

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інформаційних електронних систем
(повне найменування інституту, назва факультету (відділення))
Кафедра інформаційних радіоелектронних технологій і систем
(повна назва кафедри (предметної, циклової комісії))

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему:

**«РОЗРОБКА ЗАСОБУ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ У
САЛОНІ ЛІТАКІВ ГРОМАДСЬКОЇ АВІАЦІЇ»**

Виконав: студент 4-го курсу, групи KIBT-206
спеціальності 152 - Метрологія та інформаційно-
вимірювальна техніка

(шифр і назва напрямку підготовки, спеціальності)

Тартасюк Д.С.
(прізвище та ініціали)

Керівник: к.т.н., доц., доц. каф. ІРТС

Дудатьєв І.А.
(прізвище та ініціали)

«14» 08 2024 р.

Рецензент: доц. каф. БМІОЕС, к.т.н., доц.

Тужанський С.Є.
(прізвище та ініціали)

«14» 08 2024 р.

Допущено до захисту

Завідувач кафедри ІРТС

Осадчук О.В.
(прізвище та ініціали)

«14» 08 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інформаційних електронних систем
Кафедра інформаційних радіоелектронних технологій і систем
Рівень вищої освіти І-й (бакалаврський)
Галузь знань 15 «Автоматизація та приладобудування»
Спеціальність 152 - Метрологія та інформаційно-вимірвальна техніка
Освітньо-професійна програма Комп'ютеризовані інформаційно-вимірвальні технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІРТС

д.т.н., проф. Осадчук О.В.

15.03.2024 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Тартасюку Дмитру Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка засобу контролю параметрів мікроклімату у салоні літаків громадської авіації

керівник роботи к. т. н. доц., Дудатьєв І.А.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «11» 03 2023 року № 80

2. Строк подання студентом роботи 13.06.2024 р.

3. Вихідні дані до роботи: Об'єкт контролю – запиленість, чадний газ, CO₂, температура тиск. Час оновлення даних вимірювання – <30 сек. Спосіб передачі даних – безпроводний. Похибка по кожному сенсору засобу контролю – не більше 1%

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Огляд існуючих інформаційних технологій та систем для моніторингу якості повітря. Розробка структурної схеми засобу контролю параметрів мікроклімату у салоні літаків громадської авіації. Розробка апаратно-програмної частини засобу контролю параметрів мікроклімату у салоні літаків громадської авіації. Розробка програмного забезпечення обробки результатів спостережень та візуалізації даних

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Аналізатор якості повітря NDIR. Аналізатор якості повітря Z-WAVE. Аналізатор якості повітря Boseon. Аналізатор якості повітря Ajax. Газоаналізатор. Модуль датчика якості повітря MQ-135. Тестер запиленості Walcom

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Основна частина	доцент кафедри ІРТС к.т.н. доц., Дудатьєв ІА.		

7. Дата видачі завдання 17.03. 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Вибір, узгодження та затвердження теми БДР	14.02.2024-28.02.2024	
2.	Огляд та аналіз літературних джерел.	01.03.2024-23.03.2024	
3.	Затвердження теми. Розробка завдання на БДР.	20.03.2024-31.03.2024	
4.	Попередня розробка основних розділів. Аналіз шляхів вирішення поставленої задачі.	01.04.2024-15.05.2024	
5.	Математичне моделювання. Розробка структурної схеми.	16.05.2024-20.05.2024	
6.	Розробка графічної частини БДР	21.05.2024-28.05.2024	
7.	Оформлення пояснювальної записки та графічної частини.	28.05.2024-06.06.2024	
8.	Нормоконтроль	07.06.2024-09.06.2024	
9.	Попередній захист БДР, доопрацювання, рецензування БДР	10.06.2024-14.06.2024	
10.	Захист БДР ЕК	18.06.2024	

Студент


(підпис)

Тартасюк Д.С.

Керівник роботи


(підпис)

Дудатьєв І.А.

АНОТАЦІЯ

УДК 504.3.054

Тартасюк Д.С. Розробка засобу контролю параметрів мікроклімату у салоні літаків громадської авіації. Бакалаврська дипломна робота. – Вінниця: ВНТУ, 2024. – 78 с. На українській мові. Бібліогр.: 13 назв; табл.: 1; рис. 61.

У цій бакалаврській дипломній роботі розглядається розробка засобу контролю параметрів мікроклімату у салоні літаків громадської авіації. Основна мета роботи полягає у створенні ефективного та надійного інструменту для моніторингу та регулювання температури, вологості, концентрації CO₂ та інших параметрів мікроклімату з використанням сучасних технологій.

Розроблений засіб контролю параметрів мікроклімату може бути використаний для забезпечення комфортних умов під час польотів, що сприятиме підвищенню якості обслуговування пасажирів та покращенню загальної безпеки на борту літака.

Ключові слова: мікроклімат, сенсори, сенсор пилу, сенсор CO₂, якість повітря

ABSTRACT

D.S. Tartasyuk Development of a means of controlling microclimate parameters in the cabin of civil aviation aircraft. Bachelor thesis. – Vinnytsia: VNTU, 2024. – 78 p. In Ukrainian language. Bibliography: 13 titles; table: 1; Fig. 61.

This bachelor's thesis addresses the development of a means to control microclimate parameters in the cabins of commercial aircraft. The main goal of the work is to create an effective and reliable tool for monitoring and regulating temperature, humidity, CO₂ concentration, and other microclimate parameters using modern technologies.

The developed microclimate control device can be used to ensure comfortable conditions during flights, which will contribute to improving passenger service quality and enhancing overall safety on board the aircraft.

Keywords: microclimate, sensors, dust sensor, CO₂ sensor, air quality

ЗМІСТ

ВСТУП.....	6
1 ОГЛЯД ІСНУЮЧИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ ДЛЯ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ	9
1.1 Особливості об’єкту дослідження. Актуальність розробки засобу контролю параметрів мікроклімату у салоні літаків громадської авіації	9
1.2 Система моніторингу FooBot	16
1.3 Висновки до розділу 1.....	21
2 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ЗАСОБУ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ У САЛОНІ ЛІТАКІВ ГРОМАДСЬКОЇ АВІАЦІЇ.....	23
2.1. Схема електрична структурна засобу контролю параметрів мікроклімату у салоні літаків громадської авіації	23
2.2 Розрахунок нормуючого пристрою для датчика вологості	24
2.3 Розрахунок нормуючого пристрою для датчика пилу	25
2.4 Висновки до розділу 2.....	27
3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОЇ ЧАСТИНИ ЗАСОБУ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ У САЛОНІ ЛІТАКІВ ГРОМАДСЬКОЇ АВІАЦІЇ.....	28
3.1. Огляд давачів для вимірювання показників якості повітря	28
3.2 Вибір оптимального програмного середовища для вимірювання показників якості повітря	35
3.3 Висновки до розділу 3.....	47
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОБРОБКИ РЕЗУЛЬТАТІВ СПОСТЕРЕЖЕНЬ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ.....	48
4.1 Підключення та налаштування датчиків моніторингу повітря	48
4.2 Опис протоколу MQTT для обміну між давачами.....	63
4.3 Висновки до розділу 4.....	68
ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
Додаток А (Обов’язковий) Ілюстративна частина.....	72
Додаток Б (Обов’язковий) Протокол перевірки роботи.....	76

ВСТУП

Актуальність.

Комфорт пасажирів та екіпажу під час авіаперельотів є одним із ключових аспектів, що визначають якість обслуговування та безпеку польотів. Мікроклімат у салоні літака, зокрема температура, вологість, рівень вуглекислого газу та інші параметри, суттєво впливають на самопочуття та здоров'я людей на борту. Недостатня увага до цих факторів може призвести до зниження комфорту, появи нездужань та навіть до серйозних проблем зі здоров'ям у пасажирів і членів екіпажу.

Сучасні системи контролю мікроклімату у салонах літаків громадської авіації мають низку обмежень, пов'язаних з їхньою ефективністю, надійністю та здатністю адаптуватися до змінних умов польоту. Впровадження нових технологій та підходів до регулювання параметрів мікроклімату може суттєво підвищити якість польотів, забезпечуючи більш комфортні умови для пасажирів та зменшуючи навантаження на системи життєзабезпечення літака.

Актуальність даної роботи обумовлена необхідністю розробки інноваційного засобу контролю параметрів мікроклімату, який зможе забезпечити оптимальні умови перебування пасажирів на борту літаків громадської авіації. Така система повинна бути ефективною, надійною та енергозберігаючою, відповідати сучасним вимогам авіаційної галузі та мати можливість інтеграції з існуючими бортовими системами.

Ця робота спрямована на розробку і впровадження такого засобу контролю, що дозволить не тільки підвищити комфорт та безпеку пасажирів, але й оптимізувати роботу систем життєзабезпечення літака, зменшивши при цьому експлуатаційні витрати та негативний вплив на навколишнє середовище.

Актуальність. Комфорт та безпека пасажирів є ключовими пріоритетами у громадській авіації. Мікроклімат у салоні літака, включаючи температуру, вологість, концентрацію CO₂ та інші параметри, безпосередньо впливає на самопочуття пасажирів та екіпажу. Ефективний контроль та регулювання цих параметрів є необхідними для забезпечення комфортних умов польоту, зниження ризиків пов'язаних зі здоров'ям та покращення загальної якості обслуговування.

Метою роботи є розробка засіб контролю параметрів мікроклімату у салоні літаків громадської авіації, що дозволить забезпечити оптимальні умови для пасажирів під час польоту.

Для досягнення поставленої мети в роботі сформульовані та вирішені такі задачі:

- Провести аналіз існуючих систем контролю мікроклімату у салонах літаків.
- Визначити ключові параметри мікроклімату, які впливають на комфорт пасажирів.
- Розробити концептуальну модель засобу контролю параметрів мікроклімату.
- Обрати апаратні та програмні засоби для реалізації системи.
- Провести моделювання та тестування розробленої системи.
- Оцінити ефективність та надійність засобу контролю у різних умовах експлуатації.

Об'єктом дослідження в роботі є процес вимірювального контролю параметрів мікроклімату у салоні літаків громадської авіації.

Предметом дослідження є методи та засоби, які можна застосувати при побудові засобу контролю параметрів мікроклімату у салоні літаків громадської авіації.

Методи дослідження. Для досягнення поставлених завдань будуть використані методи системного аналізу, комп'ютерного моделювання, експериментальні дослідження, а також методи статистичної обробки даних.

Практичне значення одержаних результатів.

У роботі отримані такі практичні результати:

1. Застосування ефективних засобів контролю мікроклімату дозволяє забезпечити оптимальні умови температури, вологості, чистоти повітря та вентиляції, що позитивно впливає на самопочуття та комфорт пасажирів і екіпажу під час польоту.

2. Забезпечення оптимального мікроклімату у салоні сприяє підтриманню високого рівня пильності та працездатності екіпажу, що є критично важливим для забезпечення безпеки польотів.

3. Розробка засобу передбачає інтеграцію з автоматичними системами управління кліматом у літаку, що дозволяє автоматично підтримувати оптимальні умови без постійного втручання екіпажу.

4. Розроблена структурна схема передбачає використання різноманітних сенсорів для вимірювання параметрів мікроклімату, таких як температура, вологість, тиск, рівень CO₂ та інші. Це забезпечує комплексний підхід до моніторингу мікроклімату. Структурна схема розроблена таким чином, що її можна легко масштабувати та модифікувати відповідно до потреб конкретного літака чи авіакомпанії. Модульний підхід дозволяє додавати або замінювати компоненти без значних витрат.

Особистий внесок здобувача.

Основні положення та результати, що представлені в роботі, одержані автором самостійно.

1 ОГЛЯД ІСНУЮЧИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ ДЛЯ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ

1.1 Особливості об'єкту дослідження. Актуальність розробки засобу контролю параметрів мікроклімату у салоні літаків громадської авіації

Останнім часом кількість пасажирів, які надають перевагу подорожам літаками, збільшується з року в рік. У сучасних літаках підтримуються різні та змінювані параметри внутрішнього середовища, що стосуються температури, тиску та вологості. Характеристики повітря в салонах літаків схожі з параметрами внутрішнього повітря будинків та офісних будівель.

Проте салони літаків все ж відрізняються від будівель у багатьох аспектах, зокрема простором, що припадає на одного пасажирів, необхідністю герметизації салону, вимушеним браком активності під час польоту тощо. На пасажирів, які можуть мати різні захворювання та належати до різних вікових категорій, впливають такі атмосферні фактори, як недостатня вологість, знижений тиск повітря, наявність у повітрі різних забруднюючих речовин (озон, вуглекислий газ та інші органічні речовини і біологічні структури).

Сучасні літаки обладнані системою контролю внутрішнього середовища (Environmental Control System – ECS), що забезпечує безпечний, комфортний та не загрозливий для здоров'я політ як для пасажирів, так і для екіпажу. У цій системі зовнішнє повітря забирається силовою системою літака, змішується з відфільтрованим внутрішнім повітрям і подається в салон. Система ECS призначена для зниження концентрації в повітрі забруднюючих речовин, які можуть проникнути в салон, а також для регулювання температури, тиску, вологості та вентиляції салону літака.

Чистота повітря в приміщенні визначає якість внутрішнього повітря в цьому приміщенні. Чисте повітря – це повітря, концентрація забруднюючих речовин у якому не перевищує певного рівня, встановленого стандартами, і яке не викликає дискомфорту у людей.

У цій статті коротко розглядаються сучасні системи ECS, зазначаються ризики, пов'язані з забруднюючими речовинами в повітрі, загрозою здоров'ю та зниженням комфорту в контексті якості внутрішнього повітря.

Системи контролю внутрішнього середовища та якості внутрішнього повітря в літаках

Внутрішнє середовище в сучасних літаках має фізичні характеристики, які не дозволяють людям перебувати в такому середовищі без будь-якого захисту. Тому в сучасних літаках використовуються досить складні системи ECS. Ці системи призначені для задоволення фізіологічних потреб людей у будь-яких умовах польоту та для надання їм певного рівня комфорту під час польоту.

На рисунку 1 представлені основні компоненти ECS, що забезпечують задоволення фізіологічних потреб пасажирів і створюють певний комфорт у салоні літака. Конструкція системи вентиляції салону заснована на подачі вентиляційного повітря в кожен ряд крісел і на зборі та видаленні відпрацьованого повітря з простору того ж ряду. Такий підхід знижує ризик поширення інфекцій по всьому салону літака.

Перші реактивні літаки не мали системи рециркуляції повітря в салоні. Основною причиною цього було те, що двигуни цих літаків мали низький ККД, і все повітря, що всмоктувалося двигуном, використовувалося для створення сили тяги. Удосконалення технології виробництва двигунів призвело до створення двигунів нового типу – так званих турбовентиляторних двигунів. Перші турбовентиляторні двигуни мали ступінь двоконтурності 2:1, завдяки чому забезпечувалося більш економічне витрачання палива. Проте таке вдосконалення, що зробило можливим використання зовнішнього повітря на 100 %, не було в той час розроблено до необхідного ступеня.

Більш пізні турбовентиляторні двигуни мали ступінь двоконтурності 5:1. Але в цих двигунах збільшилося споживання палива через відводиме повітря. Оскільки тоді ще не було систем рециркуляції, подача в салон тільки зовнішнього повітря викликала значне збільшення експлуатаційних витрат у літаках. Завдяки дослідженням, проведеним у цій області, була досягнута значна економія палива без зниження якості внутрішнього повітря. У літаках почали використовувати си-

стеми рециркуляції, що застосовують високоефективні фільтри і дозволяють знизити частку відводимого повітря.

Фактори, що впливають на якість внутрішнього повітря в літаках, можуть бути розбиті на п'ять категорій:

- тиск;
- вміст кисню;
- температура;
- вологість;
- концентрація забруднюючих речовин у повітрі.

Раптова зміна рівня одного чи кількох з цих факторів або взаємодія між ними можуть викликати погіршення якості внутрішнього повітря і, відповідно, можуть негативно впливати на здоров'я пасажирів і екіпажу.

Тиск. Оскільки на великій висоті щільність повітря дуже мала, сила опору, що діє на літак, також мала. Це обставина робить ефективним політ на великих висотах. З цієї причини для забезпечення безпеки екіпажу і пасажирів салони майже всіх комерційних літаків герметизовані. Діючий стандарт герметизації салонів літаків визначено в розділі 25.841 Федеральних авіаційних приписів (FAR) Федерального авіаційного агентства (FAA).

Згідно з цим стандартом, тиск у салоні за нормальних робочих умов не повинен бути вищим за тиск на висоті 2 450 м. Навіть якщо тиск у салоні підтримується на постійному рівні, що відповідає 2 450 м, це тиск нижчий за тиск на рівні моря. Такий низький тиск може негативно позначатися на фізіології пасажирів. Знижений тиск викликає розширення бульбашок повітря, які знаходяться в клітинах тіла людини, що може бути причиною поганого самопочуття, а у людей з підвищеною чутливістю це може викликати більш серйозні загрози здоров'ю.

Кисень. При стабільному стані атмосфери величина атмосферного тиску на рівні моря дорівнює 760 мм ртутного стовпа. При цьому парціальний тиск кисню становить 160 мм ртутного стовпа (приблизно 21%). Але оскільки в цих умовах відсотковий вміст вуглекислого газу і водяної пари зростає, парціальний тиск кисню знижується приблизно до величини 105 мм ртутного стовпа. Таким чином, парціальний тиск, до якого звик людський організм, дорівнює 105 мм

ртутного стовпа. Якщо парціальний тиск кисню нижчий за цей рівень, надходження кисню в кров знижується, і звичайний ритм організму людини порушується.

Температура. Температура в салоні в більшій мірі впливає на комфорт людей, які летять у літаку. Однак разом з іншими фізичними і біологічними факторами вона може чинити негативний вплив і на здоров'я, оскільки впливає на інтенсивність втрати рідини пасажирями та членами екіпажу. Крім того, вологість, яка безпосередньо пов'язана з температурою, також впливає на тепловий комфорт.

Температура також впливає на сприйняття якості повітря. При підвищенні температури повітря в приміщенні сприймається людьми, які там перебувають, як більш забруднене. Крім цього, температура суттєво впливає на інтенсивність виділення летких органічних сполук (Volatile Organic Compounds – VOC), джерелами яких є тіло людини та матеріали салону. Відповідно, концентрація забруднюючих речовин у повітрі салону також буде зростати.

Вологість. Відносна вологість повітря в салоні має дві важливі характеристики, що стосуються якості повітря: комфорт пасажирів і вплив на конструкцію літака та безпеку польоту. Висока відносна вологість (вище 70%), особливо при високій температурі, створює некомфортне середовище для пасажирів. У такій ситуації в салоні випадає конденсат, і краплі води можуть викликати корозію, що може становити небезпеку для конструкції літака. Крім того, деякі небезпечні для людини біологічні організми можуть розмножуватися в умовах підвищеної вологості.

Удосконалені системи ECS сучасних літаків осушують забортне повітря за допомогою відокремлювачів вологи. Таким чином, основними джерелами вологості в салоні є дихання пасажирів і випаровування з поверхні людського тіла. Змішуванням зовнішнього і рециркуляційного повітря із салону досягається відносна вологість близько 10-20%. Ці значення нижчі за показники, задані стандартами комфорту ASHRAE (Американське товариство інженерів з опалення, охолодження та кондиціонування повітря).

Забруднюючі речовини в повітрі салону літака. Забруднення внутрішнього повітря усувається вентиляційним забортним повітрям. Після того як забруднюючі речовини внутрішнього повітря розбавляються вентиляційним повітрям, внутрішнє повітря подається в салон. Витрата зовнішнього повітря є важливим фактором ступеня забрудненості повітря в салоні. Приписи FAR 25 встановлюють, що мінімальна кількість забираного свіжого повітря на одну людину становить 12,5 м³/год.

Основними джерелами забруднення внутрішнього повітря є дихання пасажирів, запахи їжі, туалет тощо або джерела, що знаходяться поза салоном. Для очищення рециркуляційного повітря від патогенних і небезпечних часток дуже важлива ефективність фільтрації застосовуваних фільтрів. У вентиляційних системах літаків застосовуються HEPA-фільтри ((високоєфективні повітряні фільтри твердих часток (HEPA filter – High Efficiency Particulate Air Filter))

. Такі фільтри здатні затримувати віруси і бактеріальні структури, вони характеризуються середньою ефективністю 94-99,97%. Однак для ефективної роботи ці фільтри повинні регулярно перевірятися.

Забруднення від зовнішніх джерел. Вентиляційне повітря, що подається в салон, забирається ззовні. Тому забруднюючі речовини, що знаходяться в зовнішньому повітрі, можуть проникнути в пасажирський салон. Більшість аеропортів розташовано поблизу великих міст, для яких характерні значні проблеми, пов'язані із забрудненням повітря. Крім того, ще одним джерелом забруднення є системи виведення відпрацьованого повітря в аеропортах. Після посадки літака вихлопні гази і пари протиобліднювальних хімічних речовин, що проникають через відчинені двері, можуть створити серйозну загрозу здоров'ю.

Основне забруднення зовнішнього повітря. Основне забруднення зовнішнього повітря відбувається на дуже низькому рівні. На висоті польоту основною забруднюючою речовиною є озон (O₃). Атмосферний озон утворюється в результаті перетворення атмосферного кисню під дією ультрафіолетового випромінювання сонця. Стандарти, що стосуються цього питання, визначені FAA (FAR 25.832 та FAR 121.578). Озон може бути причиною проблем з диханням, він

загострює астматичні прояви та негативно впливає на імунну систему організму людини.

Внутрішні джерела забруднень. У внутрішньому повітрі салону можуть знаходитися віруси, бактеріальні організми та інші мікроорганізми, що виділяються членами екіпажу та пасажирями. Джерелами цього типу забруднення можуть бути одяг, дихання та випаровування з поверхні шкіри. Пасажири та члени екіпажу є джерелами різних бактерій, вірусів та алергенів.

Крім того, найбільш важливим джерелом забруднення є вуглекислий газ, що є побічним продуктом дихання. Чохли сидінь та килимки на підлозі можуть містити пил, мікроорганізми та алергени, тому вони також можуть бути джерелами забруднення внутрішнього повітря.

Під час перерв між польотами салони чистяться. Під час процесу чистки застосовуються різні чистячі та миючі засоби і розчинники. Ці чистячі засоби при випаровуванні створюють забруднення на поверхнях та у внутрішньому повітрі.

Для захисту літака від комах та інших організмів використовуються різні пестициди. У багатьох країнах є обов'язковою процедура знищення комах у літаках, що прилетіли з країн, у яких поширена малярія та трапляються епідемії жовтяниці. Хімічні речовини, які використовуються в ході такої процедури, при неправильному їх застосуванні можуть становити небезпеку для здоров'я.

Проблеми, які можуть бути викликані поганою якістю повітря в салоні літака.

Гіпоксія. Гіпоксія – це ослаблення функцій організму людини, викликане нестачею кисню в клітинах крові та тканинах. Нестача кисню може бути викликана кількома причинами, але основною з них є зниження парціального тиску кисню в легенях, що є результатом зменшення тиску кисню в навколишньому середовищі. Гіпоксія може виявлятися за певними симптомами, але іноді які-небудь симптоми можуть бути відсутніми. Якщо гіпоксія розвивається без симптомів, це є найгіршим випадком для пасажирів, оскільки вони не знають про зниження функцій свого організму.

Під час цих явищ можуть спостерігатися наступні симптоми: сповільнення мислення та невірні обчислення, втрата пам'яті, замедлення реакції, часте дихання

та ціаноз, ослаблене відчуття, погіршення координації м'язової активності та втрата свідомості.

Тромбоз глибоких вен. Тромбоз глибоких вен - це утворення згортка крові в глибоких венах ніг, хоча він може виникати й у інших частинах тіла. Небезпека полягає в тому, що частина цього згортка може відірватися (емболія) та почати рухатися по кровоносних судинах легень, де вона може застрягти та заблокувати одну з легеневих артерій. Це називається легеневою тромбоемболією. Наслідком цього може бути зниження рівня кисню в крові, помилки в обчисленнях, втома та смерть.

Дослідження показали, що ця хвороба зазвичай виникає при тривалих перельотах, особливо у пожилого пасажирства. Під час довгих перельотів симптоми цієї хвороби можуть виявлятися або в перші години польоту, безпосередньо після зльоту, або через декілька днів після польоту, в середньому через чотири дні. Розвиток тромбозу в основному залежить від таких факторів, як недостатність руху, спазми, сидяче положення, обезвожування та атмосферний тиск в салоні.

Передача інфекції. Іноді під час тривалих польотів деякі люди скаржаться на те, що пасажирів та члени екіпажу заразили їх інфекційним захворюванням. Багато пасажирів бояться заразитися від сусідів або через вдихання рециркуляційного повітря в салоні. Очевидно, що під час тривалих перельотів за межі країни існує підвищений ризик зараження вірусами та патогенними організмами. Основними джерелами інфекції в салоні є забруднена вода та їжа, туалети, безпосередні контакти та/або виділення тіла, комахи та збудники хвороб, що поширюються від людини до людини повітряним шляхом.

У салонах літаків здійснюється змішування використаного та вентиляційного повітря в пропорції 50 на 50%. Якщо фільтри, які повинні ефективно видаляти з повітря салону віруси та бактерії, працюють неефективно або не піддаються регулярному технічному обслуговуванню, зростає ризик погіршення якості внутрішнього повітря. Проводячи регулярну перевірку фільтрів, необхідно переконатися, що в повітрі салону відсутні мікроорганізми.

Інші фактори. Ймовірно, температура повітря може впливати на здоров'я людей та якість повітря косвенно:

Оскільки при високій температурі люди зазвичай зменшують активність, при зростанні температури повітря салону збільшується ризик тромбозу глибоких вен.

Комбінація високої температури та вологості може підвищити життєздатність патогенних організмів, що поширюються повітрям, і негативно позначитися на імунній системі організму, що бореться з хворобами.

У салонах багатьох літаків, в залежності від типу літака, вологість може змінюватися в межах від 15 до 19%. Хоча низька вологість не є "забруднюючою речовиною" для повітря салону, вона є фактором, що впливає на самопочуття пасажирів та членів екіпажу.

Недостатня вологість може спричинити сухість поверхні тіла (слизової оболонки та шкіри) і подразнення очей. Висихання слизової оболонки очей викликає слъзотечу та біль в очах. Недостатня вологість може серйозно погіршити самопочуття людей, що страждають астмою, респіраторними захворюваннями та захворюваннями верхніх дихальних шляхів. Порушення добового ритму організму, що є звичайним явищем при тривалих польотах, викликається перетином за короткий час кількох часових поясів. Порушення біологічного ритму зазвичай проявляється таким чином: втома, зниження уваги та концентрації, ослаблення пам'яті, тошнота та порушення травлення.

1.2 Система моніторингу FooBot

FooBot — це система моніторингу навколишнього середовища, яка може виконувати комплексні вимірювання якості повітря в салоні літаків та контролювати наявність формальдегіду, бензолу, етиленгліколю, ацетону, оксидів і вуглекислого газу, а також вимірювати температуру та відносну вологість (див. рис. 1.1).



Рисунок 1.1 – Система моніторингу навколишнього середовища FooBot

Зібрана інформація використовується для порівняння зі звичайними характеристиками повітря, обробляється спеціальними алгоритмами та передається на смартфон користувача, щоб він міг вжити заходів для оптимізації циркуляції повітря у своєму будинку чи квартирі.

Пристрій являє собою невеликий циліндр, що містить різні датчики, який бездротовим способом підключається до планшета або смартфона користувача і дозволяє регулярно контролювати якість вашого домашнього середовища за допомогою зручного графічного дисплея.

Враховуючи сучасний глобальний ринок таких систем, розвиток систем моніторингу екосистем є доцільним через зростання попиту на портативні обчислювальні пристрої та вбудовані рішення. Сучасна концепція IoT означає, що всі сучасні пристрої, незалежно від платформи, повинні мати можливість взаємодіяти з іншими пристроями та службами, створюючи єдину взаємопов'язану екосистему, а не існуючи ізольовано.

1.3 Аналіз засобів прийому-передачі даних про якість повітря у салонах літаків

Датчики та джерела живлення. Інтернет починається або закінчується однією подією: простою дією, зміною температури або, можливо, важелем, який за-

криває замок. На відміну від багатьох існуючих ІТ-пристроїв, IoT передусім передбачає фізичні дії або події. Він формує реакцію на певні фактори реального світу. Іноді один датчик може генерувати великі обсяги даних, як, наприклад, акустичні датчики, які використовуються для профілактичних перевірок обладнання. В інших випадках достатньо лише одного біта даних, щоб передати важливу інформацію про стан здоров'я пацієнта. У будь-якому випадку сенсорні системи продовжують розвиватися і, відповідно до закону Мура, скоротилися до субнанометрового рівня та стали дешевшими. Це саме те, до чого закликають ті, хто прогнозує, що мільярди пристроїв будуть підключені до Інтернету речей, тому ці прогнози мають сенс. Тому, розглядаючи Інтернет речей, необхідно враховувати MEMS, датчики та інші типи недорогих периферійних пристроїв і їхні електрофізичні властивості. Це також стосується систем живлення та енергії, необхідних для живлення цих периферійних пристроїв. Не можна вважати, що пристрої Edge працюють за замовчуванням. Мільярди крихітних датчиків все ще вимагають багато енергії. Організаційні питання хмарних сервісів IoT також включають питання живлення.

Передача даних. Розвиток Інтернету речей приділяє велику увагу встановленню з'єднань і роботі мережі. Інтернет речей не існував би без надійної технології передачі даних із найвіддаленіших і неблагополучних регіонів до найбільших центрів збору даних Google, Amazon, Microsoft і IBM. Фраза «Інтернет речей» містить слово «Інтернет», тому необхідно вивчати суміжні питання, такі як мережеві технології, обмін даними і навіть теорія сигналізації. Основною підтримкою Інтернету речей є не датчики чи програми, а можливість встановлення з'єднань.

Передача даних і встановлення мережевих з'єднань базуються на системах зв'язку малого радіусу дії – персональних мережах (PAN), які зазвичай будуються без дотримання правил протоколу IP (див. рис. 1.2). Це може бути дротова мережа і бездротова мережа. Бездротові мережі/протоколи IoT зазвичай включають протоколи Bluetooth, mesh, Zigbee і Z-Wave. Для IIoT він також має Wireless HART і ISA100. Це яскравий приклад різноманіття систем бездротового зв'язку IoT. Спи-

сок дротових мереж ще більший, оскільки включає всі можливі промислові мережі та протоколи.

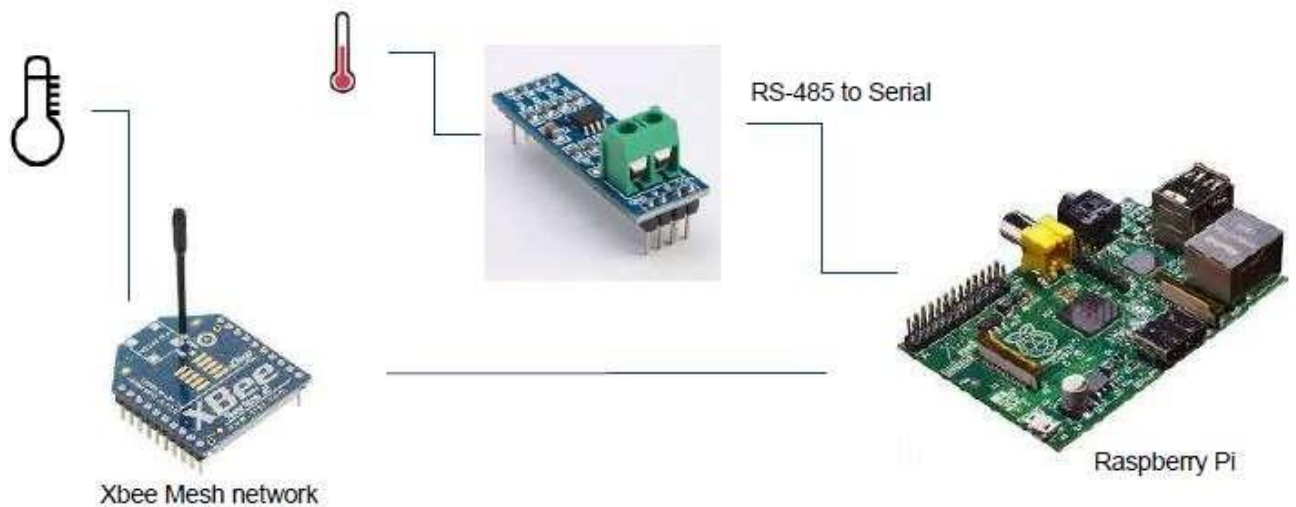


Рисунок 1.2 – Передача даних

Окрім PAN, також використовуються бездротові локальні мережі та системи зв'язку на основі протоколів IP, включаючи розгалужені мережі Wi-Fi на основі стандарту IEEE 802.11, 6LoWPAN та технології Thread. Телекомунікації на основі стандартів стільникового зв'язку (3G, 4G LTE) і нових стандартів, що забезпечують IoT і міжмашинну взаємодію (наприклад, Cat-1 і Cat-NB), а також спеціальні протоколи LoRaWAN і Sigfox, спеціально для IoT (див. рисунок 1.3) .

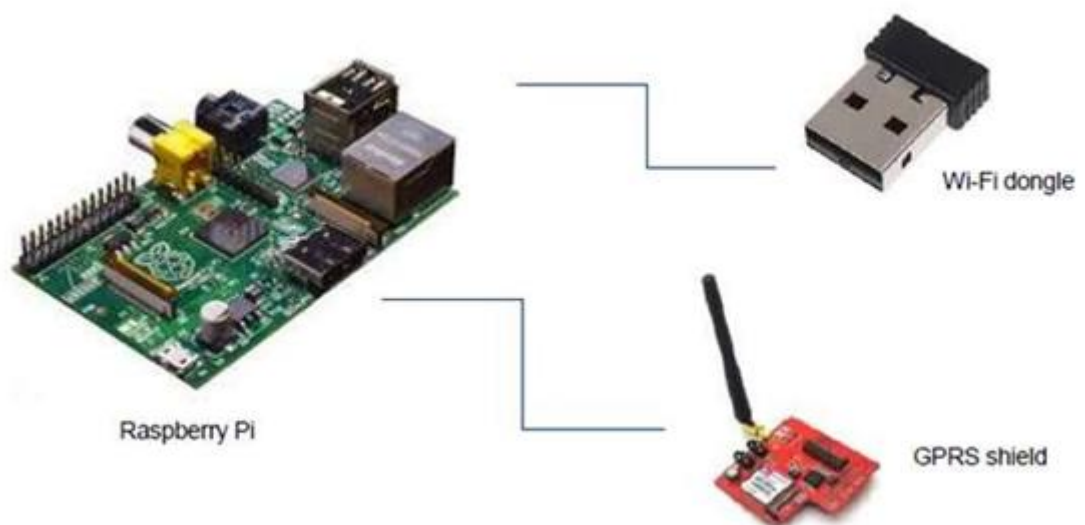


Рисунок 1.3 – Передача даних

Маршрутизація. Для передачі даних від датчиків в інтернет-простір необхідні дві технології: шлюзи маршрутизаторів і підтримка інтернет-протоколів, які забезпечують ефективний обмін даними. Маршрутизатори особливо важливі в таких аспектах, як безпека, керування та маршрутизація даних. Прикордонні маршрутизатори керують і контролюють стан окремих сітчастих мереж, а також коригують і підтримують якість даних. Конфіденційність і безпека даних також дуже важливі. Маршрутизатори відіграють важливу роль у створенні віртуальних приватних мереж, віртуальних локальних мереж і програмно-визначених глобальних мереж. Насправді вони можуть містити тисячі вузлів, які обслуговуються одним граничним маршрутизатором, у певному сенсі маршрутизатор діє як розширення хмари (граничний пристрій).

Цей рівень використовує багато протоколів, необхідних для обміну даними між вузлами, маршрутизаторами та хмарними службами в системі IoT. Інтернет речей проклав шлях для нових протоколів IoT, які можна порівняти з традиційними протоколами HTTP та SNMP, які використовувалися десятиліттями. Передача даних IoT потребує ефективних, енергозберігаючих протоколів із низькою затримкою, які можуть легко та безпечно надсилати дані в хмару та з неї. Тут особливо використовуються загальні протоколи, такі як MQTT, AMQP і CoAP.

Туманні і граничні обчислення, аналітика і машинне навчання. На цьому етапі вам потрібно вирішити, як обробляти потік даних від периферійного пристрою до хмарної служби. Щоб навчитися правильно оцінювати, як система буде розвиватися і рости, необхідно розуміти вплив, який усі складності та затримки архітектури хмарної системи мають на системи IoT. Крім того, не все слід надсилати в хмару. Пересилання всіх даних IoT набагато дорожче, ніж обробка даних на межі мережі (граничні обчислення) або включення периферійних маршрутизаторів у зону, що обслуговується хмарним сервісом (туманні обчислення). Обчислення туману також стандартизується, зокрема існує такий стандарт обчислення туману, як архітектура OpenFog.

Значну вагу можуть мати дані, отримані шляхом перетворення аналогових фізичних ефектів у цифрові сигнали. Саме тут вступають у гру інструменти аналітики та процесори правил систем IoT. Складність впровадження системи IoT за-

лежить від рішення, яке проектується. У деяких випадках все дуже просто: наприклад, коли вам потрібно встановити простий процесор правил для моніторингу аномальних стрибків температури на межовому маршрутизаторі, який контролює кілька датчиків. Інший сценарій полягає в тому, що великі обсяги структурованих і неструктурованих даних передаються в хмарні озера даних у режимі реального часу, що вимагає високої швидкості обробки (для прогнозової аналітики) і довгострокових прогнозів на основі високотехнологічних моделей машинного навчання, таких як Loop Predictive. Пакет аналізу сигналів Нейронні мережі залежать від часу. Тут є деякі аналітичні проблеми та складності, які можна вирішити за допомогою різних підходів і методів, таких як формування складних обробників подій, байєсівських мереж і нейронних мереж.

Загроза і безпека в Інтернеті речей. Багато систем IoT не будуть обмежуватися безпечним простором будинку чи офісу. Їх можна знайти в громадських місцях, у дуже віддалених районах, у транспортних засобах, що рухаються, і навіть усередині людей. Інтернет речей

Величезна єдина ціль для хакерів усіх видів. Було виявлено численні освітні атаки, організовані зломи та навіть уразливості національної безпеки, націлені на пристрої IoT. Розробники рішень IoT повинні розуміти характеристики таких уразливостей і способи їх нейтралізації, а також стандартні заходи, призначені для захисту IoT або будь-яких мережевих компонентів.

1.3 Висновок до розділу 1

У першому розділі бакалаврської дипломної роботи зроблено огляд та аналіз існуючих систем екологічного моніторингу. Дослідивши ці питання, було зроблено наступні висновки:

- З'ясовані причини забруднення повітря;
- Розкрито значення економічного екологічного моніторингу;
- Проведено дослідження існуючих систем економічного моніторингу;
- Огляд та експертиза засобів отримання та розповсюдження інформації в інтернеті.

На сьогоднішній день від пасажирів та членів екіпажу отримано багато скарг, пов'язаних з умовами польоту, параметрами внутрішнього середовища салону (тиском і вологістю), вагомими у повітрі твердими частинками та забруднюючими речовинами (очищувальними засобами, гідравлічною рідиною, озон), а також з фізіологічними проявами та незручностями польоту (втомою, шумом, необхідністю тривалий час сидіти в замкненому просторі, порушенням біологічних ритмів).

Однак через те, що до цього часу не проводився централізований збір та аналіз цих скарг, дуже складно встановити зв'язок між факторами, що впливають на якість внутрішнього повітря, та потенційною небезпекою цих факторів.

Немає сумніву, що дослідження в цій області допоможуть встановити необхідні параметри якості внутрішнього повітря в літаках і що ці зусилля призведуть до підготовки стандарту якості внутрішнього повітря для літаків. Крім того, вони сприятимуть забезпеченню, особливо для тривалих польотів, якості внутрішнього повітря, що відповідає потребам високого рівня комфорту та не представляє загрози для здоров'я людей та безпеки польотів.

2 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ЗАСОБУ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ У САЛОНІ ЛІТАКІВ ГРОМАДСЬКОЇ АВІАЦІЇ

2.1 Схема електрична структурна засобу контролю параметрів мікроклімату у салоні літаків громадської авіації

Формальна структура проекту системи контролю якості повітря в салонах літаків громадської авіації проілюстрована на рисунку 2.1.

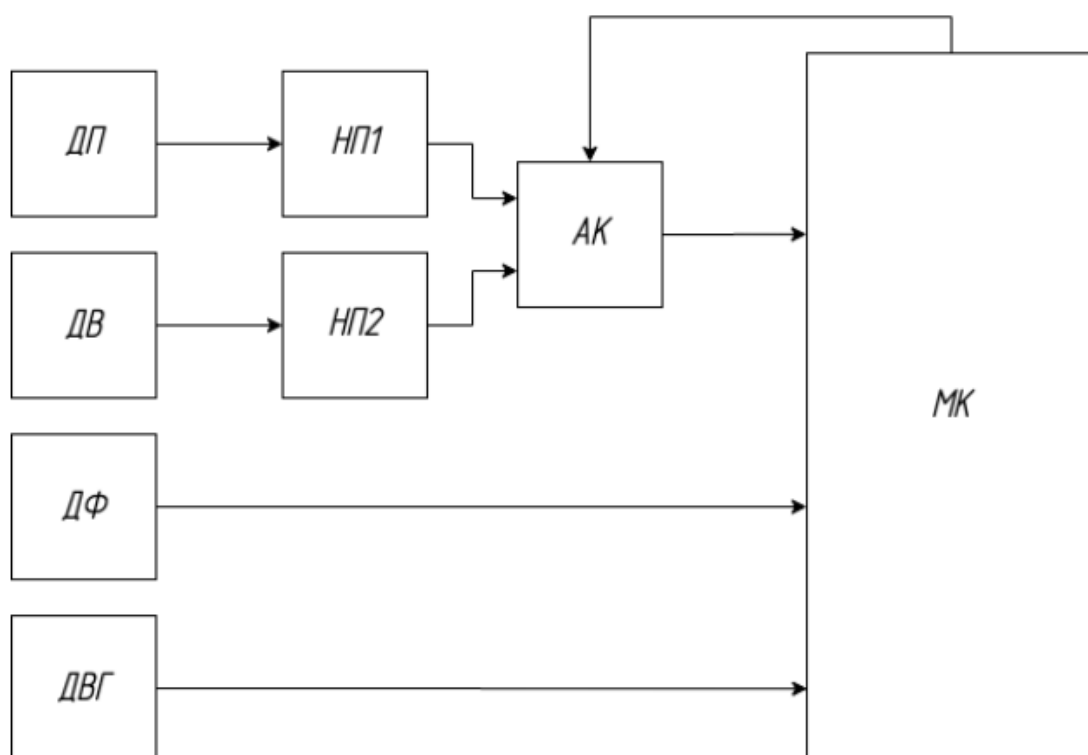


Рисунок 2.1 – Структурна схема системи контролю якості повітря в салонах літаків громадської авіації

На рис. 2.1 представлено наступні позначення: ДП – Давач пилу; ДВ – Давач вологості; ДФ – Давач формальдегіду; ДВГ – Давач вуглекислого газу; НП1 – Нормалізуючий пристрій 1; НП2 – Нормалізуючий пристрій 2; АК – Аналоговий комутатор; МК – Мікроконтролер.

Виходячи з технічного завдання, структурна схема повинна включати датчики, які вимірюють такі параметри, як концентрація пилу, концентрація формальдегіду, вуглекислого газу та вологості, контролер, який є центральною частиною, яка розпізнає, обробляє відповідно до відповідного алгоритму, використовується вихідний код для представлення результатів, пристрої, які нормалізують рі-

вень сигналу від виходу датчика до входу АЦП, і перемикач, який є аналоговим. Вихідним сигналом датчика концентрації пилу є напруга, але цей сигнал необхідно посилити, для цього канал необхідно підключити до нормуючого пристрою. Знижений сигнал надсилається на аналоговий перемикач, який потім направляє сигнал на входи контролера. Оскільки вихідним сигналом датчика вологості є струм, а вхідним сигналом АЦП є напруга, то цей канал повинен мати пристрій, що нормалізує вхідний і вихідний сигнали. Результуючий сигнал потім вводиться на аналоговий комутатор, оскільки еволюція сигналу дуже повільна. Тільки після цього моменту сигнал надходить на входи контролера. Датчики формальдегіду та вуглекислого газу надсилають цифрові сигнали на цифрові входи контролера.

2.2 Розрахунок нормуючого пристрою для датчика вологості

Оскільки вихідним значенням датчика вологості є постійний струм 5-20мА, то вхідним значенням АЦП, який входить до складу контролера, є напруга, тому наш пристрій з нормуючою функцією (НН) буде служити " перетворювач струм - напруга. Рисунок 2.2 ілюструє схему перетворювача.

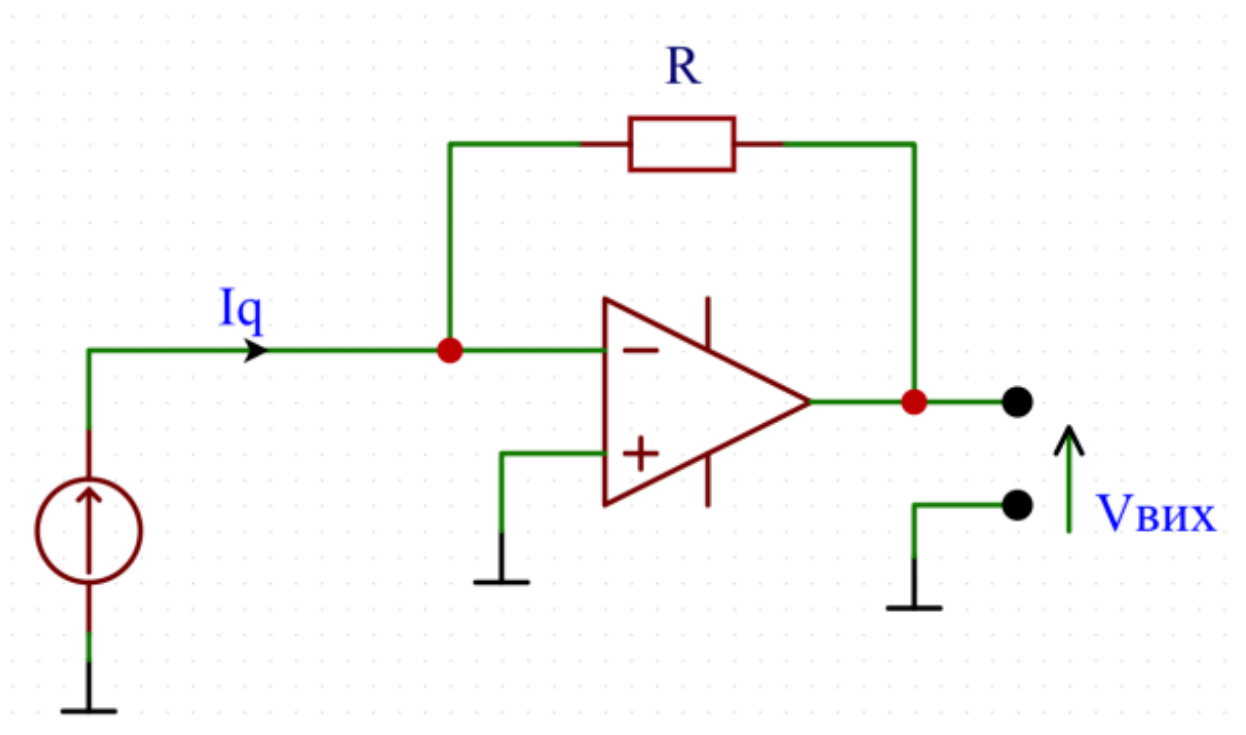


Рисунок 2.2 – Схема перетворювача струму

Вихідна напруга НП $U_{\text{вих}}$ має залежати від струму датчика I_q як:

$$U_{\text{вих}} = I_q \cdot R \quad (2.1)$$

Точність процесу перетворення оцінюється за величиною відхилення від номінального значення, яке відноситься до невизначеності опору зворотного зв'язку, значення цього опору потім множиться на 100. Тобто значення номінального опору R може відрізнятись на $\pm \delta \cdot 100\%$ ($\delta = \Delta R / R$). Якщо ми зробимо наступний крок, ми повинні знайти максимальне та мінімальне значення відхилення від номінального значення R : $\Delta R = \delta \cdot R / 100$. Допускається рівномірний розподіл можливих значень відхилення від номінального значення.

Точність перетворення, яка характеризується невизначеністю, визначається потенційним розкидом значень опору, а саме: $U(R) = \Delta R / \sqrt{3}$. Оскільки номінальне значення $I_q = 20$ мА, вибираючи $R = 250$ Ом, ми забезпечити номінальну вхідну напругу АЦП зі значенням 5В. Вибираємо резистор зворотного зв'язку з величиною $\pm 0,5\%$, тоді відносна невизначеність перетворення «струм – напруга» буде: $U(V) = 0,5 / \sqrt{3} = 0,3\%$, що повністю задовольняє вимірювання вологості, який написано в завданні.

2.3 Розрахунок нормуючого пристрою для датчика пилу

Реальна вихідна потужність датчика становить 80 мВ. Щоб привести вхідне значення АЦП до номінального рівня, необхідно виконати підвищення напруги в 62,5 рази. Вхідний опір датчика 1 кОм має бути високим. Ось чому ми обираємо стратегію. Рис. 2.3 ілюструє склад перетворювача.

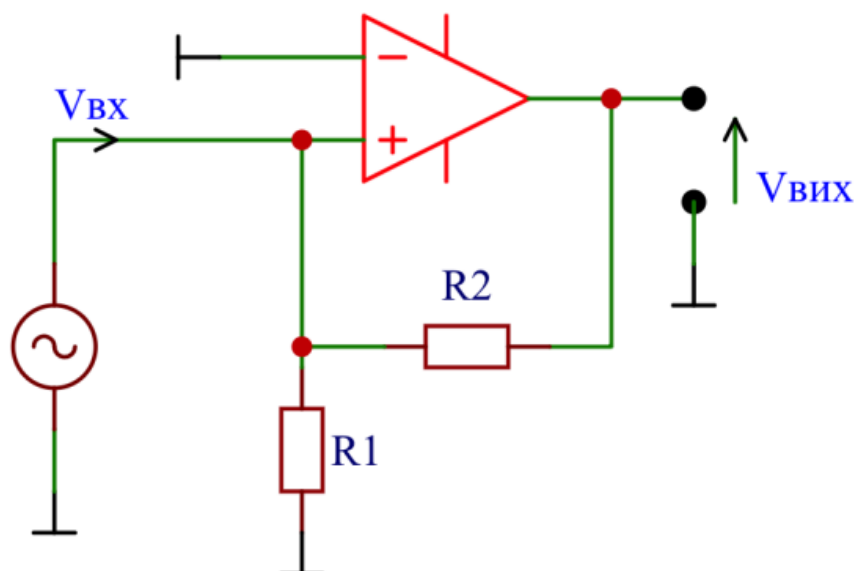


Рисунок 2.3 – Схема перетворювача напруги

$$K_V = \frac{U_{\text{ВИХ}}}{U_{\text{ВХ}}} = \frac{R_1 + R_2}{R_1} \quad (2.2)$$

Щоб оцінити вірність перетворення, запишемо вираз (2.2) у вигляді:

$$K_V = 1 + \frac{R_2}{R_1} \quad (2.3)$$

Так як $R_2 \gg R_1$, то приблизно $K_V \approx R_2 / R_1$, то похибка перетворення масштабу дорівнює [9]:

$$U(K_V) = \sqrt{C_{R1}^2 \cdot U_{R1}^2 + C_{R2}^2 \cdot U_{R2}^2} \quad (2.4)$$

Де $CR1 = \partial K_V / \partial R1$, $CR2 = \partial K_V / \partial R2$ – коефіцієнти, що відображають вплив неточності відповідно на шкалу неточності $CR1 = 1 / R1$ та $CR2 = 1 / R2$. Давайте виберемо резистор із величиною 0,5%, тоді: $UR1 = 0,2 / \sqrt{3} = 0,3\%$ Підставляємо значення $R1$, $R2$ і обчислюємо залишок від ділення (KV).

$$V(KV) \approx 1,017 (2.5)$$

2.4 Висновок до розділу 2

У другому розділі бакалаврської дипломної роботи розроблена структурна схема засобу контролю параметрів мікроклімату в салоні літаків громадської авіації. В процесі роботи було досягнуто таких результатів:

- Аналіз вимог та нормативів: Було проаналізовано існуючі стандарти та нормативи щодо параметрів мікроклімату в салонах літаків, зокрема температури, вологості, тиску та рівня CO₂.

- Вибір сенсорів та обладнання: Було здійснено відбір необхідних датчиків і пристроїв для вимірювання параметрів мікроклімату, включаючи температурні сенсори, гігрометри, барометри та датчики вмісту CO₂. Особлива увага була приділена точності, швидкодії та надійності обраного обладнання.

- Розробка структурної схеми: Створена структурна схема засобу контролю, яка включає в себе сенсори, модуль збору даних, обчислювальний блок для обробки інформації та модуль інтерфейсу для взаємодії з користувачем. Було передбачено можливість інтеграції з бортовою системою літака та зовнішніми інформаційними системами.

Таким чином, розроблена структурна схема засобу контролю параметрів мікроклімату забезпечує необхідний рівень комфорту та безпеки для пасажирів та екіпажу літаків громадської авіації. Впровадження цієї системи дозволить покращити якість обслуговування та підвищити загальний рівень безпеки польотів.

3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОЇ ЧАСТИНИ ЗАСОБУ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ У САЛОНІ ЛІТАКІВ ГРОМАДСЬКОЇ АВІАЦІЇ

3.1 Огляд давачів для вимірювання показників якості повітря

ESP82 NodeMcu V3. Платформа NodeMCU є платформою з відкритим кодом для IoT (рис. 3.1). Він охоплює мікропрограму, що працює на SoC ESP8266 Wi-Fi від Espressif Systems, а також апаратне забезпечення на основі компонента ESP-12. За домовленістю термін «NodeMCU» використовується для опису програмного забезпечення, а не набору для розробки. Прошивка написана на мові програмування Lua. Він походить від проекту eLua та базується на Espressif Non-OS SDK для ESP8266. Він містить багато відкритих проєктів, таких як luacjson і SPIFFS.

NodeMCU було випущено після випуску ESP8266. 30 грудня 2013 року компанія Espressif розпочала виробництво ESP8266. ESP8266 — це SoC на основі Wi-Fi із вбудованим ядром Tensilica Xtensa LX106.

Оскільки Arduino.cc починає створювати нові плати на основі процесорів, що не належать до AVR, таких як мікроконтролери ARM/SAM, що використовуються в Arduino Due, їм потрібно змінити Arduino IDE, щоб мати можливість легко перейти на нову платформу. дозволено для цих нових комп'ютерів Compile Arduino C/C++. Вони досягли цього, представивши Board Manager і aSAM Core. «Базовий» — це набір програмних компонентів, необхідних для керування платою Arduino та ID для створення вихідних файлів Arduino. C/C++ є рідною мовою передбачуваного MCU. Деякі ентузіасти ESP8266 створили ядро для ESP8266 WiFi SoC, яке часто називають «Ядром ESP8266 для Arduino IDE». Це стала найпопулярнішою платформою для роз-

робки програмного забезпечення для багатьох проектів на основі ESP8266, включаючи NodeMCU.

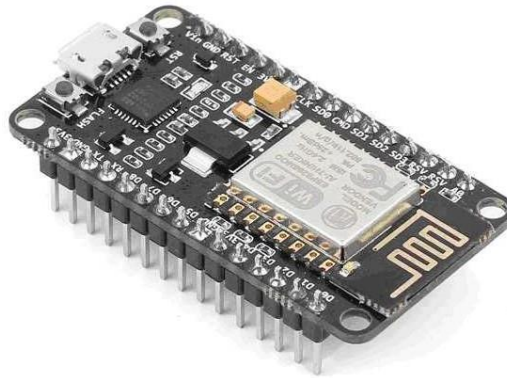


Рисунок 3.1 – ESP8266 NodeMcu V3

Особливості платформи: VCC, напруга (+3,3 В; може працювати до 3,6 В); GND, земля (напруга 0); RX, отримання даних щодо X; TX, передає перший біт даних X; CH_PD, припинить живлення; RST, перезапуск; GPIO 0, вхід загального призначення №0; GPIO 2, номер введення/виведення загального призначення №2.

ESP8266 — це недорогий мікрочіп Wi-Fi із повним стеком TCP/IP і мікроконтролером, виготовлений компанією Espressif Systems у Шанхаї, Китаї.

Західні виробники вперше помітили чіп у серпні 2014 року за допомогою модуля ESP-01 стороннього виробника Ai-Thinker. Цей невеликий модуль дозволяє мікроконтролеру підключатися до мережі Wi-Fi і створювати просте з'єднання TCP/IP за допомогою команд у стилі Хейс. Однак спочатку було мало документації англійською мовою про чіп і команди, які він приймав. Дуже низька ціна та той факт, що модуль має небагато зовнішніх компонентів, свідчить про те, що він може виявитися дуже дешевим з точки зору

розміру, що привабить багатьох хакерів для вивчення модуля, мікросхеми та програмного забезпечення на ньому та перекладу китайської документації.

ESP8266 з 1 МБ вбудованої флеш-пам'яті підтримує однокристалні пристрої з підтримкою Wi-Fi. Наступником цих чіпів мікроконтролерів став ESP32, випущений у 2016 році.

Плати покоління V1 і V2 можна розрізнити за такими ознаками, вони бувають різних розмірів. Крім того, у другому поколінні є вдосконалення чіпа ESP-12 і 4 МБ флеш-пам'яті. Перша версія застаріла і оформлена у вигляді яскраво-жовтої платформи. Користуватися незручно, оскільки охоплює 10 виходів макета. Плата другого покоління була розроблена, щоб усунути цей недолік - вона вузча і вихід ідеально підходить до контактів плати. Плати V3 зовні нічим не відрізняються від V2, вони мають більш надійний вихід USB. LoLin виготовляє плату V3, яка відрізняється від попередньої плати тим, що один із двох виходів зарезервовано для додаткової землі, а інший – для живлення через USB. Крім того, дошка має більші розміри порівняно з попередніми видами.

Живлення чіпа NodeMcu: 5-18 вольт через отвір Vin; 5 В підключається через порт USB або через контакт VUSB; Через вихід 3В. Переваги NodeMcu V3: Інтерфейс UART-USB має порт micro USB, який можна підключити до комп'ютера; 4 МБ флеш-пам'яті; Є можливість оновити прошивку через USB.

Можливість створювати скрипти в LUA і зберігати їх у файловій системі. Недоліки NodeMcu. Основним недоліком цієї плати є те, що вона може виконувати лише сценарії LUA, розташовані в оперативній пам'яті. Такого типу пам'яті недостатньо, розмір всього 20 Кб, тому написання більших скриптів може викликати багато проблем. По-перше, весь алгоритм необхідно розділити на лінійні фрагменти. Ці блоки повинні бути записані в окремі системні файли. Усі ці модулі виконуються за допомогою оператора dofile.

Розпіновку NodeMcu v3 показано на рис.3.2. Модуль V3 має 11 вхідних розеток загального призначення. Крім того, деякі результати мають додаткові цілі, окрім запланованих: D1-D10 - виходи з широтно-імпульсної модуляцією; D1, D2 - результати для інтерфейсу I²C /TWI; D5-D8 — сокети для протоколу SPI; D9, D10 – UART; A0 - вхід з АЦП.

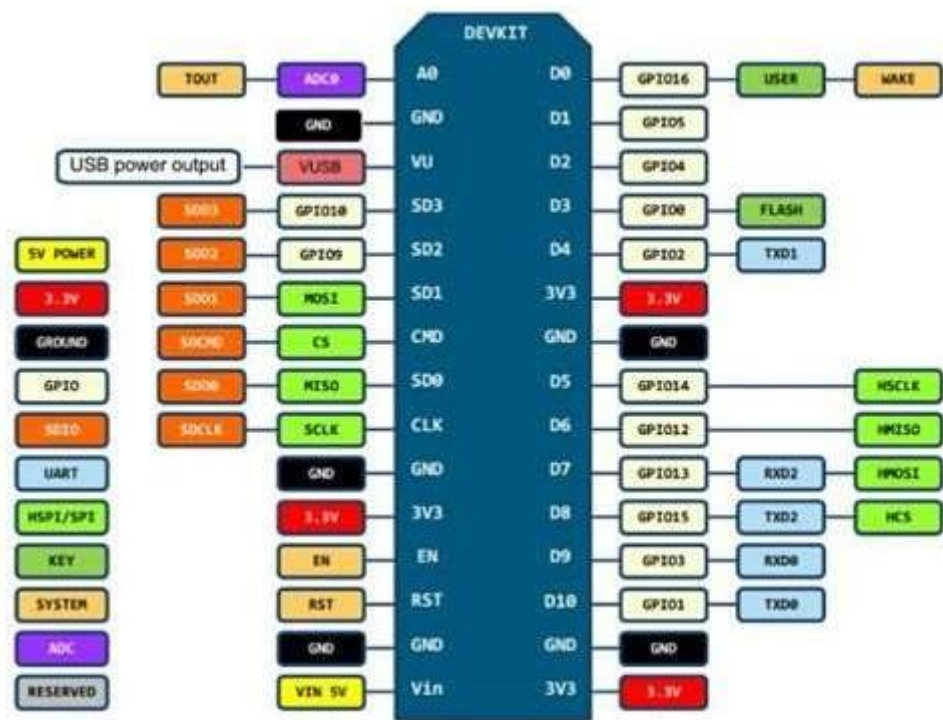


Рисунок 3.2 – Розпіновка NodeMcu v3

Датчик виявлення вуглекислого газу MH-Z19. Окрім цього, MH-Z19 (див. рис. 3.3) унікальний тим, що йому не потрібна певна напруга чи додаткове живлення, замість цього він може передавати дані через UART та ШІМ.



Рисунок 3.3 – Датчик вуглекислого газу MH-Z19

Hd –калібрування нуля. Калібрування непотрібне, якщо Hd= 7с; SR - не працює; Tx - напруга - 3,3В; Rx становить 3,3 В, а також працює з 5 В; Vo - максимальна вихідна напруга становить 3,3 В, а потрібно не більше 10 мА; Gnd -земля; Vin - напруга живлення 3,6 - 5,5В.

UART дозволяє запитувати концентрацію CO₂ і виконувати два різні калібрування. Після ознайомлення з документацією на датчик перехід в робочий режим займає близько трьох хвилин. При першому включенні він вироблятиме концентрацію 5000 або 400 000 частин на мільйон. Реакція датчика на збільшення CO₂ сповільнюється приблизно на одну хвилину. Якщо концентрація перевищує 5000 частин на мільйон, ви використовуєте його протягом тривалого періоду часу в одну хвилину. Деякий час це дасть неточні показання, що призведе до зниження рівня CO₂.

SHT31 Температура & SHT31-D вологаЦифровий модуль датчика температури та вологості SHT-31D відкриває новий рівень вимірювальних технологій. Функції датчика включають схеми обробки та підсилення сигналу, блоки калібрувальної пам'яті, схеми АЦП і скидання живлення, а також широкую напругу живлення від 2,5 до 5,5 В. Цей модуль можна використовувати

з Arduino або Raspberry Pi. Особливості: діапазон вимірювання відносної вологості: 0-100%; діапазон відносної вологості: 2%; Позначення інтерфейсу: I2C; Роздільна здатність: 16 біт; Точність: +/- 0,3 C; Напруга живлення: 2,5-5,5 В; Робоча напруга: 800 вА; Робоча температура: від -40 до +125°C.

Датчик Nova PM SDS011 визначає запилення. SDS011 використовує принцип лазерного розсіювання для отримання концентрації частинок 0,3~10 мкм у повітрі (див. рис. 2.4). Цифровий вихід, вбудований вентилятор, стабільний і надійний.

Висока роздільна здатність: 0,3 мкг/м³. Як працює лазерне розсіювання:

Частинки викликають розсіювання світла, коли вони проходять через область виявлення. Розсіяне світло перетворюватиметься на електричні сигнали, а ці сигнали посилюватимуться та оброблятимуться.

Кількість і діаметр частинок можна визначити шляхом аналізу, оскільки форма сигналу пов'язана з діаметром частинок.

Протокол послідовного зв'язку: 9600 8N1. (Частота — 9600 бод, кількість парності — 0, а стоп-біт — 1).

Формат повідомлення циклічного звіту: кадр даних 1+0,5 секунди (10 байтів): голова повідомлення + послідовність + дані (6 байтів) + контрольна сума + хвіст повідомлення.



Рисунок 3.4 – Nova PM датчик пилу SDS011

HDC1080 CCS811 хімічний склад вуглекислого газу CO₂. Цей модуль поєднує два різних датчики: CCS811 і HDC1080 на одній платі (див. рис. 3.5). Відстань між двома датчиками можна виміряти окремо за допомогою ПС. HDC1080 — малопотужний, високоточний, маломасштабний цифровий вимірювач температури та вологості.

CCS811 — малопотужний мініатюрний датчик газу, який вимірює якість повітря. Інтегровані датчики оксиду металу (MOX), які вимірюють загальну кількість органічних парів, еквівалентну eTVOC, і загальну кількість CO₂, еквівалентну eCO₂. Мініатюрний комп'ютер MCU вбудований в чіп. Значення вмісту газу, визначене CCS811, автоматично регулюється шляхом зчитування значення температури та вологості датчика HDC1080 і запису його у внутрішній вимірювач температури та вологості CCS811.

Параметри HDC1080: Робоча температура: від -20 градусів Цельсія до 70 градусів Цельсія; Діапазон вимірювання датчика вологості: 0-100% відносної вологості; Точність датчика вологості: похибка становить 14 цифр із похибкою 2%; Відтворюваність вологості: $\pm 1\%$ відносної вологості; Час, необхідний для активації датчика вологості, становить 15 секунд; Час перетворення вологості: 8 біт 2,5 мілісекунди, 11 біт 3,85 мілісекунди, 14 біт 6,5 мілі-

секунди; Час перетворення температури: 11-біт 3,85 мс, 14-біт 6,5 мс; Частота зв'язку: 400кГц; Робоча адреса ІС: 1000000х (х байт керування читанням/записом). Джерело живлення: очікується 3,3 В.

Параметри CCS811: Рекомендована напруга для джерела живлення становить 3,3 В; Потужність: максимум 60 мВт; Частота зв'язку: 400кГц; Діапазон виявлення eTVOC: 0 ~ 3,76 мільйона частин на мільярд; Межа виявлення eCO₂: 400-840 ppm.

Сенсор CCS811 предствлений на рис. 3.5

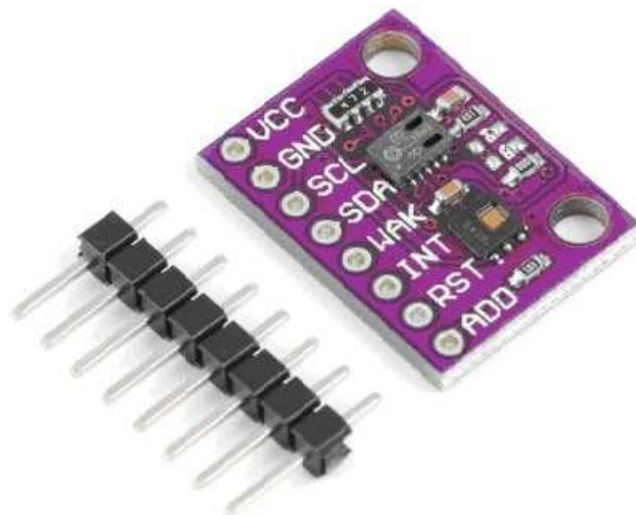


Рисунок 3.5 – HDC1080 CCS811 чадний газ

3.2 Вибір оптимального програмного середовища для вимірювання показників якості повітря

Аналіз системи збору інформації (influxDB). InfluxDB — це база даних із мітками часу, яка призначена для обробки великої кількості записів і запитів. Це невід’ємна частина купи TICK. InfluxDB призначений як доповнення до сховища, яке підтримує великі обсяги даних із мітками часу, які

стосуються різних випадків використання, включаючи моніторинг DevOps, показники додатків, датчики Інтернету речей та аналіз у реальному часі.

Ось список функцій, які зараз підтримує InfluxDB, це ідеально підходить для роботи з даними часових рядів.

Спеціально створене високопродуктивне сховище даних, призначене для даних часових рядів. Механізм TSM забезпечує швидку передачу та введення даних.

Повністю написано на Go. Його об'єднано в один бінар, якому не вистачає зовнішньої допомоги. Базовий, ефективний API для написання та запиту HTTP.

Підтримка плагінів призначена для полегшення прийому додаткових протоколів даних, зокрема Graphite, Collectd і OpenTSDB.

SQL-подібна мова запитів, яка призначена для легкого запиту зведеної інформації. Теги сприяють швидкому та ефективному виконанню запитів. Безкоштовна вихідна версія InfluxDB реалізована на одній машині. Перш ніж ініціювати процес InfluxDB, ви повинні зрозуміти фундаментальні концепції бази даних (див. Таблицю 3.1).

Таблиця 3.1 - Ключові терміни бази даних InfluxDB

Поняття	Визначення
aggregation	Функція InfluxQL, яка повертає загальне значення для набору точок. Повний список доступних і майбутніх комбінацій.
Batch	Набір значень даних у форматі рядка протоколу influxDB, розділених символами нового рядка (0x0A). Пакет позначок можна надіслати до бази даних за допомогою одного запиту до місця запису. Це підвищує ефективність записів, створених за допомогою API influxDB, зменшуючи кількість HTTP-трафіку.InfluxData

continuous query(CQ)	Запит InfluxQL, який виконується автоматично та періодично до бази даних. Для постійних запитів потрібна процедура в реченні SELECT, яка має включати GROUP BY time()пункт.
Database	Логічне зберігання користувачів, стратегії збереження, постійні запити та дані на основі часу.
Duration	Атрибут політики, який визначає період часу, протягом якого InfluxDB зберігає дані. Дані, які перевищують максимальну тривалість, автоматично видаляються з бази даних.
Field	Пара ключ-значення в структурі даних influxDB, яка зберігає інформацію про дані та фактичне значення. Поля є необхідними в структурах даних influxDB і не індексуються - запити щодо значень полів досліджують усі точки за певний період часу, як наслідок, вони неефективні щодо тегування.
Function	Збитки, селектори та перетворення InfluxQL.
Identifier	Токени, пов'язані з постійними запитами, іменами баз даних, іменами полів, іменами вимірювань, іменами політики збереження, іменами підписок, іменами тегів та іменами користувачів.
Node	Самостійний influx процес
Now()	Наносекундна мітка локального сервера.
Query	Операція, що отримує дані з InfluxDB

SQL Server не призначений для обробки даних із міткою часу. Бази даних SQL Server здатні обробляти часові ряди, але не розроблені спеціально для цієї мети. Загалом, InfluxDB призначений для швидкого збору великих обсягів даних часових рядів і аналізу їх у режимі реального часу.

В InfluxDB позначка часу визначає одну точку в будь-якому даному ряді даних. Це як таблиця бази даних SQL, де первинний ключ попередньо встановлений системою та завжди має час.

InfluxDB також усвідомлює, що параметри схеми можуть змінюватися з часом. InfluxDB не вимагає попередньо визначеної схеми. Точка даних може мати одне поле вимірювання, усі поля вимірювання або будь-яке число між ними. Ви можете додати нове поле до опитування, просто записавши для нього бал. Якщо вам потрібне пояснення термінів вимірювання, міток і полів, вам слід звернутися до наступного розділу бази даних SQL, щоб перекласти термінологію InfluxDB.

InfluxQL і SQL. InfluxQL — це SQL-подібна мова запитів для взаємодії з InfluxDB. Він розроблений, щоб відчувати себе знайомим для людей, які працюють з іншими SQL або подібними середовищами SQL, і забезпечує функціональність, специфічну для зберігання та аналізу даних часових рядів.

Оператори InfluxQL мають форму SELECT оператора SQL SELECT (див. рис. 3.6):

```
SELECT <stuff> FROM <measurement_name> WHERE <some_conditions>
```

Рисунок 3.6 – Заява InfluxQL

```
SELECT * FROM "foodships"
```

Рисунок 3.7 – Вихід у розділ вище

InfluxQL також підтримує регулярні вирази, арифметику у виразах, оператори SHOW і оператори GROUP BY. Перегляньте нашу сторінку дослідження даних для детального обговорення цих тем. Функції InfluxQL включають COUNT, MIN, MAX, MEDIAN, DERIVATIVE тощо.

Коригування дизайну InfluxDB InfluxDB — це база даних часових рядів. Оптимізація для цього випадку використання вимагає певних компромісів, насамперед для покращення продуктивності за рахунок функціональності. Нижче наведено деякі конструктивні міркування, які слід зважити:

Для випадку використання часових рядів ви можете припустити, що якщо ті самі дані надсилаються кілька разів, це ті самі дані, які клієнт щойно надіслав кілька разів.

Переваги: спрощене вирішення конфліктів покращує продуктивність журналювання.

Недоліки: неможливо зберегти дублікати даних, у рідкісних випадках вони можуть бути перезаписані.

Видалення відбувається рідко. Коли вони трапляються, вони майже завжди стосуються великої кількості старих даних у записах.

Переваги: обмеження доступу до видалень може покращити ефективність запитів і запису.

Мінуси: Функція видалення досить обмежена.

Оновлення наявних даних трапляються рідко, а суперечливих оновлень не буває. Дані часових рядів — це переважно нові дані, які ніколи не оновлювалися.

Переваги: обмеження доступу до оновлень може підвищити ефективність запитів і журналювання.

Недоліки: Функціональність оновлення дуже обмежена.

Переважна більшість записів — це дані з останньою міткою часу, і дані додайте в порядку зростання.

Переваги: додавання даних за часом у порядку зростання ефективніше.

Недоліки: Запис балів у випадковому або незростаючому часу набагато менш ефективний.

Масштаб має значення. База даних повинна мати можливість обробляти великі обсяги читання та запису.

Плюси: база даних може обробляти великі обсяги читання та запису.

Мінуси: команда розробників InfluxDB була змушена піти на компроміси, щоб покращити продуктивність.

Уміння писати та запитувати дані важливіше, ніж мати добре вирівняне подання.

Переваги: Написання та надсилення запитів до бази даних можуть виконуватися декількома клієнтами під великим навантаженням.

Недолік: якщо база даних заповнена, відповідь може не включати остаточний компонент.

Багато часових рядів є тимчасовими. Часто є серії даних, які тривають лише кілька годин, а потім зникають, наприклад, новий хост починає звітувати на деякий час, а потім вимикається.

Плюси: InfluxDB ефективно обробляє помилкові дані.

Недоліки Безсхемному дизайну не вистачає підтримки певних функцій бази даних, включаючи відсутність асоціацій між таблицями.

Ніщо не є надто значущим.

Плюси: InfluxDB має безліч потужних інструментів, які полегшують поєднання агрегованих даних і великих наборів даних.

Недоліки: окуляри не мають традиційної сертифікації, відрізняються міткою часу та серією.

Охарактеризуємо інформаційну систему візуалізації Grafana.

Grafana — це платформа з відкритим кодом для візуалізації, моніторингу та аналізу даних. Цей інструмент у поєднанні з Graylog є частиною нашої двосторонньої системи моніторингу поведінки користувачів і продуктивності системи. Grafana дозволяє користувачам створювати інформаційні панелі з панелями, кожна з яких відображає певні показники за певний час.

Кожна інформаційна панель є багатофункціональною, тому її можна налаштувати для конкретного проекту або будь-якого розвитку та/або потреб бізнесу. Редактор запитів і спеціальний синтаксис надаються в Grafana для кожного джерела даних.

Основними елементами візуалізації обраного індикатора є панелі. Grafana підтримує панелі з графікою, статусом одиниць, таблицями, тепловими картами клацання та довільним текстом, а також інтеграцію з офіційними та створеними спільнотою плагінами (такими як карти світу чи годинники) і програмами візуалізації. Ви можете налаштувати стиль і формат кожної панелі; усі панелі можна перетягувати в нові місця, змінювати порядок і розмір.

Інформаційна панель — це набір незалежних панелей, розміщених у сітці, яка містить набір змінних (наприклад, імена серверів, програм і датчиків). Змінюючи змінні, ви можете перемикає дані, які відображаються на інформаційній панелі (наприклад, дані з двох окремих серверів). Інформаційні панелі можна налаштовувати, а представлені в них дані можна сегментувати та фрагментувати відповідно до потреб користувача. Проект Grafana має велику спільноту розробників коду та користувачів, тому є багато готових інформаційних панелей для вибору різних типів даних і джерел.

На інформаційній панелі ви можете використовувати анотації для відображення певних подій на різних панелях. Анотації додаються через запити, налаштовані в Elasticsearch; на діаграмі відображаються вертикальні червоні лінії. Коли ви наводите курсор на анотацію, ви можете отримати опис події та теги, наприклад, відстеження відповідей сервера з кодами помилок 5xx або перезавантаження системи. Тому ви можете легко зіставити час, конкретні події та їхні наслідки у своїй програмі та дослідити поведінку системи.

Як використовувати Grafana у користувацькій веб-програмі Інструмент Graylog використовується для зберігання, керування журналами

веб-програми та моніторингу її продуктивності на етапі розробки та під час виконання. Grafana — це інструмент, який перетворює журнали, що зберігаються в Graylog, у візуальну форму для аналізу та моніторингу системи. В одному з наших поточних проектів інструмент Grafana, назовемо його інтерфейсом користувача, використовується для завантаження та моніторингу продуктивності веб-додатків і трафіку клієнтів.

Інструменти Graylog і Grafana відокремлені один від одного. Оскільки Graylog зберігає всі дані журналу в Elasticsearch, одному з джерел даних Grafana, підключіть Grafana до Graylog за допомогою спеціального індексу в Elasticsearch, який зберігає журнали.

Операційна система Raspberry Pi3 походить від системи Linux.

Майже всі операційні системи Raspberry Pi використовують Linux як базову операційну систему. Існує кілька офіційно підтримуваних операційних систем:

Raspbian є офіційною операційною системою і має все необхідне програмне забезпечення. Рекомендується використовувати саме цю операційну систему, щоб ознайомитися з Raspberry Pi.

OpenELEC — це медіаплеєр із відкритим кодом на базі Linux.

OSMC (Raspbmc) — це безкоштовний медіапрогравач, який є похідним від Kodi Media Center і Debian GNU/Linux.

Можливості Windows 10 на пристроях Raspberry Pi 2.

Усі операційні системи, перелічені в NOOBS, є програмами, які включають усі перераховані операційні системи, вони дозволяють легко інстальювати їх на ваш Raspberry Pi. За допомогою NOOBS ви можете встановити одну або кілька операційних систем на свій комп'ютер і вибрати потрібну систему під час запуску комп'ютера. Під час встановлення операційної системи додається спеціальний розділ Recovery, який дозволяє повернутися до початкової конфігурації системи.

Запуск кількох екземплярів однієї операційної системи на Raspberry Pi можливий трьома різними способами.

Придбайте SD-карту з попередньо встановленим Raspbian або NOOBS.

Перший спосіб найпростіший - тут більше нічого робити не потрібно. Для другого випадку процес установки виглядає наступним чином:

Спочатку карту пам'яті необхідно відформатувати, вказавши файлову систему FAT32.

Завантажте архів з NOOBS і розпакуйте його на карту пам'яті, щоб файли були в кореневому каталозі.

Підключіть до Raspberry Pi необхідну периферію: клавіатуру, мишу та монітор, підключені через USB, блок живлення. Як монітор можна використовувати телевізор, підключений через RCA.

Якщо екран підключено через RCA, натисніть «3» на клавіатурі.

У вікні виберіть операційну систему Raspbian, російську розкладку клавіатури та мову. Неможливо встановити українську мову. Натисніть «Встановити» і дочекайтеся завершення процесу інсталяції та увімкніть комп'ютер.

У третьому пункті меню «Інструменти налаштування» виберіть другий параметр для встановлення графічного інтерфейсу LXDE.

Натисніть «Готово» і дочекайтеся перезавантаження.

Якщо вам потрібно увійти під логіном і паролем, введіть рі в поле для входу, пароль raspberry і відкриється робочий стіл.

Вузол-RED

Node-RED — це програмна платформа, яка полегшує підключення апаратних пристроїв, API та онлайн-сервісів новими способами.

Він надає редактор на основі браузера, який полегшує підключення потоків до кількох вузлів у палітрі, яку можна розширити для виконання одним клацанням миші.

Node-RED надає редактор потоків на основі браузера, який дозволяє легко підключати потоки за допомогою різних вузлів на панелі. Потім просто натисніть кнопку, щоб згорнути потік до середовища виконання.

Функції JavaScript можна створювати в редакторі за допомогою редактора форматowanego тексту, а вбудовані бібліотеки дозволяють зберігати корисні функції, шаблони або потоки для повторного використання.

Потоки, створені в Node-RED, зберігаються за допомогою JSON і можуть бути легко імпортовані та експортовані, щоб поділитися з іншими. Бібліотека веб-потоків дозволяє ділитися своїми найкращими потоками зі світом.

Node-RED побудовано на Node.js, повністю використовуючи переваги його неблокуючої моделі, керованої подіями. Це робить його ідеальним для роботи на недорогому апаратному забезпеченні на межі мережі, такому як Raspberry Pi, а також у хмарі.

Процес встановлення Node-RED дуже простий. Після створення сервера увійдіть за допомогою SSH як користувач root. Ми оновлюємо вихідні коди та пакети програми до актуальної версії:

```
apt-get оновлення && apt-get -у оновлення
```

Ми встановили полегшену графічну оболонку LXDE, а щоб полегшити роботу з сервером з будь-якої машини Windows, ми встановили чудовий інструмент віддаленого керування xrdp:

```
apt-get -у встановити lxde lxde-core lxde-common xrdp
```

Тепер ви можете підключитися до сервера за допомогою стандартного клієнта Windows RDP. Зауважте, що якщо Gnome встановлено замість LXDE, xrdp не працюватиме без додаткових налаштувань. Ви також можете налаш-

тувати Midori як свій браузер - це набагато швидше, ніж стандартний браузер. Нам також потрібен менеджер пакетів NodeJS і сам механізм: `apt-get -y install midori npm nodejs-legacy` Розробники Node-RED рекомендують оновити npm до останньої версії перед установкою. Ми дотримувалися такої поради: `npm install -g npm@2.x` Тепер ви можете встановити Node-RED за допомогою однієї команди. Під час процесу інсталяції можуть виникнути помилки залежності node-gyp, але це не важливо для роботи Node-RED і може бути проігноровано: `npm install -g --unsafe-perm node-red` Після завершення інсталяції ми можемо відключитися від SSH Connect і повністю увійдіть на сервер за допомогою RDP. У Windows просто натисніть Win + R, введіть mstsc і натисніть Enter. У вікні, що відкриється, введіть IP-адресу сервера VScale і підключіться до сервера.

Відкрийте LXTerminal і запустіть середовище Node-RED за допомогою команди

Вузли червоні.

Відкрийте Midori та в адресному рядку вкажіть: `http://127.0.0.1:1880` Якщо ви не вкажете інші параметри для брандмауера, програма також буде доступна за адресою самого сервера. Але в рамках цього посібника ми не будемо розглядати це питання.

Інтерфейс складається з трьох основних частин: списку елементів, розташування елементів і властивостей об'єкта. Взявши елементи зі списку та перетягнувши їх у логіку програми, ви можете створити складні елементи, які представляють функціональність програми. Візьмемо Raspberry PI 3 с Raspbian Jessie OS як пристрій IoT. Node-RED працює з коробки, тому ми просто запускаємо його. Давайте спробуємо надіслати дані між запущеними середовищами програмування. За допомогою цього прикладу можна збирати будь-які дані, надані пристроями IoT. Ви можете підключити будь-який датчик до тієї самої Raspberry PI, наприклад датчик витoku, освітлення, температура-

тури. У прикладі я збираюся використовувати чудовий елемент під назвою Inject. Це дозволяє замінювати будь-які дані без використання реального пристрою та є важливим для налаштування.

Ми також використовуємо елементи TCP для з'єднань даних. Спочатку давайте налаштуємо наш «маленький малюнок». Виберіть елемент «Inject» із розділу «Input» і помістіть його в робочу область. Ми також розміщуємо туди елементи TCP із вихідної секції та з'єднуємо відповідні сірі точки, просто з'єднуючи їх один з одним. Ви також можете додати елементи Debug, щоб точно контролювати, що надсилається.

Будь-який елемент можна налаштувати подвійним клацанням. Спочатку ми визначаємо параметри для Inject. Наприклад, давайте надішлемо будь-який рядок на наш сервер VScale. Виберіть «Корисне навантаження - Рядок» і напишіть туди наш текст і натисніть «Готово».

Тепер давайте налаштуємо елементи TCP, щоб вказати, що ми підключимося до сервера та надішлемо йому дані. Заповніть IP-адресу сервера (XXX.XXX.XXX.XXX у прикладі) і порт (виберіть будь-який, переважно встановіть той самий порт на сервері).

Ми видаляємо параметр Debug за замовчуванням. Налаштовуємо всі параметри і запускаємо програму, натиснувши кнопку «Deploy». Тепер сплануємо серверну частину. Тут ми будемо використовувати два компоненти. З вхідного компонента ми виводимо TCP, а для візуального представлення ми виводимо проблемні аспекти вихідного компонента. Ми їх асоціюємо.

Тепер ми налаштовуємо елемент TCP, вказуючи той самий порт і вибираючи тип даних, які ми будемо отримувати.

Натисніть «Розгорнути», і через кілька секунд Raspberry Pi підключиться до нашого сервера. Під елементом з'явиться напис «Підключено».

Натиснувши синю кнопку ліворуч від елемента «Inject», ми надішлемо вказаний раніше рядок. На стороні сервера переходимо у вкладку «Debug» і ми побачимо рядки з текстом.

3.3 Висновки до розділу 3

У третьому розділі роботи були розглянуті та проаналізовані датчики, які використовуються, а саме:

- ESP82NodeMcu V3.
 - Датчик виявлення вуглекислого газу MH-Z19.
 - Температура і вологість вимірюються SHT31, а для останнього використовується окремий датчик.
 - Датчик РМ від Nova для пилку називається SDS011.
 - Програмні середовища, такі як InfluxDB, Node-RED, Grafana тощо.
- Інша доречна інформація також була розглянута.

В результаті третьої частини розробки ми отримали технічну основу для створення системи.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОБРОБКИ РЕЗУЛЬТАТІВ СПОСТЕРЕЖЕНЬ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ

4.1 Підключення та налаштування датчиків моніторингу повітря

По-перше, ми повинні зробити апаратне забезпечення доступним, підключивши DHT-11, HDC1080, MH-Z19 і SDS011 до нашого Arduino. Після підключення та перевірки активуйте живлення та створіть власну точку доступу для підключення до сервера. З цього моменту ви можете створити ім'я користувача та пароль (рис. 4.1).

ESP Easy Mega: B1

△Main ⊗Config **Controllers** ⚙Hardware 🔍Devices →Rules 📧Notification

Protocol: OpenHAB MQTT ▼

Locate Controller: Use IP address ▼

Controller IP: 10.245.76.13

Controller Port: 1883

Minimum Send Interval: 100 [ms]

Max Queue Depth: 10

Max Retries: 10

Full Queue Action: Ignore New ▼

Check Reply: Ignore Acknowledgement ▼

Client Timeout: 1000 [ms]

Controller User: openhabian

Controller Password:

Controller Subscribe: /%sysname%/#

Рисунок 4.1 – Створення точки доступу

Після створення точки доступу ми додаємо та налаштовуємо пристрої, з яких будемо отримувати покази, відображати значення, в даному випадку їх частота прийому 60 секунд. Рекомендовано не менше 30 секунд. і налаштувати пристрій (див. рис. 4.2).

31

Controllers Hardware **Devices** Rules Notifications Tools

Enabled	Device	Name	Port	Ctr (IDX)	GPIO	Values
✓	Environment - BMx280	BME280_room		① ②	GPIO-4 GPIO-5	TempBME280_room: 20.51 HumBME280_room: 41.35 PressBME280_room: 722.86
✓	Environment - BMx280	BME280_box		① ②	GPIO-4 GPIO-5	TempBME280_box: 20.50 HumBME280_box: 41.30 PressBME280_box: 722.80
✗	Dust - SDS011/018/198					PM2.5: 0.00 PM10: 0.00
✓	Gases - CO2 MH-Z19	MH-Z19		① ②	GPIO-15 GPIO-13	MH-Z19_PPM: 2000.00 MH-Z19_Temp: 19.00 MH-Z19_U: 0.00
✓	Environment - SHT30/31/35 [TESTING]	SHT31PRO_out		②	GPIO-4 GPIO-5	TempSHT31PRO_out: 3.97 HumSHT31PRO_out: 78.47

Рисунок 4.2 – Налаштування пристрою

Після завершення основних налаштувань і робочої частоти ми можемо продовжувати використовувати ресурси мережі. Щоб налаштувати їх, необхідно перейти на вкладку Контролер, обрати Редагувати та встановити необхідні ресурси. Наприклад, створимо контролер для OpenHAB MQTT (див. рис. 4.3).

ESP Easy Mega: B1

Main Config **Controllers** Hardware Devices Rules Notifications Tools

	Nr	Enabled	Protocol	Host	Port
Edit	①	✗	OpenHAB MQTT	io.adafruit.com	1883
Edit	②	✓	OpenHAB MQTT	10.245.76.13	1883

Рисунок 4.3 – Створюємо контролери

Щоб вони функціонували належним чином, налаштуємо контролери, встановивши їх власну адресу або ресурс (див. рис. 4.4).

Рисунок 4.4 – Задаємо адрес та вмикаємо контролер

Після встановлення початкової точки ми переходимо до використання бази даних для зберігання часових рядів InfluxDB. Починаємо процес (див. рисунок 4.5).

```
sudo systemctl start influxdb
```

Рисунок 4.5 – Початок роботи InfluxDB

Щоб служба продовжувала працювати після перезавантаження, необхідно ввести наступну команду (див. рис. 4.6).

```
sudo systemctl enable influxdb
```

Рисунок 4.6 – Процес перезавантаження

Робимо перевірку (див. рисунок 4.7).

```
influx
```

Рисунок 4.7 – Команда перевірки роботи сервісу

В разі успішного підключення сервісу на екрані буде наступне зображення (див. рисунок 4.8),

```
Connected to http://localhost:8086 version 1.3.5  
InfluxDB shell version: 1.3.5
```

Рисунок 4.8 – Успішне підключення

При новій (першій) установці InfluxDB потрібно створити базу даних (див. рисунок 4.9),

```
> create database testdb
```

Рисунок 4.9 – Створення нової бази даних

Результат створення бази даних показано на екрані (див. рисунок 4.10):

```
> show databases  
name: databases  
name  
----  
_internal  
testdb
```

Рисунок 4.10 – Результат створення БД

Обираємо створену базу даних (див. рис. 4.11).

```
> use testdb  
Using database testdb
```

Рисунок 4.11 – Вибір бази даних

На даний момент використано повну функціональність InfluxDB. Переходимо до наступного кроку у використанні Node-RED. Ми будемо використовувати Node-RED як тестову точку даних, яка деконструюватиме кори-

сне навантаження та впроваджуватиме його в InfluxDB, яку Grafana надсилатиме запит для створення інформаційної панелі.

Спочатку вам потрібно встановити вузол InfluxDB у Node-RED. Це можна зробити, перейшовши на панель адміністратора, вибравши вкладку «Встановлення» та ввівши пошуковий запит «InfluxDB». Ми встановимо сам вузол (див. рис. 4.12).

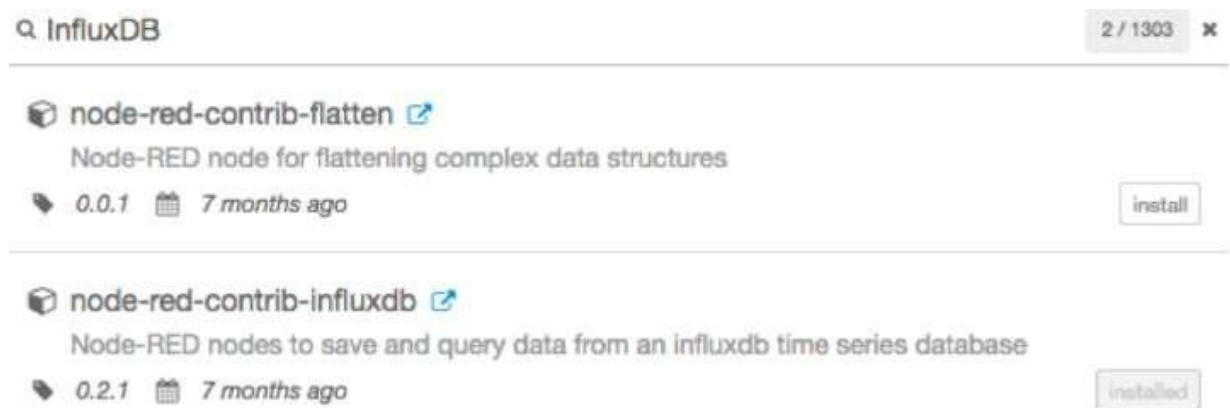


Рисунок 4.12 – Вузол InfluxDB в Node-RED

Процес імпортування створює кінцевий пункт призначення та функції для збору, перетворення та завантаження в InfluxDB. Це потрібно змінити, щоб відобразити поточну конфігурацію InfluxDB.

Адреса за замовчуванням – `http://127.0.0.1:8086`, а ім'я бази даних – «testdb» (див. рис. 4.13).

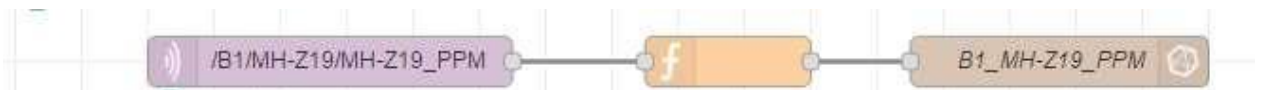


Рисунок 4.13 – Схематичне зображення створення передачі даних

Наведений приклад (4.13) демонструє, що ми створюємо передачу даних для всіх метрик (див. рис. 4.14).



Рисунок 4.14 – Передача даних

Візуалізація даних, що стосуються показників якості повітря

Починаємо оформляти приладову панель в Графана. Щоб досягти цього, ми повинні зробити наступне: Ми починаємо з Grafana (див. рис. 4.15).

```
sudo service grafana-server start
```

Рисунок 4.15 – Запуск Grafana

За замовчуванням Grafana запуститься на порту 3000.Переходимо в браузері на <http://host:3000/login>, і бачимо вікно(див. рисунок 4.16):



Рисунок 4.16 – Домашня сторінка сервісу

Увійдіть у головне меню Grafana та виберіть джерело даних (див. рис. 4.17):

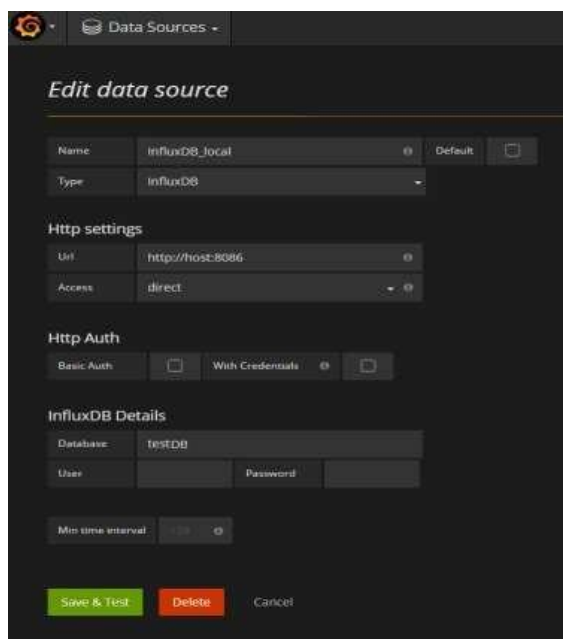


Рисунок 4.17 – Налаштування джерела даних

За замовчуванням інформаційна панель матиме назву «TestDB», тому натисніть кнопку «Імпортувати», щоб продовжити.

Тепер, коли у нас є дані в InfluxDB, нам потрібно створити інформаційну панель і налаштувати її. Поверніться на домашню сторінку, натисніть кнопку «Нова інформаційна панель» і заповніть наступне (див. рис. 4.18):

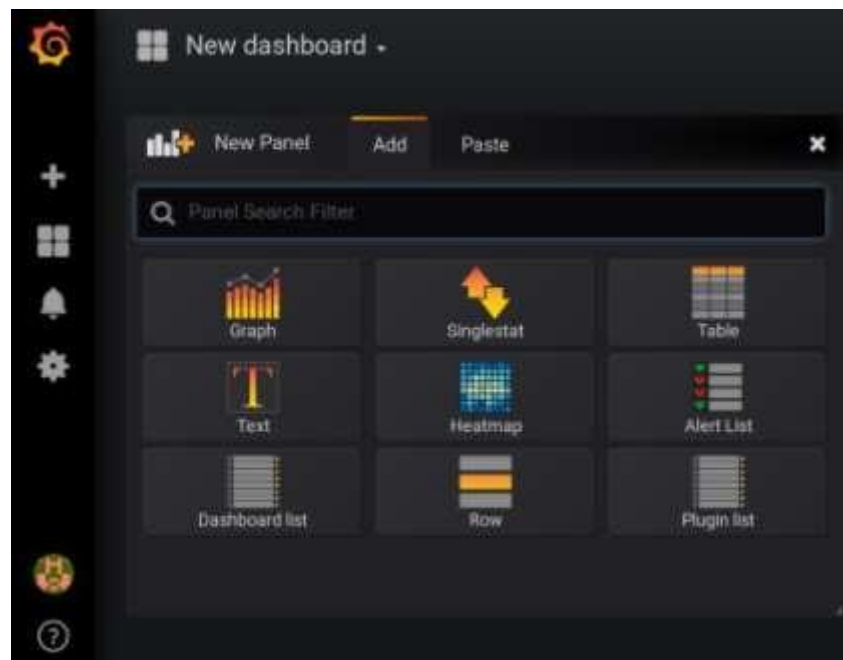


Рисунок 4.18 – Створення та заповнення нового дашборда

Щоб зробити графік потрібно використати функцію «Graph» (див. рисунок 4.19).

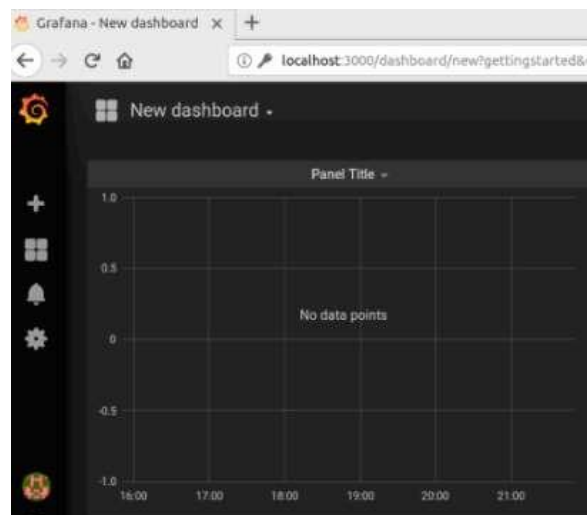


Рисунок 4.19 – Віджет «Graph»

Обираємо з меню Edit (див. рисунок 4.20).

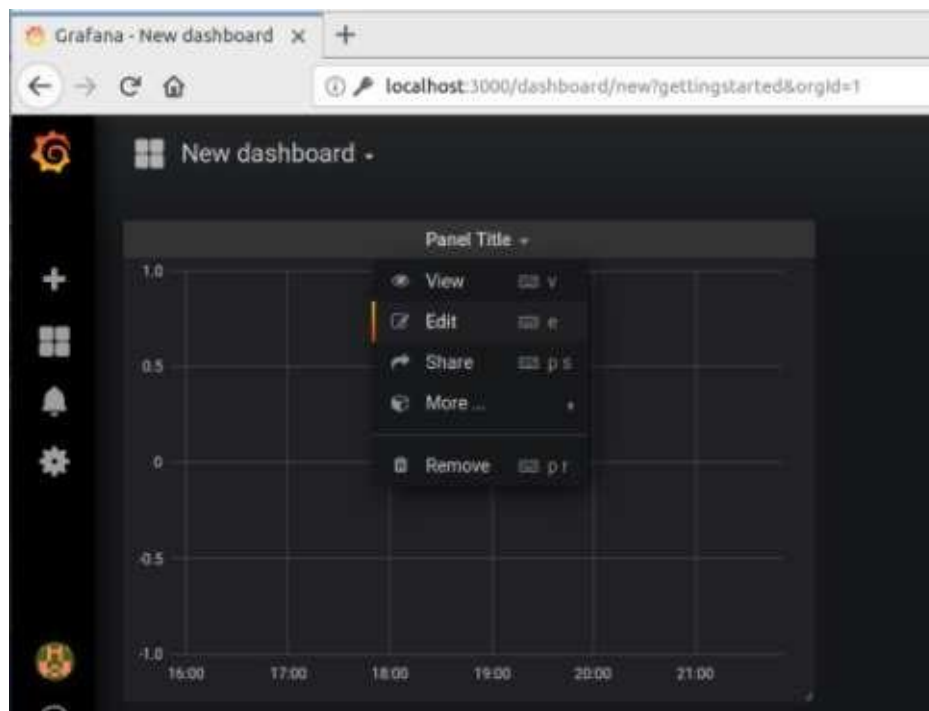


Рисунок 4.20 – Редагування графіку

Ми заповнюємо всі необхідні дані, щоб діаграма могла використовувати дані, накопичені в нашій базі даних (див. рис. 4.21).

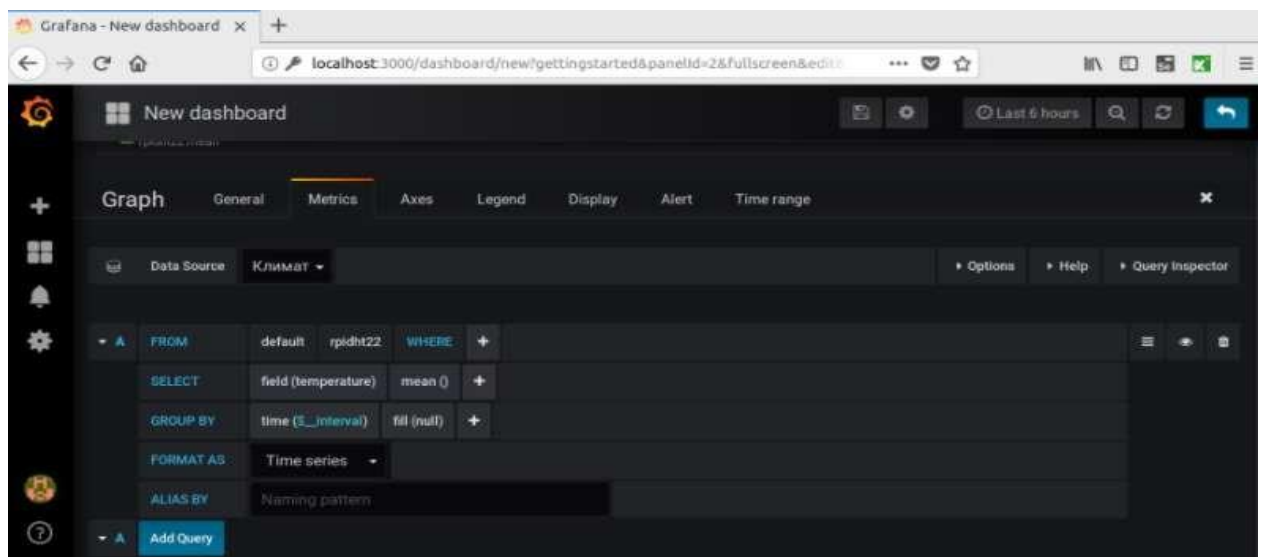


Рисунок 4.21 – Редагування графіку

Ми вибираємо період часу, протягом якого ми хочемо оцінити показники моніторингу. Як видно з рис. 4.22, ми вибираємо тривалість часу 11 днів. Також будуть проведені інші приклади: три дні та один місяць, щоб спостерігати, як індикатор змінюється протягом різного часу.

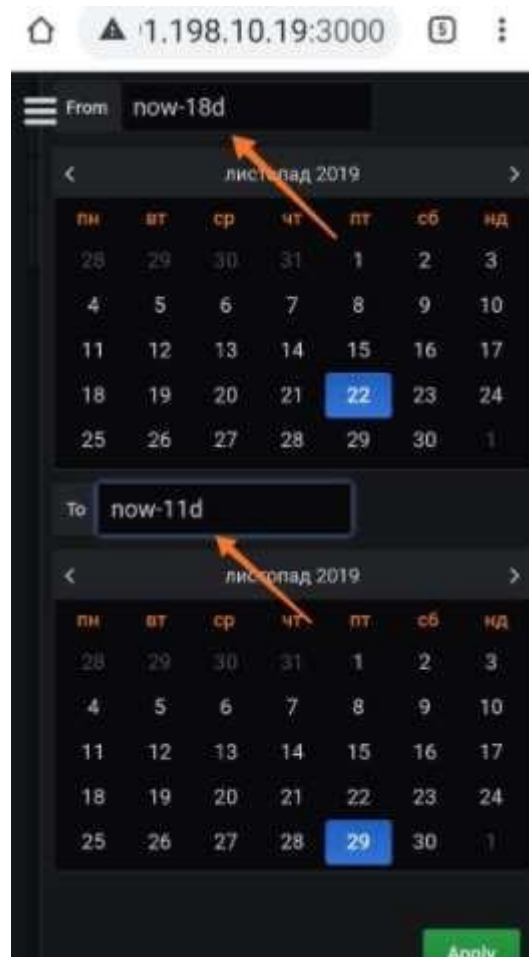


Рисунок 4.22 – Вибраний період часу для показників

Після виконання всіх необхідних налаштувань отримуємо наступний результат:

Індикатор температури. На рис. 4.23 показані показники температури за останні 11 днів. На рисунку 4.24 показані показники температури за останні три доби. На рисунку 4.25 показані дані температури за один день, а на рис. 4.26 – дані температури за останній місяць.

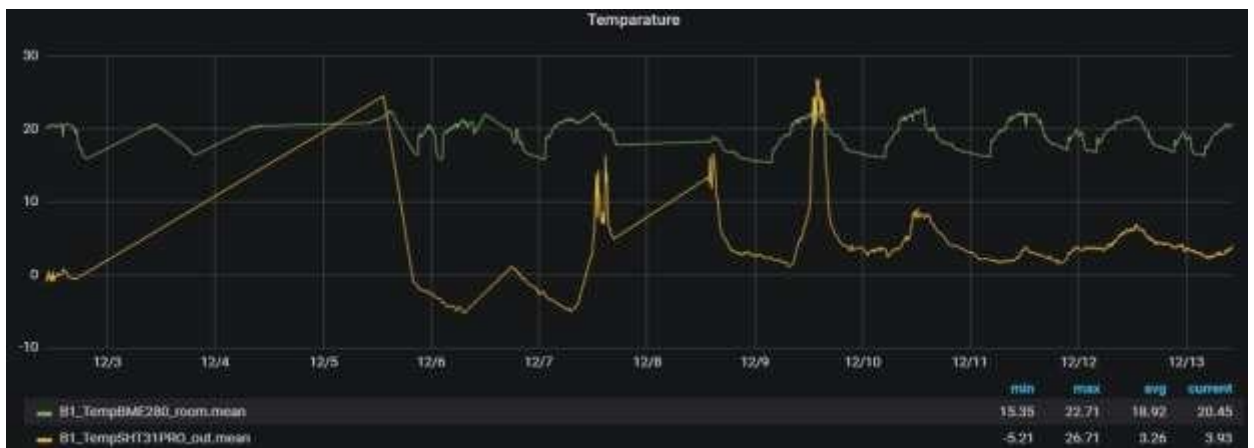


Рисунок 4.23 – Показник температури за певну кількість часу

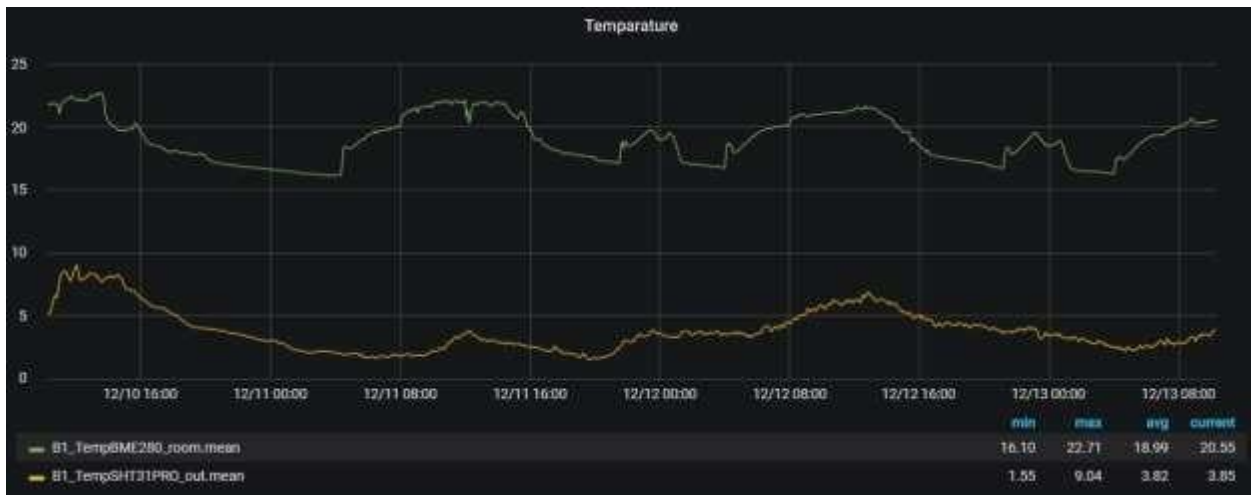


Рисунок 4.24 – Показник температури за останні три дні

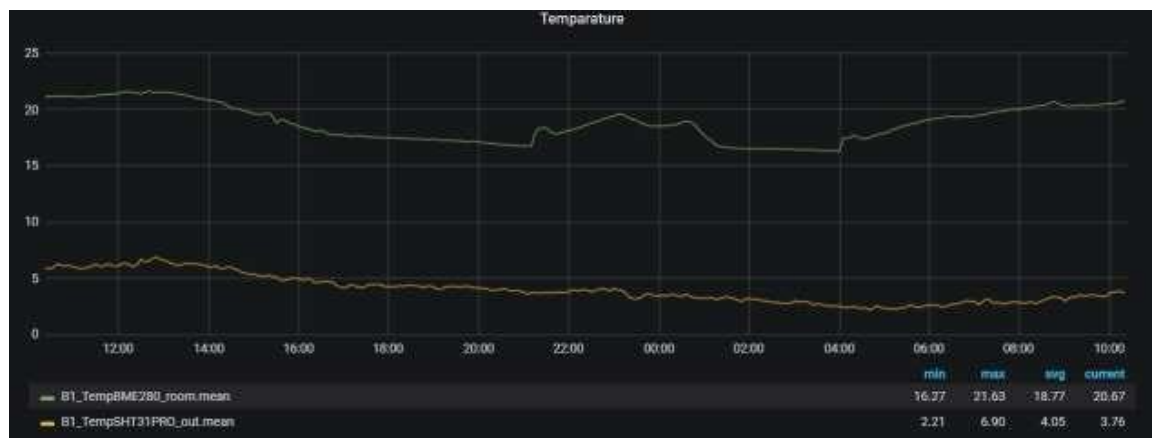


Рисунок 4.25 – Показник температури за один день

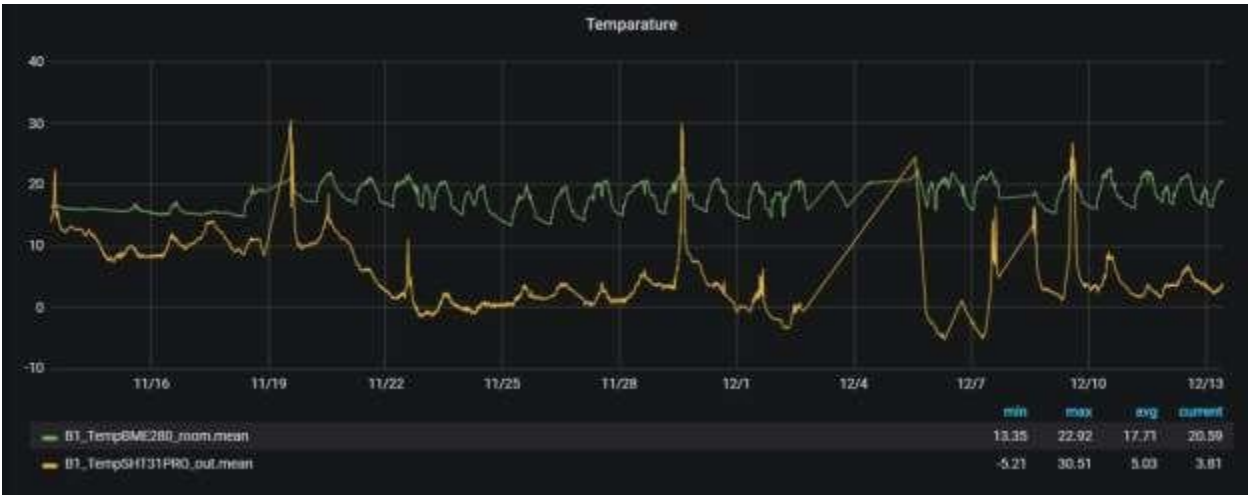


Рисунок 4.26 – Показник температури за останній місяць

Наступний показник, який ми можемо побачити графічно, це показник вологості. На рис.4.27 показані значення вологості за останні 11 днів. На рис. 4.28 наведено значення вологості за останні три дні. На рис. 4.29 показані значення вологості за одну добу, а на рис. 4.30 – за останній місяць.



Рисунок 4.27 – Показник вологості за останній одинадцять днів

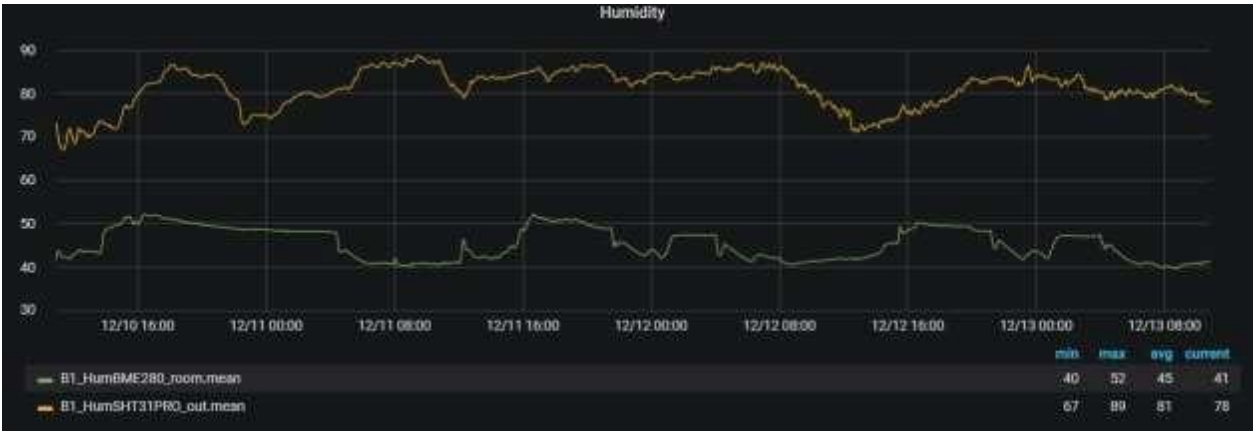


Рисунок 4.28 – Показник вологості за останні три дня

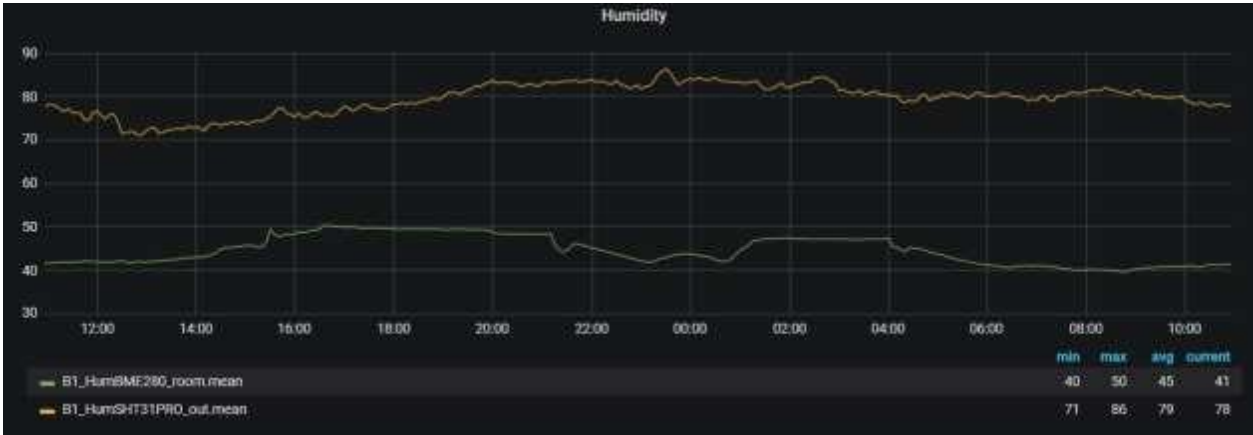


Рисунок 4.29 – Показник вологості за останній день



Рисунок 4.30 – Показник вологості за останній місяць

Індикатор тиску. Рисунок 4.31 ілюструє показники тиску за останні 11 днів. На рис. 4.32 показані показання манометра за останні три доби. На рисунку 4.33 показано показники тиску за один день, а на рисунку 4.34 – за по-

переній місяць.



Рисунок 4.31 – Показник тиску за останні одинадцять днів

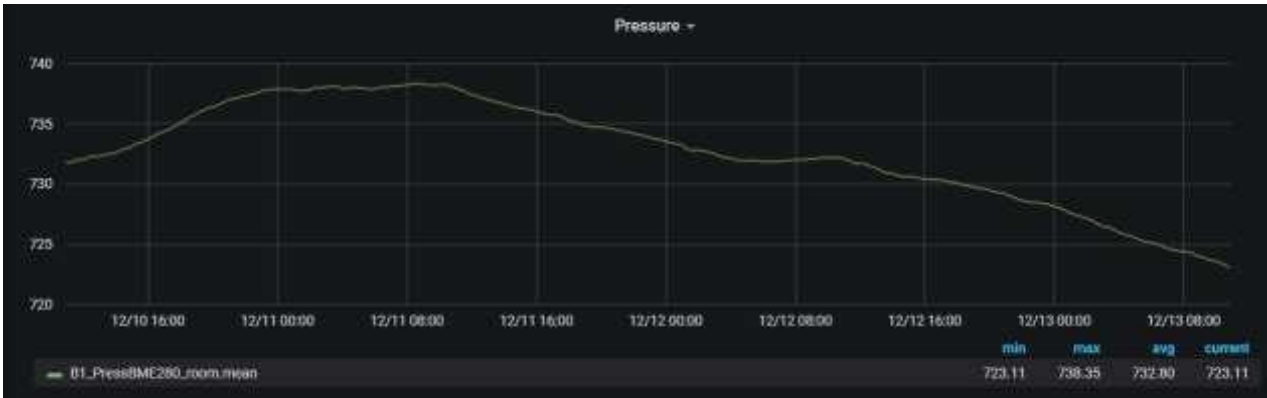


Рисунок 4.32 – Показник тиску за останні три дня

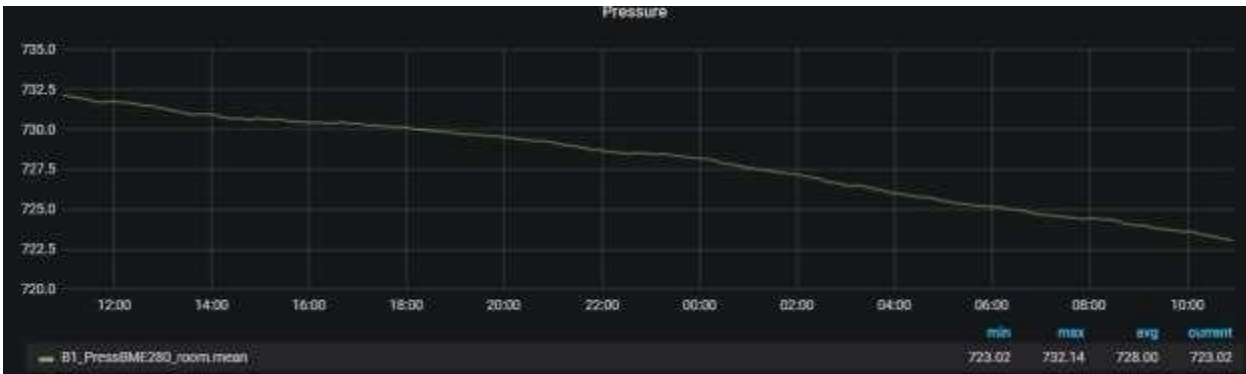


Рисунок 4.33 – Показник тиску за останній день

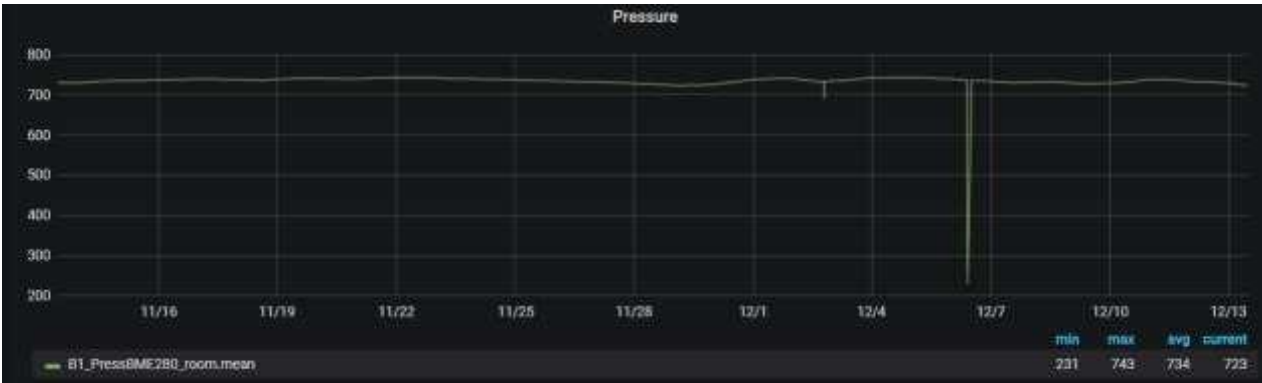


Рисунок 4.34 – Показник тиску за останній місяць

Індекс CO₂. На рисунку 4.35 показано показники CO₂ за останні 11 днів. На рис. 3.36 показані значення CO₂ за останні три дні. На рисунку 4.37 показано значення CO₂ за один день, а на рис. 4.38 показано значення CO₂ за останній місяць.

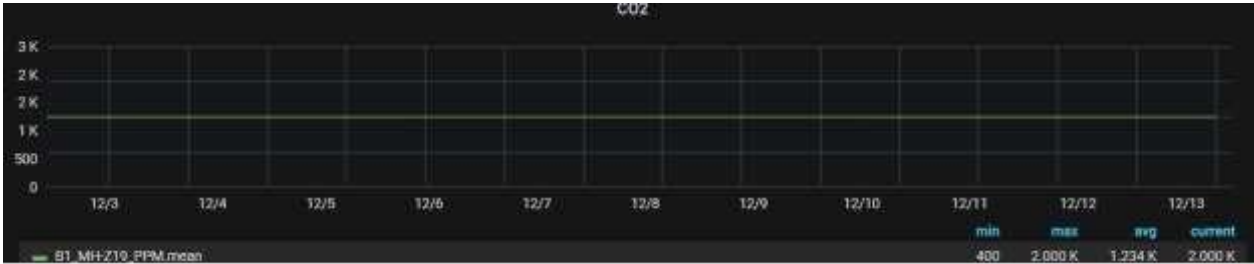


Рисунок 4.35 – Показник CO₂ за останні одинадцять днів

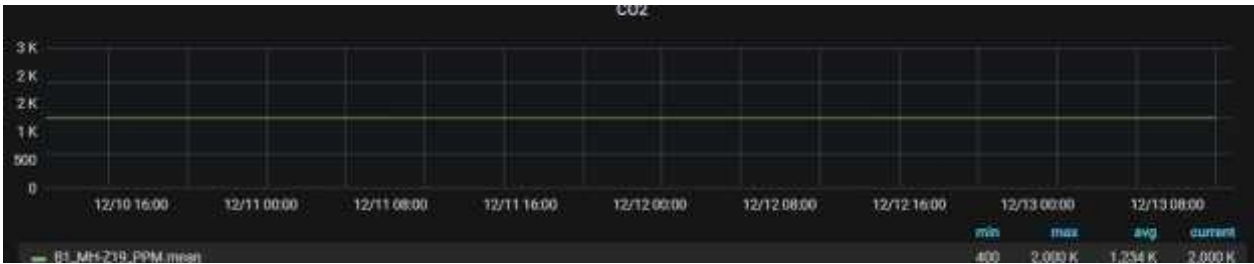


Рисунок 4.36 – Показник CO₂ за останні три дні

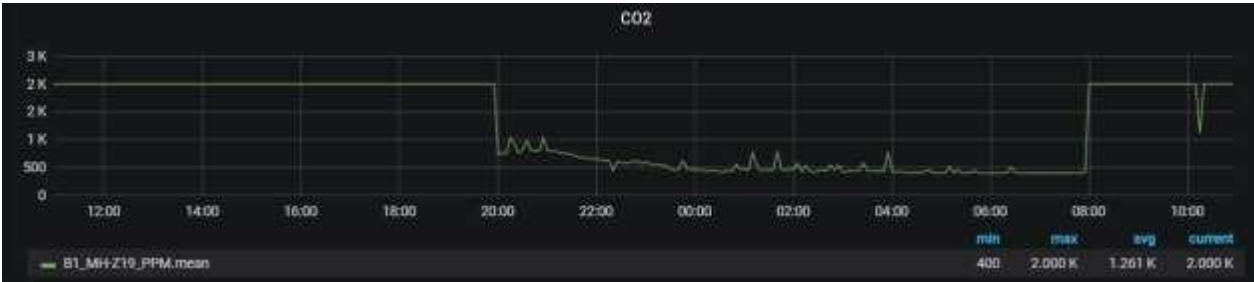


Рисунок 4.37 – Показник CO₂ за останній день



Рисунок 4.38 – Показник CO₂ за останній місяць

На рисунку 4.39 показані показники пилуки за останній день.

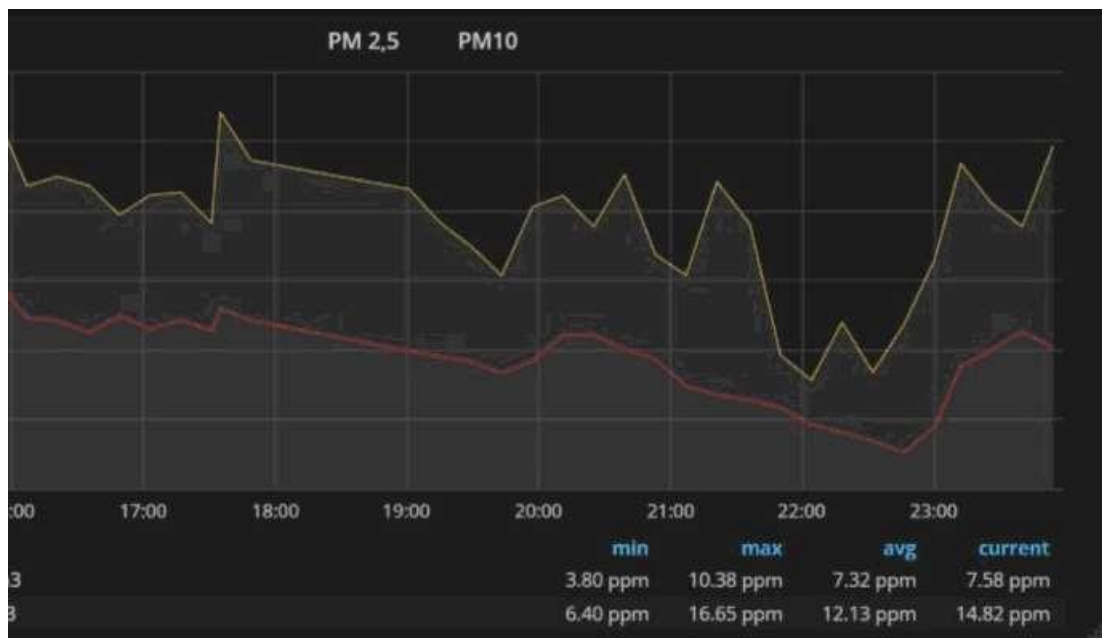


Рисунок 4.39 – Показники пилуки за останній день

4.2 Опис протоколу MQTT для обміну між давачами

MQTT або Message Queuing Telemetry Transport — це легкий, компактний і відкритий протокол обміну даними, створений для передачі даних у віддалені місця, де потрібен малий розмір коду та існують обмеження пропускну здатності каналу (див. рис. 4.40). Перераховані вище переваги дозволяють використовувати його в системах M2M та IIoT.

Існує також версія цього протоколу, MQTT-SN (MQTT для сенсорних мереж), раніше відома як MQTT-S, для вбудованих бездротових пристроїв, які не підтримують мережі TCP/IP, наприклад Zigbee. Основні особливості протоколу MQTT: Асинхронний протокол; Компактне повідомлення; Працювати в умовах, коли зв'язок по лінії передачі даних нестабільний; Підтримує кілька рівнів якості обслуговування (QoS) ;

Протокол MQTT працює на прикладному рівні поверх TCP/IP і за замовчуванням використовує порт 1883 (8883 під час підключення через SSL).

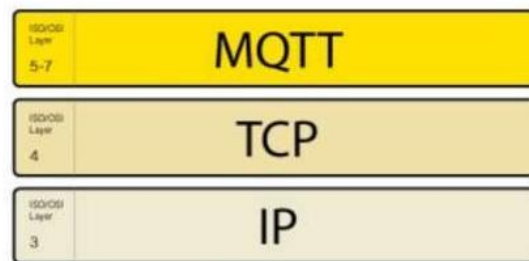


Рисунок 4.40 – Рівні роботи MQTT, TCP, IP

Шаблон публікації/підписки (також називається pub/sub) є різновидом традиційного дизайну клієнт-сервер. У моделі клієнт-сервер клієнт спілкується безпосередньо з одержувачем.

Семантика теми. Теми – це символи в кодуванні UTF-8. Ієрархічна структура тем має «деревоподібний» формат, що спрощує організацію тем і доступ до даних. Теми складаються з одного або кількох рівнів, розділених символом "/". Приклад теми, де датчик температури, розташований у спальні, публікує дані агенту: /Головна/Житлові приміщення/Вітальня1/Температура

Користувачі можуть отримувати дані з кількох тем одночасно, для цього існують символи підстановки. Вони бувають двох видів: одношарові і багатошарові. Щоб зрозуміти це простіше, розглянемо кожну з них на прикладі: Символ підстановки першого рівня. Для його використання використо-

вуйте символ «+». Наприклад, нам потрібно отримати дані про температуру всіх класів салону літака:

/ home / living-space / + / temperature

В результаті отримуємо дані з топиків:

/ Home / living-space / living-room1 / temperature

/ Home / living-space / living-room2 / temperature

/ Home / living-space / living-room3 / temperature

Багаторівневий wildcard. Для його використання застосовується символ

«#» Наприклад, щоб отримати дані з різних датчиків всіх класів салону літака:

/ home / living-space / #

В результаті отримуємо дані з топиків:

/ Home / living-space / living-room1 / temperature

/ Home / living-space / living-room1 / light1

/ Home / living-space / living-room1 / light2

/ Home / living-space / living-room1 / humidity

/ Home / living-space / living-room2 / temperature

/ Home / living-space / living-room2 / light1

Структура повідомлень MQTT складається з декількох частин (див. рисунок 4.41):

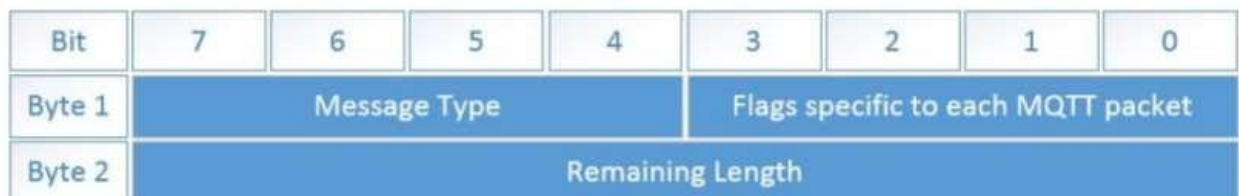


Рисунок 4.41 – Структура повідомлення

Тип повідомлення – це тип повідомлення, наприклад підключення, підписки, публікація тощо.

Довжина, що залишилася – вказує на довжину повідомлення, яка може становити від 1 до 4 байтів.

Чотири старші біти першого байта фіксованого заголовка прапора зарезервовані для спеціальних прапорів (див. рис. 4.41).



Рисунок 4.41 – Прапори повідомлень

DUP – встановлює позначку повторення, коли клієнт повторно надсилає пакет (для типів публікації, підписки, скасування підписки, Pubrel). Коли цей прапор встановлено, заголовок змінної має містити ідентифікатор повідомлення. QoS - якість обслуговування (0,1,2). Під час публікації даних із встановленим прапором збереження клієнт збереже їх. Наступного разу, коли ви підпишетесь на тему, клієнт негайно надішле повідомлення з цим прапорцем. Використовується лише для повідомлень типу "Опублікувати". Рівень якості обслуговування (QoS) — це угода між відправником і одержувачем повідомлення, яка визначає гарантії доставки для конкретного повідомлення. MQTT має 3 рівні QoS: Не більше одного разу (0); Принаймні один раз (1) ; Рівно 1раз (2) ;

Деякі заголовки містять змінні заголовки. Він містить такі дані:

- Ідентифікатор пакета – ідентифікатор пакета присутній у всіх типах повідомлень, крім: Connect, Connack, Publish (QoS<1), Pingreq, Pingresp, Disolect. ;

- Прапорці підключення - прапорці, що вказують на поведінку клієнта під час підключення;
- Ім'я користувача – якщо цей прапорець присутній, ім'я користувача має бути вказано в «load» (для автентифікації клієнта) Пароль – якщо цей прапорець присутній, пароль має бути вказано в «load» (для автентифікації клієнта). Збереження волі - під час інсталяції клієнт зберігає заповітні повідомлення. Will QoS – якість обслуговування Will Message. Якщо встановлено прапорець Will, Will QoS і Will keep є обов'язковими;
- Will Flag – якщо встановлено цей прапорець, після того, як клієнт від'єднається від Брокера без надсилання команди Disconnect (у разі неочікуваного відключення тощо), клієнт сповістить усіх підключених до нього через так званий клієнт Will Message. Очистити сеанс - очистити сеанс. Присутній сеанс - використовується в повідомленнях типу Connack.

Робота клієнта Mosquito MQTT. У Linux є кілька клієнтів MQTT для одноплатних комп'ютерів. Одним з найпопулярніших є москитний. По-перше, перед початком встановлення нам потрібно оновити систему. Щоб підключитися до Mosquito, введіть: `sudo apt-get update sudo apt-get upgrade`.

Проте звичайна бібліотека Raspberry Pi (Orange Pi) 'apt-get' не містить останньої версії програмного забезпечення Mosquitto. Тому нам потрібно оновити бібліотеки перед встановленням. Якщо на сервері не встановлена остання версія клієнта, під час роботи можуть виникати помилки. Додаємо ключ і оновлюємо репозиторій.

Перевірка статусу MQTT клієнта завжди можна через термінал:

```
sudo /etc/init.d/mosquitto status
```

Після введення даної команди в терміналі буде показана інформація про стан сервера(див. рисунок 4.42).

```

pi@orangezero:~$ sudo /etc/init.d/mosquitto status
[sudo] password for pi:
● mosquitto.service - LSB: mosquitto MQTT v3.1 message broker
   Loaded: loaded (/etc/init.d/mosquitto; bad; vendor preset: enabled)
   Active: active (running) since Tue 2017-10-03 14:28:45 MSK; 1min 12s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 479 ExecStart=/etc/init.d/mosquitto start (code=exited, status=0/SUCCESS)
    CGroup: /system.slice/mosquitto.service
            └─517 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Oct 03 14:28:43 orangezero systemd[1]: Starting LSB: mosquitto MQTT v3.1 message broker...
Oct 03 14:28:45 orangezero mosquitto[479]: * Starting network daemon: mosquitto
Oct 03 14:28:45 orangezero mosquitto[479]:   ...done.
Oct 03 14:28:45 orangezero systemd[1]: Started LSB: mosquitto MQTT v3.1 message broker.

```

Рисунок 4.42 – Статус сервера MQTT

4.3 Висновок до розділу 4

У четвертому розділі було розпочато створення програмного забезпечення для обробки результатів спостережних досліджень, а також розглянуто два питання: комбінування та налаштування датчиків для моніторингу повітря. Під час виконання цієї частини бакалаврської дипломної роботи такі сервіси, як EAP Easy Mega, InfluxDB і Node-RED, використовуються для підключення та налаштування датчиків моніторингу повітря, сервіс Grafana використовується для візуалізації результатів цих датчиків.

В даному розділі розглядаються питання «Протокол MQTT для обміну даними між провайдерами IoT» і «Робота клієнта Mosquito MQTT».

Проаналізовано та проведено оцінку продуктивності та надійності клієнта Mosquitto MQTT у випадку практичного використання для дослідження моніторингу якості повітря.

ВИСНОВКИ

У бакалаврській дипломній роботі розроблено засіб контролю параметрів мікроклімату у салоні літаків громадської авіації.

Очікується, що розроблена система дозволить автоматично контролювати та регулювати параметри мікроклімату у салоні літака, підвищуючи комфорт та безпеку пасажирів. Вона також сприятиме оптимізації енергоспоживання та покращенню експлуатаційних характеристик літаків громадської авіації.

Система включає датчики температури, вологості, та якості повітря, які постійно моніторять стан мікроклімату у салоні. Зібрані дані аналізуються в режимі реального часу, і система автоматично вносить необхідні корективи для підтримання оптимальних умов. Крім того, система може інтегруватися з іншими бортовими системами, такими як вентиляція та кондиціонування, для забезпечення комплексного підходу до керування мікрокліматом.

Розробка та впровадження ефективного засобу контролю параметрів мікроклімату у салонах літаків громадської авіації є важливим кроком у забезпеченні високого рівня комфорту та безпеки пасажирів. Запропоновані рішення можуть бути використані для вдосконалення існуючих систем клімат-контролю на борту літаків. Це не лише покращує умови перебування пасажирів на борту, але й сприяє зменшенню експлуатаційних витрат за рахунок підвищення енергоефективності. Застосування новітніх технологій в даній сфері допоможе авіакомпаніям відповідати сучасним стандартам та вимогам, підвищуючи конкурентоспроможність на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Build your own ESP8266 Web Server Tutorial [Електронний ресурс]
– Режим доступу: <http://www.14core.com/build-your-own-esp8266-web-server-tutorial/> – Дата доступу: 17.06.2024.
2. ESP8266 webserver serving multiple pages [Електронний ресурс] – Режим доступу: <http://www.instructables.com/id/ESP8266-webserver-serving-multiple-pages/?ALLSTEPS> – Дата доступу: 17.06.2024.
3. ESP8266 Thing Hookup Guide [Електронний ресурс] – Режим доступу: <https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/all> – Дата доступу: 18.06.2019.
4. Triple Server Update – Part 3: More Server Features [Електронний ресурс] – Режим доступу: <http://internetofhomethings.com/homethings/?cat=10> – Дата доступу: 18.06.2024.
5. Internet of Things. M2M from research to standards: The next steps. – IEEE Communications. [Електронний ресурс] – Режим доступу : <http://www.comsoc.org/commag/cfp/internetthingsm2m-research-standards-next-steps>. – Дата доступу: 09.12.2024
6. Моніторинг навколишнього природного середовища // Юридична енциклопедія : [в 6-ти т.] / ред. кол. Ю. С. Шемшученко (відп. ред.) [та ін.] – К. : Українська енциклопедія ім. М. П. Бажана, 2001. – Т. 3 : К – М. – 792 с. – ISBN 966-7492-03-6
7. Internet of Things: Privacy & Security in a Connected. World Federal Trade Commission (FTC). Staff Report January, 2015 [Електронний ресурс] – Режим доступу : <https://www.ftc.gov/system/files/documents/reports/federal-trade-commission->

staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf. – Дата доступу: 14.12.2024.

8. Blum P., Goff B. 'Internet Of Things' 101: Legal Concerns. Law360, New York (April 14, 2014) [Електронний ресурс] – Режим доступу: <http://www.law360.com/articles/526266/internet-of-things-101-legal-concerns> – Дата доступу: 18.12.2024.

9. Якість атмосферного повітря: короткий опис Директив ЄС та графіку їх впровадження. Київ: Проект технічної допомоги ЄС «Додаткова підтримка Міністерства екології та природних ресурсів України у впровадженні секторальної бюджетної підтримки», 2015. 15 с.

10. Полетаєва Л. М., Сафранов Т. А. Моніторинг навколишнього природного середовища. Навчальний посібник. К.: КНТ, 2007. 172 с. 5. Чугай А. В. Моніторинг довкілля. Методи вимірювань параметрів навколишнього середовища. Конспект лекцій. Одеса: ТЕС, 2014. 66 с.

11. Моніторинг якості атмосферного повітря: український та міжнародний досвід. [Аналітична записка] / Кольцов М., Шевченко Л. Київ: ГО «Фундація «Відкрите Суспільство», 2018. 13 с.

12. Проблеми забруднення атмосферного повітря при експлуатації аеропортів цивільної авіації: брошура / [О. Запорожець, К. Синило, К. Ульянова, А. Крупко, В. Паращанов]; за ред. К. Синило. Київ: НЕЦУ, 2018. 20 с.

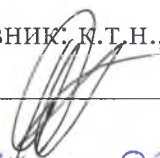
Додаток А
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА
РОЗРОБКА ЗАСОБУ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІ-
МАТУ У САЛОНІ ЛІТАКІВ ГРОМАДСЬКОЇ АВІАЦІЇ

Виконав: студент 4-го курсу, групи КІВТ-20
спеціальності 152 - Метрологія та інформаційно-
вимірювальна техніка

(шифр і назва напрямку підготовки, спеціальності)

 Тарасюк Д.С.
(прізвище та ініціали)

Керівник: к.т.н., доц., доц. каф. ІРТС
 Дудатьєв І.А.

(прізвище та ініціали)

« 14 »  2024 р.



Рисунок 1 – Аналізатор якості повітря NDIR



Рисунок 2 – Аналізатор якості повітря Z-WAVE



Рисунок 3 – Аналізатор якості повітря Boseon



Рисунок 4 – Аналізатор якості повітря Ajax



Рисунок 5 – Газоаналізатор

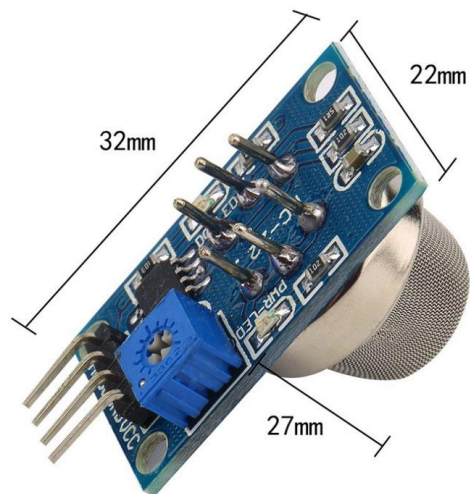


Рисунок 6 – Модуль датчика якості повітря MQ-135

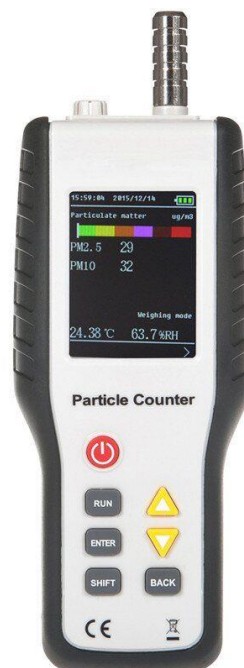


Рисунок 7 – Тестер запыленості Walcom

Додаток Б
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ РОБОТИ

**РОЗРОБКА ЗАСОБУ КОНТРОЛЮ ПАРАМЕТРІВ МІКРОКЛІМАТУ У
САЛОНІ ЛІТАКІВ ГРОМАДСЬКОЇ АВІАЦІЇ**

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Розробка засобу контролю параметрів мікроклімату у салоні літаків громадської авіації»

Тип роботи: БДР
(БДР, МКР)

Підрозділ ІРТС, ІЕС
(кафедра, факультет)

Показники звіту подібності Unichesk

Оригінальність 91.4% Схожість 8.06%

Аналіз звіту подібності (відмітити потрібне):

- ☒ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку 
(підпис)

Олександр ЗВЯГІН
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи


(підпис)

Дмитро ТАРТАСЮК
(прізвище, ініціали)

Керівник роботи


(підпис)

Ігор ДУДАТЬЄВ
(прізвище, ініціали)