

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розроблення автоматизованої системи збору та аналізу метеоданих
з використанням технологій IoT (комплексна тема)

Виконав(ла): студент(ка) IV курсу, групи КТЗ-41
спеціальності 151 Автоматизація та
комп'ютерно - інтегровані технології
(шифр і назва спеціальності)

		Русінка О.В.
	(підпис)	Чепчур М.О. (прізвище та ініціали)
Керівник	(підпис)	Голотенко О.С. (прізвище та ініціали)
Нормоконтроль	(підпис)	Чихіра І.В. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Микитишин А.Г. (прізвище та ініціали)
Рецензент	(підпис)	 (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра комп'ютерно-інтегрованих технологій
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Микитишин А.Г.

(підпис)

(прізвище та ініціали)

«__» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 151 Автоматизація та комп'ютерно - інтегровані технології
(шифр і назва спеціальності)

студенту Русінці Олександр Василювичу, Чепчуру Мар'яну Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматизованої системи збору та аналізу метеоданих з використанням технологій IoT (комплексна тема)

Керівник роботи Голотенко Олександр Сергійович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «31» березня 2025 року № 4/7-240

2. Термін подання студентом завершеної роботи 12.06.2025р.

3. Вихідні дані до роботи технічна документація

4. Зміст роботи (перелік питань, які потрібно розробити)

1. Аналітична частина;

2. Проєктна частина;

3. Спеціальна частина;

4. Безпека життєдіяльності, основи охорони праці.

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

Титульний слайд

Актуальність роботи

Завдання роботи

Основна частина

Висновки

7. Дата видачі завдання[illegible]

(підпис)

(прізвище та ініціали)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Розроблення автоматизованої системи збору та аналізу метеоданих з використанням технологій IoT (комплексна тема) // Кваліфікаційна робота бакалавра // Русінка Олександр Васильович, Чепчур Мар'ян Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет прикладних інформаційних технологій та електроінженерії, кафедра комп'ютерно-інтегрованих технологій, група КТз-41 // Тернопіль, 2025 // с. – 74, рис. – 15, табл. – 22, додат. – 4, бібліогр. – 14.

Ключові слова: метеодані, метеостанція, Інтернет речей, мікроконтролер, датчик, ESP32, WiFi.

Кваліфікаційна робота присвячена розробленню автономної системи збору метеоданих на базі мікроконтролера ESP32 з вбудованим Wi-Fi-модулем для автоматизованого збору та аналізу метеоданих з використанням технологій інтернету речей.

Метою проєкту є створення компактного, енергоефективного та доступного пристрою, здатного в реальному часі здійснювати моніторинг основних метеопараметрів: температури, вологості, атмосферного тиску та якості повітря. У процесі розробки було здійснено аналіз існуючих аналогів, сформульовано технічні вимоги, обґрунтовано вибір апаратних та програмних компонентів системи.

Спроектовано структурну схему пристрою. Також здійснено обґрунтований вибір необхідних апаратних і програмних компонентів, з яких складається прототип автоматизованої системи збору та аналізу метеоданих

Передбачено зручний веб-інтерфейс, що дозволяє переглядати графіки показників у браузері без підключення до зовнішніх серверів.

Результати тестування підтвердили працездатність системи, її точність та зручність у використанні. Розроблений пристрій може бути застосований у домашньому, освітньому або дослідницькому середовищі для контролю стану мікроклімату.

ЗМІСТ

АНОТАЦІЯ	4
ЗМІСТ	5
ВСТУП.....	7
1. АНАЛІТИЧНА ЧАСТИНА	9
1.1 Класифікація систем збору метеоданих	9
1.2 Аналіз наявних рішень	10
1.3 Вибір варіанту реалізації проєкту та його обґрунтування	14
1.4 Основні вимоги до створюваної системи.....	15
1.4.1 Вимоги до апаратної частини.....	15
1.4.2 Вимоги до програмної частини.....	18
1.4.3 Функціональні можливості системи.....	19
2. ПРОЄКТНА ЧАСТИНА	21
2.1. Вибір компонентів апаратної частини.....	21
2.1.1 Вибір мікроконтролера	22
2.1.2 Вибір датчиків	26
2.1.3 Вибір дисплея	28
2.1.4 Вибір джерела живлення	29
2.2 Вибір компонентів програмної частини.....	30
2.2.1 Вибір мови програмування.....	31
2.2.2 Вибір середовища розробки	31
2.2.3 Вибір веб-сервера та способу зберігання інформації.....	32
2.3 Конструктивні особливості корпусу.....	33
2.4 План реалізації проєкту.....	34
3. СПЕЦІАЛЬНА ЧАСТИНА	36
3.1 Підключення компонентів розробленої системи	36
3.1.1 Модуль заряджання TP4056 та джерело живлення.	38
3.1.2 Датчик BME280 та OLED-дисплей	40
3.1.3 Датчик MQ135 та RGB-світлодіод	44
3.1.4 Загальна схема підключення компонентів	46

	6
3.2 Розробка програмного забезпечення	49
3.2.1 Файлова структура проєкту.....	51
3.2.2 Вміст основних файлів.....	51
4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	54
4.1. Загальна характеристика приміщення і робочого місця.....	54
4.2. Аналіз потенційно небезпечних і шкідливих виробничих факторів на робочому місці	56
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТОК А.....	67
ДОДАТОК Б.....	69
ДОДАТОК В.....	71
ДОДАТОК Г	72

ВСТУП

У сучасному світі, що стрімко змінюється під впливом глобального потепління, екстремальних погодних явищ та антропогенного навантаження, збір, зберігання й аналіз метеорологічних даних набуває першочергового значення. Надійне моніторингове забезпечення стає не лише інструментом для наукових досліджень, а й практичним засобом прогнозування небезпечних умов, своєчасного попередження та ухвалення управлінських рішень.

Одним із головних завдань метеорології сьогодні є систематичне спостереження за змінами таких параметрів, як температура повітря, відносна вологість, атмосферний тиск, швидкість і напрямок вітру, кількість опадів, рівень сонячної радіації та концентрація вуглекислого газу. Ці показники дозволяють будувати точні моделі кліматичних процесів, прогнозувати розвиток стихійних лих та оцінювати їхній вплив на природне середовище і діяльність людини.

Сучасні метеостанції, оснащені цифровими сенсорами, модемами зв'язку та обчислювальними модулями, забезпечують автоматичний та безперервний збір даних у режимі реального часу [1, 9]. Завдяки цьому метеорологи, екологи, аграрії, транспортні служби та навіть пересічні користувачі можуть отримувати актуальну інформацію про стан навколишнього середовища.

Інформаційно-аналітична складова метеосистем дозволяє виявляти кліматичні тренди, сезонні відхилення, зональні коливання, а також аномальні погодні патерни. Ці дані є основою для довгострокового планування в енергетиці, сільському господарстві, будівництві, авіації, морській логістиці та інших сферах.

Окрім практичного застосування, метеодані мають вагоме наукове значення. Вони використовуються в численних наукових дослідженнях, пов'язаних із моделюванням кліматичних змін, адаптацією до них, а також оцінкою ризиків, спричинених змінами у довкіллі.

Розвиток мікроелектроніки та інтернету речей (IoT) призвів до створення компактних, енергоефективних і доступних побутових метеостанцій, які дедалі частіше використовуються для локального моніторингу [2]. Це створює

можливість створення розподілених мереж приватних спостережних пунктів, які, за умови інтеграції в централізовану систему, можуть стати додатковим джерелом цінної інформації.

Крім того, метеодані можуть застосовуватися в системах "розумного будинку" — для автоматичного керування обігрівом, вентиляцією чи поливом рослин, що робить побут ефективнішим і екологічно відповідальнішим [2].

Актуальність розробки сучасної системи збору метеоданих також обумовлена потребою у створенні доступних, автономних та легко масштабованих рішень, які не потребують складного технічного обслуговування. Такі системи повинні поєднувати простоту у використанні, низьку вартість, енергоефективність та можливість розширення функціоналу відповідно до потреб користувача [3].

Таким чином, дослідження, пов'язане з розробкою та реалізацією компактної метеостанції, має високу практичну та наукову значущість, а також сприяє популяризації екологічного моніторингу серед широкого кола користувачів.

1. АНАЛІТИЧНА ЧАСТИНА

1.1 Класифікація систем збору метеоданих

Система збору метеоданих (або пристрій збору метеоданих) — це технічний засіб або комплекс приладів, котрий призначений для вимірювання, фіксації та передавання метеорологічних параметрів отриманих з навколишнього середовища. Така система може інтегрувати в себе датчики температури, вологості, атмосферного тиску, швидкості й напрямку вітру, а також опадів [3, 4]. Узагальнено цей пристрій часто називають метеостанцією — незалежно від її масштабу чи рівня складності..

Метеостанції можна класифікувати за кількома критеріями [5]:

- ***За функціональними можливостями:***

- *Базові метеостанції* — забезпечують вимірювання основних метеопоказників: температури повітря, атмосферного тиску, відносної вологості повітря, швидкості та напрямку вітру.
- *Розширені або професійні метеостанції* — додатково оснащені давачами опадів, ультрафіолетового випромінювання, рівня сонячної радіації, ультразвуковими далекомірами та іншими сенсорами.

- ***За принципом роботи:***

- *Аналогові* — використовують механічні або електромеханічні елементи, наприклад, біметалеві термометри або анероїдні барометри.
- *Цифрові* — базуються на сучасних електронних сенсорах, що забезпечують підвищену точність, автоматичний запис і передавання даних.

- ***За сферою застосування:***

- Побутові (домашні), для саду чи сільського господарства, морські та авіаційні, наукові та дослідницькі метеостанції.

- **За типом живлення:**

- Моделі, що працюють від батарейок, сонячних панелей, USB-порту або електромережі.

- **За точністю вимірювання:**

- Побутові метеостанції для пересічних користувачів, професійні прилади для фахівців, а також високоточні наукові комплекси.

У даній роботі основна увага приділена створенню побутової домашньої метеостанції, що зорієнтована на користувача-аматора, але з можливістю розширення функціоналу.

1.2 Аналіз наявних рішень

На сьогодні ринок насичений великою кількістю готових рішень для збору метеоданих, які здатні виконувати базові або розширені функції моніторингу навколишнього середовища. Проте в багатьох випадках такі системи не враховують індивідуальні вимоги користувача: специфіку розміщення, необхідні параметри чи особливості інтеграції з іншими пристроями [3, 4, 5].

У таких ситуаціях доцільним варіантом є розробка власної системи збору метеоданих, адаптованої до конкретних умов та завдань. У межах цієї роботи буде спроектована та розроблена така індивідуальна метеостанція.

Зокрема ми проведемо аналіз існуючих комерційних рішень, сформулюємо технічні вимоги до нової системи, розробимо архітектуру та основні функціональні блоки.

У метеорології традиційно використовуються такі датчики [6]:

- **термометри** — для вимірювання температури;
- **барометри** — для визначення атмосферного тиску;
- **гігрометри** — для фіксації вологості повітря.

У таблиці 1 наведено порівняльні характеристики деяких популярних домашніх метеостанцій.

Таблиця 1.1 - Порівняльні характеристики популярних домашніх метеостанцій

Модель	Виробник	Температура (°C)	Вологість (%)	Живлення	Орієнтовна вартість (грн)	Особливості
Bresser TemeoTrend JC	Bresser (Німеччина)	-20...+60	Н/Д	6 АА батарейок	~3 000	Великий дисплей, підсвічування, радіоконтроль часу
La Crosse WS6811	La Crosse (США)	-40...+60	Н/Д	2 АА батарейок	~2 500	Компактність, простота у використанні
Oregon Scientific BAR208S	Oregon Scientific (США)	-20...+60	25–95	3 АА батарейок	~3 500	Прогноз погоди, індикатор заморозків
Xiaomi Mi Temp. & Hum. Monitor 2	Xiaomi (Китай)	0...+60	0–99	CR2032 батарейка	~600	Bluetooth, інтеграція з екосистемою Xiaomi
Rexant 70-0501	Rexant (Китай)	Н/Д	Н/Д	2 ААА батарейок	~1 200	Простий інтерфейс, доступна ціна
TFA Dostmann 35.1140	TFA Dostmann (Німеччина)	-40...+60	1–99	3 АА батарейок	~6 800	Вимірювання вітру, прогноз погоди
AcuRite Iris (5-in-1)	AcuRite (США)	-40...+70	1–99	Сонячна батарея (сенсор) + електромережа (дисплей)	~8 000	Компактний дизайн, підтримка додаткових датчиків
VEVOR 7-in-1 Wi-Fi Solar	VEVOR (Китай)	Н/Д	Н/Д	Сонячна батарея з акумулятором	~7 500	7" TFT-дисплей, Wi-Fi, автоматичний моніторинг

Також поширені професійні метеостанції. Їх основна відмінність від побутових полягає у набагато вищій точності вимірювань та розширеному функціоналу. Такі метеостанції здатні фіксувати ширший спектр метеорологічних параметрів і оснащені високоточними сенсорами, що дозволяє отримувати більш достовірні та стабільні дані.

Крім того, професійні системи збору метеоданих підтримують найновіші функції обробки інформації, зокрема [6, 7]:

- автоматичне збереження і обробку показників;

- аналіз тенденцій (трендів);
- побудову графіків та діаграм;
- передачу даних до комп'ютера або на сервер через інтернет.

Ще однією перевагою професійних метеостанцій є міцніша конструкція, що забезпечує надійну роботу в складних кліматичних умовах: за високої вологості, сильного вітру, екстремальних температур (як низьких, так і високих).

Водночас, такі метеостанції мають і певні недоліки:

- висока вартість у порівнянні з побутовими моделями;
- складніший монтаж та налаштування;
- великий розмір і вага, що ускладнює транспортування та встановлення у різних місцях.

У Таблиці 2 наведено приклади сучасних професійних метеостанцій.

На основі проведеного аналізу даних з таблиць і порівняння типів метеостанцій зроблено висновок, що кожен тип має свої переваги та недоліки, які визначаються цілями використання та вимогами користувача.

Зокрема, професійні метеостанції вирізняються високою точністю та ширшим функціоналом, але при цьому зазвичай значно дорожчі, потребують спеціального фахового налаштування та обслуговування. Натомість домашні метеостанції в основному мають доступнішу ціну та забезпечують базовий набір функцій, проте можуть поступатися у точності вимірювань та технічних можливостях.

У процесі розробки власної системи доцільно реалізувати такі функції:

- здійснювати вимірювання температури та атмосферної вологості повітря як всередині, так і зовні приміщення;
- фіксування атмосферного тиску, швидкості і напрямку вітру;
- прогнозування погоди, сформоване на основі отриманих даних;
- графічне відображення отриманих показників на дисплеї;
- можливість інтегрувати керування додатковими пристроями (наприклад, обігрівачем чи кондиціонером) відповідно до змін погодних умов;
- інтеграція з мобільним додатком для дистанційного моніторингу даних;
- контроль якості повітря всередині приміщення.

Таблиця 1.2 - Порівняльні характеристики професійних метеостанцій

Модель	Виробник	Діапазон температури (°C)	Діапазон вологості (%)	Тип живлення	Орієнтовна ціна (грн)	Особливості
Davis Vantage Pro2 Plus (SKU 6262)	Davis Instruments (США)	-40...+65	0–100	Сонячна батарея + резервна батарея + АС	~63 000	Високоточна, модульна конструкція, розширені можливості, UV та сонячна радіація
Ambient Weather WS-5000	Ambient Weather (США)	-40...+60	0–100	Сонячна батарея з акумулятором	~17 000	Ультразвукові датчики, Wi-Fi, кольоровий дисплей
KestrelMet 6000 AG	Kestrel (США)	-40...+60	0–100	Сонячна батарея з акумулятором	~37 000	Призначена для аграрного моніторингу, точні вимірювання, Wi-Fi
Sainlogic FT0300	Sainlogic (США)	-40...+60	0–100	Сонячна батарея з акумулятором	~7 000	Wi-Fi, мобільний додаток, кольоровий дисплей
Rain Bird WS-PRO2-PH	Rain Bird (США)	-40...+60	0–100	Сонячна батарея з акумулятором	~52 000	Призначена для аграрного моніторингу, точні вимірювання, Wi-Fi
Davis Vantage Pro2 GroWeather (SKU 6820)	Davis Instruments (США)	-40...+65	0–100	Сонячна батарея + резервна батарея + АС	~37 000	Призначена для аграрного моніторингу, точні вимірювання, Wi-Fi
Columbia Weather Magellan MX500	Columbia Weather Systems (США)	-40...+60	0–100	Сонячна батарея з акумулятором	~100 000	Портативна, GPS, компас, Wi-Fi
La Crosse V61 Wi-Fi Professional	La Crosse Technology (США)	-40...+60	0–100	Сонячна батарея з акумулятором	~7 000	Wi-Fi, мобільний додаток, кольоровий дисплей

1.3 Вибір варіанту реалізації проєкту та його обґрунтування

Враховуючи наявні рішення та принципи їх роботи, визначено кілька підходів до реалізації системи:

1) Взаємодія зі станцією з допомогою Telegram-бота.

Дає змогу надсилати повідомлення про поточні значення метеоданих, встановлювати порогові значення та отримувати сповіщення при їх досягненні. Однак такий підхід може обмежити гнучкість і функціонал системи.

2) Підключення до мережі internet та передавання даних на сервер.

Цей метод забезпечує більші можливості для обробки, зберігання та візуалізації даних (наприклад, у вигляді графіків), але вимагає постійного інтернет-з'єднання та наявності окремого сервера.

3) Реалізація веб-сервера безпосередньо на мікроконтролері.

Дає змогу швидко отримувати та переглядати дані локально, без необхідності стороннього сервера. Проте ресурси мікроплати (пам'ять, обчислювальна потужність) обмежені, а налаштування може бути складнішим.

4) Створення мобільного додатка.

Додаток зчитує дані через Bluetooth або Wi-Fi та відображає їх на екрані смартфона. Зручно для кінцевого користувача, проте потребує розробки клієнтської програми.

5) Автономна система з записом даних на SD-карту.

Дані зберігаються локально на micro SD карті пам'яті й можуть бути (за потреби) передані на ПК для аналізу. Такий підхід не потребує підключення до мережі, але не забезпечує доступ у реальному часі.

В нашому випадку ми будемо реалізовувати веб-сервер безпосередньо на платі.

Переваги:

- Швидкий доступ до даних, оскільки вони зберігаються локально.
- Мінімальні витрати на обслуговування, відсутня потреба в оплаті хостингу або сервера.
- Автономність, система не залежить від сторонніх сервісів.

Недоліки:

- Обмежені ресурси плати (пам'ять, обробка даних).
- Складність масштабування, у разі збільшення обсягу даних.
- Складніша конфігурація, порівняно з серверним рішенням.

Однак головним недоліком є лише технічна складність реалізації, яка не є критичною в межах кваліфікаційної роботи. За потреби, функціональність можна розширити, наприклад, за рахунок встановлення micro SD карти для збільшення об'єму пам'яті.

1.4 Основні вимоги до створюваної системи

Ключові завдання:

а) Збір метеоданих.

Потрібно визначити, які датчики будуть використовуватись для моніторингу температури, вологості, тиску, швидкості та напрямку вітру, а також інших параметрів.

б) Зберігання даних.

Необхідно реалізувати базу даних, яка дозволить зберігати й накопичувати всі отримані показники. Важливо реалізувати можливість масштабування для довготривалого збереження даних.

в) Обробка даних.

Потрібно створити алгоритми аналізу, які будуть формувати звіти, виявляти зміни та будувати графіки залежно від часу.

г) Надання доступу до інформації.

Необхідно реалізувати веб-інтерфейс, через який користувачі зможуть моніторити результати вимірювань у зручному поданні.

1.4.1 Вимоги до апаратної частини

До апаратної частини розроблюваної системи належать такі основні компоненти:

- джерело живлення;
- датчики;

- пристрій керування (мікроконтролер);
- засоби виводу та індикації;
- модуль Wi-Fi.

Кожен із цих елементів виконує важливу роль у забезпеченні функціонування метеостанції, і до них висуваються відповідні технічні вимоги щодо надійності, сумісності та енергоефективності.

На рисунку 1 подано прототип структурної схеми системи, котрий ілюструє взаємозв'язки між основними апаратними модулями.

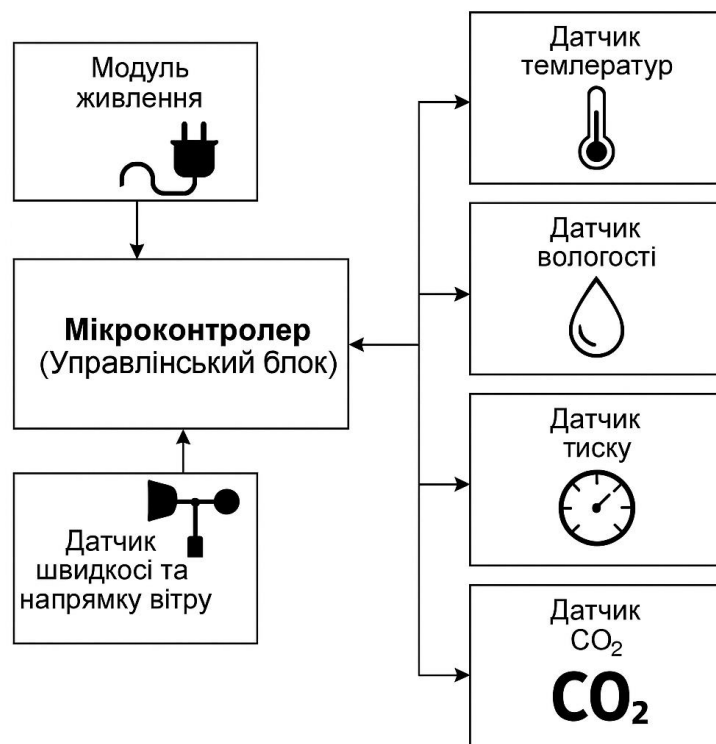


Рис. 1.1. Прототип структурної схеми системи

– Джерело живлення.

Необхідне для забезпечення електроживлення всієї системи. Вихідна напруга повинна бути не нижчою за мінімальну напругу, яка необхідна для роботи мікроконтролера. Конкретне значення споживаного струму залежить від характеристик підключених датчиків, отже, перед вибором блока живлення доцільно провести розрахунок загального струмоспоживання.

Фізичні розміри джерела живлення варто підібрати такими, щоб вони не були більшими за розміри контролера — це дозволить розмістити розроблену

систему у компактному захищеному корпусі. Підвищену автономність можна забезпечити, використовуючи джерело живлення з модулем зарядки та акумулятором.

– ***Датчики.***

Це фізичні пристрої, які перетворюють фізичні величини (температура, тиск, вологість тощо) у електричний сигнал, що читається мікроконтролером. Датчики є основним функціональним елементом системи, і саме вони забезпечують отримання метеорологічних параметрів. Нам потрібно забезпечити:

- вимірювання температури повітря;
- вимірювання відносної вологості;
- вимірювання атмосферного тиску;
- визначення концентрації CO₂ у повітрі;
- сумісність інтерфейсів підключення датчиків з мікроконтролером (наприклад, I²C, SPI, UART).

– ***Контролер (пристрій керування).***

Це центральний елемент системи, який виконує функції збирання, обробки та передачі даних, а також забезпечує підключення до internet чи керування іншими пристроями.

Вимоги до контролера:

- низька вартість;
- простота у використанні;
- наявність значної кількості бібліотек;
- достатня обчислювальна потужність і об'єм пам'яті для реалізації запланованих функцій метеостанції.

– ***Пристрої виводу та індикації.***

Ці компоненти спрощують взаємодію користувача з системою. До них відносять дисплеї (наприклад, OLED, LCD або TFT), світлодіоди тощо.

Основні вимоги:

- можливість відображення даних у реальному часі;
- низьке енергоспоживання;
- підтримка необхідних інтерфейсів мікроконтролера;

- доступність бібліотек і прикладів використання.
- **Wi-Fi-модуль.**

Забезпечує підключення системи до internet для відправлення та отримання даних.

Основні вимоги до модуля:

- підтримка частоти 2.4 ГГц s 5 ГГц;
- низьке енергоспоживання;
- наявність драйверів, бібліотек та документації для налаштування.

Загальні вимоги до апаратної частини системи:

- **Компактність** — усі елементи мають легко інтегруватися в компактний корпус.
- **Доступність і дешевизна** — варто використовувати недорогі, але надійні компоненти.
- **Простота налаштування** — мінімальні вимоги до початкової конфігурації, зручність розгортання навіть для користувача з базовими технічними знаннями.

1.4.2 Вимоги до програмної частини

До програмної частини проєкту належать: мова програмування, середовище розробки, способи зберігання та обробки даних, методи відображення інформації, а також налаштування веб-сервера.

– Мова програмування.

Слід обрати мову, з якою вже є досвід роботи і яка є ефективною для мікроконтролерів з обмеженими ресурсами. Така мова повинна дозволяти максимально ефективно використовувати пам'ять і такти процесора, мати активну **спільноту розробників**, наявність готових прикладів, бібліотек і фреймворків.

– Середовище розробки.

Воно має підтримувати обрану мову програмування та відповідати таким критеріям:

- інтуїтивно зрозумілий інтерфейс;
- підтримка обраної мікроплати;
- зручне керування бібліотеками;
- наявність засобів налагодження, перевірки коду та пошуку помилок.

– ***Способи зберігання та обробки даних.***

Згідно з обраною архітектурою, зберігання даних здійснюватиметься у вбудованій пам'яті мікроконтролера, з урахуванням її обмеженого обсягу.

Обробка даних — це процес перетворення "сирих" вимірювань з датчиків у готові значення, придатні для відображення у веб-інтерфейсі.

– ***Відображення інформації.***

Дані відображатимуться двома способами:

- на дисплеї — користувач зможе бачити поточні значення температури, вологості тощо без підключення до інтернету;
- на веб-сервері — графічна візуалізація даних усіх датчиків у вигляді інтерактивних графіків, доступна через браузер.

– ***Веб-сервер.***

Його головне завдання — відображення веб-сторінки з поточними показниками сенсорів і періодичне оновлення графіків у реальному часі.

1.4.3 Функціональні можливості системи

Спроектована система буде володіти такими функціональними можливостями:

- **збір і обробка даних** з датчиків температури, вологості, атмосферного тиску та вмісту CO₂;
- **відображення поточних значень** температури, вологості та тиску на вбудованому дисплеї;
- **збереження історії вимірювань** у пам'яті мікроконтролера та **відображення її на веб-сторінці** у вигляді трьох графіків;
- **індикація рівня CO₂** за допомогою зміни кольору RGB-світлодіода (зелене — низький рівень, червоне — високий);

- **автономна робота** від власного джерела живлення;
- **можливість підзарядки акумулятора**, що підвищує незалежність і мобільність системи.

2. ПРОЄКТНА ЧАСТИНА

В даному розділі кваліфікаційної роботи, на основі сформульованих раніше вимог до системи та її функціональних можливостей, здійснено вибір основних апаратних компонентів. Визначено доцільність використання вибраного мікроконтролера, підібрано відповідні датчики, дисплей для відображення даних, а також джерело живлення, що забезпечить стабільну автономну роботу всієї системи.

Проведено розрахунок максимально можливого енергоспоживання, враховуючи особливості підключених модулів та передбачуваний режим роботи. Обґрунтовано вибір мови програмування та середовища розробки, що дозволяють ефективно реалізувати програмну частину проєкту з урахуванням ресурсів мікроконтролера.

Визначено оптимальний підхід до зберігання та візуалізації даних на веб-сервері, враховуючи вимоги до доступності, швидкості оновлення інформації та обсягу доступної пам'яті. Усі текстові описи доповнено графічними схемами, таблицями та діаграмами, що візуалізують структуру системи, її функціонування та компонування.

Також наведено перелік базових знань і навичок, необхідних для реалізації проєкту: від роботи з мікроконтролерами й цифровими сенсорами до веб-програмування та роботи з електронними компонентами.

Надано текстовий опис зовнішнього вигляду пристрою, а також детально пояснено принцип роботи всієї метеосистеми — від моменту фіксації даних сенсорами до виведення їх на дисплей чи у веб-браузері.

2.1. Вибір компонентів апаратної частини

Під час відбору апаратних компонентів необхідно врахувати вимоги до функціональності, енергоефективності та компактності системи, визначені на попередніх етапах проєктування. Апаратна частина системи охоплює ключові елементи, необхідні для її стабільної та автономної роботи.

До складу апаратної частини входять:

- мікроконтролер, який виступає як центральний керуючий елемент;

- датчики, що здійснюють вимірювання температури, вологості, тиску, рівня CO₂ тощо;
- дисплей, призначений для виведення актуальних даних у зручному для користувача вигляді;
- RGB-світлодіод, що забезпечує візуальну індикацію рівня вуглекислого газу або інших критичних параметрів;
- Wi-Fi-модуль, який дозволяє системі передавати дані та забезпечує інтеграцію з веб-інтерфейсом;
- система живлення, яка складається з джерела живлення та зарядного модуля, що забезпечують енергетичну автономність пристрою.

Усі компоненти слід вибирати з урахуванням енергоспоживання, сумісності між собою, можливостей розширення та простоти інтеграції у загальну конструкцію. Це дозволить створити технічно збалансовану платформу для подальшої реалізації метеостанції [6, 8].

2.1.1 Вибір мікроконтролера

Вибір мікроконтролера є одним із ключових рішень при розробці будь-якої електронної системи, зокрема метеостанції. Саме мікроконтролер відповідає за збір, обробку та передавання даних від датчиків, керування дисплеєм, зв'язок із мережею та іншими модулями. Від його обчислювальної потужності, об'єму пам'яті та енергоефективності залежить стабільність і функціональність усієї системи. Крім того, важливим є наявність підтримки бездротових інтерфейсів (Wi-Fi, Bluetooth), а також широка підтримка спільноти та бібліотек, що спрощує розробку програмного забезпечення.

При розгляді апаратної платформи для реалізації системи збору метеоданих було проаналізовано декілька популярних мікроконтролерів та одноплатних комп'ютерів, кожен з яких має свої особливості, переваги та обмеження.

В таблиці 2.1 представлено порівняльні характеристики популярних мікроконтролерів [3], котрі застосовуються для реалізації подібних проєктів.

Таблиця 2.1 - Порівняльні характеристики популярних мікроконтролерів

Плата	Процесор	Частота CPU	ОЗП (RAM)	Wi-Fi	Bluetooth	Ethernet	USB-порти	Орієнтовна ціна (USD)
Arduino Uno	ATmega328P (8-bit AVR)	16 MHz	2 KB SRAM	✗	✗	✗	1 × USB-B	≈ 25–30 \$
Arduino Mega	ATmega2560 (8-bit AVR)	16 MHz	8 KB SRAM	✗	✗	✗	1 × USB-B	≈ 40 \$
Raspberry Pi 3	Broadcom BCM2837 (ARMv7, 4-ядерний)	1.2 GHz	1 GB	✓ (2.4 GHz)	✓ Bluetooth 4.1	✓ 100 Mbps	4 × USB 2.0	≈ 35 \$
STM32	STM32F407 (або аналог, Cortex-M4)	до 180 MHz	до 192 KB SRAM	✗*	✗*	✗*	1 × Micro-USB	≈ 15–25 \$
Nucleo	(STM32F4xx/Nucleo-серія)	≈ 72–180 MHz	≈ 64–192 KB SRAM	✗*	✗*	✗*	1 × Micro-USB	≈ 15–25 \$
ESP32	Xtensa LX6 (2 ядра, 32-bit)	160–240 MHz	520 KB SRAM	✓ 802.11 b/g/n	✓ Bluetooth 4.2	✗ (тільки MAC)	1 × USB-to-Serial	≈ 6–12 \$

*Ethernet і Wi-Fi можуть бути додані за допомогою додаткових модулів, тоді як USB у STM32/Nucleo використовується зазвичай лише для програмування й комунікації.

Arduino Uno — одна з найпоширеніших платформ для початківців. Це 8-бітна плата з простим у використанні середовищем розробки та великою кількістю прикладів і бібліотек. Її перевагами є простота підключення, надійність, стабільна робота. Проте вона має обмежену пам'ять, невисоку тактову частоту та не підтримує бездротові інтерфейси без зовнішніх модулів.

Arduino Mega — це розширена версія Uno, яка має значно більше виводів та більший об'єм оперативної пам'яті. Вона підходить для проєктів із великою кількістю підключень, але так само не має вбудованих засобів зв'язку (Wi-Fi або Bluetooth), що обмежує її можливості в автономних бездротових рішеннях.

Raspberry Pi 3 — повноцінний одноплатний комп'ютер з операційною системою Linux. Має високі обчислювальні характеристики, підтримку Wi-Fi, Bluetooth і Ethernet. Це ідеальне рішення для складних систем, які потребують графічного інтерфейсу, обробки великих обсягів даних або роботи з хмарними

сервісами. Проте Raspberry Pi споживає більше енергії, ніж класичні мікроконтролери, і потребує окремої операційної системи для запуску.

STM32 — серія 32-бітних мікроконтролерів із архітектурою ARM Cortex-M, які вирізняються високою продуктивністю та гнучкістю. Їх часто застосовують у промислових та реального часу системах. STM32 має значно більше ресурсів, ніж Arduino, але вимагає глибших знань програмування на мові C, а також окремих засобів розробки. Вбудованих засобів зв'язку зазвичай немає.

Nucleo — це апаратна платформа, розроблена компанією STMicroelectronics на основі STM32. Вона має доступну ціну, сумісність із бібліотеками STM32Cube, достатню продуктивність для багатьох завдань, але, як і звичайний STM32, потребує зовнішніх модулів для бездротового зв'язку та спеціалізованих навичок програмування.

ESP32 — двоядерний 32-бітний мікроконтролер з архітектурою Xtensa, що має на борту Wi-Fi і Bluetooth. Він ідеально підходить для IoT-проектів завдяки поєднанню компактності, низького енергоспоживання, високої тактової частоти (до 240 МГц), великого обсягу оперативної пам'яті та підтримки сучасних протоколів бездротової передачі даних.

Таким чином, проаналізувавши вищенаведені мікроконтролери і одноплатні ПК ми дійшли до висновку, що ESP32 забезпечує оптимальне поєднання функціональності, компактності та ціни, що робить його найкращим вибором для створення сучасної метеостанції з підтримкою бездротового зв'язку та автономного доступу до даних.

ESP32 — це компактний, високопродуктивний, двоядерний мікроконтролер з вбудованою підтримкою Wi-Fi та Bluetooth, розроблений компанією Espressif Systems. Вперше представлений у 2016 році, ESP32 швидко здобув популярність серед розробників завдяки поєднанню високої обчислювальної потужності, енергоефективності та багатофункціональності в одному рішенні. Його часто використовують у пристроях Інтернету речей (IoT), системах автоматизації, смарт-пристроях та телеметрії.

Основні характеристики ESP32:

- Модуль: ESP32-WROOM, що базується на чипі ESP32-D0WDQ6;

- Процесор: 2-ядерний 32-бітний Tensilica Xtensa LX6;
- Тактова частота: до 240 МГц;
- Пам'ять:
 - Внутрішня Flash: 448 КБ ROM;
 - SRAM: 520 КБ;
 - Зовнішня Flash: 4 МБ;
- Мережеві інтерфейси:
 - Wi-Fi 2,4 ГГц (IEEE 802.11 b/g/n);
 - Bluetooth 4.2 (BLE + BR/EDR);
- GPIO (виводи загального призначення): до 25;
- Інші інтерфейси:
 - 15 аналогових входів (ADC, 12 біт);
 - 2 цифрових виходи (DAC, 8 біт);
 - 21 канал PWM (ШИМ), 16 біт;
 - 8 ємнісних сенсорних каналів;
 - $3 \times \text{SPI}$, $3 \times \text{UART}$, $2 \times \text{I}^2\text{C}$, $2 \times \text{I}^2\text{S}$;
- Підтримка апаратних переривань: до 25 пінів;
- Живлення:
 - Напруга логіки: 3,3 В;
 - Вхідна напруга (через пін Vin): 5–14 В;
 - Максимальний струм із GPIO: 12 мА;
 - Струм на виході 3V3: до 1 А;
- Додаткові функції:
 - Підтримка багатозадачності;
 - Апаратне прискорення для складних обчислень;
 - Вбудований датчик Холла та сенсор температури.

Конструктивні особливості плати ESP32 DevKit:

- ✓ USB-UART перетворювач на чипі CP2102 забезпечує зручне підключення до комп'ютера через мікро-USB, а також створення віртуального COM-порту;

- ✓ Кнопка EN — виконує функцію скидання (аналог RESET);
- ✓ Кнопка BOOT — переводить плату в режим прошивки вручну;
- ✓ Регулятор напруги AMS1117-3.3 — лінійний стабілізатор, який знижує напругу живлення до 3,3 В з максимальною силою струму до 1 А, достатньою для роботи всієї системи.

ESP32 — це оптимальний вибір для розробки метеостанції завдяки поєднанню бездротових інтерфейсів, високої продуктивності, компактності та великої кількості доступних бібліотек (рисунк 2.1). Він дозволяє реалізувати як збір і обробку даних, так і створення локального веб-сервера без потреби у зовнішньому комп'ютері чи мережевій інфраструктурі.

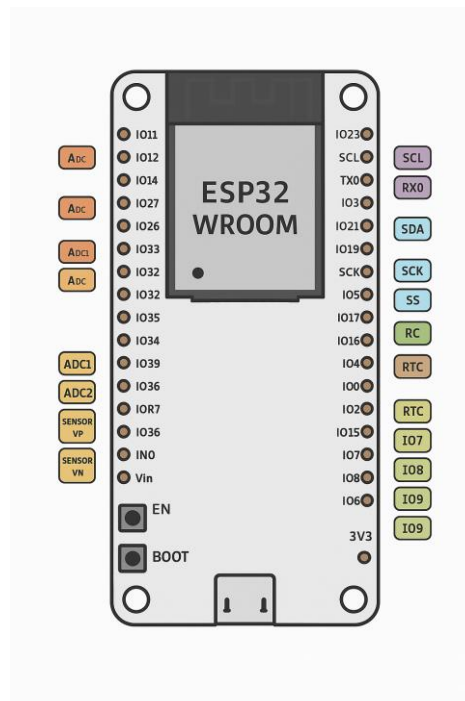


Рис. 2.1. Схема ESP32

2.1.2 Вибір датчиків

Датчики є ключовими елементами системи збору метеоданих, оскільки саме вони забезпечують перетворення фізичних величин на цифрову інформацію, яку можна аналізувати й відображати [9]. Від точності, стабільності та надійності обраних сенсорів безпосередньо залежить якість усієї системи та достовірність отриманих результатів. В таблиці 2.1 наведено порівняльні характеристики популярних датчиків.

Таблиця 2.1 - Порівняльні характеристики датчиків

Датчик	Що вимірює	Діапазон / точність	Інтерфейс	Живлення	Ціна (USD)	Переваги
BME280	Температура, тиск, вологість	-40...+85 °C, ±1 °C; 300–1100 hPa, ±1 hPa; 0–100 % RH, ±3 % RH	I ² C/SPI	1.8–3.6 V	≈ 14–15 \$	Компактний, точний, багатофункційний, низьке споживання
BMP280	Температура, тиск	-40...+85 °C, ±1 °C; 300–1100 hPa, ±1 hPa	I ² C/SPI	~1.8–3.6 V	≈ 9–12 \$	Надійний, дешевий, висока точність тиску
DHT22	Температура, вологість	0...+80 °C, ±0.5 °C; 0–100 % RH, ±2–5 % RH	1-wire цифровий	3–5 V	≈ 5 \$	Дешевий, готовий модуль, простий
SHT3x (SHT30/31)	Температура, вологість	-40...+125 °C, ±0.3 °C; 0–100 % RH, ±2 % RH	I ² C	2.4–5.5 V	≈ 10–12 \$	Висока точність, швидкий
MQ-135	CO ₂ , аміак, альдегід, бензол та ін.	10–1000 ppm (газ), потребує попереднього нагріву	Analog + цифровий	5 V / ~150 mA	≈ 7–11 \$	Широкий спектр детекції газів, налаштовуване порогове значення

Для розробки системи збору метеоданих було обрано комбінацію датчиків, що забезпечують оптимальний баланс між функціональністю, точністю вимірювань та енергоефективністю. Основним датчиком температури, вологості та атмосферного тиску став BME280, який поєднує в собі три сенсори в одному корпусі та забезпечує високу точність при низькому енергоспоживанні.

Для оцінки якості повітря до системи додано датчик MQ-135, який дозволяє визначати наявність шкідливих газів, зокрема CO₂, аміаку та органічних сполук. Така комбінація дозволяє не лише фіксувати погодні параметри, а й контролювати мікроклімат у приміщенні, що особливо актуально для розумних систем вентиляції й кондиціонування [9, 10].

Таким чином, вибрані датчики повністю відповідають технічним вимогам і забезпечують достатню точність для побутового та дослідницького використання.

2.1.3 Вибір дисплея

У системах збору метеоданих можуть застосовуватись різні типи дисплеїв, кожен з яких має свої технічні особливості, переваги та обмеження.

Рідкокристалічні дисплеї (LCD) є одним із найпоширеніших рішень завдяки своїй доступності, низькому енергоспоживанню та простоті використання. Вони працюють на основі рідких кристалів, що змінюють свою прозорість під дією електричного струму, створюючи таким чином зображення. Однак, LCD-дисплеї потребують підсвічування, що може знижувати контрастність при яскравому освітленні.

OLED-дисплеї (органічні світлодіодні) вирізняються високим рівнем контрасту, яскравими кольорами та здатністю відображати інформацію без підсвітки, завдяки самосвітним пікселям. Вони споживають мало енергії, особливо при частковому або монохромному виведенні даних. До недоліків можна віднести вищу вартість порівняно з LCD-аналогами та потенційно менший ресурс служби, зумовлений вигоранням пікселів при довготривалому статичному зображенні.

LED-дисплеї (на основі класичних світлодіодів) мають відмінну яскравість, добре читаються навіть під прямим сонячним світлом, характеризуються довговічністю та ефективністю. Проте такі дисплеї можуть бути дорогими, мати підвищене енергоспоживання та обмежений кут огляду.

Вибір конкретного дисплея залежить від поставлених технічних вимог, умов експлуатації, бюджету проекту та очікуваної зручності для користувача.

Враховуючи специфіку розроблюваної системи, було визначено такі ключові критерії:

- наявність підтримуваного інтерфейсу підключення до мікроконтролера (I²C);
- низьке енергоспоживання;
- наявність бібліотек для роботи з ESP32;
- невелика, але достатня роздільна здатність для відображення основних метеопоказників (температура, вологість, тиск);
- невисока вартість.

З огляду на ці вимоги, було обрано OLED-дисплей діагоналлю 0,91 дюйма, який ідеально підходить для компактних рішень. Він забезпечує чітке відображення інформації навіть при низькому освітленні та не потребує підсвітки, що значно знижує споживання енергії в автономному режимі.

2.1.4 Вибір джерела живлення

Перед вибором джерела живлення для системи збору метеоданих необхідно здійснити розрахунок загального енергоспоживання всіх компонентів. Це дозволяє підібрати джерело, яке забезпечить стабільну й тривалу автономну роботу пристрою.

Для початку розглянемо орієнтовне споживання струму кожного з основних елементів системи:

- Мікроконтролер ESP32 у режимі роботи з Wi-Fi-модулем споживає від 70 до 200 мА, залежно від режиму передачі даних та потужності сигналу.
- Газовий датчик MQ-135 споживає близько 150 мкА в режимі очікування та до 50 мА в активному режимі.
- OLED-дисплей на базі SSD1306 може споживати до 25 мА при активному відображенні інформації (залежить від яскравості та контрасту).
- Датчик BME280 має низьке енергоспоживання: до 1 мА в режимі очікування та до 3 мА під час активного вимірювання.
- Індикаторний світлодіод може споживати від 5 до 20 мА, залежно від кольору та яскравості.

Підсумовуючи максимальні значення, отримаємо загальне пікове навантаження близько 300 мА:

$$200 \text{ мА (ESP32)} + 50 \text{ мА (MQ-135)} + 25 \text{ мА (OLED)} + 3 \text{ мА (BME280)} + 20 \text{ мА (світлодіод)} = 298 \text{ мА}$$

Це значення є орієнтовною оцінкою пікового споживання, тобто в момент, коли всі компоненти працюють одночасно в активному режимі.

Можливі варіанти живлення системи:

- Звичайна батарея – доступне, просте рішення, але має обмежений ресурс і не підходить для тривалого використання без заміни.

- Адаптер живлення – забезпечує стабільну подачу струму, однак порушує ключову вимогу про автономність пристрою, тому не є придатним у цьому проєкті.
- Сонячна панель – цікаве автономне рішення, але потребує додаткової електроніки, є дорожчим, а також залежить від погодних умов.

Обране рішення:

Було обрано літій-іонний акумулятор US18650VTC6 виробництва Sony.

Його характеристики:

- Номінальна напруга: 3,6 В
- Ємність: 3120 мА·год
- Максимальний струм розряду: до 30 А

Згідно з формулою:

Час автономної роботи (год) = ємність акумулятора (мА·год) / споживання (мА) отримуємо:

$3120 \text{ мА} \cdot \text{год} / 300 \text{ мА} \approx 10$ годин автономної роботи без підзарядки.

Для продовження часу автономної роботи систему можна доповнити модулем зарядки TP4056, який дозволяє заряджати акумулятор через USB або сонячну панель, не розбираючи пристрій.

2.2 Вибір компонентів програмної частини

Вибір компонентів програмної частини має ключове значення для стабільної, ефективної та зручної роботи всієї системи збору метеоданих. Саме мова програмування, середовище розробки, тип веб-сервера та спосіб зберігання інформації визначають, наскільки швидко система буде реагувати на події, як легко її буде масштабувати, модернізувати чи обслуговувати [11]. Правильне програмне середовище дозволяє мінімізувати помилки, спростити розробку та прискорити впровадження нових функцій. Крім того, добре підібрані інструменти забезпечують кращу сумісність з обраними мікроконтролерами, датчиками та інтерфейсами, що напряму впливає на надійність системи загалом [12].

2.2.1 Вибір мови програмування

Мова програмування є фундаментальною основою будь-якої вбудованої системи, оскільки саме вона визначає стиль взаємодії з апаратною частиною, ефективність виконання коду та зручність розробки. У межах проєкту метеостанції було обрано мову C++, яка є найпоширенішою для мікроконтролерів, зокрема Arduino-сумісних плат, включаючи ESP32. C++ дозволяє створювати як прості скетчі, так і масштабовані об'єктно-орієнтовані проєкти з гнучкою структурою.

Крім того, C++ забезпечує прямий доступ до регістрів мікроконтролера, що дозволяє оптимізувати споживання енергії та підвищити швидкодію системи. Завдяки відкритому коду численних бібліотек (наприклад, для роботи з датчиками BME280, MQ135 або OLED-дисплеями), розробник може глибше зрозуміти роботу кожного модуля й адаптувати його під конкретні потреби [7]. Популярність мови гарантує наявність великої кількості навчальних матеріалів, активних форумів та підтримки з боку світової спільноти розробників.

2.2.2 Вибір середовища розробки

Вибір середовища розробки (IDE) значно впливає на комфорт розробника, ефективність тестування, швидкість налагодження та зручність керування проєктом. Основним середовищем, обраним для реалізації системи метеостанції, є Arduino IDE — класична платформа для роботи з мікроконтролерами, що має мінімальний поріг входу, підтримку численних плат і бібліотек, а також інтуїтивно зрозумілий інтерфейс.

Arduino IDE дозволяє швидко підключати зовнішні бібліотеки, створювати прості та надійні програми, а також проводити налагодження за допомогою послідовного монітору. З огляду на те, що ESP32 підтримується офіційно через розширення esp32-core, робота з нею у цьому середовищі не викликає труднощів навіть у початківців.

Однак для більш досвідчених користувачів існують розширені альтернативи:

- PlatformIO — потужна багатоплатформна екосистема, яка інтегрується з Visual Studio Code і пропонує розширене керування проєктами, бібліотеками, середовищами та налаштуваннями компіляції. Підтримує автозбірку, CI/CD, unit-тести та роботу з git.
- Eclipse + ESP-IDF — підходить для проєктів, де потрібно повний контроль над усією системою та низькорівневими функціями. Вимагає більше часу для конфігурації, але відкриває ширші можливості.

Залежно від складності завдань і досвіду розробника, кожне середовище може бути корисним, однак для навчальних і середніх проєктів Arduino IDE залишається найбільш зручним та практичним рішенням.

2.2.3 Вибір веб-сервера та способу зберігання інформації

Завдяки можливостям ESP32 розробник має змогу реалізувати повноцінний веб-сервер без зовнішньої інфраструктури, що значно спрощує архітектуру системи та зменшує витрати. Найкращим вибором для таких цілей є використання бібліотеки ESPAsyncWebServer, яка дозволяє створювати асинхронний сервер — тобто такий, що може обслуговувати кілька клієнтів одночасно без блокування головного циклу програми.

Асинхронна модель особливо ефективна у пристроях реального часу, де критично важливо уникнути затримок під час обробки даних з сенсорів або оновлення дисплея. Веб-інтерфейс, створений на базі ESPAsyncWebServer, може відображати не лише текстові значення, але й графіки зміни метеопараметрів, таблиці, лог-файли, повідомлення про події тощо.

Щодо зберігання даних, система використовує вбудовану пам'ять ESP32, що дозволяє уникнути застосування зовнішніх SD-карт і зменшує складність проєкту. У пам'яті можна організувати структуру зберігання у вигляді JSON-об'єктів, масивів або простих текстових файлів, використовуючи файлові системи LittleFS або SPIFFS. Такий підхід є ефективним, енергоощадним та економним рішенням, особливо коли система не потребує тривалого архівування великих обсягів інформації [13, 14].

Крім того, ESP32 дозволяє створити простий REST API, що відкриває перспективи для віддаленого моніторингу, інтеграції з мобільними додатками чи іншими IoT-сервісами в майбутньому.

2.3 Конструктивні особливості корпусу

Корпус для системи збору метеоданих виконує не лише естетичну, але й функціональну роль — він має забезпечити надійний захист електронних компонентів, доступ до роз'ємів і портів, а також зручність у використанні та обслуговуванні. Розмір і форма корпусу мають відповідати геометрії встановлених компонентів, при цьому всі елементи повинні бути розташовані щільно, логічно й акуратно.

На рисунку 2.2 представлено 3D-модель корпусу системи. Конструкція складається з двох акрилових пластин товщиною 3 мм:

- 1 – нижня пластина (основа) розміром 11×13 см;
- 2 – верхня пластина розміром 8×11 см;
- 3 – акумуляторна батарея;
- 4 – макетна плата;
- 5 – відсік для акумулятора.

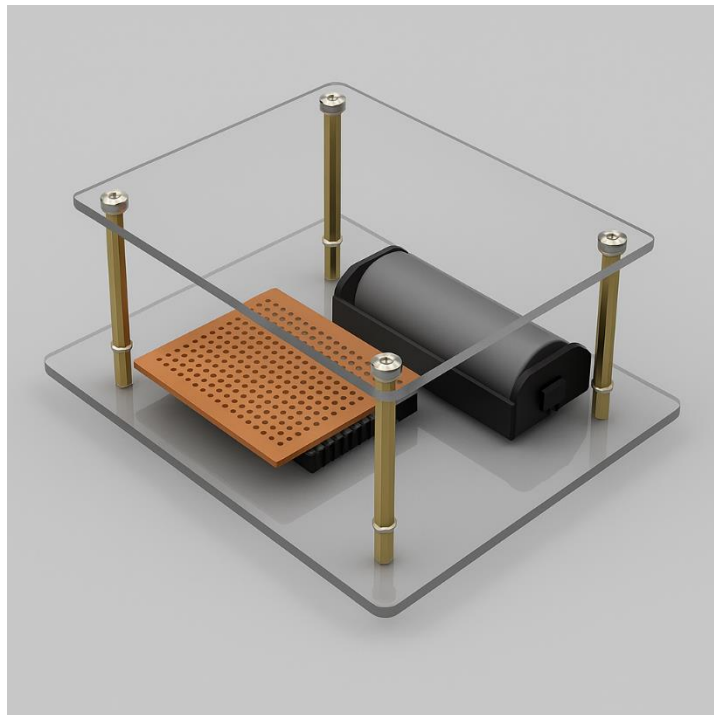


Рисунок 2.2 – 3D модель корпусу системи моніторингу метеоданих

Кожна пластина має по чотири отвори М3, розміщених у кутах з відступом 1 см від країв, що дозволяє зручно з'єднати всі елементи за допомогою гвинтів.

Процес складання корпусу:

1. На нижню пластину за допомогою чотирьох гвинтів М3×30 мм закріплюється макетна плата зі встановленими штифтовими роз'ємами та розпаяними доріжками під інтерфейс І²С, живлення тощо.

2. У вільному просторі фіксується відсік для акумулятора (клеєм або подвійним скотчем).

3. Верхня пластина прикручується до гвинтів за допомогою чотирьох гайок М3, після чого конструкція перевіряється на надійність і стійкість.

Такий корпус забезпечує вентиляцію, видимість елементів (завдяки прозорому матеріалу), а також захист від механічних пошкоджень, пилу та випадкових замикань.

2.4 План реалізації проєкту

Для реалізації проєкту було розроблено чіткий покроковий план, що охоплює всі основні етапи — від підбору обладнання до впровадження системи:

1. Вибір обладнання — визначити необхідні компоненти: мікроконтролер ESP32, датчики BME280, MQ-135, OLED-дисплей, акумулятор, зарядний модуль тощо, придбати все необхідне.

2. Підготовка середовища розробки — встановити Arduino IDE, додати бібліотеки для роботи з ESP32 та відповідними модулями.

3. Написання коду — розробити програмне забезпечення для зчитування даних із датчиків, передачі інформації на дисплей і веб-сервер, обробки даних, реалізації інтерфейсу керування.

4. Збірка та тестування — під'єднати всі компоненти згідно зі схемою, перевірити правильність роботи кожного модуля; у разі виявлення проблем — локалізувати і усунути.

5. Завантаження прошивки — прошити ESP32 створеним кодом через Arduino IDE.

6. Тестування та налагодження – перевірити, як система реагує на зміну показників, протестувати обробку, зберігання та відображення даних.
7. Деплой веб-сервера – розгорнути веб-сервер безпосередньо на ESP32, протестувати відображення графіків, оновлення даних та загальну стабільність роботи.
8. Розміщення у корпусі – акуратно змонтувати всі компоненти всередині корпусу, перевірити фіксацію та доступ до роз'ємів.
9. Використання – запустити систему у реальних умовах, моніторити температуру, вологість, тиск та якість повітря в реальному часі.
10. Обслуговування – час від часу перевіряти стан компонентів, заряд акумулятора, при потребі замінювати чи перепрошивати пристрій.

3. СПЕЦІАЛЬНА ЧАСТИНА

3.1 Підключення компонентів розробленої системи

Перед тим як здійснювати підключення датчиків, дисплея та інших периферійних модулів до плати, важливо ознайомитися з розташуванням та функціоналом виводів плати ESP32 DevKit v1. Схематичне зображення розпіновки плати наведено на рисунку 3.1.

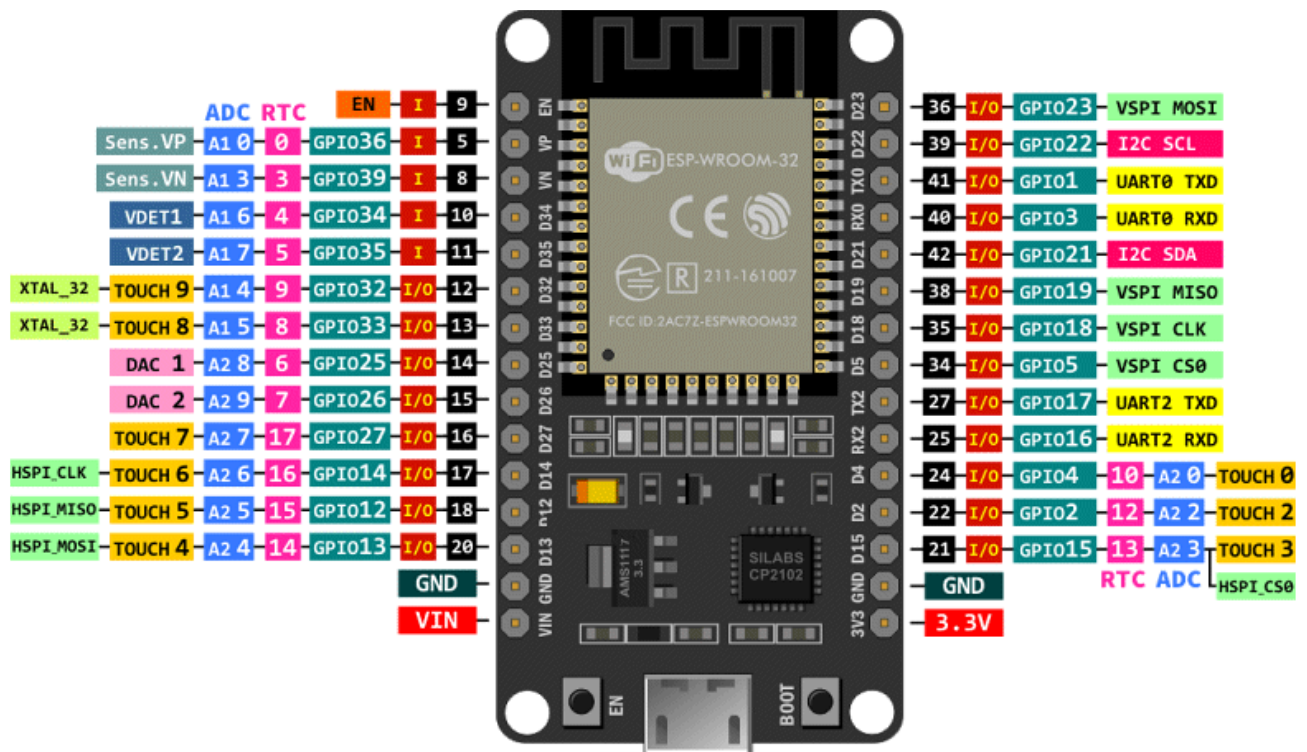


Рисунок 3.1 – Схема виходів плати ESP32 DevKit v1

Опис основних контактів ESP32 DevKit v1

- GND — загальний провід (земля), необхідний для замикання електричних кіл.
- VIN — вхід для зовнішнього джерела живлення (5 В). Вбудований стабілізатор AMS1117-3.3 дозволяє подавати напругу в діапазоні від 5 до 10 В, хоча теоретично витримує до 15 В. Втім, за високих напруг може виникати перегрів, якщо відсутнє додаткове охолодження.
- 3.3V — стабілізоване живлення 3,3 В від внутрішнього стабілізатора. Використовується для живлення зовнішніх модулів, які підтримують роботу на такій напрузі.

- GPIO (General Purpose Input/Output) — виводи загального призначення. Можуть конфігуруватися як входи, виходи або комбіновані I/O, та програмно призначатися під будь-який інтерфейс або функцію.
- EN (Enable) — дозволяє вивести плату з режиму сну або перезавантажити її (аналог Reset).
- ADC (Analog to Digital Converter) — аналогово-цифрові перетворювачі. ESP32 має вбудований АЦП з роздільністю 12 біт (значення в діапазоні 0–4095). Напруга на вході не повинна перевищувати 3.3 В. Роздільність і діапазон можна змінити програмно.
- DAC (Digital to Analog Converter) — цифро-аналогові перетворювачі. Два 8-бітних DAC-канали дозволяють виводити аналоговий сигнал (напругу), згенерований із цифрових значень.
- UART (Universal Asynchronous Receiver-Transmitter) — асинхронний послідовний інтерфейс для обміну даними з іншими пристроями (датчиками, GPS-модулями тощо). Виводи UART можуть бути переназначені на будь-який GPIO.
- SPI (Serial Peripheral Interface) — синхронний інтерфейс високошвидкісного зв'язку. ESP32 підтримує два SPI (HSPI і VSPI), що можуть працювати як у режимі ведучого, так і підлеглого.
- I2C (Inter-Integrated Circuit) — популярна двопровідна послідовна шина для підключення численних периферійних пристроїв (датчики, дисплеї тощо). В ESP32 шини SDA і SCL можна переназначати програмно.
- TOUCH — виводи для ємнісних сенсорів. Реагують на зміну ємності, спричинену дотиком пальця.
- RTC GPIO — спеціальні виводи з підтримкою режиму наднизького енергоспоживання (ULP – Ultra Low Power). Можуть використовуватися для виведення з глибокого сну або контролю з боку співпроцесора, який працює навіть тоді, коли основне ядро відключене.
- VDET — аналогові виводи, пов'язані з RTC-ядром. Дають змогу виводити контролер із режимів енергозбереження на основі напруги на вході.

- XTAL_32 — виводи для підключення зовнішнього кварцового резонатора на частоті 32.768 кГц, необхідного для точного обліку часу в енергозберігаючих режимах.
- Sens_VP / Sens_VN — входи вбудованого датчика Холла, який здатен виявляти зміни магнітного поля. Цей датчик інтегрований у сам мікроконтролер і не потребує додаткових підключень.

Загальні рекомендації перед підключенням:

1. Перевірити сумісність рівнів напруги — усі периферійні модулі мають працювати з логікою 3.3 В.
2. Під час підключення кількох пристроїв слід планувати розташування I2C, UART, SPI-пристроїв з урахуванням обмежень кількості шин або пінів.
3. Уникати підключення до GPIO, які вже використовуються під час завантаження прошивки (наприклад, GPIO0, GPIO2, GPIO15 — мають спеціальні функції при старті плати).
4. Використовувати резистори підтягування для шини I²C при необхідності.

3.1.1 Модуль заряджання TP4056 та джерело живлення.

Для забезпечення автономної роботи системи збору метеоданих використовується акумуляторна батарея типу Li-Ion, у поєднанні з модулем зарядки TP4056, який призначений для безпечного й ефективного керування процесом заряджання.

TP4056 — це спеціалізована мікросхема, призначена для зарядки однокоміркових літій-іонних акумуляторів. Вона підтримує вхідну напругу 5 В через micro-USB, максимальний струм зарядки — до 1 А, а ємність акумуляторів може варіюватися від 300 мА·год до 4000 мА·год і більше.

Ключові особливості TP4056:

- Автоматичне завершення циклу заряджання при досягненні напруги 4.2 В і зниженні струму до 1/10 від початкового значення;
- Вбудовані захисти від перегріву та перевантаження по струму;

➤ Інтеграція з індикаторами стану зарядки: світлодіод CHARGE сигналізує активний процес зарядки, а FULL – завершення процесу.

На платі модуля, окрім основного чипа TP4056, також присутні два додаткові компоненти:

- DW01 – мікросхема захисту від перезаряду, перерозряду, перевищення струму та короткого замикання;
- ML8205A – здвоєний MOSFET, що працює у парі з DW01 для апаратного розмикання ланцюга живлення у критичних ситуаціях.

На рисунку 3.2 зображено модуль TP4056, включаючи основні роз'єми:



Рисунок 3.2 – Модуль заряджання TP4056.

- B+ / B- — підключення до акумулятора (відповідно "+" і "-");
- OUT+ / OUT- — вихід живлення для навантаження (у нашому випадку – плата ESP32);
- +5V / GND — вхід для зовнішнього джерела живлення (через micro-USB або контакти);
- LED-індикатори — показують стан зарядки (зелене — заряд завершено, червоне — заряд триває).

На рисунку 3.3 наведено схему підключення живлення до плати ESP32. Вихідні контакти модуля TP4056 OUT+ та OUT- підключаються відповідно до пінів VIN і GND на ESP32 DevKit v1.

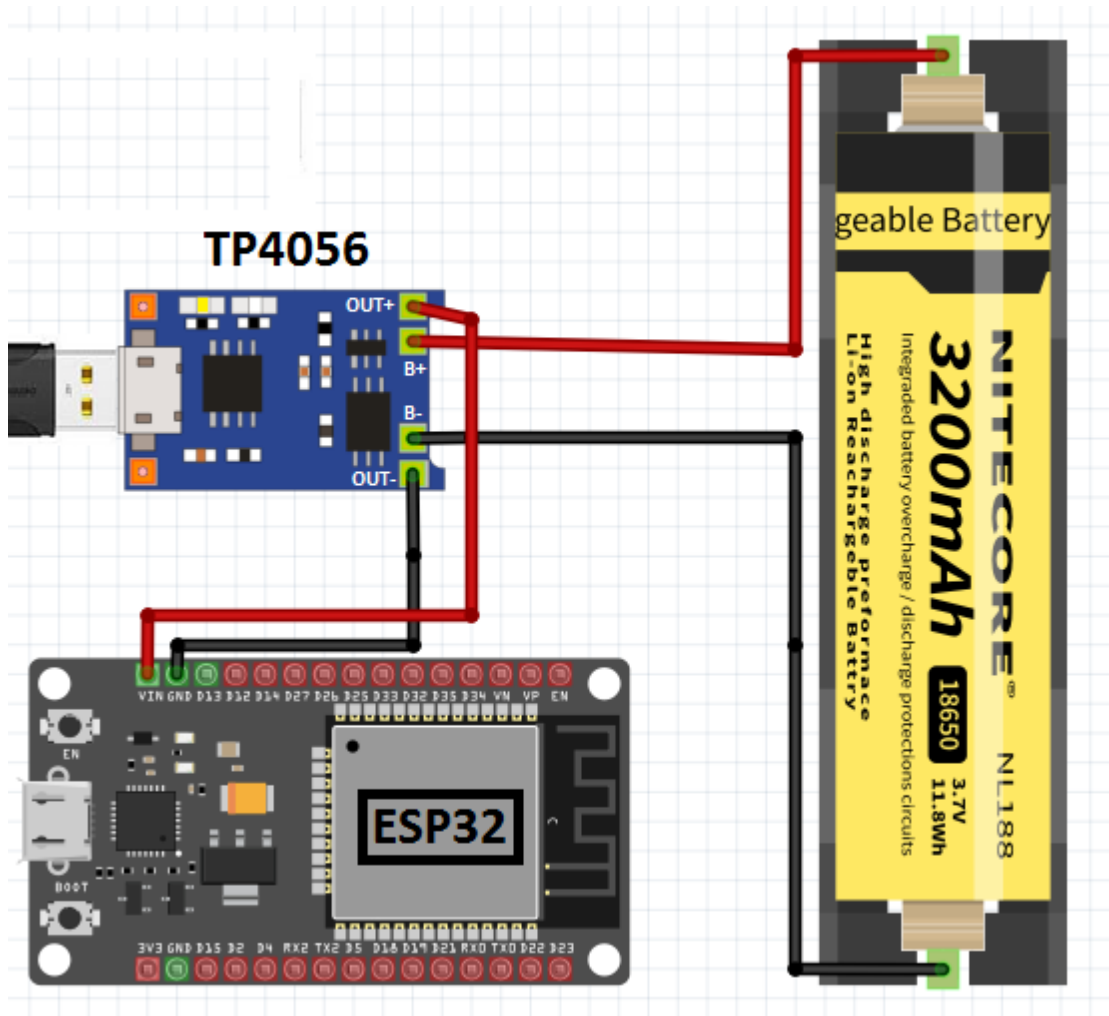


Рисунок 3.3 – Схема підключення системи живлення до плати ESP32.

Таке підключення дозволяє:

- жити плату безпосередньо від акумулятора;
- одночасно заряджати батарею й працювати з платою (режим UPS);
- забезпечити повну автономність пристрою.

3.1.2 Датчик BME280 та OLED-дисплей

У системі збору метеоданих для вимірювання температури, вологості та атмосферного тиску використовується багатофункціональний модуль з вбудованим сенсором BME280, що підключається через інтерфейс I²C. Фізично

модуль являє собою компактну друковану плату, на якій розміщено сам датчик і чотири виводи: VCC (живлення), GND (земля), SCL (тактовий сигнал) та SDA (лінія даних). Конструкція модуля зображена на рисунку 3.4.

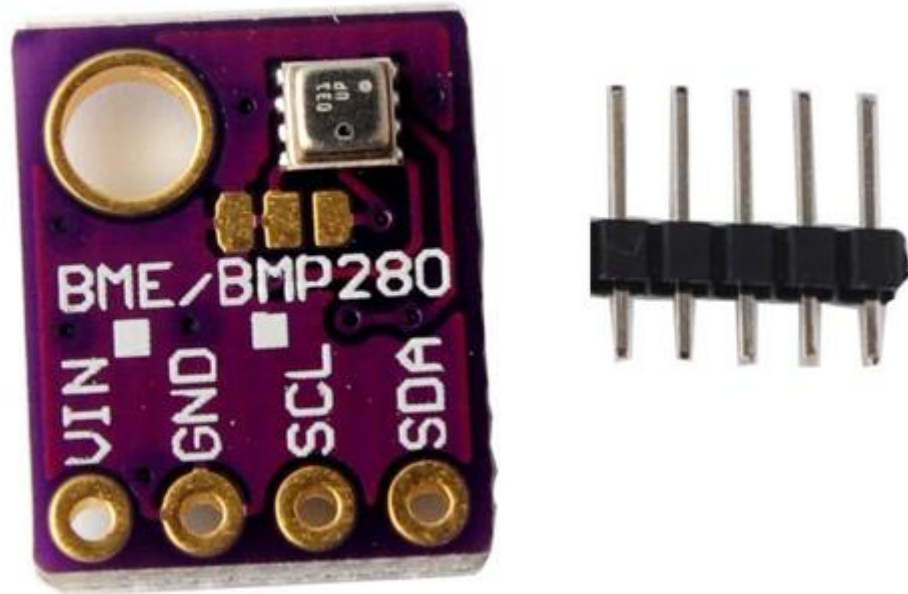


Рисунок 3.4 – Модуль BME280 .

Технічні характеристики BME280:

- Інтерфейси: I²C, SPI (вибір програмно);
- Робоча напруга: 3.3 В;
- Діапазон вимірювання температури: $-40 \dots +85$ °C (точність ± 0.01 °C);
- Діапазон вимірювання вологості: $0 \dots 100$ % RH (точність ± 3 %);
- Діапазон тиску: 300–1100 hPa (точність ± 0.01 hPa);
- Енергоспоживання:
 - у режимі вимірювання: ~ 2.74 нА;
 - у режимі очікування: ~ 0.1 нА.

Датчик має високу точність, стабільність та низьке енергоспоживання, що робить його ідеальним для автономних екологічних та погодних систем.

Для виведення зібраної інформації на екран у проєкті використовується OLED-дисплей з діагоналлю 0,91 дюйма та роздільною здатністю 128×32 пікселі, реалізований на основі драйвера SSD1306. Дисплей працює по шині I²C, що дозволяє з'єднувати його паралельно з іншими I²C-пристроями (наприклад, датчиком BME280) через ті самі лінії SDA і SCL. На рисунку 3.5 представлено зовнішній вигляд OLED-модуля.



Рисунок 3.5 – Модуль OLED .

Основні характеристики OLED-дисплея:

- Тип дисплея: монохромний OLED (білий текст);
- Інтерфейс: I²C;
- Живлення: 3.3 В;
- Роздільна здатність: 128×32 пікселі;
- Контролер: SSD1306;
- Енергоспоживання: низьке, змінне залежно від кількості

активних пікселів.

Підключення до ESP32

На рисунку 8 наведено схему підключення датчика BME280 та OLED-дисплея до плати ESP32. Обидва модулі з'єднані з мікроконтролером за допомогою 4-х ліній:

- Червоний провід – живлення (VCC, 3.3 В);
- Чорний провід – земля (GND);
- Зелений провід – SDA (дані, GPIO 21);
- Жовтий провід – SCL (тактова лінія, GPIO 22).

ESP32 підтримує гнучке призначення I²C-виводів, однак GPIO21 (SDA) та GPIO22 (SCL) є стандартними пін-контактами для шини I²C, рекомендованими у документації.

Таке підключення дозволяє обом модулям спільно працювати на одній шині без конфлікту, а використання бібліотек Adafruit для BME280 і SSD1306 спрощує ініціалізацію та вивід даних на дисплей.

Потрібно наочно продемонструвати спосіб підключення OLED-дисплея до плати ESP32. Це забезпечує чітке розуміння принципу роботи шини I²C та дозволяє уникнути помилок під час монтажу системи. Схема такого підключення наведена на рисунку 3.6. У ній відображено стандартні пін-контакти ESP32, до яких під'єднується дисплей: GPIO21 (лінія даних SDA) та GPIO22 (лінія тактування SCL), а також лінії живлення (3.3 V) і заземлення (GND). Такий спосіб підключення є універсальним і широко використовується в мікроконтролерних проєктах із виводом інформації на дисплей.

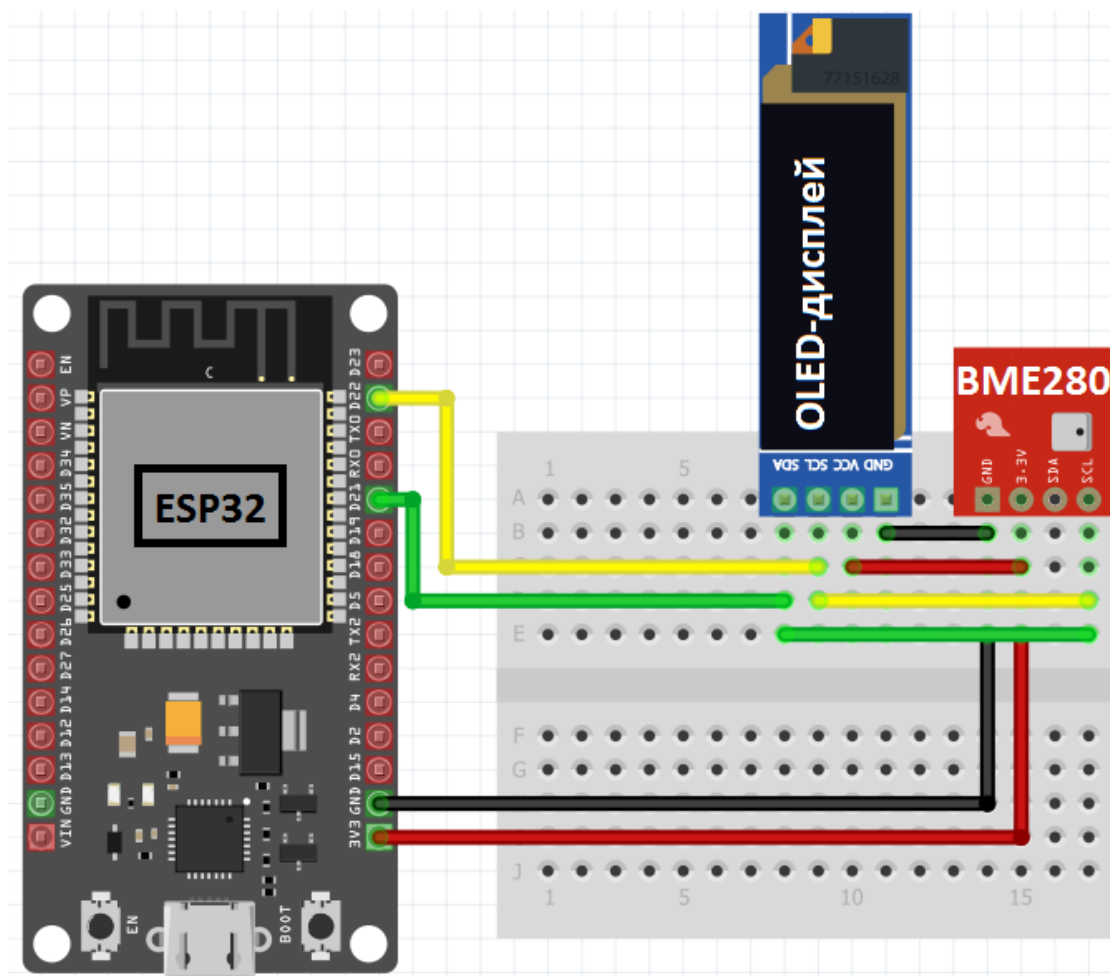


Рисунок 3.6 – Схема підключення датчика BME280 і OLED-дисплея до плати ESP32

На схемі зображено з'єднання плати ESP32 з датчиком BME280 за допомогою чотирьох провідників:

- червоний провід — лінія живлення (VCC, 3.3 В),
- чорний провід — загальний провід (GND),
- зелений провід — лінія даних SDA (GPIO21),
- жовтий провід — лінія тактових імпульсів SCL (GPIO22).

Виводи D21 (SDA) і D22 (SCL) на ESP32 відповідають стандартному інтерфейсу I²C, що дозволяє підключати одразу кілька пристроїв (наприклад, BME280 та OLED-дисплей) через одну і ту ж пару шин. Така конфігурація забезпечує стабільну передачу даних між датчиком і мікроконтролером.

3.1.3 Датчик MQ135 та RGB-світлодіод

У системі моніторингу якості повітря застосовується газовий сенсор MQ135, що здатний виявляти наявність ряду шкідливих речовин, зокрема аміаку, парів спирту, бензину та інших летких органічних сполук. Сенсор виконаний у вигляді компактного модуля з чотирма виводами:

- VCC — живлення;
- GND — земля;
- DO — цифровий вихід (0 або 1 від компаратора LM393);
- AO — аналоговий вихід (напруга, пропорційна концентрації газів).

Також на модулі присутній резистор для налаштування (потенціометр) для встановлення порогу спрацьовування компаратора, індикатор живлення (LED Power) та індикатор виходу компаратора (LED DO), що відображає факт перевищення порогового рівня. Зовнішній вигляд модуля зображено на рисунку 3.7.

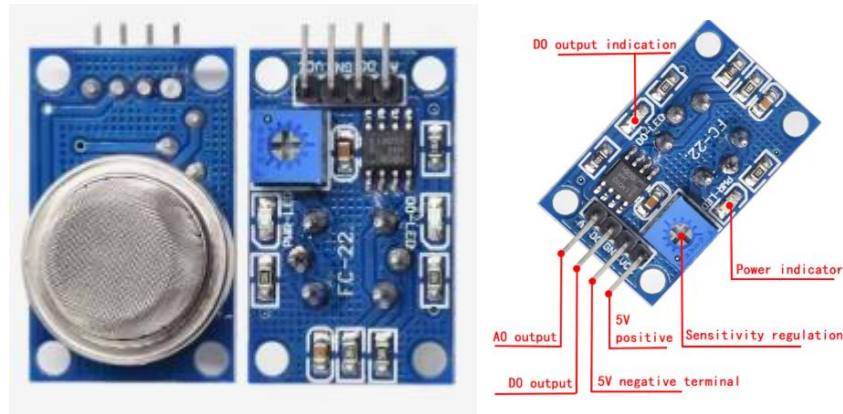


Рисунок 3.7 – Зовнішній вигляд модуля MQ135 та його будова.

Основні технічні характеристики MQ135:

- Напруга живлення нагрівального елементу: 5 В;
- Живлення модуля: 3.3–5 В;
- Споживання струму: до 150 мА;
- Час прогріву після вмикання: до 1 хвилини;
- Діапазон вимірювання (типовий):
 - Аміак (NH_3): 10–300 ppm;
 - Бензин: 10–1000 ppm;
 - Пари алкоголю: 10–300 ppm.

Для візуального відображення рівня забруднення повітря використовується RGB-світлодіод із загальним катодом, що забезпечує зміну кольору в залежності від рівня виявлених газів. До системи підключено дві кольорові компоненти — червону та зелену, якими керують плати ESP32 через пін-контакти GPIO34 та GPIO35 відповідно. Синя компонента у даній реалізації не використовується.

Підключення датчика та світлодіода наведено на рисунку 3.8.

Сенсор MQ135 з'єднується з ESP32 таким чином:

- Червоний провід — живлення (VCC, 3.3 В);
- Чорний провід — GND;
- Аналоговий вихід АО — до піну GPIO4, що підтримує ADC (аналогово-цифрове перетворення).

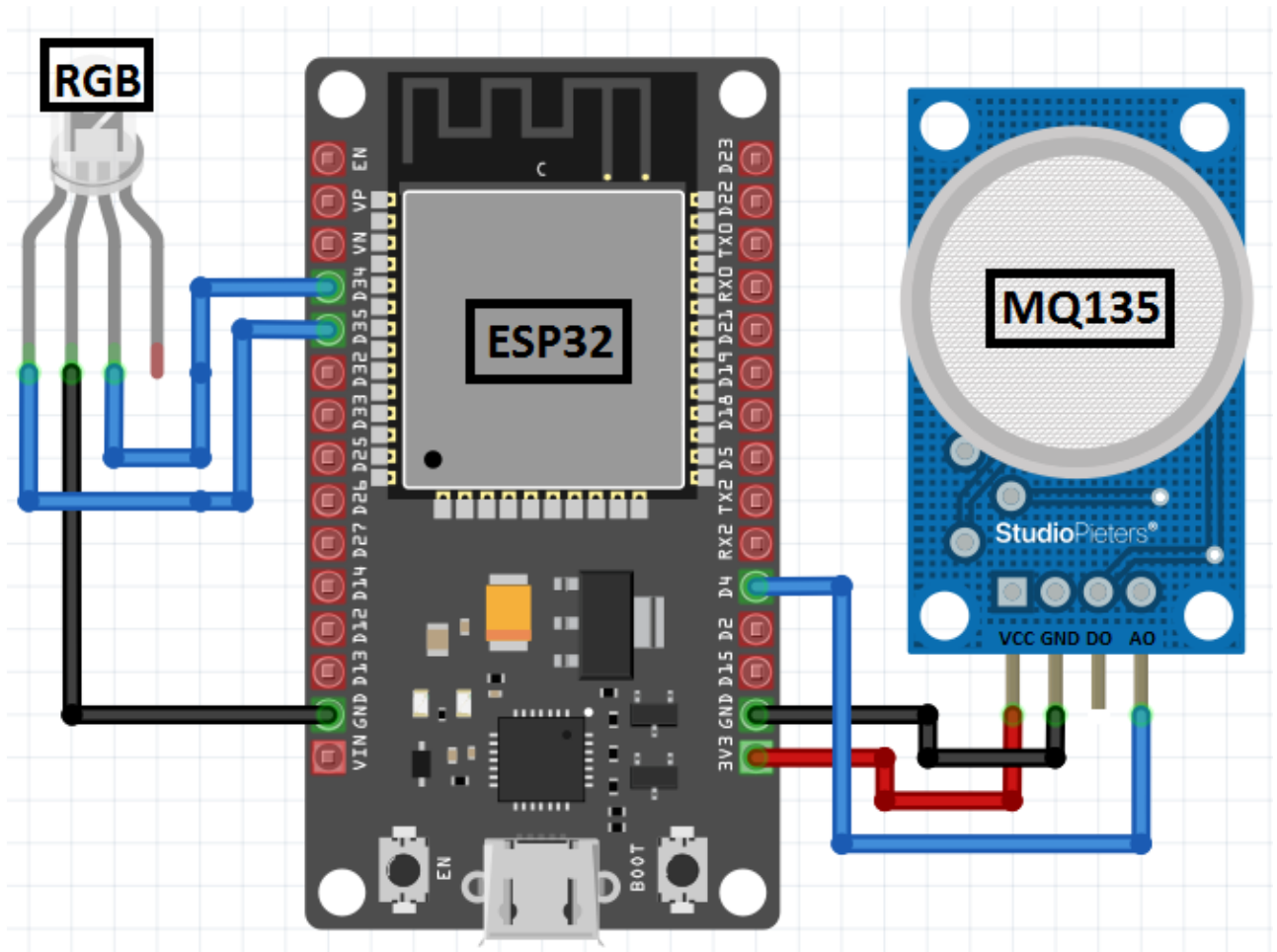


Рисунок 3.8 – Схема підключення світлодіода та датчика MQ135.

RGB-світлодіод підключений наступним чином:

- Загальний катод (–) — до землі (GND);
- Аноди червоної та зеленої компонент — до пінів GPIO34 і GPIO35, через резистори (для обмеження струму).

Це дає змогу індикації: наприклад, зелений колір може означати допустимий рівень якості повітря, а червоний — перевищення порогу концентрації шкідливих речовин.

3.1.4 Загальна схема підключення компонентів

Усі електронні компоненти системи підключені між собою за допомогою проводів з різним кольоровим маркуванням, що дозволяє легко розрізняти типи з'єднань та спрощує процес складання.

Загальна схема підключення компонентів наведена на рисунку 3.9.

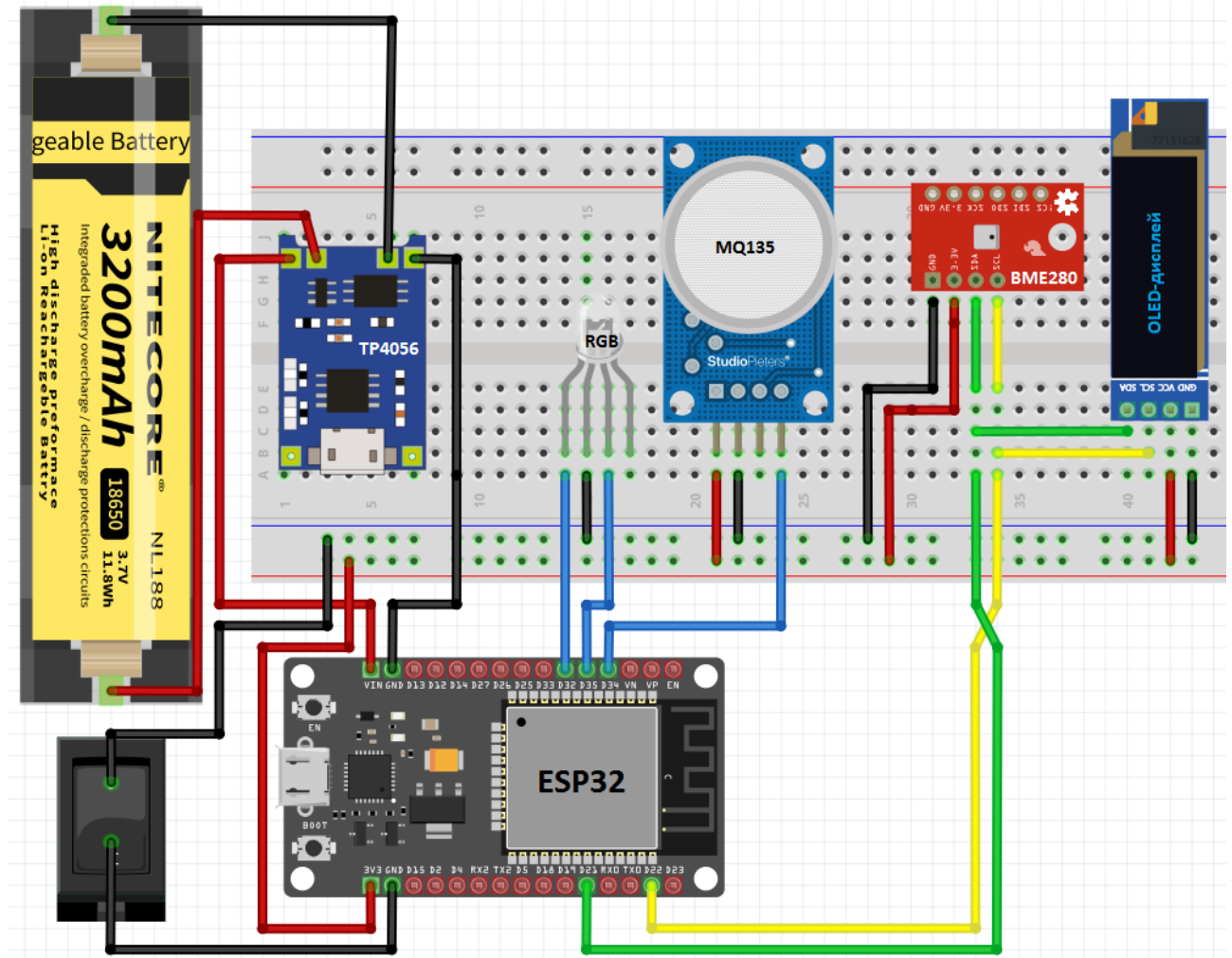


Рисунок 3.9 – Загальна схема підключення компонентів системи моніторингу.

Зокрема:

- червоний провід відповідає за лінію живлення (VCC),
- чорний — заземлення (GND),
- жовтий — лінія тактування SCL (Serial Clock Line),
- зелений — лінія передачі даних SDA (Serial Data Line),
- синій — використовується для інших з'єднань за потреби.

Усі компоненти розміщуються на макетній платі (breadboard) — це зручне рішення для створення прототипів без необхідності пайки. Такий підхід дозволяє легко змінювати конфігурацію під час тестування та налагодження.

У системі реалізовано перемикач режимів живлення, який дозволяє обирати між автономним живленням від акумулятора та живленням від зовнішнього джерела через порт micro-USB.

На плату ESP32 завантажується програмна прошивка, яка керує всіма підключеними модулями. Для взаємодії з OLED-дисплеєм та датчиком BME280 використовується інтерфейс I²C, що працює через виводи D22 (SCL) та D21 (SDA) мікроконтролера.

RGB-світлодіод підключений відповідно до раніше розглянутої схеми — через виводи мікроконтролера, що керують кольоровими компонентами світіння.

Газовий датчик MQ135 зчитується через аналоговий пін D34, який має вбудований аналогово-цифровий перетворювач (ADC). Це дозволяє мікроконтролеру отримувати числові значення концентрації газів у повітрі для подальшої обробки та відображення.

Загальна схема підключення компонентів була адаптована відповідно до вимог проєкту та перенесена з макетної плати на друковану плату розміром 7×9 см. Це дозволило зробити розташування елементів більш компактним і зручним для розміщення в корпусі. Усі з'єднання на платі були спроектовані так, щоб мінімізувати довжину провідників і забезпечити стабільну роботу пристрою. Завдяки використанню друкованої плати зменшився ризик обривів контактів або випадкових замикань, що є поширеною проблемою на макетках. Компоненти розміщено з урахуванням зручного доступу до пінів програмування та зчитування даних. На рисунку 3.10 представлено оновлену схему розміщення елементів на друкованій платі. Такий підхід підвищив надійність та ергономічність системи.

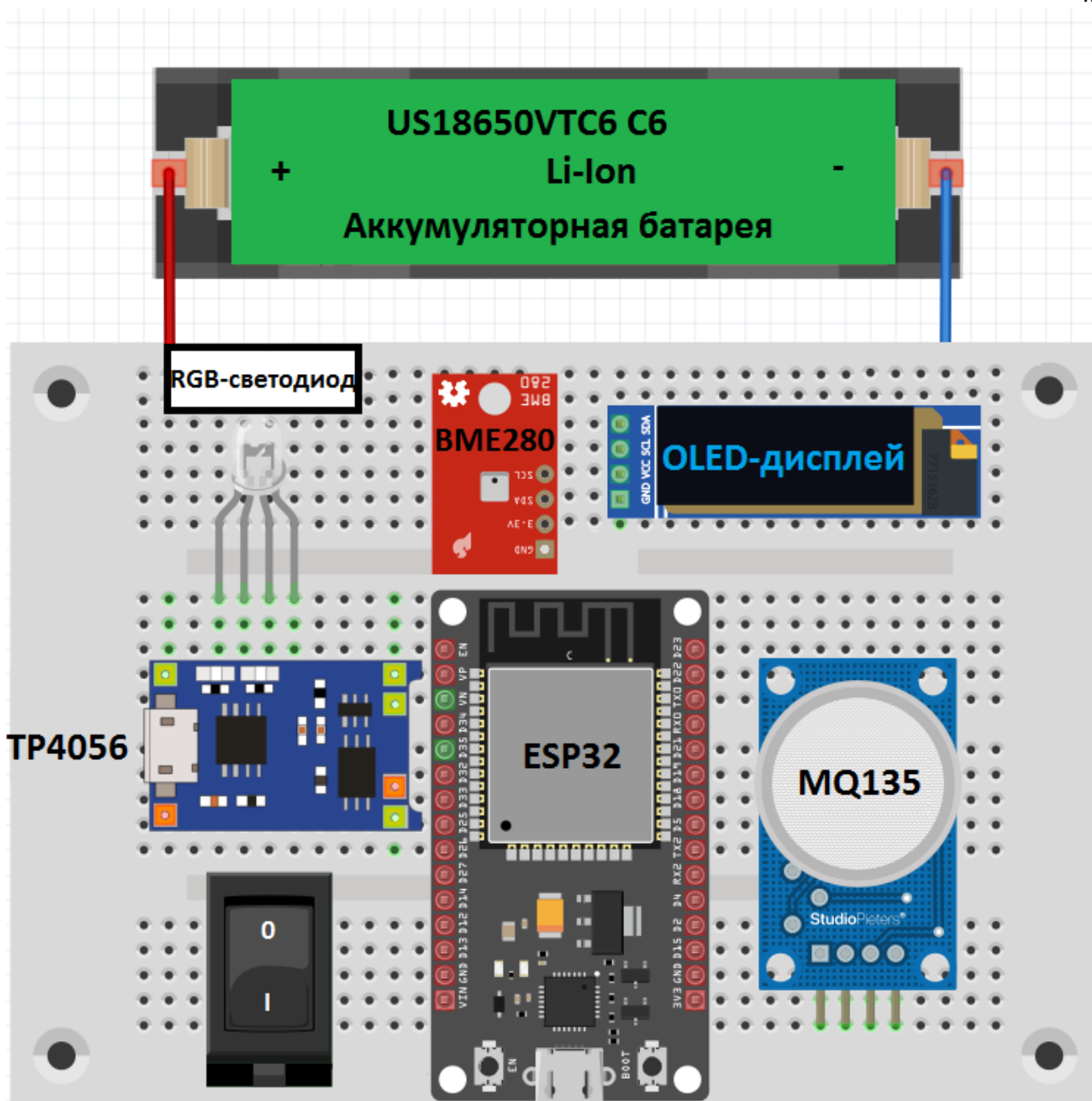


Рисунок 3.10 – Розташування компонентів на друкованій платі.

3.2 Розробка програмного забезпечення

Програмна частина є критично важливою складовою розробки будь-якої вбудованої системи, зокрема системи збору метеоданих. Вона забезпечує інтелектуальне управління усіма компонентами, починаючи від ініціалізації сенсорів і закінчуючи передачею зібраних даних на веб-сервер. Саме програмне забезпечення дозволяє перетворити апаратну базу на функціональний пристрій, здатний взаємодіяти з користувачем, мережею та іншими пристроями.

Код відповідає за обробку показників з датчиків, керування індикацією, відображення інформації на дисплеї, збереження в базі даних та виведення даних у вигляді графіків через веб-інтерфейс. Без правильно написаного програмного забезпечення навіть найкраще підібрані апаратні компоненти не будуть здатні повноцінно взаємодіяти між собою.

Крім того, програмна частина визначає енергоефективність, стабільність та масштабованість всієї системи. Вона дозволяє реалізовувати нові функції без зміни апаратної частини, а також забезпечує можливість оновлення прошивки у разі необхідності. Таким чином, програмне забезпечення є серцем системи — саме воно перетворює концепцію в працюючий інноваційний продукт.

Нами здійснено повний цикл роботи над програмною частиною системи: написання, тестування, налагодження та коментування коду, а також розробка та пояснення файлової структури проєкту. Особлива увага приділяється структурованій організації всіх файлів, які забезпечують роботу системи збору метеоданих. На рисунку 3.11 наведена загальна структура проєкту, що дозволяє швидко орієнтуватися в його складових.

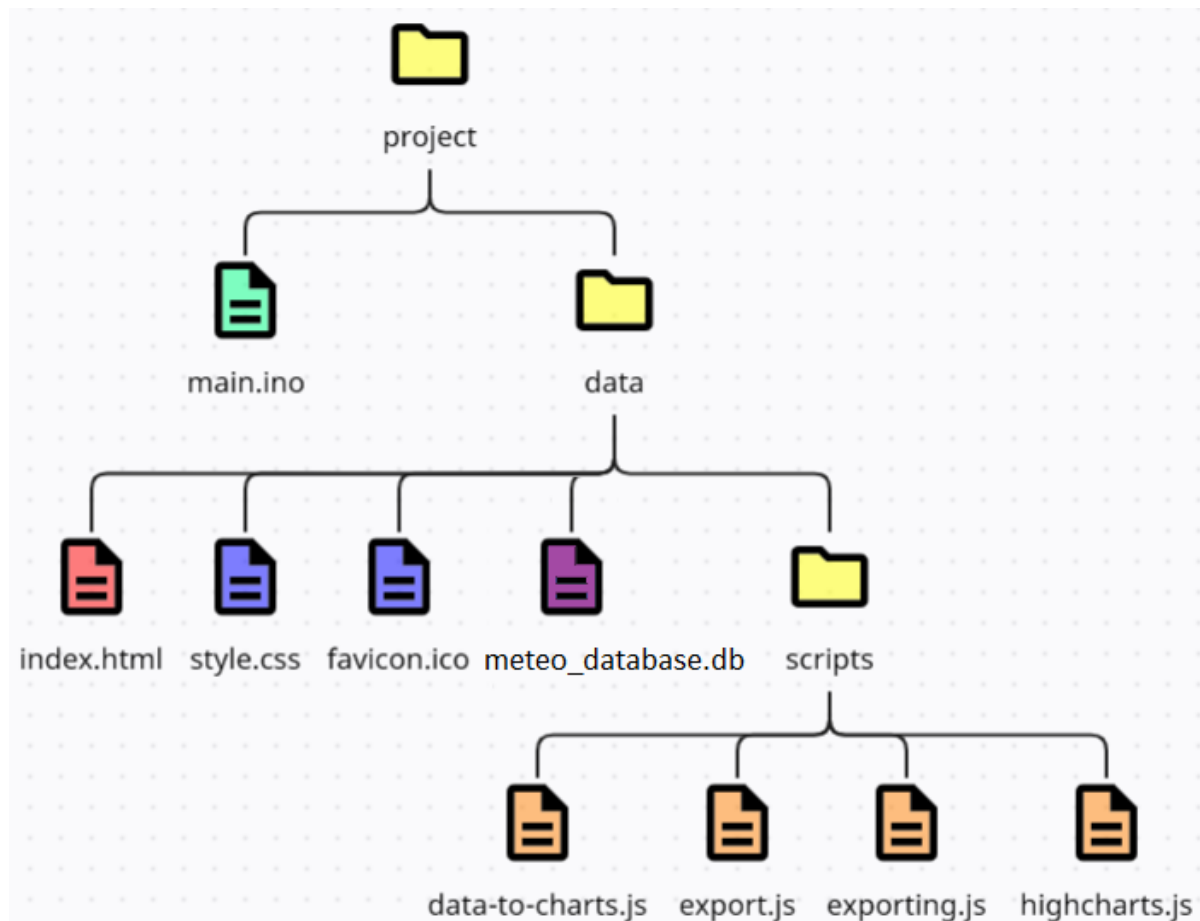


Рисунок 3.10 – Файлова структура проєкту.

3.2.1 Файлова структура проєкту

Каталог «projects».

Цей каталог є основним у проєкті. Тут знаходиться головний скетч — файл `main.ino`, написаний мовою C++ в середовищі Arduino IDE. Саме цей файл прошивається в ESP32 і керує усією логікою пристрою. Також у каталозі є підкаталог `data` — це коренева директорія файлової системи SPIFFS, де зберігаються файли для веб-інтерфейсу, які під час прошивки записуються до флеш-пам'яті контролера.

Каталог «data»

У цьому каталозі розміщено файли, які використовуються веб-сервером:

- `index.html` — основна HTML-сторінка;
- `style.css` — таблиця стилів;
- `favicon.ico` — іконка сторінки;
- `scripts/` — папка зі скриптами JavaScript;
- `meteo_database.db` — база даних, у якій зберігаються показники з датчиків.

Каталог «scripts»

Містить наступні JavaScript-файли:

- `data-to-charts.js` — відповідає за побудову графіків на веб-сторінці;
- `highcharts.js` — бібліотека для побудови графіків;
- `export.js` та `exporting.js` — модулі для експорту даних з графіків у вигляді зображення чи таблиці.

3.2.2 Вміст основних файлів

Файл `main.ino`

У цьому файлі реалізована основна логіка системи. Функція `setup()` ініціалізує роботу пристрою: налаштовуються пін-конфігурації, відбувається

підключення до Wi-Fi та запуск веб-сервера. Наприклад, у рядках 80–82 ініціюється відповідь сервера на запит GET / шляхом виводу файлу index.html з пам'яті SPIFFS.

Функція `loop()` виконується циклічно й забезпечує:

- зчитування показників з датчиків;
- відображення значень на OLED-дисплеї;
- контроль стану RGB-світлодіода згідно з показниками MQ135.

Важливий блок у рядках 129–154 відповідає за зміну яскравості світлодіодів залежно від рівня CO₂ у повітрі. За допомогою функції `map()` значення `ppm` лінійно трансформується у діапазон яскравості LED, що дозволяє візуально сигналізувати про стан повітря.

Файл data-to-charts.js

Цей файл обробляє динамічне оновлення графіків на сторінці без її перезавантаження. За допомогою AJAX-запитів дані (температура, вологість, тиск) запитуються з контролера кожну секунду, а потім додаються на відповідні графіки за допомогою Highcharts. Особливо цікавий блок коду у рядках 46–49, який додає нову точку на графік, перерисовує його, анімує зміну та при потребі видаляє найстарішу точку, зберігаючи лише актуальні дані.

Аналогічні функції реалізовано для решти параметрів — вологості (`chartH`) і тиску (`chartP`), кожен із власною кольоровою схемою та підписами осей.

Файл meteo_database.db

Це SQLite-база даних, яка слугує для зберігання показників із сенсорів. У таблиці `meteo_table` фіксуються значення температури, вологості та тиску, кожне з яких має унікальний ідентифікатор. Формат зберігання – тип `float`, що забезпечує точність до сотих часток. Така структура дозволяє швидко витягувати дані, формувати історію змін і будувати аналітику. На рисунку 3.11 зображено логічну модель цієї бази даних.

meteo_table		
KEY	NAME	TYPE
PK	id	INT PRIMARY KEY AUTO_INCREMENT
	temperature_value	REAL
	humidity_value	REAL
	pressure_value	REAL

Рисунок 3.11 – Логічна модель бази даних.

У результаті, програмна частина проєкту реалізує усі необхідні функції для збору, обробки, візуалізації та збереження метеоданих у режимі реального часу. Вона є масштабованою, зручною у підтримці, а її структура дозволяє легко додавати нові функції або сенсори.

4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам законодавства. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або для людей, які його оточують, або для виробничого середовища чи довкілля. Він зобов'язаний негайно повідомити про це безпосереднього керівника або роботодавця. Факт наявності такої ситуації за необхідності підтверджується спеціалістами з охорони праці підприємства за участю представника профспілки, членом якої він є, або уповноваженої працівниками особи з питань охорони праці (якщо професійна спілка на підприємстві не створювалася), а також страхового експерта з охорони праці. Завдання охорони праці – звести до мінімуму ушкодження та захворювання працівника з одночасним забезпеченням комфорту при максимальній продуктивності праці. Основними цілями охорони праці є формування в спеціалістів необхідних знань і практичних навичок по правових і організаційних питаннях охорони праці, виробничій санітарії, техніці безпеки, пожежній безпеці.

4.1. Загальна характеристика приміщення і робочого місця

Розробка системи керування виконується в приміщенні, яке знаходиться на четвертому поверсі восьмиповерхового будинку з загальним та місцевим освітленням. В приміщенні одностороннє освітлення, вікна орієнтовані на схід, на вікнах є ролети. Стеля білого кольору з коефіцієнтом відбиття 0,7, стіни

цегляні світлого кольору з коефіцієнтом відбиття 0,5. В приміщенні працює 4 людини, відповідно до цього отримуємо вхідні дані для аналізу потенційно-небезпечних і шкідливих виробничих факторів, які наведено в табл. 4.1.

Таблиця 4.1

Вхідні дані

Параметри приміщення	Значення
Площа	40,26 м ²
Об'єм	108,70 м ³
Номер робочого місця	Специфіка роботи
I робоче місце	Front-end програміст (спеціаліст з розробки клієнтської частини веб- застосунків)
II робоче місце	Back-end програміст (спеціаліст з розробки серверної частини веб застосунків та проектування баз даних)
III робоче місце	Бізнес-аналітик (також виконує роль менеджера продукту)
IV робоче місце	UI-UX веб-дизайнер
Технічні засоби (кількість)	Назва та характеристики
Монітор (4 шт.)	HP 22Xi/21,5"/1920x1080px/IPS
Комп'ютер (4 шт.)	HP ProBook 440 G6, екран 14" IPS (1920x1080) Full HD, Intel Core i7-8565U (1.8 - 4.6 ГГц)/RAM 16 ГБ/SSD 256 ГБ
Підлоговий кулер (1 шт.)	CRYSTAL YLR3-5V208
Кондиціонер (1 шт.)	DEKKER DSH105R/G/26м ² /2,65кВт-2,9кВт/25x74,5x19,5см/9 кг
Світильники загального призначення (3 шт.)	Світильник растровий вмонтований 4x18W
Світильники (4 шт.)	DeLux Décor TF-05 / 1 x 40Вт

Згідно НПАОП 0.00-7.15-18 площа S' , виділена для одного робочого місця з персональною ЕОМ, повинна бути не менше 6 м² і об'єм – не менше 20 м³. У приміщенні розташовано 4 робочі місця, що повністю відповідає необхідним нормам.

Розрахуємо фактичні значення цих показників, розділивши об'єм приміщення та загальну площу на кількість працюючих.

Отже, виходячи з отриманих результатів за характеристиками площі та об'єму, приміщення відповідає нормам.

Характеристики робочого місця

№	Найменування параметру	Значення	
		фактичне	нормативне
1.	Висота робочої поверхні, мм	780	680 – 800
2.	Ширина робочої поверхні, мм	1500	не менше 600
3.	Глибина робочої поверхні, мм	750	не менше 600
4.	Висота простору для ніг, мм	750	не менше 600
5.	Ширина простору для ніг, мм	800	не менше 500
6.	Глибина простору для ніг, мм	750	не менше 450
7.	Висота поверхні сидіння, мм	480	400 – 500
8.	Ширина сидіння, мм	500	не менше 400
9.	Глибина сидіння, мм	500	не менше 400
10.	Висота опорної поверхні спинки, мм	550	не менше 300
11.	Ширина поверхні спинки, мм	470	не менше 380
12.	Довжина підлокітників, мм	300	не менше 250
13.	Ширина підлокітників, мм	60	50 – 70
14.	Відстань від очей до екрану, мм	650	600 – 700

Можна зробити висновок, що розміри робочого місця програміста відповідають встановленим нормам, виходячи з заданих параметрів.

4.2. Аналіз потенційно небезпечних і шкідливих виробничих факторів на робочому місці

При створенні системи аналізу та візуалізації робота виконується сидячи без фізичних зусиль, тому відноситься до категорії легка Іа.

Під час роботи на працівника діє ряд небезпечних і шкідливих чинників, які наведені у табл. 4.3 та табл. 4.4.

Таблиця 4.3

Шкідливі чинники на робочому місці

Фізичні	Психофізіологічні
Підвищений рівень шуму	Розумове перенапруження

Підвищений рівень електромагнітного випромінювання	Монотонність праці
Підвищений рівень статичної електрики	Перенапруження аналізаторів
Недостатній рівень освітленості	
Неоптимальний мікроклімат	

Таблиця 4.4

Аналіз шкідливих факторів, пов'язаних з мікрокліматом

№	Шкідливий фактор	Наслідки
1	Відхилення вологості повітря від оптимальних параметрів	Тимчасове погіршення самопочуття і зниження працездатності, хвороби, роздратованість
2	Відхилення t від оптимальних параметрів	Відсутність теплового комфорту, тимчасове погіршення самопочуття і зниження працездатності, хвороби
3	Відхилення V руху повітря від оптимальних параметрів	Тимчасове погіршення самопочуття і зниження працездатності, хвороби

У таблиці 4.5 та 4.6 наведені нормативні та фактичні показники мікроклімату.

Таблиця 4.5

Мікроклімат в теплий період року

Параметр мікроклімату			
Найменування	Значення		
	Фактичне		Оптимальне
t , °C	21	21 – 23	18 – 27
w , %	55	60 – 40	до 75
V , м/с	0,2	0,3	0,4 – 0,2

Таблиця 4.6

Мікроклімат в холодний період року

Параметр мікроклімату			
Найменування	Значення		
	Фактичне		Оптимальне
t , °C	18	21 – 23	18 – 27
w , %	70	60 – 40	до 75
V , м/с	0,4	0,3	0,4 – 0,2

Заходи для запобігання встановлених мікрокліматичних порушень норм

подані в таблиці 4.7.

Приміщення для роботи мають бути обладнані системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією відповідно до ДБН В.2.5-67:2013. Нормовані параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин відповідають вимогам ДСН 3.3.6.042-99, ГН 2152-80, ГОСТ 12.1.005-88, ДСТУ ГОСТ 12.0.230:2008 та ДСТУ ГОСТ 12.4.041:2006. Під вентиляцією розуміють сукупність заходів та засобів, призначених для забезпечення на постійних місцях та зонах обслуговування приміщень метеорологічних умов та чистоти повітряного середовища, що відповідають гігієнічним та технічним вимогам. Основне завдання вентиляції – вилучити із приміщення забруднене, вологе або нагріте повітря та подати чисте свіже повітря.

Таблиця 4.7

Запобіжні заходи в теплий та холодний періоди року

№	Технічні	Організаційні	ЗІЗ
1	Контроль параметрів за допомогою анемометра Extech AN100; використання кондиціонеру DEKKER DSH105R/G (для кондиціонування і провітрювання)	відсутні	відсутні
2	Контроль параметрів за допомогою термометра La Crosse WS8005; використання кондиціонеру DEKKER DSH105R/G (для кондиціонування і провітрювання)	Перерви в роботі з метою провітрювання кімнати; вологе прибирання на робочих місцях	відсутні
3	Контроль параметрів за допомогою психрометра Т-04; використання зволожувача повітря ZELMER AH1500	Перерви в роботі з метою провітрювання кімнати; вологе прибирання на робочих місцях	відсутні

Джерелами шуму в приміщенні є вентилятор системного блоку, ноутбуку та кондиціонер (табл. 5.8). Звук, що створюється вентилятором та кондиціонером, можна класифікувати як постійний.

Джерела шуму

Джерело шуму	Фактичний рівень шуму, дБ	Оптимальний рівень шуму, дБ	Час роботи, год.
Кондиціонер DEKKER SH105R/G	22	< 50	8
Кулер комп'ютеру HP Probook 4530s	20		8

Наслідки шуму та вібрації подано у таблиці 4.9.

Таблиця 4.9

Шум і вібрація

Шкідливий фактор	Наслідки
Підвищений рівень шуму	Погіршення слуху, підвищення ймовірності виникнення помилки, зниження продуктивності роботи
Вібрації на робочому місці	Роздратування, зниження працездатності, погіршення самопочуття

Запобіжні заходи, які здійснюються для уникнення наслідків шкідливих факторів, наведено в табл. 4.10.

Таблиця 4.10

Запобіжні заходи

№	Технічні	Організаційні	ЗІЗ
1	Контроль параметрів за допомогою приладу для виміру шуму DT-8852; якісний монтаж окремих вузлів комп'ютера	Проведення планового попереджувального ремонту (чищення від пилу і інших забруднень)	Відсутні
2	Контроль параметрів за допомогою приладу для виміру вібрацій TV260; встановлення спеціальної підставки під ноутбук	Проведення планового попереджувального ремонту (чищення від пилу й інших забруднень)	Відсутні

Відповідно до ДБН В.2.5-28:2018 робота відноситься до розряду зорових робіт. Передбачається використання природного, штучного та змішаного освітлення. В табл. 4.11 наведені шкідливі фактори порушень норм яскравості світла.

Таблиця 4.11

Шкідливі фактори порушень норм яскравості світла

№	Шкідливий фактор	Наслідки
1	Недостатня освітленість робочої зони	Погіршення зору і самопочуття, втомлюваність, підвищення ризику
2	Підвищена яскравість світла	здійснення помилки

У таблиці 4.12 відображено фактичні та оптимальні значення для параметрів освітлення.

Таблиця 4.12

Параметри освітлення

Найменування	Значення	
	Фактичне	Оптимальне
При змішаному освітленні	450	400
При загальному освітленні	300	300
Коефіцієнт природного освітлення	1,23	1,2

Для уникнення наслідків неправильного освітлення вживаються такі запобіжні заходи (табл. 4.13):

Таблиця. 4.13

Запобіжні заходи

№	Технічні	Організаційні	ЗІЗ
1	Контроль параметрів за допомогою люксметра DT-1308; використання нових світильників загального призначення ELSTEAD FINSBURY PARK FP6 POL NICKEL; урахування природного освітлення кімнати	Встановлення мінімального рівня освітлення; чищення скла вікон та світильників; заміна ламп, що перегоріли	Додаткове освітлення на робочих місцях (світильники DeLux Décor TF-05); окуляри для роботи з комп'ютером.
2	Контроль параметрів за допомогою люксметра DT-1308; використання регульованих пристроїв для відкривання вікон, а також жалюзі; використання світильників нового типу	Відсутні	Окуляри для роботи з комп'ютером.

ЕОМ є однофазним споживачем електроенергії, що живиться від змінного струму 220В від мережі із заземленою нейтраллю. IBM PC відноситься до електроустановок до 1000В закритого виконання, всі струмопровідні частини знаходяться в кожухах. За способом захисту людини від ураження електричним струмом, ЕОМ і периферійна техніка повинні відповідати 1 класу захисту.

Технічні методи захисту від ураження струмом зводиться до застосування струму безпечної напруги, захисту у випадку випадкового доторкання до струмоведучих частин і від надмірних струмів, захисту у випадку переходу напруги на неструмоведучі металеві частини установки.

Безпечну напругу одержують від сітки підвищеної напруги (110-120 В) за допомогою знижувальних трансформаторів.

Захисту від доторкання до струмоведучих частин установки досягають за допомогою ізоляції, відгородження застосування блокуючих пристроїв запобіжної сигналізації та неприступності розташування установок.

Розподільні щитки поміщають у закриті металеві кожухи-ящики.

Запобіжну сигналізацію застосовують у вигляді плакатів і надписів. Найкращими світловими сигналізаціями є подвійні, яких при наявності напруги горить червона лампочка, а при її відсутності - зелена.

В табл. 4.14 наведені небезпечні фактори ураження людини електричним струмом.

Таблиця 4.14

Небезпечні фактори ураження людини електричним струмом

№	Шкідливий фактор	Наслідки	Заходи
1	Небезпечний рівень напруги струмопровідних частин обчислювальної та побутової техніки	Зростання ризику ураження електричним струмом	Релейний захист струму дотику, захисні заземлюючі корпуси. Попереджувальні знаки про рівень напруги.

Захист від надмірних струмів – короткого замикання і струмів перевантаження, які можуть спричинити займання ізоляції, здійснюється запобіжниками й автоматичними вимикачами, а захист від переходу напруги на струмоведучі частини за допомогою захисного заземлення і захисного

вимикання.

У таблиці 4.15 відображено фактичні та оптимальні значення для параметрів електропостачання.

Таблиця 4.15

Параметри електропостачання на робочому місці

Значення	Напруга, В	Частота, Гц	Тип розетки/вилки	Тип фази
Фактичне	220	50	F	Однофазна, трипровідна
Оптимальне	220	50	C, F	Однофазна, трипровідна

Вживаються такі запобіжні заходи для уникнення наслідків ураження людини електричним струмом (табл. 4.16):

Таблиця 4.16

Запобіжні заходи

№	Технічні	Організаційні	ЗІЗ
1	Релейний захист струму дотику, захисні заземлюючі корпуси	Проведення робіт з електричним обладнанням лише проінструктованим персоналом. Створення плану короткострокових відпочинків.	відсутні

Таблиця 4.17

Шкідливі фактори, пов'язані з пожежною безпекою

№	Шкідливий фактор	Наслідок
1	Коротке замикання, електротравми, пожежі, летальні наслідки	Коротке замикання, пожежі, електротравми, летальні наслідки
2	Коротке замикання	Електротравми, пожежі, летальні наслідки
3	Порушення протипожежного режиму	Електротравми, пожежі, летальні наслідки

Запобігання пожежі досягається виключенням утворення джерел загорянь і горючого середовища. У таблиці 4.17 приведено шкідливі фактори.

В цьому приміщенні можливі пожежі таких класів: А – горіння твердих речовин, Е – горіння електроустановок під напругою. Для забезпечення цих

категорій застосовуються заходи, що вказані в таблиці 4.18.

Таблиця 4.18

Запобіжні заходи

№	Технічні	Організаційні	ЗІЗ
1	Контроль параметрів за допомогою термометра La Crosse WS8005; використання кондиціонера DEKKER DSH105R/G (для кондиціонування і провітрювання)	Розвантаження електровузлів після виконання роботи; ознайомлення з інструкціями по використанню електроприладів;	відсутні
2	Наявність вогнегасника порошкового типу ОП-5 та автоматичної системи “ГАРАНТ-Р” (ПО-2), узгоджений план евакуації	Ознайомлення з інструкціями по використанню протипожежних засобів; узгоджений план евакуації	відсутні
3	Наявність вогнегасника порошкового типу ОП-5 та автоматичної системи “ГАРАНТ-Р” (ПО-2), узгоджений план евакуації	Ознайомлення з інструкціями по використанню протипожежних засобів; узгоджений план евакуації	відсутні

ВИСНОВКИ

В даній роботі проведено ґрунтовне дослідження поставленої задачі - здійснено аналіз існуючих аналогів та прототипів метеостанцій. Виявлено їхні сильні й слабкі сторони, що дозволило обґрунтувати актуальність створення власної системи збору метеоданих. Також розглянуто альтернативні підходи до реалізації та визначено найоптимальніший варіант на основі заданих умов. Узагальнено вагомі функціональні можливості системи, а також розглянуто варіанти реалізації та обґрунтовано вибір оптимального підходу — створення веб-сервера безпосередньо на мікроконтролері ESP32.

Головною метою кваліфікаційної роботи стало розроблення та створення автоматизованої системи збору та аналізу метеоданих з використанням технологій IoT, яка зможе повністю задовольнити запити користувачів.

Керуючись отриманою інформацією та поставлених цілях, спроектовано структурну схему пристрою. Також здійснено обґрунтований вибір необхідних апаратних і програмних компонентів, з яких мав складатися прототип автоматизованої системи збору та аналізу метеоданих.

Реалізовано основну функцію проєкту – побудовано систему динамічного візуального відображення даних із сенсорів у режимі реального часу з використанням бібліотеки Highcharts. Це забезпечує наочний моніторинг показників та сприяє глибшому розумінню змін навколишнього середовища. Передбачено зручний веб-інтерфейс, що дозволяє переглядати графіки показників у браузері без підключення до зовнішніх серверів.

Окрім цього, виконано повну апаратну збірку пристрою, завантаження програмного забезпечення у мікроконтролер та проведено тестування. У результаті підтверджено, що розроблена система повністю відповідає вимогам, виконує поставлені задачі та готова до практичного використання.

Загалом, у результаті роботи над розділами були створені чіткі технічні, апаратні та програмні основи для реалізації повноцінної системи збору, обробки та візуалізації метеорологічних даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білоус, В. О. Розробка метеостанції з бездротовою передачею даних / В. О. Білоус // Системи обробки інформації. — 2021. — № 4 (164). — С. 115–119.
2. Al-Fuqaha, A. Internet of Things: A survey on enabling technologies, protocols, and applications / A. Al-Fuqaha, M. Guizani, M. Mohammadi et al. // IEEE Communications Surveys & Tutorials. — 2015. — Vol. 17, no. 4. — P. 2347–2376.
3. Кононенко, В. М. Розробка розподіленої IoT-системи моніторингу погодних умов / В. М. Кононенко, Л. Ю. Данилюк // Збірник наукових праць НАУ. — 2021. — № 4. — С. 75–81.
4. Stanko, A., Wieczorek, W., Mykytyshyn, A., Holotenko, O., & Lechachenko, T. (2024). Realtime air quality management: Integrating IoT and Fog computing for effective urban monitoring. CITI, 2024, 2nd.
5. Котляр, І. С. Застосування технологій Інтернету речей у системах моніторингу клімату / І. С. Котляр // Вісник Київського національного університету технологій та дизайну. — 2020. — № 1. — С. 93–97.
6. Misra, S. Guide to Wireless Sensor Networks / S. Misra, I. Woungang, S. C. Misra. — London: Springer, 2009. — 707 p.
7. Гузій, М. І. Автоматизовані системи екологічного моніторингу на основі мікроконтролерів / М. І. Гузій, Т. О. Петренко // Науковий вісник НУВГП. — 2022. — № 3 (99). — С. 120–124.
8. Kumar, P. Smart Weather Station Using Raspberry Pi / P. Kumar, A. Shukla, S. Verma // International Journal of Advanced Research in Computer and Communication Engineering. — 2019. — Vol. 8, No. 2. — P. 44–48.
9. Руснак, О. Ю. Дослідження застосування сенсорних мереж для метеомоніторингу / О. Ю. Руснак // Інформаційні технології та комп'ютерна інженерія. — 2020. — № 2. — С. 48–54.
10. О Голотенко, А Бойчун, «Розробка автоматизованої системи моніторингу мікроклімату складських приміщень транспортної компанії з використанням технологій IoT», Матеріали XI науково-технічної

- конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя. – 2023. – С. 146.
11. Іванченко, С. В. Інтелектуальні системи вимірювання екологічних параметрів на основі IoT / С. В. Іванченко, А. Л. Паламарчук // Сучасні інформаційні технології у сфері безпеки та оборони. — 2022. — № 2 (42). — С. 33–38.
 12. А. Станько, А. Микитишин и О. Голотенко, "Комунікаційні технології в енергосистемах," XI Міжнародна науково-практична конференція молодих учених та студентів "Актуальні задачі сучасних технологій", 2022.
 13. Bernas, M., Mykytyshyn, A., Kartashov, V., Levytskyi, V. & Martjanov, D. (2023) The Role of Cyber-Physical Systems and Internet of Things in Development of Smart Cities for Industry 4.0. CEUR Workshop Proceedings. Vol. 3468,. P. 91–102.
 14. Vasisht, D. FarmBeats: An IoT platform for data-driven agriculture / D. Vasisht, A. Kapoor, J. Liu // 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI). — 2017. — P. 515–529.
 15. Gubbi, J. Internet of Things (IoT): A vision, architectural elements, and future directions / J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami // Future Generation Computer Systems. — 2013. — Vol. 29, no. 7. — P. 1645–1660.

ДОДАТОК А

Лістинг коду «main.ino»

```
// Підключення бібліотек
#define SEALEVELPRESSURE_HPA (1017) //Тиск над рівнем моря в hPa
// Номери портів світлодіодів
// Налаштування властивостей ШІМ-сигналу
// Частота
// Роздільна здатність = 256
// Номер червоного каналу
// Номер зеленого каналу
// Налаштування точки доступу
const char* ssid = "meteo"; // Ім'я точки доступу
AsyncWebServer server(80); // Присвоєння екземпляру класу імені та
призначення порту
Adafruit_BME280 bme; // Присвоєння екземпляру класу імені
Adafruit_SSD1306 display(128, 32); // Передача розмірів дисплея
екземпляру класу
IPAddress ip(192, 168, 0, 20); // Локальна статична IP-адреса
IPAddress gateway(192, 168, 0, 20); // IP-адреса шлюзу
IPAddress subnet(255, 255, 255, 0); // Маска підмережі
// Функція відображення значень температури, вологості та тиску на
OLED-дисплеї
display.setTextColor(WHITE); // Встановлення кольору тексту
display.clearDisplay(); // Очищення дисплея
display.setCursor(0,0); // Встановлення позиції курсора
display.print("Temperature: "); // Вивід мітки температури
display.print(temperature); // Вивід температури
display.println(" C"); // Додавання одиниць виміру та перехід на
наступний рядок
display.print("Humidity: "); // Вивід мітки вологості
display.print(humidity); // Вивід вологості
display.println(" %"); // Додавання одиниць виміру та перехід на
наступний рядок
display.print("Pressure: "); // Вивід мітки тиску
display.print(pressure / 100); // Вивід тиску
display.println(" hPa"); // Додавання одиниць виміру та перехід на
наступний рядок
display.print("Altitude: "); // Вивід мітки рівня
display.print(round(altitude)); // Вивід рівня над морем
display.println(" m"); // Додавання одиниць виміру та перехід на
наступний рядок
display.display(); // Вивід на дисплей даних з оперативної пам'яті
// Функція setup, яка виконується один раз при запуску
// Налаштовуємо канали
// Прив'язуємо порти до каналів
bme.begin(0x76); // Ініціалізація датчика за його адресою
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Ініціалізація
дисплея
delay(500); // Затримка
SPIFFS.begin(); // Ініціалізація файлової системи
// Відкриваємо файл
// Отримуємо розмір файлу
```

```

// Закриваємо файл
// Відкриття з'єднання з базою даних
// Видаляємо всі записи в таблиці, якщо розмір файлу перевищує 1
МБ (1048576 байт)
delay(500); // Затримка
/*
Обробник запиту для кореневого шляху,
який викликає лямбда-функцію request і надсилає index.html
при зверненні клієнта до кореневого каталогу файлової системи
*/
// Закриття бази даних
server.begin(); // Запуск сервера
delay(1000);
void loop(){ // Функція loop() виконується в нескінченному циклі і
містить основну логіку програми
float temperature = bme.readTemperature(); // Зчитуємо температуру
float humidity = bme.readHumidity(); // Зчитуємо вологість
float pressure = bme.readPressure(); // Зчитуємо тиск
float altitude = bme.readAltitude(SEALEVELPRESSURE_HPA); //
Отримуємо висоту над рівнем моря
// Відкриття з'єднання з базою даних
// Формування SQL-запиту для вставки даних в таблицю meteo_table
// Закриття бази даних
int sensorValue = analogRead(4); // Зчитування аналогового
значення з піна датчика
float voltage = sensorValue * (3.3 / 4096.0); // Перетворення
аналогового значення в напругу (3.3 - напруга живлення ESP32, 4096
- роздільна здатність АЦП ESP32)
float ppm = (1.1 * voltage - 0.1) * 1000 / 3; // Перетворення
напруги в концентрацію CO2 в ppm (значення 1.1 і 0.1 можуть
змінюватися в залежності від калібрування датчика)
// Зміна кольору світлодіода в залежності від значення ppm
if (ppm <= 400){ // зелений колір при значеннях <= 400ppm
...
else if (ppm >= 1800){ // Червоний колір при значеннях більше >=
1800ppm
...
delay(1000); // Затримка перед наступною ітерацією

```

ДОДАТОК Б

Лістинг коду «index.html»

```
<!DOCTYPE HTML>
<html>
<head>
  <!-- Встановлюємо заголовок сторінки вебсайту -->
  <title>Метеостанція</title>
  <!-- Визначаємо параметри відображення сторінки на мобільних пристроях -->
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Встановлюємо значок (favicon) для вебсторінки -->
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <!-- Підключаємо зовнішній CSS-файл для стилізації сторінки -->
  <link rel="stylesheet" type="text/css" href="style.css">
  <!-- Підключаємо бібліотеку Highcharts для побудови графіків -->
  <script src="scripts/highcharts.js"></script>
  <!-- Підключаємо скрипт для експорту даних з графіків -->
  <script src="scripts/export-data.js"></script>
  <!-- Підключаємо скрипт для відображення кнопок експорту графіків -->
  <script src="scripts/exporting.js"></script>
</head>
<body>
  <!-- Створюємо блок з заголовком сторінки -->
  <div class="header">
    <!-- Виводимо заголовок сторінки -->
    <h1>СИСТЕМА ЗБОРУ МЕТЕОДАНИХ</h1>
  </div>

  <!-- Створюємо блок для відображення графіків -->
  <div class="charts">
    <!-- Створюємо сітку для розташування графіків всередині блоку -->
    <div class="card-grid">
      <!-- Створюємо картку для кожного графіка -->
      <div class="card">
        <!-- Створюємо контейнер з унікальним ідентифікатором "chart-temperature" для графіка температури -->
        <div id="chart-temperature">
          </div>
        </div>
      </div>

      <!-- Графік вологості -->
      <div class="card-grid">
        <div class="card">
          <div id="chart-humidity">
            </div>
        </div>
      </div>

      <!-- Графік тиску -->
      <div class="card-grid">
        <div class="card">
          <div id="chart-pressure">
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```
</div>
</div>
</div>

<!-- Створюємо блок підвалу сторінки -->
<div class="footer">
  <!-- Відображаємо інформацію про автора сторінки -->
  <h1>Студенти групи КТз-41</h1>
</div>

<!-- Підключаємо скрипт, який обробляє дані та будує графіки на їх основі -->
<script src="scripts/data-to-charts.js"></script>
</body>
</html>
```

ДОДАТОК В

Лістинг коду «style.css»

```
html { /* Стиль усієї сторінки */
  font-family: Arial, Helvetica, sans-serif; /* Шрифт */
  display: inline-block; /* Щоб застосовувати вирівнювання та інші властивості блочного
елемента */
  text-align: center; /* Вирівнює текст всередині елемента по центру */
}
body {
  margin: 0; /* Відступи для всього вмісту сторінки */
}
.header { /* Заголовок */
  overflow: hidden; /* Приховує будь-який вміст, який не поміщається всередині блока */
  background-color: #0b1c52; /* Колір фону блока */
  color: white; /* Колір тексту всередині блока */
}
.card-grid {
  padding: 5%; /* Відступи навколо вмісту блока */
  max-width: 1800px; /* Максимальна ширина блока */
  margin: 0 auto; /* Горизонтальне вирівнювання блока по центру */
  display: grid; /* Встановлює блок як контейнер сітки */
  grid-gap: 2rem; /* Проміжок між елементами сітки */
  /*
  Шаблон колонок сітки .auto-fit дозволяє елементам автоматично розтягуватися для
заповнення доступного простору,
а minmax(200px, 1fr) встановлює мінімальну ширину кожної колонки в 200 пікселів,
але також дозволяє їм розтягуватися за допомогою fr (частка від доступного простору).
*/
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
}
.card {
  background-color: white; /* Колір фону картки */
  box-shadow: 2px 2px 12px 1px rgba(140,140,140,.6); /* Тінь */
}
.footer {
  overflow: hidden; /* Приховує будь-який вміст, який не поміщається всередині блока */
  background-color: #0b1c52; /* Колір фону */
  color: white; /* Колір тексту всередині блока */
}
```

ДОДАТОК Г

Лістинг коду «data-to-charts.js»

```
// Створюється новий екземпляр графіка Highcharts,
// пов'язаний з елементом з ідентифікатором 'chart-temperature'
var chartT = new Highcharts.Chart({
  chart:{ renderTo : 'chart-temperature' },
  // Заголовок графіка встановлюється на 'Температура'
  title: { text: 'Температура' },
  // Створюється серія даних для графіка з порожнім масивом значень
  series: [{
    showInLegend: false,
    data: [],
    name: 'bme280'
  }],
  // Опції для відображення лінії графіка
  plotOptions: {
    line: { animation: false},
    series: { color: '#059e8a'}
  },
  // Вісь X налаштована для відображення дати та часу
  xAxis: { type: 'datetime'},
  // Вісь Y з заголовком 'Temperature, °C'
  yAxis: { title: { text: 'Temperature, °C' }},
  // Підпис графіка в правому нижньому куті вимкнено
  credits: { enabled: false },
  // Установлено використання локального часу, а не UTC
  time: { useUTC: false},
});

// Ця функція буде виконуватися з певним інтервалом часу
setInterval(function () {
  // Створюється новий об'єкт XMLHttpRequest, який використовується для надсилання
  // HTTP-запитів на сервер
  var xhttp = new XMLHttpRequest();
  // Встановлюється обробник подій зміни стану запиту
  // Функція викликається кожного разу, коли змінюється стан XMLHttpRequest
  xhttp.onreadystatechange = function() {
    // Перевіряється, що стан запиту дорівнює 4 (завершено), а статус відповіді сервера
    // дорівнює 200 (успішно)
    if (this.readyState == 4 && this.status == 200) {
      // Створюється змінна x — поточний час у мілісекундах
      // Використовується як мітка часу на графіку
      var x = (new Date()).getTime(),
      // Створюється змінна y — числове значення з відповіді сервера
      // Вона відображає значення температури з датчика
      y = parseFloat(this.responseText);
      // Перевіряється довжина масиву даних на графіку
      // Якщо вона перевищує 30 елементів, додається нова точка, найстаріша видаляється,
      // графік прокручується
      if(chartT.series[0].data.length > 30) {
        chartT.series[0].addPoint([x, y], true, true, true);
      } else {
```



```

        chartT.series[0].addPoint([x, y], true, false, true);
    }
}
};
// Встановлюється метод і URL для відправки GET-запиту на сервер
// У цьому випадку — GET-запит на "/temperature"
xhttp.open("GET", "/temperature", true);
// Надсилається HTTP-запит
xhttp.send();
// Інтервал 1000 мілісекунд (1 секунда)
}, 1000);

// Графік вологості
var chartH = new Highcharts.Chart({
    chart:{ renderTo:'chart-humidity' },
    title: { text: 'Вологість' },
    series: [{
        showInLegend: false,
        data: [],
        name: 'bme280'
    }],
    plotOptions: {
        line: { animation: false }
    },
    xAxis: { type: 'datetime' },
    yAxis: { title: { text: 'Humidity, %' } },
    credits: { enabled: false },
    time: { useUTC: false },
});

setInterval(function () {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var x = (new Date()).getTime(),
                y = parseFloat(this.responseText);
            if(chartH.series[0].data.length > 30) {
                chartH.series[0].addPoint([x, y], true, true, true);
            } else {
                chartH.series[0].addPoint([x, y], true, false, true);
            }
        }
    };
    xhttp.open("GET", "/humidity", true);
    xhttp.send();
}, 1000);

// Графік тиску
var chartP = new Highcharts.Chart({
    chart:{ renderTo:'chart-pressure' },
    title: { text: 'Тиск' },
    series: [{
        showInLegend: false,
        data: [],

```

```

    name: 'bme280'
  }],
  plotOptions: {
    line: { animation: false},
    series: { color: '#18009c'}
  },
  xAxis: { type: 'datetime'},
  yAxis: { title: { text: 'Pressure, hPa' }},
  credits: { enabled: false },
  time: { useUTC: false},
});

setInterval(function () {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      var x = (new Date()).getTime(),
      y = parseFloat(this.responseText);
      if(chartP.series[0].data.length > 30) {
        chartP.series[0].addPoint([x, y], true, true, true);
      } else {
        chartP.series[0].addPoint([x, y], true, false, true);
      }
    }
  };
  xhttp.open("GET", "/pressure", true);
  xhttp.send();
}, 1000);

```