

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-апаратний засіб "Погодна станція на ESP32" із відображенням даних у

Blynk
Назва теми

КВРКІ 210106.21.01.42.ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

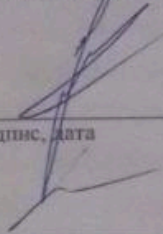
Виконав: студент IV курсу, група K12-22-1


Підпис

Денис ВІТЮК

Ініціали, прізвище

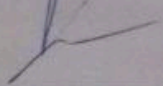
Керівник


Підпис, дата

Андрій НІЧЕПОРУК

Ініціали, прізвище

Нормоконтролер


Підпис, дата

Тетяна КИСЛІВ

Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«12» червня 2025 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Вітюку Денису Віталійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-апаратний засіб "Погодна станція на ESP32" із відображенням даних у Vlynk

Керівник проекту (роботи) Андрій Нічепорук, д.т.н...

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

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

Кіберфізична система адаптивного застосування моніторингових елементів погодної станції на ESP32 та постановка задачі щодо її удосконалення.

Проектування системи обробки інформації у кіберфізичній системі адаптивного застосування моніторингових елементів погодної станції на ESP32.

Програмно-апаратна реалізація кіберфізичної системи адаптивного застосування моніторингових елементів погодної станції на ESP32 із відображенням даних у Vlynk.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Структура пристрою

Схема електрична

Панель керування Vlynk

6. Консультанти розділів дипломного проекту (роботи)

Підпис, дата

Розділ	Прізвище, ініціали та посада консультанта	завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

2025 р.

7. Дата видачі завдання

« 10 » 01

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	ВИКОНАНО
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	ВИКОНАНО
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	ВИКОНАНО
4	Робота над розділом 2 – вибір компонентів для проєктування системи моніторингу погоди	01.04.2025	ВИКОНАНО
5	Робота над розділом 3 – проєктування системи моніторингу погоди	29.04.2025	ВИКОНАНО
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	ВИКОНАНО
7	Попередній захист ВКР	26.05.2025	ВИКОНАНО
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Підпис

Керівник роботи

Підпис

Денис ВІТЮК
Ініціали, прізвище

Андрій НІЧЕПОРУК
Ініціали, прізвище

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л л я с т і в		П р и м і т к я
				№	ск з	
1		КвРКІ 210106.21.01.42.ПЗ	Текстові документи	60		
			Пояснювальна записка			
2		КвРКІ 210106.21.01.42.ПЗ	Графічні матеріали	1		
			Структура проекту			
3		КвРКІ 210106.21.01.42.ПЗ	Електрична схема	1		
4		КвРКІ 210106.21.01.42.ПЗ	Панель керування ВІУК	1		

КвРКІ 210106.21.01.42.ВП

Зм	Арк	№ докум	Пілляс	Дата
Розробив		Вітюк		
Перевір		Нічепорук		12.06.25
Н. монір		Кисель		
Затв.		Павлов		

Відомість проекту

Літера	Аркуш	Аркушів
У	1	1

ХНУ, К12-21-1

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратний засіб "Погодна станція на ESP32" із відображенням даних у Blynk».

Автор роботи: Денис Вітюк.

Керівник роботи: Нічепорук Андрій Олександрович.

Пояснювальна записка: 60 с., 2 табл., 4 дод., 40 джерел.

Графічна частина: 2 креслення.

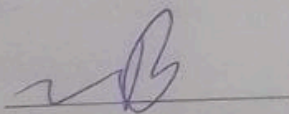
МІКРОКОНТРОЛЕР, ХМАРНЕ СХОВИЩЕ, ІНТЕРНЕТ РЕЧЕЙ,
АПАРАТНО-ПРОГРАМНА АРХІТЕКТУРА

Мета дослідження - проєктування та розробка програмно-апаратного засобу для виконання моніторингу метеорологічних параметрів за допомогою мікроконтролера ESP32 з відображенням даних у платформі Blynk.

Об'єкт дослідження - програмне забезпечення для мікроконтролерної системи збору та передачі метеорологічних даних.

Предмет дослідження - мікроконтролери ESP32 та системи моніторингу погодних умов.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації, а також практичне тестування апаратної та програмної частин для виявлення ефективності роботи станції.



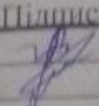



Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП		3
1 АНАЛІЗ ВІДОМИХ ПРОГРАМНО-ТЕХНІЧНИХ РІШЕНЬ		
ВІМІРЮВАННЯ КЛІМАТИЧНИХ ПОКАЗНИКІВ		4
1.1	Аналіз предметної області та наявних рішень	4
1.2	Загальні принципи функціонування та методи вимірювання метеостанцій	12
1.3	Висновки до першого розділу	19
2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ		
"ПОГОДНА СТАНЦІЯ НА ESP32"		20
2.1	Вимоги до програмно-апаратного засобу	20
2.2	Структура програмно апаратного засобу	21
2.3	Порівняльний аналіз мікроконтролерних систем	27
2.4	Програмне забезпечення та середовища розробки	31
2.5	Порівняльний аналіз хмарних платформ для візуалізації даних з мікроконтролерів	34
2.6	Вибір та аналіз вартості компонентів для погодної станції	39
2.7	Висновок до розділу 2	48
3.1	Розгортання та налаштування хмарної платформи Blynk	49
3.2	Створення android версії хмарної платформи	53
3.3	Програмна реалізація клієнтської частини програмно-апаратного засобу	54
3.4	Налаштування сповіщень на платформі Blynk	57
ВИСНОВКИ		59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ		60
ДОДАТОК А		64
ДОДАТОК Б		65
ДОДАТОК В		66
ДОДАТОК Г		67

КвРКІ 210106.21.01.42.ПЗ								
Зм.	Арк.	№ докум.	Підпис	Дата	Програмно-технічний засіб "Погодна станція на ESP32" із відображенням даних у Blynk	Літера	Аркуш	Аркушів
Виконав		Денис ВІТЮК		12.06.25		у	2	72
Перевір.		Андрій ПРИБОРУК			ХНУ КІ2-21-1			
І.контр.		Тетяна КИСІЛЬ		12.06.25				
Затвер.		Єлиза ПAVЛОВА						

ВСТУП

Програмно-технічні рішення для автоматизації все більше входять у наше життя, використовуючись у безлічі сфер. Одним з цікавих напрямків є розробка особистих систем моніторингу та управління, які можна впроваджувати безпосередньо в свій простір.

Сучасні технології Інтернету речей (IoT) дозволяють створювати компактні та доступні пристрої для збору даних про навколишнє середовище. Особливу увагу привертають системи метеорологічного моніторингу, які забезпечують отримання актуальної інформації про погодні умови в конкретній локації. Традиційні метеостанції часто розташовані на великих відстанях і не завжди відображають локальні особливості мікроклімату, що створює потребу в персональних рішеннях.

Мікроконтролер ESP32 є перспективним рішенням для створення автономних систем збору даних завдяки своїй функціональності та доступності. Він поєднує обчислювальні можливості з вбудованими модулями бездротового зв'язку, що дозволяє легко інтегрувати різноманітні датчики та передавати дані в хмарні сервіси. Платформа Blynk надає зручні інструменти для візуалізації даних та створення користувацьких інтерфейсів без складного програмування.

В такому контексті створення програмно-технічного засобу "Метеостанція на ESP32" з візуалізацією даних через Blynk є корисним і важливим рішенням, яке поєднує практичну цінність з технологічною доступністю.

Мета дослідження - визначення особливостей застосування програмно-апаратного засобу "Погодна станція на ESP32" та оцінка ефективності збору й відображення метеорологічних даних у реальному часі через платформу Blynk.

Об'єктом дослідження є функціонування елементів погодної станції.

Предметом дослідження є процес збору, обробки та передачі метеорологічних даних за допомогою ESP32 і Blynk.

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ВІДОМИХ ПРОГРАМНО-ТЕХНІЧНИХ РІШЕНЬ ВИМІРЮВАННЯ КЛІМАТИЧНИХ ПОКАЗНИКІВ

1.1 Аналіз предметної області та наявних рішень

З розвитком мікроелектроніки відбулися суттєві трансформації у сфері моніторингу довкілля. Якщо раніше метеорологічні станції представляли собою складні та масивні системи, які встановлювалися переважно у спеціалізованих установах, то сьогодні спостерігається тенденція до їх мініатюризації та здешевлення.

Крім того, сучасні технології забезпечують автоматизований збір, передачу та обробку метеорологічних даних, що усуває потребу у ручному веденні записів і підвищує ефективність моніторингу параметрів навколишнього середовища, таких як температура, вологість і атмосферний тиск.

Пристрої, здатні в режимі реального часу аналізувати параметри навколишнього середовища та передавати відповідну інформацію для подальшого прийняття рішень, перестають бути виключно теоретичною розробкою.

Зокрема, метеостанція на базі мікроконтролера ESP32 є прикладом такої системи, яка забезпечує не лише вимірювання базових метеорологічних показників (температури, вологості, якості повітря), а й їх інтеграцію у ширший контекст систем автоматизації та Інтернету речей (ІоТ).

Сучасні метеостанції є інтегрованими системами обладнання та датчиків, які можуть бути розташовані як в приміщеннях, так і на відкритих ділянках. Вони об'єднані в єдину мережу, що дозволяє здійснювати управління через комп'ютери, смартфони та інші цифрові пристрої. Головною метою таких систем є забезпечення ефективного моніторингу погодних умов та полегшення процесу їх спостереження.

Метеостанції налаштовуються відповідно до індивідуальних потреб користувача, що робить їх універсальними для застосування в різних сферах. Одним з основних напрямів використання таких систем є виявлення і сповіщення про значні зміни в погодних умовах. За допомогою швидкого фіксування таких

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

параметрів, як атмосферний тиск, вологість, швидкість вітру та кількість опадів, метеостанції дають змогу своєчасно реагувати на потенційно небезпечні явища, такі як шторми, заморозки та інші катастрофічні погодні зміни [1].

Іншим важливим аспектом є ефективне використання ресурсів. Метеостанції допомагають оптимізувати споживання води та електроенергії. Наприклад, у сільському господарстві вони дозволяють здійснювати контроль вологості ґрунту, що забезпечує ефективне зрошення, а в теплицях - автоматичне регулювання мікроклімату, що сприяє створенню оптимальних умов для вирощування рослин.

Автоматизовані метеосистеми також підвищують зручність та продуктивність. Вони забезпечують отримання даних в режимі реального часу без необхідності постійного моніторингу з боку користувача, а також дозволяють зберігати історію показників для подальшого аналізу та прогнозування.

Для належного функціонування метеостанції необхідно використання різноманітних пристроїв, кожен з яких виконує певну роль у загальній системі. Вимірювальні блоки є основними компонентами, які реєструють параметри навколишнього середовища, зокрема температуру повітря, атмосферний тиск, відносну вологість, інтенсивність сонячного випромінювання, напрямок та швидкість вітру, кількість опадів і інші показники. Деякі сенсори можуть також виявляти наявність снігу, рівень ультрафіолетового випромінювання або концентрацію пилу в повітрі, що дозволяє здійснювати більш детальний аналіз атмосферних умов.

Важливою складовою є контроль мікроклімату. Це включає в себе датчики температури, вологості повітря, атмосферного тиску та якості повітря. Завдяки цим датчикам система може відстежувати зміни погодних умов, передбачати заморозки чи спеку, а також автоматично регулювати роботу кліматичного обладнання, забезпечуючи оптимальні умови для різних процесів [2].

Ще одним важливим елементом є забезпечення безпеки та попередження про небезпечні погодні явища. Деякі метеостанції здатні виявляти різкі зміни погоди, такі як шторми, сильні опади або екстремальну спеку. У разі перевищення певних

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

порогових значень система може сповістити користувача про можливі ризики для врожаю, техніки або інфраструктури.

Інформаційно-аналітична підтримка є ще однією складовою функціоналу метеостанцій. Цей модуль дозволяє збирати статистичні дані, будувати графіки, порівнювати погодні параметри за різні періоди та експортувати дані для подальшого аналізу. Така функція є корисною для фермерів, наукових установ і комунальних служб, які потребують точного аналізу погодних умов для прийняття обґрунтованих рішень.

З розвитком інтегрованих метеорологічних систем з'явилися високотехнологічні пристрої, що адаптуються під різноманітні потреби користувачів - погодні станції. Ці системи відзначаються різними функціями, технологією з'єднання та структурою, що сприяє появі класифікацій метеостанцій за кількома критеріями. Зокрема, за організаційною формою управління метеостанції поділяються на централізовані та децентралізовані.

Централізовані системи мають єдиний контролер, що здійснює збір та аналіз усіх даних, тоді як децентралізовані системи складаються з автономних модулів, що можуть функціонувати незалежно один від одного. Цей підхід підвищує стійкість системи в разі відмови окремих елементів, що особливо важливо для критично важливих об'єктів, де необхідна безперервність збору даних.

Такі пристрої можна також класифікувати як універсальні або спеціалізовані. Універсальні погодні станції призначені для широкого ринку і забезпечують вимірювання кількох параметрів, таких як температура, вологість, швидкість вітру, кількість опадів тощо. Натомість спеціалізовані метеостанції фокусуються на виконанні конкретних завдань, таких як моніторинг ультрафіолетового випромінювання, рівня забруднення повітря або інших специфічних параметрів.

Розглянемо приклади найпопулярніших комерційних погодних станцій на сучасному ринку:

Однією з провідних систем є Netatmo Weather Station, розроблена французьким брендом Netatmo у 2012 році. Ця система заснована на модульній

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

архітектурі з бездротовим з'єднанням між компонентами. Основу становлять два головні модулі: внутрішній, який вимірює температуру, концентрацію CO₂, вологість повітря, рівень шуму та атмосферний тиск, та зовнішній модуль для визначення температури та вологості довкілля.



Рисунок 1.1 – Netatmo Weather Station [1]

Система використовує власний радіопротокол 433 МГц для бездротового зв'язку між зовнішнім сенсором та базовою станцією, а також IEEE 802.11b/g/n (Wi-Fi) для підключення до інтернету. Протоколи керування доступом реалізовані через WPA/WPA2 шифрування для Wi-Fi з'єднання та базову HTTP автентифікацію для веб-доступу. Інтеграція з зовнішніми системами забезпечується через Weather Underground API та підтримку протоколу Ecowitt для сумісності з іншими метеостанціями. Методи обробки даних включають вбудовану обробку сигналів з 12-бітним аналого-цифровим перетворювачем, локальне зберігання в EEPROM пам'яті та експорт у XML форматі. Операційною системою виступає вбудована

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

датчиків температури та вологості. Система заснована на гнучкій архітектурі, що дозволяє користувачам самостійно конфігурувати набір сенсорів відповідно до потреб.



Рисунок 1.3 – Ecowitt HP2551 [3]

Система використовує IEEE 802.11 (Wi-Fi) протокол для передачі даних до хмарних сервісів та власний радіопротокол 915 МГц для комунікації з сенсорами. Протоколи керування доступом реалізовані через WPA2-PSK шифрування для Wi-Fi та токени автентифікацію для API доступу. Інтеграція з зовнішніми платформами здійснюється через WeatherCloud API та Weather Underground протокол, а також підтримку MQTT для IoT інтеграцій. Методи обробки даних включають локальну обробку на базі ARM Cortex-A7 процесора, SQLite базу даних для зберігання історичних даних та автоматичний експорт у CSV форматі. Операційна система базується на Linux-ядрі з власною прошивкою, що забезпечує стабільну роботу розподіленої архітектури з множинними сенсорами.

Технологічна база включає підтримку анемометра, ультрафіолетових та сонячних датчиків з можливістю одночасного моніторингу до 16 різних параметрів. Система забезпечує гнучкість конфігурації, високу точність вимірювань та багатоплатформенну інтеграцію з можливістю експорту даних для подальшого аналізу.

Німецька система TFA Dostmann VIEW Breeze представляє підхід, орієнтований на дистанційне управління та прогнозування. Архітектура включає базову станцію з кольоровим дисплеєм та зовнішній датчик 5-в-1, що інтегрує моніторинг температури, вологості, швидкості та напрямку вітру, кількості опадів.



Рисунок 1.4 – TFA Dostmann VIEW Breeze [4]

Система функціонує на базі IEEE 802.11b/g/n (Wi-Fi) протоколу для підключення до інтернету та власного радіопротоколу 868 МГц для зв'язку з зовнішніми сенсорами. Протоколи керування доступом включають WPA2 шифрування для бездротового з'єднання та OAuth 2.0 автентифікацію для хмарних сервісів. Інтеграція з голосовими помічниками реалізована через Amazon Alexa

					КвРКІ 210106.21.01.42.ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

Skills API та Google Assistant Actions SDK. Методи обробки даних базуються на REST API архітектурі для мобільного додатку TFA VIEW, хмарному зберіганні даних на серверах AWS та JSON форматі передачі інформації. Операційна система базової станції використовує вбудовану Linux-систему з веб-сервером для віддаленого доступу, що забезпечує гібридну архітектуру з локальним відображенням та хмарною синхронізацією.

Багатофункціональна система Bresser 7-in-1 Weather Center демонструє комплексний підхід до метеомоніторингу. Система заснована на інтегрованому зовнішньому сенсорі, що здійснює одночасне вимірювання семи параметрів: температури, вологості, швидкості та напрямку вітру, опадів.



Рисунок 1.5 – Bresser 7-in-1 Weather Center [5]

Система використовує комбінований підхід передачі даних через IEEE 802.11 (Wi-Fi) протокол для хмарної синхронізації та власний радіопротокол 868 МГц для зв'язку між сенсором та базовою станцією. Протоколи керування доступом включають WPA2-Enterprise шифрування для корпоративних мереж та базову

					КВРКІ 210106.21.01.42.ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

HTTP автентифікацію для веб-доступу. Інтеграція з зовнішніми системами реалізована через підтримку Weather Underground API, WeatherCloud протоколу та MQTT для IoT платформ.

Методи обробки даних базуються на вбудованому 32-бітному ARM мікроконтролері з власним алгоритмом калібрування, FAT32 файловою системою для SD-карт та автоматичним архівуванням даних у CSV та XML форматах. Операційна система використовує власну вбудовану прошивку з підтримкою багатозадачності, що забезпечує стабільну роботу розподіленої архітектури з локальним та хмарним зберіганням.

Архітектура системи підтримує підключення до семи додаткових датчиків через протокол розширення та забезпечує автономну роботу від батарейного живлення з оптимізованим енергоспоживанням. Система забезпечує детальний аналіз погодних умов, надійне дублювання даних через локальне та хмарне зберігання, можливість масштабування функціоналу шляхом додавання модулів.

1.2 Загальні принципи функціонування та методи вимірювання метеостанцій

Функціонування сучасних метеорологічних станцій базується на комплексі фізичних принципів та технологічних рішень, що забезпечують точне вимірювання атмосферних параметрів. Розуміння цих принципів є ключовим для правильного використання та інтерпретації отриманих даних, а також для розробки нових технологічних рішень у сфері метеорологічного моніторингу.

Температурні сенсори в метеостанціях працюють на основі різних фізичних явищ, кожне з яких має свої особливості застосування та характеристики точності. Найпоширеніші термочутливі резистори (термістори) змінюють свій електричний опір залежно від температури за експоненціальним законом, що описується рівнянням Стейнхарта-Харта. Ці датчики характеризуються високою чутливістю, особливо в діапазоні навколишніх температур, проте мають нелінійну характеристику, що потребує складних алгоритмів лінеаризації.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

Платинові термометри опору (RTD) демонструють лінійну залежність опору від температури, що забезпечує високу точність вимірювань у широкому діапазоні від -200°C до $+850^{\circ}\text{C}$. Стандарт PT100 використовує платиновий резистор з номінальним опором 100 Ом при 0°C , що змінюється приблизно на 0,385 Ом на кожен градус Цельсія.

Термопари генерують електричну напругу на основі ефекту Зеебека при різниці температур між спаями різнорідних металів, причому найпоширенішими є термопари типу К (хромель-алюмель), J (залізо-константан) та Т (мідь-константан), кожна з яких має специфічний діапазон робочих температур та характеристики стабільності.

Вимірювання вологості здійснюється переважно за допомогою ємнісних та резистивних гігрометрів, принцип роботи яких базується на фізико-хімічних властивостях гігроскопічних матеріалів. Ємнісні датчики використовують полімерні діелектрики, діелектрична проникність яких змінюється при поглинанні водяної пари з навколишнього середовища.

Сучасні ємнісні сенсори використовують полімери на основі целюлози або синтетичних матеріалів, що забезпечує швидку реакцію на зміни вологості та високу стабільність показників протягом тривалого періоду експлуатації. Резистивні сенсори базуються на зміні електричного опору гігроскопічних матеріалів при зміні вологості навколишнього середовища, де найчастіше використовуються полімерні плівки або керамічні матеріали з додаванням провідних домішок.

Температурна компенсація цих датчиків здійснюється через вбудовані температурні сенсори та математичні алгоритми корекції, оскільки чутливість гігроскопічних матеріалів до вологості значно залежить від температури навколишнього середовища.

Барометричні вимірювання атмосферного тиску реалізуються через п'єзорезистивні датчики, де механічна деформація мембрани під дією атмосферного тиску призводить до зміни опору напівпровідникових елементів, що

					КвРКІ 210106.21.01.42.ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

сформовані у вигляді містків Вітстона для підвищення чутливості та зменшення температурного дрейфу.

Альтернативно використовуються ємнісні барометри, де зміна тиску викликає деформацію діафрагми і, відповідно, зміну ємності конденсатора, утвореного між рухомою діафрагмою та нерухожими електродами. Сучасні MEMS-барометри поєднують високу точність (до $\pm 0,1$ гПа) з компактними розмірами та низьким енергоспоживанням, що робить їх ідеальними для портативних метеостанцій.

Калібрування барометричних датчиків здійснюється відносно еталонних значень атмосферного тиску на рівні моря (1013,25 гПа при 15°C) з урахуванням висоти розташування станції над рівнем моря та локальних атмосферних умов.

Анемометричні системи для вимірювання швидкості вітру функціонують за різними принципами, кожен з яких має специфічні переваги та обмеження застосування. Чашкові анемометри перетворюють обертальний рух, спричинений вітровим потоком, у електричні імпульси через магнітні датчики Холла або оптичні енкодери, де калібрування здійснюється через емпіричні коефіцієнти, що враховують аеродинамічні характеристики чашок та механічне тертя підшипників.

Ультразвукові анемометри вимірюють час проходження ультразвукових імпульсів між передавачами, що змінюється залежно від швидкості та напряму повітряного потоку, причому використання кількох пар передавач-приймач дозволяє визначати не лише швидкість, але й тривимірний вектор швидкості вітру.

Термоанемометри базуються на принципі конвективного охолодження нагрітого елемента повітряним потоком, де інтенсивність охолодження прямо пропорційна швидкості вітру, проте цей метод потребує значної кількості енергії та чутливий до температурних змін навколишнього середовища.

Лазерні доплерівські анемометри використовують ефект Доплера для вимірювання швидкості частинок у повітряному потоці, що забезпечує надзвичайно високу точність вимірювань, проте їх застосування обмежене високою вартістю та складністю обслуговування.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

Плювіометричні системи для реєстрації опадів базуються на принципі накопичення дощової води в каліброваних ємностях з подальшою реєстрацією кількості через різні методи детекції. Механічні дощоміри використовують балансірні системи, де накопичення певного об'єму води (зазвичай 0,1 або 0,2 мм опадів) спричиняє перекидання механізму та генерацію електричного імпульсу, що реєструється лічильником.

Оптичні дощоміри застосовують інфрачервоне випромінювання для детекції крапель води, що перетинають світловий промінь, причому аналіз інтенсивності та тривалості перерв у променях дозволяє визначати не лише кількість, але й інтенсивність опадів.

П'єзоелектричні дощоміри реєструють механічні удари крапель дощу об чутливу поверхню, перетворюючи кінетичну енергію крапель у електричні сигнали, інтенсивність яких корелює з розміром крапель та інтенсивністю опадів.

Ваговий метод вимірювання опадів використовує прецизійні ваги для безперервного зважування накопиченої води, що забезпечує найвищу точність вимірювань, особливо для твердих опадів (снігу, граду), проте потребує складних систем температурної компенсації та захисту від вітрового навантаження.

Сучасні метеостанції використовують багаторівневі системи калібрування для забезпечення точності вимірювань, що включають заводське калібрування, періодичне калібрування в експлуатації та автоматичне калібрування за еталонними зразками. Температурна компенсація датчиків атмосферного тиску здійснюється через математичні алгоритми, що враховують залежність чутливості сенсора від температури навколишнього середовища, використовуючи поліноміальні функції другого або третього порядку для корекції показників. Автокалібрування відносної вологості проводиться з використанням еталонних значень при відомих температурних умовах, часто використовуючи насичені розчини солей (наприклад, NaCl для 75% вологості або MgCl₂ для 33% вологості) як еталони відносної вологості. Крос-калібрування між різними датчиками однієї станції дозволяє виявляти систематичні похибки та дрейф характеристик окремих

					КвРКІ 210106.21.01.42.ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

сенсорів через порівняння з розрахунковими значеннями, отриманими з інших параметрів.

Компенсація систематичних похибок реалізується через статистичні методи обробки даних, включаючи фільтрацію викидів за критерієм Граббса або методом міжквартильного розмаху, згладжування короточасних флуктуацій за допомогою фільтрів Калмана або експоненційного згладжування, та кореляційний аналіз між різними параметрами для виявлення аномальних показників. Це дозволяє підвищити достовірність метеорологічних даних та усунути вплив зовнішніх завад, таких як електромагнітні перешкоди, вібрації, температурні шоки або механічні деформації корпусу станції. Адаптивні алгоритми компенсації використовують методи машинного навчання для автоматичного виявлення та корекції систематичних похибок на основі накопичених історичних даних та порівняння з сусідніми метеостанціями або супутниковими даними.

Метеорологічні станції використовують різноманітні енергетичні рішення для забезпечення безперервної роботи в різних кліматичних умовах та географічних локаціях. Батарейне живлення з літєвими елементами забезпечує тривалу автономність за рахунок низького саморозряду (менше 2% на рік) та стабільної напруги протягом більшої частини розрядного циклу, причому сучасні літій-тіонілхлоридні батареї можуть працювати при температурах від -55°C до $+85^{\circ}\text{C}$. Сонячні панелі з акумуляторними системами дозволяють створювати повністю автономні станції для віддалених локацій, де монокристалічні або полікристалічні фотоелементи забезпечують ефективність перетворення сонячної енергії на рівні 15-20%, а контролери заряду з MPPT-технологією максимізують ефективність зарядки акумуляторів. Вітрові генератори можуть використовуватися як додаткове джерело енергії у регіонах з постійними вітрами, проте їх застосування обмежене механічною складністю та потребою в регулярному обслуговуванні.

Оптимізація енергоспоживання досягається через циклічні режими роботи, де датчики активуються лише під час вимірювань за заздалегідь запрограмованими

					КвРКІ 210106.21.01.42.ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

інтервалами (зазвичай кожні 1-10 хвилин), а в проміжках часу переходять у режим низького споживання з током спокою менше 1 мкА. Інтелектуальне управління живленням включає моніторинг рівня заряду акумуляторів через вбудовані кулонометри або вольтметри, автоматичне відключення некритичних функцій при низькому заряді, та динамічне масштабування частоти вимірювань залежно від доступної енергії. Технології збирання енергії з навколишнього середовища (energy harvesting) включають термоелектричні генератори, що використовують різницю температур між навколишнім повітрям та землею, п'єзоелектричні елементи для перетворення механічних вібрацій у електричну енергію, та радіочастотні збирачі енергії для використання енергії радіохвиль у густонаселених районах.

Телеметричні системи метеостанцій використовують різні протоколи передачі даних залежно від відстані, обсягу інформації та доступної інфраструктури зв'язку. Короткодистанційний зв'язок реалізується через радіочастотні канали в діапазонах 315 МГц, 433 МГц, 868 МГц (Європа) або 915 МГц (США) з модуляцією FSK (Frequency Shift Keying), OOK (On-Off Keying) або LoRa для підвищення дальності та надійності передачі при низькому енергоспоживанні. Протокол LoRaWAN забезпечує двосторонній зв'язок на відстані до 15 км у сільській місцевості з енергоспоживанням, що дозволяє працювати від батарей протягом кількох років. Zigbee та Thread протоколи використовуються для створення mesh-мереж датчиків з автоматичним маршрутизуванням та самовідновленням мережевої топології.

Дальня передача даних здійснюється через стільникові мережі GSM/GPRS/3G/4G/5G або супутникові канали зв'язку для станцій у віддалених регіонах без наземної інфраструктури зв'язку. Стільникові модеми з підтримкою LTE Cat-M1 або NB-IoT забезпечують енергоефективну передачу невеликих обсягів даних з можливістю роботи від батарей протягом кількох років. Супутникові системи зв'язку, такі як Iridium, Globalstar або нові низькоорбітальні констеляції, забезпечують глобальне покриття, проте характеризуються високою вартістю передачі даних та значним енергоспоживанням. Протоколи синхронізації

					КвРКІ 210106.21.01.42.ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

забезпечують узгодження часових міток вимірювань через GPS-приймачі з точністю до мікросекунд або мережеві протоколи NTP (Network Time Protocol) з точністю до мілісекунд при наявності інтернет-з'єднання.

Буферизація даних у локальній пам'яті дозволяє зберігати інформацію під час тимчасових перерв у зв'язку, використовуючи енергонезалежну Flash-пам'ять або EEPROM з ресурсом перезапису 100,000-1,000,000 циклів. Алгоритми стиснення зменшують обсяг переданих даних без втрати критичної інформації, використовуючи методи дельта-кодування для повільно змінюваних параметрів, Huffman-кодування для текстових даних або спеціалізовані алгоритми стиснення часових рядів. Протоколи забезпечення цілісності даних включають контрольні суми CRC-16 або CRC-32 для виявлення помилок передачі, механізми підтвердження доставки (ACK/NACK) та автоматичне повторення передачі (ARQ) при виявленні помилок.

Первинна обробка метеорологічних даних включає цифрову фільтрацію для усунення високочастотних завад через фільтри низьких частот Баттерворта або Чебишева, медіанні фільтри для усунення імпульсних завад, та статистичну обробку для виявлення аномальних значень за критеріями Граббса, Діксона або методом міжквартильного розмаху. Алгоритми скользячої медіани та експоненційного згладжування з адаптивними коефіцієнтами дозволяють отримати стабільні показники при збереженні швидкості реакції на реальні зміни погодних умов, причому коефіцієнти згладжування можуть динамічно змінюватися залежно від швидкості зміни параметрів або часу доби.

Валідація даних здійснюється через перехресну перевірку показників різних датчиків з використанням фізичних співвідношень між метеопараметрами, порівняння з історичними даними через статистичні тести на відповідність нормальному розподілу або сезонним трендам, та аналіз фізичної узгодженості параметрів через рівняння стану атмосфери. Наприклад, розрахунок точки роси на основі температури та вологості за формулою Магнуса дозволяє перевірити коректність цих вимірювань, а співвідношення між атмосферним тиском та

					КвРКІ 210106.21.01.42.ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

висотою за барометричною формулою може використовуватися для валідації барометричних даних. Алгоритми виявлення аномалій використовують методи машинного навчання, такі як ізоляційні ліси, автоенкодери або статистичні методи на основі Z-score або модифікованого Z-score для автоматичного виявлення підозрілих значень.

Прогностичні алгоритми використовують методи часових рядів, такі як ARIMA (AutoRegressive Integrated Moving Average), експоненційне згладжування Холта-Вінтерса для сезонних даних, або сучасні методи машинного навчання, включаючи рекурентні нейронні мережі (RNN), LSTM (Long Short-Term Memory) мережі або трансформери для передбачення короткострокових змін погодних умов на основі поточних тенденцій та історичних даних. Це дозволяє метеостанціям не лише фіксувати поточний стан атмосфери, але й надавати користувачам попередження про очікувані зміни погоди з горизонтом прогнозування від кількох годин до кількох днів, інтегруючи локальні вимірювання з регіональними метеорологічними моделями та супутниковими даними.

1.3 Висновки до першого розділу

У першому розділі дипломної роботи було проведено ґрунтовний аналіз предметної області сучасних метеорологічних систем, розглянуто тенденції розвитку, класифікацію та основні технічні рішення у сфері автоматизованого моніторингу довкілля.

Встановлено, що з розвитком мікроелектроніки та інформаційних технологій метеостанції еволюціонували від громіздких і складних систем до компактних, доступних та високофункціональних пристроїв, які інтегруються у системи автоматизації та Інтернету речей (IoT).

					КвРКІ 210106.21.01.42.ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ "ПОГОДНА СТАНЦІЯ НА ESP32"

2.1 Вимоги до програмно-апаратного засобу

У попередньому розділі було проведено аналіз предметної області та розглянуто існуючі рішення для моніторингу метеорологічних даних. Визначено основні параметри, які вимірюються сучасними погодними станціями, а також виявлено ключові недоліки деяких існуючих підходів.

На основі цього було сформульовано наступні вимоги до програмно-апаратного засобу "Погодна станція на ESP32" із використанням платформи Blynk для відображення даних у режимі реального часу.

Система повинна забезпечувати точне вимірювання температури навколишнього середовища з похибкою не більше $\pm 0.5^{\circ}\text{C}$ у діапазоні від -10°C до $+50^{\circ}\text{C}$. Частота оновлення температурних показників має становити не рідше одного разу на хвилину для забезпечення актуальності інформації. Пристрій має здійснювати моніторинг відносної вологості повітря з точністю $\pm 3\%$ у повному діапазоні від 0% до 100% RH.

Контроль якості повітря реалізується через вимірювання концентрації шкідливих газів та пилу з відображенням результатів у вигляді індексу якості повітря. Система повинна реагувати на підвищення рівня забруднення та сигналізувати про потенційно небезпечні умови.

Передача даних у реальному часі повинна здійснюватись через стабільне Wi-Fi з'єднання з хмарною платформою Blynk із затримкою не більше 30 секунд. Це дозволяє користувачам отримувати актуальну інформацію про стан навколишнього середовища незалежно від їх місцезнаходження.

Система автоматичного сповіщення має інформувати користувача про критичні зміни параметрів середовища через push-повідомлення та електронну пошту. Користувач може самостійно налаштовувати порогові значення для кожного параметра відповідно до своїх потреб.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Відображення інформації повинно бути реалізоване через інтуїтивно зрозумілий веб-інтерфейс та мобільний додаток з представленням поточних та історичних даних у вигляді графіків, діаграм та цифрових індикаторів. Інтерфейс повинен бути адаптивним та зручним для використання на різних пристроях.

Пристрій має підтримувати стабільне бездротове з'єднання стандарту Wi-Fi 802.11 b/g/n з можливістю автоматичного відновлення зв'язку після перерв. Споживання електроенергії не повинно перевищувати 3 Вт для забезпечення економічної експлуатації при постійному підключенні до мережі живлення.

Налаштування параметрів системи має здійснюватись через зручний користувацький інтерфейс з можливістю конфігурування інтервалів вимірювань, порогових значень для сповіщень та інших системних параметрів без необхідності програмування або технічних знань.

2.2 Структура програмно апаратного засобу.

Погодна станція на ESP32 складається з взаємопов'язаних апаратної та програмної частин, які функціонують у тісній інтеграції для забезпечення ефективного збору і надійної передачі даних про основні метеорологічні параметри - температуру, вологість та якість повітря навколишнього середовища.

Основою всієї системи є потужний мікроконтролер ESP32, який виконує ключову роль у процесі обробки інформації - він безперервно отримує дані з підключених датчиків через відповідні інтерфейси, здійснює їх первинну обробку та форматування, після чого відправляє оброблену інформацію до хмарної платформи Blynk через стабільне Wi-Fi з'єднання для подальшого зберігання та візуалізації.

Для забезпечення наочного та зрозумілого представлення принципів роботи всієї системи моніторингу було розроблено три основні блок-схеми :

Перша схема показує загальну структуру пристрою та відображає принципи організації всієї системи.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

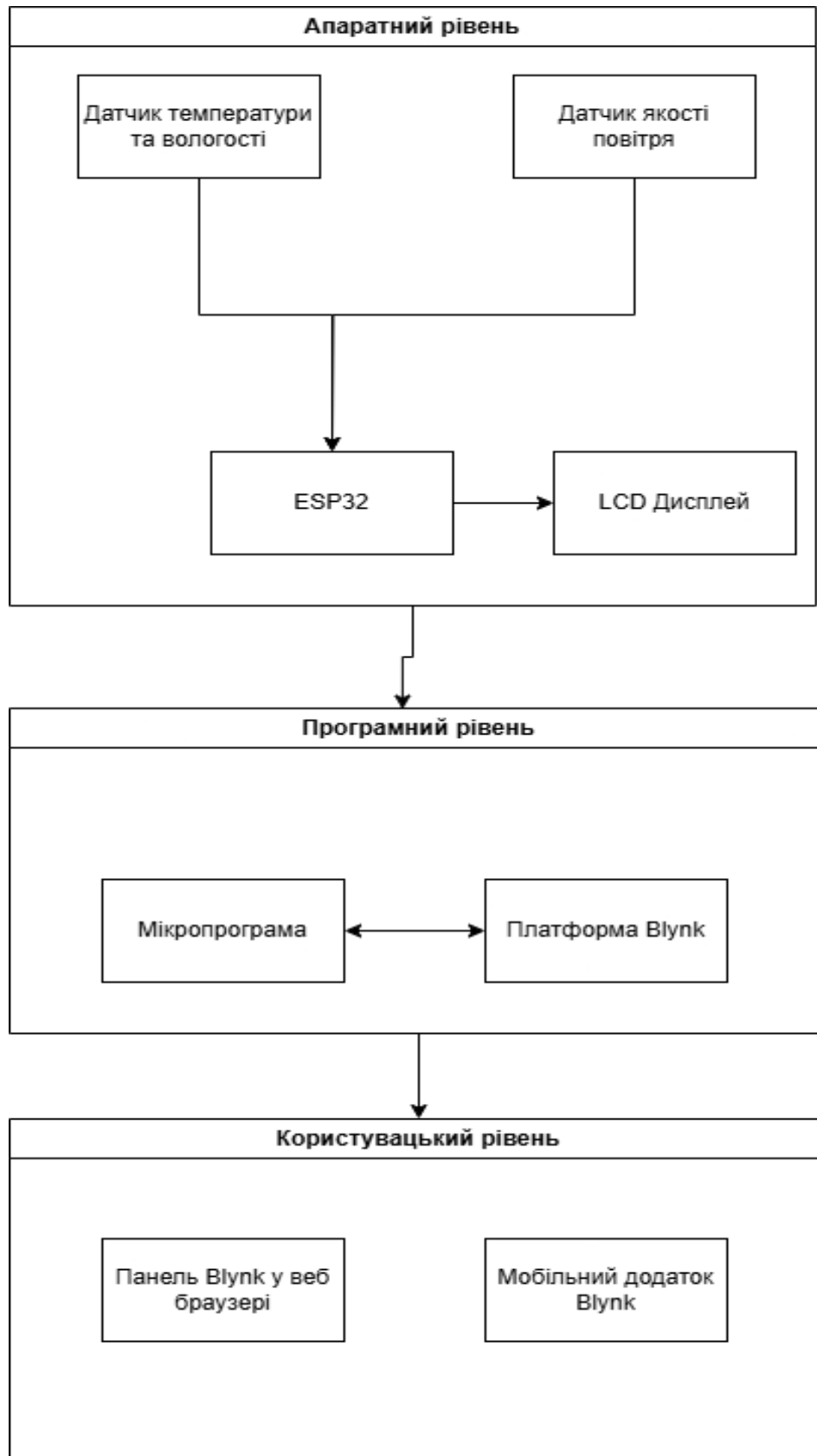


Рисунок 2.1 – Загальна структура пристрою.

Друга схема описує алгоритм роботи ESP32 від початку роботи до передачі даних. Вона включає процес підключення до Wi-Fi, з'єднання з Blynk сервером, циклічне зчитування датчиків .

Блок-схему можна умовно розділити на дві частини :

1. Ініціалізація системи – на цьому етапі відбувається увімкнення системи, підключення до мережі Wi Fi та платформи Blynk.

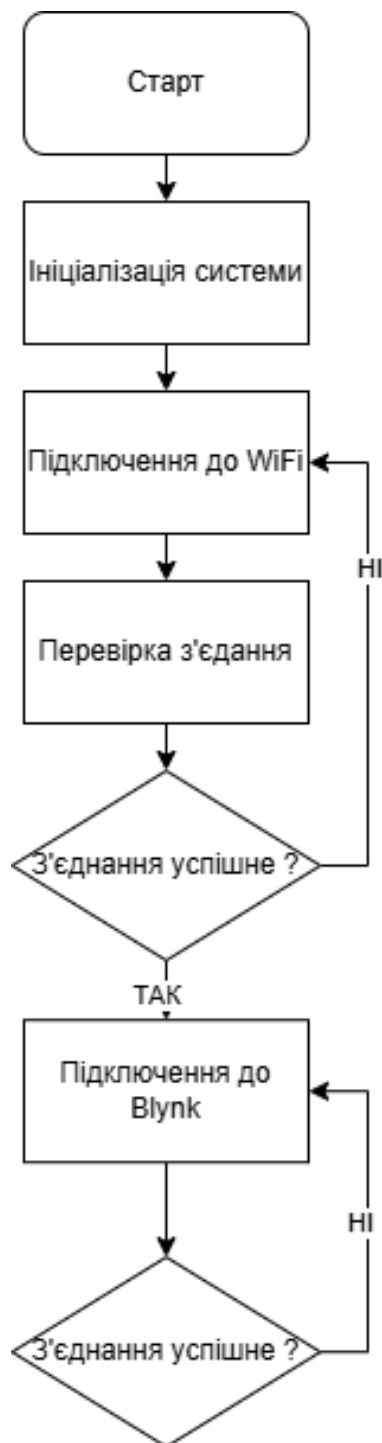


Рисунок 2.2 – Схема ініціалізації системи

2. Основний цикл – на цьому етапі система зчитує дані з датчиків , обробляє їх , відображає на дисплеї та надсилає на сервер Blynk.

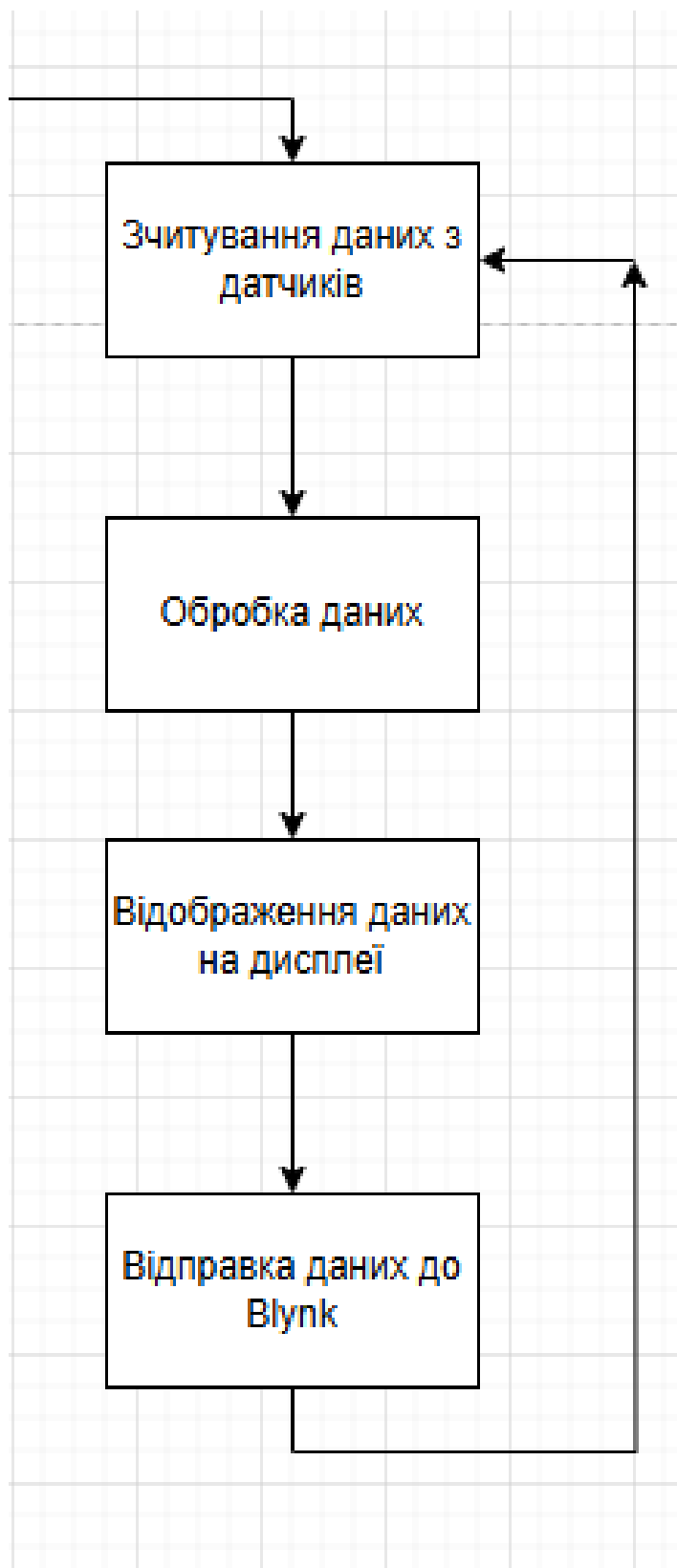


Рисунок 2.3 – Основний цикл

Зм.	Арк.	№ докум.	Підпис	Дата

Третя схема описує роботу Vlynk платформи - як сервер обробляє отримані дані, перевіряє їх на перевищення встановлених меж, генерує сповіщення користувачу та оновлює графіки в реальному часі.



Рисунок 2.4 – Схема роботи Vlynk

Також було розроблено електричну схему пристрою, на ній зображено всі елементи, їх підключення до ESP32, а також підключення лінії 5v.

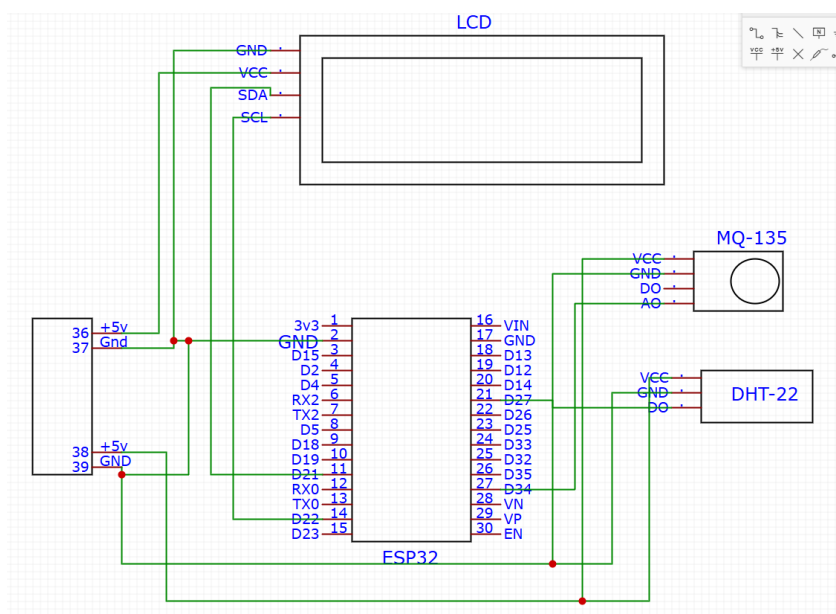


Рисунок 2.5 – схема електрична.

Представлена схема розроблена у середовищі автоматизованого проектування EasyEDA та побудована за принципом централізованої архітектури з використанням мікроконтролера ESP32 як основного обчислювального вузла.

Мікроконтролер ESP32 представлений у схемі у вигляді функціонального блоку з детальною специфікацією призначення кожного виводу. У верхній частині схеми розташований LCD-дисплей, який функціонує як підсистема візуалізації даних. Підключення дисплея здійснюється через послідовний інтерфейс I2C, що забезпечує ефективне використання портів вводу-виводу мікроконтролера. Лінії SDA (Serial Data) та SCL (Serial Clock) з'єднують дисплей з відповідними GPIO портами 21 та 22 мікроконтролера, формуючи двопровідну шину обміну даними.

Сенсорна частина системи представлена двома спеціалізованими датчиками. Датчик DHT-22 реалізує функцію вимірювання температурно-вологісних параметрів навколишнього середовища. Його підключення здійснюється через цифровий інтерфейс до одного з портів GPIO мікроконтролера, що забезпечує надійну передачу цифрових даних з високою завадостійкістю.

Зм.	Арк.	№ докум.	Підпис	Дата

Датчик якості повітря MQ-135 інтегрований у схему через аналоговий інтерфейс, підключений до аналого-цифрового перетворювача (АЦП) мікроконтролера. Така конфігурація дозволяє здійснювати безперервний моніторинг концентрації газоподібних речовин у повітрі з можливістю кількісної оцінки їх вмісту.

Схема електроживлення реалізована за принципом багаторівневого живлення з використанням напруги 5v що забезпечує функціонування датчика MQ-135, який потребує підвищеної напруги для стабільної роботи нагрівального елемента, а також для живлення мікроконтролера ESP32, LCD-дисплея та датчика DHT-22, що відповідає їх технічним специфікаціям.

Загальна точка схеми (GND) об'єднує всі компоненти в єдину електричну систему, забезпечуючи стабільний потенціал відліку для всіх електричних сигналів.

2.3 Порівняльний аналіз мікроконтролерних систем

Наразі на ринку існує три найпопулярніші мікроконтролери ESP32, Arduino та Raspberry Pi пісо. Проведемо порівняльний аналіз даних систем для вибору найбільш оптимальної для створення погодної станції. Спочатку розглянемо мікроконтролерну систему ESP32.

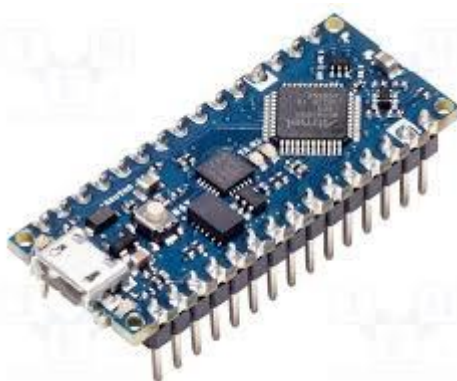


Рисунок 2.1 – ESP32 [6]

					КвРКІ 210106.21.01.42.ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

В основі архітектури ESP32 лежить двоядерний процесор Tensilica Xtensa LX6, що функціонує на частоті до 240 МГц. Пристрій оснащений 520 КБ оперативної пам'яті типу SRAM та передбачає можливість підключення зовнішньої флеш-пам'яті об'ємом до 16 МБ.

Визначальною характеристикою ESP32 є інтегрована підтримка бездротових інтерфейсів, що включає Wi-Fi з підтримкою режимів STA/AP/STA+AP та Bluetooth в версіях класичного Bluetooth і Bluetooth Low Energy (BLE). Чіп також оснащений багатим набором периферійних інтерфейсів: 34 програмованими GPIO виводами, 18 каналами 12-бітного АЦП, 2 каналами 8-бітного ЦАП, інтерфейсами SPI, I2S, I2C, UART, CAN, інфрачервоним передавачем, підтримкою ємнісних сенсорних датчиків та вбудованим датчиком Холла.

Архітектурна конструкція ESP32 передбачає наявність спеціалізованих апаратних блоків для криптографічних операцій, що підтримують шифрування AES, SHA-2, RSA, ECC, а також апаратного генератора випадкових чисел, що є критичним для забезпечення безпеки IoT-пристроїв. Система працює в температурному діапазоні від -40°C до +125°C, що робить її придатною для використання в екстремальних умовах.

ESP32 характеризується низьким енергоспоживанням, що особливо важливо для автономних пристроїв: у режимі глибокого сну споживання складає лише 10 мкА. Архітектура ESP32 оптимізована для ефективного керування енергоспоживанням завдяки наявності RTC контролера та різних режимів сну, що значно подовжує термін експлуатації пристроїв з батарейним живленням.

Не менш популярною є мікроконтролерна систем Arduino. Це сімейство мікроконтролерних плат, більшість з яких базується на 8-бітних мікроконтролерах ATmega виробництва компанії Microchip Technology (раніше Atmel). Флагманська модель Arduino Uno R3, що стала референтною для екосистеми, використовує мікроконтролер ATmega328P з тактовою частотою 16 МГц, 2 КБ оперативної пам'яті SRAM та 32 КБ флеш-пам'яті для програмного коду.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

В архітектурі Arduino важливу роль відіграє стандартизоване розташування пінів, що забезпечує сумісність з широким спектром модулів розширення, відомих як "шилди" (shields). Це дозволяє ефективно масштабувати функціональність базової плати з мінімальними зусиллями.



Рисунок 2.2 – Arduino [7]

Типова конфігурація Arduino Uno включає 14 цифрових входів/виходів (з яких 6 можуть використовуватися як ШІМ-виходи) та 6 аналогових входів, що працюють з 10-бітним АЦП. Плата містить інтерфейси UART, SPI, I2C для комунікації з периферійними пристроями.

Важливою архітектурною особливістю Arduino є наявність вбудованого завантажувача (bootloader), який значно спрощує процес програмування мікроконтролера через USB-інтерфейс без необхідності використання зовнішнього програматора. Це рішення суттєво знизило поріг входу для початківців у сфері електроніки.

З часом сімейство Arduino розширилося, включивши більш потужні моделі, такі як Arduino Mega (на основі ATmega2560 з 256 КБ флеш-пам'яті), Arduino Zero (з 32-бітним ARM Cortex-M0+ процесором на 48 МГц), Arduino Due (з 32-бітним ARM Cortex-M3 процесором на 84 МГц). Ці вдосконалені моделі пропонують більше пам'яті, вищу обчислювальну потужність та розширений набір периферійних пристроїв, проте зберігають сумісність з базовою екосистемою Arduino на рівні програмного забезпечення.

Ще однією популярною системою є Raspberry Pi, що являє собою повноцінний одноплатний комп'ютер, архітектурно суттєво відмінний від ESP32 та Arduino. Актуальна модель Raspberry Pi піко побудована на основі 64-бітного чотириядерного процесора ARM Cortex-A72 з тактовою частотою 1,5 ГГц. Варіанти комплектації передбачають наявність 2, 4 або 8 ГБ оперативної пам'яті типу LPDDR4.

Архітектура Raspberry Pi піко включає графічний процесор VideoCore VI з підтримкою відтворення відео 4K, звуковий процесор, модуль управління живленням та численні інтерфейси вводу-виводу. Зберігання операційної системи та даних забезпечується картою microSD, а додаткова пам'ять може підключатися за допомогою USB або мережевих сховищ.

Raspberry Pi піко має багатий набір інтерфейсів, що включає: 40 GPIO виводів, підтримку I2C, SPI, UART, PWM, USB 3.0 та USB 2.0 порти, Gigabit Ethernet, Wi-Fi 802.11ac, Bluetooth 5.0, два мікро-HDMI порти з підтримкою двох 4K моніторів, композитний відеовихід, аудіо вихід та роз'єм для камери. Присутність USB-C порту для живлення з можливістю використання джерела на 5В/3А розширює можливості енергопостачання.



Рисунок 2.3 – Raspberry pi [8]

Ключовою архітектурною відмінністю Raspberry Pi піко від ESP32 та Arduino є підтримка повноцінних операційних систем, зокрема, спеціально адаптованого

					КвРКІ 210106.21.01.42.ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

дистрибутиву Linux - Raspberry Pi OS (раніше відомого як Raspbian), а також Ubuntu, Windows 10 IoT Core, Android та інших. Це дозволяє використовувати на платформі стандартне програмне забезпечення з екосистеми ПК, а також програмувати пристрій з використанням широкого спектру мов програмування.

2.4 Програмне забезпечення та середовища розробки

Проведемо порівняльний аналіз середовищ розробки для розглянутих вище мікроконтролерних систем.

Програмування ESP32 може здійснюватися за допомогою декількох програмних фреймворків та середовищ розробки, що демонструє гнучкість цієї платформи. Офіційний ESP-IDF (IoT Development Framework) від Espressif представляє собою повноцінний набір інструментів для розробки на мові програмування C, оптимізований для максимального використання апаратних можливостей ESP32.

Фреймворк базується на FreeRTOS - операційній системі реального часу, що забезпечує ефективне управління задачами, семафорами, таймерами та іншими системними ресурсами. ESP-IDF надає розробникам доступ до всіх низькорівневих функцій мікроконтролера та дозволяє досягти максимальної продуктивності.

Для спрощення процесу розробки ESP32 інтегрований в екосистему Arduino через Arduino Core для ESP32, що дозволяє використовувати звичне середовище Arduino IDE та синтаксис мови, зберігаючи при цьому доступ до специфічних функцій ESP32. Ця інтеграція відкриває платформу для широкого кола розробників, що мають досвід роботи з Arduino, забезпечуючи плавний перехід до більш потужної системи.

Альтернативними середовищами для розробки ESP32 є PlatformIO - кросплатформений інструмент, що інтегрується з Visual Studio Code, а також підтримка мови програмування MicroPython, що дозволяє писати код на інтерпретованій мові Python безпосередньо на мікроконтролері. MicroPython

					КВРКІ 210106.21.01.42.ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

робить ESP32 доступним для розробників, які віддають перевагу високорівневим мовам програмування, жертвуючи певною мірою продуктивності заради зручності та швидкості розробки.

ESP32 також підтримує LUA через проєкт NodeMCU, що є ще одним варіантом для швидкого прототипування. Крім того, існує можливість використання JavaScript через фреймворк Espruino, що розширює аудиторію потенційних розробників.

Для забезпечення безпеки та оновлення прошивки "по повітрю" в ESP32 реалізована підтримка протоколів TLS/SSL, що забезпечує шифрування при передачі даних, а також механізми безпечного оновлення ПЗ через Wi-Fi, що є критичним для IoT-пристроїв, розміщених у віддалених локаціях.

Натомість екосистема програмного забезпечення Arduino побудована навколо принципу максимальної доступності та простоти використання. Основою цієї екосистеми є інтегроване середовище розробки Arduino IDE, що забезпечує єдиний інтерфейс для написання коду, компіляції та завантаження програм на різні моделі мікроконтролерів. Arduino IDE пропонує мінімалістичний інтерфейс з базовими функціями редагування коду, швидким доступом до бібліотек та простим механізмом компіляції та завантаження.

Мова програмування Arduino базується на C++ з певними спрощеннями і доповненнями, що робить її доступною для початківців при збереженні потужності C++ для досвідчених розробників. Процес компіляції програм Arduino відбувається за допомогою компілятора GCC, а бінарні файли завантажуються на мікроконтролер через послідовний порт.

Важливою складовою екосистеми Arduino є обширна бібліотека програмних компонентів, що забезпечують абстракцію низькорівневих операцій та спрощують роботу з різноманітними датчиками, дисплеями, модулями зв'язку тощо. Ці бібліотеки дозволяють швидко інтегрувати різноманітні функції без необхідності глибокого розуміння низькорівневих аспектів взаємодії з периферійними пристроями.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

Для більш досвідчених розробників існують альтернативні підходи до програмування Arduino, включаючи використання AVR-GCC напряму без Arduino IDE, що дозволяє отримати більший контроль над процесом компіляції та оптимізації коду. Також можливе використання середовища PlatformIO, що надає більш професійний набір інструментів з підтримкою версійного контролю, автодоповнення коду та інтеграцією з популярними редакторами, такими як Visual Studio Code.

Аналізуючи програмування Arduino, важливо відзначити відсутність операційної системи як такої. Програми для Arduino виконуються безпосередньо на "голому залізі" (bare metal), що з одного боку спрощує архітектуру системи і мінімізує накладні витрати на управління ресурсами, а з іншого, покладає на розробника відповідальність за ефективне управління часом та ресурсами мікроконтролера.

Програмне забезпечення Raspberry Pi кардинально відрізняється від ESP32 та Arduino, оскільки ця платформа функціонує під управлінням повноцінних операційних систем. Стандартною операційною системою для Raspberry Pi є Raspberry Pi OS (раніше Raspbian) - оптимізований дистрибутив Linux на базі Debian, спеціально адаптований для архітектури ARM та апаратних особливостей Raspberry Pi. Ця операційна система надає графічний інтерфейс користувача та повноцінне середовище для виконання різноманітних програм.

Крім Raspberry Pi OS, плата підтримує інші дистрибутиви Linux (Ubuntu, Fedora, Arch Linux ARM), спеціалізовані медіа-центри на основі OSMC або LibreELEC, а також обмежені версії Windows 10 IoT Core та Android. Таке різноманіття операційних систем робить Raspberry Pi універсальною платформою для різних сценаріїв використання.

Розробка програмного забезпечення для Raspberry Pi може здійснюватися з використанням практично будь-якої мови програмування, підтримуваної в Linux-середовищі: Python, C/C++, Java, JavaScript (Node.js), Ruby, Perl, Bash та багатьох інших. Особливо популярним є використання Python, що активно підтримується

					КвРКІ 210106.21.01.42.ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

спільнотою Raspberry Pi Foundation через навчальні матеріали та бібліотеки для роботи з GPIO та периферійними пристроями.

Для низькорівневого доступу до апаратних можливостей Raspberry Pi існують спеціалізовані бібліотеки, такі як RPi.GPIO, pigpio, WiringPi (для Python та C), що забезпечують програмний інтерфейс для роботи з GPIO, I2C, SPI та іншими інтерфейсами. Крім того, для розробки додатків з графічним інтерфейсом доступні стандартні фреймворки, такі як Qt, GTK, Electron та інші.

Важливою особливістю програмного забезпечення Raspberry Pi є підтримка віртуалізації та контейнеризації, що дозволяє розгортати Docker-контейнери та віртуальні машини для ізоляції середовищ виконання різних додатків. Ця функціональність робить Raspberry Pi придатним для використання в якості локального сервера для тестування та розробки.

Raspberry Pi підтримує стандартні механізми оновлення програмного забезпечення, характерні для Linux-систем, через систему управління пакетами APT, що забезпечує централізоване отримання оновлень безпеки та нових версій програм з офіційних репозиторіїв.

2.5 Порівняльний аналіз хмарних платформ для візуалізації даних з мікроконтролерів

В контексті даної роботи необхідно візуалізувати дані з мікроконтролера за допомогою хмарної платформи. Проведемо порівняння найпопулярніших платформ на ринку.

Першою такою системою є BlynkIoT яка представляє собою комплексну платформу для розробки додатків у сфері Інтернету речей, яка поєднує апаратні та програмні компоненти для забезпечення взаємодії між фізичними пристроями та мобільними додатками. Архітектура BlynkIoT базується на тривірневій моделі, що включає: серверну частину (Blynk Server), клієнтську бібліотеку (Blynk Library) та

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

мобільний додаток (Blynk App). Така структура забезпечує гнучкість та ефективність у процесі розробки IoT-рішень.

Серверна частина BlynkIoT відповідає за встановлення з'єднання між мікроконтролерами та мобільними пристроями, здійснює керування сеансами, зберігає дані проектів та забезпечує аутентифікацію користувачів. Важливо відзначити, що BlynkIoT пропонує як хмарний сервер, що значно спрощує процес розробки, так і можливість розгортання локального сервера, що надає додаткові переваги з точки зору безпеки та контролю над даними.

Клієнтська бібліотека BlynkIoT призначена для інтеграції з мікроконтролерами та забезпечує надійний зв'язок з сервером. Вона підтримує широкий спектр апаратних платформ, включаючи Arduino, ESP8266, ESP32, Raspberry Pi та інші. Бібліотека характеризується компактністю та оптимізованим використанням ресурсів, що є особливо важливим для пристроїв з обмеженими обчислювальними можливостями.

Мобільний додаток BlynkIoT представляє собою графічний інтерфейс для взаємодії з підключеними пристроями. Він дозволяє створювати інтерактивні панелі керування за допомогою готових віджетів, таких як кнопки, повзунки, графіки, дисплеї та інші елементи. Важливою особливістю додатку є можливість створення інтерфейсу методом "drag-and-drop", що значно спрощує процес розробки та не вимагає глибоких знань у програмуванні мобільних додатків.

З точки зору функціональності, BlynkIoT пропонує широкий спектр можливостей для візуалізації та аналізу даних. Платформа підтримує відображення даних у реальному часі, створення історичних графіків, генерацію сповіщень на основі заданих умов та можливість експорту даних для подальшого аналізу. Окрім цього, BlynkIoT надає інструменти для створення автоматизованих сценаріїв, що дозволяють реагувати на певні події або зміни у показниках датчиків.

Однією з ключових переваг BlynkIoT є простота інтеграції з різними протоколами зв'язку, включаючи Wi-Fi, Bluetooth, GSM, Ethernet та інші. Це забезпечує гнучкість у виборі технологій зв'язку в залежності від конкретних вимог

					КвРКІ 210106.21.01.42.ПЗ	Арк. 35
Зм.	Арк.	№ докум.	Підпис	Дата		

проекту та наявної інфраструктури. Крім того, платформа підтримує інтеграцію з іншими сервісами через API, що розширює можливості для створення комплексних рішень.

З точки зору масштабованості, BlynkIoT пропонує різні тарифні плани, орієнтовані на проекти різного масштабу – від простих аматорських розробок до промислових систем з великою кількістю пристроїв. При цьому, важливо відзначити, що при переході на більш високі тарифні плани розширюються не лише квоти на кількість пристроїв та обсяг даних, а й функціональні можливості платформи.

Далі розглянемо ThingSpeak, що є відкритою платформою для аналізу даних IoT, розробленою компанією MathWorks, відомою своїм програмним забезпеченням MATLAB. Архітектура ThingSpeak базується на концепції "каналів" (channels), які представляють собою контейнери для збору, зберігання та візуалізації даних з підключених пристроїв. Кожен канал може містити до 8 полів даних, метадані (опис, розташування, тощо) та статуси (наприклад, інформацію про останнє оновлення).

Серверна інфраструктура ThingSpeak побудована на основі RESTful API, що забезпечує простоту інтеграції з різними пристроями та системами. Для відправлення даних на платформу використовуються HTTP-запити, що робить процес комунікації універсальним та не залежним від конкретної апаратної платформи. Крім того, ThingSpeak пропонує спеціалізовані бібліотеки для популярних мікроконтролерів, таких як Arduino, Particle та ESP8266/ESP32, що додатково спрощує процес інтеграції.

Однією з ключових особливостей ThingSpeak є тісна інтеграція з середовищем MATLAB, що надає потужні можливості для аналізу та обробки даних. Користувачі мають доступ до функцій MATLAB безпосередньо через веб-інтерфейс ThingSpeak, що дозволяє реалізувати складні алгоритми аналізу без необхідності встановлення додаткового програмного забезпечення. Це особливо

					КвРКІ 210106.21.01.42.ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

корисно для проектів, що вимагають статистичного аналізу, фільтрації даних, виявлення аномалій та прогнозування.

У контексті візуалізації даних, ThingSpeak пропонує різноманітні інструменти для створення графіків та діаграм. Платформа підтримує різні типи візуалізацій, включаючи лінійні графіки, гістограми, діаграми розсіювання та інші. Користувачі можуть налаштовувати параметри відображення, такі як часові діапазони, кольори, підписи осей та легенди. Крім того, ThingSpeak дозволяє вбудовувати створені візуалізації на зовнішні веб-сайти за допомогою iframe, що розширює можливості для публікації та розповсюдження даних.

Важливою складовою функціональності ThingSpeak є концепція "TimeControl", що дозволяє планувати виконання аналітичних завдань за розкладом. Це надає можливість автоматизувати процеси обробки даних, генерації звітів та сповіщень. Наприклад, можна налаштувати щоденний аналіз даних з датчиків температури для виявлення аномальних значень та відправлення повідомлень у разі їх виявлення.

З точки зору безпеки, ThingSpeak підтримує різні механізми аутентифікації та авторизації. Кожен канал має унікальні ключі API для запису та читання даних, що забезпечує контроль доступу до інформації. Крім того, платформа підтримує шифрування даних при передачі через HTTPS, що захищає інформацію від перехоплення. Однак, варто зазначити, що деякі розширені функції безпеки, такі як двофакторна аутентифікація, доступні лише у комерційній версії платформи.

Щодо масштабованості, ThingSpeak пропонує як безкоштовний план для невеликих проектів та експериментів, так і комерційні плани для більш вимогливих застосувань. Безкоштовна версія має певні обмеження, зокрема щодо кількості повідомлень (8,200 на рік), частоти оновлень (не частіше ніж раз на 15 секунд) та кількості каналів (не більше 4). Комерційні плани знімають ці обмеження та надають додаткові можливості, такі як збільшений обсяг зберігання даних, пріоритетна підтримка та доступ до розширених функцій безпеки.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

Ще однією популярною системою є Ubidots яка є хмарною платформою для Інтернету речей, орієнтованою на промислові та комерційні застосування. Архітектура Ubidots побудована навколо концепції "змінних" (variables), які організовані в "пристрої" (devices), що, в свою чергу, можуть бути згруповані у "колекції" (collections). Така ієрархічна структура забезпечує гнучкість у організації та управлінні даними, особливо у проектах з великою кількістю пристроїв та датчиків.

Серверна інфраструктура Ubidots забезпечує надійне зберігання та обробку даних з можливістю масштабування відповідно до потреб проекту. Платформа підтримує різні протоколи зв'язку, включаючи HTTP/HTTPS, MQTT, TCP/UDP та Sigfox, що робить її сумісною з широким спектром IoT-пристроїв та мереж. Крім того, Ubidots пропонує спеціалізовані бібліотеки для популярних мікроконтролерів, таких як Arduino, ESP8266/ESP32, Raspberry Pi та інших, що спрощує процес інтеграції та зменшує час на розробку.

Ubidots надає розширені можливості для аналізу даних та створення автоматизованих сценаріїв. Платформа підтримує складні математичні операції над даними, включаючи агрегацію, фільтрацію, нормалізацію та інші трансформації. Користувачі можуть створювати "синтетичні" змінні, які обчислюються на основі інших змінних за заданими формулами. Це дозволяє реалізувати комплексні аналітичні алгоритми без необхідності модифікації коду на стороні пристроїв.

Особливу увагу в Ubidots приділено механізмам тригерів та подій. Платформа дозволяє визначати умови, при яких генеруються сповіщення або виконуються певні дії. Наприклад, можна налаштувати відправлення електронного листа, SMS-повідомлення або веб-хука при перевищенні певного порогового значення або при виявленні аномального стану системи. Підтримка часових умов та комбінованих тригерів дозволяє реалізувати складні сценарії реагування на події.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

Важливою особливістю Ubidots є розширені можливості для інтеграції з іншими системами та сервісами. Платформа підтримує взаємодію з системами управління підприємством (ERP), сервісами аналітики (наприклад, Google Analytics), хмарними платформами (AWS, Azure, Google Cloud) та іншими сервісами через API та веб-хуки. Це дозволяє інтегрувати IoT-рішення у існуючу інфраструктуру підприємства та забезпечити безперервний потік даних між різними системами.

Щодо масштабованості, Ubidots пропонує диференційовані тарифні плани, орієнтовані на різні сегменти ринку. Платформа має безкоштовну версію STEM, призначену для освітніх та експериментальних цілей, а також комерційні плани для бізнес-застосувань. Комерційні плани відрізняються за квотами на кількість пристроїв, обсяг даних та кількість користувачів, а також за доступністю певних розширених функцій. Важливо відзначити, що всі тарифні плани Ubidots підтримують необмежену кількість дашбордів та віджетів, що надає гнучкість у створенні візуалізацій.

2.6 Вибір та аналіз вартості компонентів для погодної станції

Основу досліджуваної системи складає обчислювальний модуль ESP32 LuaNode32 Type-C з 38 контактними виводами, що представляє собою передовий мікроконтролер з інтегрованими функціями Wi-Fi та Bluetooth комунікації. Даний модуль характеризується високою продуктивністю завдяки двоядерному процесору з тактовою частотою до 240 МГц, що дозволяє виконувати складні обчислювальні операції в режимі реального часу. Доцільність використання даного модуля обумовлена оптимальним співвідношенням обчислювальної потужності, комунікаційних можливостей та енергоефективності, що дозволяє реалізувати широкий спектр функціональності при відносно помірній вартості, яка складає 375 гривень.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

Електроживлення системи забезпечується блоком живлення Sunny з вихідними параметрами 6В та максимальним струмом 2А, що відповідає потужності 12 Вт. Даний блок живлення характеризується високою стабільністю вихідної напруги при змінах навантаження, що є критично важливим для забезпечення стабільної роботи мікроконтролера та сенсорів. Конструктивно блок живлення виконаний у вигляді адаптера з вбудованим стабілізатором напруги та захистом від короткого замикання, перевантаження та перегріву, що підвищує надійність та безпеку експлуатації системи.




Вибір даного блоку живлення обумовлений необхідністю забезпечення стабільного електроживлення всіх компонентів системи з урахуванням пікових навантажень при одночасній роботі сенсорів та модуля бездротової передачі даних, особливо під час встановлення Wi-Fi з'єднання, коли споживання струму може досягати 250 мА. Вартість даного блоку живлення становить 179 гривень, що є обґрунтованим з урахуванням його технічних характеристик та надійності.

Додатковим елементом системи електроживлення є спеціалізований модуль живлення для макетної плати, що забезпечує стабілізовані напруги 5В та 3.3В, які є стандартними для більшості сучасних електронних компонентів.

Конструктивно модуль виконаний у вигляді плати з штирьовими контактами, що забезпечують легке підключення до макетної плати без необхідності пайки. Використання такого модуля дозволяє оптимізувати розподіл електроенергії та зменшити ризик пошкодження чутливих електронних компонентів через перенапругу або неправильне підключення. Вартість даного модуля становить 85 гривень, що є економічно обґрунтованим з огляду на його функціональність та роль у забезпеченні стабільної роботи всієї системи.

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Основні компоненти системи

Назва компоненту	Фото	Короткий опис
Блок живлення Sunny 6В 2А 2.1x5.5мм		Блок живлення, вхід 100 - 240 В змінного струму, вихід 6В 2А постійного струму.
Wi-Fi модуль ESP32 LuaNode32 Type-C (38-pin)		Модуль розробника LuaNode32 Type-C побудований на мікромодулі ESP-WROOM-32 .
Живлення для макетної плати 5В /3.3В		Модуль живлення для макетної плати. Є перемикач 3.3 / 5В. Живиться від зовнішнього блоку живлення 6,5 - 9В.

Сенсорний блок системи представлений двома основними компонентами: модулем датчика якості повітря MQ135 та датчиком вологості і температури DHT21 (AM2301). Модуль MQ135 призначений для виявлення та вимірювання концентрації шкідливих газів, включаючи аміак, оксиди азоту, бензол, дим, вуглекислий газ та інші летючі органічні сполуки. Технологічно даний сенсор відноситься до категорії напівпровідникових газових сенсорів, що працюють за

						КвРКІ 210106.21.01.42.ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата			

принципом зміни електропровідності чутливого шару при адсорбції молекул газів на його поверхні. Для забезпечення стабільних показань сенсор потребує попереднього нагрівання до робочої температури, що забезпечується вбудованим нагрівальним елементом з напругою живлення 5В.

Конструктивно модуль MQ135 складається з самого сенсора та плати з мікросхемою компаратора LM393, що дозволяє отримувати як аналоговий сигнал, пропорційний концентрації газу, так і дискретний сигнал при перевищенні встановленого порогу концентрації. Наявність на платі потенціометра дозволяє регулювати поріг спрацьовування дискретного виходу, що розширює функціональні можливості модуля та забезпечує гнучкість його використання в різних умовах експлуатації. Діапазон вимірювання концентрації газів знаходиться в межах від 10 до 1000 ppm, що відповідає більшості практичних застосувань у сфері контролю якості повітря в побутових та промислових приміщеннях.

Економічна доцільність використання даного сенсора визначається його високою чутливістю, стабільністю показань та конкурентоспроможною ціною у порівнянні з альтернативними рішеннями аналогічного призначення. Вартість модуля MQ135 становить 110 гривень, що є помірною з урахуванням його функціональності та технологічної складності. Додатковою перевагою даного модуля є широка розповсюдженість та наявність докладної документації, що спрощує його інтеграцію в різноманітні електронні системи та зменшує витрати на розробку програмного забезпечення.

Датчик DHT21 (AM2301) представляє собою комбінований сенсор, що забезпечує одночасне вимірювання температури та відносної вологості повітря з високою точністю. Даний сенсор використовує ємнісний принцип вимірювання вологості та термісторний принцип вимірювання температури, що забезпечує високу чутливість та стабільність показань. Діапазон вимірювання відносної вологості складає від 0% до 100% з точністю $\pm 3\%$, а діапазон вимірювання температури - від -40°C до $+80^{\circ}\text{C}$ з точністю $\pm 0.5^{\circ}\text{C}$, що задовольняє вимоги більшості практичних застосувань.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

Конструктивно датчик DHT21 виконаний у компактному пластиковому корпусі з трьома виводами для підключення: живлення (3.3-5.5В), земля та цифровий сигнал. Даний сенсор характеризується цифровим інтерфейсом передачі даних з використанням спеціалізованого протоколу з однопровідним підключенням, що спрощує його інтеграцію в систему та підвищує завадостійкість вимірювань.

Вартість датчика DHT21 (AM2301) становить 195 гривень, що є обґрунтованою з урахуванням його функціональності, діапазону вимірювань та точності показань, що відповідають вимогам більшості практичних застосувань. Порівняно з більш простим аналогом DHT11, даний сенсор забезпечує вищу точність вимірювань та ширший діапазон робочих температур, що обумовлює його вищу вартість, але забезпечує кращі експлуатаційні характеристики в різноманітних умовах використання.

Таблиця 2.2 – Сенсорні компоненти системи

Назва компоненту	Фото	Короткий опис
Модуль датчика якості повітря MQ-135		Модуль датчика якості повітря MQ135
Датчик вологості та температури dht21/am2301a		Датчик вологості та температури DHT21/AM2301A – датчик температури та вологості

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

КВРКІ 210106.21.01.42.ПЗ

Арк.
43

Для візуалізації даних, отриманих від сенсорів, у системі використовується символний дисплей LCD 2004 з інтерфейсом I2C, що має 4 рядки по 20 символів з синім підсвічуванням. Технологічно даний дисплей відноситься до категорії рідкокристалічних дисплеїв, що використовують ефект зміни оптичних властивостей рідких кристалів під впливом електричного поля. Символи на такому дисплеї формуються за допомогою матриці точок розміром 5×8 пікселів, що забезпечує хорошу розбірливість тексту при відносно низькій вартості компонента.

Конструктивно дисплей LCD 2004 представляє собою модуль, що складається з рідкокристалічної панелі, контролера HD44780 (або сумісного) та плати розширення з перетворювачем інтерфейсу I2C, який базується на мікросхемі PCF8574. Використання інтерфейсу I2C дозволяє зменшити кількість необхідних з'єднань з мікроконтролером до чотирьох проводів (живлення, земля, SDA та SCL), що спрощує монтаж та зменшує ймовірність помилок при підключенні. Крім того, технологія I2C забезпечує можливість підключення до однієї шини кількох пристроїв з різними адресами, що розширює масштабованість системи.

Дисплей LCD 2004 з інтерфейсом I2C характеризується відносно низьким енергоспоживанням (близько 20 мА без підсвічування та до 100 мА з увімкненим підсвічуванням) та широким діапазоном робочих температур (від -20°C до +70°C), що дозволяє використовувати його в різноманітних умовах експлуатації. Наявність регулювання контрастності та яскравості підсвічування забезпечує оптимальну видимість інформації при різних рівнях освітлення.

Економічна доцільність використання даного дисплея обумовлена оптимальним співвідношенням інформаційної ємності (загальна кількість символів, що можуть бути відображені одночасно - 80) та вартості, яка становить 275 гривень. Порівняно з більш простими дисплеями (наприклад, LCD 1602 з 2 рядками по 16 символів), даний модуль забезпечує більший обсяг відображуваної інформації при незначному збільшенні вартості, що є важливим для систем моніторингу, де необхідно одночасно відображати кілька параметрів

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

навколишнього середовища з відповідними поясненнями та одиницями вимірювання.

Синє підсвічування дисплея не тільки забезпечує краще візуальне сприйняття інформації в умовах недостатньої освітленості, але й надає пристрою більш сучасний та естетичний вигляд, що може бути важливим фактором при використанні системи в житлових приміщеннях або офісах. Альтернативні варіанти підсвічування (зелене, жовте, біле) також доступні на ринку за аналогічною ціною, що дозволяє вибрати оптимальний варіант відповідно до дизайну корпусу та умов експлуатації.

Використання контролера HD44780 (або сумісного) забезпечує широку програмну підтримку, включаючи наявність готових бібліотек для різних платформ розробки, що спрощує програмування та зменшує витрати на розробку програмного забезпечення. Крім відображення стандартних символів ASCII, дисплей підтримує можливість створення та відображення користувацьких символів, що розширює його функціональні можливості та дозволяє відображати спеціальні символи, графічні елементи або символи національних алфавітів.

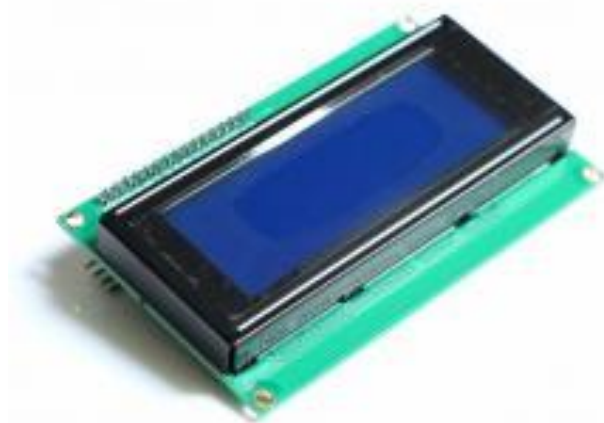


Рисунок 2.4 – Lcd дисплей

Конструктивна база системи представлена макетною платою МВ-102 з 830 отворами, що забезпечує можливість прототипування та тестування схемних рішень без необхідності пайки компонентів. Технологічно макетна плата МВ-102 виконана з високоякісного пластику з нанесеними на поверхню струмопровідними доріжками, що забезпечують електричне з'єднання між відповідними контактними отворами за заданою схемою. Конструкція плати передбачає наявність двох силових шин живлення та двох шин заземлення, що проходять по краях плати вздовж усієї її довжини, а також множини груп по п'ять з'єднаних між собою контактних отворів, розташованих перпендикулярно до силових шин.

Загальна кількість контактних отворів (830) забезпечує можливість реалізації досить складних схем з використанням великої кількості компонентів різного призначення. Стандартний крок між контактними отворами (2.54 мм) відповідає кроку виводів більшості електронних компонентів у корпусах DIP та модулів, що забезпечує їх легке та надійне встановлення. Робоча площа макетної плати складає приблизно 16.5×5.5 см, що є достатнім для розміщення всіх компонентів системи з можливістю їх оптимального розташування для забезпечення зручності з'єднань та зменшення довжини з'єднувальних провідників.

Використання даної макетної плати є економічно обґрунтованим на етапі розробки та відлагодження системи, оскільки дозволяє оперативно вносити зміни до схеми з мінімальними часовими та матеріальними витратами. Вартість макетної плати МВ-102 з 830 отворами становить 65 гривень, що є незначною частиною загального бюджету проекту, але забезпечує суттєве спрощення та прискорення процесу розробки. Довговічність даного компонента при правильній експлуатації складає кілька років, що дозволяє використовувати його для реалізації багатьох різних проектів з можливістю багаторазового перезбирання схем.

Для забезпечення електричних з'єднань між компонентами використовується комплект перемичок Arduino, що містить 120 провідників різної довжини та кольорів. У комплект входять перемички типу "штир-штир" (male-to-male), "гніздо-гніздо" (female-to-female) та "штир-гніздо" (male-to-female), що забезпечує


					КвРКІ 210106.21.01.42.ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

можливість створення з'єднань між компонентами різного типу. Кольорове маркування провідників (червоний, чорний, жовтий, зелений, синій та білий) дозволяє легко ідентифікувати різні типи з'єднань (живлення, земля, сигнальні лінії різного призначення), що спрощує налагодження схеми та зменшує ймовірність помилок при підключенні.

Якість виготовлення перемичок (використання багатожильного мідного провідника в гнучкій ізоляції, надійне кріплення контактів) забезпечує стабільність електричних з'єднань та довговічність при багаторазовому використанні.

Даний комплект представляє собою оптимальне рішення з точки зору співвідношення кількості та різноманітності перемичок до їх загальної вартості, яка складає 119 гривень, що дозволяє забезпечити всі необхідні з'єднання в системі з певним запасом на випадок пошкодження окремих провідників.

Таблиця 2.4 – Конструктивні елементи системи

Назва компоненту	Фото	Короткий опис
Макетна плата mb-102 830 отворів		Велика макетна плата (830 отворів) з двома лініями для живлення
Комплект перемичок мама-мама, тато-тато, мама-тато		Комплект перемичок для макетування довжиною 20 см.

Таким чином аналіз вибраних компонентів демонструє раціональність їх використання з точки зору функціональності, надійності та вартості. Обчислювальний модуль ESP32 LuaNode32 Type-C забезпечує високу продуктивність і широкі комунікаційні можливості

Таким чином, сформована система є оптимальною за співвідношенням ціна/можливості, що дозволяє реалізувати широкий спектр завдань моніторингу параметрів навколишнього середовища з мінімальними витратами та високою надійністю.

2.7 Висновок до розділу 2

На основі аналізу проведеного вище, було обрано мікроконтролерну платформу ESP32 у поєднанні з BlynkIoT як оптимальне рішення для реалізації системи моніторингу якості повітря.

ESP32 обраний як апаратна основа системи через повну відповідність технічним вимогам проекту. Платформа забезпечує необхідну функціональність для підключення датчиків температури, вологості та якості повітря завдяки наявності достатньої кількості GPIO портів з підтримкою різних інтерфейсів передачі даних (I2C, SPI, UART). Крім того, ESP32 надає можливості бездротової передачі даних через вбудовані модулі Wi-Fi та Bluetooth без використання додаткових модулів, що спрощує архітектуру системи та знижує її вартість.

Важливою перевагою ESP32 є його енергоефективність та можливість роботи в режимах низького енергоспоживання, що особливо актуально для автономних систем моніторингу. Двоядерна архітектура процесора дозволяє ефективно розподіляти навантаження між збором даних з датчиків та їх передачею в хмарний сервіс.

BlynkIoT вибрано як хмарне рішення для збору, обробки та візуалізації даних через його спеціалізацію на IoT проектах та простоту інтеграції. Платформа пропонує готову інфраструктуру для реалізації всіх програмних вимог системи.

					КВРКІ 210106.21.01.42.ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ "ПОГОДНА СТАНЦІЯ НА ESP32" ІЗ ВІДОБРАЖЕННЯМ ДАНИХ У BLYNK

3.1 Розгортання та налаштування хмарної платформи Blynk

Розглянемо алгоритм дії для налаштування хмарної платформи Blynk

1. Реєстрація та створення проекту

Розпочати роботу з платформою Blynk необхідно з реєстрації облікового запису на офіційному веб-сайті <https://blynk.io>.

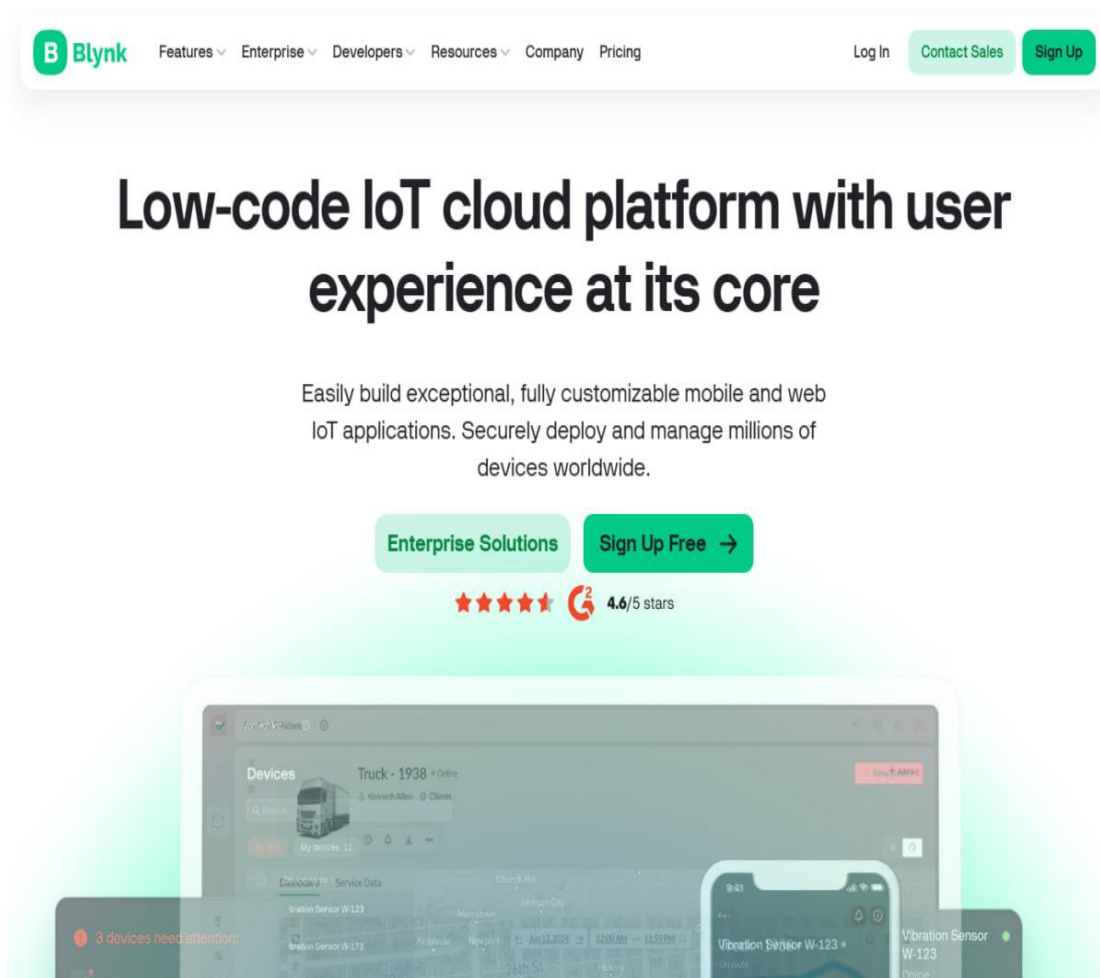


Рисунок 3.1 – Сайт Blynk.io

Процес реєстрації вимагає використання дійсної електронної адреси для підтвердження створеного облікового запису. Після успішної реєстрації та входу в

									КВРКІ 210106.21.01.42.ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата						

систему користувач отримує доступ до розділу "Developer Console", де можна створювати та керувати проектами.

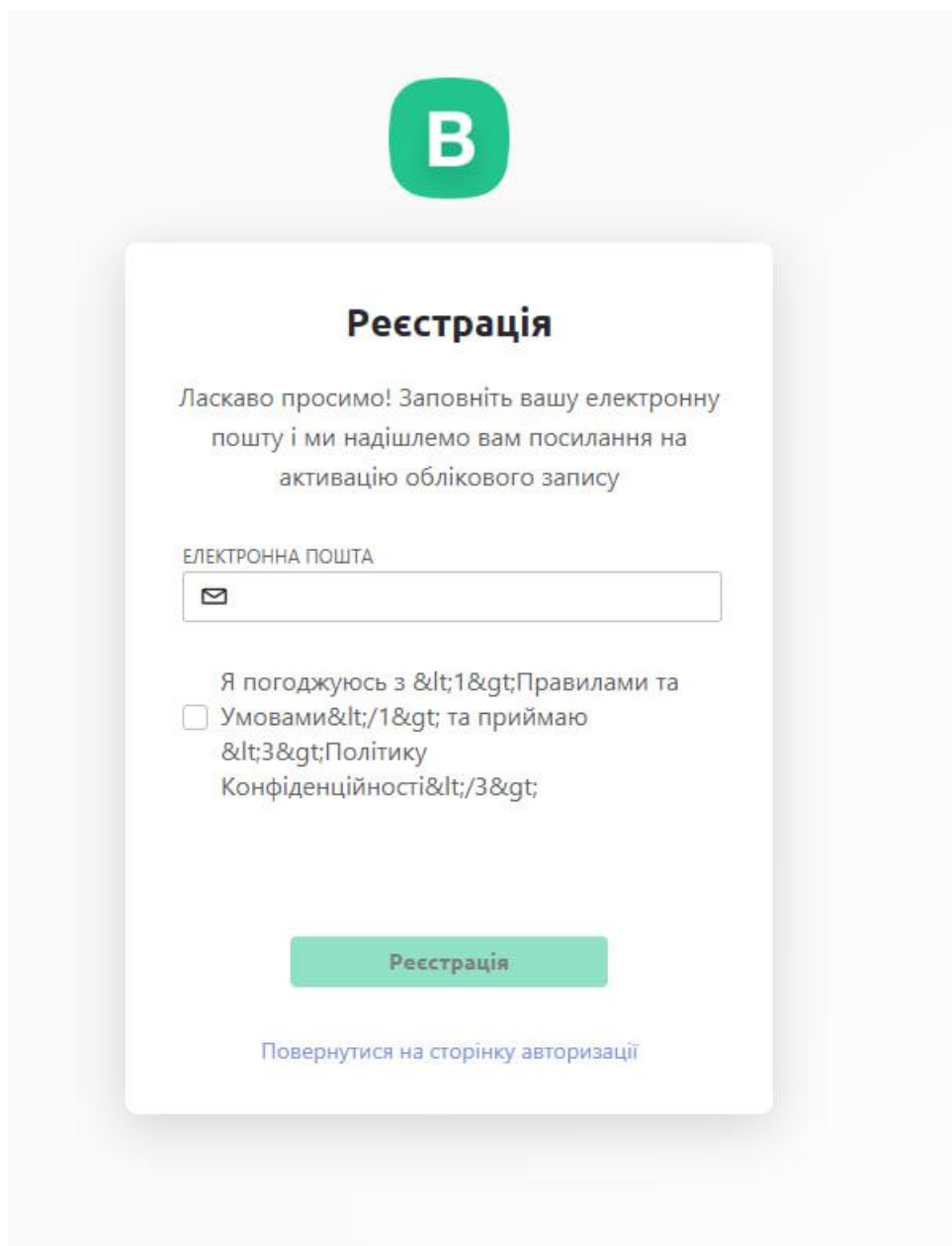


Рисунок 3.2 – Поле реєстрації на сайті Blynk

Для створення нового проекту метеорологічної станції необхідно натиснути кнопку "New Project" та вказати назву "ESP32 Weather Station". Важливим кроком є вибір типу пристрою "ESP32 Dev Board" із запропонованого переліку апаратних платформ та вибір типу з'єднання "WiFi".

					КвРКІ 210106.21.01.42.ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

Система автоматично генерує унікальний аутентифікаційний токен після створення проекту, який є критично важливим елементом безпеки для забезпечення безпечного з'єднання між метеостанцією на базі ESP32 та хмарним сервером. Зберігання токена повинно здійснюватися з дотриманням усіх необхідних заходів захисту для запобігання несанкціонованого доступу до системи.

2. Конфігурація інтерфейсу метеорологічної станції

Успішне створення проекту відкриває можливість для налаштування користувацького інтерфейсу, призначеного для відображення метеорологічних показників, що надходять від датчиків ESP32. З огляду на особливості вимірюваних параметрів, інтерфейс потребує створення спеціалізованих віджетів, таких як "Gauge" для візуалізації температури повітря з відповідним діапазоном значень, аналогічні віджети для відображення вологості та показників якості повітря, а також текстові віджети "Value Display" для представлення даних у текстовому форматі.

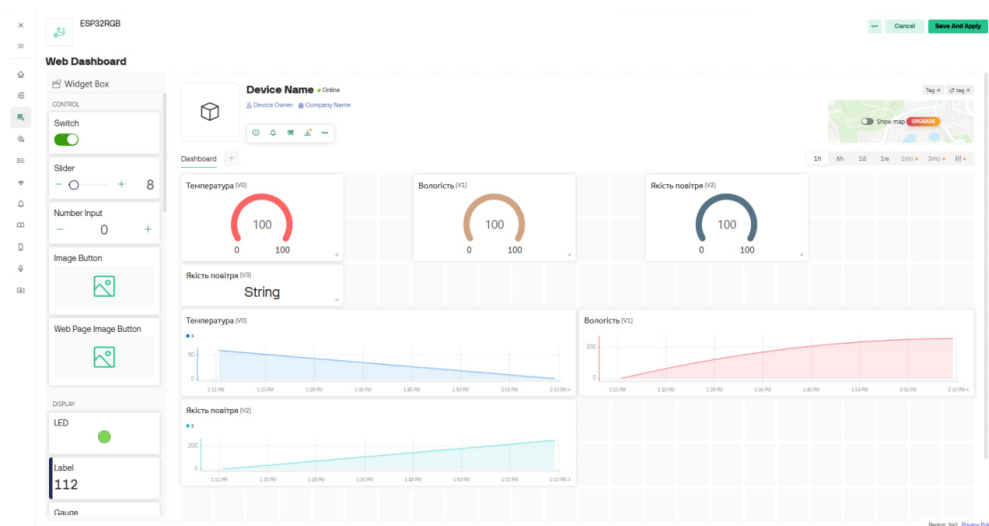


Рисунок 3.3 – Конструктор інтерфейсу Vynk IoT

Кожен віджет потребує налаштування властивостей, включаючи призначення відповідного віртуального піна, встановлення інтервалу оновлення даних у режимі PUSH, та налаштування кольорової індикації для різних діапазонів

Зм.	Арк.	№ докум.	Підпис	Дата

значень. Наприклад, для температури можна використовувати синій колір для низьких значень та червоний для високих, а для вологості - зелений колір для оптимального діапазону.

Важливим елементом інтерфейсу є графіки, що дозволяють відслідковувати динаміку показників у часі. Налаштування графіків вимагає вибору параметрів відображення, можливостей вибору часового інтервалу та встановлення коректного масштабування осей для оптимального представлення даних.

3. Налаштування серверної частини та збереження даних

Ефективна робота хмарної платформи Vlynk неможлива без належної організації процесів збору, обробки та зберігання даних, що надходять від метеорологічної станції. Ключовим елементом цього процесу є активація функції "Data Streams" для визначених віртуальних пінів, що забезпечує накопичення та збереження історичних показників у хмарному сховищі.

Важливим параметром є частота опитування датчиків та передачі даних на сервер, яка повинна забезпечувати актуальність інформації без надмірного навантаження на мережу. Рекомендований інтервал становить 15-30 секунд. Також необхідно врахувати обмеження обраного тарифного плану щодо періоду зберігання даних, оскільки безкоштовна версія дозволяє зберігати інформацію лише протягом одного місяця.

Для забезпечення своєчасного реагування на аномальні показники метеорологічної станції корисною є можливість налаштування системи сповіщень, яка генерує повідомлення при виході параметрів за встановлені межі. Система сповіщень потребує конфігурації каналів доставки, таких як електронна пошта або push-повідомлення, а також налаштування затримок між повторними сповіщеннями для запобігання їх надмірної кількості при коливанні показників навколо критичних значень.

Додатковою можливістю є налаштування механізмів експорту даних для їх подальшого аналізу або використання в інших системах. Це може включати конфігурацію API-інтеграції або налаштування періодичного експорту даних у

					КвРКІ 210106.21.01.42.ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

формати CSV чи JSON для проведення додаткової аналітики або створення резервних копій.

3.2 Створення android версії хмарної платформи

Для створення android версії платформи необхідно спочатку завантажити застосунок Blynk IoT у PlayMarket. Далі необхідно авторизуватись в додатку і перейти на екран створення.

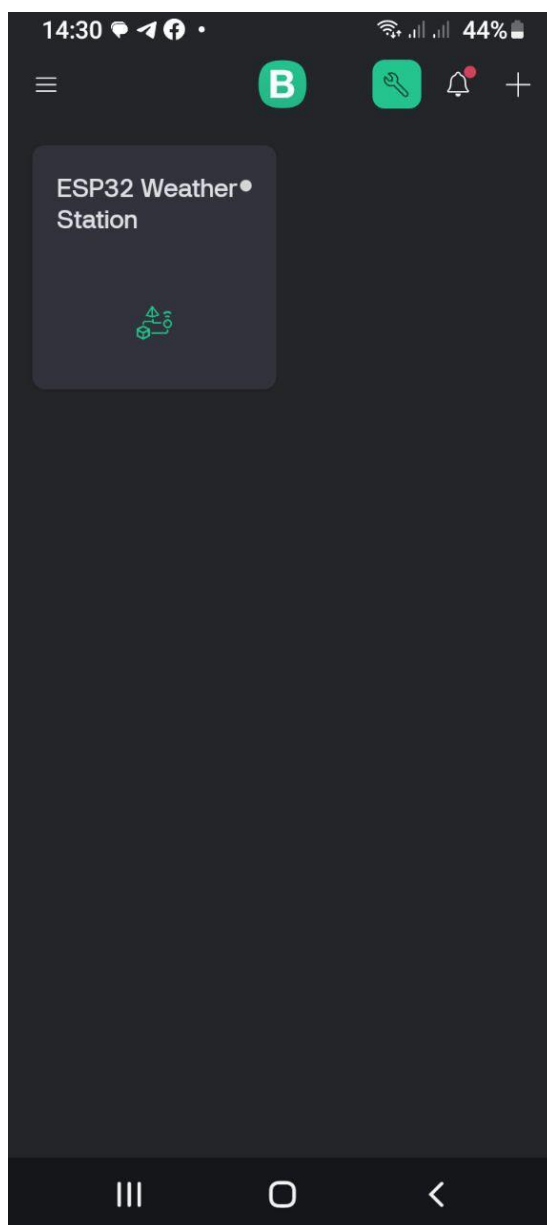


Рис 3.4 – Екран створення в мобільному застосунку Blynk

					КвРКІ 210106.21.01.42.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

Після цього необхідно додати бажані віджети на екран і налаштувати потоки даних.



Рисунок 3.5 – Налаштований екран мобільного застосунку

3.3 Програмна реалізація клієнтської частини програмно-апаратного засобу

Структура програмного коду погодної станції на базі ESP32 являє собою класичний приклад добре організованої програми для вбудованих систем. На початку коду здійснюється визначення необхідних директив препроцесора та

					КвРКІ 210106.21.01.42.ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

включення відповідних бібліотек. Директиви `#define BLYNK_TEMPLATE_ID` та `#define BLYNK_TEMPLATE_NAME` задають ідентифікатори для платформи Blynk, що є критичним для правильної конфігурації зв'язку з хмарним сервісом.

Подальше включення бібліотек `Wire.h`, `LiquidCrystal_I2C.h`, `DHT.h`, `WiFi.h` та `BlynkSimpleEsp32.h` забезпечує доступ до функціональності для роботи з I2C інтерфейсом, РК-дисплеєм, датчиком температури та вологості, WiFi підключенням та сервісом Blynk відповідно.

Наступний блок коду присвячений визначенню констант та пінів підключення. Константи `DHTPIN` і `DHTTYPE` визначають пін підключення та тип датчика температури та вологості. `BLYNK_AUTH_TOKEN` містить ключ автентифікації для платформи Blynk. `MQ135_DO` та `MQ135_AO` визначають цифровий та аналоговий виходи датчика якості повітря MQ135.

Після визначення констант ініціалізуються глобальні об'єкти датчика DHT та РК-дисплея. Створення об'єкта `dht` відбувається з використанням раніше визначених констант `DHTPIN` та `DHTTYPE`. Об'єкт `lcd` ініціалізується з адресою `0x27` та розмірами дисплея `20x4` символи, що відповідає стандартним параметрам для багатьох I2C РК-модулів.

Важливим елементом структури програми є масиви `ssidList` та `passList`, які містять дані для підключення до різних WiFi мереж. Такий підхід забезпечує гнучкість системи та дозволяє автоматично перебирати можливі варіанти підключення, що значно підвищує надійність роботи станції в різних умовах.

Пороги якості повітря визначаються константами `lowThreshold`, `mediumThreshold` та `highThreshold`, що використовуються пізніше для категоризації якості повітря.

Функціональна структура програми включає кілька чітко визначених допоміжних функцій. Функція `centerText()` реалізує алгоритм центрування тексту на дисплеї, розраховуючи необхідну кількість пробілів перед виведенням рядка.

Основна структура програми дотримується класичної моделі Arduino з двома ключовими функціями: `setup()` та `loop()`. Функція `setup()` виконує початкову

ініціалізацію всіх компонентів системи: налаштування серійного порту, ініціалізацію датчика DHT та РК-дисплея, підключення до WiFi мережі та, за умови успішного підключення, ініціалізацію з'єднання з сервісом Blynk.

Функція loop() забезпечує безперервний цикл роботи погодної станції. Вона починається з виклику функції Blynk.run() для підтримки зв'язку з сервісом Blynk. Далі здійснюється зчитування даних з датчиків: температури та вологості з DHT21, а також аналогового значення з MQ135.

Після оновлення даних з датчиків відбувається очищення дисплея та виведення інформації. Дані розміщуються на чотирьох рядках дисплея: назва станції, температура та вологість, рівень забруднення у відсотках та якісна оцінка стану повітря.

Завершальна частина функції loop() відповідає за відправку даних на платформу Blynk. Значення температури, вологості, якості повітря у відсотках та текстової оцінки якості повітря надсилаються на віртуальні піни V0, V1, V2 та V3 відповідно.

Структура коду включає елементи обробки помилок та забезпечення стійкості системи. Функція connectToWiFi() містить алгоритм послідовного перебору мереж з обмеженням кількості спроб підключення до кожної мережі. Це запобігає зависанню системи при неможливості підключення до жодної з мереж.

Перевірка статусу підключення до WiFi перед ініціалізацією з'єднання з Blynk є прикладом захисного програмування, що запобігає виникненню помилок при відсутності мережевого з'єднання. Система продовжує працювати в автономному режимі навіть при відсутності підключення до мережі, відображаючи дані на РК-дисплеї.

Структура виведення даних на дисплей з використанням функції centerText() забезпечує коректне відображення інформації незалежно від довжини рядків, що підвищує стійкість інтерфейсу користувача до змінних умов роботи.

Структура коду мережевої взаємодії включає два основні компоненти: підключення до WiFi та інтеграцію з хмарною платформою Blynk. Функція

connectToWiFi() реалізує алгоритм послідовного перебору мереж з масивів ssidList та passList, що забезпечує гнучкість та надійність підключення.

Інтеграція з платформою Blynk здійснюється через функцію Blynk.begin(), яка ініціалізує з'єднання з використанням токена автентифікації та параметрів WiFi підключення. Підтримка з'єднання забезпечується через регулярний виклик функції Blynk.run() на початку кожної ітерації циклу loop().

Відправка даних на платформу Blynk реалізується через виклики функції Blynk.virtualWrite() з відповідними віртуальними пінами та значеннями. Така структура дозволяє організувати ефективний двосторонній зв'язок між погодною станцією та хмарним сервісом.

3.4 Налаштування сповіщень на платформі Blynk

Система базується на мікроконтролері ESP32, інтегрованому з датчиками DHT21 (температура та вологість) та MQ135 (якість повітря). Дані передаються через віртуальні піни платформи Blynk до хмарного сервісу, де обробляються.

Система використовує наступну схему розподілу віртуальних пінів:

- 1) V0: Температура (°C)
- 2) V1: Відносна вологість (%)
- 3) V2: Індекс забрудненості повітря (%)
- 4) V3: Текстова оцінка якості повітря

Перш за все необхідно створити події (Events) у Blynk Console для кожного типу сповіщень:

ID	Name	Code	Color	Type	Notifications	Description	Exj	Actions
4	hight_pollution	hight_pollution	Grey	Info	Enabled	Увага ! Повітря надто з...	CC	Green
5	hight_temp	hight_temp	Red	Critical	Enabled	Увага температура при...	CC	Green
6	hight_humidity	hight_humidity	Red	Critical	Enabled	Увага ! Вологість прими...	CC	Green

Рисунок 3.6 – Створення Event в Blynk

Після створення подій необхідно налаштувати автоматизації у розділі "Automations" Blynk Console:

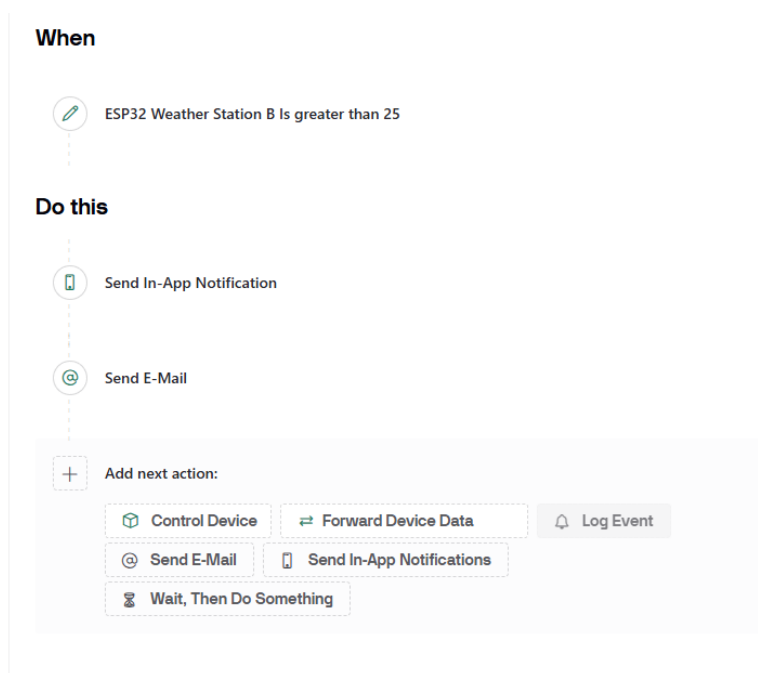


Рис 3.7 Приклад налаштування Automations в Blynk

Після цього при перевищенні певним показиком порогового значення, Blynk автоматично надішле лист на електронну пошту а також сповіщення.

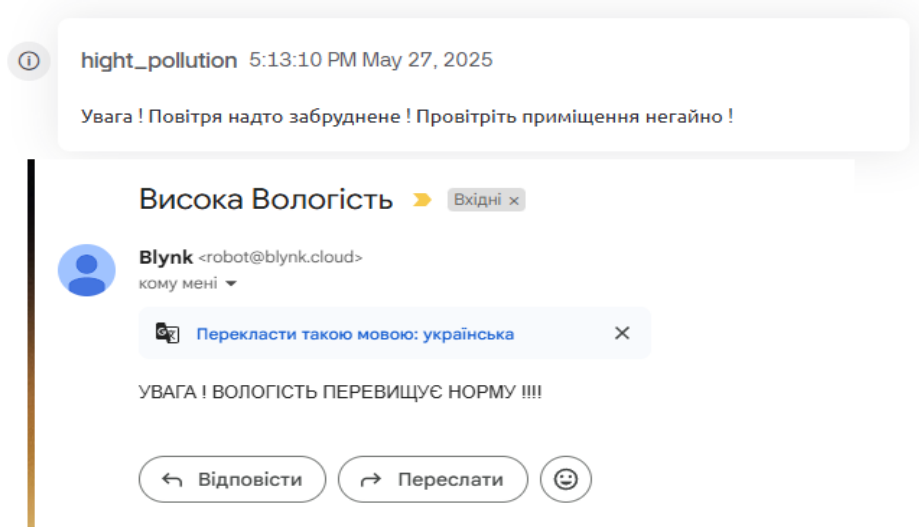


Рисунок 3.8 – Приклад сповіщень в Blynk

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було розроблено програмно-апаратний засіб "Погодна станція на ESP32" із відображенням даних у Blynk, який дозволяє здійснювати ефективний моніторинг метеорологічних параметрів у реальному часі.

Проведено всебічний аналіз предметної області, визначено актуальність застосування IoT-рішень у сфері метеоспостереження, обґрунтовано вибір апаратної платформи та хмарного сервісу для візуалізації даних, а також реалізовано прототип системи з урахуванням сучасних вимог до мобільності, енергоефективності та зручності користування.

У першому розділі проведено аналіз предметної області, етапів розвитку автоматизованих погодних станцій, їхньої архітектури та функціональних можливостей. Розглянуто сучасні апаратно-програмні рішення, їх класифікацію, особливості функціонування, порівняння популярних моделей погодних станцій, а також виявлено переваги та недоліки існуючих технологій. Особливу увагу приділено аналізу сенсорів, контролерів та платформ IoT.

У другому розділі проведено проектування системи обробки інформації в межах програмно-апаратного засобу, зокрема визначено вимоги до апаратного та програмного забезпечення, обґрунтовано вибір мікроконтролера ESP32 та хмарної платформи Blynk.

У третьому розділі реалізовано програмно-апаратний засіб, описано принцип роботи апаратної частини, алгоритми взаємодії з датчиками, структуру передачі та збереження даних. Створено інтерфейс у середовищі Blynk, що дозволяє в режимі реального часу відображати показники температури, вологості, атмосферного тиску та якості повітря. Оцінено ефективність роботи розробленої системи в умовах реального застосування, виявлено переваги розробленого рішення щодо простоти використання, розширюваності та доступності для кінцевого користувача.

					КвРКІ 210106.21.01.42.ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1.ESP32 Resources. Espressif Systems. URL: <https://www.espressif.com/en/products/socs/esp32/resources> (дата звернення 09.06.2025)
- 2.Arduino Uno – Getting Started. Arduino. URL: <https://www.arduino.cc/en/Guide/ArduinoUno> (дата звернення 09.06.2025)
- 3.Blynk Documentation. Blynk IoT Platform. URL: <https://docs.blynk.io/en/> (дата звернення 09.06.2025)
- 4.Blynk Library on GitHub. URL: <https://github.com/blynkkk/blynk-library> (дата звернення 09.06.2025)
- 5.ESP32 with Blynk 2.0 – Complete Guide. Random Nerd Tutorials. URL: <https://randomnerdtutorials.com/esp32-blynk-2-0/>. (дата звернення 09.06.2025)
- 6.Arduino Language Reference. Arduino. URL: <https://www.arduino.cc/reference/en/> (дата звернення 09.06.2025)
- 7.PlatformIO for ESP32. PlatformIO. URL: <https://platformio.org/platforms/espressif32> (дата звернення 09.06.2025)
- 8.BME280 – Humidity Sensor. Bosch Sensortec. URL: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/> (дата звернення 09.06.2025)
- 9.BME280 Combined Humidity and Pressure Sensor – Datasheet. Adafruit. URL: <https://cdn-shop.adafruit.com/datasheets/BME280.pdf> (дата звернення 09.06.2025)
10. DHT22 Temperature-Humidity Sensor – Datasheet. SparkFun. URL: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf> (дата звернення 09.06.2025)
11. DHT22 Temperature and Humidity Sensor. Adafruit. URL: <https://www.adafruit.com/product/385> (дата звернення 09.06.2025)
12. Modeling and Implementation of a Low-Cost IoT-Smart Weather Station Measuring Air Quality. PLoS ONE. 2022.

					КВРКІ 210106.21.01.42.ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

13. Pandey P., Kumar S. IoT Based Weather Monitoring System Using ESP32. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2022. Vol.8(2). P. 101–106.

14. Kumar A., Sharma N. IoT Based Weather Station Using ESP32 and Blynk. *International Journal of Computer Applications*. 2022. Vol.184(21). P. 15–20.

15. Singh N., Kaur J. IoT Based Weather Monitoring Using ESP32 and Blynk. *International Journal of Engineering and Technology*. 2023. Vol.15(2). P. 120–125.

16. Mishra R., Gupta A. IoT Based Weather Station Using ESP32 and Blynk. *International Journal of Advanced Research in Computer Science*. 2023. Vol.14(3). P. 110–115.

17. Kumar S., Singh P. IoT Based Weather Monitoring System Using ESP32. *International Journal of Computer Science and Mobile Computing*. 2022. Vol.11(4). P. 40–46.

18. Sutar S. S., Patil S. S. IoT Based Weather Monitoring System Using ESP32. *International Journal of Research in Engineering, Science and Management*. 2020. Vol.3(5). P. 1–4.

19. Suryawanshi S., et al. IoT Based Weather Monitoring System Using ESP32. *International Journal of Innovative Research in Computer and Communication Engineering*. 2021. Vol.9(4). P. 1–4.

20. Bhattacharjee S., et al. IoT based weather monitoring system using ESP32. *International Journal of Computer Applications*. 2020. Vol.176(25). P. 1–5.

21. Kaur P., et al. Design and implementation of IoT based weather monitoring system. *Materials Today: Proceedings*. 2021. Vol.46. P. 10834–10838.

22. Prathibha S. R., et al. IOT based weather monitoring system. *International Journal of Engineering Science and Computing*. 2017. Vol.7(3). P. 5486–5489.

23. Chavan P. S., et al. IoT Based Weather Monitoring System. *International Journal of Advanced Research in Computer and Communication Engineering*. 2018. Vol.7(6). P. 1–5.

					КВРКІ 210106.21.01.42.ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

24. Gurav P., et al. IoT Based Weather Monitoring System Using NodeMCU. *International Journal of Engineering Research & Technology*. 2018. Vol.7(6). P. 1–4.

25. Choudhary S., et al. IoT based weather station using ESP32 and cloud. *International Journal of Scientific & Technology Research*. 2020. Vol.9(2). P. 1234–1237.

26. Sahu S., et al. IoT based weather monitoring system using ESP32 and cloud. *International Journal of Engineering and Advanced Technology*. 2020. Vol.9(3). P. 1456–1460.

27. Patel D., et al. IoT based weather monitoring and reporting system using ESP32. *International Journal of Research and Analytical Reviews*. 2019. Vol.6(2). P. 112–116.

28. Singh S., et al. IoT based weather monitoring system using ESP32 and ThingSpeak. *International Journal of Computer Applications*. 2019. Vol.178(7). P. 9–13.

29. Kumar S., et al. IoT based weather monitoring system using ESP32 and Blynk. *International Journal of Research in Engineering and Technology*. 2021. Vol.8(5). P. 1–6.

30. Rathi S., Mehta A. IoT weather monitoring using ESP32 and cloud platforms. *International Journal of Engineering Research & Technology*. 2023. Vol.12(8). P. 150–155.

31. Sharma M., Gupta P. IoT weather station using ESP32 and mobile applications. *International Journal of Engineering and Technology*. 2023. Vol.15(4). P. 140–145.

32. Patel R., Singh S. ESP32 based weather station with IoT integration. *International Journal of Computer Science and Mobile Computing*. 2022. Vol.11(7). P. 75–80.

33. Kumar D., Singh V. IoT based weather station using ESP32 and Android interface. *International Journal of Engineering and Advanced Technology*. 2023. Vol.12(7). P. 120–125.

					КВРКІ 210106.21.01.42.ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

34. Banerjee P., Saha S. IoT based weather station using ESP32 and cloud services. *International Journal of Computer Applications*. 2022. Vol.184(40). P. 37–42.

35. Sharma A., Kumar N. ESP32 based weather monitoring system with IoT. *International Journal of Advanced Research in Computer Science*. 2023. Vol.14(5). P. 140–145.

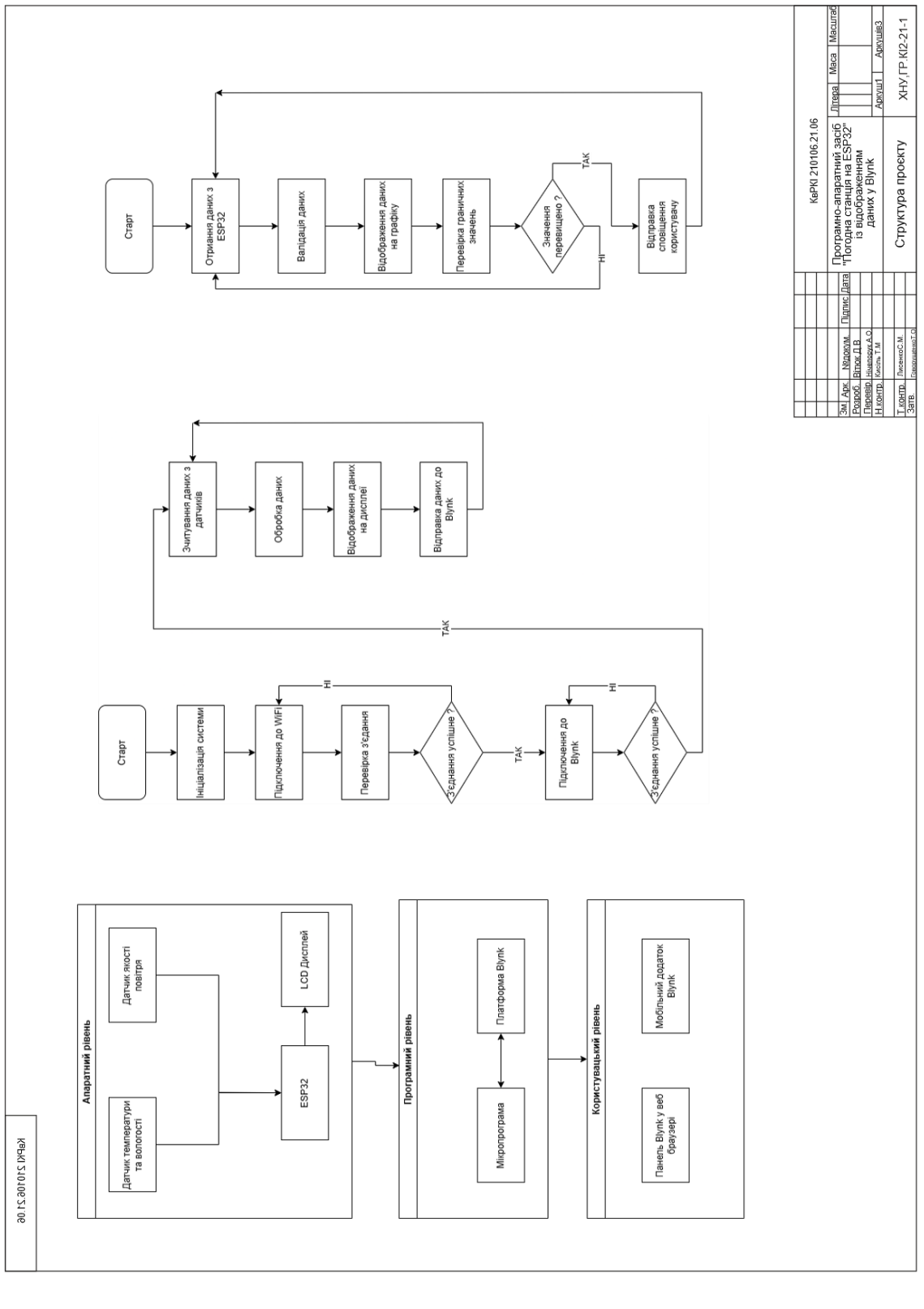
36. Singh S., Patel R. IoT weather station using ESP32 and mobile platforms. *International Journal of Engineering Research & Technology*. 2023. Vol.12(9). P. 160–165.

37. Kumar R., Gupta S. ESP32 based IoT weather station and data visualization. *International Journal of Computer Science and Mobile Computing*. 2022. Vol.11(8). P. 85–90.

					КВРКІ 210106.21.01.42.ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

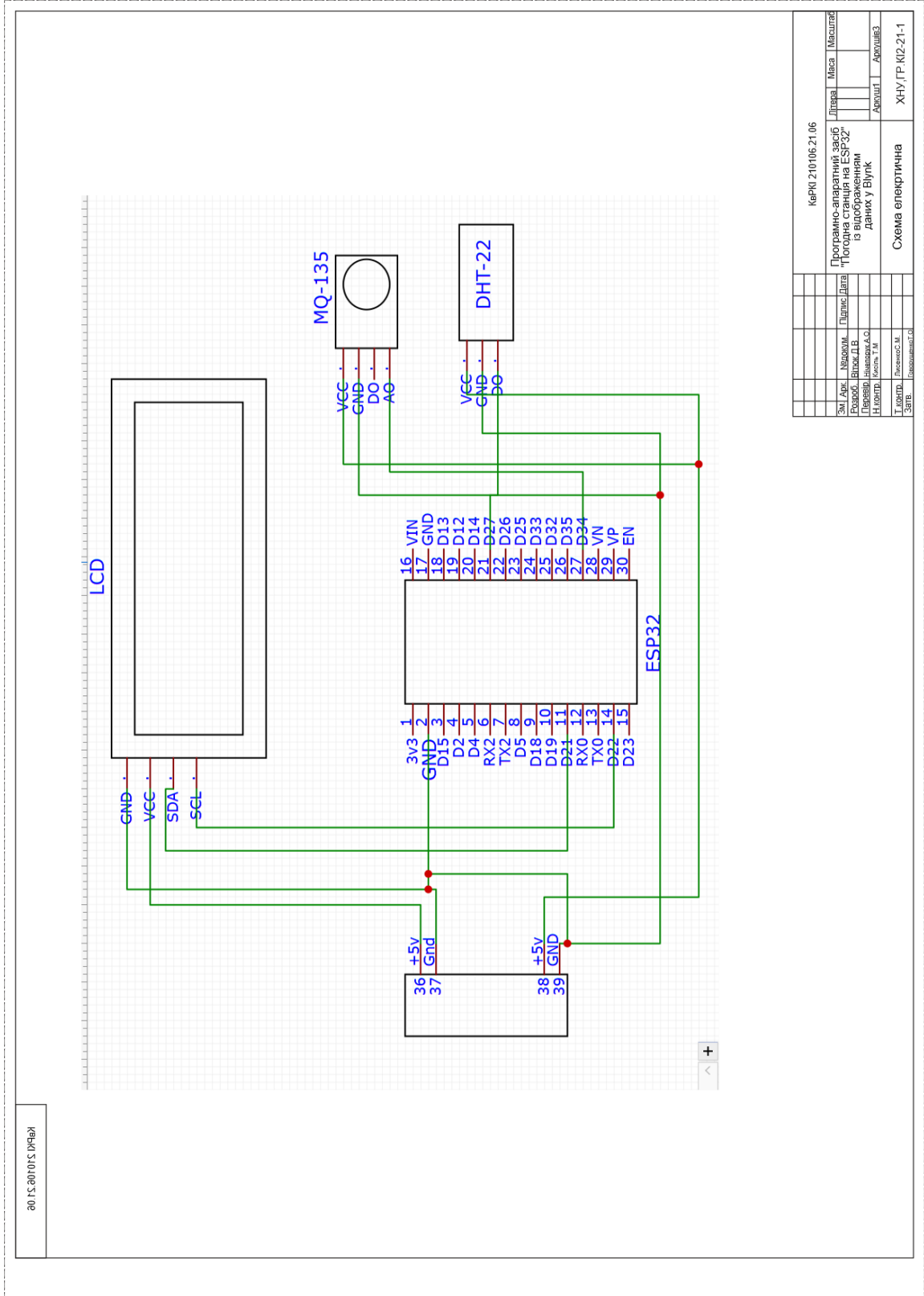
Додаток А (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «СТРУКТУРА ПРОЄКТУ»



Додаток Б (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «СХЕМА ЕЛЕКТРИЧНА»



автор: [unreadable]

керівник		Масштаб	
Зав. АРМ	Молодий	Листопад	Місяць
Керівник	Володимир	Початок	Діагональ
Н. Керівник	Керівник	Август	Август
Т. Керівник	Керівник	Август	Август
Зав. АРМ	Керівник	Август	Август
Програмно-апаратний засіб "Погодна станція на ESP32" із відображенням даних у Вульк		ХНУ/ГР КІ2-21-1	
Схема електрична			

Додаток Г
(обов'язковий)
КОД ПРОШИВКИ ESP32

```
#define BLYNK_TEMPLATE_ID "TMPL4yQXiASXR"
#define BLYNK_TEMPLATE_NAME "ESP32RGB"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#define DHTPIN 27
#define DHTTYPE DHT21
#define BLYNK_AUTH_TOKEN "wCjznQRoG97rcPB31_F2gktKMq10QNHf"
#define MQ135_DO 32
#define MQ135_AO 34
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 20, 4);
char auth[] = BLYNK_AUTH_TOKEN;
const char* ssidList[] = {"netis", "Galaxy"};
const char* passList[] = {"200419801952", "12345678"};
int networksCount = sizeof(ssidList) / sizeof(ssidList[0]);
int lowThreshold = 20;
int mediumThreshold = 30;
int highThreshold = 40;
unsigned long lastAlertTime = 0;
unsigned long alertInterval = 5 * 1000;
bool lastHighPollution = false;
bool lastHighTemp = false;
bool lastHighHumidity = false;
void centerText(int row, String text) {
    int spaces = (20 - text.length()) / 2;
    lcd.setCursor(spaces, row);
    lcd.print(text);
}
void loadingScreen() {
    lcd.clear();
    centerText(0, "Weather Station");
    centerText(2, "Loading...");
    for (int i = 0; i < 20; ++i) {
        lcd.setCursor(i, 3);
        lcd.print((char)255);
        delay(50);
    }
}
void connectToWiFi() {
    loadingScreen();
}
```

```

for (int i = 0; i < networksCount; ++i) {
  Serial.print("Connecting to ");
  Serial.println(ssidList[i]);
  WiFi.begin(ssidList[i], passList[i]);
  int attempts = 0;
  while (WiFi.status() != WL_CONNECTED && attempts < 20) {
    delay(500);
    Serial.print(".");
    attempts++;
  }
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nConnected to " + String(ssidList[i]));
    Serial.println("IP Address: " + WiFi.localIP().toString());
    lcd.clear();
    centerText(1, "WiFi Connected!");
    delay(1000);
    return;
  }
  Serial.println("\nFailed to connect to " +
String(ssidList[i]));
}
Serial.println("Could not connect to any WiFi network.");
}
void setup() {
  Serial.begin(115200);
  dht.begin();
  lcd.init();
  lcd.backlight();
  lcd.clear();
  connectToWiFi();
  if (WiFi.status() == WL_CONNECTED) {
    Blynk.begin(auth, WiFi.SSID().c_str(), WiFi.psk().c_str());
    Serial.println("Connected to Blynk");
    lcd.clear();
    centerText(1, "Connected to Blynk");
    delay(1000);
  }
}

void sendNotifications(float temp, float hum, float air_quality)
{
  unsigned long currentTime = millis();
  if (currentTime - lastAlertTime < alertInterval) return;
  bool shouldSendAlert = false;
  if (air_quality > 45 && !lastHighPollution) {
    Blynk.logEvent("high_pollution", "High pollution detected!");
    Serial.println("Alert: High pollution detected!");
    shouldSendAlert = true;
    lastHighPollution = true;
  } else if (air_quality <= 45) {
    lastHighPollution = false;
  }
}

```

```

if (temp > 30 && !lastHighTemp) {
  Blynk.logEvent("high_temp", "High temperature detected!");
  Serial.println("Alert: High temperature detected!");
  shouldSendAlert = true;
  lastHighTemp = true;
} else if (temp <= 30) {
  lastHighTemp = false;
}
if (hum > 70 && !lastHighHumidity) {
  Blynk.logEvent("high_humidity", "High humidity detected!");
  Serial.println("Alert: High humidity detected!");
  shouldSendAlert = true;
  lastHighHumidity = true;
} else if (hum <= 70) {
  lastHighHumidity = false;
}
if (shouldSendAlert) {
  lastAlertTime = currentTime;
}
}
void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    Blynk.run();
  } else {
    Serial.println("WiFi disconnected, attempting to
reconnect...");
    connectToWiFi();
    return;
  }
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();

  if (isnan(temp) || isnan(hum)) {
    Serial.println("Error: Failed to read from DHT sensor!");
    lcd.clear();
    centerText(1, "Sensor Error!");
    centerText(2, "DHT not responding");
    delay(2000);
    return;
  }
  int mq135_analog = analogRead(MQ135_AO);
  float air_quality = map(mq135_analog, 0, 1023, 0, 100);
  air_quality = fmin(100.0, air_quality);
  lcd.clear();
  centerText(0, "Weather Station");
  String line2 = "T: " + String(temp, 0) + (char)223 + "C H: " +
String(hum, 0) + "%";
  centerText(1, line2);
  String line3 = "Pollution: " + String(air_quality, 0) + "%";
  centerText(2, line3);
  String quality;
  if (air_quality < lowThreshold) quality = "Air: Good";

```

```

        else if (air_quality < mediumThreshold) quality = "Air:
Moderate";
    else if (air_quality < highThreshold) quality = "Air: Polluted";
    else quality = "Air: Hazardous";
        centerText(3, quality);
    if (Blynk.connected()) {
        Blynk.virtualWrite(V0, temp);
        Blynk.virtualWrite(V1, hum);
        Blynk.virtualWrite(V2, air_quality);
        Blynk.virtualWrite(V3, quality);
        sendNotifications(temp, hum, air_quality);
    } else {
        Serial.println("Blynk not connected, skipping data upload");
    }
    int rssi = WiFi.RSSI();
    Serial.println("WiFi Signal Strength (RSSI): " + String(rssi)
+ " dBm");
    Serial.printf("Temp: %.1f°C, Humidity: %.1f%%, Air Quality:
%.1f%%\n",
                    temp, hum, air_quality);
    delay(2000);
}

```

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Денис ВІТЮК

Співавтор:

Назва: Вітюк_Програмно-апаратний засіб "Погодозна станція на ESP32" із відображенням даних Blynk

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:1.7%

Коефіцієнт подібності 2:0.5%

Мікроробіли: 37

Заміна букв: 2

Інтервали: 0

Білі знаки: 4

Дата створення звіту: 2025-06-09 20:01:45.0

Після аналізу Звіту подібності констатуємо наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові створення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-10

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 12%**

ID: 244292 Title: БКР Програмно-апаратний засіб "Погодна станція на ESP32" із відображенням даних у Blynk Added in a DB: 2025-06-09 Authors: Денис ВІТЮК Heads: Андрій НІЧЕПОРУК Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	85498	555	1781 (2%)	17 (3%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Вітюк Денис Віталійович

Тема: Синтез та моделювання погодної станції на esp32

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 75

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є синтез та моделювання погодної станції на основі esp32.
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано теорію погодних станцій, а також розроблення і опис схем електричних функціональної і принципової) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено моделювання та проєктування пристрою, а саме: виконано формалізований опис погодної станції; розроблено змістовну схему алгоритму; закодовано верхівки змістовної схеми алгоритму; розроблено закодовану мікроопераційну схему алгоритму; закодовано мікрокоманди; розроблено закодовану мікрокомандну схема алгоритму; В третьому розділі кваліфікаційної роботи виконано апаратну реалізацію погодної станції.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага моделюванню схеми пристрою в середовищі EasyEDA.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Петровський Сергій Володимирович, доцент кафедри
Комп'ютерних наук, к.т.н., доцент

"13" ДБ 2025 р.

ЛВ (підпис)

Завідувачу кафедри КНС
д-р. філософії, доц. Ользі ПАВЛОВИЙ

Дениса ВІПЮКА

ІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-1

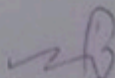
ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академіч добросчесності у Хмельницькому національному університеті» від 01.07.2022, згідно яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи захисту та застосування заходів дисциплінарної та академічної відповідальності ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарі університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

8 06 2025 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМПІСІЇ КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-апаратний засіб "Погодна станція на ESP32" із відображенням даних у Blynk

Автор Денис ВІТЮК

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський) рівень

Спеціальність 123- Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент Андрій НІЧЕПОРУК

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	Відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить наміски текстові спотворення, передбачувані спроби укріптя текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	Не виявлено

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
 - усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 1.67% і адресується до 9 першоджерел; та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____

Андрій НІЧЕПОРУК

Гарант ОП _____

Андрій НІЧЕПОРУК

Завідувач кафедри КІС _____

Ольга ПАВЛОВА