

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеню вищої освіти – магістр
за освітньо-професійною програмою
«Киберфізичні системи в промисловості, бізнесі та транспорті»

зі спеціальності

174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

тема роботи:

«Система дистанційного моніторингу якості повітря з використанням технологій інтернет речей »

Виконав ст. гр. АКІТР-23-2м.	_____	Білецький Я.А.
Керівник	_____	Рубан С.А.
Нормоконтроль	_____	Маринич І. А.
Завідувач кафедри	_____	Рубан С. А.

Кривий Ріг – 2024

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Магістр

Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

ЗАТВЕРДЖУЮ

Зав. кафедри: к.т.н. Рубан С.А.

« 5 » липня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

студентові групи АКІТР-23-2м. Білецький Ярослав Андрійович

1. Тема кваліфікаційної роботи: «Розробка системи дистанційного моніторингу
якості повітря з використанням інтернет речей

затверджено наказом по університету № 595с від 04.07.2024 р.

2. Термін здачі кваліфікаційної роботи: 01.12.2024 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка обсягом 81с., додатки,
презентація у Microsoft PowerPoint (24 слайдів) в електронному та друкованому
вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-3

зав. Кафедри Рубан С.А.

Нормоконтроль

доц. Маринич І. А.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>12.07.24</i>
2	<i>Розділ 1</i>	<i>19.07.24</i>
3	<i>Розділ 2</i>	<i>24.08.24</i>
4	<i>Розділ 3</i>	<i>10.09.24</i>
5	<i>Висновки</i>	<i>13.10.24</i>
6	<i>Оформлення кваліфікаційної роботи</i>	<i>17.11.24</i>
7	<i>Підготовка презентації та графічного матеріалу</i>	<i>27.11.24</i>
8	<i>Підготовка доповіді до захисту</i>	<i>01.12.24</i>

6. Дата видачі завдання: 28.06.2024р.

Керівник _____ /Рубан С.А./

7. Запевнення: Я, Білецький Ярослав Андрійович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Здобувач _____ / Білецький Я.А./

АНОТАЦІЯ

Білецький Я.А. «Розробка системи дистанційного моніторингу якості повітря з використанням технологій інтернет речей».

Кваліфікаційна робота на здобуття ступеня вищої освіти магістр, за освітньо-професійною програмою «Кіберфізичні системи в промисловості, бізнесі та транспорті» зі спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка. – Криворізький національний університет, Кривий Ріг, 2024.

Об'єктом дослідження є системи моніторингу якості повітря, та методи обробки даних.

У першому розділі було розглянуто основи вимірювання рівня забруднення, застосування індексації якості повітря (AQI) і вплив основних забруднювачів на довкілля та здоров'я людини. Проаналізовано концентрації ключових забруднюючих речовин, та розглянуто існуючі системи та пристрої моніторингу якості повітря..

У другому розділі було розглянуто технолоію IoT та ідентифікована математича моделі та синтез керування процесом системою дистанційного моніторингу якості повітря з використанням технологій інтернет.

У третьому розділі було обрано компоненти для пристрою та фізично і програмно реалізовано системою дистанційного моніторингу якості повітря, налаштовано через MQTTброкер, Node Red та Телеграм бот.

Ключові слова:

ЯКІСТЬ ПОВІТРЯ, IoT, ESP 32, NODE RED, ТЕЛЕГРАМ БОТ.

ANNOTATION

Biletsky Ya.A. "Development of a remote air quality monitoring system using Internet of Things technologies".

Qualification work for the degree of higher education Master, in the educational and professional program "Cyber-physical systems in industry, business and transport" in the specialty 174 - Automation, computer-integrated technologies and robotics. - Kryvyi Rih National University, Kryvyi Rih, 2024.

The object of the study is air quality monitoring systems and data processing methods.

The first section considered the basics of measuring the level of pollution, the application of air quality indexation (AQI) and the impact of major pollutants on the environment and human health. The concentrations of key pollutants were analyzed, and existing air quality monitoring systems and devices were reviewed.

The second section considered IoT technology and identified mathematical models and synthesis of process control of a remote air quality monitoring system using Internet technologies.

In the third section, the components for the device were selected and physically and software implemented by a remote air quality monitoring system, configured via MQTT broker, Node Red and Telegram bot.

Keywords:

COUNTY QUALITY, IoT, ESP 32, NODE RED, TELEGRAM BOT.

ЗМІСТ

АНОТАЦІЯ	4
ANNOTATION	5
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП	9
РОЗДІЛ 1 МОНІТОРИНГ ТА ОЦІНКА ЯКОСТІ АТМОСФЕРНОГО ПОВІТРЯ. ОСНОВИ ВИМІРЮВАННЯ ТА АНАЛІЗ ЗАБРУДНЕННЯ.....	11
1.1 Індексація якості повітря: визначення та практичне застосування	11
1.2 Організація мікроклімату в приміщеннях	20
1.3 СИСТЕМИ ТА ПРИЛАДИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ В УКРАЇНІ.....	24
1.3.1 ПРОЕКТУ SAVE ESOBOT	24
1.3.2 ПРОЕКТ ESO CITY	25
1.3.3 SAVEECOSENSOR 3.0	25
1.3.4 ПРИЛАД РОС AIRC	27
<i>Висновки до розділу</i>	29
РОЗДІЛ 2 ВИЗНАЧЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ТА ПРОЕКТУВАННЯ АЛГОРИТМІВ КЕРУВАННЯ ПРОЦЕСОМ.....	31
2.1 IoT-технології	31
2.2 Ідентифікація математичної моделі та синтез керування процесом	33
2.2.1 Метод експоненціального згладжування (метод Брауна).....	33
2.2.2 Метод лінійної регресія.....	40
<i>Висновки до розділу</i>	52

РОЗДІЛ 3 ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ.....	54
3.1 Вибір пристроїв та датчиків	54
3.1.1 Датчик якості повітря AGS02MA (TVOC)	54
3.1.2 BMP280	57
3.1.3 Датчик АНТ20	60
3.1.4 OLED SSD1306 128x32 0.91	61
3.1.5 Плата ESP32 DevKit v1 на базі мікроконтролера ESP-WROOM-32.....	65
3.2 Приклад реалізації проекту	65
3.2.1 Опис пристрою	65
3.2.2 Принципова схема.....	70
3.2.3 Робота з платою ESP32:.....	71
3.2.3 Робота з Node Red та Телеграм ботом.	74
<i>Висновки до розділу</i>	77
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	81
ДОДАТОК А.....	87
ДОДАТОК Б	92

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

ГДК – гранично допустима концентрація компоненти в повітрі

ЛОС – леткі органічні сполуки

СО – окис вуглецю (чадний газ)

CO₂ – діоксид вуглецю

NO₂ – Діоксид азоту

SO₂ – Діоксид сірки

VOC – Volatile Organic Compound— Леткі органічні сполуки

O₃ – Озон

PM (Particulate Matter) — Зважені частки, дрібнодисперсні частки (наприклад, PM2.5, PM10).

IoT- технологія інтернет речей

GPS (Global Positioning System) — Глобальна система позиціонування для визначення геолокації пристроїв.

Wi-Fi (Wireless Fidelity) — Технологія бездротового зв'язку для підключення пристроїв до мережі.

MQTT (Message Queuing Telemetry Transport) — Протокол обміну даними для IoT пристроїв

DHT (Digital Humidity and Temperature) — Датчик температури та вологості повітря.

AQMS (Air Quality Monitoring System) — Система моніторингу якості повітря.

ВСТУП

Щороку близько 7 мільйонів людей у світі помирають через вплив забрудненого повітря. Це проблема, яка торкається кожного, незалежно від місця проживання, адже забруднення не знає кордонів. Україна залишається серед країн з найгіршими показниками якості повітря, де щорічно фіксується понад 100 смертей на 100 000 жителів через токсичні викиди. Забруднення повітря спричиняє серйозні наслідки для здоров'я, включаючи захворювання дихальної системи, серцево-судинні хвороби та навіть рак.

Основні джерела забруднення в світі, Україні та зокрема в Кривому Розі пов'язані з промисловістю транспортом, та побутовими потребами.

Кривий Ріг є один із найбільших промислових центрів України, який відомий своєю металургійною галуззю та видобутком залізної руди. Через це місто є одним із лідерів за рівнем забруднення повітря в країні.

ВООЗ зазначає, що забруднене повітря є ключовим фактором ризику для багатьох захворювань. Окрім хронічних хвороб, таких як астма та бронхіт, довготривале перебування в зоні з високим рівнем забруднення збільшує ризик інсульту та інфаркту. Діти особливо вразливі до забрудненого повітря, адже їх дихальна система ще не до кінця сформована, що може спричинити незворотні наслідки.

Українська система моніторингу якості повітря суттєво відстає від європейських стандартів. Багато станцій працюють за застарілими методиками, які не здатні фіксувати нові види забруднювачів. У великих містах, таких як Київ, Харків чи Львів, моніторингових станцій недостатньо для створення об'єктивної картини стану повітря.

Проте сучасні технології, зокрема мікроелектронні датчики, здатні забезпечити більш точний аналіз. Їх використання допоможе не лише відстежувати забруднення, а й прогнозувати його вплив на здоров'я населення.

Актуальність теми зумовлена зростаючою загрозою забруднення повітря, яке чинить негативний вплив на здоров'я людини та навколишнє

середовище. У сучасному суспільстві якість повітря є важливим показником екологічної безпеки, особливо у великих містах і промислових районах. Традиційні системи моніторингу часто не встигають вчасно збирати та аналізувати дані, що ускладнює реагування на зміну умов навколишнього середовища. Використання технології Інтернету речей (IoT) дає змогу створювати інтерактивні автономні системи моніторингу, що забезпечують точний і швидкий аналіз забруднення та сприяють ухваленню ефективних рішень зі зниження впливу на навколишнє середовище.

Мета даного дослідження є розробка та впровадження системи віддаленого моніторингу якості повітря на основі технології IoT, що дасть змогу оперативно збирати, аналізувати та візуалізувати дані про стан повітря і полегшить ухвалення управлінських та екологічних рішень.

Предмет дослідження є принципи побудови та функціонування системи віддаленого моніторингу якості повітря на базі IoT, включно зі структурою мережі датчиків, методами передавання даних та оцінки отриманої інформації.

Об'єкт дослідження є якість повітря в навчальних приміщеннях та в приміщеннях в цілому.

РОЗДІЛ 1

МОНІТОРИНГ ТА ОЦІНКА ЯКОСТІ АТМОСФЕРНОГО ПОВІТРЯ. ОСНОВИ ВИМІРЮВАННЯ ТА АНАЛІЗ ЗАБРУДНЕННЯ

1.1 Індексація якості повітря: визначення та практичне застосування

Індекс якості повітря (AQI) — це стандартизований показник, який дозволяє оцінити рівень забруднення повітря та його потенційний вплив на здоров'я людини. Він використовується для того, щоб зробити інформацію про якість повітря доступною та зрозумілою для широкої аудиторії. AQI враховує концентрації ключових забруднювачів, таких як дрібнодисперсний пил (PM_{2.5} і PM₁₀), діоксид азоту (NO₂), діоксид сірки (SO₂), озон (O₃) та оксид вуглецю (CO). Кожен із цих забруднювачів має встановлені межі безпеки, визначені на основі досліджень впливу на здоров'я людини.

Для обчислення AQI використовуються спеціальні формули, які переводять концентрації забруднювачів у стандартизовану шкалу. Ця шкала поділяється на кілька діапазонів, кожен із яких відповідає певному рівню небезпеки для здоров'я. Наприклад, якщо значення AQI перебуває в зеленій зоні, це означає, що повітря є чистим і ризики для здоров'я мінімальні. Жовта зона свідчить про помірне забруднення, яке може становити загрозу для чутливих груп, таких як діти, літні люди чи особи з хронічними респіраторними захворюваннями. Коли значення AQI переходить у червону зону, це вже означає небезпеку для всіх груп населення, а фіолетова зона сигналізує про надзвичайно небезпечний рівень забруднення.

Окрім класифікації рівнів якості повітря, AQI також супроводжується рекомендаціями для населення. Наприклад, у разі високих значень AQI людям радять уникати фізичної активності на відкритому повітрі, особливо вразливим групам, таким як діти, вагітні жінки чи люди з астмою. Також можуть бути надані рекомендації щодо використання захисних масок або встановлення очищувачів повітря в приміщеннях.

Показник AQI може суттєво змінюватися залежно від часу доби, сезону

чи погодних умов. Наприклад, у великих містах рівень забруднення часто зростає під час годин пік через інтенсивний автомобільний рух. У зимовий період ситуацію ускладнює використання твердого палива для опалення, особливо у приватному секторі. Несприятливі метеорологічні умови, такі як штиль чи температурна інверсія, також сприяють накопиченню забруднювачів у повітрі, оскільки відсутність вітру перешкоджає їх розсіюванню.

Індекс якості повітря виконує важливу роль у підвищенні обізнаності суспільства про стан навколишнього середовища. Завдяки ньому люди можуть планувати свою діяльність так, щоб мінімізувати ризики для здоров'я. Окрім того, AQI є важливим інструментом для державних органів і екологічних організацій, які використовують його дані для моніторингу забруднення, розробки екологічної політики та прийняття рішень щодо зменшення викидів шкідливих речовин в атмосферу.

Дрібнодисперсні частинки (PM_{2.5} та PM₁₀) є одними з найбільш небезпечних забруднювачів атмосферного повітря, які суттєво впливають на здоров'я людини та навколишнє середовище. Ці частинки складаються з твердих або рідких речовин, що можуть тривалий час залишатися в повітрі, переміщуючись на значні відстані. Розмір частинок визначає їхню здатність проникати у різні частини дихальної системи людини. Частинки PM₁₀ мають розміри від 2,5 до 10 мікрометрів, тоді як PM_{2.5} є ще меншими – до 2,5 мікрометра.

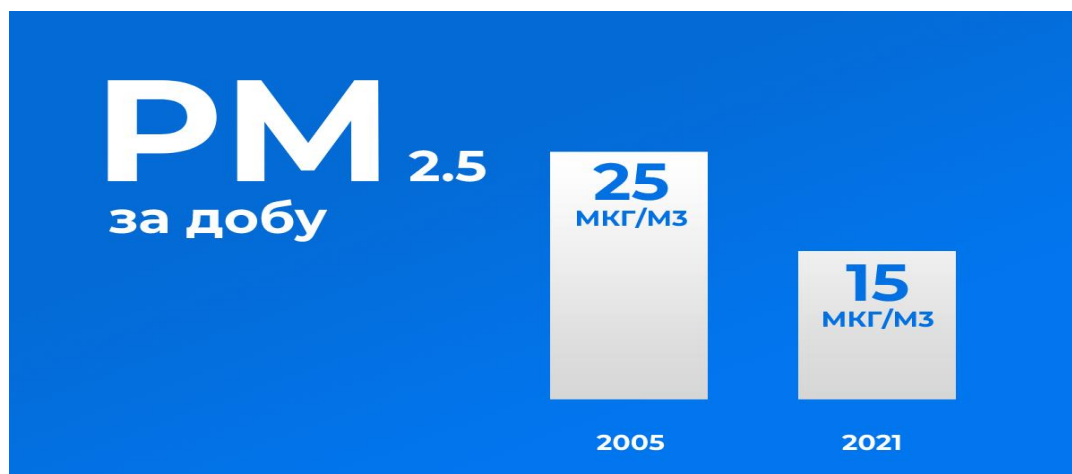


Рисунок 1.1 – Статистичні дані концентрації частинок PM_{2.5}

Через свій невеликий розмір частинки PM10 здатні проникати у верхні відділи дихальних шляхів і легень, викликаючи їхнє подразнення та блокування. Натомість PM2.5 можуть досягати глибоких альвеол легень, проходити через аерогематичний бар'єр і потрапляти в кровоносну систему. Це робить PM2.5 надзвичайно небезпечними, оскільки вони впливають не лише на легені, а й на серцево-судинну систему, спричиняючи запальні процеси та посилюючи ризик хронічних захворювань.

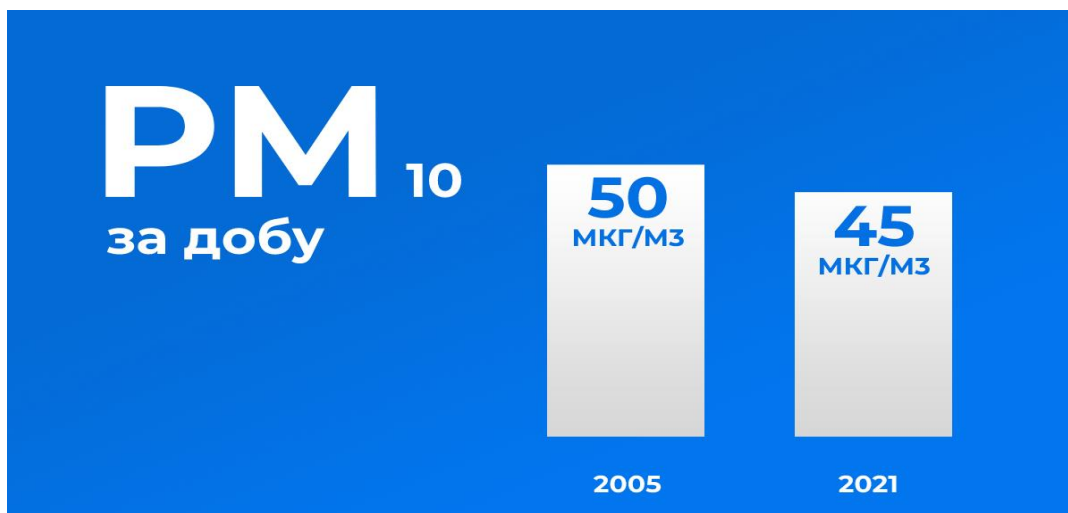


Рисунок 1.2 – Статистичні данні концентрації частинок PM10

Всесвітня організація охорони здоров'я (ВООЗ) вважає доведеним зв'язок між впливом дрібнодисперсних частинок і такими захворюваннями, як хронічна обструктивна хвороба легень, серцево-судинні захворювання, рак легень, а також інфекції нижніх дихальних шляхів. Дослідження також підтверджують, що навіть короточасний вплив високих концентрацій PM2.5 може підвищити ризик госпіталізацій через серцеві напади, інсульти та загострення астми.

Джерела походження цих частинок різняться. PM10 найчастіше утворюються внаслідок будівельних робіт, підняття дорожнього пилу та природних процесів, таких як ерозія ґрунту чи лісові пожежі. PM2.5 в основному є продуктом спалювання, зокрема під час роботи транспортних засобів, опалення будівель, спалювання відходів та промислових процесів. Концентрації цих частинок можуть значно збільшуватися у періоди стабільної

погоди, коли повітря застоюється, особливо в промислово розвинених регіонах. Наприклад, найбільші концентрації PM2.5 та PM10 в Україні спостерігаються у промислових зонах, таких як Донецька, Дніпропетровська та Запорізька області.

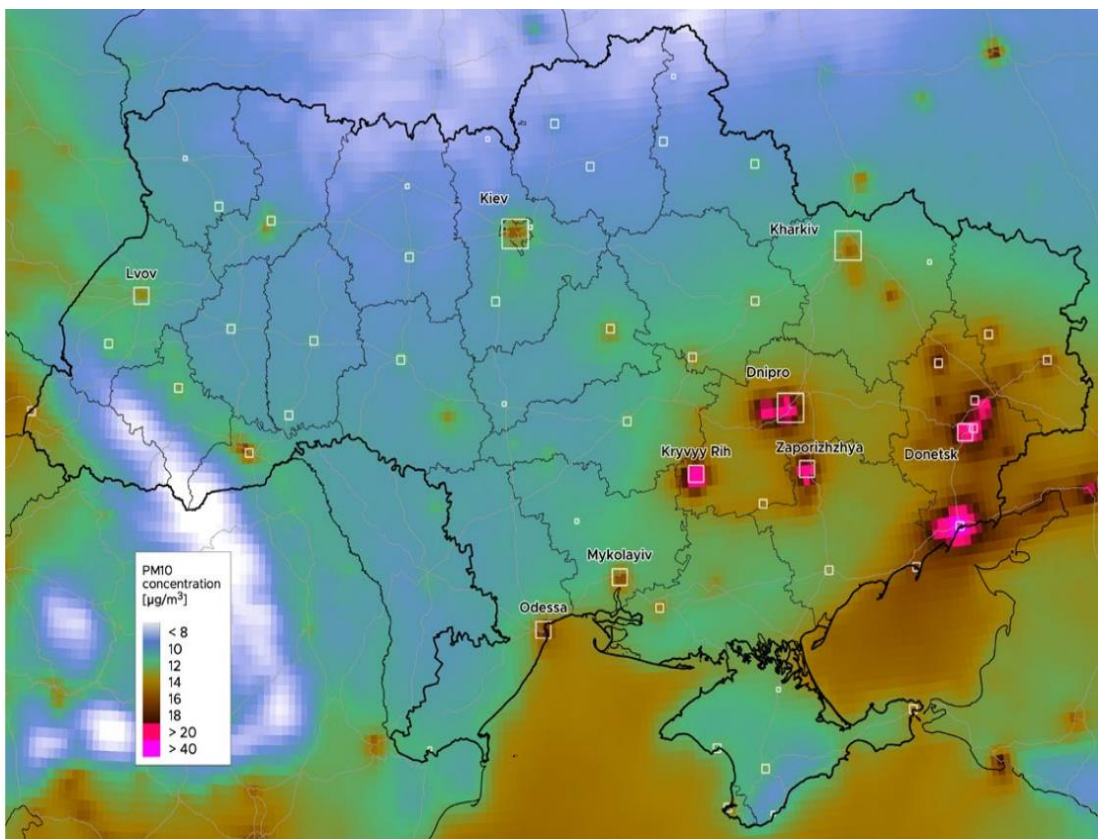


Рисунок 1.4 – Середні концентрації PM10 на території України у період з липня 2017 року по липень 2020 року.

Озон

Приземний озон (O_3), який формується у нижніх шарах атмосфери, є одним із ключових компонентів фотохімічного смогу. Його виникнення обумовлене хімічними реакціями між оксидами азоту (NO_2) і леткими органічними сполуками (ЛОС) під впливом сонячного світла. Таким чином, найвищі концентрації озону спостерігаються у теплі сонячні дні, коли погодні умови сприяють його утворенню.

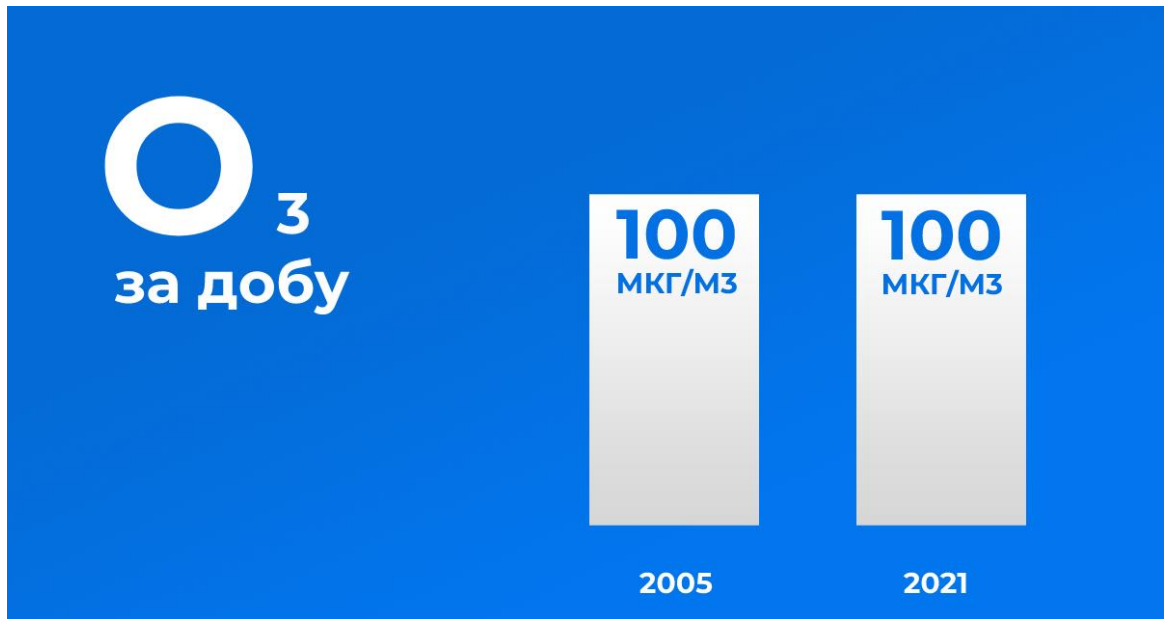


Рисунок 1.5 – Рекомендовані граничні концентрації O_3

Приземний озон є потужним подразником дихальних шляхів. Він може викликати порушення дихання, зниження функції легень і провокувати розвиток або загострення астми. Люди, які вже мають респіраторні чи серцево-судинні захворювання, особливо чутливі до підвищених рівнів озону. За даними ВООЗ, тривале перебування в умовах високих концентрацій озону підвищує ризик захворювань дихальної системи, включаючи хронічні обструктивні захворювання легень і бронхіт.

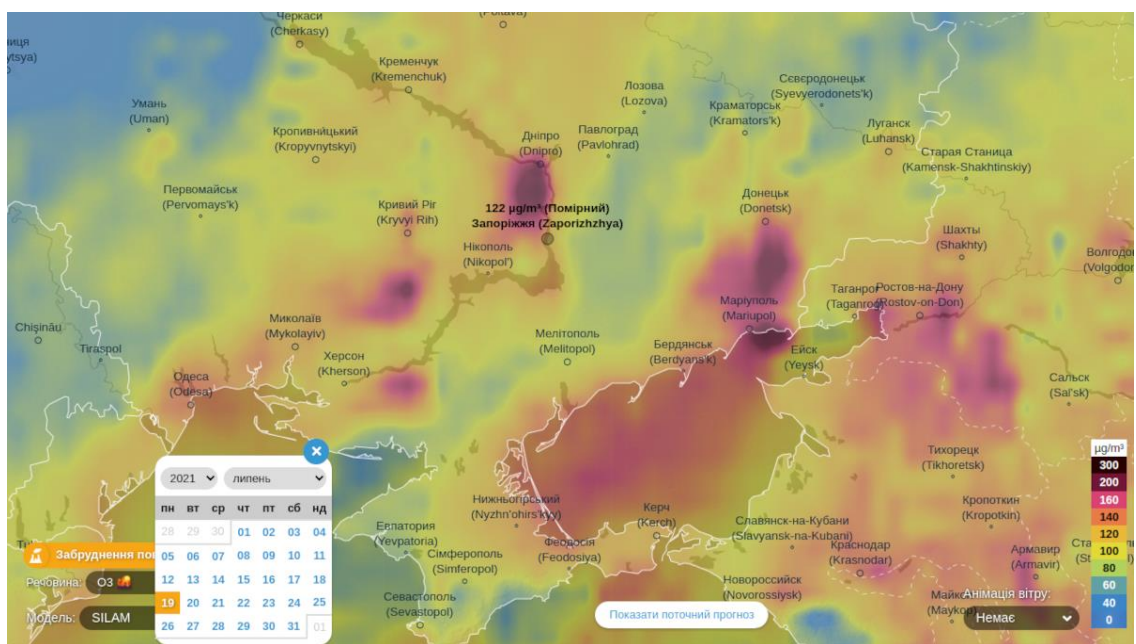


Рисунок 1.6 – Мапа максимальної концентрація озону

Джерелами забруднення, що сприяють утворенню озону, є транспорт, промислові викиди, спалювання біомаси та використання розчинників. В Україні найбільш забруднені озоном території розташовані у південно-східних областях, де промислові процеси та густота транспорту створюють умови для формування смогу.

Діоксид азоту (NO₂)

Є потужним забруднювачем повітря, що виникає внаслідок процесів спалювання. Найбільший внесок у концентрацію NO₂ роблять вихлопні гази автомобілів, особливо у міських районах з інтенсивним трафіком. За даними Європейського агентства з навколишнього середовища, понад 60% NO₂ в містах Європи припадає саме на транспорт.

Цей газ є серйозною загрозою для здоров'я людини, особливо для дітей і людей із хронічними респіраторними захворюваннями. При вдиханні до 90% NO₂ потрапляє в організм і проникає у кров. Він здатний викликати запалення дихальних шляхів, підвищуючи вразливість до бактеріальних і вірусних інфекцій. Дослідження ВООЗ підтверджують, що високі концентрації NO₂ сприяють розвитку бронхіту, астми, а також погіршенню функції та росту легень у дітей.

Найвищі рівні забруднення NO₂ в Україні фіксуються у великих містах, таких як Київ, Дніпро, Харків та у промислових регіонах.



Рисунок 1.7 – Рекомендовані граничні концентрації NO₂

Діоксид сірки (SO₂)

Це безбарвний газ із характерним запахом, який утворюється при спалюванні викопного палива, такого як вугілля та нафта. Основними джерелами SO₂ є вугільні електростанції, металургійна промисловість і спалювання біомаси.

Вплив SO₂ на здоров'я людини проявляється у вигляді подразнення слизових оболонок очей, дихальних шляхів і легенів. У дні з високими концентраціями SO₂ зростає кількість госпіталізацій через серцево-судинні та респіраторні захворювання. Довготривалий вплив навіть низьких концентрацій цього газу може сприяти розвитку хронічного бронхіту та астми.

Максимальні концентрації SO₂ в Україні зазвичай спостерігаються в зимовий період через збільшення викидів від опалення.



Рисунок 1.9 – Рекомендовані граничні концентрації SO₂

Чадний газ (CO)

Чадний газ, або оксид вуглецю (CO), є одним із найбільш небезпечних забруднювачів повітря, який характеризується безбарвністю, відсутністю

запаху і смаку. Це робить його особливо підступним, оскільки виявити його присутність у повітрі без спеціального обладнання неможливо. Основним джерелом утворення CO є процеси неповного згоряння палива, коли кількість кисню є недостатньою для повного окиснення вуглецю до діоксиду вуглецю (CO₂).

Основними джерелами чадного газу в атмосферу є:

1. Транспорт – особливо автомобілі з двигунами внутрішнього згоряння, які працюють на бензині або дизелі. Недостатнє спалювання палива в двигунах сприяє значним викидам CO. У великих містах з інтенсивним дорожнім рухом це джерело є основним.

2. Побутове використання палива – газові плити, печі, обігрівачі, каміни і навіть генератори, які використовуються в умовах недостатньої вентиляції, можуть виділяти чадний газ у небезпечних концентраціях.

3. Промисловість – металургійні підприємства, цементні заводи, виробництво хімікатів і спалювання відходів часто є джерелами CO.

4. Природні процеси – лісові пожежі, виверження вулканів і навіть розкладання органічних речовин також виділяють CO в атмосферу.

Чадний газ є надзвичайно токсичним для живих організмів через його здатність зв'язуватися з гемоглобіном у крові. Коли людина вдихає CO, цей газ швидко вступає в хімічну реакцію з гемоглобіном, утворюючи карбоксигемоглобін (HbCO). Ця сполука є стабільною і блокує здатність гемоглобіну переносити кисень до тканин і органів, що призводить до гіпоксії – кисневого голодування.

Вплив CO залежить від його концентрації та тривалості експозиції:

Низькі концентрації (до 50 ppm) можуть викликати головний біль, нудоту, запаморочення і загальну слабкість.

Середні концентрації (50-200 ppm) провокують серйозніші симптоми, такі як задишка, біль у грудях, порушення концентрації уваги і координації.

Високі концентрації (понад 200 ppm) можуть призвести до втрати свідомості, серцевих нападів і навіть смерті протягом короткого часу.

Особливо вразливими до впливу СО є люди з хронічними захворюваннями серця і легень, вагітні жінки, діти та літні люди. У вагітних чадний газ може спричинити гіпоксію у плода, що підвищує ризик ускладнень під час вагітності.

Чадний газ навіть у низьких концентраціях може негативно впливати на когнітивні функції, спричиняти постійні головні болі та знижувати працездатність. У людей, які перебувають під впливом СО протягом тривалого часу, зростає ризик розвитку серцево-судинних захворювань.



Рисунок 1.10 – Рекомендовані граничні концентрації СО

СО також має непрямий вплив на клімат, хоча сам газ не є парниковим, він бере участь у хімічних реакціях у атмосфері, що збільшують концентрації інших парникових газів, таких як метан (CH_4) і тропосферний озон (O_3). Це відбувається через його здатність взаємодіяти з гідроксиллом ($\text{OH}\cdot$), який відповідає за очищення атмосфери.

Системи моніторингу якості повітря дозволяють оперативно виявляти підвищені рівні СО, особливо в міських районах із щільним транспортним потоком або в промислових зонах. У побутових умовах важливо

використовувати детектори чадного газу, особливо в приміщеннях із газовими приладами або камінами.

Для зменшення викидів СО важливими заходами є впровадження екологічно чистих технологій, перехід на електромобілі, використання відновлюваних джерел енергії та дотримання правил безпеки під час використання палива.

Чадний газ залишається серйозною загрозою для людства, і боротьба з його негативним впливом вимагає комплексного підходу на всіх рівнях – від державного регулювання до індивідуальної обережності.

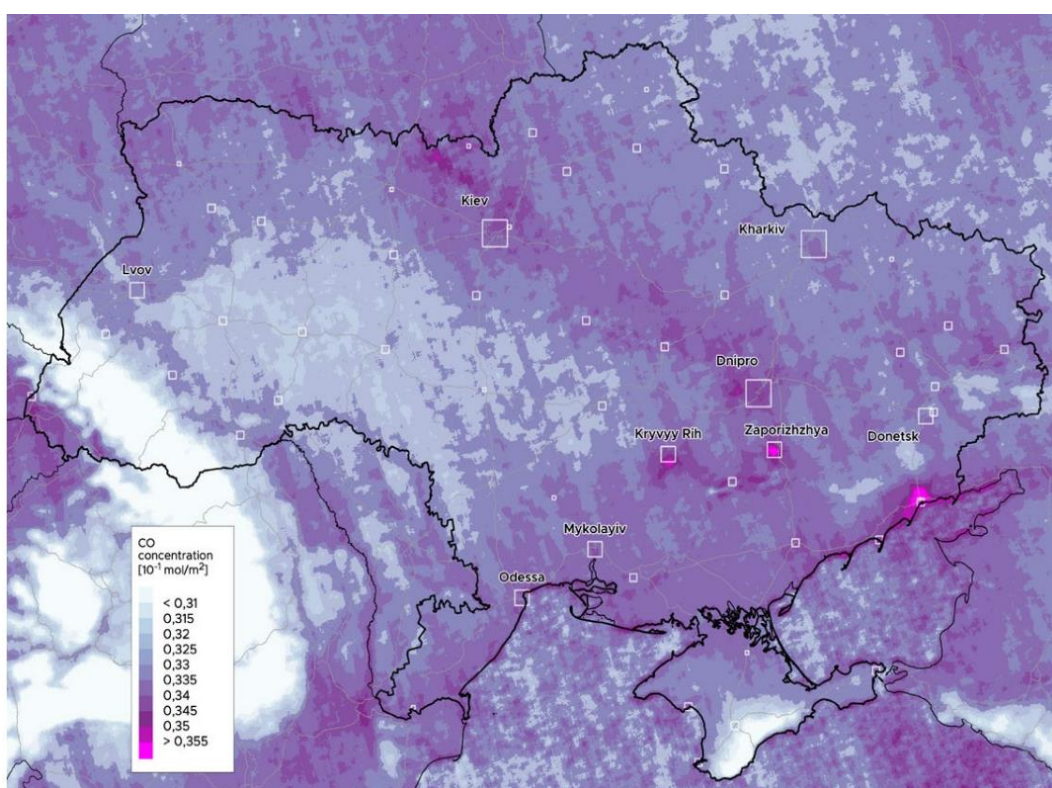


Рисунок 1.12 – Середні концентрації СО на території України у період з травня 2018 року до квітня 2020 року

1.2 Організація мікроклімату в приміщеннях

Повітря в навчальних аудиторіях

Повітря в навчальних аудиторіях має дуже важливе значення для здоров'я та ефективності навчання. За дослідженнями Всесвітньої організації

охорони здоров'я, якість повітря в приміщеннях може бути значно гіршою, ніж на вулиці. Це пов'язано з тим, що всередині приміщень накопичуються шкідливі речовини, а провітрювання відбувається недостатньо часто. Особливо це стосується навчальних закладів, де наявність великої кількості людей у невеликому просторі тільки погіршує ситуацію.

Однією з причин забруднення є використання недорогих будівельних матеріалів і меблів, наприклад, ДСП. Цей матеріал часто виділяє формальдегід — небезпечну хімічну сполуку, яка викликає алергії та інші захворювання. Інші джерела забруднення — це фарби, клеї, підлогові покриття, які виділяють леткі органічні сполуки. З часом вони накопичуються в повітрі, особливо якщо приміщення не провітрюється.

Організація мікроклімату в приміщеннях: ключові аспекти

Для забезпечення безпеки працівників у приміщеннях визначаються гранично допустимі концентрації (ГДК) шкідливих речовин у повітрі. Ці нормативи встановлені для того, щоб уникнути негативного впливу токсичних речовин на здоров'я людей. Усі шкідливі речовини поділяються на чотири класи залежно від рівня небезпеки:

Надзвичайно небезпечні речовини (1-й клас) мають ГДК менше 0,1 мг/м кубічних . До них належать свинець, ртуть, озон тощо.

Високонебезпечні речовини (2-й клас) характеризуються ГДК у межах 0,1–1,0 мг/м кубічних. Це, наприклад, сірчана й соляна кислоти, хлор, фенол, луги.

Помірно небезпечні речовини (3-й клас) мають ГДК від 1,1 до 10,0 мг/м кубічних. Сюди входять толуол, ксилол, метанол та інші хімікати.

Малонебезпечні речовини (4-й клас) із ГДК понад 10 мг/м кубічних, як-от аміак, бензин, ацетон.

Методи контролю шкідливих речовин

Щоб оцінити рівень забруднення повітря, використовуються такі методи:

Експрес-методи. Ці способи базуються на зміні кольору індикаторного

порошку під впливом шкідливих речовин. Вони дозволяють швидко отримати результати безпосередньо в приміщенні. Наприклад, використовуються прилади УГ-2, ГХ-4, СТЧ-17.

Лабораторні методи. У цьому випадку проби повітря беруться для подальшого аналізу у спеціалізованих лабораторіях. Методи, як-от хроматографія чи фотохромометрія, забезпечують високоточні результати, проте потребують більше часу.

Безперервний моніторинг. Для цього застосовуються автоматичні газоаналізатори, які постійно контролюють рівень шкідливих речовин.

Оцінка запиленості повітря

Для визначення концентрації пилу в повітрі використовують кілька методів:

Гравіметричний метод: спеціальні фільтри зважують до та після пропускання повітря, що дозволяє визначити кількість пилу в міліграмах на кубічний метр.

Фотометричні та електроіндукційні методи: ці способи дозволяють оцінювати концентрацію мікрочастинок у повітрі.

На сучасному ринку доступні портативні аналізатори, які визначають рівень частинок розміром від 0,3 мкм до 10 мкм.

Частота перевірки залежить від ступеня небезпеки речовин і умов роботи. Моніторинг може здійснюватися: Постійно, якщо є ризик високих концентрацій небезпечних речовин або Періодично, щодня, щотижня чи щомісяця залежно від специфіки діяльності підприємства.

Для підвищення безпеки у випадках можливого раптового підвищення концентрації токсичних речовин обов'язково встановлюють системи сигналізації. Вони попереджають про перевищення допустимих значень.

Таким чином, правильна організація мікроклімату в приміщенні є ключовим фактором для збереження здоров'я працівників і забезпечення безпечних умов праці.

Ще одна проблема — вуглекислий газ (CO_2), який видихає кожна

людина. У приміщеннях із недостатньою вентиляцією рівень CO₂ дуже швидко зростає. Якщо цей показник перевищує 1000 ppm, з'являється сонливість, важкість у голові, знижується увага. У навчальних аудиторіях це особливо помітно під час пар, коли в одному класі перебуває багато студентів. Чим довше приміщення не провітрюється, тим більше накопичується CO₂, що погіршує самопочуття і продуктивність.

Також слід звернути увагу на пил, який осідає в приміщеннях. Особливо небезпечні мікрочастинки розміром до 10 мкм, які легко потрапляють у легені. У навчальних закладах це може бути пил із підлоги, меблів, чи навіть залишки крейди. Якщо прибирання проводиться нерегулярно або лише сухе, це тільки посилює проблему.

Для вимірювання рівня шкідливих речовин у повітрі існують різні методи. Наприклад, використовуються портативні газоаналізатори, які можуть швидко показати рівень CO₂, формальдегіду або пилу. Більш точні результати можна отримати в лабораторіях, але це займає більше часу. У сучасних аудиторіях також встановлюють датчики, які постійно вимірюють рівень забруднень і сигналізують, якщо показники перевищують норму.

Як оцінити якість повітря

Експрес-методи використання портативних газоаналізаторів (наприклад, для CO₂, формальдегіду). Це швидкий спосіб оцінки рівня токсичних речовин.

Лабораторний аналіз забір проб повітря з подальшим дослідженням у лабораторії (хроматографія, фотометрія). Метод точний, але потребує часу.

Постійний моніторинг сучасні автоматизовані системи забезпечують безперервний контроль рівнів CO₂, пилу, формальдегіду та інших забруднювачів.

Оцінка запиленості для визначення концентрації пилу використовують фільтри або фотометричні прилади.

Щоб покращити якість повітря, потрібно дотримуватися кількох простих правил. Перше — це регулярне провітрювання. Якщо немає

автоматичної системи вентиляції, достатньо кілька разів на день відкривати вікна, щоб забезпечити притік свіжого повітря. Друге — використовувати екологічно чисті меблі та оздоблювальні матеріали. Також варто уникати дешевих матеріалів, які виділяють шкідливі речовини. Третє — забезпечити регулярне прибирання із застосуванням вологих методів, щоб зменшити кількість пилу. Крім того, варто розглянути можливість встановлення очищувачів повітря моніторингових систем датчики для відстеження рівнів CO₂ і токсичних речовин дозволяють вчасно реагувати на проблеми.

Ще один цікавий спосіб очищення повітря — це використання рослин. Наприклад, хлорофітум і сансев'єрія відомі своєю здатністю поглинати CO₂ і токсини. Такі рослини не тільки прикрашають аудиторію, а й покращують якість повітря, створюючи комфортні умови для навчання.

Отже, якість повітря у навчальних аудиторіях — це проблема, на яку варто звертати увагу. Вона впливає на здоров'я студентів і викладачів, а також на ефективність навчального процесу. Якщо вчасно впровадити прості заходи, можна значно покращити атмосферу в приміщеннях і забезпечити комфортні умови для всіх.

1.3 Системи та прилади моніторингу якості повітря в Україні

1.3.1 SaveEcoBot – це унікальна в Україні екологічна система, яка об'єднує дані про стан довкілля, рівень забруднення, джерела забруднення та інструменти для його захисту. Вона забезпечує інформацію про:

Щогодинні концентрації дрібнодисперсного пилу фракції 2.5 мікронів (PM_{2.5}) та індекс якості атмосферного повітря (AQI) відповідно до формули NowCast, розробленої Агентством із захисту довкілля США (US EPA), для населених пунктів України.

Первинні показники якості повітря, які отримуються зі станцій моніторингу, встановлених жителями населених пунктів, незалежними проєктами, організаціями та органами місцевого самоврядування.

Система автоматично збирає й агрегує ці дані, щогодини розраховує середнє арифметичне значення концентрації PM2.5 для кожної станції моніторингу та обчислює AQI PM2.5 за методикою NowCast. На основі цих даних обраховується середнє значення концентрації PM2.5 та AQI PM2.5 для кожного населеного пункту.

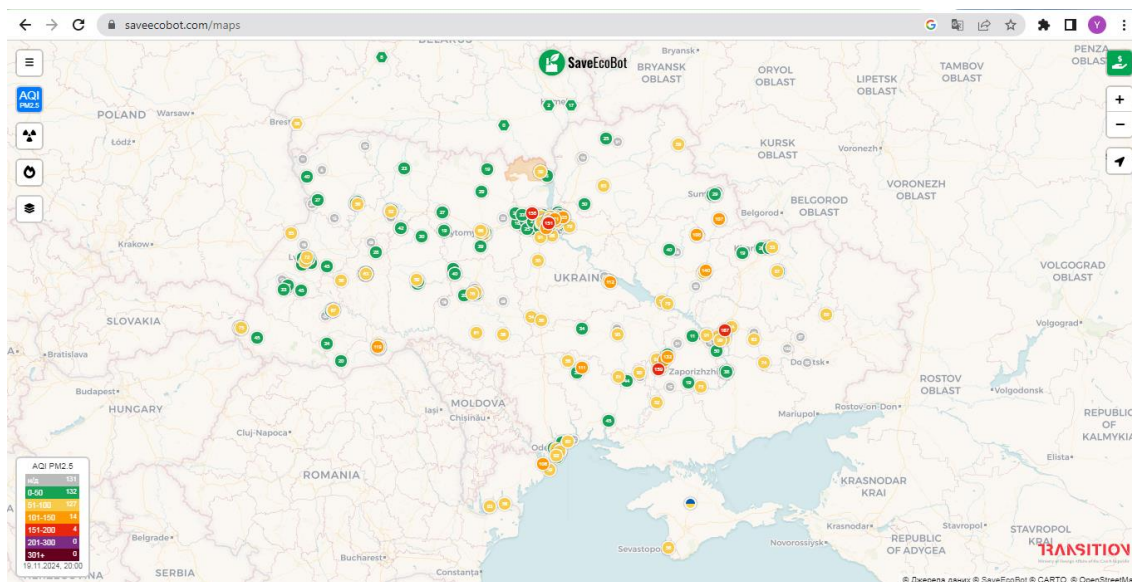


Рисунок 1.13 – Еко-мапа, проекту SaveEcoBot

1.3.2 Проект ECO CITY

Надає данні в відкритому доступі зі стаціонарних станцій, у великих містах України

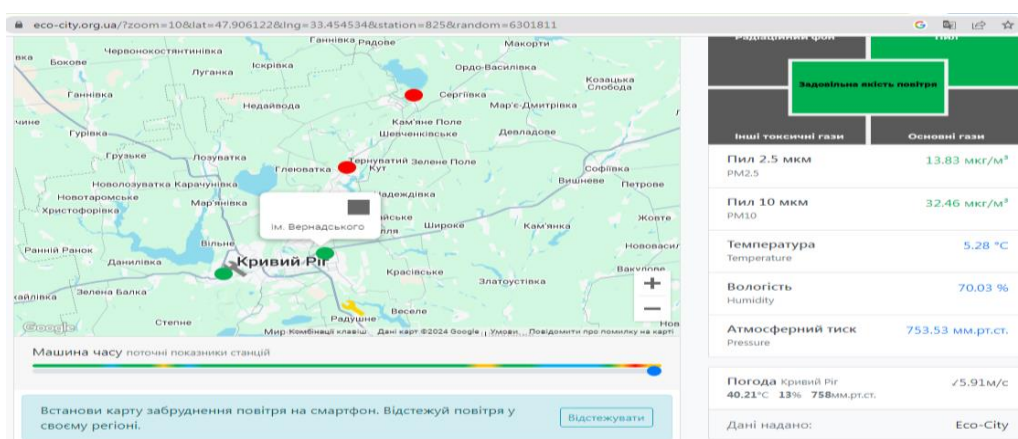


Рисунок 1.14 – Еко-мапа, проекту ECO CITY

1.3.3 SaveEcoSensor 3.0 – це сучасна станція моніторингу якості повітря, яка створена для високоточного вимірювання рівня забруднення атмосфери.

Пристрій забезпечує контроль за вмістом пилу фракцій PM2.5 та PM10 у повітрі – найбільш небезпечних для здоров'я форм забруднення, які утворюються внаслідок процесів горіння. Завдяки своїй продуманій конструкції та технічним рішенням, SaveEcoSensor 3.0 гарантує надійність та точність вимірювань навіть у несприятливих погодних умовах.

Особливості вимірювань:
SaveEcoSensor 3.0 оснащений сенсорами, які фіксують вміст дрібнодисперсного пилу PM2.5 і PM10, температури, вологості та атмосферного тиску. Завдяки інтегрованому сенсору температури, вологості й тиску (BME280), пристрій здатен автоматично коригувати результати вимірювань, враховуючи погодні фактори. Вбудована камера підігріву знімає вплив туману, опадів та низьких температур, що дозволяє отримувати точні дані навіть за складних умов.

Вплив дрібнодисперсного пилу:
PM2.5 і PM10 – це частинки, які мають настільки малий розмір, що здатні проникати глибоко в дихальні шляхи, потрапляти до кровотоку та навіть у мозок. Вони викликають серйозні проблеми зі здоров'ям, такі як серцеві захворювання, хронічні хвороби дихальної системи, загострення астми та навіть передчасну смерть. Контроль за рівнем цих забруднень є ключовим для збереження здоров'я населення.

Технічні переваги:
Пристрій використовує пиломір SDS011 і контролер Wemos D1 mini V2 Pro, які забезпечують високоточне вимірювання та стабільну роботу. Камера підігріву з можливістю моніторингу її внутрішніх параметрів (температура та відносна вологість) підтримує коректну роботу сенсора за екстремальних умов. Вбудований перетворювач живлення та апаратний сторожовий таймер (Watchdog) підвищують стійкість пристрою до збоїв.

SaveEcoSensor 3.0 має можливість розширення функціоналу за допомогою підключення додаткового сенсора для вимірювання концентрації двоокису вуглецю (CO2) – Senseair S8.

При виробництві пристрою застосовується припій без важких металів, що відповідає вимогам RoHS. Маркування корпусу здійснюється за допомогою ультрафіолетового друку, що зменшує використання полімерів та виключає застосування ПВХ-стрічки, яка негативно впливає на довкілля.

SaveEcoSensor 3.0 – це не просто інструмент моніторингу якості повітря, а приклад екологічно відповідального підходу до виробництва та використання технологій для захисту довкілля і здоров'я людей.



Рисунок 1.15 – Станція моніторингу якості повітря SaveEcoSensor 3.0

1.3.4 Прилад PoC AirC

Це потужний прилад для моніторингу якості повітря, що забезпечує точні вимірювання основних показників забруднення, таких як CO (чадний газ), NO₂ (діоксид азоту), SO₂ (діоксид сірки), O₃ (озон), а також дрібнодисперсного пилу PM_{2.5} та PM₁₀, серед інших. Цей прилад є частиною екологічних проектів, що мають на меті покращити стан атмосферного повітря через точний контроль і збору даних про забруднення.

Технічні характеристики PoC AirC:

1. SPEC digital sensors — прилад оснащений цифровими сенсорами для вимірювання основних забруднювачів, таких як чадний газ (CO), діоксид азоту (NO₂), діоксид сірки (SO₂) і озон (O₃). Ці сенсори дозволяють

отримувати точні та стабільні показники цих шкідливих компонентів в атмосфері, що є важливим для контролю якості повітря.

2. Датчик пилу PM2.5 та PM10 (NOVA SDS011) — один з основних компонентів приладу, здатний точно вимірювати концентрацію дрібнодисперсного пилу в повітрі. Частинки PM2.5 і PM10 є одними з найбільш небезпечних для здоров'я, оскільки можуть проникати в легені, кров і навіть до мозку, викликаючи серйозні захворювання, включаючи хронічні обструктивні хвороби легень і серцеві недуги.

3. CCS811 Air Quality Sensor — датчик якості повітря, який вимірює рівень CO₂ та загальні леткі органічні сполуки (TVOCs). Це допомагає визначити загальний рівень забруднення повітря та є важливим для моніторингу стану приміщень і навколишнього середовища.

4. Комплексний сенсор HCHO (CH₃, C₆H₆, H₂CO, DFRobot Gravity HCHO Sensor) — цей сенсор здатний визначати рівень формальдегіду (HCHO) та інших органічних летких сполук, таких як бензол (C₆H₆). Важливість цього сенсора полягає в тому, що формальдегід та бензол є канцерогенними речовинами, які можуть значно погіршити здоров'я людини, особливо при тривалому впливі.

5. Модуль погодних умов BME280 — сенсор, що вимірює температуру, вологість та атмосферний тиск. Ці показники важливі для коригування та точного аналізу даних про якість повітря, оскільки вони можуть впливати на рівень забруднення та його поширення в атмосфері.

6. GlobalLogic Starter Kit на основі STM32F104 DISCOVERY, FreeRTOS — це основа апаратного забезпечення приладу, що включає мікроконтролер STM32F104, який забезпечує високу обчислювальну потужність і здатний ефективно обробляти дані з різних сенсорів. Використання операційної системи реального часу (RTOS) дозволяє приладу здійснювати моніторинг в реальному часі без збоїв.

7. OrangePi Zero з Armbian Linux — це апаратне рішення для управління та обробки даних, яке забезпечує гнучкість у роботі з приладом,

зокрема, дозволяє обробляти та передавати зібрані дані для подальшого аналізу або відображення в реальному часі. Armbian Linux надає стабільну та ефективну платформу для виконання задач моніторингу та збору даних.

Використання PoC AirC в екологічних проектах: PoC AirC дозволяє організаціям, підприємствам та громадянам моніторити якість повітря в реальному часі, що сприяє покращенню екологічної ситуації, зниженню рівня забруднення та підвищенню обізнаності про важливість чистого повітря для здоров'я. Цей прилад також може використовуватися в рамках різних програм моніторингу навколишнього середовища, урядових ініціатив та проектів по захисту здоров'я населення.



Рисунок 1.16 PoC AirC

Висновки до розділу

Моніторинг і оцінка якості атмосферного повітря є важливими елементами забезпечення екологічної безпеки та здоров'я населення. У розділі розглянуто основи вимірювання рівня забруднення, застосування індексації якості повітря (AQI) і вплив основних забруднювачів на довкілля та здоров'я

людини.

Проаналізовано концентрації ключових забруднюючих речовин, таких як тверді частинки (PM_{2.5} та PM₁₀), озон (O₃), діоксид азоту (NO₂), діоксид сірки (SO₂) та чадний газ (CO). Встановлено, що підвищення концентрацій цих речовин прямо впливає на розвиток серцево-судинних та респіраторних захворювань, а також на погіршення екосистем. Найбільші рівні забруднення спостерігаються у промислових регіонах і великих містах України.

Результати свідчать про необхідність посилення заходів для зниження викидів, удосконалення системи моніторингу та підвищення поінформованості населення. Особливу увагу слід приділяти впровадженню сучасних технологій зменшення викидів у промисловості, транспорті та енергетиці, а також розробці національних програм адаптації до змін клімату, які сприяли б зменшенню впливу атмосферного забруднення.

Таким чином, системний підхід до контролю якості повітря є важливою умовою для збереження здоров'я громадян, сталого розвитку країни та підтримки природного балансу екосистем.

РОЗДІЛ 2

ВИЗНАЧЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ТА ПРОЕКТУВАННЯ АЛГОРИТМІВ КЕРУВАННЯ ПРОЦЕСОМ

2.1 IoT-технології

Технологія інтернет речей(IoT) – це технологія, яка змінює наше розуміння взаємодії між пристроями. Це концепція, де будь-який пристрій, що має можливість підключення до Інтернету, може обмінюватися даними з іншими пристроями. IoT охоплює розумні лампи, димові сигналізації, термостати та безліч інших речей, які можна підключити до мережі. Такі пристрої оснащені датчиками, електронікою та програмним забезпеченням, що дозволяє їм працювати автономно або разом із іншими пристроями.

Свої витoki IoT бере ще в 1982 році, коли в Університеті Карнегі-Меллона встановили автомат Coca-Cola, який передавав інформацію про кількість напоїв і стан пристрою. У 1990-х роках концепція розвивалася завдяки спробам інтеграції пристроїв через мережі. Хоча тодішні платформи, як-от Microsoft at Work або NEST від Novell, не були успішними, вони стали основою для сучасного IoT.

У сучасному вигляді IoT працює через збирання даних із датчиків, їх обробку та передачу результатів користувачеві. Ці процеси організовані через чотири основні рівні. Сенсорний рівень фіксує зміни в середовищі, наприклад, рух, температуру чи освітлення. Зібрані дані передаються через мережевий рівень за допомогою Wi-Fi, Bluetooth чи інших технологій до рівня обробки даних. На цьому етапі аналізуються отримані показники, визначаються закономірності, приймаються рішення, які потім передаються на рівень прикладних додатків. Останній рівень відповідає за взаємодію з користувачем, представляючи дані через зручний інтерфейс.

Хмарні технології є невід'ємною частиною IoT. Вони дозволяють зберігати великі обсяги даних, забезпечувати швидкість обробки та доступність з будь-якої точки світу. Наприклад, у розумних будинках хмарні

платформи координують роботу пристроїв, мінімізуючи потребу у локальній інфраструктурі. Крім того, хмара допомагає захистити дані користувачів через сучасні протоколи безпеки, такі як шифрування та багаторівнева автентифікація.

The Internet of Things Ecosystem

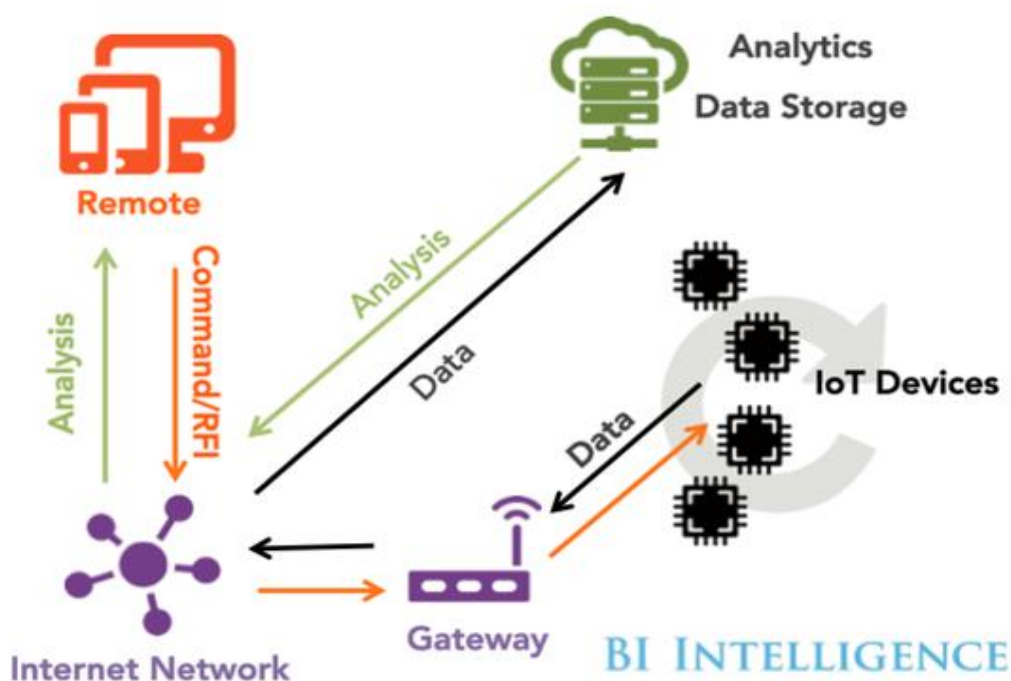


Рисунок 2.1 – Взаємодія ІОТ з іншими пристроями

Сфери використання ІоТ дуже широкі. У розумних будинках ці технології застосовуються для автоматизації побутових пристроїв, таких як термостати, освітлення чи кормушки для тварин. У медицині ІоТ допомагає лікарям дистанційно контролювати стан пацієнтів, доставляти ліки дронами та аналізувати дані про здоров'я. В автомобільній промисловості ІоТ використовується для розробки безпілотних машин, систем навігації та дистанційного керування функціями автомобіля. У розумних містах технології спрямовані на покращення інфраструктури, наприклад, створення систем розумного паркування, моніторингу якості повітря чи управління міським транспортом.

Попри безліч переваг, ІоТ має певні недоліки. Наприклад, відсутність

єдиного стандарту для пристроїв різних виробників створює проблеми сумісності. Велика кількість даних, що зберігаються в IoT-системах, може стати мішенню для хакерів, а також викликає питання конфіденційності. Проте розвиток технологій дозволяє поступово вирішувати ці виклики, роблячи IoT більш безпечним і доступним.

Отже, IoT є важливою складовою сучасних технологій, яка знаходить застосування в багатьох сферах нашого життя. Завдяки своїй здатності автоматизувати процеси, зберігати ресурси та покращувати взаємодію між пристроями, IoT стає невід'ємною частиною цифрового майбутнього.

2.2 Ідентифікація математичної моделі та синтез керування процесом

2.2.1 Метод експоненціального згладжування (метод Брауна)

Метод експоненційного згладжування, запропонований Р.Г. Брауном, базується на вирівнюванні часового ряду за допомогою зваженої ковзної середньої, ваги якої підкоряються експоненціальному закону. Така зважена ковзна середня відображає стан процесу в кінці аналізованого періоду та служить середньою характеристикою рівнів ряду. Ця властивість активно використовується для прогнозування динаміки. З огляду на те, що соціально-економічні процеси проявляють інерцію, тобто в прогнозованому періоді продовжують розвиватися аналогічно до досліджуваного, застосування зваженої ковзної середньої для прогнозування є обґрунтованим.

Припустимо, задано часовий ряд $y_t = (t = 1, 2, \dots, n)$, який описується многочленом p -го порядку:

$$y_t = a_0 + a_1 t + \frac{a_2}{2!} t^2 + \dots + \frac{a_p}{p!} t^p + u_i, \quad (2.1)$$

де $a_i = (i = 1, 2, \dots, p)$ — коефіцієнти ряду; u_i — випадкова складова.

Необхідно, використовуючи відомі рівні ряду, оцінити значення на

момент $t+1$ ($l = 1, 2, \dots, L$), таким чином, щоб наступним спостереженням відповідали більші ваги, ніж попереднім. Прогноз для рівнів ряду на момент часу $t+1$ ($t=n$) отримують, використовуючи розклад ряду в ряд Тейлора:

$$\hat{y}_{t+1} = y_t^{(0)} + l y_t^{(1)} + \frac{l^2}{2!} y_t^{(2)} + \dots + \frac{l^i}{i!} y_t^{(i)} + \dots + \frac{l^p}{p!} y_t^{(p)}, \quad (2.2)$$

Де $y_t^{(i)}$ - i -та похідна на момент часу t .

Згідно з теоремою Брауна - Маєра довільна i -та похідна ($i= 0, 1, 2, \dots, p$) до $(p + i)$ -го порядку.

За теоремою Брауна-Маєра, будь-яку i -ту похідну ($i=0,1,2,\dots,p$) можна виразити через лінійні комбінації експоненційних середніх до $(p + i)$ -го порядку. Таким чином, експоненційне згладжування спрямоване на поступове уточнення оцінок коефіцієнтів ряду.

Експоненціальну середню першого порядку для ряду y визначають за формулою:

$$S_t^{[k]}(y) = a \sum_{i=0}^n (1 - a)^i y_{t-i}, \quad (2.3)$$

Де a - параметр згладжування ($0 < a < 1$).

Експоненціальна середня k -го порядку обчислюється як:

$$S_t^{[k]}(y) = a \sum_{i=0}^n (1 - a)^i S_{t-i}^{\{k-1\}}(y), \quad (2.4)$$

Для обчислення експоненціальної середньої Браун вивів рекурентну формулу (2.5) що дозволяє обчислювати середню як лінійну комбінацію

і початкових умов

$$\begin{cases} S_0^{[1]}(y) = a_0 - \frac{1-\alpha}{\alpha} a_1 \\ S_0^{[2]}(y) = a_0 \frac{2(1-\alpha)}{\alpha} a_1 \end{cases} \quad (2.11)$$

Прогноз для рівнів ряду на момент часу $t+1$ записується як:

$$\hat{y}_{t+1} = \hat{a}_0 + l\hat{a}_1, \quad (2.12)$$

Де l - період прогнозу.

Похибку прогнозу можна оцінити за формулою:

$$\sigma_{y1} \approx \sqrt{\frac{a}{(2-\alpha)^2} [1 + 4(1-\alpha) + 5(1-\alpha)^2 + 2a(4-3\alpha) + 2a^2l^3]}, \quad (2.13)$$

Де σ_{y1} - середньоквадратична похибка, обчислена для відхилень від лінійного тренду,

$$\sigma_{y1} = \sqrt{\frac{\sum_{i=1}^n (y_i - \tilde{y}_i)^2}{f}} \quad f = n-2, \quad (2.14)$$

$y_i = i, \tilde{y}_i (i = 1, n)$ - відповідно фактичні й теоретичні (знайдені на основі прогнозу) значення членів ряду; f -число ступенів свободи.

Якщо часовий ряд описується параболою другого порядку:

$$y_t = a_0 + a_1 t + \frac{1}{2} a_2 t^2 + u_t, \quad (2.15)$$

то основні показники експоненціального згладжування розраховуються за такими формулами:

Початкові умови:

$$\begin{cases} S_0^{[1]}(y) = a_0 - \frac{1-\alpha}{\alpha} a_1 + \frac{(1-\alpha)(2-\alpha)}{2a^2} a_2 \\ S_0^{[2]}(y) = a_0 - \frac{2(1-\alpha)}{\alpha} a_1 + \frac{(1-\alpha)(3-2\alpha)}{2a^2} a_2, \\ S_0^{[3]}(y) = a_0 - \frac{3(1-\alpha)}{\alpha} a_1 + \frac{(1-\alpha)(4-3\alpha)}{2a^2} a_2 \end{cases} \quad (2.16)$$

2) Експоненціальні середні:

$$\begin{cases} S_t^{[t]}(y) = \alpha y_t - (1-\alpha)S_{t-1}^{[1]}(y); \\ S_t^{[2]}(y) = \alpha S_0^{[1]}(y) + (1-\alpha)S_{t-1}^{[2]}(y), \\ S_t^{[3]}(y) = \alpha S_0^{[2]}(y) + (1-\alpha)S_{t-1}^{[3]}(y) \end{cases} \quad (2.17)$$

3) Оцінки коефіцієнтів тренду

$$\begin{cases} a_0 = 3 \left[S_t^{(1)}(y) - S_t^{(2)}(y) \right] + S_t^{(3)}(y) \\ a_1 = a_0 - \frac{1-\alpha}{2(1-\alpha)^2} \left[(6-5\alpha)S_t^{(1)}(y) - 2(5-4\alpha)S_t^{(2)}(y) + (4-3\alpha)S_t^{(3)}(y) \right] \\ a_2 = \frac{a^2}{(1-\alpha)^2} \left[S_t^{(1)}(y) - S_t^{(2)}(y) + S_t^{(3)}(y) \right] \end{cases} \quad (2.18)$$

4) модель, за якою робиться прогноз,

$$y_{t+l} = a_0 + a_1 l = \frac{1}{2} a_2 l^2, \quad (2.19)$$

5) похибка прогнозу

$$\sigma_{y_{t+l}} \approx \sigma_{y_t} \sqrt{2a + 3a^2 + 3a^3 t}, \quad (2.20)$$

Оцінку параметрів a_0 , a_1 , a_2 найкраще знаходити методом найменших квадратів.

У методі експоненціального згладжування важливим є вибір параметра згладжування a , оскільки при різних його значеннях прогнозні значення рівнів також різні. Якщо a наближається до одиниці, це свідчить про великий вплив на прогноз лише останніх спостережень; якщо a наближається до нуля, то ваги рівнів зменшуються повільно, тобто при прогнозуванні враховуються всі (або майже всі) попередні спостереження.

Коли є впевненість, що початкові умови досить правильні, параметр a беруть таким, що наближається до нуля;

у протилежному разі слід брати великі значення a . На практиці для знаходження значень параметра згладжування користуються формулою

$$a = \frac{2}{n+1}, \quad (2.20)$$

яка відображає залежність лише від загальної кількості рівнів ряду динаміки n .

Метод експоненціального згладжування забезпечує найкращі результати при роботі з великими часовими рядами. У випадку коротких рядів (15–20 спостережень) він може бути менш ефективним, оскільки не встигає врахувати всі зміни. Це особливо критично для економічних часових рядів, які зазвичай короткі та мають низькі темпи приросту. Точність прогнозу залежить від правильного вибору початкових умов та параметра a .

Експоненціальне згладжування також використовується для фільтрації випадкових шумів у даних, що особливо корисно для аналізу інформації, зібраної в реальному часі. Наприклад, параметри, такі як температура,

вологість чи тиск, схильні до короткострокових коливань через зміну умов (відкриття дверей, робота кондиціонера тощо). Використання експоненціального згладжування дозволяє зменшити вплив таких коливань.

Формула для обчислення згладженого значення:

$$X'_t = \alpha X_t + (1 - \alpha)X'_{t-1}, \quad (2.21)$$

$X_t=22.5^\circ\text{C}$ — температура, виміряна зараз.

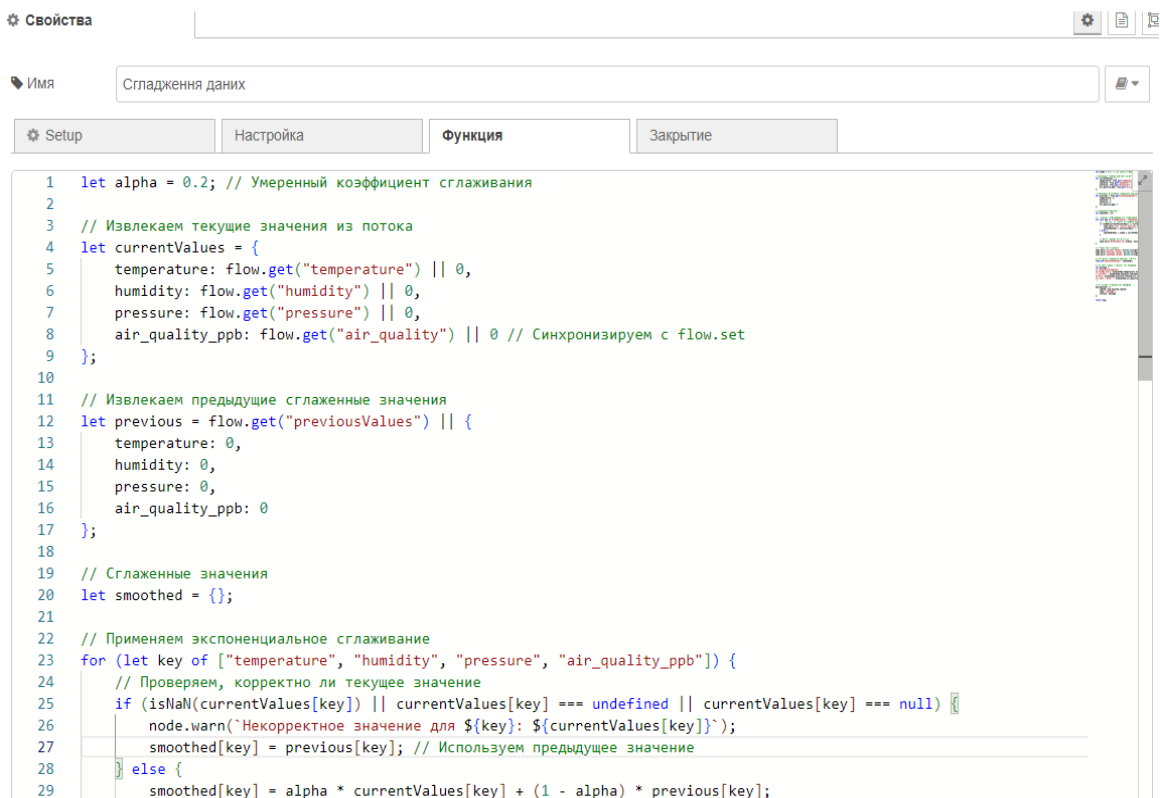
$X'_{t-1}=22.7^\circ\text{C}$ — згладжене значення температури попереднього моменту.

Якщо $\alpha=0.2$;

$$X'_t = 0.2 * 22.5 + 0.8 * 22.7 = 22.66^\circ\text{C}, \quad (2.22)$$

Таке згладжене значення мінімізує шум.

Приклад реалізації коду в функції Node-red:



```
1 let alpha = 0.2; // Умеренный коэффициент сглаживания
2
3 // Извлекаем текущие значения из потока
4 let currentValues = {
5   temperature: flow.get("temperature") || 0,
6   humidity: flow.get("humidity") || 0,
7   pressure: flow.get("pressure") || 0,
8   air_quality_ppb: flow.get("air_quality") || 0 // Синхронизируем с flow.set
9 };
10
11 // Извлекаем предыдущие сглаженные значения
12 let previous = flow.get("previousValues") || {
13   temperature: 0,
14   humidity: 0,
15   pressure: 0,
16   air_quality_ppb: 0
17 };
18
19 // Сглаженные значения
20 let smoothed = {};
21
22 // Применяем экспоненциальное сглаживание
23 for (let key of ["temperature", "humidity", "pressure", "air_quality_ppb"]) {
24   // Проверим, корректно ли текущее значение
25   if (isNaN(currentValues[key]) || currentValues[key] === undefined || currentValues[key] === null) {
26     node.warn("Некорректное значение для ${key}: ${currentValues[key]}");
27     smoothed[key] = previous[key]; // Используем предыдущее значение
28   } else {
29     smoothed[key] = alpha * currentValues[key] + (1 - alpha) * previous[key];
```

```

31
32 // Вывод разницы для проверки
33 node.warn(`Difference for ${key}: ${currentValues[key]} - previous[key]`);
34 }
35
36 // Отладочные сообщения
37 node.warn(`Current values: ${JSON.stringify(currentValues)}`);
38 node.warn(`Previous values: ${JSON.stringify(previous)}`);
39 node.warn(`Smoothed values: ${JSON.stringify(smoothed)}`);
40
41 // Сохраняем сглаженные значения в поток
42 flow.set("previousValues", smoothed);
43
44 // Форматирование сообщения для Telegram
45 let message = `
46 📊 Сглажені показники:
47 🌡️ Температура: ${smoothed.temperature.toFixed(2)} °C
48 💧 Вологість: ${smoothed.humidity.toFixed(2)} %
49 📏 Тиск: ${smoothed.pressure.toFixed(2)} hPa
50 🌬️ Якість повітря: ${smoothed.air_quality_ppb.toFixed(2)} ppb
51 `;
52
53 // Формируем сообщение для Telegram
54 msg.payload = {
55   chatId: msg.payload.chatId,
56   type: "message",
57   content: message
58 };
59
60 return msg;
61

```

Рисунок 2.2 – Код програми методу сгладжування в Node-red

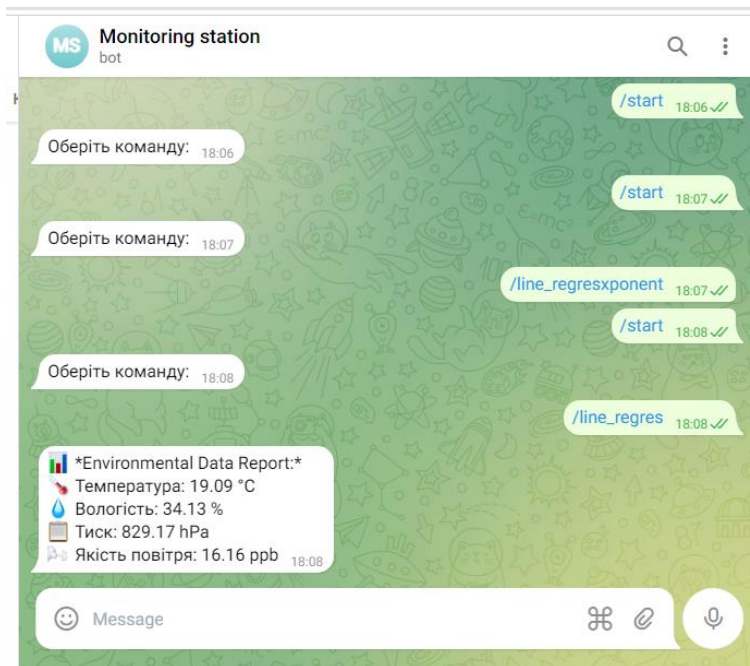


Рисунок 2.3 – Вивід даних в телеграм бот

2.2.2 Метод лінійної регресія

Метод лінійної регресії є ефективним і відносно простим інструментом для прогнозування значень залежних змінних на основі їхнього зв'язку з незалежними змінними. Наприклад, цей метод можна використовувати для прогнозування концентрації вуглекислого газу (CO₂) при моніторингу якості повітря в приміщенні.

Головна мета дослідження полягає у використанні його результатів для прогнозування майбутнього стану об'єкта або явища. Приклади такого підходу широко представлені у природничих, технічних та економічних дисциплінах. Якщо явище аналізується у взаємозв'язку з іншими, дослідник визначає чинники, які впливають на нього, оцінює ступінь їхнього впливу та характер взаємозв'язку.

Змінна, яку досліджують, називається залежною змінною і позначається літерою Y . Змінні, що впливають на Y , називаються незалежними змінними і позначаються x_1, x_2, \dots, x_k . Усі змінні вважаються числовими.

Нехай залежність між x і y має вигляд:

$$y = \alpha^0 + \alpha^1 x + \varepsilon, \quad (2.23)$$

де α^0, α^1 – постійні коефіцієнти, що називаються параметрами моделі; ε – випадкова величина з математичним сподіванням 0 і дисперсією s^2 . В цьому випадку рівняння регресії перетворюється на рівняння прямої

$$\bar{y}(x) = M\left(\frac{y}{x}\right) = a_0 + a_1 x, \quad (2.24)$$

Передбачимо, що незалежна змінна набула значень x_1, x_2, \dots, x_n , внаслідок чого залежна змінна набула значень y_1, y_2, \dots, y_n . У припущенні лінійної залежності отримуємо n рівностей

$$y_i = a_0 + a_1 x_i + \varepsilon_i, \quad i = 1 \dots n, \quad (2.25)$$

де ε_i – незалежні і розподілені так само, як ε .

Невідомі (x_i, y_i) . потрібно оцінити за значеннями пари. a_0, a_1 Як вже було сказано, кожна задача оцінювання пов'язана з певним критерієм якості.

У нашій теорії таким критерієм є критерій найменших квадратів:

$$\sum_{i=1}^n \varepsilon_i^2 - \min \quad (2.26)$$

Давайте запишемо цю суму по-іншому, щоб була помітніша закономірність від a_0, a_1

$$\sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n [\bar{y}(x) - y_i]^2 = \sum_{i=1}^n [y_i - a_0 - a_1 x_i]^2, \quad (2.27)$$

Це означає знаходження таких значень невідомих параметрів a_0, a_1 , за яких значення функції (2.28) мінімізується.

$$Q(a_0, a_1) = \sum_{i=1}^n [y_i - a_0 - a_1 x_i]^2, \quad (2.28)$$

Метод розв'язання цієї задачі відповідає тому, що викладається у вищих математичних класах. Знаходження частинних похідних функції Q і прирівнювання їх до нуля приводить до одночасного лінійного рівняння:

$$\begin{cases} \frac{\partial Q}{\partial a_0} = -2 \sum_{i=1}^n [y_i - a_0 - a_1 x_i] = 0 \\ \frac{\partial Q}{\partial a_1} = -2 \sum_{i=1}^n [y_i - a_0 - a_1 x_i] x_i = 0 \end{cases} \quad (2.29)$$

Після очевидного перетворення виходить вираз:

$$\begin{cases} n a_0 + a_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \\ a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i \end{cases} \quad (2.29)$$

Визначимо вибірккову середню.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (2.30)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad (2.31)$$

$$\overline{xy} = \frac{1}{n} \sum_{i=1}^n x_i y_i, \quad (2.32)$$

$$\overline{x^2} = \frac{1}{n} \sum_{i=1}^n x_i^2, \quad (2.33)$$

У цих позначеннях, розділивши кожне з одночасних рівнянь на n , ми отримаємо такий вигляд,

$$\begin{cases} a_0 + a_1 \bar{x} = \bar{y} \\ a_0 \bar{x} + a_1 \bar{x}^2 = \overline{xy} \end{cases}, \quad (2.33)$$

а її рішення (шукані оцінки коефіцієнтів рівняння регресії) буде таким:

$$\widehat{a_0} = \frac{\overline{x^2 y} - \overline{xy} * \bar{x}}{\overline{x^2} - (\bar{x})^2}, \quad (2.34)$$

$$\widehat{a_1} = \frac{\bar{x} * \bar{y} - \overline{y} * \bar{x}}{\overline{x^2} - (\bar{x})^2}, \quad (2.35)$$

Уведемо також позначення $S_x^2 = \overline{x^2} - (\bar{x})^2$ яке перетворює вираз a_0 :

$$\widehat{a_0} = \frac{\overline{x^2 y} - \overline{xy} * \bar{x}}{S_x^2} = \frac{S_x^2 \bar{y} - \bar{x}(\overline{xy} - \bar{y} \bar{x})}{S_x^2} = \bar{y} - \widehat{a_1} \bar{x}, \quad (2.36)$$

то оцінка функції регресії набуде такого вигляду:

$$\bar{y}(x) = \widehat{a_0} + \widehat{a_1} x = \bar{y} - \widehat{a_1} \bar{x} + \widehat{a_1} x = \widehat{a_1} (x - \bar{x}), \quad (2.37)$$

Множинна регресія та етапи побудови

Регресія називається множинною, якщо вона відображає залежність результативної змінної від декількох факторів. Побудова моделі множинної регресії включає кілька етапів. Вибір форми зв'язку (рівняння регресії); Вибір характеристик факторів; Забезпечення достатнього обсягу вибірки для отримання незміщеної оцінки. Розглянемо кожен з них більш детально. Дані, зібрані для регресійного аналізу, зазвичай є «історичними» фактами, тобто цифрами, що показують значення фактора за попередній період або для певної географічної області. Вони використовуються для обчислення оцінок коефіцієнтів регресії та визначення того, наскільки модель відповідає фактичним змінам результативної змінної. Для досягнення цих цілей можна використовувати різні методи, але найбільш універсальним і широко застосовуваним є описаний вище метод найменших квадратів: (2.26) При використанні математичних інструментів вибір форми зв'язку іноді буває складним, оскільки зв'язок між змінними теоретично може бути виражений кількома різними функціями. Незалежно від форми зв'язку, вибирається декілька рівнянь, які можуть певною мірою описати цей зв'язок. Найзручніший спосіб визначити форму вихідного рівняння регресії - перевірити його за допомогою різних рівнянь.

Суть цього методу полягає в тому, що кілька рівнянь (моделей) регресії, відібраних для опису зв'язку між соціально-економічними явищами і процесами, реалізуються на комп'ютері за допомогою спеціально розроблених алгоритмів пошуку з подальшою статистичною перевіркою, заснованою в основному на t-критерії Стьюдента і F-критерії Фішера.

Основною метою цього є пошук найкращого можливого розв'язку задачі рівняння регресії. Алгоритм пошуку базується на t-критерії Стьюдента та F-критерії Фішера. З іншого боку, цей метод пошуку є дуже трудомістким і передбачає великий обсяг обчислювальної роботи.

Важливим етапом побудови попередньо відібраного рівняння множинної регресії є відбір і подальше включення факторних ознак. Складність побудови рівнянь множинної регресії пов'язана з тим, що майже

всі факторні ознаки є взаємозалежними. Питання розмірності реляційної моделі, тобто визначення оптимальної кількості факторних ознак, є однією з основних проблем при побудові рівнянь множинної регресії. З одного боку, чим більше факторних ознак включено в рівняння, тим краще описується явище. Однак моделі з більш ніж 100 факторними ознаками складні в реалізації і вимагають багато машинного часу. Зменшення розміру моделі за рахунок виключення дрібних, економічно та статистично несуттєвих факторів сприяє простоті та якості реалізації. Водночас побудова регресійних моделей низької розмірності може призвести до того, що такі моделі не будуть адекватно відображати досліджуване явище або процес. Найбільш прийнятним методом відбору факторних ознак є покроковий регресійний аналіз.

Суть методу покрокової регресії полягає в послідовному включенні факторів у рівняння регресії з подальшою перевіркою їх значущості. Фактори вводяться в рівняння по одному за допомогою так званого «прямого методу». При перевірці значущості вхідних факторів визначається, наскільки зменшується сума квадратів залишків і наскільки збільшується значення коефіцієнта множинної кореляції. У цьому випадку використовується зворотний метод, тобто виключення факторів, які стали незначущими за t -критерієм Стьюдента. Фактор вважається незначущим, якщо його включення в рівняння регресії змінює лише значення коефіцієнта регресії, не зменшуючи суму квадратів залишків і не збільшуючи їх значення. Якщо включення відповідної факторної змінної в модель не змінює коефіцієнт регресії (або змінює несуттєво), але збільшує значення коефіцієнта множинної кореляції, то ця змінна є значущою і повинна бути включена в рівняння регресії. Однак, якщо включення факторної ознаки в модель змінює не тільки величину, але й знак коефіцієнта регресії, але не збільшує коефіцієнт множинної кореляції, вважається недоцільним включати цю факторну ознаку в кореляційну модель. Розглянемо побудову множинної регресії на прикладі вище. Оскільки відомо, що на частку ринку впливають інші фактори, окрім ціни, і оскільки майже всі

маркетингові проблеми пов'язані з кількома різними факторами, у більшості випадків слід використовувати модель множинної регресії. Концепції та методи, що використовуються в множинному регресійному аналізі, по суті, однакові, з деякими змінами та доповненнями, пов'язаними з одночасним вивченням кількох факторів. Побудова моделі множинної регресії пов'язана з іншими складнощами. По-перше, неможливо дослідити взаємозв'язки за допомогою діаграм розсіювання. Звичайно, їх можна використовувати для того, щоб показати зв'язок між результуючою змінною та одним із незалежних факторів по черзі. Однак слід пам'ятати, що незалежні змінні часто впливають один на одного, а також на результативну змінну, тому інформація, отримана таким чином, має обмежену цінність. Розглянемо приклад пошуку моделі множинної регресії з двома змінними в рівнянні.

$$\begin{cases} \Sigma y_i = nb^0 + b^1 \Sigma x_i^1 + b^2 \Sigma x_i^2 \\ \Sigma y_i x_i^1 = b^0 \Sigma x_i^1 + b^1 \Sigma x_i^1 x_i^1 + b^2 \Sigma x_i^1 x_i^2 \\ \Sigma y_i x_i^2 = b^0 \Sigma x_i^2 + b^1 \Sigma x_i^1 x_i^2 + b^2 \Sigma x_i^2 x_i^2 \end{cases}, \quad (2.39)$$

Її можна розв'язати за допомогою методу Крамера та формул:

$$b^0 = \bar{y} - b^1 \bar{x}^1 - b^2 \bar{x}^2, \quad (2.40)$$

$$b^1 = \frac{\Sigma(x_i^1 - \bar{x}^1)(y_i - \bar{y})\Sigma(x_i^2 - \bar{x}^2)^2 - \Sigma(x_i^2 - \bar{x}^2)(y_i - \bar{y})\Sigma(x_i^1 - \bar{x}^1)(x_i^2 - \bar{x}^2)}{\Sigma(x_i^1 - \bar{x}^1)^2 \Sigma(x_i^2 - \bar{x}^2)^2 - (\Sigma(x_i^1 - \bar{x}^1)(x_i^2 - \bar{x}^2))^2}, \quad (2.41)$$

$$b^2 = \frac{\Sigma(x_i^2 - \bar{x}^2)(y_i - \bar{y})\Sigma(x_i^1 - \bar{x}^1)^2 - \Sigma(x_i^1 - \bar{x}^1)(y_i - \bar{y})\Sigma(x_i^1 - \bar{x}^1)(x_i^2 - \bar{x}^2)}{\Sigma(x_i^1 - \bar{x}^1)^2 \Sigma(x_i^2 - \bar{x}^2)^2 - (\Sigma(x_i^1 - \bar{x}^1)(x_i^2 - \bar{x}^2))^2}, \quad (2.42)$$

Розглянемо етапи проведення регресійного аналізу:

1. Розробка системи факторів, які суттєво впливають на результативний показник
2. Розробка моделі, яка відображає загальний зміст досліджуваного зв'язку та кількісне визначення її параметрів
3. Перевірка якості моделі - оцінка впливу окремих факторів.

На першому етапі відбираються фактори, які суттєво впливають на результативні показники. Це робиться насамперед на основі змістовного аналізу. Для отримання достовірних оцінок модель не повинна включати багато факторів, кількість яких не повинна перевищувати третини аналізованих даних. Однак на ранніх стадіях розробки моделі дослідник не має чіткої відповіді на питання про істотний набір факторів, тому при використанні комп'ютера відбір факторів зазвичай здійснюється безпосередньо в процесі створення моделі методом послідовної регресії. Суть цього методу полягає в послідовному включенні в модель додаткових факторів і оцінці впливу додаткових факторів. Інший підхід полягає в тому, щоб не накладати особливих обмежень на фактори, включені в попередню модель, а оцінювати і відбирати їх лише на наступному етапі. Другий етап починається з розробки моделі, яка відображає загальний зміст досліджуваного зв'язку. Регресійна модель - це, з точки зору дослідника, формула (або система формул), яка вказує, які фактори слід аналізувати з точки зору взаємозв'язку. Рівняння регресії також дають уявлення про форму зв'язку. Класифікація рівнянь регресії може бути показана у вигляді схеми на рисунку (2.4). Якщо регресія відображає залежність між результуючою змінною і одним фактором, вона називається однопарною регресією. Якщо регресія відображає залежність результативної змінної від декількох факторів, вона називається множинною регресією. Якщо залежність є лінійною по відношенню до параметрів (але не обов'язково по відношенню до незалежних змінних), регресія називається лінійною. Якщо ні, то регресія називається нелінійною.

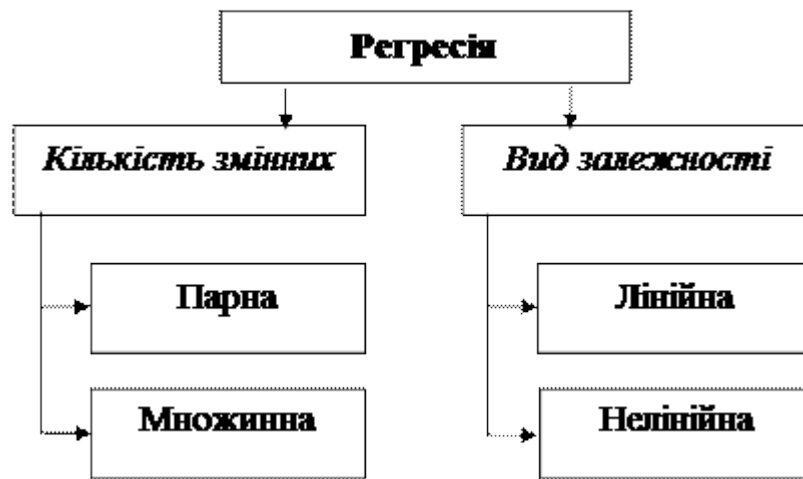


Рисунок 2.4 – Класифікація регресійної залежності

Після вибору змінних та їх представлення наступним кроком є визначення форми рівняння регресії. Основою для ідентифікації та встановлення аналітичної форми зв'язку є використання певних математичних функцій - лінійної, логарифмічної, степеневі, експоненціальної, поліноміальної та деяких інших. На практиці для розрахунку параметрів функцій використовуються спеціальні комп'ютерні програми, серед яких найбільшій можливості для інтерпретації результатів надає програма лінійного регресійного аналізу. Саме тому більшість аналітиків віддають перевагу саме їй. Дані, зібрані для регресійного аналізу, зазвичай є «історичними даними», тобто цифрами, що показують значення кожного фактора в кожному з попередніх періодів або географічних регіонів. Вони використовуються для отримання оцінок коефіцієнтів регресії та визначення того, наскільки добре модель відповідає змінам результативної змінної. Для досягнення цих цілей можна використовувати різні методи, але найбільш універсальним і найчастіше застосовуваним є метод найменших квадратів. Оцінка за методом найменших квадратів - це таке значення коефіцієнтів рівняння регресії, яке мінімізує суму квадратів відхилень спостережуваних значень результативної змінної (Y_i) від значень, отриманих з рівняння $Y(X_i)$. Для оцінки якості моделі та повноти набору пояснювальних змінних зазвичай використовують коефіцієнт детермінації R^2 . Його також називають наближеною ймовірністю

або довірчою ймовірністю. Коефіцієнт детермінації R^2 - це відношення дисперсії, що пояснюється регресійним аналізом, до загальної дисперсії. Коефіцієнт детермінації кількісно визначає ступінь аналізованого зв'язку. Він показує, яка частина варіації результативної ознаки знаходиться під впливом досліджуваного фактора, тобто визначає, яка частка варіації ознаки Y врахована в моделі і пояснюється впливом незалежних факторів: чим ближче R^2 до 1, тим точніше рівняння регресії пояснює досліджуваний фактор (для функціонального зв'язку R^2 дорівнює 1. а для відсутності зв'язку R^2 дорівнює 0

Етапи реалізації методу лінійної регресії

Збір даних налаштовуємо ESP32 для періодичного зчитування даних із датчиків (температура, вологість, тиск, концентрація забруднень). Відправляємо ці данні на сервер через Wi-Fi.

Зібрані дані необхідно проаналізувати, щоб виявити кореляції між незалежними змінними (температура, вологість, тиск) і залежною змінною (наприклад, концентрацією CO_2).

Розрахунок коефіцієнтів лінійної регресії:

$$Y = a + b_1X_1 + b_2X_2 + a_3X_3 , \quad (2.43)$$

де:

X_1 — температура.

X_2 — вологість.

X_3 — тиск.

Y — концентрація CO_2

Використовуємо метод найменших квадратів для визначення коефіцієнтів a , b_1 , b_2 , b_3 .

$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (2.40)$$

Де:

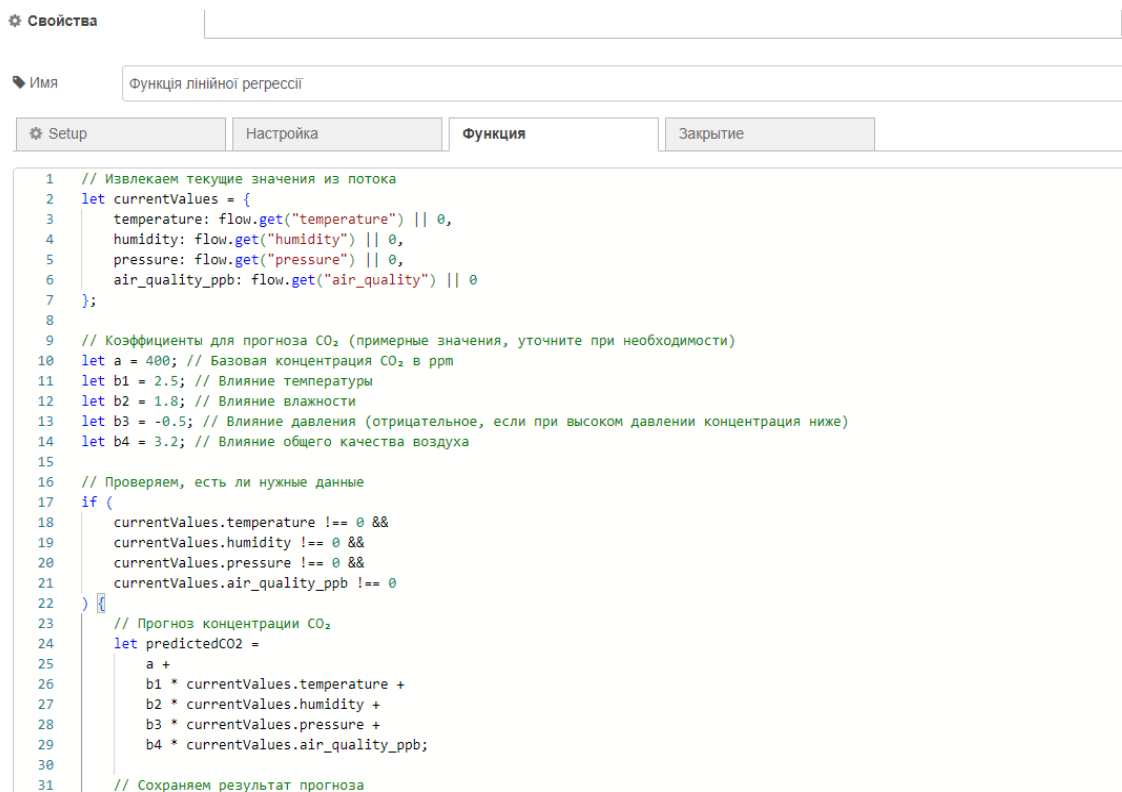
Y_i — фактичні значення

\hat{Y}_i — прогнозовані значення

n — кількість спостережень

В Node-RED, ці обчислення автоматизуються за допомогою бібліотек, таких як NumPy

Приклад реалізації коду в функції Node-red:



```
1 // Извлекаем текущие значения из потока
2 let currentValues = {
3   temperature: flow.get("temperature") || 0,
4   humidity: flow.get("humidity") || 0,
5   pressure: flow.get("pressure") || 0,
6   air_quality_ppb: flow.get("air_quality") || 0
7 };
8
9 // Коэффициенты для прогноза CO2 (примерные значения, уточните при необходимости)
10 let a = 400; // Базовая концентрация CO2 в ppm
11 let b1 = 2.5; // Влияние температуры
12 let b2 = 1.8; // Влияние влажности
13 let b3 = -0.5; // Влияние давления (отрицательное, если при высоком давлении концентрация ниже)
14 let b4 = 3.2; // Влияние общего качества воздуха
15
16 // Проверяем, есть ли нужные данные
17 if (
18   currentValues.temperature !== 0 &&
19   currentValues.humidity !== 0 &&
20   currentValues.pressure !== 0 &&
21   currentValues.air_quality_ppb !== 0
22 ) {
23   // Прогноз концентрации CO2
24   let predictedCO2 =
25     a +
26     b1 * currentValues.temperature +
27     b2 * currentValues.humidity +
28     b3 * currentValues.pressure +
29     b4 * currentValues.air_quality_ppb;
30
31   // Сохраняем результат прогноза
```

```

32     context.set("lastPrediction", {
33         predictedCO2: predictedCO2.toFixed(2),
34         temperature: currentValues.temperature,
35         humidity: currentValues.humidity,
36         pressure: currentValues.pressure,
37         airQuality: currentValues.air_quality_ppb
38     });
39
40     // Форматирование сообщения для Telegram
41     let message = `
42     Функция лінійної регресії:
43     🌡️ Температура: ${currentValues.temperature.toFixed(2)} °C
44     💧 Вологість: ${currentValues.humidity.toFixed(2)} %
45     📏 Тиск: ${currentValues.pressure.toFixed(2)} hPa
46     🌬️ Якість повітря: ${currentValues.air_quality_ppb.toFixed(2)} ppb
47     🌫️ Прогнозована концентрація CO2 : ${predictedCO2.toFixed(2)} ppm
48     `;
49
50     // Формируем сообщение для Telegram
51     msg.payload = {
52         chatId: msg.payload.chatId, // ID чата для Telegram
53         type: "message", // Тип сообщения
54         content: message // Сформированное сообщение
55     };
56
57     return msg;
58 } else {
59     node.warn("Недостаточно данных для вычисления прогноза концентрации CO2");
60     return null;
61 }
62

```

Рисунок 2.4 – Код програми методу лінійної регресії в Node-red



Рисунок 2.5 – Вивід даних лінійної регресії в телеграм бот

Заключний алгоритм роботи:

Ініціалізація системи.

Зчитування даних із датчиків.

Обчислення управляючої дії.

Передача та обробка даних:

Виведення на OLED-дисплей.

Відправка на MQTT.

Отримання та обробка даних за допомогою Node-Red

Обмін запитами з Telegram-bot

Цей підхід дозволяє ефективно отримати, обробити та вивести данні, з віддаленої станції, моніторинг якості повітря за допомогою ІОТ.

Висновки до розділу

У другому розділі виконано детальне дослідження математичних методів обробки даних, які використовуються для підвищення точності вимірювань у системах моніторингу якості повітря. Основна увага приділялася експоненціальному згладжуванню та лінійній регресії.

Метод експоненціального згладжування було проаналізовано як ефективний підхід для зменшення випадкових коливань даних, отриманих від датчиків. Встановлено, що цей метод дозволяє згладжувати вплив короточасних змін, викликаних, наприклад, технічними похибками чи змінними погодними умовами, і таким чином забезпечує стабільність та якість отриманих даних. Це є особливо важливим у контексті моніторингу якості повітря, де вимірювання можуть змінюватися динамічно.

Лінійна регресія розглянута як інструмент для встановлення залежностей між рівнем забруднення повітря та зовнішніми факторами, такими як температура, вологість чи атмосферний тиск. Аналіз показав, що використання цього методу дозволяє не лише виявляти тенденції у даних, але й створювати прогностичні моделі для оцінки змін якості повітря за певних умов. Це робить систему більш адаптивною та дозволяє розробляти рекомендації для попередження негативного впливу забруднень.

Отже, в цьому розділі закладено основу для розробки програмного забезпечення, яке забезпечує стабільність, надійність та точність роботи

системи моніторингу. Використання описаних методів дозволяє обробляти великі обсяги даних у реальному часі, що є важливим для забезпечення оперативності реагування.

РОЗДІЛ 3

ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ

3.1 Вибір пристроїв та датчиків

3.1.1 Датчик якості повітря AGS02MA (TVOC)

Датчик AGS02MA є високопродуктивним пристроєм для виявлення летких органічних сполук (TVOC), який відзначається високою чутливістю, низьким енергоспоживанням, довгостроковою стабільністю та високою надійністю.

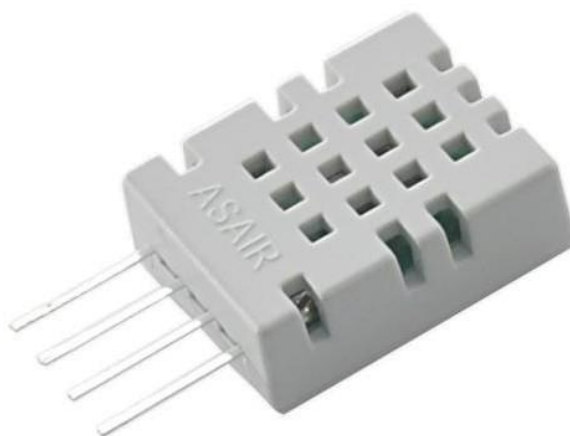


Рисунок 3.1 — AGS02MA

Оснащений спеціальним чіпом ASIC, датчик використовує передові технології цифрової обробки сигналів і аналізу газів. Завдяки цьому він повністю відкалібрований і готовий до використання одразу після доставки. Його основне призначення – моніторинг якості повітря, зокрема виявлення таких органічних газів, як етанол, аміак, сульфід, бензолні пари та дим. Датчик знаходить застосування в очищувачах повітря, системах вентиляції, побутовій техніці та іншому обладнанні, яке контролює якість повітря.

Датчик виготовлений із використанням сучасної MEMS-технології (мікроелектромеханічних систем). У його основі лежить чутливий матеріал, створений на базі оксиду металу, що нанесений на кремнієву підкладку. У

чистому повітрі цей матеріал має низьку провідність. При контакті із леткими органічними сполуками провідність матеріалу змінюється залежно від концентрації газу: чим більше виявленого газу, тим вищою стає провідність. На основі цих змін спеціальна інтегральна схема перетворює провідність у цифровий сигнал, пропорційний концентрації виявленого газу.

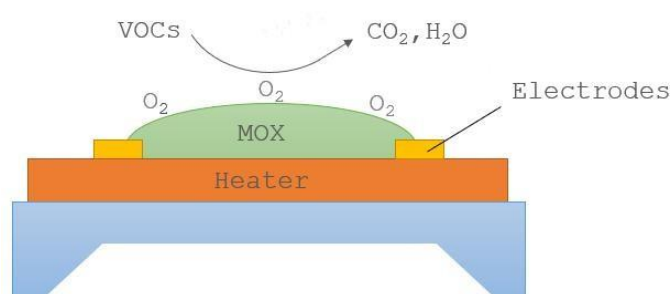


Рисунок 3.2 — Принцип роботи AGS02MA

Датчик працює при напрузі від 3,3 до 5,5 В постійного струму і споживає струм на рівні 28 ± 5 мА. Його типове енергоспоживання становить 98 мВт при напрузі 3,3 В та 130 мВт при 5 В. Для коректної роботи потрібен період вибірки щонайменше 2 секунди, а також час попереднього нагріву не менше 120 секунд. Робочий температурний діапазон становить від 0 до 50 °С, а допустима вологість – від 0 до 95% RH. Датчик розрахований на тривалий термін служби понад 5 років за умови експлуатації при температурі 25 °С у чистому повітрі.

AGS02MA має діапазон вимірювання від 0 до 99999 ppb (часток на мільярд) із точністю до 25% від виміряного значення при стандартних умовах (25 °С і 50% RH). Основним газом, для якого проводиться калібрування датчика, є етанол. Його вихідний сигнал задається в одиницях ppb, що дозволяє чітко оцінювати концентрацію виявлених речовин.

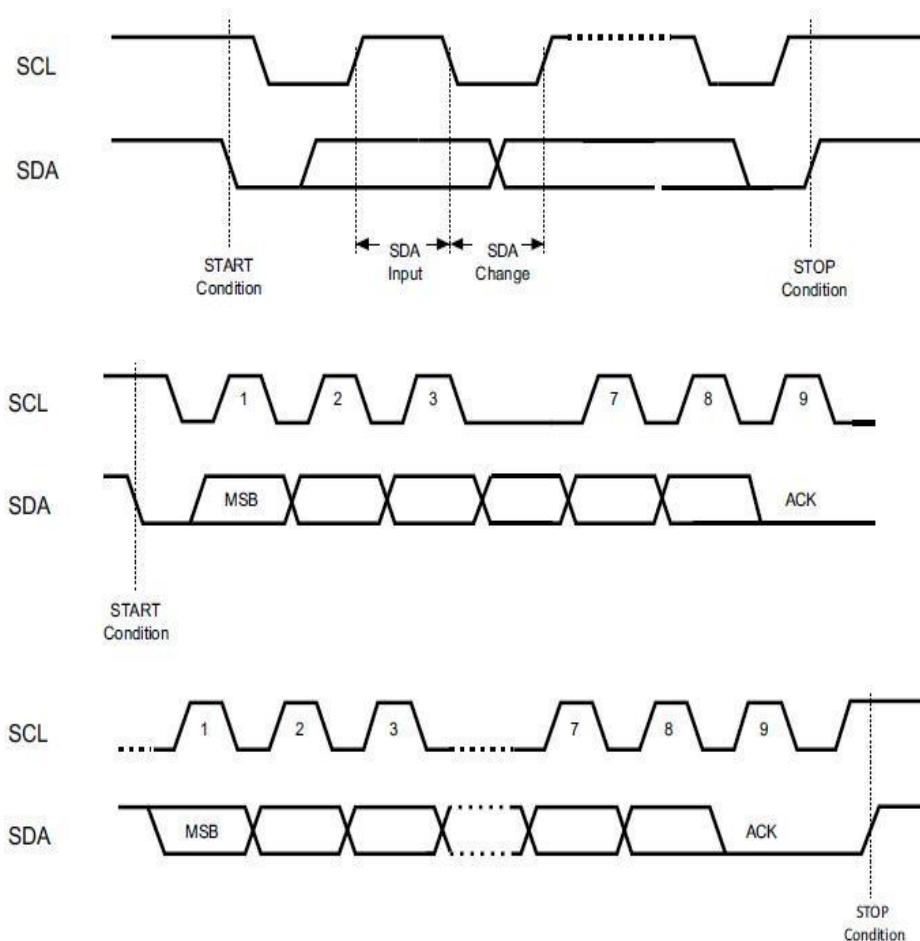


Рисунок 3.3 — Тимчасова діаграма шини I2C

Датчик підтримує стандартний протокол зв'язку I2C, що забезпечує простоту підключення до різних пристроїв. Для роботи використовуються дві основні лінії: послідовні дані (SDA) і послідовний годинник (SCL), які підключаються до джерела живлення через підтягувальні резистори номіналом від 1 до 10 кОм. Адреса датчика в I2C-протоколі складає 0x1A (у 7-бітному режимі), а швидкість зв'язку не повинна перевищувати 30 кГц. Команди для запису та читання даних мають значення 0x34 та 0x35 відповідно. Цей датчик є універсальним і широко використовується для моніторингу якості повітря у системах очищення, вентиляції та побутових пристроях, що забезпечують комфортні та безпечні умови в приміщеннях.

Таблиця 3.1 Інтерпретація показань TVOC

TVOC (ppb)	Шкала	Опис
<= 220	1	Добрий
<= 660	3	Помірна
<= 1430	7	Поганий
<= 2200	10	Нездоровий
<= 3300	15	Дуже шкідливо для здоров'я
<= 5500	25	Небезпечний
> 5500	50	Надзвичайно небезпечний

3.1.2 Датчик тиску BMP280

Датчик BMP280, створений компанією Robert Bosch, є одним із найпопулярніших рішень для вимірювання барометричного тиску в сучасних портативних пристроях. Завдяки своїй точності, низькому енергоспоживанню та компактному дизайну, він широко використовується в смартфонах, GPS-модулях, фітнес-трекерах та інших гаджетах. BMP280 є удосконаленою версією попередньої моделі BMP180, пропонуючи кращу продуктивність і більше функцій.



Рисунок 3.4 — Датчик тиску BMP280

Цей датчик працює на основі п'єзорезистивної технології вимірювання тиску, що характеризується високою стійкістю до електромагнітних завад, відмінною лінійністю і довготривалою стабільністю. Його виробництво базується на інноваційній технології Bosch MEMS APSM, яка забезпечує сумісність із CMOS-технологіями та високу герметичність компонентів. Завдяки цьому BMP280 має чудові експлуатаційні характеристики навіть у складних умовах.

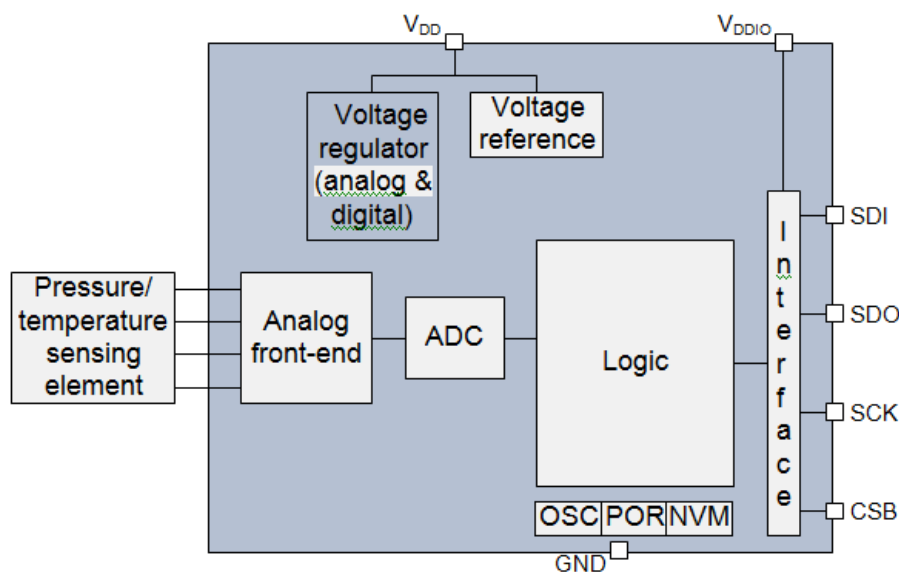


Рисунок 3.5 — спрощена блок-схема BMP280

Пристрій виготовлений у компактному корпусі з розмірами $2,0 \times 2,5$ мм і висотою 0,95 мм. Металева кришка забезпечує додатковий захист від впливів зовнішнього середовища. Низьке енергоспоживання датчика робить його ідеальним для пристроїв на батарейках. Наприклад, у режимі роботи з частотою 1 Гц він споживає всього 2,7 мкА. BMP280 підтримує інтерфейси SPI та I2C, що дозволяє інтегрувати його з різними електронними системами.

Датчик може працювати в кількох режимах: у сплячому режимі він не проводить вимірювань, у нормальному режимі чергуються цикли вимірювання і очікування, а у форсованому режимі здійснюється одноразове вимірювання з поверненням до енергозберігаючого стану. Крім того, користувач може налаштувати передискретизацію вимірювань для адаптації точності до потреб конкретного застосування. Це дозволяє знижувати

енергоспоживання в простих задачах або забезпечувати високу точність у критичних сценаріях.

Датчик BMP280 здатний ефективно вирішувати завдання, пов'язані з навігацією в приміщеннях, моніторингом погодних умов, контролем висоти та виявленням змін тиску. Наприклад, у навігаційних системах він дозволяє точно визначати висоту або навіть поверх у будівлі завдяки точності $\pm 0,12$ гПа, що еквівалентно похибці висоти ± 1 м. У метеорологічних системах його можна налаштувати на енергозберігаючий режим для тривалої автономної роботи.

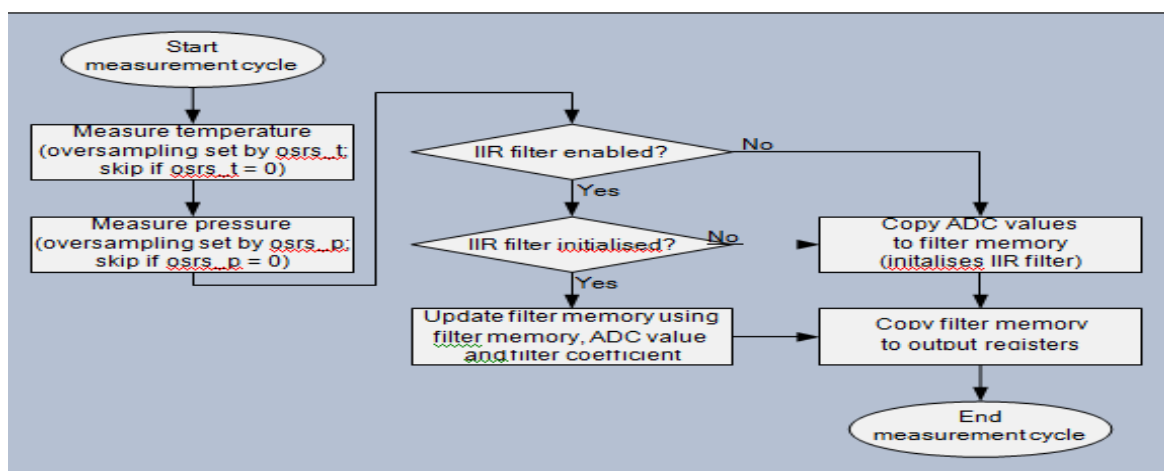


Рисунок 3.6 — Цикл вимірювання BMP280

Принцип роботи датчика базується на вимірюванні тиску і температури за допомогою п'єзорезистивного елемента та ASIC-чіпа, який виконує аналого-цифрове перетворення і корекцію сигналу. Вбудований ІІР-фільтр дозволяє мінімізувати короточасні шуми, такі як різкі звуки чи вібрації. Температурна компенсація гарантує стабільність вимірювань навіть за змінних температур, а низький температурний коефіцієнт зсуву забезпечує мінімальний дрейф даних.

Датчик має два контакти живлення: один забезпечує роботу аналогових і цифрових компонентів, а інший живить цифровий інтерфейс. Якщо живлення подається лише на контакт цифрового інтерфейсу, датчик автоматично переходить у високий імпедансний стан, що дозволяє використовувати шину

інтерфейсу без порушення роботи інших пристроїв.

Завдяки своїй високій продуктивності, багатофункціональності та здатності працювати в умовах низького енергоспоживання, BMP280 є ідеальним вибором для мобільних додатків, які вимагають точного вимірювання тиску та температури. Він легко адаптується для різних потреб, від метеорологічного моніторингу до портативних систем навігації.

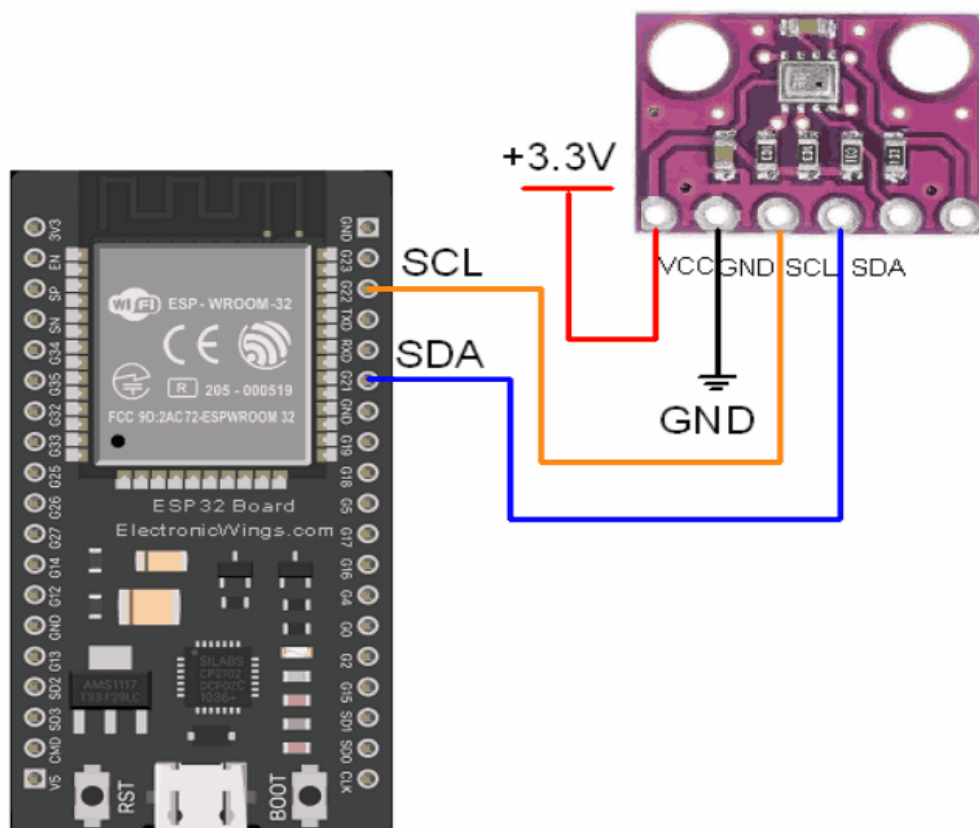


Рисунок 3.7 — Схема підключення BMP 280 до ESP 32

3.1.3 Датчик АНТ20

Датчиків температури та вологості, АНТ20 встановлює новий стандарт щодо розміру та продуктивності: дворядний плоский штифтовий корпус без SMD, придатний для пайки оплавленням, нижня поверхня 3x3 мм, висота 1,0 мм. Датчик виводить калібровані цифрові сигнали. у стандартному форматі ІС.



Рисунок 3.8 АНТ20 Датчик вологості та температури

АНТ20 оснащений нещодавно розробленим спеціальним чіпом ASIC, покращеним MEMS-напівпровідниковим ємнісним елементом датчика вологості та стандартним елементом датчика температури на чіпі. Його продуктивність була значно покращена або навіть перевищила рівень надійності датчиків попереднього покоління.

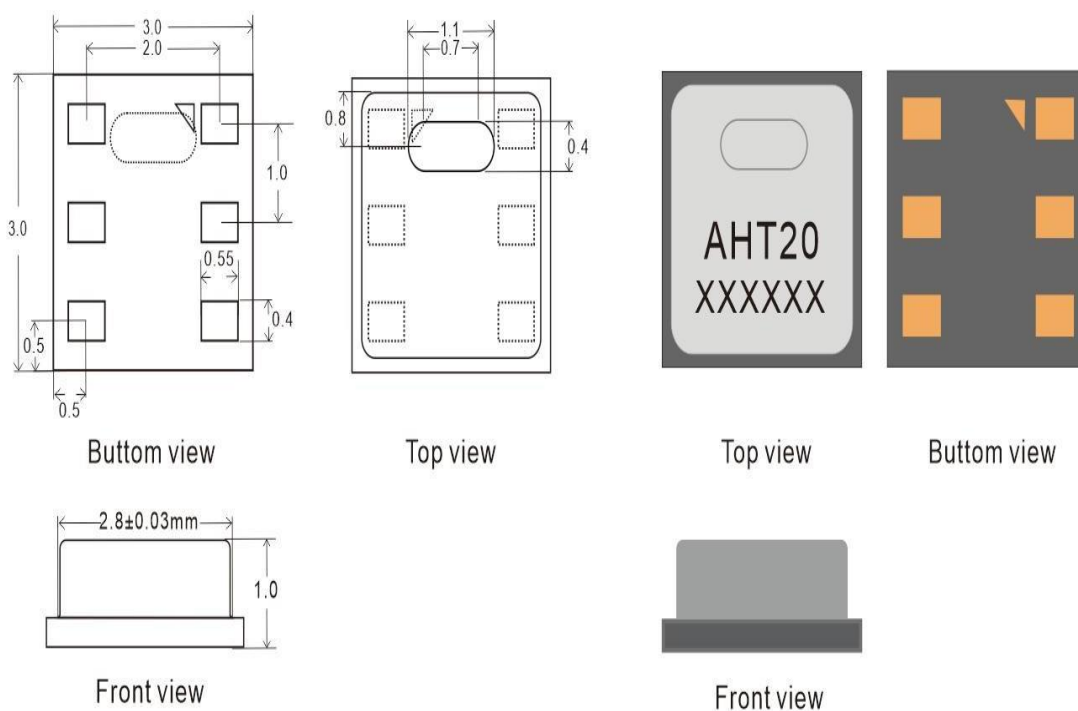


Рисунок 3.9 — Схема датчика АНТ20 (Одиниці вимірювання: мм
Допуск: $\pm 0,1$ мм).

Покоління датчиків температури та вологості було вдосконалено, щоб

зробити їх роботу більш стабільною в суворих умовах. Кожен датчик відкалібровано та перевірено, а номер партії продукту надруковано на поверхні продукту. Завдяки вдосконаленням і мініатюризації датчика, він є економічно ефективнішим, і в кінцевому підсумку всі пристрої отримують переваги від передових енергозберігаючих режимів роботи

3.1.4 OLED-дисплей SSD1306 128x32 0.91

Це компактний графічний модуль, оснащений контролером SSD1306, який широко використовується для керування органічними світлодіодними панелями. Завдяки інтегрованим функціям, таким як регулювання контрастності, вбудований SRAM-буфер для зберігання графічних даних і внутрішній генератор сигналів, дисплей забезпечує стабільну і зручну роботу. Це робить його популярним вибором для застосування у портативних пристроях, включаючи медичне обладнання, MP3-плеєри, калькулятори, гаджети та допоміжні екрани для мобільних телефонів.



Рисунок 3.10 — OLED SSD1306

Дисплей має роздільну здатність 128x32 пікселі, що дозволяє відображати текстову інформацію, символи або просту графіку. Завдяки діагоналі 0.91 дюйма, модуль є компактным і легко інтегрується у пристрої з обмеженим простором. Контролер SSD1306 підтримує два основні типи інтерфейсів — I2C та SPI, що дозволяє легко інтегрувати його у різноманітні проекти на базі мікроконтролерів. Вбудований SRAM-буфер обсягом 128x64 біт забезпечує ефективне управління пам'яттю і швидке оновлення відображуваного зображення.

Для роботи дисплея необхідне живлення від кількох джерел. Напруга

для живлення логічної частини (VDD) становить від 1,65 до 3,3 В. Напруга для підсилювача (VBAT) має бути у межах від 3,3 до 4,2 В, а для живлення панелі OLED (VCC) використовується зовнішнє джерело з напругою від 7 до 15 В. Модуль є енергоефективним, споживаючи максимальний струм лише 15 мА, а струм для кожного сегмента дисплея обмежується до 100 мкА. Крім цього, пристрій підтримує 256-рівневе регулювання яскравості, що дозволяє налаштувати його для роботи в різних умовах освітлення.

SSD1306 забезпечує додаткові функції, які підвищують його універсальність. Він має можливість програмування частоти кадрів, підтримує функцію прокручування зображення в горизонтальному та вертикальному напрямках, а також дозволяє змінювати систему координат відповідно до потреб користувача. Завдяки широкому діапазону робочих температур від -40°C до +85°C, дисплей може працювати у складних умовах експлуатації, що робить його придатним для використання у промислових і комерційних застосунках.

Таким чином, OLED SSD1306 128x32 0.91 — це надійний і функціональний дисплей, який поєднує компактні розміри, енергоефективність і широкий спектр можливостей. Він ідеально підходить для розробки пристроїв, де важливі невеликі розміри, низьке енергоспоживання і висока якість відображення графіки.

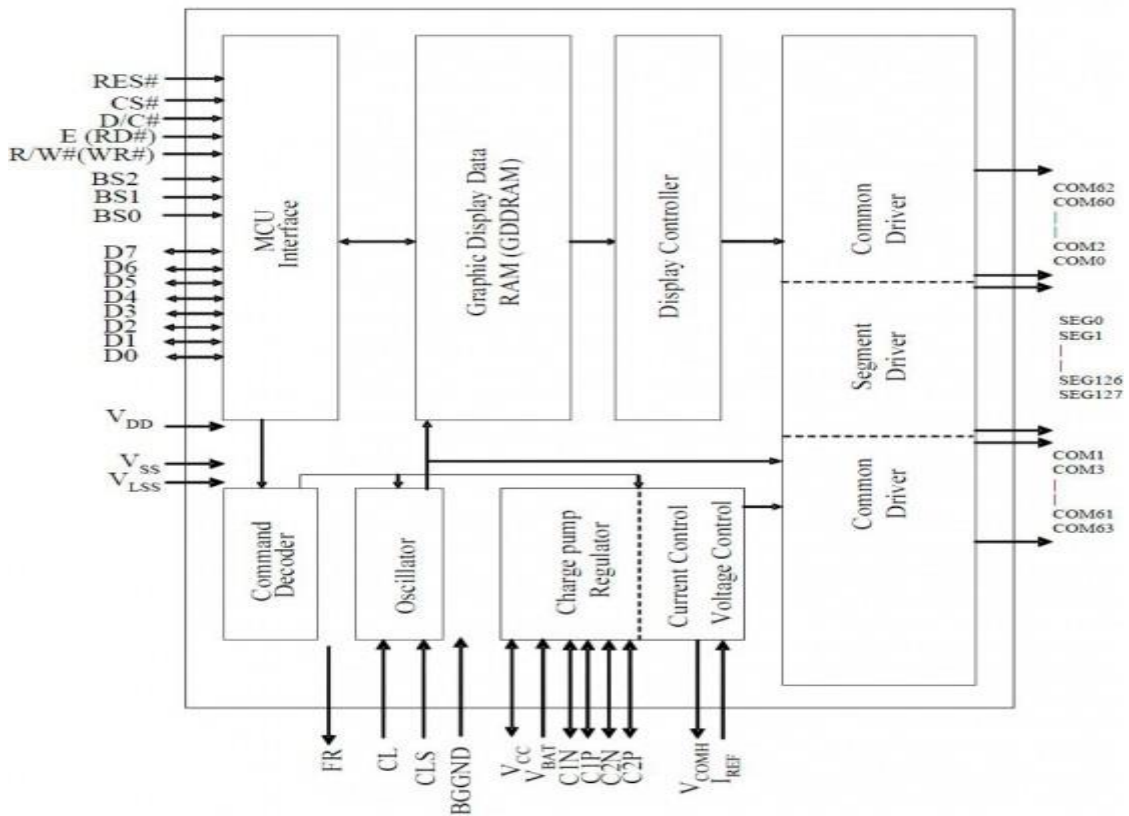


Рисунок 3.11 — Блок схема драйвера

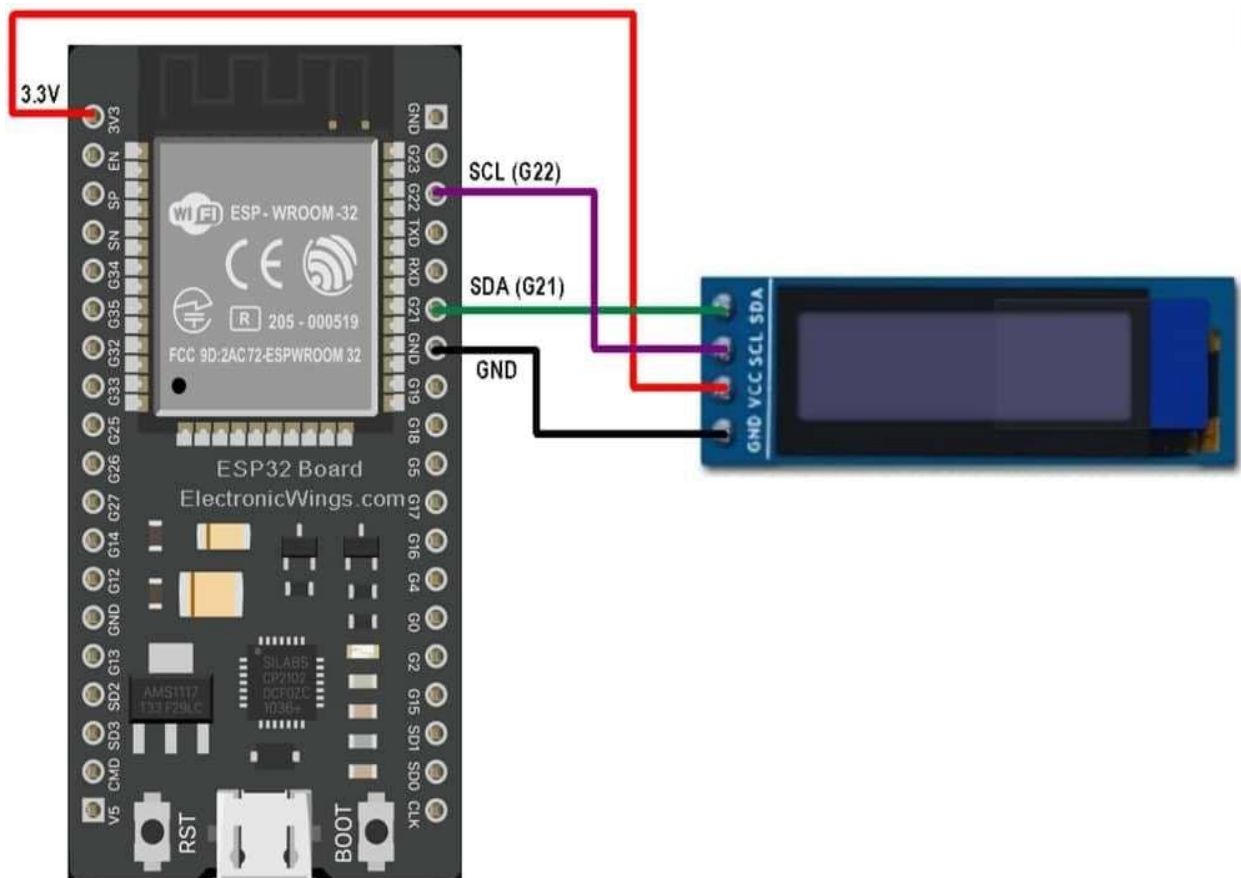


Рисунок 3.12 — Схема підключення екрану до esp 32

3.1.5 Плата ESP32 DevKit v1 на базі мікроконтролера ESP-WROOM-32

ESP32 DevKit v1 — це платформа на основі мікроконтролера ESP32-WROOM, розроблена компанією Espressif Systems. Вона призначена для створення різноманітних IoT-проектів та інтеграції у системи, що вимагають бездротового зв'язку, таких як автоматизація, метеостанції чи портативні пристрої. Основою є потужний чіп ESP32-D0WDQ6, який об'єднує можливості Wi-Fi та Bluetooth у компактному модулі.

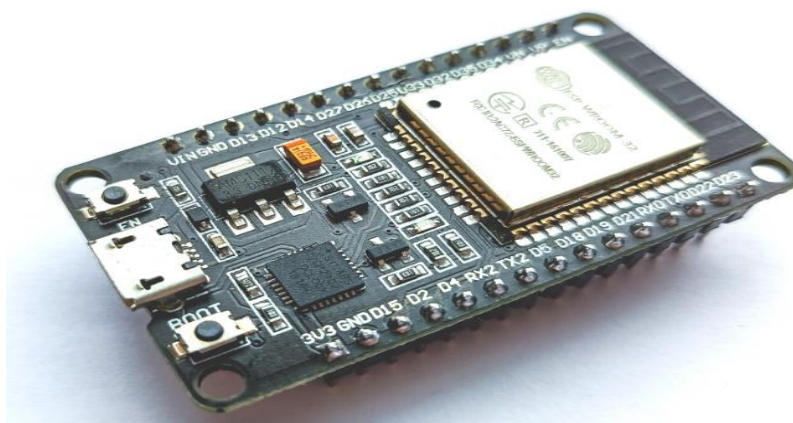


Рисунок 3.13 — ESP32 DevKit v1

Модуль ESP32-WROOM має вбудовану Flash-пам'ять обсягом 4 МБ і всі необхідні компоненти, зібрані під металевим екраном. Металевий корпус виконує роль клітки Фарадея, що мінімізує електромагнітні завади і підвищує стабільність роботи. Okремо передбачена мініатюрна друкована антена, виконана у формі доріжки-змійки, яка забезпечує ефективну передачу сигналу на частоті 2,4 ГГц.

Для підключення до комп'ютера використовується USB-UART перетворювач на базі мікросхеми CP2102. Він дозволяє взаємодіяти з мікроконтролером через віртуальний COM-порт. Всі операції, включаючи прошивку і налагодження, здійснюються через роз'єм micro-USB, який також забезпечує живлення плати.

На платі присутній світлодіодний індикатор живлення, а також додатковий користувацький світлодіод, який підключений до другого GPIO.

Його стан можна змінювати програмно шляхом встановлення високого або низького логічного рівня. Для ручного скидання системи використовується кнопка EN (аналог кнопки RESET), а кнопка BOOT дозволяє перевести модуль у режим прошивки.

Живлення плати забезпечується лінійним регулятором напруги AMS1117-3.3, який видає стабільні 3,3 В при струмі до 1 А. Вхідна напруга на пін Vin може бути у межах 5–14 В, що дозволяє використовувати широкий спектр джерел живлення.

Платформа має широкі можливості для взаємодії із зовнішніми пристроями завдяки наявності 25 пінів введення-виводу, з яких 15 підтримують роботу з АЦП розрядністю 12 біт, а 2 працюють як ЦАП із розрядністю 8 біт. Крім того, є 21 пін із підтримкою ШІМ (широкоімпульсної модуляції) із 16-бітною розрядністю. Апаратно доступні 3 інтерфейси SPI, 3 UART, 2 I²C і 2 I²S, що дозволяє використовувати плату в різноманітних проектах, де потрібна комунікація з датчиками, дисплеями, сервоприводами або іншими модулями.

Завдяки потужному ядру Tensilica Xtensa LX6 із максимальною тактовою частотою до 240 МГц, ESP32 DevKit v1 забезпечує високу обчислювальну здатність. Чіп підтримує сучасні протоколи бездротового зв'язку Wi-Fi стандарту 802.11b/g/n і Bluetooth BLE 4.2, що дозволяє створювати розумні пристрої з мінімальними витратами. Невеликі габарити плати — 51×28 мм — дозволяють використовувати її в компактних корпусах.

Платформа має розширені можливості, зокрема підтримку роботи в діапазоні температур від -40°C до +85°C, що робить її придатною для промислового використання. Завдяки своїм технічним характеристикам, низькій ціні та широкій функціональності, ESP32 DevKit v1 є відмінним вибором для будь-яких IoT-проектів.

ESP32 DEVKIT V1 – DOIT version with 36 GPIOs

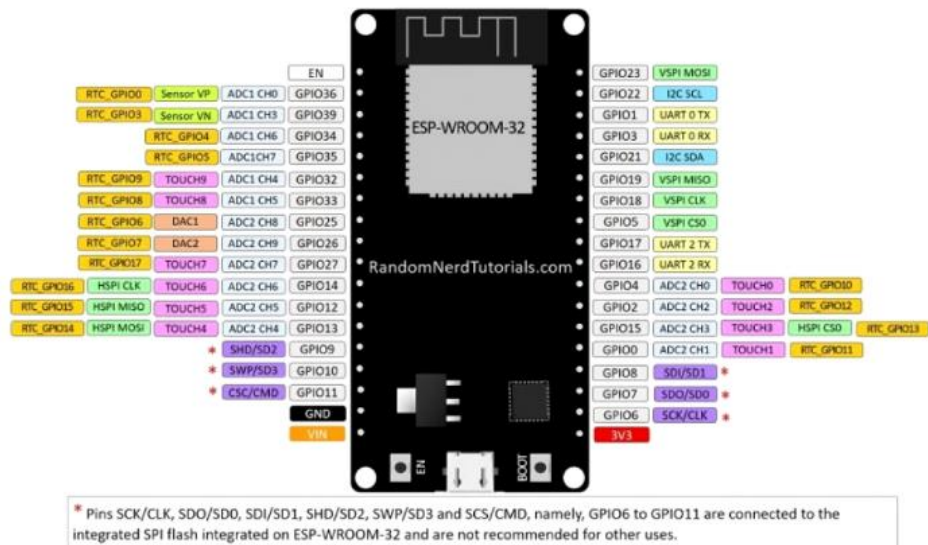


Рисунок 3.12 — Піни esp 32

3.2 Приклад реалізації проекту

3.2.1 Опис пристрою

Мною було розроблене апаратне забезпечення та написаний код програми, який передає отримані данні з датчиків, на mqtt сервер. Звідти програма Node-Red забирає данні: температури, тиску, вологості та якості повітря, та дає можливість реалізувати візуалізацію отриманої інформації в виді графіків, дозволяє застосувати метод експоненційного згладжування яке прибирає випадкові шуми, та метод лінійної регресія, що дозволяє спрогнозувати концентрацію CO2. Також Node-Red дає можливість віддалено контролювати показники та отримувати цю інформацію через Телеграм-бота, робота якого буде наведена, далі. Що в свою чергу, розширює можливості для проекту IoT моніторингу якості повітря.

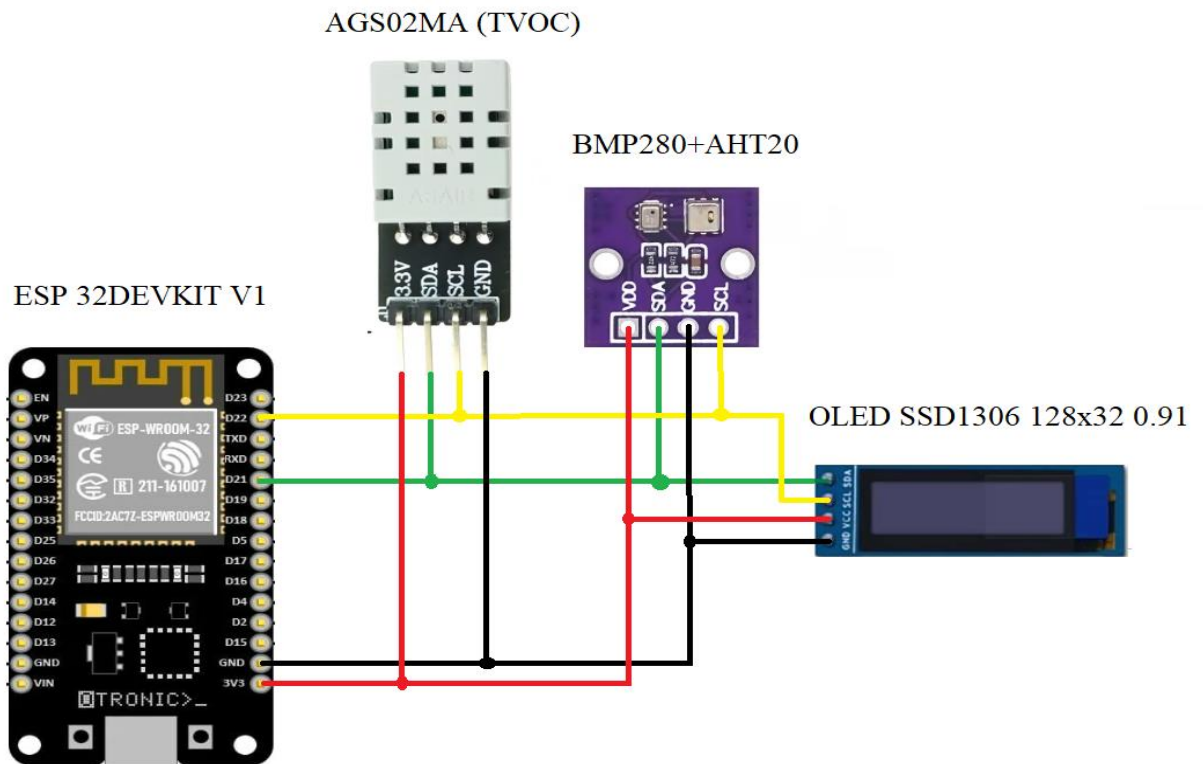


Рисунок 3.13 — Схема пристрою моніторингу повітря

Схема може бути зібрана я безпосереднім підєднанням дротів до модулів, і плати, а можна використати макетну плату, як зробив я. Живлення плати може бути як від micro-usb через ПК, чи Power Bank, так і від джерела живлення 5V 800mA, можна використовувати акумулятор 18650, з контролером заряду, та перетворювачем струму, якщо джерело живлення більше 5V. Живлення плати під'єднують через контакти VIN та GND.

ESP32 має вбудовану підтримку I2C, і ви можете призначити будь-які GPIO як SDA та SCL (за замовчуванням SDA — GPIO21, SCL — GPIO22).

I2C (Inter-Integrated Circuit) — це двопровідний протокол для передачі даних між мікроконтролером і периферійними пристроями. У ньому використовуються SDA (Serial Data) для передачі даних та SCL (Serial Clock) для передачі тактових сигналів для синхронізації.

Важливо! BMP280 працює на 3.3V. Якщо ви живите його від 5V, перевірте, чи ваш модуль датчика має вбудований регулятор напруги. Якщо модуль не має вбудованих pull-up резисторів, додайте зовнішні резистори (10

кОм) між SDA і VCC, а також між SCL і VCC.

OLED дисплей підключається за тим же принципом що і датчики по цифровому інтерфейсу, та слугує первинним інтерфейсом, для виводу інформації з датчиків, та розуміння етапу загрузки плати.

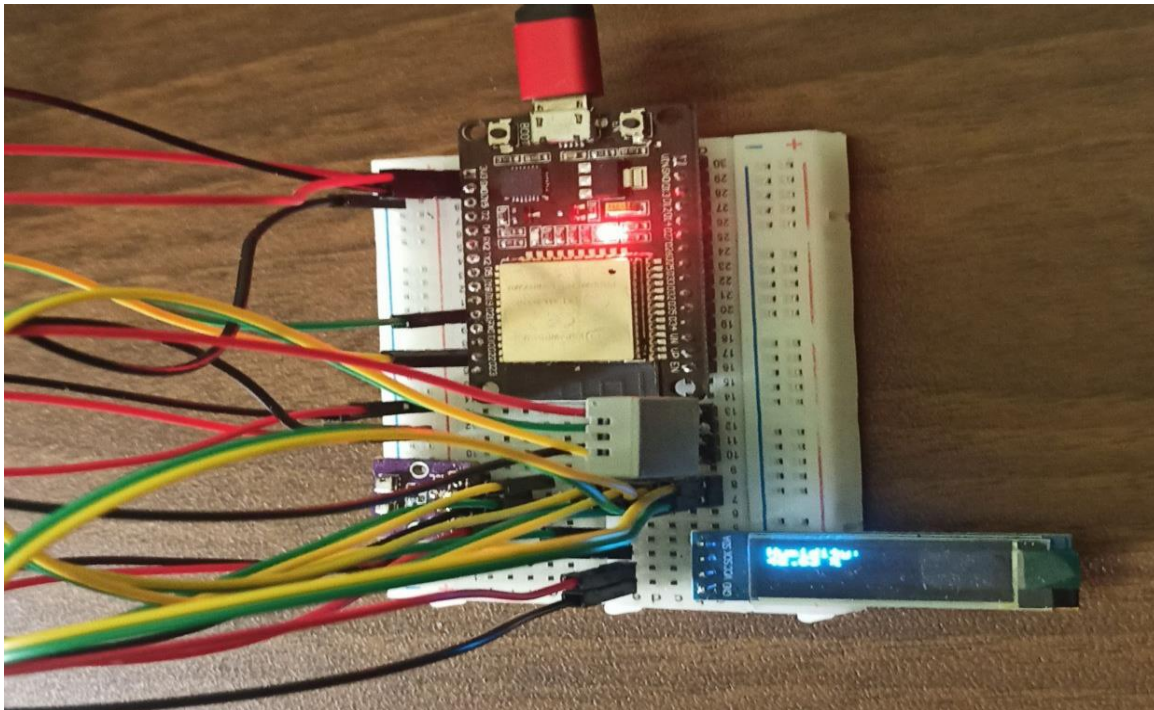


Рисунок 3.14 — Фізично зібраний пристрій

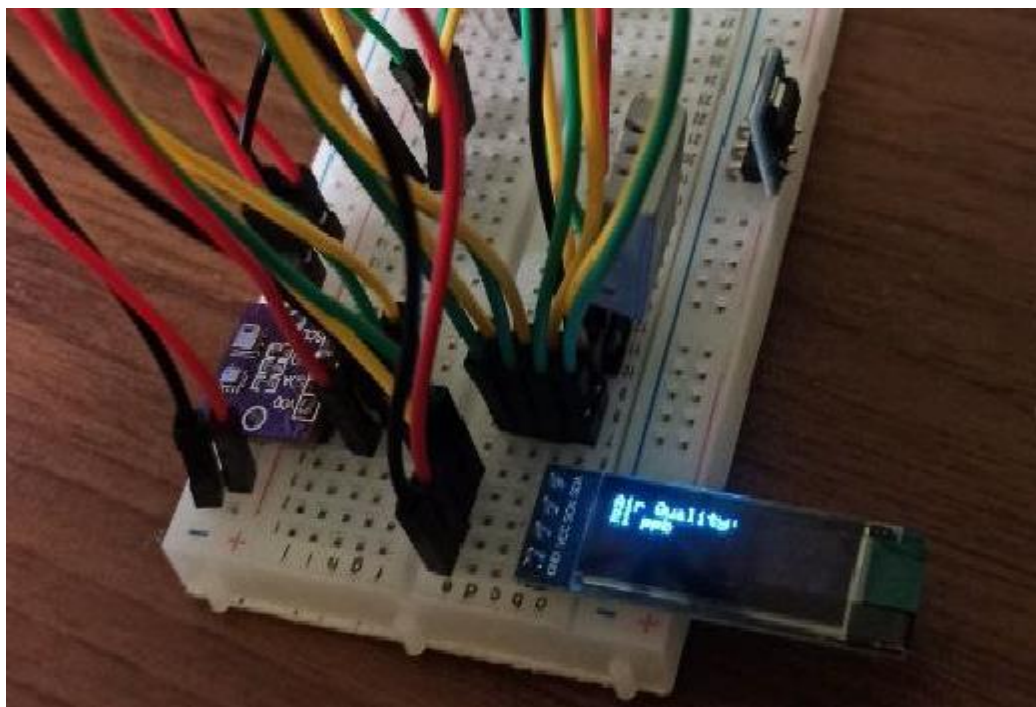


Рисунок 3.15 — Приклад роботи дисплея та датчиків

Виведення на OLED-дисплей даних про параметри середовища які циклічно відображаються:

Температура.

Вологість.

Тиск.

Якість повітря (VOC).

Приклад коду для виводу:

```
display.clearDisplay();
```

```
display.setCursor(0, 0);
```

```
display.println("Temperature: " + String(temp.temperature, 2) + " C");
```

```
display.display();
```

Відправка даних на MQTT, ESP32 публікує дані у форматі JSON на брокер MQTT

```
String payload = "{\"temperature\": " + String(temp.temperature) +  
    ", \"humidity\": " + String(humidity.relative_humidity) +  
    ", \"pressure\": " + String(pressure) +  
    ", \"air_quality_ppb\": " + String(airQualityPPB) + " }";
```

```
client.publish("esp32/sensorData", payload.c_str());
```

3.2.2 Принципова схема

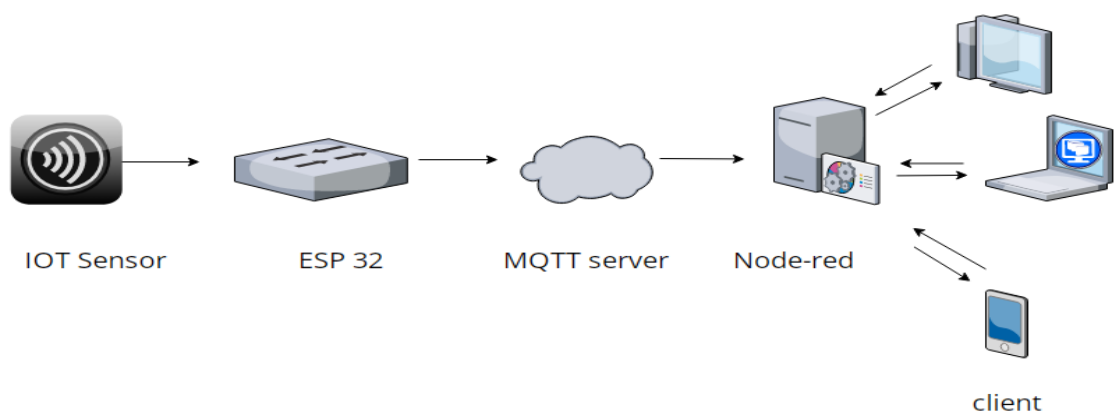


Рисунок 3.16 — Принципова схема реалізації системи

Містить такі елементи:

Сенсори якості повітря (температура, вологість, тиск, TVOC).

Мікроконтроллер ESP32 DevKit v1 з Wi-Fi та Bluetooth модулем для передачі даних.

MQTT сервер для виводу даних з датчиків

Сервер Node-Red для візуалізації отриманих даних та веб інтерфейсу

Веб-інтерфейс для відображення даних телеграм-боту

3.2.3 Робота з платою ESP32:

Плата ESP32 може бути запрограмована за допомогою програмного забезпечення Arduino IDE. Це зручне середовище програмування, яке дозволяє працювати на платформі ESP32 з підтримкою потрібних бібліотек.

Інтегроване середовище розробки Arduino (IDE)-це кросплатформна програма, написана на Java, похідна від IDE, для мови програмування Processing та проектів Wiring. Щоб використовувати ESP32 з Arduino IDE, вам потрібно додати відповідний менеджер плат.

ESP32 має вбудований інсталятор, який дозволяє завантажувати новий код без зовнішнього програматора, програмування здійснюється через інтерфейс USB з використанням протоколу UART. Більшість карт ESP32 підтримують автоматичне перемикання в режим завантаження, що спрощує процес завантаження коду.

Мікроконтролер налаштований на отримання програми завантажувача під час запуску системи управління. Якщо нової програми немає, програма установки запустить раніше збережену програму.

Функція `main()` є фактичною точкою входу в програму і викликається першою.

`void setup()` – функція `setup()` викликається, коли стартує скетч.

Наприклад, вона використовується для ініціалізації бібліотек, які будуть використовуватися. Функція налаштування виконується лише один раз після увімкнення або перезавантаження плати Arduino.

```
sketch_dec2a.ino
1  #include <Wire.h>
2  #include <WiFi.h>
3  #include <PubSubClient.h>
4  #include <Adafruit_GFX.h>
5  #include <Adafruit_SSD1306.h>
6  #include <Adafruit_AHTX0.h>
7  #include <Adafruit_BMP280.h>
8  #include "AGS02MA.h"
9
10 // Налаштування Wi-Fi
11 const char* ssid = "TOTOLINK_N150RT"; // Ім'я Wi-Fi мережі
12 const char* password = ""; // Пароль Wi-Fi мережі
13
14 // Налаштування MQTT
15 const char* mqttServer = "test.mosquitto.org"; // Адреса MQTT-сервера
16 const int mqttPort = 1883; // Порт MQTT
17 const char* mqttTopic = "esp32/sensorData"; // MQTT-тема для публікації даних
18
19 WiFiClient espClient;
20 PubSubClient mqttClient(espClient);
21
22 // Налаштування дисплея
23 #define SCREEN_WIDTH 128
24 #define SCREEN_HEIGHT 32
25 #define OLED_ADDRESS 0x3C
26 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
27
28 // Налаштування пінів світлодіодів
29 #define HEATER_LED 32 // Червоний світлодіод (обігрівач)
30 #define AC_LED 33 // Синій світлодіод (кондиціонер)
31 #define FAN_LED 25 // Зелений світлодіод (вентиляція)
32
33 // Датчики
34 Adafruit_AHTX0 aht; // Датчик температури та вологості
35 Adafruit_BMP280 bmp; // Датчик атмосферного тиску
36 AGS02MA ags02ma(0x1A, &Wire); // Датчик якості повітря
37
38 // Шляхи визначення параметрів сенсору
```

Рисунок 3.17 — Програмне забезпечення Arduino IDE

Програмування плати полягає в тому щоб правильно та послідовно ініціалізувати усі потрібні компоненти, бібліотеки які знадобляться в процесі роботи та компіляції програми, та налаштувати їх, в коді програми видно (Додаток А) що спочатку оголошуються бібліотеки, постійні змінні, та пристрої, налаштовується підключення до WI-FI та підключення до Mqtt брокера, для передачі інформації.

Через Mqtt Explorer ми бачимо що наші данні, окрім того що відображаються на дисплеї, виводяться і в інтернет, та можна побачити первинну інформацію вже на комп'ютері, та далі займатися обробкою цих даних.

Наші данні доступні на test.mosquitto.org порт 1883, це безкоштовний сервер для виведення інформації з різних IoT пристр

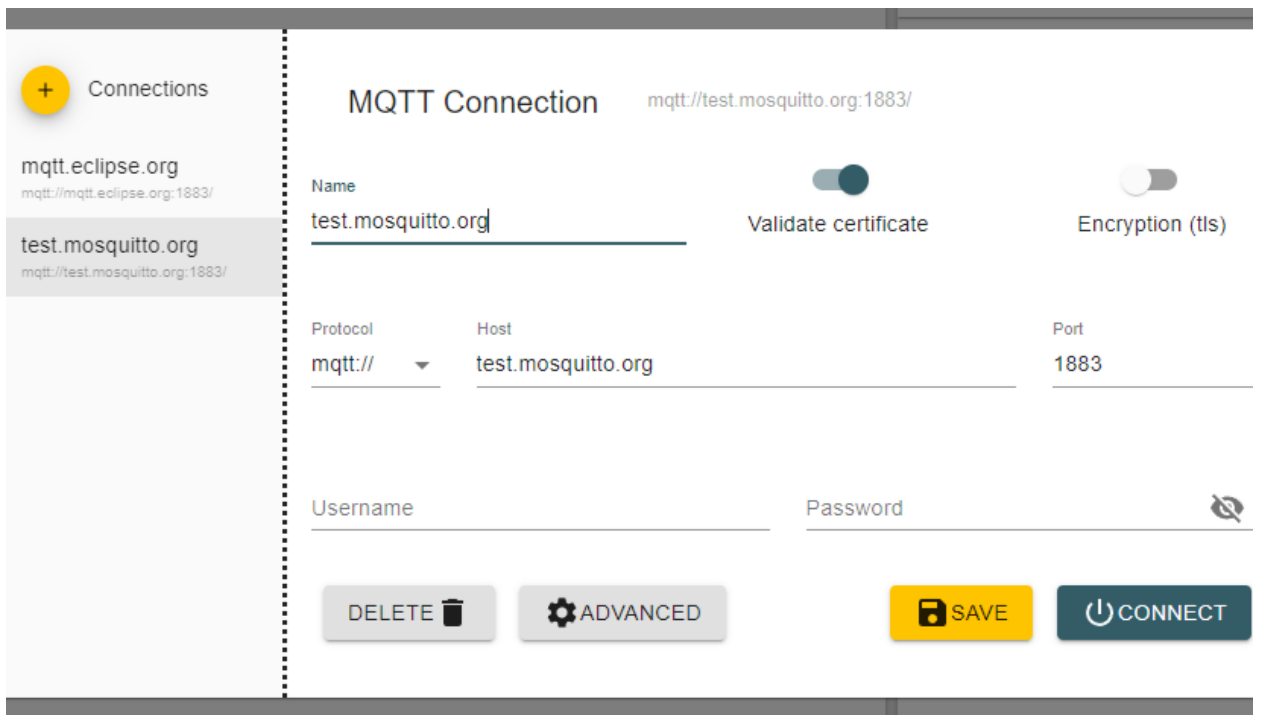


Рисунок 3.18 — Стартова сторінка Mqtt Explorer

А тут ми вже за шляхом esp32/sensorData ми можемо побачити наші данні, та первинні графіки.

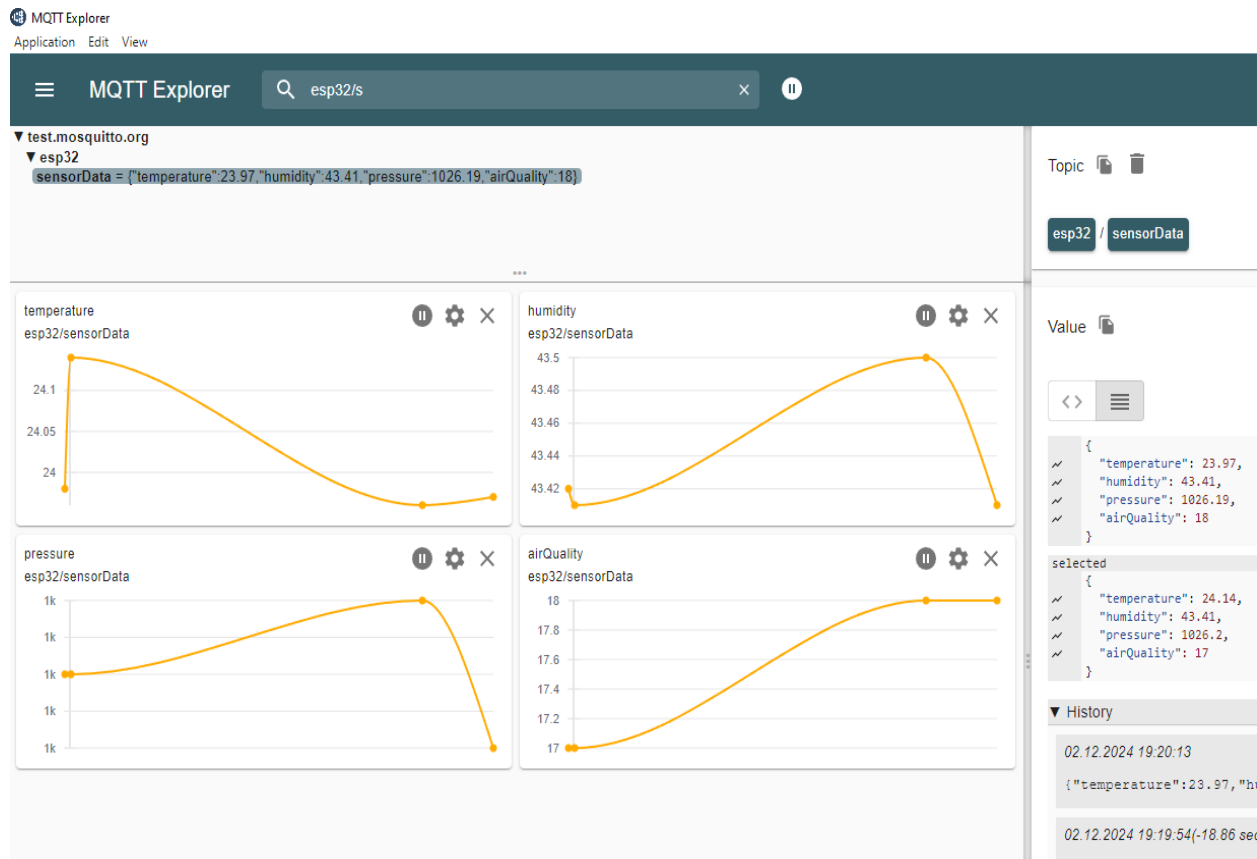


Рисунок 3.19 — Головна сторінка MQTT серверу

3.2.3 Робота з Node red та Телеграм ботом.

Наступним кроком є налаштування Node-Red першим є блок збереження який отримує та зберігає данні.

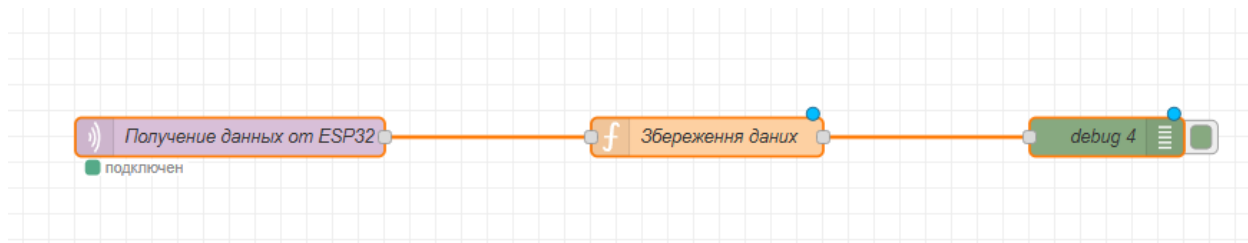


Рисунок 3.20 — Блок збереження інформації.

Имя: Збереження даних

Setup | Настройка | **Функция** | Закрытие

```
1 flow.set("temperature", msg.payload.temperature);
2 flow.set("humidity", msg.payload.humidity);
3 flow.set("pressure", msg.payload.pressure);
4 flow.set("air_quality", msg.payload.airQuality);
5 return msg;
```

Рисунок 3.21 — Код функції збереження даних

В цьому блоці данні які надійшли з контролеру через mqtt брокер збережені в функції, та виведені на графік через наступні функції.

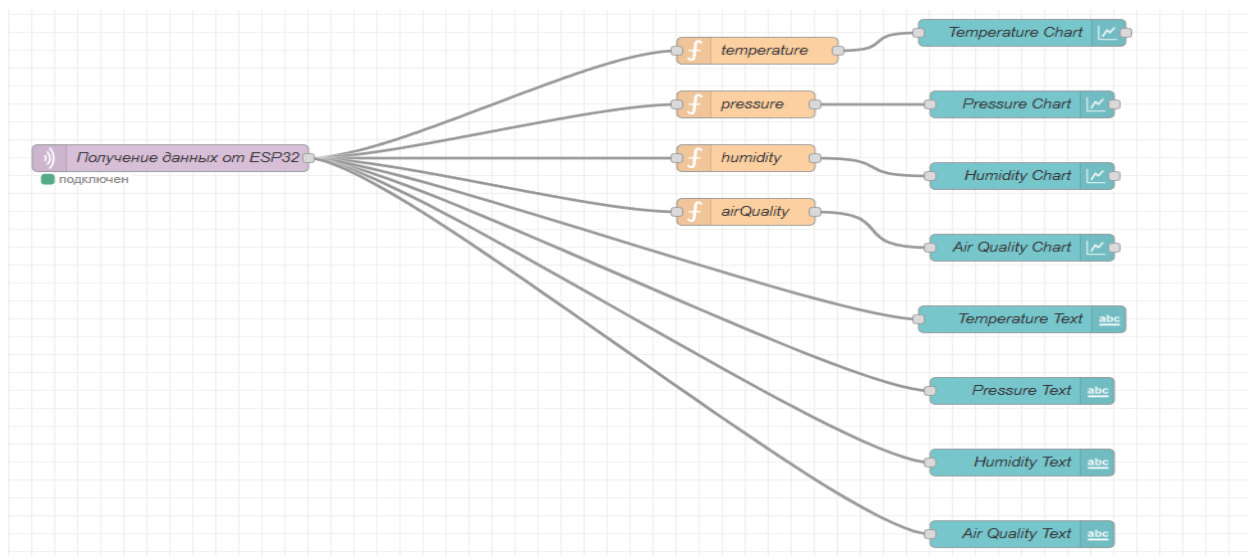


Рисунок 3.22 — Блок візуалізації отриманих даних



Рисунок 3.23 — Сторінка візуалізації отриманих даних в Node-Red

Блок реалізації основних математичних функції та налаштування telegram-bota та його взаємодія з Node-red дозволяє отримувати відповіді на усі реалізовані функції в телеграм, що в свою чергу є реалізацією віддаленого моніторингу за допомогою IoT.

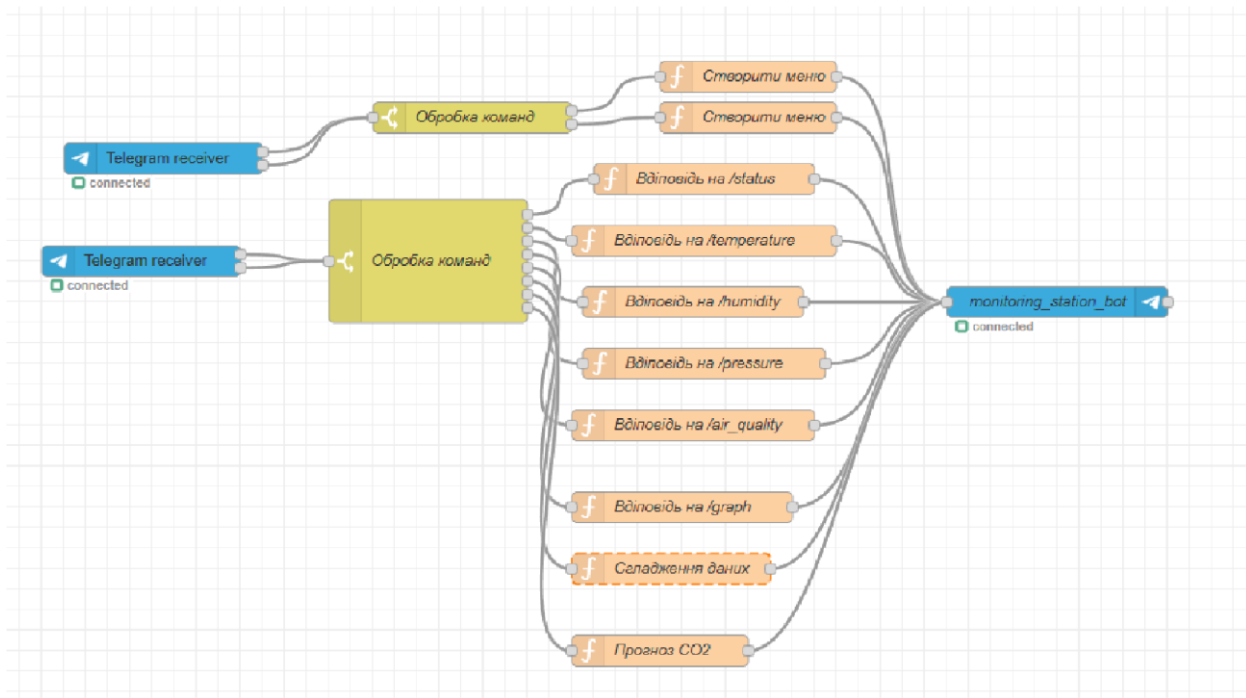


Рисунок 3.24 — Блок реалізації основних математичних функції та налаштування telegram-bota та його взаємодія з Node-red

Свойства

Имя: Відповідь на /status

Setup Настройка **Функция** Закрытие

```

1  msg.payload = {}
2    chatId: msg.payload.chatId,
3    type: "message",
4    content: `Загальний статус:
5  Температура: ${flow.get("temperature") || "данні відсутні"}
6  Вологість: ${flow.get("humidity") || "данні відсутні"} %
7  Тиск: ${flow.get("pressure") || "данні відсутні"} гПа
8  Якість повітря: ${flow.get("air_quality") || "данні відсутні"}
9  `;
10 return msg;

```

Рисунок 3.25 — Код реалізації однієї з команд телеграм-бота

Доступ до телеграм-бота можна отримати, ввівши в пошуку телеграму: @Monitoring_station_bot, зайти в чат с ботом та нажати старт, або ввести команду /menu, для виводу основних команд бота.

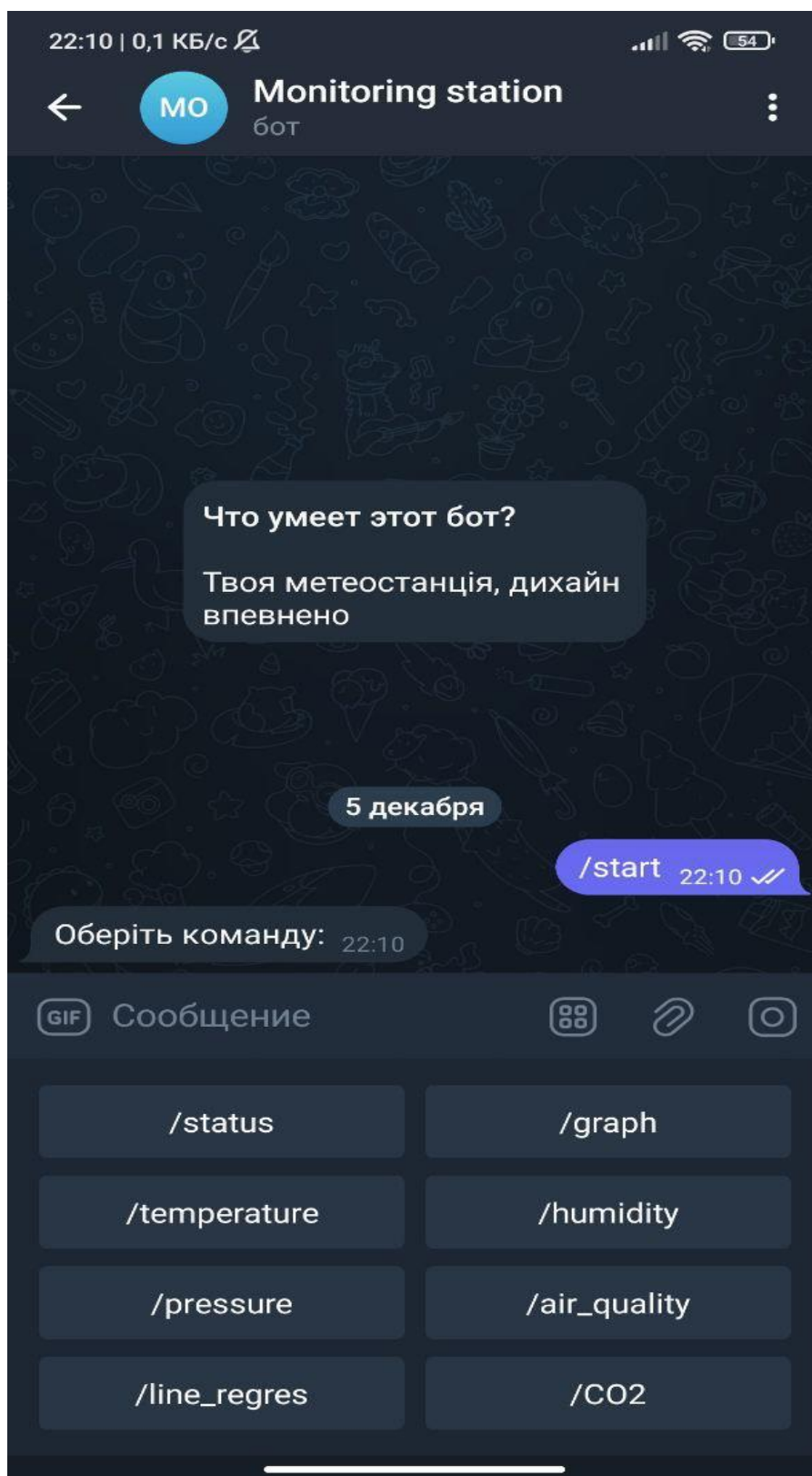


Рисунок 3.26 — Стартова сторінка бота

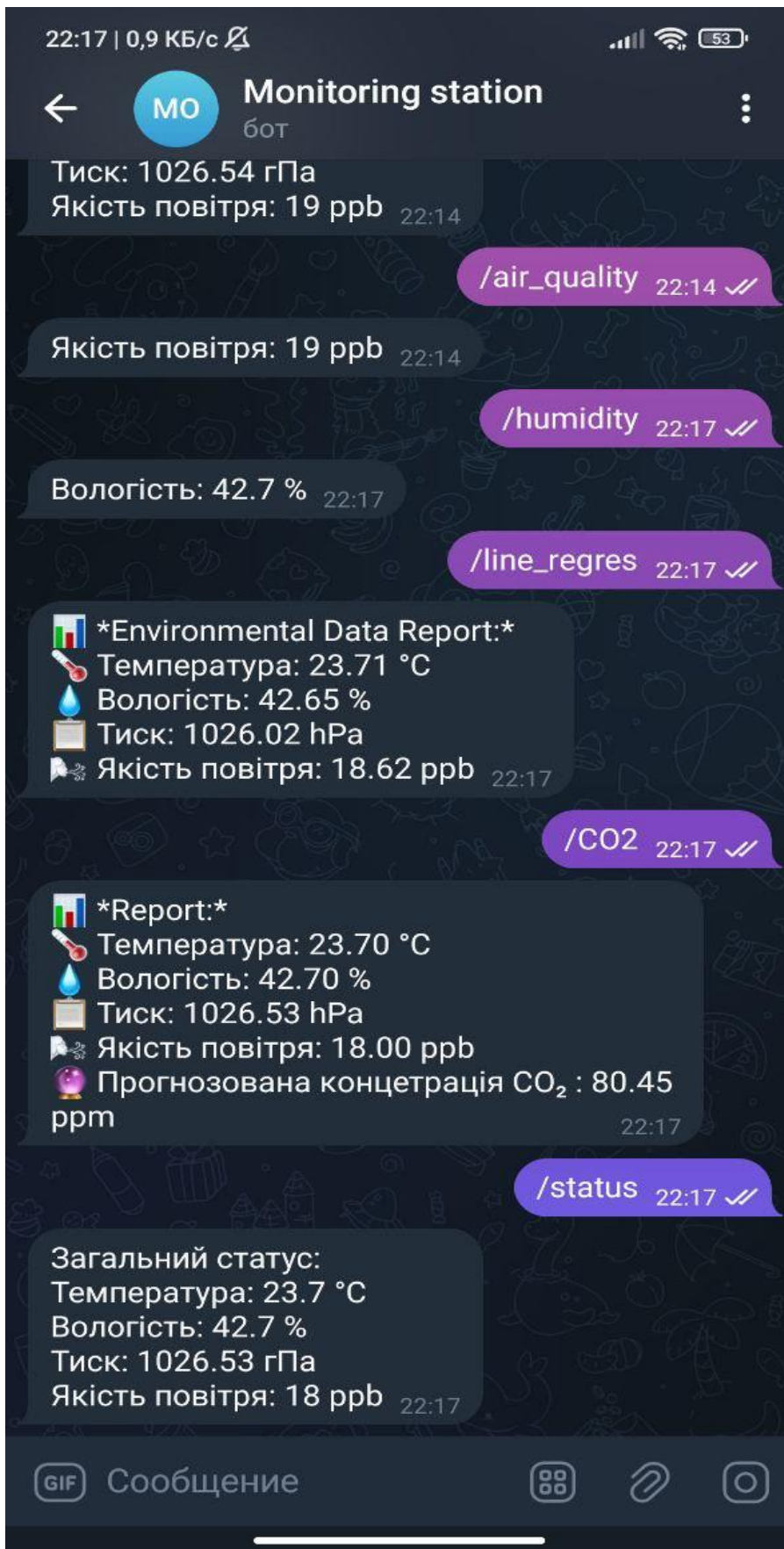


Рисунок 3.12 — Вивід інформації статусу, сгладжування, та прогнозованого забруднення CO₂

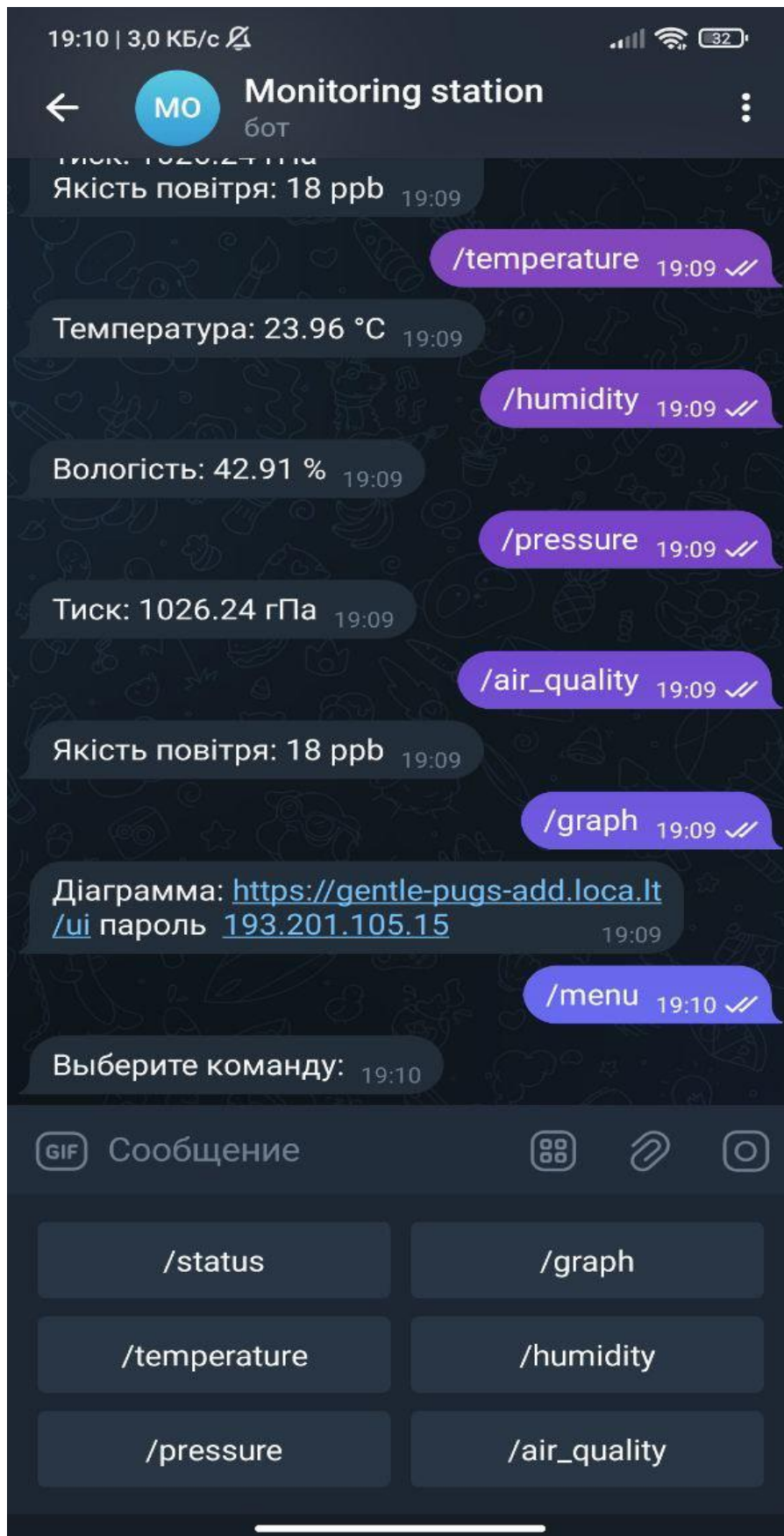


Рисунок 3.28 — Відповіді на основні команди бота

Висновки до розділу

У третьому розділі було реалізовано розробку програмного забезпечення та технічне впровадження системи моніторингу якості повітря на базі мікроконтролера ESP32. У ході роботи:

Розроблено архітектуру системи, що включає датчики забруднення повітря, контролер ESP32, дисплей для локального відображення інформації та хмарну платформу для збору та аналізу даних.

Реалізовано програмний код, що забезпечує взаємодію між усіма компонентами системи, а також передачу даних у хмарне середовище.

Проведено тестування системи в реальних умовах, яке підтвердило її ефективність у вимірюванні рівня забруднення повітря, передачі даних у режимі реального часу та забезпеченні доступу до них у зручному для користувача форматі.

ВИСНОВКИ

Під час виконання магістерської роботи розробки система дистанційного моніторингу якості повітря з використанням технологій інтернет речей були розглянуті:

У першому розділі сучасні підходи до створення систем моніторингу якості повітря з використанням технологій Інтернету речей. Проведено огляд існуючих рішень, їх плюсів і мінусів.

Проведено аналіз систем, що діють в Україні та країнах ЄС. При цьому враховуються критерії оцінки якості повітря, а також основні види забруднюючих речовин, такі як тверді частинки, гази і леткі органічні сполуки. Був проведений аналіз впливу цих речовин на здоров'я людини і екологію.

Також розглядаються основні компоненти системи моніторингу: датчики, контролери, модулі передачі даних, хмарна платформа для аналізу інформації. Висвітлюються сучасні тенденції, такі як інтеграція з хмарними сервісами, використання бездротових технологій і мінімізація витрат на обладнання.

У другому розділі було розглянуто математичний аналіз експоненціального згладжування та лінійної регресії. Цей розділ присвячений методам обробки даних для підвищення точності вимірювань.

Метод експоненціального згладжування може бути використаний для усунення випадкових коливань показників, одержуваних датчиком, і підвищення стабільності даних. Це важливо для аналізу якості повітря, так як показники можуть змінюватися через вплив погодних умов і технічних помилок.

Лінійна регресія використовується для встановлення взаємозв'язків з різними факторами, такими як рівень забруднення та погодні умови. Це дозволяє прогнозувати зміни якості повітря за певних умов, що робить систему моніторингу більш адаптованою.

У третьому розділі було розроблено програмне забезпечення та технічна

реалізація системи, описано практичну реалізацію системи моніторингу на базі мікроконтролера ESP32.

Система складається з датчиків, які визначають рівень забруднення повітря, контролера ESP32, який збирає дані, і дисплея, на якому відображається інформація. Дані також передаються на хмарну платформу для виведення та аналізу.

В ході впровадження був створений програмний код, налаштовано взаємодію між компонентами і проведено тестування системи в реальних умовах. Випробування показали, що система може ефективно вимірювати рівень забруднення, передавати дані в режимі реального часу та аналізувати їх у зручному форматі.

Система готова до використання і може бути легко адаптована для моніторингу атмосферного повітря в різних умовах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kalman R. 'A New Approach to Linear Filtering and Prediction Problems.' Journal of Basic Engineering, 1960. (дата звернення: 17.11.24)
2. Welch G., Bishop G. 'An Introduction to the Kalman Filter.' Technical Report, UNC-Chapel Hill, 2006. (дата звернення: 17.11.24)
3. Espressif Systems: 'ESP32 Technical Reference Manual. (дата звернення: 17.11.24)
4. В. В. Шликов Мікропроцесорна техніка: Практикум: навч. посіб. для студ. спеціальності 163 «Біомедична інженерія» та 152 «Метрологія та інформаційно- вимірювальна техніка»/ В. В. Шликов; КПІ ім. Ігоря Сікорського.– Київ : КПІ ім. Ігоря Сікорського, 2018. – 144 с. (Електронний ресурс) Режим доступу до ресурсу :https://drive.google.com/file/d/117AvmKwxZe37hTTO_j_NaN6ej02O-FyUX/view?usp=sharing (дата звернення: 17.11.24)
5. Загальні санітарно-гігієнічні вимоги до повітря робочої зони: ГОСТ 12.1.005-88. Режим доступу до ресурсу: <https://drive.google.com/file/d/1t9J2jokxwSFFxIWBvkzwZiLvfeUZ74-Z/view?usp=sharing> (дата звернення: 17.11.24)
6. Яковенко В.В. Система збору, збереження та відображення інформації про навколишнє середовище/ Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»/кер. доц. к.ф.м. наук_Заворотний В.Ф. Виконавець: Яковенко В.В: Київ : КПІ ім. Ігоря Сікорського, 2019. – 65с. (дата звернення: 17.11.24)
7. Welcome to fritzing. Welcome to Fritzing. URL: <https://fritzing.org/> (дата звернення: 17.11.24).
8. Data sheet CCS811 Ultra-Low Power Digital Gas Sensor for Monitoring Indoor Air Quality (Електронний ресурс) (дата звернення: 17.11.24)
9. Internet of things, iot. IT-Enterprise – your one-stop ecosystem for reengineering | it.ua. URL: <https://www.it.ua/knowledge-base/technology->

innovation/internet-veschej-internet-of-things-iot (дата звернення: 17.11.24)

10. IoT analytics - thingspeak internet of things. IoT Analytics - ThingSpeak Internet of Things. URL: <https://thingspeak.com/> (дата звернення: 17.11.24).

11. S. Srilatha, R. Uma, A. R. Pallavi, and S. Koppu, —Quantitative performance analysis of face recognition system,|| in Proceedings - International Conference on Trends in Electronics and Informatics, ICEI 2017, Feb. 2018, vol. 2018-January, pp. 1143–1146, doi: 10.1109/ICOEI.2017.8300892. (дата звернення: 17.11.24).

12. G. S. Hsu and C. H. Tang, — health monitoring in Ukraine,|| IEEE Access, vol. 8, pp. 147765–147775, 2020, doi: 10.1109/ACCESS.2020.3014877.

13. <https://okna.ua/ua/library/vymohy-do-yakosti-povitrya-staly-bilsh> (дата звернення: 17.11.24)

14. Маринич І. А., Тронь В. В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2022. 50с.

15. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).

16. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).

17. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація)

18. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення Київ, Держстандарт України, 1998. 27 с.

(Інформація та документація).

19. Горбенко, Л.В. Інтернет речей: концепції, архітектури та технології / Л.В. Горбенко. – Харків: ХНУРЕ, 2020. – 320 с. (Дата звернення: 17.11.2024).

20. Коваленко, С.В. Інформаційні системи екологічного моніторингу / С.В. Коваленко. – К.: Наук. думка, 2019. – 256 с. (Дата звернення: 17.11.2024).

21. Чорний, М.О. Системи моніторингу якості повітря: аналітичний огляд / М.О. Чорний // Екологія та техніка. – 2021. – №3. – С. 34–41. (Дата звернення: 17.11.2024).

22. Новиков, В.О. Інтернет речей та його застосування в моніторингу навколишнього середовища / В.О. Новиков. – Одеса: ОНУ, 2018. – 188 с. (Дата звернення: 17.11.2024).

23. Ivanov, P. Development of IoT-based air quality monitoring systems // Proceedings of the International Conference on Environmental Monitoring. – 2020. – P. 12–19. (Дата звернення: 17.11.2024).

24. ThingSpeak Documentation. IoT Analytics Platform [Електронний ресурс]. – Режим доступу: <https://thingspeak.com/docs> (Дата звернення: 17.11.2024).

25. Blynk Application for IoT. Features and Capabilities [Електронний ресурс]. – Режим доступу: <https://blynk.io> (Дата звернення: 17.11.2024).

26. Комп'ютерна система для дистанційного контролю якості атмосферного повітря // Вісник ТНТУ. – 2020. – №4. – С. 45–52 [Електронний ресурс]. – Режим доступу: <https://elartu.tntu.edu.ua> (Дата звернення: 17.11.2024).

27. Система моніторингу якості повітря на базі технологій IoT // Наукові записки НУВГП. – 2021. – №1. – С. 29–34 [Електронний ресурс]. – Режим доступу: <https://itconfdoc.nuwm.edu.ua> (Дата звернення: 17.11.2024).

28. Computerized System for Monitoring Dust Levels // Вісник ТНТУ. – 2021. – №5. – С. 38–44 [Електронний ресурс]. – Режим доступу: <https://elartu.tntu.edu.ua> (Дата звернення: 17.11.2024).

29. Monitoring of Air Pollution using IoT and Raspberry Pi // Lviv Polytechnic Journal. – 2022. – №3. – С. 55–61 [Електронний ресурс]. – Режим доступу: <https://science.lpnu.ua> (Дата звернення: 17.11.2024).
30. Системи моніторингу якості повітря: методи та алгоритми // Науковий вісник ВНТУ. – 2021. – №2. – С. 42–50 [Електронний ресурс]. – Режим доступу: <https://ir.lib.vntu.edu.ua> (Дата звернення: 17.11.2024).
31. Лінчук, Т.В. Математичні методи в екологічному моніторингу / Т.В. Лінчук. – Київ: КНУ, 2018. – 140 с. (Дата звернення: 17.11.2024).
32. Analysis of Air Quality Data Using Linear Regression and Smoothing Methods // Journal of Applied Environmental Studies. – 2019. – №7. – Р. 85–92. (Дата звернення: 17.11.2024).
33. IoT in Environmental Monitoring: A Case Study of Smart Sensors // International Journal of Smart Technologies. – 2020. – №4. – Р. 21–28. (Дата звернення: 17.11.2024).

Код C++ для Arduino IDE

```
1. #include <Wire.h>
2. #include <WiFi.h>
3. #include <PubSubClient.h>
4. #include <Adafruit_GFX.h>
5. #include <Adafruit_SSD1306.h>
6. #include <Adafruit_AHTX0.h>
7. #include <Adafruit_BMP280.h>
8. #include "AGS02MA.h"
9.
10.// Налаштування Wi-Fi
11.const char* ssid = "TOTOLINK_N150RT"; // Ім'я Wi-Fi мережі
12.const char* password = ""; // Пароль Wi-Fi мережі
13.
14.// Налаштування MQTT
15.const char* mqttServer = "test.mosquitto.org"; // Адреса MQTT-
сервера
16.const int mqttPort = 1883; // Порт MQTT
17.const char* mqttTopic = "esp32/sensorData"; // MQTT-тема для
публікації даних
18.
19.WiFiClient espClient;
20.PubSubClient mqttClient(espClient);
21.
22.// Налаштування дисплея
23.#define SCREEN_WIDTH 128
24.#define SCREEN_HEIGHT 32
25.#define OLED_ADDRESS 0x3C
```

```

26.Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, -1);
27.// Датчики
28.Adafruit_AHTX0 aht;           // Датчик температури та вологості
29.Adafruit_BMP280 bmp;         // Датчик атмосферного тиску
30.AGS02MA ags02ma(0x1A, &Wire); // Датчик якості повітря
31.
32.// Цільові значення параметрів середовища
33.float targetTemperature = 22.0; // Цільова температура (°C)
34.float targetHumidity = 50.0; // Цільова вологість (%)
35.float targetAirQuality = 800; // Цільова якість повітря (ppb)
36.
37.// Змінні для циклу
38.unsigned long lastUpdateTime = 0;
39.int displayState = 0;
40.
41.// Функція підключення до MQTT
42.void reconnectMQTT() {
43. while (!mqttClient.connected()) {
44.   Serial.print("Connecting to MQTT...");
45.   if (mqttClient.connect("ESP32Client")) {
46.     Serial.println(" connected!");
47.   } else {
48.     Serial.print(" failed, rc=");
49.     Serial.print(mqttClient.state());
50.     Serial.println(". Retrying in 5 seconds...");
51.     delay(5000);
52.   }
53. }
54.}

```



```

55.
56. void setup() {
57.   Serial.begin(115200);
58.
59.   // Налаштування дисплея
60.   if (!display.begin(0x3C, OLED_ADDRESS)) { // Используем адрес
I2C напрямую
61.     Serial.println("SSD1306 allocation failed");
62.     while (true);
63.   }
64.   display.clearDisplay();
65.   display.setTextSize(1);
66.   display.setTextColor(SSD1306_WHITE);
67.
68.   // Підключення до Wi-Fi
69.   display.println("Connecting to Wi-Fi...");
70.   display.display();
71.   WiFi.begin(ssid, password);
72.   while (WiFi.status() != WL_CONNECTED) {
73.     delay(500);
74.     Serial.print(".");
75.   }
76.   Serial.println(" Wi-Fi connected!");
77.   display.clearDisplay();
78.   display.println("Wi-Fi connected!");
79.   display.println(WiFi.localIP());
80.   display.display();
81.
82.   // Налаштування MQTT
83.   mqttClient.setServer(mqttServer, mqttPort);

```

```

84.
85. // Ініціалізація датчиків
86. if (!aht.begin()) {
87.   Serial.println("АHT initialization failed!");
88. }
89. if (!bmp.begin(0x77)) {
90.   Serial.println("BMP280 initialization failed!");
91. }
92. if (!ags02ma.begin()) {
93.   Serial.println("AGS02MA initialization failed!");
94. }
95.}
96.
97.void loop() {
98. // Підключення до MQTT, якщо відключено
99. if (!mqttClient.connected()) {
100.     reconnectMQTT();
101.   }
102.
103.   // Зчитування даних із датчиків
104.   sensors_event_t humidityEvent, tempEvent;
105.   aht.getEvent(&humidityEvent, &tempEvent);
106.   float temperature = tempEvent.temperature;
107.   float humidity = humidityEvent.relative_humidity;
108.   float pressure = bmp.readPressure() / 100.0F; // Тиск у гПа
109.   uint32_t airQuality = ags02ma.readPPB(); // Якість повітря
у ppb
110.
111.   // Виведення даних у Serial Monitor
112.   Serial.println("--- Sensor Data ---");

```

```

113. Serial.printf("Temperature: %.2f °C\n", temperature);
114. Serial.printf("Humidity: %.2f %%\n", humidity);
115. Serial.printf("Pressure: %.2f hPa\n", pressure);
116. Serial.printf(" Air Quality: %u ppb\n", airQuality);
117. Serial.println("-----");
118. // Відправка даних до MQTT
119. String payload = String("{\"temperature\":") + temperature +
120.     "\",\"humidity\":") + humidity +
121.     "\",\"pressure\":") + pressure +
122.     "\",\"air_quality_ppb\":") + airQuality + "}";
123. if (mqttClient.publish(mqttTopic, payload.c_str())) {
124.     Serial.println("Data published to MQTT!");
125. } else {
126.     Serial.println("Failed to publish data to MQTT!");
127. }
128.
129. // Оновлення дисплея
130. if (millis() - lastUpdateTime > 2000) { // Інтервал 2 секунди
131.     lastUpdateTime = millis();
132.     display.clearDisplay();
133.     display.setCursor(0, 0);
134.
135.     switch (displayState) {
136.         case 0:
137.             display.println("Temperature:");
138.             display.println(String(temperature) + " °C");
139.             break;
140.         case 1:
141.             display.println("Humidity:");
142.             display.println(String(humidity) + " %");

```

```
143.         break;
144.         case 2:
145.             display.println("Pressure:");
146.             display.println(String(pressure) + " hPa");
147.             break;
148.         case 3:
149.             display.println("Air Quality:");
150.             display.println(String(airQuality) + " ppb");
151.             break;
152.     }
153.     display.display();
154.     displayState = (displayState + 1) % 4;
155. }
156.
157. mqttClient.loop();
158. delay(500); // Затримка для зниження частоти опитування
}
```

Json Node-Red

```
[
  {
    "id": "08b3c4155c049b4a",
    "type": "tab",
    "label": "Поток 5",
    "disabled": false,
    "info": "",
    "env": []
  },
  {
    "id": "ceca83cd4532060b",
    "type": "telegram sender",
    "z": "08b3c4155c049b4a",
    "name": "monitoring_station_bot",
    "bot": "5cc3c3b6a83274af",
    "haserroroutput": false,
    "outputs": 1,
    "x": 1020,
    "y": 1160,
    "wires": [
      []
    ]
  },
  {
    "id": "bb2f29804b027119",
    "type": "switch",
    "z": "08b3c4155c049b4a",
    "name": "Обробка команд",
```

```
"property": "payload.content",
"propertyType": "msg",
"rules": [
  {
    "t": "eq",
    "v": "/status",
    "vt": "str"
  },
  {
    "t": "eq",
    "v": "/temperature",
    "vt": "str"
  },
  {
    "t": "eq",
    "v": "/humidity",
    "vt": "str"
  },
  {
    "t": "eq",
    "v": "/pressure",
    "vt": "str"
  },
  {
    "t": "eq",
    "v": "/air_quality",
    "vt": "str"
  },
  {
    "t": "eq",
```

```
        "v": "/graph",
        "vt": "str"
    }
],
"checkall": "true",
"repair": false,
"outputs": 6,
"x": 450,
"y": 1120,
"wires": [
    [
        "e09d92839bee7c19"
    ],
    [
        "8e8eb5df88ae9764"
    ],
    [
        "7a9c940929e50ab9"
    ],
    [
        "de6c03aacddc94c1"
    ],
    [
        "febefe31d171c396"
    ],
    [
        "7cabb7e9170b6de9"
    ]
]
},
```

```

{
  "id": "e09d92839bee7c19",
  "type": "function",
  "z": "08b3c4155c049b4a",
  "name": "Вдiповiдь на /status",
  "func": "msg.payload = {\n  chatId: msg.payload.chatId,\n  type:\n  \"message\", \n  content: `Загальний статус:\nТемпература:\n${flow.get(\"temperature\") || \"даннi вiдсутнi\"} °C\nВологiсть:\n${flow.get(\"humidity\") || \"даннi вiдсутнi\"} %\nТиск: ${flow.get(\"pressure\")\n|| \"даннi вiдсутнi\"} гПа\nЯкiсть повітря: ${flow.get(\"air_quality\") || \"даннi\nвiдсутнi\"} ppb`\n};\nreturn msg;",
  "outputs": 1,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 700,
  "y": 1040,
  "wires": [
    [
      "сеса83cd4532060b"
    ]
  ]
},
{
  "id": "8e8eb5df88ae9764",
  "type": "function",
  "z": "08b3c4155c049b4a",
  "name": "Вдiповiдь на /temperature",

```



```

    "func": "msg.payload = {\n    chatId: msg.payload.chatId,\n    type:
\"message\", \n    content: `Температура: ${flow.get(\"temperature\")} || \"данні
відсутні\"} °C`\n};\nreturn msg;\",
    \"outputs\": 1,
    \"timeout\": \"\",
    \"noerr\": 0,
    \"initialize\": \"\",
    \"finalize\": \"\",
    \"libs\": [],
    \"x\": 700,
    \"y\": 1100,
    \"wires\": [
      [
        \"c5ca83cd4532060b\"
      ]
    ]
  },
  {
    \"id\": \"7a9c940929e50ab9\",
    \"type\": \"function\",
    \"z\": \"08b3c4155c049b4a\",
    \"name\": \"Відповідь на /humidity\",
    \"func\": \"msg.payload = {\n    chatId: msg.payload.chatId,\n    type:
\"message\", \n    content: `Вологість: ${flow.get(\"humidity\")} || \"данні
відсутні\"} %`\n};\nreturn msg;\",
    \"outputs\": 1,
    \"timeout\": \"\",
    \"noerr\": 0,
    \"initialize\": \"\",
    \"finalize\": \"\",

```

```

    "libs": [],
    "x": 690,
    "y": 1160,
    "wires": [
      [
        "ceca83cd4532060b"
      ]
    ]
  },
  {
    "id": "de6c03aacddc94c1",
    "type": "function",
    "z": "08b3c4155c049b4a",
    "name": "Відповідь на /pressure",
    "func": "msg.payload = {\n  chatId: msg.payload.chatId,\n  type:\n  \"message\",\n  content: `Тиск: ${flow.get(\"pressure\")} || \"данні відсутні\"}\n  rПа`\n};\nreturn msg;",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 700,
    "y": 1220,
    "wires": [
      [
        "ceca83cd4532060b"
      ]
    ]
  }
]

```

```

    },
    {
      "id": "febefe31d171c396",
      "type": "function",
      "z": "08b3c4155c049b4a",
      "name": "Вдiповiдь на /air_quality",
      "func": "msg.payload = {\n  chatId: msg.payload.chatId,\n  type:\n  \"message\", \n  content: `Якiсть повітря: ${flow.get(\"air_quality\")} || \"даннi\n  вiдсутнi`} ppb`\n};\nreturn msg;",
      "outputs": 1,
      "timeout": "",
      "noerr": 0,
      "initialize": "",
      "finalize": "",
      "libs": [],
      "x": 690,
      "y": 1280,
      "wires": [
        [
          "ceca83cd4532060b"
        ]
      ]
    },
    {
      "id": "5857f4f126d10715",
      "type": "telegram receiver",
      "z": "08b3c4155c049b4a",
      "name": "",
      "bot": "5cc3c3b6a83274af",
      "saveDataDir": "",

```

```
"filterCommands": false,
"x": 190,
"y": 1120,
"wires": [
  [
    "bb2f29804b027119"
  ],
  [
    "bb2f29804b027119"
  ]
]
},
{
  "id": "1c26f7a6f3640b08",
  "type": "ui_chart",
  "z": "08b3c4155c049b4a",
  "name": "Temperature Chart",
  "group": "9582adecad5fdddd",
  "order": 1,
  "width": "0",
  "height": "0",
  "label": "Temperature (°C)",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm:ss",
  "interpolate": "linear",
  "nodata": "",
  "dot": false,
  "ymin": "-50",
  "ymax": "50",
```

```
"removeOlder": 1,  
"removeOlderPoints": "",  
"removeOlderUnit": "3600",  
"cutout": 0,  
"useOneColor": false,  
"useUTC": false,  
"colors": [  
    "#ff0000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000"  
],  
"outputs": 1,  
"useDifferentColor": false,  
"className": "",  
"x": 940,  
"y": 300,  
"wires": [  
    []  
]  
},  
{  
    "id": "83f5f030e28d5c0c",  
    "type": "ui_chart",  
    "z": "08b3c4155c049b4a",
```

```
"name": "Pressure Chart",
"group": "9582adecad5fdddd",
"order": 2,
"width": 6,
"height": 4,
"label": "Pressure (hPa)",
"chartType": "line",
"legend": "false",
"xformat": "HH:mm:ss",
"interpolate": "linear",
"nodata": "",
"dot": false,
"ymin": "600",
"ymax": "1200",
"removeOlder": 1,
"removeOlderPoints": "",
"removeOlderUnit": "3600",
"cutout": 0,
"useOneColor": false,
"useUTC": false,
"colors": [
  "#0000ff",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000"
```

```

    ],
    "outputs": 1,
    "useDifferentColor": false,
    "className": "",
    "x": 940,
    "y": 380,
    "wires": [
        []
    ]
},
{
    "id": "6e34d139ff61ceb1",
    "type": "ui_chart",
    "z": "08b3c4155c049b4a",
    "name": "Humidity Chart",
    "group": "9582adecad5fdddd",
    "order": 3,
    "width": 6,
    "height": 4,
    "label": "Humidity (%)",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "0",
    "ymax": "100",
    "removeOlder": 1,
    "removeOlderPoints": "",

```

```
"removeOlderUnit": "3600",
"cutout": 0,
"useOneColor": false,
"useUTC": false,
"colors": [
  "#00ff00",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000",
  "#000000"
],
"outputs": 1,
"useDifferentColor": false,
"className": "",
"x": 940,
"y": 460,
"wires": [
  []
]
},
{
  "id": "1e9ea8c8ec6481d4",
  "type": "ui_chart",
  "z": "08b3c4155c049b4a",
  "name": "Air Quality Chart",
  "group": "9582adecad5fdddd",
```



```
"order": 4,  
"width": 6,  
"height": 4,  
"label": "Air Quality (PPB)",  
"chartType": "line",  
"legend": "false",  
"xformat": "HH:mm:ss",  
"interpolate": "linear",  
"nodata": "",  
"dot": false,  
"ymin": "0",  
"ymax": "10000",  
"removeOlder": 1,  
"removeOlderPoints": "",  
"removeOlderUnit": "3600",  
"cutout": 0,  
"useOneColor": false,  
"useUTC": false,  
"colors": [  
    "#800080",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000",  
    "#000000"  
],  
"outputs": 1,
```

```
"useDifferentColor": false,
"className": "",
"x": 940,
"y": 540,
"wires": [
  []
]
},
{
  "id": "e61b2a62d9643063",
  "type": "ui_text",
  "z": "08b3c4155c049b4a",
  "group": "9582adecad5fdddd",
  "order": 5,
  "width": 6,
  "height": 1,
  "name": "Temperature Text",
  "label": "Current Temperature",
  "format": "{{msg.payload.temperature}} °C",
  "layout": "row-left",
  "className": "",
  "style": false,
  "font": "",
  "fontSize": "",
  "color": "#000000",
  "x": 940,
  "y": 620,
  "wires": []
},
{
```

```
"id": "8a8c1e8e6de0aec2",
"type": "ui_text",
"z": "08b3c4155c049b4a",
"group": "9582adecad5fdddd",
"order": 6,
"width": 6,
"height": 1,
"name": "Pressure Text",
"label": "Current Pressure",
"format": "{{ msg.payload.pressure }} hPa",
"layout": "row-left",
"className": "",
"style": false,
"font": "",
"fontSize": "",
"color": "#000000",
"x": 940,
"y": 700,
"wires": []
},
{
  "id": "1227145a75062a65",
  "type": "ui_text",
  "z": "08b3c4155c049b4a",
  "group": "9582adecad5fdddd",
  "order": 7,
  "width": 6,
  "height": 1,
  "name": "Humidity Text",
  "label": "Current Humidity",
```

```
"format": "{{msg.payload.humidity}} %",
"layout": "row-left",
"className": "",
"style": false,
"font": "",
"fontSize": "",
"color": "#000000",
"x": 940,
"y": 780,
"wires": []
},
{
  "id": "02c5ab610ea6f4d3",
  "type": "ui_text",
  "z": "08b3c4155c049b4a",
  "group": "9582adecad5fdddd",
  "order": 8,
  "width": 6,
  "height": 1,
  "name": "Air Quality Text",
  "label": "Current Air Quality (PPB)",
  "format": "{{msg.payload.airQuality}} ppb",
  "layout": "row-left",
  "className": "",
  "style": false,
  "font": "",
  "fontSize": "",
  "color": "#000000",
  "x": 940,
  "y": 860,
```

```
"wires": []
},
{
  "id": "5a6f992c03c2f675",
  "type": "function",
  "z": "08b3c4155c049b4a",
  "name": "temperature",
  "func": "msg.payload = msg.payload.temperature;\nreturn msg;\n",
  "outputs": 1,
  "timeout": 0,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 710,
  "y": 320,
  "wires": [
    [
      "1c26f7a6f3640b08"
    ]
  ]
},
{
  "id": "dc88757e814f71b7",
  "type": "function",
  "z": "08b3c4155c049b4a",
  "name": "pressure",
  "func": "msg.payload = msg.payload.pressure;\nreturn msg;",
  "outputs": 1,
  "timeout": 0,
```

```
"noerr": 0,
"initialize": "",
"finalize": "",
"libs": [],
"x": 700,
"y": 380,
"wires": [
  [
    "83f5f030e28d5c0c"
  ]
]
},
{
  "id": "ab535e782f89fe8c",
  "type": "function",
  "z": "08b3c4155c049b4a",
  "name": "humidity",
  "func": "msg.payload = msg.payload.humidity;\nreturn msg;",
  "outputs": 1,
  "timeout": 0,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 700,
  "y": 440,
  "wires": [
    [
      "6e34d139ff61ceb1"
    ]
  ]
}
```

```

    ]
  },
  {
    "id": "6f767415857ed74d",
    "type": "function",
    "z": "08b3c4155c049b4a",
    "name": "airQuality",
    "func": "msg.payload = msg.payload.airQuality;\nreturn msg;\n",
    "outputs": 1,
    "timeout": 0,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 700,
    "y": 500,
    "wires": [
      [
        "1e9ea8c8ec6481d4"
      ]
    ]
  },
  {
    "id": "5369d5e1e20757f8",
    "type": "switch",
    "z": "08b3c4155c049b4a",
    "name": "Обробка команд",
    "property": "payload.content",
    "propertyType": "msg",
    "rules": [

```

```
{
  "t": "eq",
  "v": "/menu",
  "vt": "str"
},
{
  "t": "eq",
  "v": "/start",
  "vt": "str"
}
],
"checkall": "true",
"repair": false,
"outputs": 2,
"x": 490,
"y": 980,
"wires": [
  [
    "e387e3cc31f35371"
  ],
  [
    "417566a48f8c7b88"
  ]
]
},
{
  "id": "e387e3cc31f35371",
  "type": "function",
  "z": "08b3c4155c049b4a",
  "name": "Створити меню",
```



```

    "func": "msg.payload = {\n    chatId: msg.payload.chatId,\n    type:
\"message\", \n    content: \"Выберите команду:\", \n    options: {\n
reply_markup: {\n    keyboard: [\n    [\"/status\", \"/graph\"], \n
[\"/temperature\", \"/humidity\"], \n    [\"/pressure\", \"/air_quality\"] \n
\n    ], \n    resize_keyboard: true, // Уменьшает клавиатуру \n
one_time_keyboard: true // Скрывает клавиатуру после выбора \n    } \n
}\n}; \nreturn msg; ",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 740,
    "y": 940,
    "wires": [
        [
            "cеса83cd4532060b"
        ]
    ]
},
{
    "id": "7cabb7e9170b6de9",
    "type": "function",
    "z": "08b3c4155c049b4a",
    "name": "Вдіповідь на /graph",
    "func": " // @ts-nocheck \nmsg.payload = {\n    chatId:
msg.payload.chatId, \n    type: \"message\", \n    content: `Діаграмма:
${\"https://gentle-pugs-add.local.lt/ui пароль 193.201.105.15\"}` \n}; \nreturn msg; ",
    "outputs": 1,

```

```
"timeout": "",
"noerr": 0,
"initialize": "",
"finalize": "",
"libs": [],
"x": 680,
"y": 1360,
"wires": [
  [
    "ceca83cd4532060b"
  ]
],
},
{
  "id": "bc79e73b23c6cc58",
  "type": "telegram receiver",
  "z": "08b3c4155c049b4a",
  "name": "",
  "bot": "5cc3c3b6a83274af",
  "saveDataDir": "",
  "filterCommands": false,
  "x": 210,
  "y": 1020,
  "wires": [
    [
      "5369d5e1e20757f8"
    ],
    [
      "5369d5e1e20757f8"
    ]
  ]
}
```

```

    ]
  },
  {
    "id": "1833d67b65ee1029",
    "type": "mqtt in",
    "z": "08b3c4155c049b4a",
    "name": "Получение данных от ESP32",
    "topic": "esp32/sensorData",
    "qos": "0",
    "datatype": "auto-detect",
    "broker": "2ec9c8b605a9e551",
    "nl": false,
    "rap": false,
    "inputs": 0,
    "x": 300,
    "y": 200,
    "wires": [
      [
        "546f247ae08287df"
      ]
    ]
  },
  {
    "id": "546f247ae08287df",
    "type": "function",
    "z": "08b3c4155c049b4a",
    "name": "Сохранение данных",
    "func": "flow.set(\"temperature\",
msg.payload.temperature);\nflow.set(\"humidity\",
msg.payload.humidity);\nflow.set(\"pressure\",

```

```
msg.payload.pressure);\nflow.set(\\"air_quality\\", msg.payload.airQuality);\nreturn  
msg;"
```

```
  "outputs": 1,  
  "timeout": "",  
  "noerr": 0,  
  "initialize": "",  
  "finalize": "",  
  "libs": [],  
  "x": 640,  
  "y": 200,  
  "wires": [  
    [  
      "6cb69aae173dc46f"  
    ]  
  ]  
},  
{  
  "id": "6cb69aae173dc46f",  
  "type": "debug",  
  "z": "08b3c4155c049b4a",  
  "name": "debug 5",  
  "active": false,  
  "tosidebar": true,  
  "console": false,  
  "tostatus": false,  
  "complete": "false",  
  "statusVal": "",  
  "statusType": "auto",  
  "x": 940,  
  "y": 200,
```

```
"wires": [],
},
{
  "id": "f949d14f4c7c4fa5",
  "type": "mqtt in",
  "z": "08b3c4155c049b4a",
  "name": "Получение данных от ESP32",
  "topic": "esp32/sensorData",
  "qos": "0",
  "datatype": "auto-detect",
  "broker": "2ec9c8b605a9e551",
  "nl": false,
  "rap": false,
  "inputs": 0,
  "x": 340,
  "y": 520,
  "wires": [
    [
      "5a6f992c03c2f675",
      "dc88757e814f71b7",
      "ab535e782f89fe8c",
      "6f767415857ed74d",
      "e61b2a62d9643063",
      "8a8c1e8e6de0aec2",
      "1227145a75062a65",
      "02c5ab610ea6f4d3"
    ]
  ]
},
{
```

```

    "id": "417566a48f8c7b88",
    "type": "function",
    "z": "08b3c4155c049b4a",
    "name": "Створити меню",
    "func": "msg.payload = {\n    chatId: msg.payload.chatId,\n    type:\n    \"message\",\n    content: \"Выберите команду:\",\n    options: {\n    reply_markup: {\n    keyboard: [\n    [\"/status\", \"/graph\"],\n    [\"/temperature\", \"/humidity\"],\n    [\"/pressure\", \"/air_quality\"]\n    ]\n    },\n    resize_keyboard: true, // Уменьшает клавиатуру\n    one_time_keyboard: true // Скрывает клавиатуру после выбора\n    }\n};\nreturn msg;";
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 740,
    "y": 980,
    "wires": [
      [
        "ceca83cd4532060b"
      ]
    ]
  },
  {
    "id": "5cc3c3b6a83274af",
    "type": "telegram bot",
    "botname": "Monitoring_station_bot",
    "usernames": "Yar_0_s1Av",

```

```
"chatids": "614098515",
"baseapiurl": "",
"testenvironment": false,
"updatemode": "polling",
"addressfamily": "",
"pollinterval": "300",
"usesocks": false,
"sockshost": "",
"socksprotocol": "socks5",
"socksport": "6667",
"socksusername": "anonymous",
"sockspassword": "",
"bothost": "",
"botpath": "",
"localbotport": "8443",
"publicbotport": "8443",
"privatekey": "",
"certificate": "",
"useselfsignedcertificate": false,
"sslterminated": false,
"verboselogging": false
},
{
  "id": "9582adecad5fdddd",
  "type": "ui_group",
  "name": "Classroom",
  "tab": "7a04e38909e6ee1d",
  "order": 1,
  "disp": true,
  "width": "6",
```

```
"collapse": false,  
"className": ""  
},  
{  
  "id": "2ec9c8b605a9e551",  
  "type": "mqtt-broker",  
  "name": "mosquitto",  
  "broker": "http://test.mosquitto.org",  
  "port": "1883",  
  "clientId": "",  
  "autoConnect": true,  
  "usetls": false,  
  "protocolVersion": "3",  
  "keepalive": "60",  
  "cleansession": true,  
  "autoUnsubscribe": true,  
  "birthTopic": "",  
  "birthQos": "0",  
  "birthRetain": "false",  
  "birthPayload": "",  
  "birthMsg": {},  
  "closeTopic": "",  
  "closeQos": "0",  
  "closeRetain": "false",  
  "closePayload": "",  
  "closeMsg": {},  
  "willTopic": "",  
  "willQos": "0",  
  "willRetain": "false",  
  "willPayload": "",
```



```
"willMsg": {},  
"userProps": "",  
"sessionExpiry": ""  
},  
{  
  "id": "7a04e38909e6ee1d",  
  "type": "ui_tab",  
  "name": "classroom",  
  "icon": "dashboard",  
  "disabled": false,  
  "hidden": false  
}  
]
```