтерфейс користувача	51
5.2 Підключення до Wi-Fi мережі.....	52
5.3 Підключення нових вузлів.....	53
5.4 Перегляд історії вимірювань	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	58

ДОДАТКИ

Додаток 1. Копії графічних матеріалів

- ІАЛЦ.468300.005 Д1. Схема підключення центрального вузла. Схема функціональна
- ІАЛЦ.468300.006 Д2. Схема імпорту програмних модулів системи. Схема структурна
- ІАЛЦ.468300.007 Д3. Алгоритм роботи центрального вузла. Схема алгоритму
- ІАЛЦ.468300.008 Д4. Алгоритм роботи периферійного вузла. Схема алгоритму

Додаток 2. Фрагменти програмного коду

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

DMA – Direct Memory Access

FAT – File Allocation Table

HAL – Hardware Abstraction Layer

I2C – Inter-Integrated Circuit

IRQ – Interrupt Request

LoRa – Long Range

NVIC – Nested Vectored Interrupt Controller

RX – Receive

SCL – Serial Clock Line

SDA – Serial Data Line

SPI – Serial Peripheral Interface

TX – Transmit

UART – Universal Asynchronous Receiver-Transmitter

Wi-Fi – Wireless Fidelity

					ІАЛЦ.468300.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		4

ВСТУП

В наш час технології Інтернету речей активно впроваджуються у різноманітні сфери життя, забезпечуючи автоматизацію процесів та безперервний збір даних із датчиків у реальному часі. Сучасні тенденції демонструють, що користувачі потребують не лише швидкого доступу до кліматичних показників, а й надійної роботи системи незалежно від масштабу приміщення чи території. Зазвичай для таких задач застосовують монолітні системи з інтегрованим Wi-Fi, проте вони мають суттєві обмеження щодо дальності зв'язку, автономності та гнучкості при побудові великих мереж. У таких випадках значною перевагою є використання мікроконтролерів сімейства STM32. Вони дозволяють побудувати розподілену архітектуру та гнучко взаємодіяти з різноманітною периферією: датчиками, далекобійними радіомодулями, картами пам'яті, дисплеями. Завдяки розвиненій системі контролера векторних переривань (NVIC) та можливості прямого доступу до пам'яті (DMA), ця платформа дає змогу ефективно оптимізувати процеси обробки – наприклад, поєднувати прийом даних, локальне логування та оновлення екрана, забезпечуючи миттєву реакцію апаратних.

Завданням дипломного проєкту є створення розподіленої програмно-апаратної системи моніторингу якості повітря на основі плати розробника сімейства STM32. У межах проєкту розроблено механізм збору даних від периферійних вузлів, їх передачі на великі відстані за допомогою технології LoRa та їх обробки на центральному вузлі. Центральний вузол забезпечує опрацювання даних у режимі реального часу, їх збереження та візуалізацію на TFT-дисплеї, а також інтеграцію з мережею через додатковий Wi-Fi модуль. Продумана структура програмного забезпечення гарантує високу надійність і безперебійність обробки інформації, а продуктивність та масштабованість STM32 роблять розроблене рішення стійким до перебоїв у каналах зв'язку та привабливим для подальшого застосування у галузі IoT.

					ІАЛЦ.468300.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		5

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Щоб визначити найбільш ефективний шлях для створення власного рішення, потрібно проаналізувати вже існуючі рішення на ринку, зрозуміти їх переваги та недоліки. Таке порівняння дозволяє побачити загальну картину ринку та складнощі, які виникають при розробці подібних пристроїв. При розробці систем моніторингу клімату важливо враховувати типи сенсорів та їх характеристики, а й зручність у розробці, можливість інтеграції з популярними платформами, тому розглянемо декілька існуючих та найбільш поширених варіантів, які пропонують виробники.

1.1 Sencor SWS4900

Sencor – японська компанія з виробництва побутової техніки, заснована у 1969 році, має великий асортимент різної техніки, серед якої є метеостанція Sencor SWS4900, складається з базової станції, яка обладнана кольоровим дисплеєм, має власні датчики температури, вологості та тиску, працює від 5В постійного струму та має резервне живлення у вигляді трьох батарейок типу АА. Також додатково комплектується одним зовнішнім датчиком SWS TH3600-4500-4900, який вимірює температуру, вологість, тиск, та передає ці дані на базову станцію по радіоканалу на частоті 433,92 МГц на відстань до 80 метрів, живитися від двох батарейок типу АА. Загалом базова станція може підключити до 3 зовнішніх датчиків. Недоліками можна назвати те, що станція працює автономно та не забезпечує перегляд та керування даними на інших пристроях чи в мережі.

1.2 Xiaomi Smart Temperature and Humidity Monitor 3

Xiaomi – китайська компанія, що спеціалізується на розумній електроніці та побутових гаджетах. Їхня метеостанція Xiaomi Smart Temperature and Humidity

					ІАЛЦ.468300.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		6

Monitor 3 вимірює температуру та вологість у приміщенні з точністю до $\pm 0.1^{\circ}\text{C}$ та $\pm 1\%$ RH. Має монохромний дисплей, який є не дорогим та енергоефективним рішенням. Пристрій може підключитися по Bluetooth до мобільного додатку Mi Home, що дає можливість переглядати дані в реальному часі, налаштовувати автоматизацію в поєднанні з іншими пристроями Xiaomi. Живиться від батарейки CR2450, що забезпечує тривалу автономність – до 18 місяців. До переваг можна віднести доступність, можливість підключення до Mi Home та інтеграції з іншими пристроями, високу автономність.

1.3 Ajax LifeQuality Jeweller

Аjax – українська компанія, заснована в 2011 році, відома своєю системою безпеки та лінійкою пристроїв “Розумного будинку” серед яких є також пристрій для контролю якості повітря у приміщенні під назвою LifeQuality Jeweller. Має сенсори для вимірювання температури, вологості та кількості CO₂ в повітрі з точністю до $\pm 0.2^{\circ}\text{C}$, $\pm 1.8\%$ RH та $\pm 30\text{ppm}$. Для роботи пристрою та відображення даних у мобільному додатку потрібний центральний контролер Ajax Hub 2 з доступом до інтернету. Пристрій може бути розміщений на відстані до 1700 метрів від керуючого пристрою або ретранслятора сигналу Ajax ReX 2, що дозволяє розміщувати пристрій на значній відстані. Інформація передається по власному захищеному радіопротоколу Jeweller. Живиться від двох батарейок CR123A, що в поєднанні з високою енергоефективністю дає можливість працювати до 3 років на одному заряді. Недоліком є висока ціна як самого пристрою так і центрального контролера, також, закрита архітектура не дає можливості власної модифікації.

1.4 uutzinger Airquality

Проект системи моніторингу параметрів якості повітря з відкритим кодом від користувача uutzinger на сайті GitHub, який побудований на базі мікроконтролера Espressif Systems ESP8266. Пристрій призначений для

					ІАЛЦ.468300.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		7

вимірювання показників якості повітря та передачі отриманих даних через бездротову мережу Wi-Fi. Система підтримує використання сенсорів BME280/BME680 для вимірювання температури, вологості, атмосферного тиску та параметрів якості повітря. Отримані дані можуть передаватися на MQTT-сервер, веб-інтерфейс або систему домашньої автоматизації Home Assistant. Для відображення інформації використовується OLED-дисплей, який дозволяє локально переглядати основні параметри роботи системи. Передача даних здійснюється через Wi-Fi, що забезпечує просту інтеграцію з локальною мережею та IoT-сервісами. Пристрій підтримує режими енергозбереження Deep Sleep, що дозволяє зменшити енергоспоживання при автономній роботі. Перевагами проекту є відкритий вихідний код, підтримка MQTT та Home Assistant, простота інтеграції з існуючими IoT-системами, а також можливість модифікації. Недоліками є залежність від Wi-Fi мережі, відносно велике енергоспоживання у порівнянні з LoRa-пристроями, а також невелика дальність бездротового зв'язку.

1.5 Обґрунтування теми дипломного проекту та постановка задачі

Аналіз вищезгаданих рішень показав, що сучасний ринок пристроїв пропонує низку варіантів для вимірювання кліматичних показників у приміщенні та на вулиці (Sencor). В цих рішеннях використовується закрита архітектура, що не дає можливості для додавання інших сенсорів, не передбачених розробником. Також ці рішення маю обмежені можливості по вимірюванню показників в інших кімнатах або на вулиці, в Sencor обмежена кількість зовнішніх датчиків, у Ажах та Хіаомі потрібно докуповувати центральні керуючі пристрої для відстеження декількох пристроїв та такі ж кліматичні станції, у випадку пристрою Ажах – за досить велику ціну. Також рішення від Ажах та Хіаомі не маю достатнього рівня захисту від вологи, що не дозволяє встановити їх на вулиці. Пристрій від Sencor немає можливості перегляду та керування даними з мобільного телефону, комп'ютера або веб браузера. Рішення

від Ajax та Xiaomi хоч і мають таку можливість, але залежать від серверів компанії та наявності підключення до інтернету.

З іншого боку, існують відкриті некомерційні проекти (Airquality від uutzinger), які частково вирішують проблеми закритості архітектури та відкрито передають дані локально. Однак подібні рішення, побудовані на базі мікроконтролерів із вбудованим Wi-Fi, мають власні недоліки для побудови розподілених систем. Вони жорстко прив'язані до радіуса дії локальної бездротової мережі, що унеможливорює збір даних на великих територіях та відзначаються відносно високим енергоспоживанням, що суттєво знижує час автономної роботи віддалених точок у порівнянні з використанням радіомодулів.

Актуальність дипломного проекту полягає у відсутності на ринку пристрою, який поєднував би в собі можливість віддаленого керування пристроєм з можливістю підключення великої кількості зовнішніх вузлів, що дозволить відстежувати клімат у кожній кімнаті в будинку та зовні, також можливість підключення багатьох датчиків на значній відстані дозволяє використовувати рішення на виробництвах для контролю як клімату так і інших даних при заміні сенсорів у простий спосіб за допомогою відкритого коду. Використання мікроконтролера STM32 дозволяє використовувати велику кількість периферійних модулів (датчики температури, вологості, тиску), висока енергоефективність дає можливість зробити велику автономність від мережі для пристрою.

2. РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ

Розробка апаратної складової є одним із визначальних етапів створення програмно-апаратної системи моніторингу периферійних датчиків. Апаратна частина відповідає за безперервний збір даних, їхню первинну обробку та подальшу передачу для відображення та аналізу, а інтерфейс забезпечує зручний доступ до даних, а також реалізує функцію керування, яка потрібна для ефективної взаємодії користувача з системою

2.1 Визначення складових компонентів системи моніторингу

Апаратна частина розробленої системи моніторингу якості повітря побудована з функціональних модулів, що забезпечують повний цикл роботи з даними – від зчитування до відображення користувачу. Основою системи є мікроконтролер, який обробляє отриману інформацію та координує роботу всіх підсистем. Апаратної частина містить в собі периферійний датчик, який здійснює вимірювання необхідних параметрів, модуль SD-карти, дисплей, модуль Wi-Fi-зв'язку, модуль LoRa-антени, взаємодія цих елементів дозволяє реалізувати безперервний цикл збору, обробки й візуалізації даних у реальному часі.

2.1.1 Мікроконтролер

Головним елементом апаратної частини центрального та периферійних вузлів системи моніторингу є мікроконтролер, який забезпечує координацію роботи всіх компонентів та обробку отриманих даних. У цій розробці використовується налагоджувальна плата STM32F411CEU6 (рис. 2.1) на базі мікроконтролера серії STM32F4, який оптимально підходить для створення вбудованих систем із помірними вимогами до обчислювальних ресурсів та енергоспоживання, таке рішення дозволяє ефективно обробляти дані з декількох

модулів у реальному часі, забезпечувати взаємодію з іншими мікроконтролерами, зберігати необхідну інформацію у пам'яті. Також він має невелике енергоспоживання, що дає високу автономність периферійних вузлів.

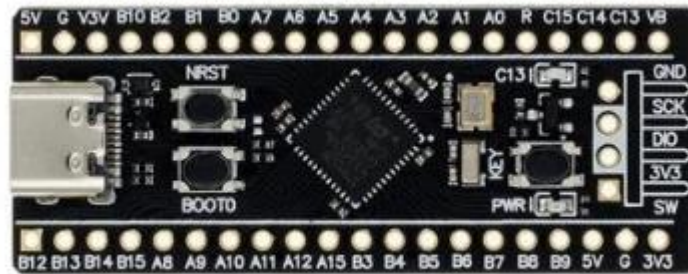


Рисунок 2.1 – STM32F411CEU6

2.1.2 Датчики

Датчики є ключовими елементами системи моніторингу, оскільки саме вони забезпечують отримання інформації про навколишнє середовище, вони здійснюють вимірювання необхідних фізичних параметрів та передають отримані дані до мікроконтролеру для подальшої обробки, відображення та збереження.

У складі розробленої системи моніторингу передбачено використання датчика BME280. Цей модуль поєднує датчик атмосферного тиску, термометр і датчик вологості повітря, він забезпечує точні вимірювання при досить низькому енергоспоживанні. Може бути використаний також для визначення висоти або зміни погодних умов. Також модуль оснащений вбудованим цифровим IIR-фільтром та підтримує механізм оверсемплінгу, що дозволяє апаратно згладжувати шуми вимірювань та підвищувати їхню точність. Датчик постачається із заводським калібруванням, індивідуальні коефіцієнти компенсації зберігаються у його внутрішній незалежній пам'яті. Використаний у проєкті модуль BME280 показано на рис. 2.2.

Використання цього датчика дозволяє забезпечити комплексний підхід до моніторингу навколишнього середовища. Датчик підтримує комунікацію за

допомогою стандартизованих інтерфейсів I2C та SPI. Це значно спрощує інтеграцію з обраним мікроконтролером.

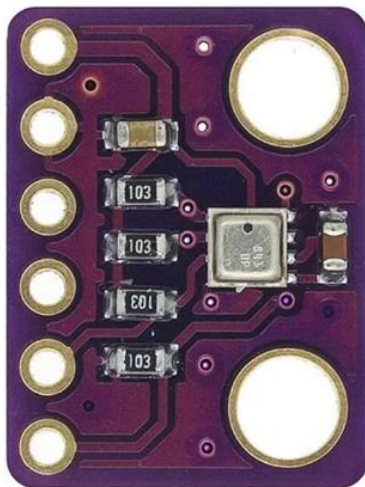


Рисунок 2.2 - BME280

2.1.3 Пам'ять

Для забезпечення тривалого зберігання зібраних даних, ведення журналу подій, а також збереження налаштувань системи використовується модуль для роботи з SD-картами. Це дозволяє створювати великі архіви показників датчиків, обсяг яких перевищує внутрішню пам'ять мікроконтролера, і забезпечує легкий доступ до даних шляхом перенесення карти на інший пристрій. У розробленій системі використовується модуль SD-карти з підтримкою інтерфейсу SPI (рис. 2.3).

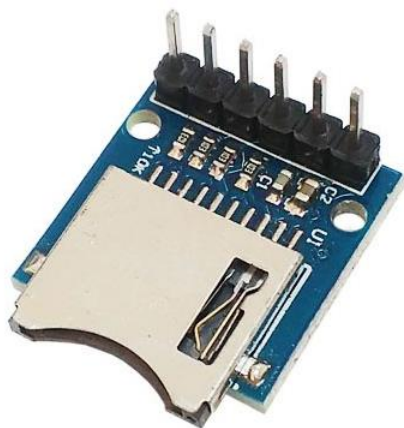


Рисунок 2.3 - Модуль SD-карти

Модулі цього типу підтримують карти пам'яті формату SD та SDHC об'ємом до 32 ГБ. Використання SD-модуля дозволяє організувати повноцінну систему логування з відмітками часу, зберігати конфігураційні файли, та гарантує цілісність даних у разі збоїв живлення, а завдяки використанню файлової системи, накопичена інформація може бути легко зчитана та проаналізована на персональному комп'ютері без використання додаткового спеціалізованого обладнання.

2.1.4 Дисплей

Для відображення показників, отриманих від периферійних датчиків, а також для візуального контролю роботи системи моніторингу, використовується кольоровий графічний TFT-дисплей.

У розробленій системі застосовується рідкокристалічний екран, побудований на базі контролера ST7735S (рис. 2.4). Даний дисплей має діагональ 1.44 дюйма та роздільну здатність 128×128 пікселів. Це забезпечує чітке відображення текстової та графічної інформації при низькому енергоспоживанні. Передача даних від мікроконтролера до дисплея здійснюється через послідовний інтерфейс SPI, використання цього протоколу дозволяє мінімізувати кількість необхідних виводів мікроконтролера, забезпечуючи при цьому достатню швидкість оновлення кадрів для виводу частоті зміни числових значень. Модуль підтримує відображення до 65 тисяч кольорів. Завдяки своїй компактності та простоті, дисплей на базі ST7735S є оптимальним рішенням для автономних систем моніторингу, де важливо поєднувати інформативність та швидкодію системи.

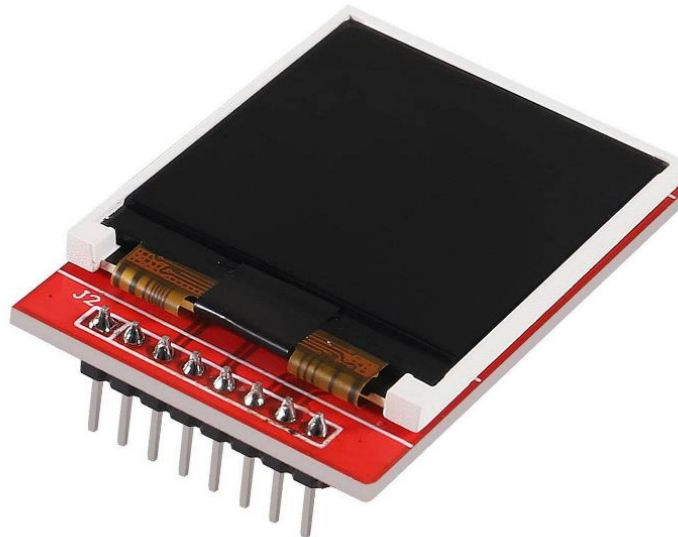


Рисунок 2.4 – TFT LCD дисплей ST7735S

2.1.5 Бездротовий зв'язок

Для передачі зібраних даних на сервер, а також для передавання даних між вузлами на великих відстанях, у системі моніторингу використано комбінований підхід з поєднанням технології Wi-Fi та LoRa.

Для підключення системи до локальної мережі та передачі інформації на власний сервер використовується малогабаритний модуль ESP-01S на базі чипа ESP8266 (рис. 2.5). Модуль працює на частоті 2.4 ГГц і підтримує стандарт 802.11 b/g/n. Взаємодія між головним мікроконтролером та ESP-01S здійснюється через інтерфейс UART за допомогою набору AT-команд та вбудованого протоколу TCP/IP, це дозволяє системі моніторингу інтегруватися в IoT-інфраструктуру та забезпечувати дистанційний доступ до показників через Wi-Fi-мережу.



Рисунок 2.5 – ESP-01S

Для забезпечення надійного зв'язку між вузлами на відстані до декількох кілометрів та мінімального енергоспоживання використано модуль XL1278-SMT (рис.2.6).



Рисунок 2.6 – XL1278-SMT

Модуль базується на чипі SX1278, який забезпечує високу стійкість при мінімальному споживанні струму. Використовуючи запатентовану Semtech техніку модуляції LoRa, SX1278 може досягти чутливості понад -148 дБм,

працює на частоті 433 МГц і підключається до мікроконтролера через інтерфейс SPI. Використання XL1278 дозволяє системі передавати дані від віддалених сенсорних вузлів до центрального шлюзу навіть у місцях, де відсутнє покриття Wi-Fi або мобільної мережі. Завдяки гнучкому налаштуванню потужності передавача та швидкості передачі даних, досягається оптимальний баланс між дальністю зв'язку та тривалістю автономної роботи пристрою.

Ці два компоненти разом утворюють гібридну систему зв'язку: LoRa забезпечує збір даних з віддалених вузлів, а ESP-01S відповідає за передачу отриманої інформації на власний сервер.

2.1.6 Система живлення

Враховуючи розподілену структуру розробленої системи моніторингу, підсистема живлення організована за таким принципом: центральний вузол працює від стаціонарної мережі, а периферійні вузли функціонують повністю автономно.

Для забезпечення безперебійної роботи центрального вузла, який виконує задачі зі збору, обробки, відображення даних та їх передачі на сервер, використовується зовнішній мережевий блок живлення, використання стаціонарного джерела знімає жорсткі обмеження на енергоспоживання, що дозволяє стабільно жити такі активні компоненти, як графічний дисплей, мікроконтролер та модулі бездротового зв'язку.

Автономне функціонування периферійних вузлів досягається завдяки інтегрований підсистемі живлення, яка складається з літій-іонного акумулятора, контролера заряду-розряду та підвищувального DC-DC перетворювача. Джерелом живлення слугує Li-ion акумулятор. Для безпечного функціонування, у пристрої задіяно модуль TP4056 із портом Type-C, він виконує роль інтелектуального контролера, оберігаючи акумуляторну батарею від перезаряду, надмірного розряду та струмових перевантажень. Для коректної роботи мікроконтролера та підключеної периферії потрібна стабільна напруга, у схему

включено модуль підвищувального перетворювача MT3608. Його завдання полягає у конвертації номінальної напруги акумулятора (3,7 В) у робочі 5 В.

Такий комбінований підхід гарантує оптимальний баланс, який дає надійність і безперервність роботи центрального вузла, високу мобільність та тривалу автономність віддалених вузлів.

2.2 Розробка апаратної частини системи моніторингу

Апаратна частина системи моніторингу складається з двох вузлів: центрального та периферійного, центральний вузол виконує функції збору, обробки та відображення даних, отриманих від периферійних вузлів, а також забезпечує передачу інформації до віддаленого сервера через Wi-Fi, периферійний вузол призначений для безпосереднього вимірювання показників навколишнього середовища та їх бездротової передачі до центрального вузла засобами LoRa. Обидва вузли побудовано на базі мікроконтролера STM32F411CEU6, що спрощує розробку програмного забезпечення та уніфікує апаратну базу системи. При проектуванні апаратної частини враховувались вимоги щодо енергоефективності, компактності та надійності передачі даних.

2.2.1 Апаратна частина центрального вузла

Після визначення необхідних апаратних компонентів було складено функціональну схему підключення модулів до інтерфейсів мікроконтролера (рис. 2.7). Інтерфейс I2C1 було виділено для підключення датчика вологості, тиску та температури BME280, інтерфейс SPI1 було розподілено між трьома модулями: дисплеєм ST7735S, модулем LoRa XL1278-SMT та SD-картою. Для зв'язку з Wi-Fi-модулем ESP-01S використовується інтерфейс USART6. Інтерфейс USART1, у свою чергу, може бути використано для підключення до персонального комп'ютера для дебагу.

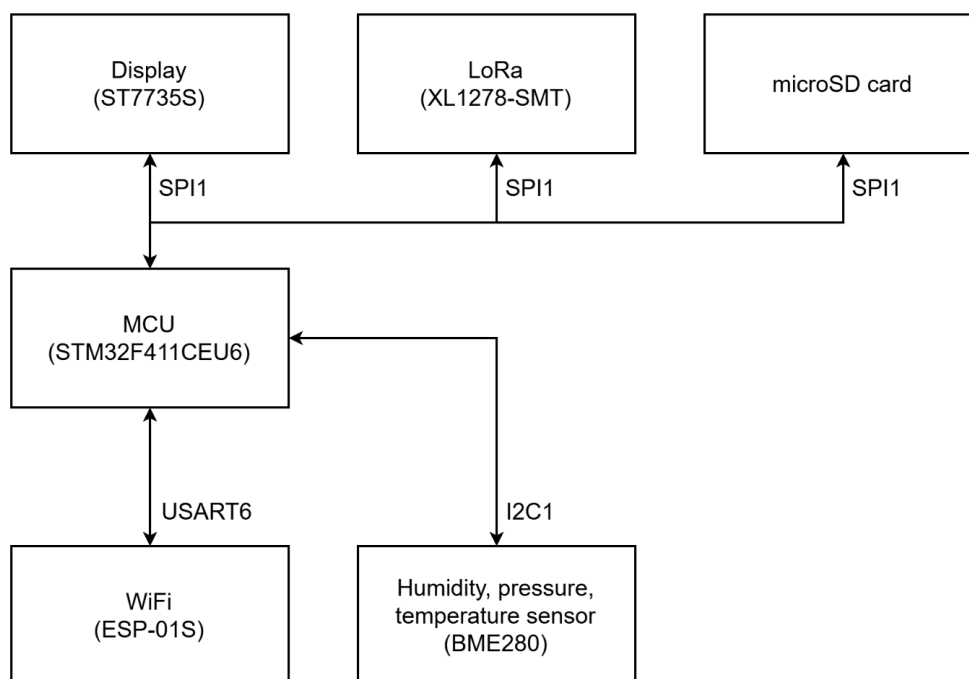


Рисунок 2.7 – Функціональна схема центрального вузла

Системи живлення центрального вузла (рис. 2.8) розрахована на роботу від стаціонарного зовнішнього джерела живлення. Напруга з зовнішнього джерела подається на модуль Breadboard Power Supply MB102, який забезпечує стабілізовану напругу для живлення мікроконтролера та всієї підключеної периферії.

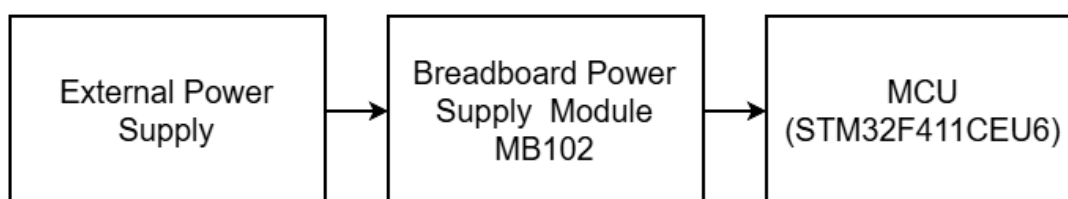


Рисунок 2.8 – Схема системи живлення центрального вузла

2.2.2 Апаратна частина периферійного вузла

Периферійний вузол побудовано на основі того ж мікроконтролера зі спрощеним набором периферії (рис. 2.9). Інтерфейс I2C1 використовується для підключення датчика BME280, який є основним вимірювальним елементом

вузла, інтерфейс SPI1 виділено для підключення модуля LoRa XL1278-SMT, який забезпечує передачу даних до центрального вузла.

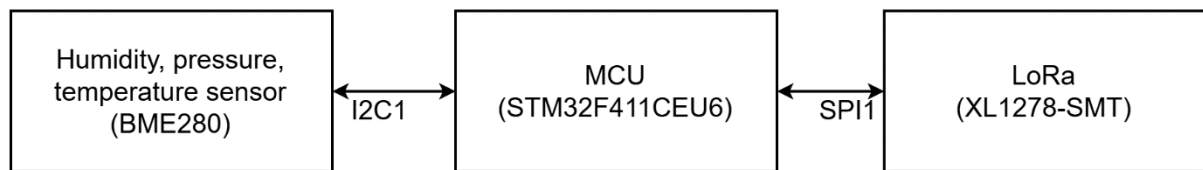


Рисунок 2.9 – Функціональна схема периферійного вузла

Схему живлення периферійного вузла (рис. 2.10) побудовано з урахуванням автономної роботи пристрою. Живлення подається від зовнішнього джерела через модуль заряджання TP4056, який керує зарядом літій-іонного акумулятора, для забезпечення стабільної напруги живлення мікроконтролера та периферії використовується підвищувальний перетворювач MT3608 Step-Up Converter, який підтримує необхідний рівень напруги навіть при розряджанні акумулятора.

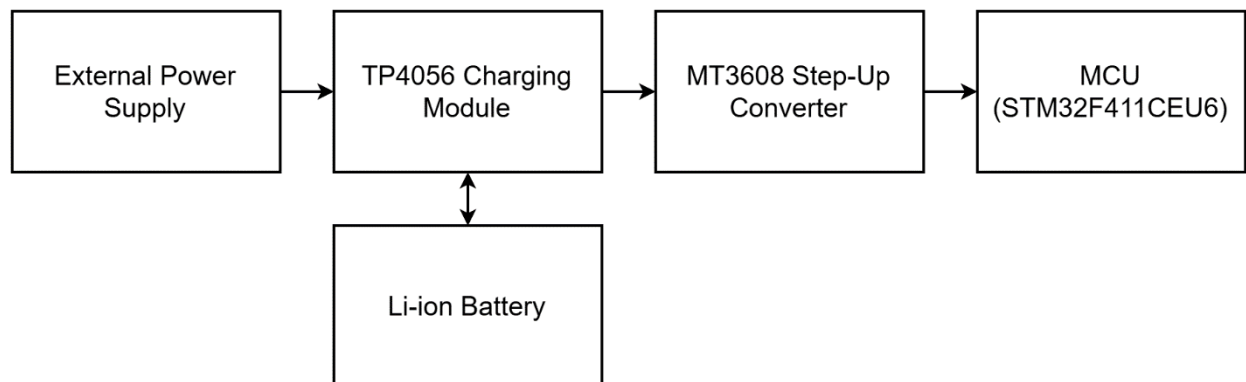


Рисунок 2.10 – Схема системи живлення периферійного вузла

3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ

3.1. Структурна схема програмних модулів системи моніторингу

У програмній частині розробленої системи реалізовано модульний підхід, що дозволяє ізолювати логіку взаємодії з різними апаратними периферійними вузлами, спростити відлагодження. Структурну схему програмних модулів та їх взаємодії показано на рис. 3.1.

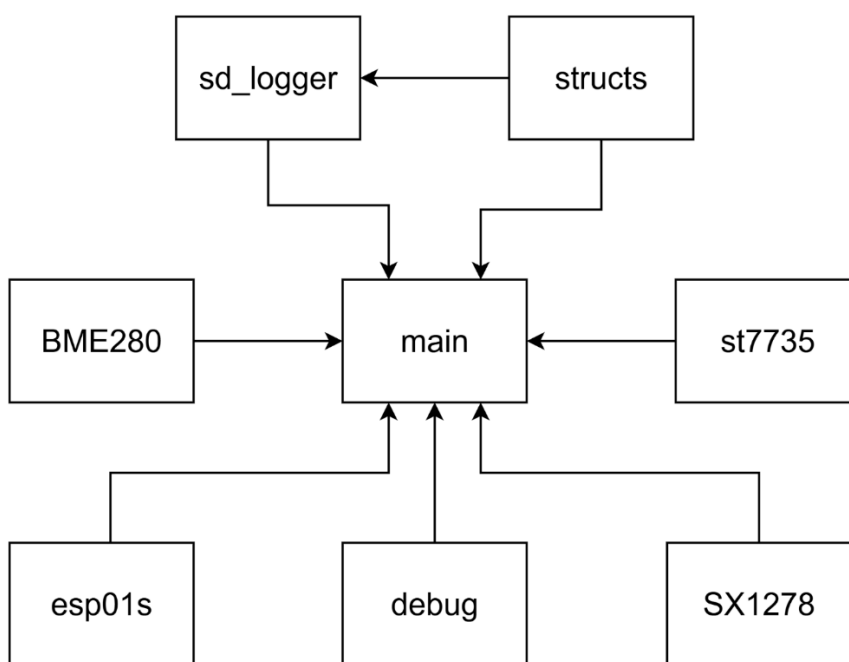


Рисунок 3.1 – Структурна схема програмних модулів

Центральним компонентом системи є модуль `main`, який виконує початкову ініціалізацію периферійних інтерфейсів мікроконтролера, координує роботу всіх підсистем та реалізує основний цикл виконання програми. Для уніфікації та стандартизації обміну інформацією використовується загальносистемний модуль `structs`, що містить визначення глобальних структур даних, типів, констант та пакетів телеметрії, які передаються до інших компонентів. Безпосередньо за роботу з картою пам'яті (SD-картою) через відповідну файлову систему відповідає модуль `sd_logger`, який виконує запис зібраних показників та системних логів. Збір кліматичних даних покладається на

модуль драйвера цифрового датчика BME280, який відповідає за конфігурування сенсора, зчитування сирих значень по інтерфейсу обміну та їх конвертацію у реальні показники температури, вологості й атмосферного тиску. Візуалізація інформації в реальному часі реалізована за допомогою модуля st7735 драйвера дисплея, який містить набір функцій для виведення текстової інформації чи графічних елементів інтерфейсу. Комунікаційні можливості пристрою забезпечуються двома модулями. Програмний модуль esp01s забезпечує встановлення бездротового з'єднання з Wi-Fi-точкою доступу та передає дані на сервер за протоколами TCP. Водночас модуль SX1278 бездротового зв'язку на базі трансивера керування LoRa-модемом для приймання та передачі інформаційних. Для моніторингу внутрішнього стану системи служить модуль відлагодження debug, призначена для формування та виведення діагностичних повідомлень, статусів виконання та помилок через інтерфейс UART у консоль розробника.

3.2 Модуль датчика показників навколишнього середовища

Програмний модуль драйвера BME280 відповідає за управління підключеним до системи моніторингу за допомогою інтерфейсу I2C датчиком температури, вологості та атмосферного тиску BME280 виробництва Bosch Sensortec. В його рамках реалізовано задачу ініціалізації, налаштування режимів роботи та циклічного зчитування вимірювальних даних з датчика. Після отримання сирих значень виконується їх компенсація відповідно до індивідуальних калібрувальних коефіцієнтів мікросхеми, після чого обчислені фізичні величини передаються для збереження та подальшої обробки. Зчитування даних відбувається за запитом системи у режимі одиночного вимірювання.

3.2.1 I2C

Інтерфейс I2C (Inter-Integrated Circuit) – це послідовна синхронна шина, яку створила компанія Philips у 1982 році. Її головне завдання забезпечити

комунікацію між мікроконтролерами та периферійними модулями на коротких дистанціях. Розробники прагнули максимально спростити з'єднання багатьох цифрових мікросхем, тому I2C використовує всього два дроти: лінію тактування (SCL) та лінію передачі даних (SDA). До однієї шини можна підключити велику кількість компонентів. Керування процесом беруть на себе один або кілька головних вузлів, а всі інші пристрої працюють у режимі підпорядкування. Обмін інформацією строго узгоджується за допомогою тактових імпульсів, які генерує головний пристрій.

3.2.2 Драйвер датчика BME280

В офіційній документації на мікросхему зазначено, що для зчитування показників датчика необхідно спочатку налаштувати реєстри керування: записати в реєстр 0xF2 (ctrl_hum) параметр вибірки вологості, в реєстр 0xF4 (ctrl_meas) – параметри вибірки температури, тиску та режим роботи, а в реєстр 0xF5 (config) — параметри фільтрації та часу очікування. Після цього зчитуються 8 байт починаючи з базової адреси 0xF7, серед яких містяться сирі показники тиску, температури та вологості. Повний список реєстрів можна побачити на рис. 3.2.

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
hum_lsb	0xFE	hum_lsb<7:0>								0x00
hum_msb	0xFD	hum_msb<7:0>								0x80
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>								0x00
press_msb	0xF7	press_msb<7:0>								0x80
config	0xF5	t_sb[2:0]			filter[2:0]			spi3w_en[0]		0x00
ctrl_meas	0xF4	osrs_t[2:0]			osrs_p[2:0]			mode[1:0]		0x00
status	0xF3				measuring[0]			im_update[0]		0x00
ctrl_hum	0xF2							osrs_h[2:0]		0x00
calib26..calib41	0xE1...0xF0	calibration data								individual
reset	0xE0	reset[7:0]								0x00
id	0xD0	chip_id[7:0]								0x60
calib00..calib25	0x88...0xA1	calibration data								individual

Registers:	Reserved registers	Calibration data	Control registers	Data registers	Status registers	Chip ID	Reset
Type:	do not change	read only	read / write	read only	read only	read only	write only

Рисунок 3.2 – Список реєстрів датчика

Перед використанням отримані дані потрібно конвертувати з урахуванням індивідуальних калібрувальних коефіцієнтів, що зчитуються з енергонезалежної пам'яті мікросхеми під час ініціалізації. Калібрувальні коефіцієнти для температури (dig_T1–dig_T3) та тиску (dig_P1–dig_P9) зберігаються в регістрах 0x88–0x9F, а коефіцієнти для вологості (dig_H1–dig_H6) — в регістрах 0xA1 та 0xE1–0xE7. Компенсація виконується у два етапи: спочатку розраховується температура разом із проміжним значенням t_{fine} , яке враховує теплові характеристики чипу та використовується для підвищення точності подальших обчислень тиску і вологості. Для реалізації компенсаційних розрахунків було використано код, наданий виробником в офіційній документації.

3.3 Модуль бездротового зв'язку

Модуль бездротового зв'язку забезпечує передачу зібраних сенсорних даних між вузлами системи моніторингу. В його рамках реалізовано підтримку двох каналів зв'язку: далекодуючого радіоканалу на основі технології LoRa для передачі вимірювальних даних між периферійними вузлами та центральним вузлом, а також Wi-Fi-з'єднання для подальшої передачі даних до локального сервера. Взаємодія мікроконтролера з відповідними модулями здійснюється через інтерфейси SPI та UART.

3.3.1 SPI

Інтерфейс SPI (Serial Peripheral Interface) використовується для високошвидкісного обміну даними між мікроконтролером та LoRa-модемом XL1278-SMT. SPI є синхронним послідовним інтерфейсом із чотирма лініями: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (тактовий сигнал) та CS (вибір пристрою). Передача даних відбувається у повнодуплексному режимі — одночасно в обох напрямках — за кожним тактовим імпульсом. Модем XL1278-SMT підтримує режим SPI Mode 0, в якому дані фіксуються по передньому фронту тактового сигналу. Доступ до внутрішніх регістрів мікросхеми здійснюється шляхом передачі адресного байта, де старший біт

визначає напрямок операції: 0 — читання, 1 — запис, після чого передається байт даних.

3.3.2 Драйвер LoRa-модему XL1278-SMT

Програмний драйвер модему XL1278-SMT реалізує повний цикл керування радіомодулем: ініціалізацію, налаштування робочих параметрів, передачу та прийом пакетів. Під час ініціалізації виконується апаратне скидання модуля через лінію RESET, після чого перевіряється ідентифікатор мікросхеми шляхом зчитування регістру 0x42 (RegVersion), значення якого для XL1278 має дорівнювати 0x12. Після підтвердження ідентифікатора модуль переводиться в режим сну (Sleep mode) через регістр 0x01 (RegOpMode) для безпечної зміни конфігурації.

Налаштування радіопараметрів включає запис робочої частоти у регістри 0x06–0x08 (RegFrMsb/Mid/Lsb), встановлення потужності передавача через регістр 0x09 (RegPaConfig), а також конфігурацію параметрів модуляції: смуги пропускання, коефіцієнта розширення спектру (Spreading Factor) та швидкості кодування через регістри 0x1D (RegModemConfig1) та 0x1E (RegModemConfig2). Після завершення налаштування модуль переводиться в режим очікування (Standby mode).

Передача пакету виконується шляхом запису корисного навантаження у FIFO-буфер модему через регістр 0x00 (RegFifo) з попереднім встановленням покажчика запису через регістр 0x0E (RegFifoTxBaseAddr). Після запису даних модуль переводиться в режим передачі (TX mode), а завершення передачі фіксується через прапорець TxDone у регістрі 0x12 (RegIrqFlags). Прийом пакету виконується аналогічно в режимі безперервного прийому (RXCONTINUOUS), а наявність прийнятого пакету визначається за прапорцем RxDone у тому самому регістрі переривань.

3.3.3 Протокол пакету та CRC перевірка

Для передачі вимірювальних даних між вузлами системи розроблено власний формат пакету фіксованої довжини 8 байт. Структура пакету наведена в таблиці 3.1. Значення температури та вологості передаються у вигляді цілих чисел, масштабованих з коефіцієнтом 100, що дозволяє зберегти два знаки після коми без використання числа з плаваючою крапкою. Так, наприклад, температура 23,45 °C передається як значення 2345. На стороні приймача виконується зворотнє перетворення діленням на 100.0. Температура представлена знаковим типом `int16_t` для коректної передачі від'ємних значень.

Таблиця 3.1 – Структура пакету передачі

Байт	Поле	Тип	Опис
0	ID	<code>uint8_t</code>	ідентифікатор вузла-відправника
1-2	Temperature	<code>int16_t</code>	температура $\times 100$, °C
3-4	Humidity	<code>uint16_t</code>	відносна вологість $\times 100$, %
5-6	Pressure	<code>uint16_t</code>	атмосферний тиск, гПа
7	Flags	<code>uint8_t</code>	службові прапорці стану вузла
8	CRC8	<code>uint8_t</code>	контрольна сума пакету

3.3.4 UART

Інтерфейс UART (Universal Asynchronous Receiver-Transmitter) використовується для обміну даними між мікроконтролером та Wi-Fi-модулем ESP-01S. На відміну від SPI, UART є асинхронним інтерфейсом — синхронізація між пристроями забезпечується узгодженою швидкістю передачі без окремої тактової лінії. Взаємодія з модулем ESP-01S здійснюється на швидкості 115200 бод з параметрами кадру 8N1 (8 біт даних, без біта парності, 1 стоп-біт). Для буферизації вхідного потоку даних від модуля на стороні мікроконтролера застосовується кільцевий буфер з обробкою в перериванні по прийому байта.

3.3.5 Драйвер Wi-Fi-модуля ESP-01S

Програмний драйвер модуля ESP-01S реалізує керування Wi-Fi-з'єднанням та передачу даних до віддаленого сервера засобами AT-команд, які модуль підтримує у стандартній прошивці Espressif. Ініціалізація починається з апаратного скидання модуля та надсилання команди AT, у відповідь на яку модуль повертає OK, що підтверджує його готовність до роботи. Далі командою AT+CWMODE=1 модуль переводиться в режим клієнта, після чого командою AT+CWLAP="SSID","password" виконується підключення до точки доступу.

Після встановлення Wi-Fi-з'єднання для передачі даних використовується TCP-з'єднання з віддаленим сервером, яке відкривається командою AT+CIPSTART="TCP","адреса",порт. Безпосередня відправка корисного навантаження виконується у два кроки: спочатку командою AT+CIPSEND=N вказується кількість байт для передачі, після чого модуль повертає символ > як запрошення, і лише тоді надсилаються самі дані. Підтвердженням успішної доставки слугує відповідь SEND OK. Після завершення сесії з'єднання закривається командою AT+CIPCLOSE. Кожна AT-команда супроводжується контролем таймауту очікування відповіді, а у разі відсутності реакції модуля протягом встановленого часу драйвер виконує повторну спробу або ініціює перезапуск модуля.

3.4 Модуль використання постійної пам'яті

Для забезпечення надійного локального зберігання даних у центральному вузлі системи моніторингу якості повітря реалізовано модуль постійної пам'яті на базі SD-карти. Необхідність локального логування обумовлена можливими перебоями у Wi-Fi-з'єднанні: навіть за відсутності зв'язку з сервером усі вимірювання зберігаються на карті та не втрачаються. Дані записуються у форматі CSV, що дозволяє їх безпосереднє відкриття в табличних редакторах без додаткової конвертації.

Модуль складається з двох рівнів: низькорівневого SPI-драйвера (user_diskio.c), що забезпечує фізичний обмін даними з картою, та

високорівневого логера (sd_logger.c), що формує рядки CSV і здійснює операції з файловою системою через бібліотеку FatFS.

3.4.1 Драйвер SD-карти

Взаємодія мікроконтролера STM32F411CEU6 з SD-картою реалізована через інтерфейс SPI з використанням бібліотеки FatFS, яка є реалізацією файлової системи FAT для вбудованих систем. FatFS виступає проміжним шаром між файловими операціями (відкриття файлу, запис рядка, закриття) та апаратним рівнем, який повністю визначається файлом user_diskio.c.

Файл user_diskio.c реалізує інтерфейс дискового введення-виведення FatFS і містить п'ять обов'язкових функцій: disk_initialize, disk_status, disk_read, disk_write та disk_ioctl. Кожна з них трансліує виклики FatFS у відповідні SPI-транзакції до карти.

Ініціалізація карти виконується у функції disk_initialize і включає такі кроки: подання не менше 74 тактових імпульсів при розібраному CS для переведення карти в SPI-режим, відправлення команди CMD0 (GO_IDLE_STATE) для скидання, команди CMD8 (SEND_IF_COND) для визначення версії карти (SDv1 або SDv2), та циклічне надсилання команди ACMD41 до отримання готовності. Для карт SDv2 додатково надсилається CMD58 для зчитування регістру OCR і визначення типу адресації – байтової (SDSC) або блокової (SDHC/SDXC).

Після успішної ініціалізації файлова система монтується викликом f_mount, результат якого перевіряється і за необхідності логується для діагностики. Таким чином, драйвер забезпечує надійний фізичний доступ до карти, на основі якого будується логіка збереження даних верхнього рівня.

3.4.2 Алгоритм збереження даних по вузлах

Логіку збереження даних реалізовано у файлі sd_logger.c. Основним публічним інтерфейсом модуля є функція SD_LogData, яка приймає ідентифікатор вузла, значення температури та вологості й записує один рядок у

CSV-файл на карті. Алгоритм роботи функції складається з послідовних кроків. Спочатку здійснюється відкриття файлу log.csv з прапором FA_OPEN_APPEND | FA_WRITE, що гарантує дописування нових записів у кінець без перезапису існуючих даних. Якщо файл ще не існує, FatFS автоматично його створює при першому відкритті з цим прапором.

Далі формується рядок CSV у буфері фіксованого розміру за допомогою snprintf у форматі:

```
<node_id>;<Temperature>;<Humidity>;<Pressure>\r\n
```

Такий формат узгоджений із протоколом передачі на сервер, де дані також передаються у вигляді id;temp;humidity;pressure, що спрощує подальшу обробку.

Сформований рядок записується функцією f_write, після чого обов'язково викликається f_sync для примусового скидання буферів FatFS на фізичний носій. Виклик f_sync є критично важливим: без нього дані можуть залишитись у внутрішніх буферах бібліотеки і втратитись у разі раптового відключення живлення або спрацювання IWDG. Завершальним кроком є закриття файлу через f_close.

Структура модуля передбачає, що функція SD_LogData викликається з основного циклу обробки подій — після успішного прийому пакету від кожного з віддалених вузлів по LoRa та перевірки його CRC8. Таким чином, для кожного прийнятого і цілісного пакету виконується один запис у лог-файл. Ідентифікатор вузла береться безпосередньо з поля ID пакету, що дозволяє в лог-файлі чітко розрізняти показники різних польових вузлів мережі.

Результуючий файл log.csv містить хронологічну послідовність вимірювань від усіх вузлів системи і може бути безпосередньо відкритий у Microsoft Excel або імпортований у будь-яку систему аналізу даних без проміжної обробки.

3.5 Модуль дисплею

Для відображення даних мікроклімату в реальному часі у центральному вузлі системи використовується TFT-дисплей ST7735 з роздільною здатністю 128×128 пікселів. Модуль дисплею реалізований у двох шарах: графічна

бібліотека примітивів (st7735.c) та логіка графічного інтерфейсу у main.c. Такий поділ забезпечує чітке розмежування між низькорівневою роботою та алгоритмом відображення даних.

3.5.1 Графічна бібліотека

Графічна бібліотека реалізована у файлі st7735.c і забезпечує повний набір примітивів для побудови інтерфейсу. В основі бібліотеки лежить адаптована під STM32 HAL версія бібліотеки Adafruit ST7735 для Arduino.

Базовою одиницею виводу є функція ST7735_DrawPixel, яка задає адресне вікно з одного пікселя командами CASET/RASET та передає двобайтовий колір у форматі RGB565 через SPI. Функція ST7735_SetAddressWindow є внутрішньою допоміжною процедурою, що встановлює прямокутну область відеопам'яті контролера перед будь-якою операцією масового запису пікселів. Формат кольору RGB565 кодує червоний канал у 5 бітах, зелений у 6, синій у 5, упаковуючи колір у 16-бітне слово:

```
#define ST7735_COLOR565(r, g, b)
(((r & 0xF8) << 8) | ((g & 0xFC) << 3) | ((b & 0xF8) >> 3))
```

Створення примітивів реалізовано на основі піксельних операцій: ST7735_DrawLine реалізує алгоритм Брезенхема для побудови довільної прямої, ST7735_DrawFastHLine та ST7735_DrawFastVLine – оптимізовані варіанти для горизонтальних і вертикальних ліній відповідно. ST7735_FillRectangle заповнює прямокутну область одним кольором, передаючи пікселі у циклі без повторного задання адресного вікна. Бібліотека також містить функції для кіл, еліпсів, трикутників та прямокутників із заокругленими кутами. Виведення тексту реалізоване через ST7735_WriteChar та ST7735_DrawString. Кожен символ задається адресним вікном шириною font.width і висотою font.height, після чого для кожного рядка зчитується бітова маска з масиву шрифту: якщо старший біт зсунутого слова встановлений — передається колір символу, інакше — колір фону:

```
b = font.data[(ch - 32) * font.height + i];
```

```
for(j = 0; j < font.width; j++) {
    if((b << j) & 0x8000) ST7735_WriteData(color_bytes, 2);
    else ST7735_WriteData(bgcolor_bytes, 2);
}
```

Функція `ST7735_DrawString` обходить рядок посимвольно, автоматично переносючи текст на новий рядок при досягненні правого краю екрана.

3.5.2 Драйвер дисплею

Низькорівнева взаємодія з контролером ST7735 будується на двох базових операціях: `ST7735_WriteCommand` для надсилання командного байту та `ST7735_WriteData` для надсилання байтів даних. Передача виконується через `HAL_SPI_Transmit` на SPI1.

Ініціалізація дисплею виконується у функції `ST7735_Init` і складається з апаратного скидання та послідовного виконання трьох командних таблиць:

```
void ST7735_Init() {
    ST7735_GPIO_Init();
    TFT_CS_L();
    ST7735_Reset();
    ST7735_ExecuteCommandList(init_cmds1);
    ST7735_ExecuteCommandList(init_cmds2);
    ST7735_ExecuteCommandList(init_cmds3);
    TFT_CS_H();
}
```

Апаратне скидання (`ST7735_Reset`) подає імпульс низького рівня на пін RES тривалістю 20 мс. Функція `ST7735_ExecuteCommandList` інтерпретує таблицю у форматі: перший байт — кількість команд, далі для кожної команди: код команди, кількість, самі аргументи та опціонально значення затримки в мілісекундах. Якщо значення затримки дорівнює 255, вона інтерпретується як 500 мс. Такий формат дозволяє компактно зберігати складну послідовність ініціалізації та легко її модифікувати.

У проєкті дисплей підключений до SPI1 разом з модулем LoRa SX1278, проте конфлікту шини не виникає завдяки роздільному керуванню CS-пінами: перед кожною операцією до дисплею активується `TFT_CS_L()`, а після

завершення — TFT_CS_H(). CS-пін модуля LoRa при цьому залишається у неактивному стані.

3.5.3 Графічний інтерфейс

Графічний інтерфейс реалізований безпосередньо у main.c і складається з трьох функцій: draw_intro, draw_interface та draw_measurements. При старті системи викликається draw_intro, яка відображає заставку. Після завершення заставки, яка виводиться одноразово на початку головного циклу, викликається draw_interface, що малює статичну рамку інтерфейсу:

```
void draw_interface() {  
    ST7735_DrawRect(0, 16, 128, 112, ST7735_WHITE);  
    ST7735_DrawFastHLine(0, 80, 128, ST7735_WHITE);  
    ST7735_DrawFastVLine(64, 80, 48, ST7735_WHITE);  
    ST7735_DrawString(98, 46, "^C", Font_11x18, ...);  
    ST7735_DrawString(9, 110, "HUMIDITY", Font_5x7, ...);  
    ST7735_DrawString(88, 110, "hPa", Font_5x7, ...);  
}
```

Цей код розміщує на екрані три зони: велике поле температури у верхній частині, поле вологості ліворуч знизу та поле тиску праворуч знизу. Підписи одиниць (^C, HUMIDITY, hPa) є статичними і малюються лише один раз.

Динамічне оновлення даних виконує функція draw_measurements, яка приймає вказівник на структуру NodeData з полями id, temp, humidity та pressure. Ключовою особливістю реалізації є механізм часткового оновлення: кожне значення порівнюється з попереднім через статичні змінні-кеші (prevId, prevTemp10, prevHumInt, prevPressureHpa), і перемальовується лише те поле, яке змінилося. Це дозволяє уникнути мерехтіння та зайвих операцій:

```
int16_t temp10 = (int16_t)(d->temp * 10 + 0.5f);  
static int prevTemp10 = -999;  
if(temp10 != prevTemp10) {  
    prevTemp10 = temp10;  
    sprintf(str, "%c%d.%d", sign, value/10, value%10);  
    ST7735_DrawString(14, 46, str, Font_16x26, ST7735_WHITE, ST7735_BLACK);  
}
```

Перемикання між вузлами реалізовано через апаратне переривання від кнопки: у обробнику HAL_GPIO_EXTI_Callback шукається наступний вузол із встановленим прапором node_valid[i], після чого встановлюється node_switch_flag = 1 та оновлюється selected_node. Головний цикл перевіряє цей прапор і викликає draw_measurements(&node_cache[selected_node]), що оновлює всі поля під новий вузол. Вузол з ID=0 (HUB) оновлюється додатково кожні 5 секунд за таймером через LOG_INTERVAL, оскільки його дані надходять від локального BME280, а не по LoRa.

Таким чином, графічний інтерфейс забезпечує наочне відображення даних якості повітря від усіх вузлів з мінімальними витратами на перемальовування, що є критично важливим при роботі з повільним побайтовим SPI-виводом тексту.

3.6 Програмна реалізація центрального вузла

Програмне забезпечення центрального вузла базується на циклі з використанням переривань для обробки критичних до часу подій. Головними завданнями мікроконтролера є збір даних з локальних датчиків, прийом пакетів від віддалених вузлів через бездротовий канал зв'язку LoRa, відображення інформації на дисплеї, а також збереження зібраних даних на SD-карту та їх передача на сервер через модуль ESP-01S.

3.6.1 Алгоритм ініціалізації

Процес ініціалізації починається з базового налаштування системи тактування мікроконтролера та підготовки периферійних модулів: портів вводу/виводу, контролера прямого доступу до пам'яті, незалежного сторожового таймера (IWDG) для захисту від зависань та годинника реального часу. Потім ініціалізуються модулі верхнього рівня: модуль SD-карти, дисплей ST7735 із виведенням графічної заставки. Радіомодуль SX1278 конфігурується на частоту 434 МГц з вихідною потужністю 17 дБм, фактором розширення SF7, смугою пропускання 125 кГц та увімкненим апаратним контролем цілісності пакетів.

Після налаштування модуль переводиться в режим очікування прийому пакетів. Датчик BME280 конфігурується для нормального режиму роботи з визначеними параметрами оверсемплінгу та IIR-фільтрації.

3.6.2 Алгоритм кешування даних та керування опитуванням вузлів

Центральний вузол створює масив `node_cache[MAX_NODES]` структур типу `NodeData`, що містять ідентифікатор, температуру, вологість та тиск. Кожен запис супроводжується двома прапорами: `node_valid[i]` вказує на те, що дані від вузла хоча б раз були отримані та придатні для відображення, `node_unsaved[i]` сигналізує про наявність нових даних, які ще не були збережені на SD-карту та не були надіслані через ESP. Робота алгоритму кешування розпочинається з періодичного збору локальних даних. Кожні десять секунд мікроконтролер ініціює опитування вбудованого датчика BME280. Отримані параметри температури, вологості та атмосферного тиску записуються в нульовий індекс масиву, який зарезервований за центральним вузлом, з одночасним встановленням прапорця очікування запису.

В головному циклі програма здійснює постійний моніторинг нових або незбережених даних. При виявленні такого прапорця інформація відправляється до Wi-Fi-модуля через контролер прямого доступу до пам'яті для забезпечення передачі на зовнішній сервер без блокування процесора. Відразу після цього ці дані копіюються у проміжний буфер підсистеми логування SD-карти, а відповідний прапорець незбережених даних скидається. Для мінімізації кількості циклів запису та відкриття файлу фізичний запис інформації з буфера у файл відбувається з періодичністю в одну хвилину або у випадку його повного заповнення. Прийом пакетів від периферійних вузлів реалізовано на базі апаратних переривань. Успішне надходження радіопакета викликає сигнал на піні DIO0 LoRa-модему, що встановлює глобальний прапорець та фіксує точну часову мітку події. Основний цикл реагує на цю подію, зчитує вміст буфера SX1278 та виконує перевірку цілісності пакету за допомогою обчислення

контрольної суми CRC8. У разі успішної верифікації пакет зберігається у комірці кешу, номер якої відповідає ідентифікатору вузла.

Для гарантування уникнення колізій передача даних від різних вузлів розділена у часі. Базовий цикл опитування мережі встановлено на рівні 60 секунд, в межах якого кожному вузлу виділяється власний часовий проміжок. Зсув визначається індивідуально і становить 500 мілісекунд, помножених на ідентифікатор пристрою. Після обробки вхідного пакета мікроконтролер вираховує часовий зсув та розраховує точний час, що залишився до наступного вікна передачі цього вузла. Отримане та скориговане значення інтервалу очікування упаковується у 6-байтовий пакет-підтвердження (АСК), який одразу відправляється назад до вузла. Приймавши пакет, вузол переходить у режим глибокого сну на вказаний час, завдяки чому забезпечується постійна та точна синхронізація всіх вузлів відносно центрального хабу. Схему алгоритму головного циклу зображено на рис 3.3.

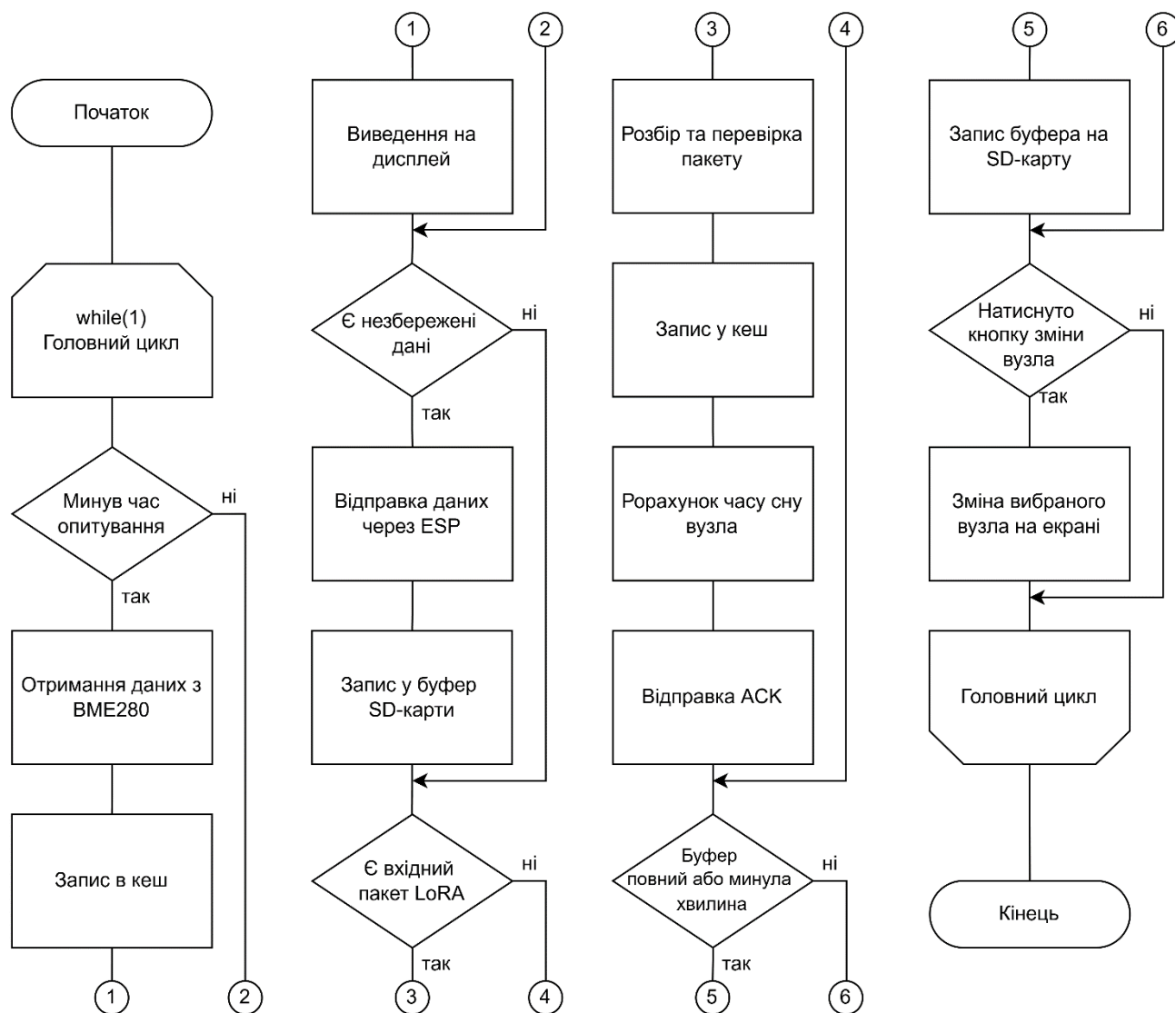


Рисунок 3.3 – Схема алгоритму головного циклу

3.7 Програмна реалізація периферійного вузла

Програмне забезпечення периферійного вузла побудована за класичною схемою циклу з використанням апаратних переривань для обробки критичних до часу подій від радіомодуля, основною метою програмної реалізації є забезпечення максимальної автономності пристрою, мінімізація часу активного стану процесора та дотримання часових інтервалів для запобігання колізіям з іншими вузлами.

3.7.1 Алгоритм збору та формування пакету даних

Робота периферійного вузла починається з ініціалізації периферії мікроконтролера та налаштування базових параметрів радіомодуля LoRa. Для запобігання одночасному виходу в ефір усіх пристроїв мережі у випадку одночасного подання живлення, в алгоритмі передбачено початкову затримку, її тривалість розраховується індивідуально для кожного вузла шляхом множення його унікального ідентифікатора на 2 секунди. В основному циклі мікроконтролер виводить радіомодуль з режиму сну та зчитує показники з датчика BME280, алгоритм формування пакету збирає дані у масив розміром 9 байтів. Перший байт містить ідентифікатор вузла. Значення температури, вологості та атмосферного тиску конвертуються з формату з плаваючою комою у цілочисельний формат шляхом множення на 100 для збереження двох знаків після коми та розбиваються на два байти. Передостанній байт зарезервовано для прапорців або налагоджувальної інформації. Останній байт містить контрольну суму, обчислену для перших восьми байтів, що гарантує цілісність даних при їх передачі. Після формування пакету радіомодуль переводиться в режим передачі, по завершенню відправки вузол перемикається в режим прийому і очікує на пакет-підтвердження від центрального вузла. У разі отримання пакету розміром 6 байтів, алгоритм перевіряє його контрольну суму та збіг ідентифікатора. Якщо все правильно, вузол бере з пакету 32-бітне значення часу в мілісекундах, через який він має здійснити наступний сеанс зв'язку, після чого вузол сигналізує про успішне отримання коротким спалахом світлодіода.

3.7.2 Режими енергозбереження

Оскільки периферійні вузли орієнтовані на автономне живлення, оптимізація енергоспоживання є критичною вимогою. Після завершення обміну даними радіомодуль SX1278 примусово переводиться в режим мінімального споживання (SX1278_sleep), після чого мікроконтролер переходить у режим глибокого сну. Для забезпечення високоточного відліку часу під час сну використовується модуль годинника реального часу, тактований від зовнішнього

низькочастотного кварцового резонатора (LSE, 32768 Гц). Для досягнення високої точності таймер пробудження налаштований з дільником частоти DIV16, що забезпечує крок розрахунку приблизно 0.488 мс.

Процедура засинання реалізована у функції Enter_Stop_Precise:

```
void Enter_Stop_Precise(uint32_t sleep_ms)
{
    HAL_RTCEx_DeactivateWakeUpTimer(&hrtc);
    HAL_RTC_WaitForSynchro(&hrtc);
    const uint32_t MAX_STEP_MS = 30000;

    while (sleep_ms > 0) {
        uint32_t current_sleep = (sleep_ms > MAX_STEP_MS) ?
            MAX_STEP_MS : sleep_ms;
        //... розрахунок ticks
        HAL_RTCEx_SetWakeUpTimer_IT(
            &hrtc, (uint16_t)ticks,
            RTC_WAKEUPCLOCK_RTCCLK_DIV16);
        HAL_SuspendTick();
        __HAL_PWR_CLEAR_FLAG(PWR_FLAG_WU);
        HAL_PWR_EnterSTOPMode(
            PWR_LOWPOWERREGULATOR_ON,
            PWR_STOPENTRY_WFI);

        SystemClock_Config();
        HAL_ResumeTick();

        sleep_ms -= current_sleep;
    }

    MX_SPI1_Init();
    MX_I2C1_Init();
}
```

Оскільки апаратний лічильник таймера має обмежену розрядність, максимальний безпечний інтервал сну за один цикл обмежено 30 секундами, щоб обійти це обмеження алгоритм автоматично розбиває загальний час очікування, отриманий від центрального вузла, на кілька ітерацій, якщо він перевищує цей ліміт. Для компенсації часу, що витрачається мікроконтролером на пробудження, вимірювання та повторне засинання між ітераціями, алгоритм віднімає константу OVERHEAD_BETWEEN_STOPS. Після пробудження мікроконтролер відновлює налаштування системи тактування та ініціалізує інтерфейси SPI та I2C для наступного циклу вимірювання та передачі даних.

3.8 Серверна частина системи моніторингу

Серверна частина системи моніторингу реалізована мовою Python і складається з двох основних підсистем, що функціонують паралельно: TCP-сервера для безперервного прийому даних від кінцевих вузлів та HTTP-сервера на базі фреймворку Flask. Flask-сервер надає доступ до поточних і статистичних показників через REST API. Для забезпечення одночасної та незалежної роботи обох підсистем використовується модуль `threading`, який дозволяє виділити операції обробки мережесокетів в окремі фонові потоки.

3.8.1 TCP-сервер для прийому даних

Мережевий TCP-сервер ініціалізується за допомогою стандартної бібліотеки `socket` з використанням сімейства адрес IPv4 та потокового протоколу TCP. Для запобігання блокуванню мережевого порту операційною системою при перезапуску програми встановлюється опція сокета `SO_REUSEADDR`. Після прив'язки до визначеної IP-адреси та порту сервер переводиться в режим прослуховування. Архітектура прийому даних побудована на багатопотоковій моделі: основний цикл сервера безперервно очікує на вхідні підключення методом `асерт()`. Для підвищення стабільності роботи встановлюється таймаут на читання тривалістю 30 секунд, що запобігає зависанню потоків у разі раптового обриву зв'язку або апаратного збою на стороні передавача.

3.8.2 Алгоритм обробки та запису даних до БД

Отриманий мережевий пакет у вигляді масиву байтів декодується за стандартом UTF-8 з ігноруванням помилкових. Алгоритм очікує вхідні дані у вигляді текстового рядка, розділеного крапкою з комою за форматом: ідентифікатор вузла, температура, вологість та атмосферний тиск. Алгоритм перевіряє цілісність пакета на наявність щонайменше трьох базових складових та конвертує текстові значень у числовий формат з плаваючою комою. У разі виявлення пошкоджених або неповних даних, що може статися при фрагментації або злипанні TCP-пакетів, такі дані відхиляються, запобігаючи зупинці коду.

Збереження валідованих даних реалізовано паралельно на двох взаємодоповнюючих рівнях. Для забезпечення миттєвого відображення показників на сайті без затримок на файлові операції, останні значення разом із поточною міткою часу записуються до глобального масиву `nodes` в оперативній пам'яті сервера. Одночасно з цим, для збереження статистики вимірювань, дані записуються до реляційної бази даних SQLite. З метою уникнення конфліктів блокування файлу під час одночасного доступу з різних потоків, база даних функціонує з увімкненим режимом журналу випереджального запису (PRAGMA journal_mode=WAL). Алгоритм формує SQL-запит вставки ідентифікатора вузла та показників до таблиці `readings` та закриває з'єднання з файлом, вивільняючи системні ресурси.

3.9 Програмна реалізація вебінтерфейсу

Клієнтська частина системи моніторингу реалізована у вигляді односторінкового вебсайту, який забезпечує взаємодію користувача з даними в режимі реального часу. Інтерфейс побудовано з використанням таких вебтехнологій: HTML5, CSS3 та чистого JavaScript для програмної частини, обмін даними із серверною частиною здійснюється через REST API за допомогою HTTP-запитів, що дозволяє динамічно оновлювати інформацію на екрані без перезавантаження вебсторінки.

3.9.1 Структура вебсайту

Структура користувацького інтерфейсу базується на гнучкій сітці, яка розділяє робочий простір на дві основні функціональні зони: бічну панель навігації та головну панель індикації. Візуальний дизайн адаптовано під темну тему оформлення. Бічна панель слугує для відображення списку всіх активних вузлів, кожен елемент списку відображає унікальний ідентифікатор вузла та останні отримані показники. Програмна логіка дозволяє користувачеві обирати необхідний вузол з бічної панелі для відображення на головній панелі. Головна панель робочого простору містить заголовок з ідентифікатором поточного вузла,

блок останніх отриманих даних та контейнери для відображення графіків зміни показників. Фонове оновлення цих даних відбувається циклічно кожну секунду, забезпечуючи швидку реакцію системи на зміни в навколишньому середовищі. Зовнішній вигляд вебсайту показано на рис 3.4.

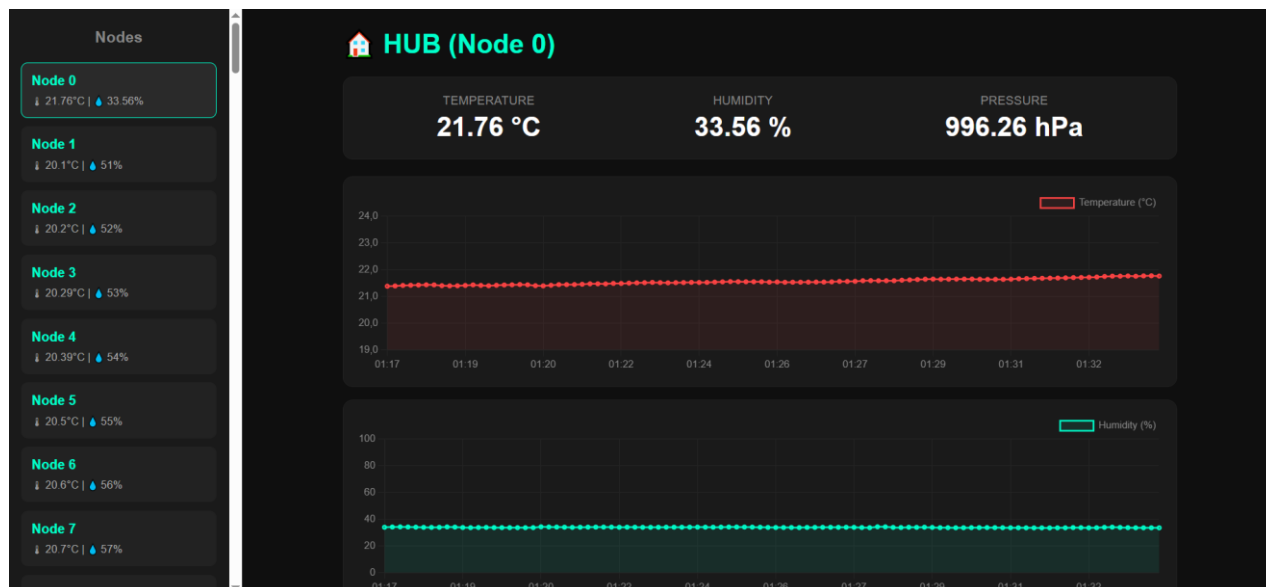


Рисунок 3.4 – Зовнішній вигляд вебсайту

3.9.2 Реалізація візуалізації даних та статистики

Для графічного зображення статистики вимірювань використовується зовнішня JavaScript-бібліотека Chart.js. Візуалізація даних відбувається автоматично – система періодично звертається до сервера, завантажує останні збережені в базі даних записи для обраного вузла і формує на їх основі графіки. Це дозволяє користувачеві відстежувати зміни температури, вологості та атмосферного тиску.

З метою оптимізації та уникнення витоків пам'яті, у коді використовується функція buildChart. Її алгоритм перевіряє наявність створеного екземпляра графіка: якщо графік вже існує, функція лише викликає метод перемальовування update(), замість руйнування та створення нового об'єкта Chart. Кожен з трьох графіків має власні налаштування осей. Для відносної вологості задано діапазон від 0 до 100% із кроком 10 одиниць, для температури та атмосферного тиску

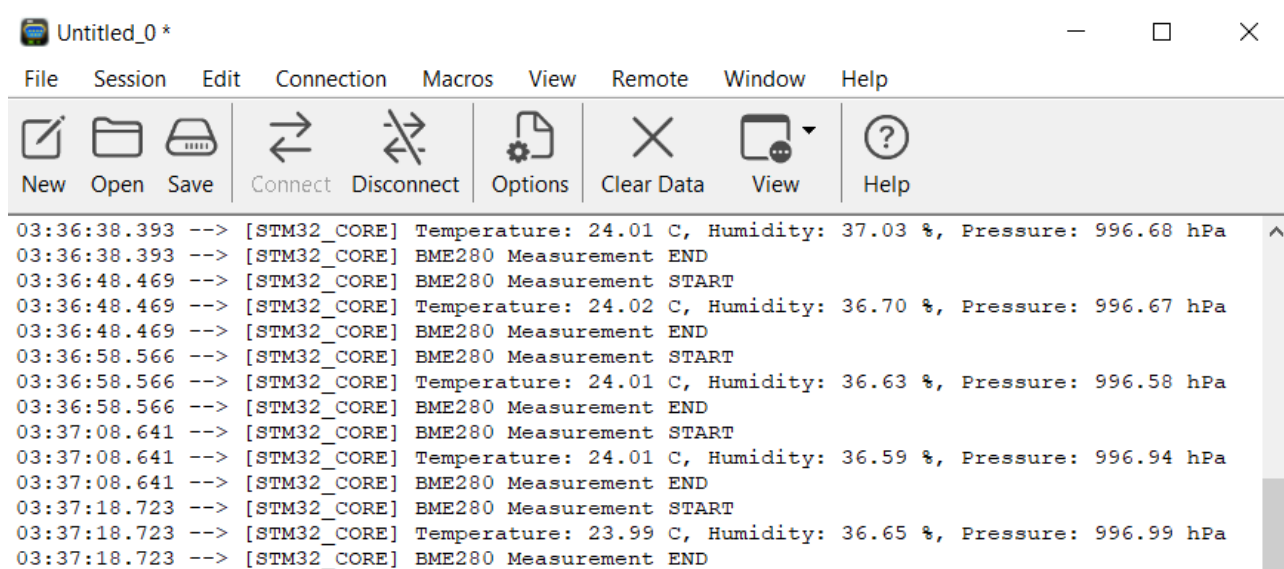
застосовується лінійна шкала з параметром $gr_{\text{асе}}$, який масштабує і додає відступи від максимальних та мінімальних значень.

					ІАЛЦ.468300.004 ПЗ	Арк.
						41
Зм	Лист	№ докум.	Підп.	Дата		

4. ТЕСТУВАННЯ

4.1 Тестування зчитування показників датчика

Після написання коду для опитування датчика, виникла необхідність протестувати коректність його ініціалізації та зчитування даних. Основним інструментом діагностики на цьому етапі став розроблений модуль логування, який передає форматовані повідомлення через інтерфейс UART до персонального комп'ютера, такий підхід дозволяє в режимі реального часу відслідковувати успішність передачі даних, періодичність вимірювань, а також перевіряти правильність математичних перетворень у фізичні величини. Приклад повідомлень в терміналі від мікроконтролера наведено на рис. 4.1.



```
03:36:38.393 --> [STM32_CORE] Temperature: 24.01 C, Humidity: 37.03 %, Pressure: 996.68 hPa
03:36:38.393 --> [STM32_CORE] BME280 Measurement END
03:36:48.469 --> [STM32_CORE] BME280 Measurement START
03:36:48.469 --> [STM32_CORE] Temperature: 24.02 C, Humidity: 36.70 %, Pressure: 996.67 hPa
03:36:48.469 --> [STM32_CORE] BME280 Measurement END
03:36:58.566 --> [STM32_CORE] BME280 Measurement START
03:36:58.566 --> [STM32_CORE] Temperature: 24.01 C, Humidity: 36.63 %, Pressure: 996.58 hPa
03:36:58.566 --> [STM32_CORE] BME280 Measurement END
03:37:08.641 --> [STM32_CORE] BME280 Measurement START
03:37:08.641 --> [STM32_CORE] Temperature: 24.01 C, Humidity: 36.59 %, Pressure: 996.94 hPa
03:37:08.641 --> [STM32_CORE] BME280 Measurement END
03:37:18.723 --> [STM32_CORE] BME280 Measurement START
03:37:18.723 --> [STM32_CORE] Temperature: 23.99 C, Humidity: 36.65 %, Pressure: 996.99 hPa
03:37:18.723 --> [STM32_CORE] BME280 Measurement END
```

Рисунок 4.1 – Термінал з повідомленнями

На зображенні можна побачити, що система стабільно виконує вимірювання з інтервалом в 10 секунд, кожен цикл супроводжується повідомленнями про початок та завершення процесу зчитування, що підтверджує стабільність роботи мікроконтролера та відсутність блокувань під час обміну даними з датчиком. Зчитані показники коректно форматуються та виводяться у стандартизованих одиницях вимірювання.

Під час роботи можна зробити висновок, що апаратно-програмна частина взаємодії з датчиком працює надійно, без втрати зв'язку чи критичних збоїв. Отримані значення температури, вологості та тиску є адекватними для умов закритого житлового приміщення, а порівняння отриманих даних з показниками побутового термометра та даними місцевих метеостанцій підтвердило, що відхилення знаходяться в межах заводської похибки датчика BME280. Результати тестування підтверджують, що модуль функціонує коректно і повністю готовий до подальшої інтеграції з системою вимірювань.

4.2 Тестування збору даних з периферійних вузлів

Наступним етапом тестування стала перевірка роботи периферійних вузлів, тестування було розділено на два етапи: перевірку точності часової синхронізації вузлів та тестування системи при максимальному навантаженні з імітацією підключення сотні вузлів. Оскільки для уникнення колізій передавачів LoRa застосовується алгоритм часового розділення, який потребує від кожного вузла високої точності синхронізації та передачі даних у свій часовий проміжок. Через наявну похибку тактових генераторів поступово виникає невелике зміщення часу, що без належного коригування може призвести до колізій у радіоефірі. Для тестування ефективності алгоритму компенсації цього зміщення було проведене тестування з виведенням повідомлень про розрахований час прийому пакета та його фактичний час надходження. Як видно з рисунка 4.2, зафіксоване зміщення часу є мінімальним і стабільно утримується в межах допустимої норми для обраної тривалості часового проміжку. Алгоритм синхронізації від центрального вузла успішно та вчасно коригує внутрішні лічильники периферійних вузлів. Це забезпечує стабільну передачу даних у виділених інтервалах і гарантує відсутність колізій навіть під час тривалої роботи.

```

21:26:24.546 --> [STM32_CORE] LoRa RX START
21:26:24.546 --> [STM32_CORE] LoRa read success
21:26:24.546 --> [STM32_CORE] Node 1 drift: -28 ms  sleep: 60415 ms
21:26:24.657 --> [STM32_CORE] LoRa RX END
21:26:24.729 --> [ESP01] DMA SEND SUCCESS
21:26:26.816 --> Flushing 7 entries...
21:26:27.176 --> Flush done, errors: 0
21:26:27.846 --> [STM32_CORE] BME280 Measurement START
21:26:27.846 --> [STM32_CORE] BME280 Measurement END
21:26:27.920 --> [ESP01] DMA SEND SUCCESS
21:26:37.941 --> [STM32_CORE] BME280 Measurement START
21:26:37.941 --> [STM32_CORE] BME280 Measurement END
21:26:38.002 --> [ESP01] DMA SEND SUCCESS
21:26:48.050 --> [STM32_CORE] BME280 Measurement START
21:26:48.050 --> [STM32_CORE] BME280 Measurement END
21:26:48.124 --> [ESP01] DMA SEND SUCCESS
21:26:58.141 --> [STM32_CORE] BME280 Measurement START
21:26:58.141 --> [STM32_CORE] BME280 Measurement END
21:26:58.221 --> [ESP01] DMA SEND SUCCESS
21:27:08.251 --> [STM32_CORE] BME280 Measurement START
21:27:08.251 --> [STM32_CORE] BME280 Measurement END
21:27:08.309 --> [ESP01] DMA SEND SUCCESS
21:27:18.347 --> [STM32_CORE] BME280 Measurement START
21:27:18.347 --> [STM32_CORE] BME280 Measurement END
21:27:18.455 --> [ESP01] DMA SEND SUCCESS
21:27:25.163 --> [STM32_CORE] LoRa RX START
21:27:25.163 --> [STM32_CORE] LoRa read success
21:27:25.163 --> [STM32_CORE] Node 1 drift: -17 ms  sleep: 60412 ms
21:27:25.269 --> [STM32_CORE] LoRa RX END
21:27:25.331 --> [ESP01] DMA SEND SUCCESS
21:27:27.785 --> Flushing 7 entries...

```

Рисунок 4.2 – Термінал з діагностикою синхронізації та відображенням часового зміщення

Друга частина тестування була спрямована на перевірку стабільності роботи центрального вузла під час опитування максимально передбаченої кількості периферійних вузлів. Збільшення кількості пристроїв суттєво підвищує навантаження на мікроконтролер. Система повинна встигати не лише безперервно обробляти переривання від радіомодуля LoRa для прийому вхідних пакетів, але й паралельно керувати іншою периферією. Результати, наведені на рисунку 4.3, підтверджують, що апаратно-програмний комплекс успішно справляється з піковим навантаженням, незважаючи на високу інтенсивність надходження радіопакетів, робота іншої периферії залишається абсолютно стабільною.

```

22:47:40.020 --> [STM32 CORE] LoRa RX END
22:47:40.067 --> [ESP01] DMA SEND SUCCESS
22:47:40.614 --> [STM32 CORE] LoRa RX START
22:47:40.614 --> [STM32_CORE] LoRa read success
22:47:40.614 --> [STM32_CORE] Node 92 drift: 9989 ms sleep: 50011 ms
22:47:40.724 --> [STM32 CORE] LoRa RX END
22:47:40.778 --> [ESP01] DMA SEND SUCCESS
22:47:41.315 --> [STM32 CORE] LoRa RX START
22:47:41.315 --> [STM32_CORE] LoRa read success
22:47:41.315 --> [STM32_CORE] Node 93 drift: 10197 ms sleep: 49803 ms
22:47:41.441 --> [STM32 CORE] LoRa RX END
22:47:41.504 --> [ESP01] DMA SEND SUCCESS
22:47:42.036 --> [STM32 CORE] LoRa RX START
22:47:42.036 --> [STM32_CORE] LoRa read success
22:47:42.036 --> [STM32_CORE] Node 94 drift: 10405 ms sleep: 49595 ms
22:47:42.163 --> [STM32 CORE] LoRa RX END
22:47:42.200 --> [ESP01] DMA SEND SUCCESS
22:47:42.757 --> [STM32 CORE] LoRa RX START
22:47:42.757 --> [STM32_CORE] LoRa read success
22:47:42.757 --> [STM32_CORE] Node 95 drift: 10613 ms sleep: 49387 ms
22:47:42.881 --> [STM32 CORE] LoRa RX END
22:47:42.933 --> [ESP01] DMA SEND SUCCESS
22:47:43.471 --> [STM32 CORE] LoRa RX START
22:47:43.471 --> [STM32_CORE] LoRa read success
22:47:43.471 --> [STM32_CORE] Node 96 drift: 10821 ms sleep: 49179 ms
22:47:43.585 --> [STM32 CORE] LoRa RX END
22:47:43.663 --> [ESP01] DMA SEND SUCCESS
22:47:44.176 --> [STM32 CORE] LoRa RX START
22:47:44.176 --> [STM32_CORE] LoRa read success
22:47:44.176 --> [STM32_CORE] Node 97 drift: 11028 ms sleep: 48972 ms
22:47:44.286 --> [STM32 CORE] LoRa RX END
22:47:44.365 --> [ESP01] DMA SEND SUCCESS
22:47:44.892 --> [STM32 CORE] LoRa RX START
22:47:44.892 --> [STM32_CORE] LoRa read success
22:47:44.892 --> [STM32_CORE] Node 98 drift: 11237 ms sleep: 48763 ms
22:47:45.020 --> [STM32 CORE] LoRa RX END
22:47:45.060 --> [ESP01] DMA SEND SUCCESS
22:47:45.597 --> [STM32 CORE] LoRa RX START
22:47:45.597 --> [STM32_CORE] LoRa read success
22:47:45.597 --> [STM32_CORE] Node 99 drift: 11444 ms
22:47:45.606 --> sleep: 48556 ms
22:47:45.748 --> [STM32 CORE] LoRa RX END
22:47:45.780 --> [ESP01] DMA SEND SUCCESS
22:47:46.334 --> [STM32 CORE] LoRa RX START
22:47:46.334 --> [STM32_CORE] LoRa read success
22:47:46.334 --> [STM32_CORE] Node 100 drift: 11652 ms sleep: 48348 ms
22:47:46.442 --> [STM32 CORE] LoRa RX END
22:47:46.504 --> [ESP01] DMA SEND SUCCESS
22:47:49.018 --> Flushing 88 entries...
22:47:50.168 --> Flush done, errors: 0

```

Рисунок 4.3 – Логування процесу збору даних з 100 підключених вузлів

Зокрема, можна побачити, що обробка зібраних кліматичних даних та їх подальший запис на SD-карту за допомогою файлової системи FatFs відбуваються без затримки та втрати пакетів, файлові операції не блокують роботу, а передача показників на сервер за допомогою Wi-Fi-модуля проходить успішно. Це показує, що ресурси мікроконтролера розподілені оптимально, і

система здатна надійно виконувати повноцінний збір та обробку даних у великій мережі.

4.3 Тестування відображення показників

Оскільки розроблена система передбачає можливість локального моніторингу без необхідності постійного підключення до Wi-Fi, важливим етапом стало тестування візуалізації даних на модулі дисплею, головною метою цього тестування була перевірка коректності виведення кліматичних показників, зчитаних з датчика BME280, читабельності інтерфейсу та стабільності оновлення інформації на екрані без візуальних артефактів. Процес тестування полягав у порівнянні значень, які мікроконтролер відправляє в UART-термінал, із тими, що безпосередньо виводиться на дисплеї. Як видно з рисунка 4.4, драйвер керування дисплеєм коректно отримує оброблені дані та формує зручний для сприйняття користувачем інтерфейс.

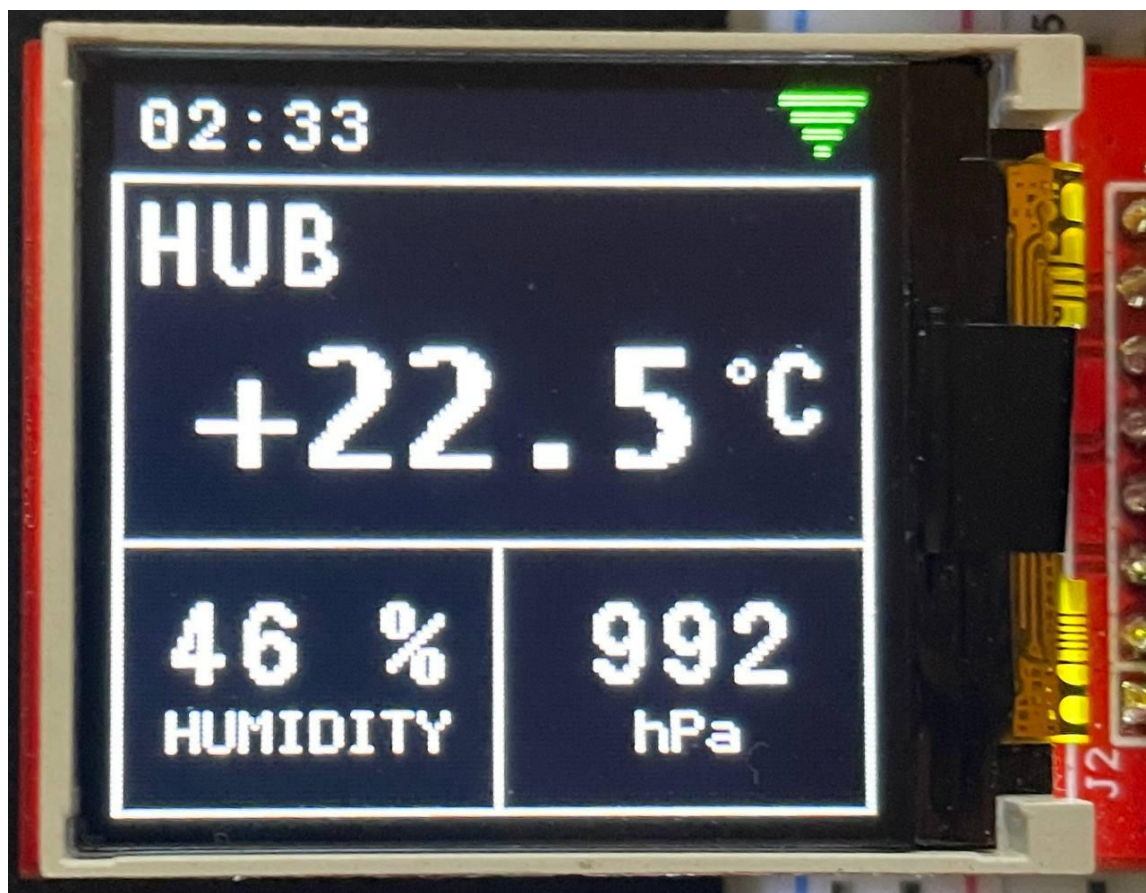


Рисунок 4.4 – Виведення поточних показників на дисплеї пристрою

Усі ключові параметри виводяться з дотриманням правильного форматування та одиниць вимірювання: температура в градусах Цельсія, відносна вологість у відсотках, а атмосферний тиск у гектопаскалях, що відповідає прийнятим метеорологічним стандартам. Окрема увага під час тестування приділялася процесу динамічного оновлення інформації. Оскільки мікроконтролер виконує зчитування даних циклічно, було перевірено реакцію дисплея на зміну показників, завдяки коректній програмній реалізації оновлення даних лише у разі їх зміни та без повного очищення екрана, що значно зменшує навантаження на мікроконтролер. Також було підтверджено відсутність конфліктів різних модулів на одній шині. Процес вимальовування графіки та тексту не створює критичних затримок у робочому циклі мікроконтролера.

Таким чином, результати тестування доводять, що пристрій успішно забезпечує наочний та безперебійний доступ до поточних даних на пристрої.

4.4 Тестування вебсайту

Наступним етапом перевірки програмно-апаратного комплексу було тестування розробленого вебсайту. Головною була перевірка коректності прийому, збереження та візуалізації цих показників у браузері. Тестування було розділено на перевірку серверної логіки та користувацького інтерфейсу. На рівні серверної частини перевірялася стабільність обробки вхідних запитів від мікроконтролера. Програмна діагностика підтвердила, що сервер успішно приймає мережеві пакети без втрат, перевіряє їх та коректно записує у базу даних, зберігаючи хронологію вимірювань.

Під час тестування користувацького інтерфейсу (рис. 4.5) перевірялася правильність відображення поточних показників. Дані з датчиків успішно підтягуються з бази даних та виводяться на інформаційній панелі у стандартизованих фізичних величинах. Також було протестовано модуль візуалізації статистичних даних, який використовується для перегляду кліматичних змін. Алгоритми побудови графіків працюють безперебійно, криві зміни параметрів навколишнього середовища відмальовуються плавно,

оновлюючись при надходженні нових даних, а масштабування осей та завантаження записів з бази відбувається без затримок і не викликає зависань вебсторінки.

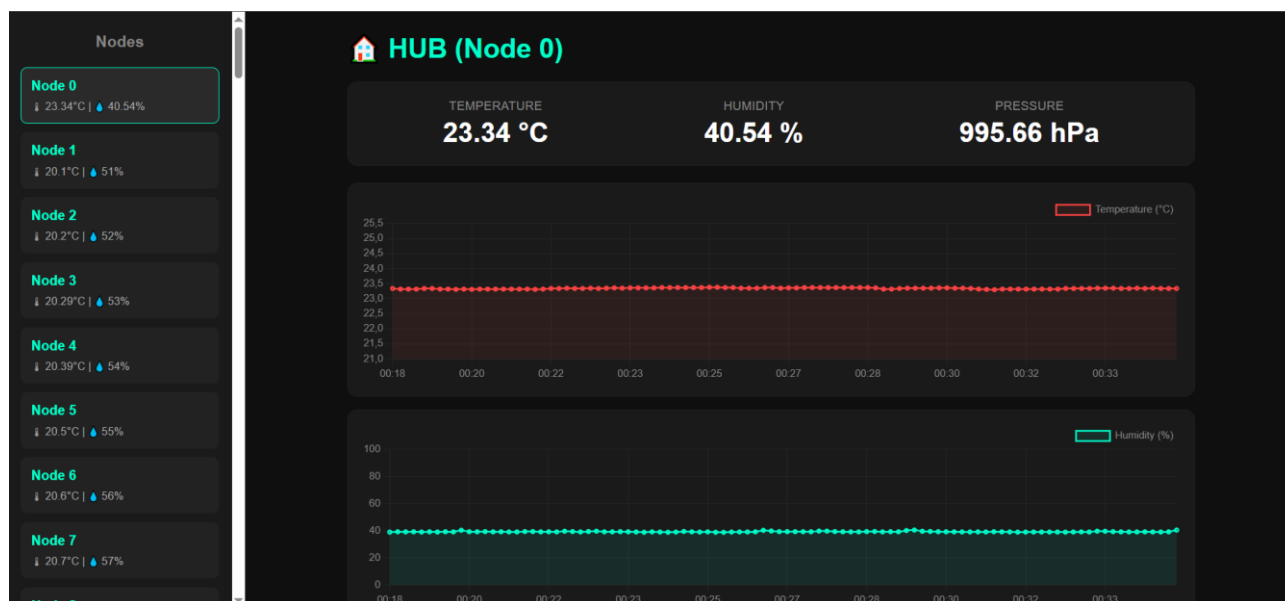


Рисунок 4.5 – Відображення кліматичних показників на сторінці сайту

У підсумку, результати тестування показують, що розроблений вебсайт є функціональним та стабільним інструментом для віддаленого моніторингу. Він успішно інтегрований з апаратною частиною системи і забезпечує зручний доступ до даних у режимі реального часу.

4.5 Тестування збереження та перегляду історії вимірювань

Було протестовано запис зібраних даних на SD-карту, яка працює під керуванням файлової системи FatFs. Тому окремим етапом стало тестування надійності апаратного запису та зручності їх подальшого перегляду. Для перевірки надійності системи збереження, пристрій було залишено в режимі безперервного збору та логування даних на тривалий період, після завершення тестування SD-карту було підключено до персонального комп'ютера для перевірки цілісності накопиченої інформації. Перегляд збереженого файлу (рис. 4.6) показав, що у файлі немає зайвих символів, пошкоджених рядків чи пропусків у часових мітках, усі записи зберегли правильне форматування з

розділювачами, що дозволяє табличним редакторам автоматично розпізнавати стовпці з даними.

	A	B	C	D	E
1	node_id	time_s	temperature	humidity	pressure
2990	0	3514	23.35	39.64	995.77
2991	0	3524	23.35	39.49	995.73
2992	0	3534	23.36	39.31	995.75
2993	0	3544	23.36	39.25	995.73
2994	0	3554	23.35	39.20	995.70
2995	0	3564	23.35	39.14	995.71
2996	0	3574	23.34	39.15	995.71
2997	0	3584	23.32	39.18	995.71
2998	0	3594	23.31	39.13	995.72
2999	0	3604	23.30	39.30	995.71
3000	0	3614	23.32	39.20	995.72
3001	0	3624	23.32	39.16	995.72
3002	0	3634	23.32	38.99	995.73
3003	0	3644	23.32	39.04	995.72
3004	0	3654	23.32	39.09	995.72
3005	0	3664	23.32	39.07	995.72
3006	0	3674	23.32	39.10	995.70
3007	0	3684	23.32	39.08	995.69
3008	0	3694	23.34	39.00	995.69
3009	0	3704	23.34	39.11	995.67
3010	0	3714	23.34	39.17	995.68
3011	0	3724	23.34	39.12	995.67
3012	0	3734	23.35	39.71	995.69
3013	0	3744	23.35	39.63	995.69
3014	0	3754	23.35	39.35	995.66
3015	0	3764	23.34	39.19	995.68
3016	0	3774	23.34	39.15	995.66
3017	0	3784	23.35	39.15	995.66
3018	0	3794	23.34	39.22	995.65

Рисунок 4.6 – Фрагмент збереженого файлу з історією показників на SD-карті

Це підтверджує, що операції запису через бібліотеку FatFs виконуються коректно і не перериваються іншими апаратними процесами мікроконтролера. Результати тестування підтверджують, що підсистема роботи з SD-картою функціонує стабільно. Пристрій гарантує безпечне довгострокове збереження зібраних показників та забезпечує розробника і користувача зручним форматом даних для їх подальшого перегляду та статистичного аналізу на ПК.

4.6 Тестування енергоефективності вузлів

В рамках тестування енергоспоживання периферійного вузла було проведено оцінку споживання електроенергії. Вузол періодично виконує вимірювання параметрів мікроклімату за допомогою датчика та передає отримані дані через модуль LoRa. Після завершення передачі система переходить у режим сну до наступного циклу вимірювання.

За результатами тестування середнє споживання струму системи становить приблизно 2 мА при періодичності передачі даних один раз на хвилину. Основна частина енергії споживається під час активної роботи мікроконтролера та передачі даних по каналу LoRa, тоді як більшу частину часу система перебуває в режимі низького енергоспоживання. Для оцінки автономності було розглянуто використання акумулятора ємністю 3000 мАг. За таких умов теоретичний час автономної роботи системи може становити близько 60 діб без підзарядки, реальний час роботи залежатиме від умов експлуатації, якості акумулятора та температури навколишнього середовища.

Отримані результати свідчать про високу енергоефективність розробленої системи та підтверджують можливість її тривалої автономної експлуатації. У подальшому енергоспоживання може бути додатково знижене шляхом оптимізації режимів роботи мікроконтролера, скорочення часу активності радіомодуля або збільшення інтервалу між передаваннями даних.

5. КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Інтерфейс користувача

Вебінтерфейс системи створено таким чином, щоб забезпечити максимально інтуїтивну та зручну роботу користувача без необхідності ручного перезавантаження вебсторінки. Візуальне оформлення інтерфейсу виконано в темній темі, робочий простір логічно розділений на дві основні взаємопов'язані зони, які включають бічну панель навігації з лівого боку екрана та головну інформаційну панель по центру.

Бічна панель призначена для динамічного відображення переліку всіх доступних метеостанцій, які зареєстровані в мережі моніторингу та передають дані. Для кожного вузла у цьому списку виводиться його унікальний текстовий ідентифікатор, а також короткий зріз останніх отриманих значень температури та відносної вологості повітря. Щоб переключити увагу на конкретний пристрій та завантажити його повну статистику, користувачеві достатньо натиснути мишкою на відповідний блок у списку, після чого обраний вузол автоматично виділяється рамкою, підтверджуючи зміну активного контексту.

Головна панель слугує для детальнішого перегляду параметрів навколишнього середовища, що надходять від обраного на бічній панелі пристрою. У верхній частині цієї зони розташовано блок оперативних показників із великими цифровими індикаторами, де відображаються найсвіжіші вимірювання. Прямо під ними розміщено блок графіків, який складається з трьох окремих лінійних діаграм для кожної фізичної величини. Ці діаграми дозволяють наочно оцінити тенденції зміни показників, використовуючи позначки часу на горизонтальній осі часу.

5.2 Підключення до Wi-Fi мережі

Для забезпечення можливості експлуатації в різних локаціях без перепрограмування мікроконтролера, пристрій має алгоритм швидкого налаштування параметрів мережі, який дозволяє підключити його за допомогою смартфона, планшета або. Щоб розпочати процес налаштування потрібно натиснути та утримувати кнопку керування пристроєм протягом трьох секунд. На екрані метеостанції з'явиться напис "SETUP MODE", який свідчить про успішний перехід у режим конфігурації(рис 5.1), на дисплеї відобразиться назва тимчасової мережі пристрою та адреса для підключення.



Рисунок 5.1 – Виведення інформації для підключення до Wi-Fi-мережі

Далі на смартфоні або комп'ютері потрібно під'єднатися до створеної пристроєм Wi-Fi-мережі, після успішного з'єднання необхідно відкрити будь-який веб-браузер і ввести у адресному рядку IP-адресу 192.168.4.1. На екрані

завантажиться сторінка(рис 5.2) з полями для введення даних точки доступу до якої має бути під'єднаний пристрій

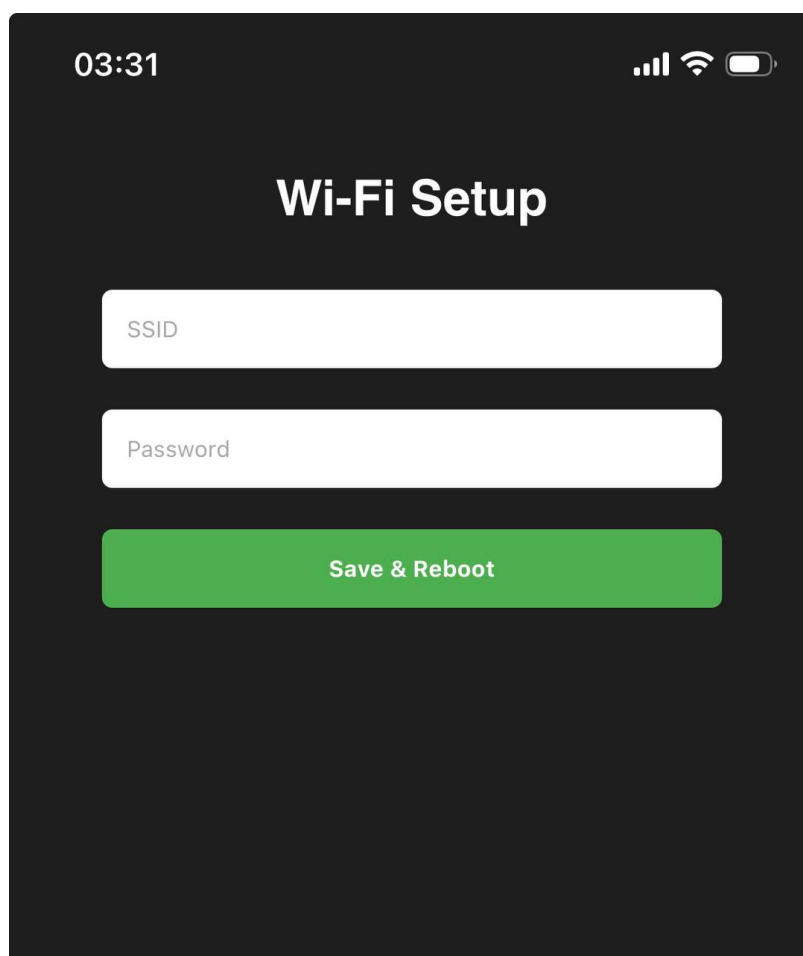


Рисунок 5.2 – Сторінка під'єднання до Wi-Fi-мережі

Після введення даних та натиснувши кнопку збереження на екрані з'явиться повідомлення про успішне збереження введених даних. Після цього метеостанція автоматично перезавантажиться і спробує підключитися до обраної мережі. Про успішне з'єднання свідчитиме зміна індикатора статусу Wi-Fi у верхньому правому куті дисплею пристрою.

5.3 Підключення нових вузлів

Для спрощення процесу підключення нових периферійних вузлів, процес інтеграції до загальної мережі відбувається повністю в автоматичному режимі й не потребує додаткових дій від користувача. Після подачі живлення на

периферійний вузол його мікроконтролер самостійно ініціалізує бездротовий інтерфейс і починає передачу службових пакетів із власним унікальним апаратним ідентифікатором. Центральна станція у штатному режимі функціонування безперервно прослуховує ефір, автоматично виявляє новий пристрій, зберігає його у внутрішній пам'яті та починає приймати показники, які одразу надходять для логування та відображення на вебсайті.

5.4 Перегляд історії вимірювань

Для аналізу змін показників протягом певного часу система підтримує функцію архівації та перегляду історії вимірювань. Доступ до накопичених ретроспективних даних може здійснюватися дистанційно через вебсайт за допомогою будь-якого веббраузера. У бічній панелі інтерфейсу користувач має можливість обрати потрібний периферійний вузол за його унікальним ідентифікатором, а також задати бажаний часовий інтервал за допомогою вбудованого інструменту вибору дати. Після підтвердження запиту система генерує інтерактивні графіки та часові шкали, що наочно відображають динаміку зміни температури, вологості та інших атмосферних параметрів для обраного модуля (рис. 5.2).

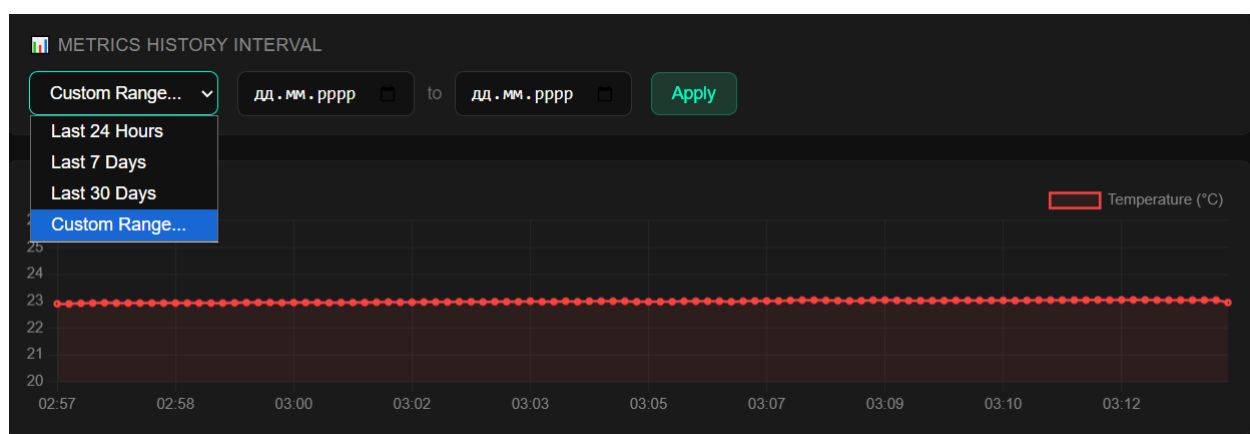


Рисунок 5.3 – Вікно перегляду історії вимірювань

Також, у системі реалізовано можливість перегляду та резервного копіювання історії за допомогою SD-карти. Усі отримані з вузлів кліматичні показники автоматично записуються на карту пам'яті у вигляді структурованих файлів у форматі CSV із прив'язкою до часу та ідентифікатора пристрою. Для

перегляду накопиченої інформації користувачеві достатньо вилучити SD-карту з центрального хабу та підключити її до персонального комп'ютера чи ноутбука. Завдяки стандартному формату запису, отримані файли можна легко відкрити у будь-яких табличних програмах для подальшого аналізу, побудови власних діаграм або архівування на локальному диску.

					ІАЛЦ.468300.004 ПЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		55

ВИСНОВКИ

У результаті виконання дипломного проєкту на тему «Програмно-апаратна система моніторингу якості повітря на основі плати розробника сімейства STM32» було розроблено, реалізовано та протестовано систему, що забезпечує збір, обробку, візуалізацію, бездротову передачу та локальне збереження даних із периферійних датчиків у реальному часі.

На початковому етапі роботи було проведено аналіз існуючих рішень. Було виявлено, що більшість доступних комерційних і відкритих проєктів мають обмежену гнучкість у розширенні, складну інтеграцію нових датчиків або недостатньо інформативний користувацький інтерфейс. Це обґрунтувало вибір концепції розподіленої системи, яка завдяки модульному підходу до розробки програмного забезпечення легко адаптується до нових апаратних конфігурацій, підтримує передачу даних на великі відстані засобами технології LoRa та забезпечує доступ до інформації через локальну мережу.

Було обрано апаратну платформу на базі мікроконтролера STM32, що поєднує високу обчислювальну потужність, низьке енергоспоживання та широкі можливості інтеграції периферії. До системи підключено периферійний датчик BME280 для вимірювання температури, вологості та атмосферного тиску, а для далекобійної бездротової передачі даних застосовано радіомодуль LoRa XL1278. Для відображення показників у реальному часі використовується графічний дисплей TFT ST7735S, а тривале зберігання зібраних даних реалізовано засобами модуля SD-карти під керуванням файлової системи FatFs. Центральний вузол доповнено Wi-Fi-модулем для забезпечення зв'язку з локальним сервером.

Програмна частина системи розроблена з урахуванням принципів модульності та розширюваності. На рівні вбудованого програмного забезпечення реалізовано драйвери для роботи з усіма апаратними компонентами, алгоритм часового розділення для уникнення колізій у мережі

вузлів LoRa, що забезпечує точну синхронізацію передачі даних. Серверна частина центрального вузла відповідає за прийом мережових пакетів, їх декодування та запис до бази даних. Розроблений веб-інтерфейс надає користувачеві зручний доступ до актуальних та архівних показників усіх вузлів мережі через браузер без необхідності встановлення додаткового програмного забезпечення. Особливу увагу під час розробки приділено стабільності та коректній обробці помилкових або пошкоджених пакетів.

Після завершення розробки систему зібрано на макетній платі та проведено всебічне тестування. Перевірено коректність взаємодії з датчиком BME280, стабільність часової синхронізації між вузлами, надійність запису та зчитування даних із SD-карти, а також коректність роботи серверної логіки та веб-інтерфейсу при імітації підключення великої кількості вузлів. Результати тестування підтвердили працездатність системи у всіх перевірених режимах роботи та відповідність її функціоналу поставленим завданням.

Розроблене рішення є надійним фундаментом для подальшого розвитку. У перспективі можливе додавання підтримки нових типів датчиків, розширення функціональності веб-інтерфейсу засобами аналітики, реалізація автоматичного керування зовнішніми пристроями на основі зібраних показників. Усе це може перетворити систему на повноцінну платформу для побудови елементів розумного будинку або для промислового моніторингу стану повітря у великих приміщеннях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. STM32F411CEU6 Datasheet. STMicroelectronics [Електронний ресурс]. – Режим доступу: <https://www.st.com/resource/en/datasheet/stm32f411ce.pdf>. – Дата звернення: травень 2026.
2. BME280 Combined Humidity and Pressure Sensor Datasheet. Bosch Sensortec [Електронний ресурс]. – Режим доступу: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf>. – Дата звернення: травень 2026.
3. SX1276/77/78/79 LoRa Modem Datasheet. Semtech [Електронний ресурс]. – Режим доступу: https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf. – Дата звернення: травень 2026.
4. ESP8266EX Datasheet. Espressif Systems [Електронний ресурс]. – Режим доступу: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. – Дата звернення: травень 2026.
5. ST7735S Datasheet. Sitronix Technology Corporation [Електронний ресурс]. – Режим доступу: <https://www.displayfuture.com/Display/datasheet/controller/ST7735.pdf>. – Дата звернення: травень 2026.
6. Sencor SWS 4900 Weather Station [Електронний ресурс]. – Режим доступу: <https://www.sencor.com/station/sws-4900>. – Дата звернення: травень 2026.
7. Mi Temperature and Humidity Monitor 3. Xiaomi [Електронний ресурс]. – Режим доступу: <https://mi.ua/ua/datchiki-dlja-signalizacij/mi-temperature-and-humidity-monitor-3-bhr9041gl.html>. – Дата звернення: травень 2026.
8. Ajax LifeQuality – Розумний монітор якості повітря. Ajax Systems [Електронний ресурс]. – Режим доступу: <https://ajax.systems/products/lifequality/>. – Дата звернення: травень 2026.

9. Uutzingер. Air Quality Monitoring System. GitHub [Електронний ресурс]. – Режим доступу: <https://github.com/uutzingер/airquality>. – Дата звернення: травень 2026.
10. Flask Web Framework Documentation. Pallets Projects [Електронний ресурс]. – Режим доступу: <https://flask.palletsprojects.com/en/stable>. – Дата звернення: травень 2026.
11. FatFs Generic FAT Filesystem Module. ChaN [Електронний ресурс]. – Режим доступу: <https://elm-chan.org/fsw/ff>. – Дата звернення: травень 2026.
12. Adafruit ST7735 Library. Adafruit Industries [Електронний ресурс]. – Режим доступу: <https://github.com/adafruit/adafruit-st7735-library>. – Дата звернення: травень 2026.
13. Chart.js Documentation. Chart.js Contributors [Електронний ресурс]. – Режим доступу: <https://www.chartjs.org/docs/latest>. – Дата звернення: травень 2026.
14. I2C-bus Specification and User Manual. NXP Semiconductors [Електронний ресурс]. – Режим доступу: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. – Дата звернення: травень 2026.
15. Python Programming Language. Python Software Foundation [Електронний ресурс]. – Режим доступу: <https://www.python.org>. – Дата звернення: травень 2026.
16. UART: A Hardware Communication Protocol. Analog Devices [Електронний ресурс]. – Режим доступу: <https://www.analog.com/en/resources/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>. – Дата звернення: травень 2026.
17. TP4056 1A Standalone Linear Li-Ion Battery Charger Datasheet. NanJing Top Power ASIC Corp [Електронний ресурс]. – Режим доступу: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf>. – Дата звернення: травень 2026.

18. MT3608 2A 1.2MHz Step-Up Converter Datasheet. Aerosemi Technology [Електронний ресурс]. – Режим доступу: <https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf>. – Дата звернення: травень 2026.
19. Learn-C.org: Interactive C Tutorial [Електронний ресурс]. – Режим доступу: <https://www.learn-c.org/>. – Дата звернення: травень 2026.
20. Description of STM32F4 HAL and Low-Layer Drivers. STMicroelectronics [Електронний ресурс]. – Режим доступу: https://www.st.com/resource/en/user_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf. – Дата звернення: травень 2026.