

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділенням

комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

підпис

« ____ » _____ 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

освітньо-професійного ступеня «фаховий молодший бакалавр»

зі спеціальності 123 «Комп'ютерна інженерія»

на тему:

«Система моніторингу метеорологічних параметрів навколишнього
середовища»

Студент групи КІ-41

Богдан КАПЛУН

(підпис)

Керівник роботи

Василь КУЗИК

(підпис)

Консультанти:

з техніко-економічного

обґрунтування

Любов МЕЛЕНЧУК

(підпис)

Нормоконтролер

Ольга СЛЄПЦОВА

(підпис)

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділенням

комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

підпис

« ____ » _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-кваліфікаційного рівня «фаховий молодший
бакалавр»

студенту Каплуну Богдану Володимировичу

(прізвище, ім'я та по-батькові студента)

1. Тема роботи «Система моніторингу метеорологічних параметрів навколишнього середовища» затверджено наказом по коледжу від “27” листопада 2023 р., №234а-н
2. Термін здачі студентом завершеного роботи “28” червня 2024 р.
3. Вихідні дані до роботи *актуальні технології та методи моніторингу метеорологічних параметрів.*
4. Перелік питань, які повинні бути розроблені в роботі:
 - а) основна частина дослідження предметної області, формалізація вимог до системи, реалізація та тестування системи
 - б) техніко-економічне обґрунтування аналіз ринку збуту продукту, дослідження системи та кошторису його реалізації; підрахунок загальної вартості розробки системи.

5. Перелік графічного матеріалу *структурна схема, функціональна схема, блок схема.*

6. Консультанти роботи:

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
3 техніко- економічного обґрунтування	Меленчук Л.І.		
	(вчена ступінь, звання		
	П.І.П. консультанта)		

КАЛЕНДАРНИЙ ПЛАН
Виконання кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1	<i>Вибір теми, ознайомлення з вимогами до кваліфікаційної роботи.</i>	24.11.2023	29.11.2023
2	<i>Дослідження предметної області, огляд типових рішень.</i>	30.11.2023	07.12.2023
3	<i>Дослідження технологій реалізації.</i>	08.12.2023	15.12.2023
4	<i>Розробка функціональних вимог до системи та робота над її структурою.</i>	16.12.2023	23.12.2023
5	<i>Проектування системи та підготовка відповідного графічного матеріалу.</i>	24.12.2023	03.02.2024
6	<i>Вибір, встановлення та налаштування середовища реалізації.</i>	04.02.2024	11.02.2024
7	<i>Реалізація системи та написання відповідного розділу кваліфікаційної роботи.</i>	12.02.2024	19.02.2024
8	<i>Реалізація системи та написання відповідного розділу кваліфікаційної роботи.</i>	20.03.2024	27.03.2024
9	<i>Опрацювання економічного розділу та написання відповідного розділу кваліфікаційної роботи.</i>	28.03.2024	04.04.2024
10	<i>Тестування системи та усунення недоліків.</i>	05.04.2024	12.04.2024
11	<i>Робота над оформленням пояснювальної записки.</i>	13.04.2024	20.05.2024
12	<i>Попередній захист кваліфікаційної роботи, доопрацювання.</i>	21.05.2024	30.05.2024
13	<i>Підготовка до захисту кваліфікаційної роботи.</i>	01.06.2024	08.06.2024

7. Дата видачі “___” _____ 202_ р. Керівник Кузик В.М./

Завдання прийняв до виконання _____

Реферат

Кваліфікаційна робота. Система моніторингу метеорологічних параметрів навколишнього середовища. Каплун Богдан. Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних технологій. Спеціальність 123 «Комп'ютерна інженерія». ГК, 2024. Сторінок 62, рисунків 24, додатків 2 .

Об'єкт розробки – система моніторингу метеорологічних параметрів навколишнього середовища

Метою роботи є створення автоматизованої системи моніторингу метеорологічних параметрів навколишнього середовища. Система повинна збирати метеорологічні дані в реальному часі та виводити їх користувачу.

Для реалізації проєкту було використано мікроконтролер ESP8266, середовище розробки Arduino IDE для програмування мікроконтролера ESP, бібліотеки для роботи з периферійними пристроями та онлайн сервіс ThingSpeak.

Найважливішими аспектами роботи системи є зручний моніторинг даних в реальному часі в мобільному застосунку або через онлайн сервіс.

У роботі розглянуто процес розробки та реалізації системи моніторингу метеорологічних параметрів навколишнього середовища, представлено ключові характеристики та переваги, а також надано рекомендації щодо її практичного застосування.

Abstract

Graduation Project. Monitoring system of meteorological parameters of the environment. Kaplun Bohdan. Vyacheslav Chornovol Halych Vocational College, Department of Computer Technologies. Specialty 123 "Computer Engineering". GK, 2024. Pages 62 , drawings 24 , appendices 2 .

The object of development is a system for monitoring meteorological parameters of the environment

The method of work is the creation of an automated system for monitoring meteorological parameters of the environment. The system must collect meteorological data in real time and display it to the user.

The project uses the ESP8266 microcontroller, the Arduino IDE development environment for programming the ESP microcontroller, libraries for working with peripheral devices, and the ThingSpeak online service.

The most important aspects of the system's operation are real-time data monitoring in a mobile application or through an online service.

The work deals with the process of development and implementation of the system for monitoring meteorological parameters of the environment, presents key characteristics and advantages, and also provides recommendations for its practical application.

ЗМІСТ

Скорочення і умовні позначки	6
Вступ	7
1 Аналіз існуючих рішень та постановка завдання	8
1.1 Аналіз предметної області	8
1.2 Аналіз існуючих рішень	9
2 Проєктування системи	16
2.1 Формалізація вимог до системи	16
2.2 Алгоритм роботи системи	23
3 Реалізація системи	26
3.1 Вибір засобів реалізації	26
3.2 Реалізація електричної схеми системи	34
3.3 Реалізація програмного забезпечення	36
3.4 Тестування роботи системи	43
4 Техніко економічне обґрунтування	49
4.1 Аналіз ринку	49
4.2 Розрахунок витрат на проєктування	50
4.3 Обґрунтування необхідності розробки	52
Висновки	54
Перелік джерел посилання	55
Додатки	56

					КР.КІ 24.530.03.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Кашун Б.В.			Система моніторингу метеорологічних параметрів навколишнього середовища	Літ.	Арк.
Перевір.		Кузик В.М.					5
Реценз.		Посвятовська О.Б.				ГФК.ВКТ.КІ-41	
Н. Контр.		Слепцова О.Я.					
Затверд.		Стефурак Н.А.					
						Акрушів	62

СКОРОЧЕННЯ І УМОВНІ ПОЗНАКИ

МК – Мікроконтролер

МП – Метеорологічні параметри

ПММП – Пристрої для моніторингу метеорологічних параметрів

IoT – Internet of Things

					КР.КІ 24.530.03.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасний світ стикається з різноманітними екологічними викликами, включаючи зміни клімату, забруднення повітря та води, а також інші проблеми, що впливають на навколишнє середовище та здоров'я людей. У зв'язку з цим стає все важливішим розвиток та застосування систем моніторингу метеорологічних параметрів навколишнього середовища. Метеорологічні параметри, такі як температура, вологість, атмосферний тиск, опади та інші, відіграють ключову роль у регулюванні та збереженні навколишнього середовища. Системи моніторингу цих параметрів дозволяють аналізувати, збирати та використовувати дані для прийняття рішень у сфері екології, геології, сільського господарства, будівництва та інших галузях. Складові системи моніторингу метеорологічних параметрів навколишнього середовища, які надають можливість збирати дані про погодні умови на місцевому рівні. Ці станції дозволяють індивідуальним користувачам, домогосподарствам та навіть малим підприємствам отримувати доступну та точну інформацію про погоду, що є важливим для планування різних видів діяльності та прийняття рішень. Ця дипломна робота спрямована на аналіз існуючих рішень та постановку завдання для системи моніторингу метеорологічних параметрів навколишнього середовища з використанням сучасних технологій та методів. Переваги таких систем включають в себе забезпечення точності та доступності даних, підвищення свідомості населення про погодні умови та їх вплив на навколишнє середовище, а також покращення можливостей вченого співтовариства у вивченні та розумінні кліматичних змін та їх наслідків. Таким чином, розробка та впровадження систем моніторингу метеорологічних параметрів навколишнього середовища є актуальною та важливою задачею, спрямованою на збереження та захист навколишнього середовища для майбутніх поколінь.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз предметної області

Аналіз предметної області є ключовим етапом у підготовці кваліфікаційної роботи, оскільки дозволяє зрозуміти сучасний стан цієї галузі, ідентифікувати основні проблеми та визначити напрямки подальшого дослідження та розвитку. Для вивчення даної предметної області проводиться комплексний аналіз різноманітних аспектів, таких як технічні засоби збору даних, методи їх обробки та аналізу, проблеми збереження та передачі даних, а також можливості використання сучасних інформаційних технологій для покращення систем моніторингу метеорологічних параметрів. Початковий етап аналізу полягає у зборі та огляді науково-технічної літератури, статей, наукових досліджень та інших джерел інформації, що стосуються систем моніторингу метеорологічних параметрів. Це дозволяє ознайомитися з актуальними досягненнями у цій галузі, ідентифікувати основні тенденції та визначити ключові проблеми. Далі проводиться аналіз технічних засобів, що використовуються для збору даних про погоду та кліматичні умови. Це можуть бути різноманітні метеорологічні станції, сучасні сенсори, супутникові системи збору даних тощо. Важливо визначити їхні технічні характеристики, точність вимірювань, можливості роботи в різних умовах тощо. Окрему увагу приділяють аналізу методів обробки та аналізу отриманих даних. Це можуть бути статистичні методи, методи машинного навчання, а також різноманітні програмні засоби для візуалізації та аналізу метеорологічних даних. Значна частина аналізу зосереджується на проблемах збереження та передачі даних. Це включає в себе аналіз методів забезпечення безпеки даних, розробку систем архівування та резервного копіювання, а також вивчення методів передачі даних у реальному часі. В заключенні проводиться узагальнення результатів аналізу та формулювання висновків, які становлять основу для подальших рекомендацій

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

щодо вдосконалення та розвитку систем моніторингу метеорологічних параметрів навколишнього середовища.

1.2 Аналіз існуючих рішень.

Огляд існуючих рішень включає в себе аналіз різноманітних пристроїв, технологій та програмного забезпечення, які використовуються для збору, обробки та аналізу погодних даних. Основні напрямки цього огляду включають в себе: Домашні метеорологічні станції, які є важливими засобами для збору та аналізу погодних даних на дому. Їхня поява сприяла зростанню інтересу громадськості до метеорології та розумінню погодних умов. У науковому дискурсі домашні метеорологічні станції розглядаються як інструменти, що дозволяють звичайним користувачам збирати дані про погоду на місцевому рівні та вносити свій внесок у вивчення кліматичних змін та погодних явищ. Ці станції зазвичай включають в себе різноманітні сенсори, такі як термометр, барометр, гігрометр та анемометр, які призначені для вимірювання температури, атмосферного тиску, вологості та швидкості вітру в реальному часі. Зібрані дані передаються до базової станції або зберігаються на локальних пристроях для подальшого аналізу та використання. Одним із важливих аспектів домашніх метеорологічних станцій є їхня доступність та простота в установці та експлуатації. Це дозволяє власникам без попереднього досвіду в галузі метеорології легко здійснювати спостереження погодних умов у своєму регіоні. Крім того, деякі моделі станцій мають можливість підключення до Інтернету, що дозволяє користувачам спостерігати за погодою в реальному часі та аналізувати зібрані дані за допомогою спеціальних веб-додатків чи програм. Прикладом такої станції може бути MISOL HP2550, її зображено на рисунку 1.1.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9



Рисунок 1.1 – Метеорологічна станція MISOL HP2550

MISOL HP2550 – це метеостанція для домашнього використання або для аматорських метеорологів. Вона дозволяє вимірювати різні параметри навколишнього середовища, такі як температура, вологість, швидкість та напрямок вітру, кількість опадів та атмосферний тиск.

Метеостанція включає великий дисплей-консоль (приймач) розмірами 19,5x13,8x1,9 см, зовнішній датчик та внутрішній датчик (термогігрометр). Пристрій підтримує до 8 термогігрометричних датчиків, які продаються окремо.

На кольоровому дисплеї відображаються такі дані: швидкість і напрямок вітру, температура, вологість, кількість опадів, ультрафіолетове та сонячне випромінювання від зовнішнього датчика бездротового зв'язку; вологість, барометричний тиск та температура від внутрішнього датчика. Крім того, відображаються розрахункові параметри, такі як індекс холоду та тепла, точка роси, середня швидкість і напрямок вітру (оновлюються щосекунди протягом 16 секунд), пориви вітру, 10-хвилинна середня швидкість і напрямок вітру, час сходу і заходу сонця, фаза місяця, прогноз погоди тощо. Зображення дисплея приставлено на рисунку 1.2.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10



Рисунок 1.2. – Дисплей програми

Головними особливостями MISOL HP2550 являються:

- Точність, використання високоякісних датчиків для забезпечення точних вимірювань.
- Надійність, можливість витримувати суворі погодні умови.
- Функціональність, використання широкого спектру функцій, які роблять її цінним інструментом для моніторингу погоди.
- Простота використання та налаштування.
- Доступність в ціні, що робить її застосованою для широкого кола користувачів.

Види MISOL HP2550:

- Стандартна модель, яка включає в себе основні датчики для вимірювання температури, вологості, тиску, швидкості та напрямку вітру, а також кількості опадів.
- Модель з Wi-Fi, яка включає в себе всі функції стандартної моделі, а також можливість підключення до Wi-Fi для завантаження даних на веб-сайт або мобільний додаток.

Незважаючи на численні переваги, MISOL HP2550 має деякі мінуси, а саме відносно дорога вартість, порівняно з деякими іншими моделями на ринку. Також варто відмітити складність налаштування та регулярне обслуговування, багато користувачів, які не мають досвіду відмовляються від цієї метеостанції через її складність налаштування та регулярне калібрування пристрою та датчиків.

Система MISOL WS-2320 (див. рис. 1.3) є автономною метеорологічною системою, яка передає дані бездротовим шляхом від сенсорів на відстань до 100 метрів. Вона призначена для моніторингу різних погодних параметрів, таких як швидкість та напрямок вітру, кількість опадів, температура (в середині та зовні), вологість повітря (в середині та зовні), атмосферний тиск, УФ і сонячне випромінювання, і індекс тепла. Система забезпечена живленням сенсорів від сонячної батареї та акумулятора, що забезпечує її автономність.



Рисунок 1.3 – Метеорологічна станція MISOL WS-2320

Метеостанція має великий дисплей для відображення всіх вимірюваних показників і підтримує Wi-Fi для швидкої передачі даних. Всі результати вимірювань даних у форматі .csv можна завантажити на ПК через Wi-Fi за допомогою програмного забезпечення, яке можна завантажити з офіційного сайту виробника. Приймач для підключення до Wi-Fi (сумісний з Wi-Fi 2,4 ГГц) завантажує дані на веб-сайти, такі як www.wunderground.com, www.weathercloud.net, та wow.metoffice.gov.

У внутрішній пам'яті зберігаються дані вимірювань (3552 записи), а інтервал запису можна встановити від 1 до 240 хвилин. Метеостанція MISOL WS-2320 оснащена датчиками швидкості і напрямку вітру, зовнішньої вологості і температури, вологості і температури в приміщенні, датчиком барометричного тиску, датчиком опадів, сонячною батареєю і спеціальним монтажним кронштейном. Система оснащена наступним. Кожна метеостанція поставляється

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

з диском з програмним забезпеченням, інструкцією по встановленню та налаштуванню українською мовою.

Можливість зберігати максимальні/мінімальні значення цих погодних параметрів і встановлювати режими оповіщення полегшує роботу оператора.

Метеостанція MISOL WS-2320 використовується для моніторингу погодних умов на відкритому повітрі, в теплицях і пластикових парниках, прогнозування погодних умов, моніторингу погодних умов, придатних для сільськогосподарських робіт (посів, зрошення, внесення добрив, засобів захисту рослин, збирання врожаю і т.д.) і попередження про раптові зміни погоди (заморозки, посуха, опади).

Метеостанції MISOL WS-2320 має 2 види, а саме:

- Стандартна модель, яка включає в себе основні датчики для вимірювання температури, вологості, тиску, швидкості та напрямку вітру, а також кількості опадів.
- Модель з Wi-Fi, яка включає в себе всі функції стандартної моделі, а також можливість підключення до Wi-Fi для завантаження даних на веб-сайт або мобільний додаток.

Дослідивши детальніше метеостанцію можливо виділити декілька переваг та недоліків. З переваг – це висока точність, оскільки використовуються високоякісні датчики для забезпечення точних вимірювань. Також варто відмітити її надійність та функціональність через те що, вона може витримувати сурові погодні умови та пропонує широкий спектр функцій.

З мінусів можливо виділити, що в ній відсутні деякі функції такі як датчик ультрафіолетового випромінювання або датчик якості повітря. Ціна достатньо завищена в порівнянні з іншими приладами на ринку.

Наступна мультипараметрична метеостанція MISOL WN2810 (див. рис. 1.4), призначена для вимірювання та моніторингу температури, вологості, абсолютного та відносного тиску і точки роси, а також для відображення поточних погодних умов. Це недорога версія простої у

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

використанні, високоточної та надійної домашньої метеостанції.



Рисунок 1.4 – Метеостанція MISOL WN2810-W1T

Вона оснащена виносним датчиком з власним невеликим дисплеєм, великим РК-дисплеєм з підсвічуванням для зручного зчитування показань, які передаються через бездротовий зв'язок з зовнішнього датчика на відстань до 100 м. Зворотна сторона метеостанція представлена на рисунку 1.5.



Рисунок 1.5 – Зображення заді MISOL WN2810-W1T

Вбудована функція фіксації мінімальних і максимальних значень для всіх параметрів вимірювання, вбудований календар, будильник зі змінним часовим поясом і ручне калібрування користувацької точності (барометричний тиск, внутрішня і зовнішня температура і вологість) значно полегшують роботу користувача.

Бездротова сенсорна домашня метеостанція MISOL WN2810-W1T використовується в різних галузях сільського господарства, для планування залежних від погоди свят і повсякденної діяльності, а також для запобігання кризових ситуацій у метеочутливого населення.

Переваги даної метеостанції – це висока точність та надійність, як для своїх компактних розмірів вона володіє високоточними датчиками та міцним корпусом. Дана метеостанція пропонує широкий спектр функцій за доступну ціну, що вирізняє її від конкурентів на ринку.

Все ж ця метеостанція має декілька мінусів, наприклад можливі проблеми з підключенням до Wi-Fi мережі, відсутність датчика якості повітря та регулярна очистка датчиків від зовнішнього забруднення.

					КР.КІ 24.530.03.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Формалізація вимог до системи

Для створення системи моніторингу метеорологічних параметрів було передбачено наступні вимоги:

- можливість дистанційного зчитування з датчиків;
- доступна ціна;
- легкий монтаж.

Системи для моніторингу метеорологічних параметрів – це комплекс приладів та програмного забезпечення, призначені для збору, обробки та візуалізації даних про різні аспекти погоди та клімату. Структура системи для моніторингу метеорологічних параметрів

Для реалізації мети кваліфікаційної роботи була розроблена структурно - функціональна схема системи моніторингу метеорологічних параметрів навколишнього середовища (СММПНС), яка зображена на рисунку 2.1.

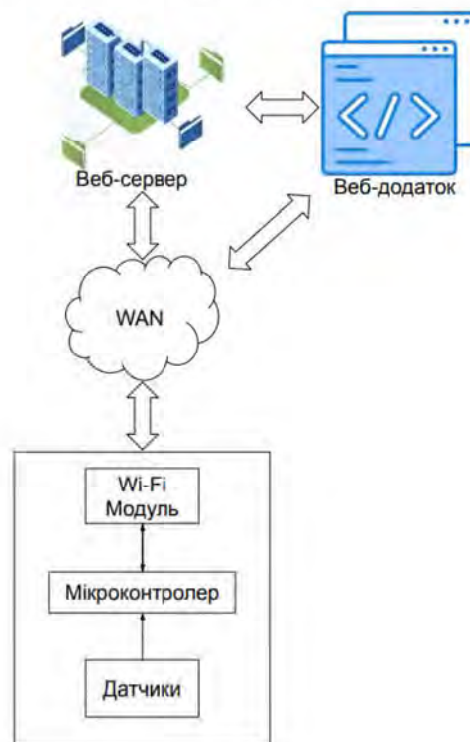


Рисунок 2.1 – Структурно-функціональна схема системи моніторингу
МП

Система спроектована із застосуванням концепції інтернету речей. Запропонована схема включає в себе такі компоненти:

– сукупність пристроїв для моніторингу метеорологічних параметрів (ПММП), які встановлені в локальних спостережних пунктах для широкого територіального охоплення;

– веб-сервер, який використовується як хмарне середовище для зберігання, опрацювання, аналізу результатів моніторингу МП навколишнього середовища. У запропонованій системі датчик передає результати вимірювання метеорологічних параметрів на мікроконтролер (МК) через цифровий інтерфейс; МК виступає основним обчислювальним блоком ПММП, приймаючи, фільтруючи, обробляючи та відображаючи виміряні дані на екрані, відповідаючи за передачу в режимі реального часу на віддалений веб-сервер через бездротовий інтерфейс Wi-Fi. Дані зберігаються в базі даних і можуть бути використані для аналізу, прогнозування та відображення в зручному для користувача форматі для конкретних застосувань. Кваліфікаційна робота передбачає розробку пристрою для вимірювання метеорологічних параметрів (в основному температури, відносної вологості та барометричного тиску). Для досягнення поставленої мети необхідно забезпечити взаємодію між мікроконтролером та високоточним датчиком погоди. Важливою функцією пристрою є обробка погодних даних, що надсилаються з Інтернету. Пристрій повинен відображати інформацію про поточні метеорологічні параметри, такі як температура, відносна вологість, чистота повітря та освітленість, а також підвищити інформативність відображення метеорологічних даних на екрані. На основі аналізу завдання на кваліфікаційну роботу, вимог до спроектованої системи та дослідження аналогічних виробів запропоновано структурну схему пристрою для моніторингу МП (рис. 2.2).

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Основні компоненти, які входять до складу ПММП:

- плата ESP8266 на основі мікроконтролера з вбудованим WiFi модулем;
- датчик вологості та температури;
- датчик забрудненості повітря;
- пасивний зумер;
- фоторезистор.

У проектованій системі за отримання, обробку, передачу та виведення даних відповідає мікроконтролер ESP8266. Він обробляє вхідні сигнали від метеодатчиків і, після обробки даних, передає результати вимірювань на верхній рівень IoT – хмарний веб-сервер.



Рисунок 2.2 – Структурна схема пристрою для моніторингу МП

Метео-датчики в даній системі призначені для вимірювання відносної вологості повітря, чистоти повітря, температури та кількості світла.

Автоматизовані системи контролюють за допомогою контролерів. Контролер це пристрій в електроніці, який використовується для управління різними процесами. Мікроконтролер (MCU, μ C, uC англ. microcontroller) це

мікрокомп'ютер створений на одній мікросхемі, є інтегральною мікропроцесорною системою, реалізованою у вигляді єдиної інтегральної схеми. Термін мікроконтролер походить від основної сфери його застосування, а саме управління електронними пристроями, такими як: офісні пристрої, медичні пристрої, пульти дистанційного керування, електроінструменти, системи управління двигуном автомобіля та інші. Контролером зазвичай називають плату створену на основі мікроконтролера. Мікроконтролер для простоти часто називають скороченою назвою контролер, без приставки мікро. Він відрізняється від мікропроцесора інтегрованими в мікросхему пристроями вводу/виводу, таймерами та іншими периферійними пристроями. Мікроконтролер, по суті, є мікросхемою, яка складається з центрального процесора (в нього також входять блок керування, регістри, постійний запам'ятовувальний пристрій) і периферійних пристроїв (включає в себе порти вводу/виводу, таймери, генератори імпульсів, аналогові перетворювачі та інші). Периферійні пристрої працюють автономно, тобто незалежно від центрального ядра. Після того як периферійний пристрій виконає своє завдання, він може повідомити про це ЦП, а може і не повідомляти. Це залежить від того, як він запрограмований.

Основною ознакою класифікації мікроконтролерів є розрядність даних, які обробляються арифметико-логічним пристроєм (АЛП). Мікроконтролери розділяються на 4-, 8-, 16-, 32- і 64-розрядні. На сьогоднішній день в світі найбільше восьмирозрядних мікроконтролерів (вони займають 50% ринку контролерів).

Будова мікроконтролера включає наступні компоненти. Центральний процесор (CPU) – є основною частиною будь-якого мікроконтролера.

За допомогою процесора всі компоненти мікроконтролера з'єднані в єдину систему. В основі процесора лежить арифметико-логічний пристрій (АЛП), що використовується для обчислення арифметичний та логічних перетворень. Таймери – даний модуль є важливою частиною кожного мікроконтролера, і більшість контролерів забезпечують один або кілька таймерів з роздільною

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

здатністю 8 та/або 16 біт. Таймери використовуються для різноманітних завдань, починаючи від простих затримок вимірювання періодів і закінчуючи генерацією форми сигналу. Основне використання таймера – його функція лічильника. Рахують вони постійно з однією швидкістю, зі швидкістю тактової частоти мікроконтролера. Але таймери, як правило, також дозволяють користувачу фіксувати час виникнення зовнішніх подій, запускати переривання після певної кількості тактових циклів. Пам'ять – зберігає всі програми і дані. Розмір пам'яті, вбудованої в мікроконтролер, обмежений. Однак, оскільки мікроконтролери найчастіше використовуються для відносно простих завдань то вони не потребують надмірного обсягу пам'яті і доцільно включати на мікросхему невелику кількість пам'яті даних і пам'яті інструкцій. Можна чітко розмежувати основні типи пам'яті відповідно до їх функції: регістрова пам'ять, що являє собою невелику пам'ять, вбудовану в центральний процесор. Використовується для тимчасового зберігання значень, з якими працює процесор, її можна назвати короткочасною пам'яттю центрального процесора: пам'ять даних, яка використовується для довгострокового зберігання, центральні процесори зазвичай використовують зовнішню пам'ять, яка набагато більша, ніж регістрова пам'ять. Мікроконтролери зазвичай мають вбудовану пам'ять даних, а саме пам'ять інструкцій, так само як пам'ять даних являється великою зовнішньою пам'яттю. Пам'ять інструкцій теж зазвичай інтегрується безпосередньо в мікроконтролер. Також розрізняють типи пам'яті за їх фізичними властивостями: енергозалежна і енергонезалежна (рис. 2.3).



Рисунок 2.3 – Типи пам'яті

Енергозалежна пам'ять зберігає свій вміст лише до тих пір, поки систему ввімкнено. Енергонезалежна пам'ять набагато повільніша і дорожча за енергозалежну. Енергозалежна пам'ять може бути статичною або динамічною: SRAM (англ. Static Random Access Memory) – статична оперативна пам'ять була першим типом енергозалежної пам'яті. Мікросхема SRAM складається з масиву комірок, кожна здатна зберігати один біт інформації; DRAM (англ. Dynamic Random Access Memory), яка являється динамічною оперативною пам'яттю. Широко застосовуються в якості ОЗП (оперативних запам'ятовуючих пристроїв). На відміну від SRAM і DRAM, енергонезалежна пам'ять (англ. non-volatile memory) зберігається після відключення живлення. Недоліками такого типу пам'яті є те, що вона набагато повільніша та порівняно складніша і дорожча. Типи енергонезалежної пам'яті: ROM (англ. Read Only Memory) – перший тип енергонезалежної пам'яті, з неї можна тільки зчитувати дані, також називається захищеним сховищем даним (ЗСД); PROM (англ. Programmable ROM) – використовується в якості альтернативи до ROM, на нього можна записати інформацію тільки один раз. Недоліком є малий обсяг даних, який можна зберегти. Перевагою є висока захищеність даних; EPROM (англ. Erasable Programmable ROM) – використовується для зберігання невеликої кількості даних.

Крім того, це дозволяє мікроконтролеру записувати і стирати дані, однак це займає набагато більше часу, ніж у оперативної пам'яті, а сам доступ до EPROM є більш складним; EEPROM (англ. Electrically Erasable and Programmable ROM) – на відміну від EPROM для програмування більше не потрібна спеціальна напруга і більше не потрібно джерело ультрафіолетового світла для стирання. Пам'ять EEPROM, на відміну від пам'яті EPROM, може бути стерта лише електричним струмом. Кількість записів даних та видалення даних на EEPROM обмежена, а кількість зчитувань нескінченна. Зараз EEPROM є ідеальним вибором для енергонезалежної пам'яті. Однак є один суттєвий недолік: він дорогий; Flash EEPROM – являється розвитком технології пам'яті

EEPROM. У Flash пам'яті стирання неможливе для кожної адреси, а лише для більших блоків або навіть усієї пам'яті. Таким чином, спрощується внутрішня логіка, що, в свою чергу, значно знижує ціну. NVRAM (англ. Non-Volatile RAM) – поєднує в собі переваги енергозалежної та енергонезалежної пам'яті. Цього можна досягти різними способами. Наприклад додати невелику внутрішню батарею до пристрою SRAM, щоб при відключенні зовнішнього живлення SRAM все ще зберігав свій вміст.

Основна периферія мікроконтролерів включає порти вводу/виводу, які використовуються для зв'язку з зовнішніми пристроями. Ці порти можуть працювати як вхід, так і вихід, дозволяючи передавати інформацію в обидва напрямки. Вони можуть бути цифровими або аналоговими, забезпечуючи можливість моніторингу та управління апаратним забезпеченням. Для перетворення аналогових сигналів у цифровий і навпаки використовується аналогово-цифровий перетворювач (АЦП), який спрощує аналоговий сигнал до дискретної форми. Цифрово-аналоговий перетворювач (ЦАП) – електронний пристрій, який перетворює цифровий сигнал в аналоговий сигнал у вигляді електричного струму або напруги, пропорційних даному числу. Іншими словами, це система, яка перетворює дискретний цифровий сигнал в еквівалентний аналоговий сигнал.

Інтерфейси зв'язку – мікроконтролери зазвичай містять кілька інтерфейсів зв'язку, а іноді навіть кілька екземплярів певного інтерфейсу, наприклад два модулі UART. Основна мета будь-якого такого інтерфейсу полягає в тому, щоб дозволити мікроконтролеру взаємодіяти з іншими блоками, будь то інші мікроконтролери, периферія або хост-ПК. Реалізація таких інтерфейсів може приймати різні форми, але в основному інтерфейси можна класифікувати за кількістю властивостей: вони можуть бути як послідовними, так і паралельними, синхронними або асинхронними, використовувати шину або зв'язок point–point, бути повнодуплексний або напівдуплексний, і може бути заснований на принципі master–slave (ведучий–ведений). Основні типи інтерфейсів;

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

SCI– інтерфейс послідовного зв'язку забезпечує асинхронний інтерфейс зв'язку (англ. Universal Asynchronous Receiver Transmitter, UART). Модуль UART використовує два дроти, передавальну (TXD) і приймальну (RXD) лінію для повного або напівдуплексного зв'язку; SPI (англ. Serial Peripheral Interface) – це простий синхронний інтерфейс point–point, заснований на принципі master–slave. Він забезпечує повнодуплексний зв'язок між ведучим (як правило, контролером) та одним (або більше) веденими пристроями (як правило, периферійними пристроями); ІІС (англ. Inter Integrated Circuit) – це синхронна шина, яка працює за принципом master–slave. Він використовує два дроти SCL (послідовна лінія годинника, англ. Serial Clock Line) та SDA (послідовна лінія передачі даних, англ. Serial Data Line) для напівдуплексного зв'язку. Протокол широко використовується для зв'язку на короткі відстані між одним або кількома контролерами та периферійними пристроями; USB (англ. Universal Serial Bus) – універсальна послідовна шина, що призначена для з'єднання мікроконтролерів та периферійних пристроїв.

2.2 Алгоритм роботи системи

Функціональна схема – схема, що роз'яснює певні процеси, що відбуваються у певних функціональних частинах виробу (устаткування) чи у виробі (устаткуванні) в цілому.

Вибір оптимальних алгоритмів роботи – це етап, на якому необхідно визначитися з тим, що та як саме буде використовуватися для виконання конкретних функцій системи.

Основні алгоритми, які буде використано у системі можна поділити на внутрішні та зовнішні.

До внутрішніх можуть відноситися ті, що необхідно реалізувати всередині ОС мікроконтролера ESP8266 і далі буде наведено приклад таких алгоритмів:

- Ідентифікація всіх зовнішніх датчиків, які будуть підключені до плати та постійне зчитування даних з них. Для реалізації цього будуть

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

використовуватися бібліотеки PyArduino.

- Фільтрація вхідних даних. Для реалізації такого алгоритму необхідно використовувати вже готові програмні та апаратні рішення.
- Перевірка даних про швидкість вітру та якості повітря чи вони не перевищують задані ліміти. При допустимих параметрах світлодіод не буде загоратися, а якщо перевищують, тоді світлодіод буде вмикатися.
- Група зовнішніх алгоритмів буде включати лише реалізацію процесу, що до чого під'єднується та як воно працює.

Результатом роботи є готова система, яка визначає погодні умови. Якщо деякі вхідні данні перевищують задану норму, такі як чистота повітря та швидкість вітру, то загорається індикатор та подається звуковий сигнал, що задана норма того чи іншого значення перевищена. Блок-схеми роботи системи представлена у додатках А.

В ході виконання даної кваліфікаційної роботи було спроектовано алгоритм функціонування моніторингу система метеорологічних параметрів навколишнього середовища. Функціональна схема системи зображено на рисунку 2.4.



Рисунок 2.4 - Функціональна схема автоматизованої метеостанції з мікроконтролерним керуванням

На схемі показано, як сенсори передають аналоговий сигнал від потенціометра. Цей аналоговий сигнал потім конвертується АЦП в цифровий і передається на мікроконтролер. Однак окремий датчик температури вже передає цифровий код 34, що є безпосереднім значенням температури, тому не потребує додаткових перетворень. Два датчики температури використовуються для вимірювання температури як у приміщенні, так і на вулиці одночасно. Також був підключений Bluetooth модуль для подальшої модернізації пристрою. Для цього потрібно створити додаток, через який модуль буде передавати всі отримані дані від інших датчиків. На поточному етапі запис, обробка та передача даних відбувається завдяки вбудованому програмному забезпеченню мікроконтролера.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

3 РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Вибір засобів реалізації

Для проектування системи моніторингу метеорологічних параметрів навколишнього середовища необхідно визначитися з елементною базою майбутнього макету, яка буде відповідати наступним вимогам:

- Точність сенсорів не більше ± 1 °C.
- Захисний корпус (для системи).
- Робота при температурі нижче 0.
- Підключення без додаткових перехідників.
- Можливість заміни датчика при несправності або модернізації приладу.
- Невеликі габарити.
- Можливість передачі даних на додаток на смартфоні.
- Доступність.
- Можливість подальшої модернізації приладу.

Під час вибору мікроконтролера було вирішено, що найбільш підходить ESP8266. Основні особливості, що зупинили вибір на ESP8266:

- ESP8266 доступний за помірну ціну, що робить його доступним для більшості хобістів та професіоналів.
- ESP8266 має вбудований Wi-Fi, що дозволяє легко з'єднувати його з мережами Інтернету та здійснювати взаємодію з іншими пристроями через бездротовий зв'язок.
- ESP8266 підтримує багато мов програмування, включаючи Arduino IDE, MicroPython, Lua і багато інших. Це дає розробникам можливість вибрати найзручніший для них інструмент для розробки.
- Мікроконтролер ESP8266 має вбудований набір GPIO пінів, що дозволяє підключати до нього різноманітні сенсори, актуатори та інші зовнішні пристрої.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

- Завдяки вбудованому Wi-Fi і можливості програмування, ESP8266 ідеально підходить для створення пристроїв для Інтернету речей, таких як датчики, звукові пристрої, контролери освітлення та багато інших.
- ESP8266 має велику спільноту користувачів, яка активно підтримує цей продукт, надає допомогу та ділиться досвідом, що робить процес розробки більш простим і приємним.

Для порівняння обрано чотири часто використовуваних плати: ESP32, Arduino Nano 33 IoT, Particle Photon, NodeMCU. Технічні характеристики плат приведені в табл.3.1

Таблиця 3.1 – Порівняння мікроконтролерів

Характеристики	Види мікроконтролерів				
	ESP8266	ESP32	Arduino Nano 33 IoT	Particle Photon	NodeMCU
Назва мікроконтролера	ESP8266	ESP32	SAMD21G18A (ARM Cortex-M0+)	STM32F205 (ARM Cortex-M3)	ESP8266
Wi-Fi	Вбудований	Вбудований	Вбудований	Вбудований	Вбудований
Bluetooth	Немає	Вбудований	Вбудований	Немає	Немає
Кількість GPIO пінів	9-11	25-36	20	18	10-11
Аналогові входи	1 (10-біт)	18 (12-біт)	8 (12-біт)	Немає	1 (10-біт)
Робоча напруга	3.3V	3.3V	3.3V	3.3V	3.3V
Швидкість процесора	До 160 MHz	До 240 MHz	До 48 MHz	До 120 MHz	До 80 MHz
Пам'ять Flash	512KB-4MB	4MB-16MB	256KB	1MB	4MB
Пам'ять RAM	32KB-160KB	520KB-320KB	32KB	128KB	32KB
Підтримка Arduino IDE	Так	Так	Так	Так	Так
Ціна	Зазвичай низька	Зазвичай помірна	Помірна	Помірна	Зазвичай низька

Виходячи з параметрів, які представлені у табл.3.1 можна зробити висновок, що найменш доцільним вибором будуть плати ESP32, Arduino Nano 33 IoT, Particle Photon через їхню ціну та невеликі переваги перед іншими.

Залишилося два типи плат, а саме ESP8266 та NodeMCU, я обрав ESP8266 через вищу швидкість процесора та більшу кількість RAM пам'яті (див. рис. 3.1) Функціональне призначення виводів модуля наведене на рис. 3.2



Рисунок 3.1 – Зображення ESP8266

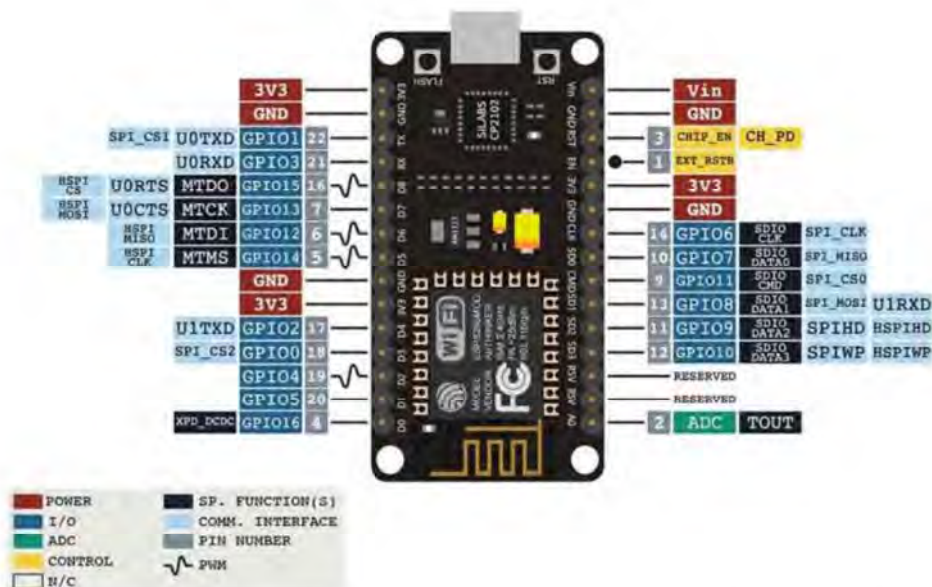


Рисунок 3.2 – Призначення виводів модуля ESP8266

Серед датчиків вологості найбільший попит у сімейства DHT. Вони добре підходять для використання в простих метеостанціях, для підтримки температури приміщень, для систем контролю за кліматом. Для порівняння обрано три найрозповсюдженіших: DHT11, DHT21, DHT22 (табл.3.2)

Таблиця 3.2 – Технічні характеристики датчиків DHT

Характеристика	Типи датчиків DHT		
	DHT11	DHT21	DHT22
Напруга живлення, В	3,3 – 5'	3,3 - 5	3,3 - 5
Діапазон вимірювання вологості, %	20 - 80	0 – 100	0 – 100
Діапазон вимірювання температури, °C	0 - 50	-40 - 80	-40 - 125
Точність вимірювання вологості, %	± 5	± 3	± 2 - 5
Точність вимірювання температури, °C	± 2	± 0,3	± 0,5
Частота опитувана в секунду	1	2	2
Габарити, мм	15,5x12x5,5	60x27x13	15,1x25x5,5

Порівнявши кілька модулів DHT для вимірювання вологості, було обрано датчик вологості та температури DHT11. Датчик DHT21 мав найбільше недоліків у порівнянні з DHT11 та DHT22 через свої великі габарити та нижчу точність вимірювання вологості. DHT11 виявився дешевшим за DHT21, але його точності було недостатньо, оскільки датчик використовувався не лише для вимірювання вологості, а й температури. Для розробки автоматичної метеостанції з мікроконтролерним керуванням було обрано датчик вологості DHT11 (рис. 3.3).



Рисунок 3.3 – Цифровий датчик вологості та температури DHT11

Завдяки достатній точності вимірювання температури, сенсор DHT11 може використовуватися як для вимірювання вологості, так і температури. Це дозволяє заощадити на придбанні окремого датчика та забезпечує компактне розташування елементів на макетній платі. Поєднання двох сенсорів в одному корпусі також усуває потребу створювати окремий отвір для іншого датчика, що покращує естетику конструкції. Сенсор DHT11 складається з термістора та ємнісного датчика вологості. Датчик вологості, що складається з двох електродів з підкладкою, реєструє зміну вологості, утримуючи вологу між електродами. Зміна вологості впливає на провідність основи або опір між електродами. Зчитування температури і вологості здійснюється чіпом всередині датчика, який передає цифровий сигнал до монітора порту. Перед підключенням сенсора DHT11 до плати було додано підтягуючий резистор 10 кОм між виводом VCC та

DATA, щоб забезпечити чітко визначений логічний рівень на виході за будь-яких умов. Потім вивід DATA підключили до 8-го піна, а виводи VCC та GND – до піна 5V та GND відповідно.

Серед датчиків вимірювання газу, було обрано датчик з сімейства MQ, оскільки вони одні з найдоступніших датчиків по ціні. Точність вимірювання достатньо висока попри їхню невелику вартість, а також в цьому сімействі є датчики, які спрямовані під певні гази. Для порівняння було обрано 3 найпопулярніші моделі, а саме: MQ-135, MQ-136 та MQ-137 технічні характеристики яких приведені в табл.3.3

Таблиця 3.3 – Технічні характеристики датчиків MQ

Характеристика	Види датчиків MQ		
	MQ-135	MQ-136	MQ-137
Детектовані гази	CO, NH ₃ , Benzene, Smoke	H ₂ S, SO ₂ , O ₃	NH ₃ , NO _x , Alcohol, Benzene
Робочий напруговий діапазон	5V	5V	5V
Вихідний сигнал	Аналоговий	Аналоговий	Аналоговий
Чутливість	Зазвичай вища	Зазвичай нижча	Зазвичай середня
Діапазон вимірювання	Зазвичай ширший	Зазвичай вузький	Зазвичай середній
Застосування	Домашні та промислові середовища	Вугільні шахти, заводи	Промислові та екологічні дослідження

Порівнявши декілька датчиків вимірювання газу з серії MQ, було обрано MQ-135, через те, що за допомогою нього можна виміряти всі необхідні гази для системи моніторингу метеорологічних параметрів навколишнього середовища або як для домашньої метеостанції. Також в нього вища чутливість і через ці параметри було обрано саме його. (рис 3.4)



Рисунок 3.4 – Датчика газів MQ-135

Обираючи пасивний зумер, вибір зупинився на моделі НУ12-065В, через його доступність та специфіки використання. Пасивний зумер зазвичай використовується для регулювання гучності звуку в аудіоапаратурі, такий як радіоприймачі, підсилювачі аудіосигналу тощо. Цей тип зумера відрізняється від активного зумера тим, що він не потребує додаткового джерела живлення. Ефективність регулювання гучності може варіюватися від моделі до моделі, а розмір пасивного зумера зазвичай залежить від його призначення та використання. Технічні характеристики пасивного зумера НУ12-065В наведені в таблиці 3.4 та рис. пасивного зумера 3.5

Таблиця 3.4 – Характеристика пасивного зумера

Характеристика	НУ12-065В
Тип	Пасивний зумер
Імпеданс	8 ом
Потужність	0.25 Вт
Діапазон частот	200 Гц - 20 кГц
Розмір	12 мм x 6.5 мм
Матеріал корпусу	Пластик
Тип кріплення	Монтаж на плату
Додаткові особливості	Внутрішнє укріплення магнітів



Рисунок 3.5 – Пасивний зумер НУ12-065В

Обираючи фоторезистор вибір зупинився на моделі GL5528, через спектральну чутливість, він сприймає світло в широкому спектрі, включаючи видиме та інфрачервоне світло. Також опір фоторезистора GL5528 змінюється від кількох кілометрів (в темряві) до кількох сотень ом (при яскравому освітленні). Деякі модулі фоторезистора можуть мати вбудований потенціометр для налаштування чутливості фоторезистора або інші додаткові функції. Модулі фоторезистора, такі як GL5528, широко використовуються в різних застосуваннях, таких як автоматичне включення вуличного освітлення, контроль освітленості в приміщеннях, фотореактивні інсталяції та багато інших. Вони є дуже корисними для реалізації функцій, пов'язаних з автоматизацією та збиранням даних про середовище. Технічні характеристики фоторезистора GL5528 наведені в таблиці 3.5 та рис. фоторезистора 3.6

Таблиця 3.6 – Характеристика фоторезистора GL5528

Характеристика	GL5528
Опір в темряві (кОм)	Зазвичай 10-20
Опір у світлі (Ом)	Зазвичай 100-1000
Спектральна відповідь	540 нм
Робоча напруга	3-5 В
Тип кріплення	Припайка



Рисунок 3.6 – Фоторезистор GL5528

Світлодіод був обраний найпростіший червоний, круглої форми (див. рис. 3.7). Також до світлодіоду образу прикріплений резистор на 330 Ом, через те, що червоний світлодіод має велику напругу (приблизно 1.8-2.2 вольт), тому для нього необхідний резистор для обмеження струму. Використовуючи резистор 330 Ом при напрузі 5 вольт, струм через червоний світлодіод буде обмежено до безпечного рівня.



Рисунок 3.7 – Червоний світлодіод з резистором

3.2 Реалізація електричної схеми системи

Для реалізації електричної схеми системи було використано сервіс Fritzing – програма для проектування електронних схем і виробництва прототипів на основі мікроконтролерів та інших електронних компонентів. Вона дозволяє користувачам візуально створювати схеми, розміщувати компоненти на віртуальній дошці, проводити з'єднання між ними, а також створювати схеми

печатних плат для реалізації проектів. Fritzing є корисним інструментом як для початківців, що навчаються основам електроніки, так і для досвідчених інженерів, які шукають зручний спосіб візуалізації своїх проектів. Інтерфейс програми зображено на рисунку 3.8.

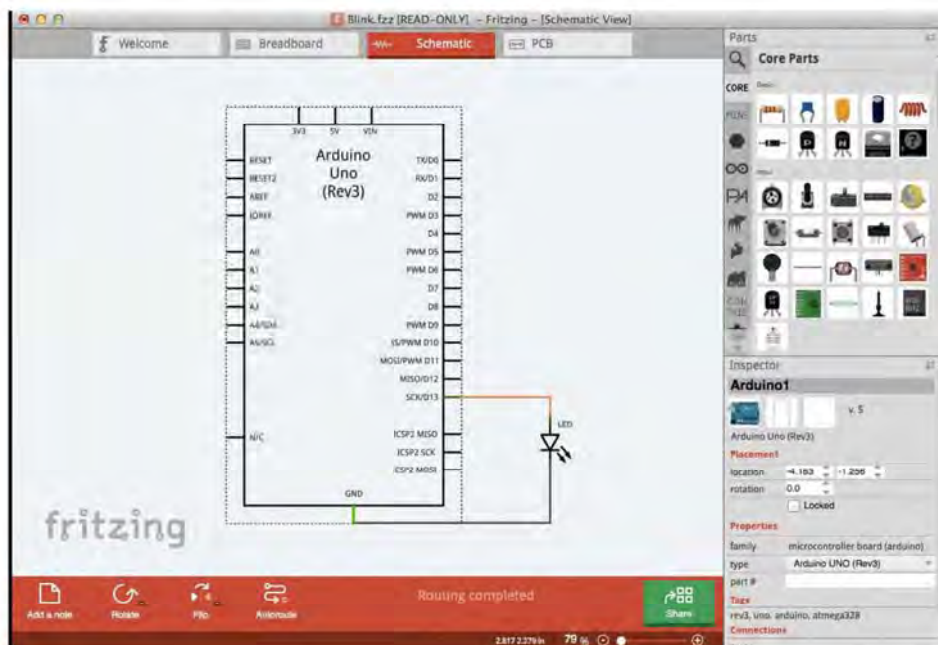


Рисунок 3.8 – Інтерфейс програми Fritzing

Після вивчення функціоналу програми було спроектовано електричну систему. В програмі доступно для вибору безліч мікросхем, контролерів та датчиків, проте в програмі їх обмежена кількість та потрібного може не виявитися, враховуючи що з кожним днем з'являються нові мікросхеми, контролери та датчики і програма не оновлює їхній асортимент. Однак, ця проблема вирішується шляхом скачування та імпортування відсутніх компонентів з їхнього офіційного сайту fritzing.org (див. рис. 3.9).

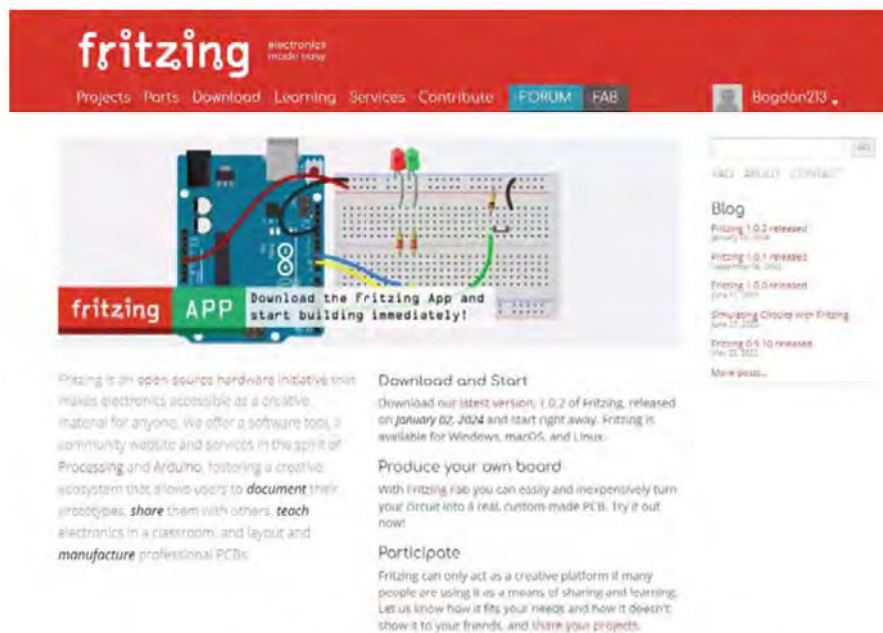


Рисунок 3.9 – Головна сторінка сайту Fritzing

3.3 Реалізація програмного забезпечення

Для реалізації системи було обрано середовище розробки Arduino IDE, яке є інтегрованим кросплатформним додатком (для ОС Windows, MacOS, Linux), написаним на мові програмування Java. Це середовище використовується для написання і завантаження програм на Arduino-сумісні плати, а також, з використанням плагінів, на плати від інших виробників. Вихідний код IDE випущено під загальнодоступною ліцензією GNU версії 2. Arduino IDE підтримує мови C і C++ із спеціальними правилами структурування коду. Крім того, IDE включає бібліотеку програмного забезпечення з проекту Wiring, яка забезпечує безліч загальних процедур введення і виведення.

Arduino IDE використовує програму avrdude для перетворення коду в текстовий файл в шістнадцятковому кодуванні, який завантажується в плату Arduino за допомогою програми-завантажувача в прошивці плати. Інтерфейс середовища Arduino IDE зображено на рисунку 3.10.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

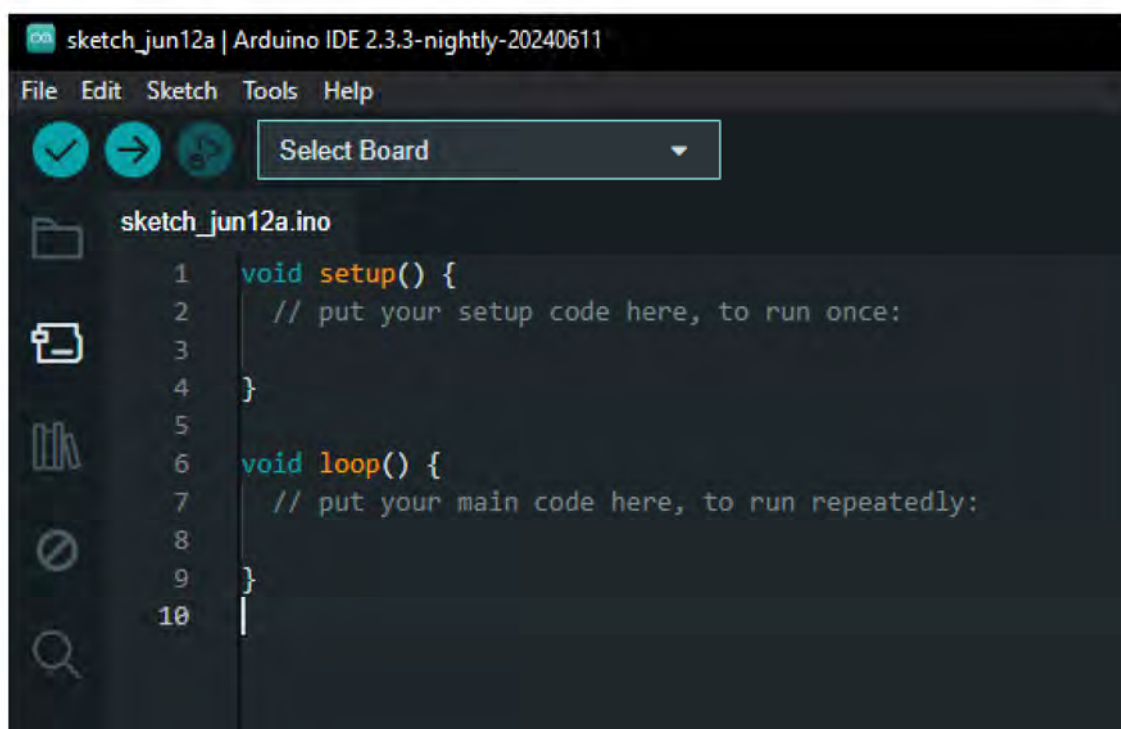


Рисунок 3.10 – Інтерфейс середовища розробки Arduino IDE

Для початку роботи необхідно підключити декілька бібліотек, фрагменти програмного коду для підключення наведено у лістингу 3.1

Лістинг 3.1 – Підключення бібліотек

```
#include <ESP8266WiFi.h>    // Бібліотека для роботи з Wi-Fi
#include <ESP8266HTTPClient.h> // Бібліотека для виконання
HTTP-запитів
#include <ArduinoJson.h>    // Бібліотека для обробки JSON
#include <WiFiClient.h>    // Клієнт Wi-Fi для підключення до
інтернету
#include <DHT.h>            // Бібліотека для роботи з датчиком DHT11
#include <ThingSpeak.h>    // Бібліотека для роботи з платформою
ThingSpeak
```

Передача даних з мікроконтролера в ThingSpeak:

Лістинг 3.2 – Дані для ThingSpeak

```
int keyIndex = 0;
unsigned long myChannelNumber = 2449765;
```



```
const char * myWriteAPIKey = "89M7BVW54JMR8LY1";
```

Підключення до локальної мережі Wi-Fi:

Лістинг 3.3 – Підключення до Wi-Fi

```
const char* ssid = "iot_21";
const char* password = "smarthome";
WiFiClient client;
HTTPClient http;
```

Програмування пінів датчиків та світлодіодів:

Лістинг 3.4 – Піни для датчиків та світлодіодів

```
const int analogInPin = A0;
int sensorValue = 0;
const int airledPin = 0; // Пін для світлодіода якості повітря
const int windledPin = 2; // Пін для світлодіода швидкості вітру
const int DHTPin = 5; // Пін для датчика DHT11
DHT dht(DHTPin, DHT11);
float insideHumidity, insideTemperature, newInsideHumidity,
newInsideTemperature;
const int buzzerPin = 12; // Пін для бузера
const int lightSensorPin = 13; // Пін для датчика освітленості
int lightSensorValue;
```

Ініціалізація серійного порту та формування URL для запиту погоди:

Лістинг 3.5 – Функція setup ()

```
void setup() {
    Serial.begin(115200);
    HTTPUrl = URL;
    HTTPUrl += "lat=" + lat + "&lon=" + lon +
"&lang=ua&units=metric&appid=" + ApiKey;
```

Ініціалізація пінів, відображення підключення до Wi-Fi та ініціалізація ThingSpeak:

Лістинг 3.6 – Ініціалізація піни та підключення до Wi-Fi

```
pinMode(airledPin, OUTPUT);
pinMode(windledPin, OUTPUT);
pinMode(buzzerPin, OUTPUT);
```

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

```

pinMode(lightSensorPin, INPUT);
digitalWrite(airledPin, LOW);
digitalWrite(windledPin, LOW);
digitalWrite(buzzerPin, LOW);

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
ThingSpeak.begin(client); // Ініціалізація ThingSpeak}

```

Зчитування даних з датчиків:

Лістинг 3.7 – Функція loop ()

```

void loop() {
    for (int i = 0; i < 2; i++) {
        newInsideHumidity = dht.readHumidity();
        delay(50);
        newInsideTemperature = dht.readTemperature();
        delay(50);
        lightSensorValue = digitalRead(lightSensorPin);
        sensorValue = analogRead(analogInPin);
    }
    if (!isnan(newInsideHumidity) ||
    !isnan(newInsideTemperature)) {
        insideHumidity = newInsideHumidity;
        insideTemperature = newInsideTemperature;
    }
    String lightState = lightSensorValue == 1 ? "Немає" : "Є";
}

```

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Перевірка Wi-Fi з'єднання та виконання HTTP запиту:

Лістинг 3.8 – Виконання HTTP запиту

```
if (WiFi.status() == WL_CONNECTED) {  
    http.begin(client, HTTPUrl);  
    int httpCode = http.GET();  
    if (httpCode > 0) {  
        String JSON_Data = http.getString();  
        Serial.println(JSON_Data);  
        DynamicJsonDocument doc(2048);  
        deserializeJson(doc, JSON_Data);  
        JsonObject obj = doc.as<JsonObject>();  
        const          char*          description          =  
obj["weather"][0]["description"];  
        const float temp = obj["main"]["temp"];  
        const float humidity = obj["main"]["humidity"];  
        const int pressure = obj["main"]["pressure"];  
        const float wspd = obj["wind"]["speed"];  
        const int wdir = obj["wind"]["deg"];
```

Визначення напрямку вітру та зміна градусної міри на сторони світу:

Лістинг 3.9 – Визначення напрямку вітру

```
String WindDir = "";  
if (wdir >= 337.5 || (wdir >= 0 && wdir < 22.5)) {  
    WindDir = "Північ";  
} else if (wdir >= 22.5 && wdir < 67.5) {  
    WindDir = "Північний схід";  
} else if (wdir >= 67.5 && wdir < 112.5) {  
    WindDir = "Схід";  
} else if (wdir >= 112.5 && wdir < 157.5) {  
    WindDir = "Південний схід";  
} else if (wdir >= 157.5 && wdir < 202.5) {  
    WindDir = "Південь";
```

```

} else if (wdir >= 202.5 && wdir < 247.5) {
    WindDir = "Південний захід";
} else if (wdir >= 247.5 && wdir < 292.5) {
    WindDir = "Захід";
} else if (wdir >= 292.5 && wdir < 337.5) {
    WindDir = "Північний захід";
}

```

Виведення інформації про погоду, температуру, швидкість вітру, тиск зовні, тощо:

Лістинг 3.10 – Виведення інформації про погоду на серійний порт

```

Serial.print("Погода ззовні: ");
Serial.println(description);
Serial.print("Температура ззовні: ");
Serial.print(temp);
Serial.println(" C");
Serial.print("Вологість ззовні: ");
Serial.print(humidity);
Serial.println(" %");
Serial.print("Повітряний тиск ззовні: ");
Serial.print(pressure);
Serial.println(" мбар");
Serial.print("Швидкість вітру: ");
Serial.print(wspd);
Serial.print(" м/с, ");
Serial.print("Напрямок: ");
Serial.println(WindDir);
Serial.print("Температура всередині: ");
Serial.print(insideTemperature);
Serial.println(" C");
Serial.print("Вологість всередині: ");
Serial.print(insideHumidity);
Serial.println(" %");
Serial.print("Наявність освітлення: ");

```



```
Serial.println(lightState);
```

Вмикання або вимикання світлодіоду при перевищенні заданої норми швидкості вітру:

Лістинг 3.11 – Управління світлодіодом швидкості вітру

```
if (wspd >= 10){ Відправка даних на ThingSpeak

    digitalWrite(windledPin, HIGH);    // Вмикає світлодіод, якщо
швидкість вітру більша або дорівнює 10 м/с
} else {
    digitalWrite(windledPin, LOW);    // Вимикає світлодіод, якщо
швидкість вітру менша за 10 м/с
}
```

Відправка зібраних даних на платформу ThingSpeak. Це дозволяє зберігати та візуалізувати дані, зібрані з датчиків:

Лістинг 3.12 – Відправка даних на ThingSpeak

```
ThingSpeak.setField(1, temp);                // Встановлює
значення температури в перше поле
ThingSpeak.setField(2, humidity);            // Встановлює
значення вологості в друге поле
ThingSpeak.setField(3, wspd);                // Встановлює
значення швидкості вітру в третє поле
ThingSpeak.setField(4, wdir);                // Встановлює
значення напрямку вітру в четверте поле
ThingSpeak.setField(5, insideTemperature);    // Встановлює
значення температури всередині в п'яте поле
ThingSpeak.setField(6, insideHumidity);       // Встановлює
значення вологості всередині в шосте поле
ThingSpeak.setField(7, sensorValue);          // Встановлює
значення якості повітря в сьоме поле
ThingSpeak.setField(8, lightSensorValue);     // Встановлює
значення освітленості в восьме поле
```

```

    int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
// Відправка даних на ThingSpeak
    if (x == 200){
        Serial.println("Channel update successful.");    // Успішне
оновлення каналу
    } else {
        Serial.println("Problem updating channel. HTTP error code "
+ String(x)); // Помилка при оновленні каналу
    }

```

Індикація якості повітря за допомогою світлодіода та бузера:

Лістинг 3.13 – Управління світлодіодом та бузером для якості повітря

```

if (sensorValue >= 750){
    digitalWrite(airledPin, HIGH);    // Вмикає світлодіод якості
повітря
    digitalWrite(buzzerPin, HIGH);    // Вмикає бузер
    delay(100);                        // Затримка 100 мс
    digitalWrite(airledPin, LOW);     // Вимикає світлодіод якості
повітря
    digitalWrite(buzzerPin, LOW);     // Вимикає бузер
} else {
    digitalWrite(airledPin, LOW);     // Вимикає світлодіод якості
повітря
    digitalWrite(buzzerPin, LOW);     // Вимикає бузер
}
Serial.print("Quality of air: ");
Serial.println(sensorValue);          // Виводить значення якості
повітря на серійний порт

```

```

delay(750);                            // Затримка 750 мс
digitalWrite(buzzerPin, LOW);          // Вимикає бузер

```

3.4 Тестування роботи системи

Тестування – це процес перевірки повної програмної системи на

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

відповідність визначеним вимогам і очікуванням користувачів. Метою цього виду тестування є виявлення дефектів у системі, які можуть вплинути на її функціональність, продуктивність та безпеку. Системне тестування охоплює перевірку всіх компонентів і модулів програми, їхню взаємодію та інтеграцію, а також відповідність кінцевого продукту встановленим стандартам і специфікаціям. Цей процес забезпечує комплексну оцінку якості програмного забезпечення перед його випуском або впровадженням у використання.

До завдань тестування не належить визначення причин невідповідності заданим вимогам (специфікаціям). Тестування один з розділів діагностики.

Тестування використовується в різних галузях і сферах для забезпечення якості та надійності програмного забезпечення. Воно є невід'ємною частиною процесу розробки програмного забезпечення, включаючи розробку веб-додатків, мобільних додатків, ігрового програмного забезпечення, вбудованих систем та корпоративних інформаційних систем. Тестування також застосовується у фінансовому секторі для перевірки банківських систем, в медичній галузі для перевірки програм для управління здоров'ям і діагностики, а також в авіаційній та автомобільній промисловості для забезпечення безпеки та надійності систем управління. Крім того, тестування важливе в телекомунікаціях для перевірки мережевих протоколів та обладнання, а також у державному секторі для перевірки різних державних систем і сервісів.

Процес тестування включає:

- Планування та розробка тестів, а саме визначення мети тестування, обсягу робіт, ресурсів і розкладу. Створення тестових випадків і сценаріїв на основі вимог до програмного забезпечення.
- Підготовка тестового середовища та виконання тестів. Створення та налаштування середовища для тестування, включаючи встановлення необхідного програмного та апаратного забезпечення і виконання тестових завдань.
- Аналіз та звітування про результати тестування, створення звітів про

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

виявлені проблеми, повторне тестування виправлених дефектів і регресійне тестування для перевірки, що зміни не призвели до нових дефектів.

У технічній діагностиці використовуються стандартизовані підходи до розробки тестів для перевірки працездатності і відповідності специфікаціям. Наприклад, в електронній промисловості формалізовані підходи до розробки тестів для цифрових пристроїв можуть ґрунтуватися на моделях статичних несправностей, таких як розриви і коротке замикання, а також на статистичних даних про несправності, що спостерігаються в схожих пристроях.

Якість – це поняття не абсолютне, а суб'єктивне. Тому тестування як процес своєчасного виявлення помилок і несправностей не може повністю гарантувати правильне функціонування програмного забезпечення. Воно лише порівнює стан і поведінку продукту зі специфікаціями.

Проведемо тестування системи моніторингу метеорологічних параметрів навколишнього середовища.

Після включення системи повинен увімкнутись червоний та зелений світлодіоди, які знаходяться на датчику якості повітря (див. рисунок 3.11) та повинен увімкнутися червоний світлодіод на датчику температури та вологості (див. рисунок 3.12)

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

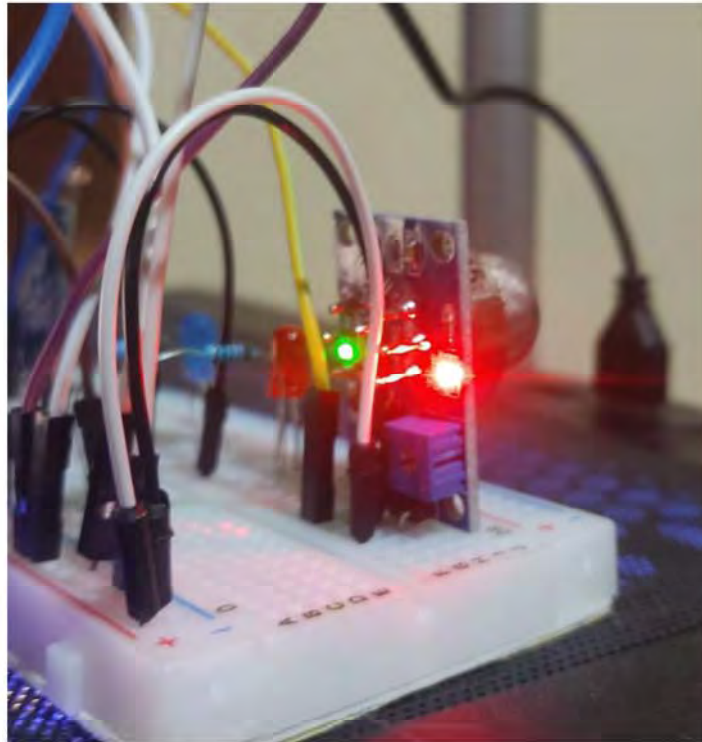


Рисунок 3.11 – Робота датчика газів MQ-135

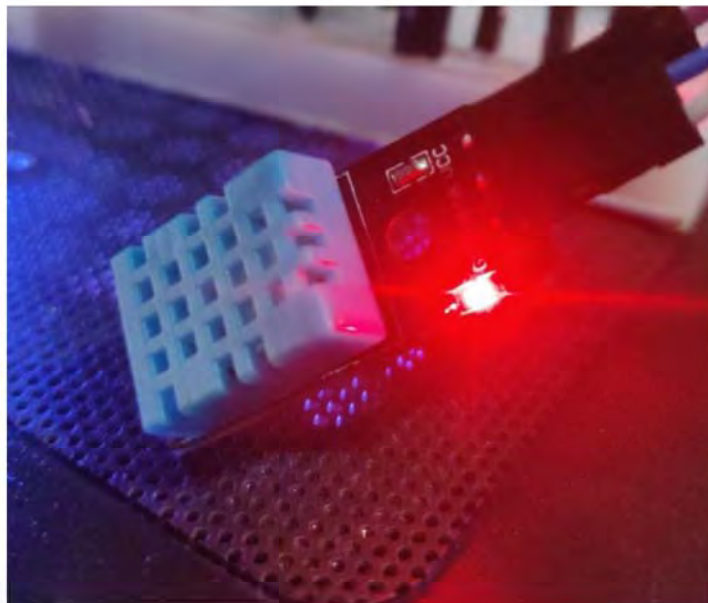


Рисунок 3.12 – Робота датчика температури та вологості DHT11

При ввімкненні живлення система починає автоматично зчитувати данні з датчиків та відправляє їх на сервер ThingSpeak з періодичністю 30 секунд (див. рисунок 3.13)

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

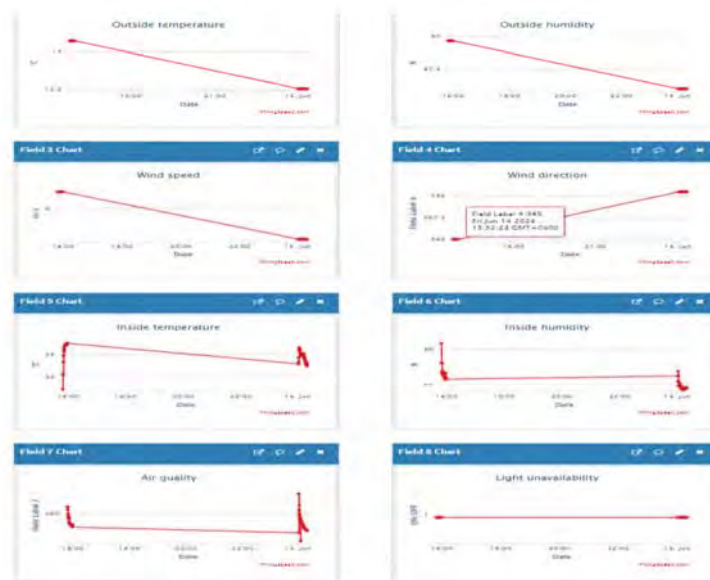


Рисунок 3.13 – Передача даних з датчиків на сервер ThingSpeak

Особливістю системи є можливість швидкого оновлення даних не зважаючи на затримку (див. рисунок 3.14), це можливо завдяки додатку “ThingView Free”, який доступний для смартфонів на базі операційної системи Android та IOS. Щоб увійти в свій канал потрібно ввести персональний API ключ та Channel ID, ці дані доступні після реєстрації на офіційному сайті на ThingSpeak.

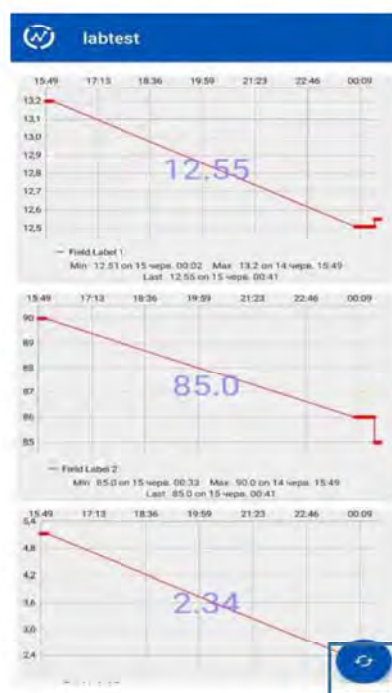


Рисунок 3.14 – Швидка обнова даних з датчиків

Змн.	Арк.	№ докум.	Підпис	Дата

КР.КІ 24.530.03.000 ПЗ

Арк.

47

Також через додаток можливо переглянути як змінювалися ті чи інші параметри протягом певного часу. Зверху зображена часова шкала та можливо відслідкувати по ній в які проміжки часу графік змінювався, посередині розміщено актуальні дані на цей період часу, зліва зображено одиниці вимірювання. При натисканні на графік відображається година, день, місяць, рік та актуальне значення на цей період часу. Знизу відображаються мінімальні та максимальні дані за період роботи системи, а також останні дані зафіксовані системою (рисунок 3.15).

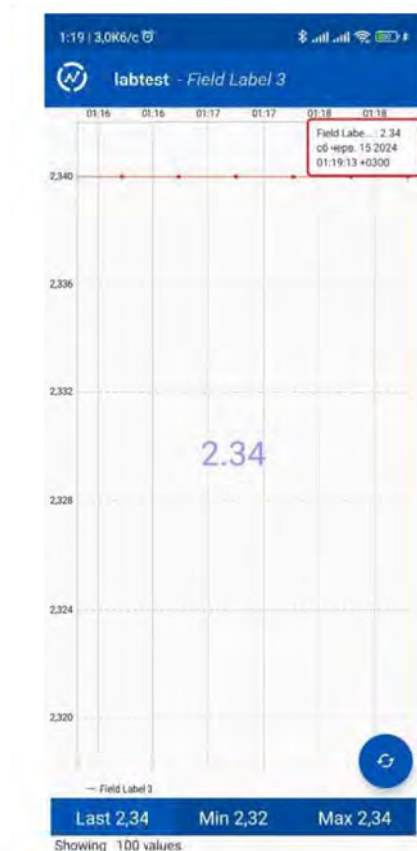


Рисунок 3.15 – Відображення даних в додатку

4 ТЕХНІКО ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

4.1 Аналіз ринку

Виробництво якісної продукції потребує проведення техніко-економічних розрахунків для визначення оптимальних рішень технічних питань на етапах підготовки, проектування, впровадження, експлуатації, обслуговування, ремонту та утилізації електрообладнання. Отже, важливою складовою є формування знань і набуття практичних навичок у вирішенні таких питань, як техніко-економічне обґрунтування проектних рішень та розрахунок вартості на проектування.

Системи моніторингу метеорологічних параметрів навколишнього середовища представляють собою ключовий сегмент у сучасних технологіях, орієнтованих на збір і аналіз даних про погодні умови. Ринок цих систем демонструє стабільний зростання, що обумовлене збільшеним попитом від різних секторів, включаючи сільське господарство, енергетику, транспорт та наукові дослідження. Основні технічні характеристики, які визначають конкурентоспроможність таких систем, включають точність вимірювань, швидкість збору та передачі даних, стійкість до погодних умов і ефективність енергоспоживання.

У контексті нового виробу, важливо врахувати його основні техніко-економічні та експлуатаційні характеристики, щоб забезпечити його ефективну роботу в умовах різних кліматичних зон. Цей виріб є модифікацією існуючих рішень на ринку з покращеною точністю та функціональністю, що спрямована на задоволення потреб сучасних користувачів.

Потенційними замовниками такого виробу є метеорологічні служби, державні органи, науково-дослідні установи, а також приватні компанії, що займаються аграрним виробництвом чи відновлюваною енергетикою. Реалізація таких систем відбувається на глобальному рівні, з урахуванням специфіки кліматичних умов різних регіонів.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Прогнозується значний попит на цей виріб, особливо в контексті змін клімату та необхідності точних прогнозів для різних галузей економіки. Для ефективного реалізації ми рекомендуємо використовувати комбінацію прямих продажів через власну мережу та партнерських відносин, а також акцентувати увагу на високоякісному сервісному обслуговуванні як під час, так і після продажу.

Рівень цін на продукцію конкурентів варіюється в залежності від багатьох факторів, включаючи технічні характеристики, якість виконання, брендову визнаність та обсяги виробництва. Зазвичай цінова політика конкурентів на ринку систем моніторингу метеорологічних параметрів може бути поділена на кілька сегментів: преміум-сегмент, середній ціновий сегмент і бюджетні рішення. Продукція в преміум-сегменті варіюється від 5 000\$ до 50 000\$ за одиницю, в середньому ціновому сегменті від близько 500\$ до 5 000\$ та в бюджетному до 500\$ за одиницю.

4.2 Розрахунок витрат на проектування

Стаття 6 Закону про оплату праці та стаття 96 Кодексу законів про працю передбачають, що мінімальний посадовий оклад (тарифна ставка) працівника не може бути меншим за розмір прожиткового мінімуму, встановленого на 1 січня календарного року. У 2024 році цей мінімальний розмір становить 8 000 грн.

Розробка систем моніторингу метеорологічних параметрів навколишнього середовища, передбачає наступний склад розробників: інженер I категорії та старший науковий співробітник.

Інженеру I категорії нараховано за повний відпрацьований місяць 8462 грн. Податкова соціальна пільга до такої заробітної плати не застосовується, оскільки вона більша за граничний розмір доходу, який дає право на податкову соціальну пільгу.

- Податок на доходи фізичних осіб: $8462 \text{ грн.} \cdot 18\% = 1523,16 \text{ грн.}$
- Військовий збір: $8462 \text{ грн.} \cdot 1,5\% = 126,93 \text{ грн.}$

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

- Єдиний внесок: 8462 грн. * 22% = 1861,64 грн.
- Утримання = 1605,09 (1523,16 грн. + 126,93 грн.)
- До виплати працівникові = 6856,91 грн. (8462 грн. – 1523,16 грн. – 126,93 грн.)

Старшому науковому співробітнику нараховано за повний відпрацьований місяць 10837 грн. Податкова соціальна пільга до такої заробітної плати не застосовується, оскільки вона більша за граничний розмір доходу, який дає право на податкову соціальну пільгу.

- Податок на доходи фізичних осіб: 10837 грн. * 18% = 1950,66 грн.
- Військовий збір: 10837 грн. * 1,5% = 162,55 грн.
- Єдиний внесок: 10837 грн. * 22% = 2384,14 грн.
- Утримання = 2113,21 грн. (1950,66 грн. + 162,55 грн.)
- До виплати працівникові = 8723,79 грн. (10837 грн. – 1950,66 грн. – 162,55 грн.)

Таблиця 4.1 – Розрахування заробітної плати

Посада	Оклад	Відрахування	Кількість		Сума
Інженер I категорії	8462 грн./міс.	1605,09 грн./міс.	1 чол.	1 міс.	6856,91 грн.
Старший науковий співробітник	10837 грн./міс.	2113,21 грн./міс.	1 чол.	1 міс.	8723,79 грн.
	Всього зарплати:				15580,07 грн.

Співробітники не були у відрядженні, тому витрат на відрядження не має.

Інші прямі витрати за місяць становлять – 8684,55 грн. (видатків на заробітну плату * 45% - (8462 грн. + 10837 грн.) * 0,45).

Всього прямих витрат за місяць 27263,08 грн. (8462 грн. + 10837 грн. + 1861,64 грн. + 2384,14 грн. + 3718,3 грн.)

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Накладні витрати за місяць становлять 92,06 грн. ($B = 3\text{Посн.} * \% B : 100 - 23017,3 * 0,4/100$)

Планове накопичення визначається у відстоках (20-30%) від суми прямих і накладних витрат.

Планові накопичення за місяць становлять 6838,78 грн. $((27263,08 + 92,06) * 0,25)$

Всього кошторисна вартість проєкту 34193,92 грн. (сума прямих, накладних та планових накопичень)

Податок на додану вартість (різниця між вартістю продукції, що випускає підприємство, та вартістю засобів виробництва) становить 5083,46 грн. $((34193,92 - 92,06 - 8684,55) * 0,2)$

Договірна ціна становить 39276,48 грн. (сума кошторисної вартості роботи та податку на додану вартість)

4.3 Обґрунтування необхідності розробки

Розробка нової системи моніторингу метеорологічних параметрів навколишнього середовища є нагальною потребою, враховуючи кілька важливих факторів. По-перше, зміни клімату та підвищення частоти екстремальних погодних явищ вимагають точного і надійного моніторингу для своєчасного прогнозування та прийняття управлінських рішень. Сучасні системи моніторингу повинні забезпечувати високу точність вимірювань, оперативність збору та аналізу даних, а також стійкість до суворих погодних умов.

По-друге, зростаючий попит на метеорологічні дані з боку різних галузей економіки, таких як сільське господарство, енергетика, транспорт і туризм, обґрунтовує необхідність впровадження нових, більш ефективних систем. Наприклад, аграрний сектор потребує точних даних для оптимізації зрошення та запобігання втратам врожаю, а енергетичні компанії використовують метеорологічні дані для підвищення ефективності вітрових і сонячних електростанцій.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

По-третє, існуючі системи часто мають обмеження в функціональності або є занадто дорогими для широкого впровадження. Розробка нової системи, яка поєднує високу точність, доступну ціну та широкий набір функцій, сприятиме її поширенню і дозволить задовольнити потреби більшої кількості користувачів.

Нарешті, технологічний прогрес в галузі сенсорів, засобів передачі даних та аналітичних платформ створює нові можливості для розробки більш досконалих систем моніторингу. Використання сучасних технологій дозволить створити інноваційний продукт, який відповідатиме вимогам ринку та забезпечить конкурентні переваги.

Таким чином, розробка нової системи моніторингу метеорологічних параметрів навколишнього середовища є не лише виправданою, але й необхідною для забезпечення ефективного управління кліматичними ризиками та задоволення зростаючих потреб різних секторів економіки.

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

ВИСНОВКИ

Було проведено системний аналіз предметної області на основі літературних джерел, по темі система метеорологічних параметрів навколишнього середовища. В ході аналізу була визначена мета розробки системи, були окреслені вхідні та вихідні дані та визначені вимоги до системи.

Було підібрано всі необхідні компоненти, а саме: мікроконтролер ESP8266, цифровий датчик вологості та температури DHT11, датчика газів MQ-135, пасивний зумер HY12-065B та фоторезистор GL5528. Також був зроблений акцент на доступність всіх компонентів та використання безкоштовного програмного забезпечення для доступу ширшому колу користувачів.

В результаті роботи було виконано поставлені завдання:

- проведено аналіз предметної області та огляд існуючих рішень;
- формалізація вимог до функціональної системи;
- проектування та реалізація структурної схеми;
- створення автоматизованої системи;
- тестування реалізованої системи.

Дану систему можливо вдосконалювати шляхом додавання інших датчиків або заміною існуючих на досконаліші аналоги. Також можливо доробити інтерфейс додатку та вдосконалити взаємодію користувача з системою, шляхом автоматичного визначення місцезнаходження користувача та надання даних саме з цієї точки, також додати прогнозування погоди на найближчий час шляхом аналізу вхідних даних.

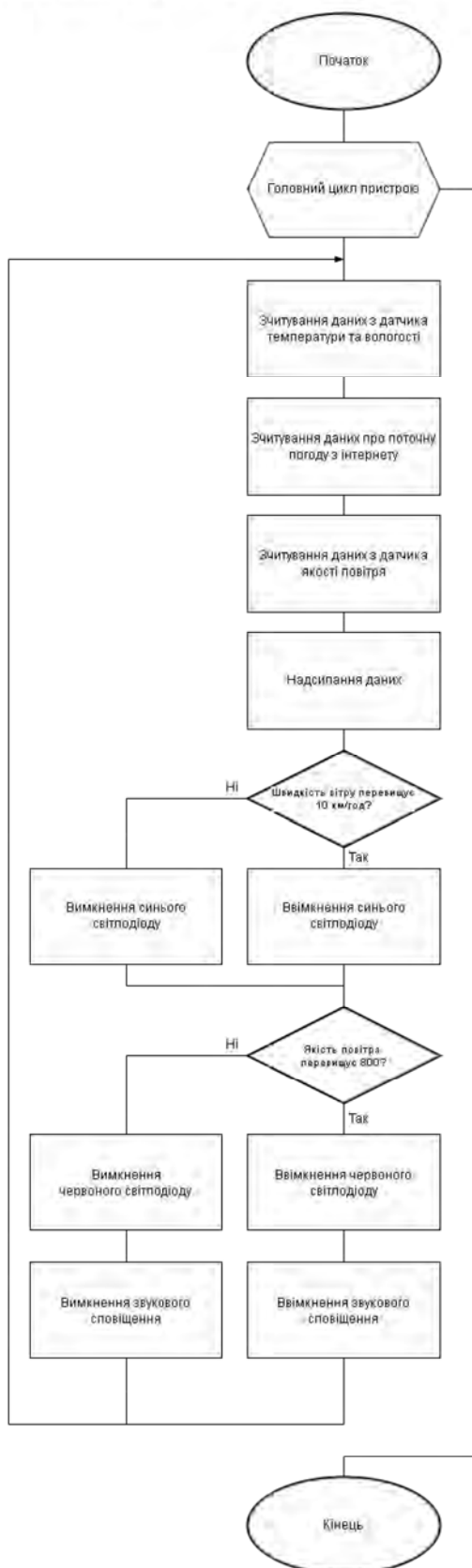
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ESP8266 WiFi Module. Espressif Systems: офіційний веб-сайт URL: <https://www.espressif.com/en/products/socs/esp8266> (дата звернення: 03.03.2024)
2. DHT11 Temperature and Humidity Sensor. Adafruit: офіційний веб-сайт URL: <https://www.adafruit.com/product/386> (дата звернення: 15.03.2024)
3. MQ-135 Gas Sensor. SparkFun: офіційний веб-сайт URL: <https://www.sparkfun.com/products/9403> (дата звернення: 25.04.2023)
4. Пасивний зумер HY12-065B. Amazon: офіційний веб-сайт URL: <https://www.amazon.com/dp/B07P4K7V5H> (дата звернення: 26.04.2024)
5. Arduino IDE. Arduino: офіційний веб-сайт URL: <https://www.arduino.cc/en/software> (дата звернення: 27.05.2024)
6. Методичні рекомендації до виконання кваліфікаційної роботи/ Павлюс В.П., Посвятовська О.Б., Кульчинська Н.З. – Галицький фаховий коледж імені В'ячеслава Чорновола, Тернопіль, 2023. 52с.
7. Закон "Про Державний бюджет України на 2022 рік" : веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/2849-20#n2352> (дата звернення: 09.06.2024).

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

ДОДАТКИ

Додаток А. Блок-схема алгоритму роботи системи



Змн.	Арк.	№ докум.	Підпис	Дата

КР.КІ 24.530.03.000 ПЗ

Арк.

56

Додаток Б
Лістинг програмного коду

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>
#include <WiFiClient.h>
#include <DHT.h>
#include <ThingSpeak.h>

int keyIndex = 0;
unsigned long myChannelNumber = 2577082;
const char * myWriteAPIKey = "U1L62CBH8ZGWMRXO";
//Put your WiFi Credentials here
const char* ssid = "Bogdan";
const char* password = "21007620";
WiFiClient client;
HTTPClient http;
//URL Endpoint for the API
String URL =
"http://api.openweathermap.org/data/2.5/weather?";
String ApiKey = "0bd75c98f407bfbcc332749988873151";
// Replace with your location Credentials
String lat = "49.55304943989859";
String lon = "25.59898239652207";
String HTTPUrl = "";
const int analogInPin = A0;
int sensorValue = 0;
const int airledPin = 0;
const int windledPin = 2;
const int DHTPin = 5;
DHT dht(DHTPin, DHT11);

float insideHumidity, insideTemperature, newInsideHumidity,
newInsideTemperature;
```

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57


```

const int buzzerPin = 12;
const int lightSensorPin = 13;
int lightSensorValue;

void setup() {
    Serial.begin(115200);
    HTTPUrl = URL;
    HTTPUrl += "lat=";
    HTTPUrl += lat;
    HTTPUrl += "&lon=";
    HTTPUrl += lon;
    HTTPUrl += "&lang=ua";
    HTTPUrl += "&units=metric&appid=";
    HTTPUrl += ApiKey;
    pinMode(airledPin, OUTPUT);
    pinMode(windledPin, OUTPUT);
    pinMode(buzzerPin, OUTPUT);
    pinMode(lightSensorPin, INPUT);
    digitalWrite(airledPin, LOW);
    digitalWrite(windledPin, LOW);
    digitalWrite(buzzerPin, LOW);
    // We start by connecting to a WiFi network
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    ThingSpeak.begin(client);
}

```

```

void loop() {
    // wait for WiFi connection
    for(int i = 0; i < 2; i++){
        newInsideHumidity = dht.readHumidity();
        delay(50);
        newInsideTemperature = dht.readTemperature();
        delay(50);
        lightSensorValue = digitalRead(lightSensorPin);
        sensorValue = analogRead(analogInPin);
    }
    if ( !(isnan(newInsideHumidity)) ||
    !(isnan(newInsideTemperature)) ){
        insideHumidity = newInsideHumidity;
        insideTemperature = newInsideTemperature;
    }
    String lightState = "";
    if(lightSensorValue == 1){
        lightState = "Hemac";
    } else if (lightSensorValue == 0){
        lightState = "C";
    }

    if (WiFi.status() == WL_CONNECTED) {

        //Set HTTP Request Final URL with Location and API key
        information
        http.begin(client, HTTPUrl);
        // start connection and send HTTP Request
        int httpCode = http.GET();
        // httpCode will be negative on error
        if (httpCode > 0) {
            //Read Data as a JSON string
            String JSON_Data = http.getString();
            Serial.println(JSON_Data);
        }
    }
}

```



```

        //Retrieve some information about the weather from the
JSON format
        DynamicJsonDocument doc(2048);
        deserializeJson(doc, JSON_Data);
        JsonObject obj = doc.as<JsonObject>();
        //Display the Current Weather Info
        const char* description =
obj["weather"][0]["description"].as<const char*>();
        const float temp = obj["main"]["temp"].as<float>();
        const float humidity =
obj["main"]["humidity"].as<float>();
        const int pressure = obj["main"]["pressure"].as<int>();
        const float wspd = obj["wind"]["speed"].as<float>();
        const int wdir = obj["wind"]["deg"].as<int>();
        String WindDir = "";
        if (wdir >= 337,5 || (wdir >= 0 && wdir < 22,5)){
            WindDir = "Північ";
        } else if (wdir >= 22,5 && wdir < 67,5){
            WindDir = "Північний схід";
        } else if (wdir >= 67,5 && wdir < 112,5){
            WindDir = "Північний схід";
        } else if (wdir >= 112,5 && wdir < 157,5){
            WindDir = "Південний схід";
        } else if (wdir >= 157,5 && wdir < 202,5){
            WindDir = "Південь";
        } else if (wdir >= 202,5 && wdir < 247,5){
            WindDir = "Південний захід";
        } else if (wdir >= 247,5 && wdir < 292,5){
            WindDir = "Захід";
        } else if (wdir >= 292,5 && wdir < 337,5){
            WindDir = "Північний захід";
        }
    }

```

```

Serial.print("Погода ззовні: ");
Serial.println(description);
Serial.print("Температура ззовні: ");
Serial.print(temp);
Serial.println(" C");
Serial.print("Вологість ззовні: ");
Serial.print(humidity);
Serial.println(" %");
Serial.print("Повітряний тиск ззовні: ");
Serial.print(pressure);
Serial.println(" мбар, ");
Serial.print("Швидкість вітру: ");
Serial.print(wspd);
Serial.print(" м/с, ");
Serial.print("Direction: ");
Serial.println(WindDir);
Serial.print("Температура всередині: ");
Serial.print(insideTemperature);
Serial.println(" C");
Serial.print("Вологість всередині: ");
Serial.print(insideHumidity);
Serial.println(" %");
Serial.print("Наявність освітлення: ");
Serial.println(lightState);
if (wspd >= 10){
    digitalWrite(windledPin, HIGH);
} else {
    digitalWrite(windledPin, LOW);
}
ThingSpeak.setField(1, temp);
ThingSpeak.setField(2, humidity);
ThingSpeak.setField(3, wspd);
ThingSpeak.setField(4, wdir);

```



```

ThingSpeak.setField(5, insideTemperature);
ThingSpeak.setField(6, insideHumidity);
ThingSpeak.setField(7, sensorValue);
ThingSpeak.setField(8, lightSensorValue);
int x = ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);
    if(x == 200){
        Serial.println("Channel update successful.");
    } else{
        Serial.println("Problem updating channel. HTTP error
code " + String(x));
    }
    } else {
        Serial.println("Error!");
    }
    http.end();

}
if (sensorValue >= 750){
    digitalWrite(airledPin, HIGH);
    digitalWrite(buzzerPin, HIGH);
    delay(100);
    digitalWrite(airledPin, LOW);
    digitalWrite(buzzerPin, LOW);
} else{
    digitalWrite(airledPin, LOW);
    digitalWrite(buzzerPin, LOW);
}
Serial.print("Quality of air: ");
Serial.println(sensorValue);
delay(750);
digitalWrite(buzzerPin, LOW);
//Wait for 30 seconds
delay(29150); }

```

					КР.КІ 24.530.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62