Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему:

Програмне забезпечення системи моніторингу та управління виробничими процесами на підприємстві на базі платформи Arduino

КбПЗ_2026

Виконав здобувач вищої освіти
IV курсу, групи КІ-22-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____Ткалич В.О.
«____» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____Кислун О.А.
«____» _____ 2026 р.
Рецензент _____

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *бакалавр*
Галузь знань ._12 _____*"*Інформаційні технології*"*_____
Спеціальність *123 "*Комп'ютерна інженерія*"*
Освітньо-професійна (освітньо-наукова) програма _*"*Комп'ютерна інженерія*"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
"____ " _____ 2026 р.

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Ткаличу Максиму Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи моніторингу та управління виробничими процесами на підприємстві на базі платформи Arduino*
2. Керівник роботи *Кислун Олег Андрійович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від "27" лютого 2026 року №136-02
3. Строк подання роботи до захисту *10.06.2026 р.*
4. Мета та завдання випускної кваліфікаційної роботи. *Метою роботи є розробка програмного забезпечення системи системи моніторингу та управління виробничими процесами*
5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію
6. Висновки.
6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи керування	*1 аркуш*
Функціональна схема системи керування	*1 аркуш*
Діаграма взаємодії процесів	*1 аркуш*
Блок-схема алгоритму роботи	*1 аркуш*
Блок-схема алгоритму реєстрації в системі	*1 аркуш*

7. Дата видачі завдання «27» лютого 2026 р.

КОПІЯ 2026

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем керування	10.03.2026	
2.	Постановка задачі, оформлення ТЗ	15.03.2026	
3.	Розробка моделі компонента	20.03.2026	
4.	Розробка структур даних	25.03.2026	
5.	Розробка алгоритмів зв’язку та відображення	30.03.2026	
6.	Програмування алгоритмів	10.04.2026	
7.	Оформлення ПЗ	17.05.2026	
8.	Попередній захист роботи	23.05.2026	

КбПЗ_2026

Дата видачі завдання
«27» лютого 2026р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
«__»_____2026р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Ткалич М.Ю. Програмне забезпечення системи моніторингу та управління виробничими процесами на підприємстві на базі платформи Arduino. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

У кваліфікаційній бакалаврській роботі розглянуто розроблення програмного забезпечення системи моніторингу та управління виробничими процесами з використанням мікроконтролерної платформи Arduino. Актуальність теми обумовлена необхідністю автоматизації виробничих процесів, підвищення ефективності роботи підприємств, зменшення впливу людського фактору та забезпечення оперативного контролю за технологічними параметрами.

У роботі проведено аналіз сучасних підходів до побудови автоматизованих систем моніторингу та управління, досліджено можливості використання платформи Arduino для реалізації систем промислової автоматизації. Розроблено програмне забезпечення, яке забезпечує збір даних із датчиків, їх обробку, відображення та передачу до центрального вузла керування. Реалізовано функції моніторингу параметрів виробничого процесу, автоматичного реагування на зміну контрольованих показників, а також можливість дистанційного керування виконавчими пристроями.

Розроблена система характеризується гнучкістю, доступністю та можливістю подальшого розширення функціональних можливостей.

Отримані результати можуть бути використані для автоматизації виробничих процесів на підприємствах різного профілю з метою підвищення продуктивності, точності контролю та ефективності управління.

Ключові слова: Arduino, автоматизація, виробничі процеси, моніторинг, мікроконтролер, програмне забезпечення, датчики, система керування.

КОПЗ-2026

ABSTRACT

Tkalych M. Software for monitoring and controlling production processes at an enterprise based on the Arduino platform. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

The qualification bachelor's thesis considers the development of software for a monitoring and control system for production processes using the Arduino microcontroller platform. The relevance of the topic is due to the need to automate production processes, increase the efficiency of enterprises, reduce the impact of the human factor and ensure operational control over technological parameters.

The paper analyzes modern approaches to building automated monitoring and control systems, explores the possibilities of using the Arduino platform for implementing industrial automation systems. Software has been developed that provides data collection from sensors, their processing, display and transmission to the central control node.

The functions of monitoring production process parameters, automatic response to changes in controlled indicators, as well as the possibility of remote control of actuators have been implemented.

The developed system is characterized by flexibility, accessibility and the possibility of further expansion of functional capabilities.

The results obtained can be used to automate production processes at enterprises of various profiles in order to increase productivity, control accuracy and management efficiency.

Keywords: Arduino, automation, production processes, monitoring, microcontroller, software, sensors, control system.

КОПЗ_2026

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 2
ВСТУП..... 3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 3
1.1 Призначення системи..... 5
1.2 Область застосування..... 12
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 13
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти) 13
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... 35
2.3 Розгорнута постановка завдання 37
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 38
3.1 Опис функціонування системи 38
3.2 Розробка структурної схеми..... 39
3.3 Розробка функціональної схеми 40
3.4 Розробка діаграми процесів..... 45
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... 46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... 46
4.2 Захист розробленого програмного забезпечення..... 67
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 71
6 ОСНОВНІ ВИСНОВКИ..... 77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 79

КПЗ – 2026

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІТ – інформаційні технології;

ІЧ – інфрачервоне;

МК – мікроконтролер;

МП – мікропроцесор;

ПЗ – програмне забезпечення;

ПК – персональний комп’ютер;

API – application programming interface – прикладний програмний інтерфейс;

AWS – Amazon Web Services – платформа хмарних обчислень;

IoT – internet of things – інтернет речей;

SQL – Structured Query Language – мова структурованих запитів;

MQTT – Message Queuing and Telemetry Transport – черга повідомлень;

WSS – Websockets Secure – захищений протокол вебсокетів;

SDK – Software development kit – набір для розробки ПЗ;

IDE – Integrated Development Environment – система для розробки ПЗ;

HTTP – Hypertext Transfer Protocol – протокол передачі даних;

WPAN – wireless personal area network – бездротові персональні мережі;

WLAN – wireless local area network – бездротова локальна мережа.

КБПЗ – 2026

ВСТУП

Останніми роками спостерігається стрімкий розвиток систем моніторингу в реальному часі на основі концепції Інтернету речей (IoT). Попит на IoT-рішення зростає експоненціально завдяки їх широкому застосуванню в різних галузях, значна частина яких і досі перебуває на етапі активного дослідження.

Використання даних про критичні показники для оптимізації виробничих і фінансових процесів давно є важливою складовою діяльності підприємств, адже це дозволяє підвищувати ефективність та зменшувати витрати.

Важливим аспектом сучасних систем є їх інтеграція з хмарними сервісами або можливість міграції до них. Зокрема, хмарна платформа Amazon Web Services (AWS) пропонує широкий спектр IoT-рішень, які дозволяють підключати та керувати великою кількістю пристроїв, збирати, зберігати й аналізувати дані для промислових, комерційних, споживчих та автомобільних застосувань. Використання хмарних технологій також сприяє підвищенню рівня безпеки даних і стабільності роботи систем.

Автоматизована система моніторингу та управління виробничими процесами на підприємстві виконує такі основні функції:

- збір критичних показників, необхідних для забезпечення стабільного функціонування виробничих процесів;
- візуалізацію даних та управління пороговими значеннями показників;
- виявлення, фіксацію та оперативне реагування на критичні ситуації.

Застосування такої системи дозволяє підвищити ефективність виробництва, зменшити ризики збитковості у довгостроковій перспективі та запобігти негативним наслідкам, пов’язаним із повним або частковим виходом з ладу виробничих елементів. Промисловий IoT забезпечує масштабованість ресурсів, дистанційне керування процесами, автоматизацію виробництва та оперативний обмін інформацією.

КБПЗ_2026

Отже, розробка автоматизованої системи моніторингу та управління виробничими процесами є актуальним завданням.

Для досягнення поставленої мети необхідно:

- проаналізувати існуючі системи моніторингу та датчикові рішення для збору критичних показників;
- дослідити типи датчиків, що застосовуються на підприємствах, і визначити актуальні метрики;
- розглянути мікроконтролерні платформи як основу апаратного забезпечення;
- обґрунтувати вибір апаратних і програмних засобів для реалізації системи;
- побудувати інфраструктуру обробки та розгортання даних із використанням хмарного сервісу Amazon Web Services, мови програмування Python і фреймворку Flask;
- розробити та реалізувати автоматизовану систему моніторингу на основі досліджених технологій.

КбПЗ_2026

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Моніторинг інформаційно-технологічних систем є важливою складовою управління інформаційною інфраструктурою підприємства. Він передбачає безперервне спостереження та періодичний аналіз компонентів робочих процесів із відстеженням динаміки змін, що відбуваються в системі. Основним завданням систем моніторингу є збір, зберігання та аналіз інформації про стан підконтрольних елементів структури компанії.

Пошук ефективних методів виявлення проблем у процесі виробничої діяльності є ключовим чинником підвищення продуктивності бізнесу. Для досягнення необхідної результативності важливо своєчасно виявляти та усувати несправності, запобігаючи критичним інцидентам і матеріальним втратам. Особливо актуально це у технологічній сфері, де нераціональне використання виробничих ресурсів або обладнання може призвести до зниження ефективності та довгострокової збитковості. Навіть базова система моніторингу здатна суттєво мінімізувати подібні ризики.

Промисловий IoT вирізняється масштабністю та функціональними можливостями. Він дозволяє здійснювати збір і передавання даних, дистанційне керування процесами та автоматизацію виробництва. Використання автоматизованих систем дає змогу зменшити потребу в додатковому персоналі, оскільки модулі обробки інформації здатні виконувати частину контрольних функцій. Отримані дані можуть застосовуватися для запобігання позаплановим простоям, поломкам обладнання, надлишковому технічному обслуговуванню та порушенням у ланцюгах постачання, що сприяє підвищенню загальної ефективності підприємства.

Передавання інформації до хмарних сервісів, наприклад Microsoft Azure, забезпечує централізоване зберігання даних, масштабованість інфраструктури та

КПІ 2026

високий рівень захисту інформації.

Актуальність впровадження систем контролю та моніторингу виробничих показників пов’язана не лише з питаннями безпеки, а й із забезпеченням стабільності бізнес-процесів. Розроблений вебзастосунок дозволяє оперативно реагувати на збої в роботі сервісів і запобігати виникненню несправностей. Саме це обґрунтовує доцільність створення автоматизованої системи моніторингу та управління виробничими процесами. Під виробничими процесами мається на увазі створення матеріального продукту, необхідного для функціонування та розвитку підприємства, а моніторинг є невід’ємним інструментом підвищення ефективності його виробництва.

1.1 Область застосування

В даній випускні кваліфікаційній бакалаврській роботі буде розроблятись програмне забезпечення системи моніторингу та управління виробничими процесами на підприємстві.

У сучасних умовах розвитку промисловості особливої актуальності набуває автоматизація виробничих процесів, що дозволяє підвищити ефективність роботи підприємств, зменшити вплив людського фактора та забезпечити безперервний контроль технологічних параметрів. Одним із ключових напрямів цифрової трансформації виробництва є впровадження систем моніторингу та управління виробничими процесами на основі мікроконтролерних та інформаційно-комунікаційних технологій.

У межах даної випускної кваліфікаційної бакалаврської роботи розробляється програмне забезпечення системи моніторингу та управління виробничими процесами на підприємстві, яке призначене для збору, обробки, аналізу та відображення технологічних параметрів у режимі реального часу.

Система забезпечує централізований контроль стану обладнання, технологічних ліній та виробничого середовища.

КПІ – 2026

Областю застосування розроблюваної системи є промислові підприємства різних галузей економіки, де необхідний постійний контроль параметрів виробництва та оперативне реагування на зміну технологічних умов. До таких підприємств належать:

– машинобудівні виробництва;
– агропромислові комплекси;
– харчова промисловість;
– енергетичні підприємства;
– переробні та складські комплекси;
– підприємства
– автоматизовані виробничі дільниці малого та середнього бізнесу.

Система може застосовуватись для контролю роботи технологічного обладнання, моніторингу температурних режимів, вологості, тиску, енергоспоживання, швидкості роботи механізмів, стану електродвигунів та інших виробничих параметрів.

Однією з важливих сфер застосування є автоматизовані виробничі лінії, де безперервність процесу має критичне значення. Використання програмного забезпечення дозволяє отримувати актуальну інформацію про стан обладнання, своєчасно виявляти відхилення від нормальних режимів роботи та запобігати аварійним ситуаціям.

Розроблювана система може функціонувати як складова концепції Індустрії 4.0, що передбачає інтеграцію фізичних виробничих процесів із цифровими системами керування. Завдяки використанню мікроконтролерних платформ, зокрема ESP32, забезпечується можливість створення розподілених мереж датчиків та виконавчих пристроїв із бездротовою передачею даних.

Особливістю області застосування системи є можливість її використання на підприємствах із різним рівнем автоматизації. Вона може бути впроваджена як:

– самостійна система моніторингу;
– допоміжна система контролю;

КБПЗ_2026

– складова існуючої SCADA або MES-системи;

– модуль модернізації застарілого обладнання.

На підприємствах агропромислового сектору система може використовуватись для контролю умов зберігання продукції, управління вентиляційними системами, моніторингу роботи насосного обладнання, сушильних установок або тепличних комплексів. Це дозволяє оптимізувати використання ресурсів та зменшити експлуатаційні витрати.

У виробничих приміщеннях система забезпечує контроль мікроклімату, що має важливе значення для стабільності технологічних процесів та забезпечення безпечних умов праці персоналу. Автоматичний збір інформації дозволяє вести журнал подій, здійснювати аналіз роботи обладнання та прогнозувати можливі несправності.

Ще однією сферою застосування є енергомоніторинг підприємств. Програмне забезпечення дозволяє відстежувати споживання електроенергії окремими вузлами виробництва, аналізувати навантаження та оптимізувати режими роботи обладнання. Це сприяє підвищенню енергоефективності та зменшенню витрат підприємства.

Система також може використовуватись у віддалених або розподілених виробничих об’єктах, де відсутня постійна присутність обслуговуючого персоналу. Завдяки використанню мережевих технологій оператор має можливість здійснювати контроль через веб-інтерфейс або мобільні пристрої незалежно від місця розташування.

Важливою областю застосування є підвищення промислової безпеки. Система дозволяє автоматично формувати попередження у випадку перевищення допустимих параметрів роботи обладнання, що знижує ризик виникнення аварійних ситуацій, простоїв виробництва або пошкодження техніки.

Розроблюване програмне забезпечення може застосовуватись як на нових підприємствах під час проєктування автоматизованих систем керування, так і при модернізації існуючих виробничих процесів без значних фінансових витрат.

КОПЗ_2026

Використання доступних мікроконтролерних платформ дозволяє значно знизити вартість впровадження у порівнянні з традиційними промисловими рішеннями.

Крім промислового використання, система може бути застосована у навчальних та дослідницьких лабораторіях для моделювання виробничих процесів, дослідження алгоритмів автоматичного керування та навчання студентів принципам побудови кіберфізичних систем.

Таким чином, область застосування розроблюваного програмного забезпечення охоплює широкий спектр виробничих та технологічних задач, пов’язаних із моніторингом, аналізом та управлінням процесами підприємства. Використання системи дозволяє підвищити ефективність виробництва, забезпечити оперативний контроль обладнання, зменшити експлуатаційні витрати та підвищити рівень автоматизації підприємства.

КБПЗ_2026

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Аналіз сучасних приладів та систем моніторингу зовнішніх показників

Під час проєктування систем моніторингу, основу яких становлять інформаційно-вимірювальні комплекси, одним із ключових завдань є обґрунтований вибір діагностичних ознак. Раціональний добір таких ознак можливий на основі побудови та аналізу математичних моделей об’єктів моніторингу або фізичних процесів, що супроводжують їх функціонування.

Моніторинг, який базується на аналізі випадкових процесів і полів, зокрема шумової діагностики, передбачає використання адекватних математичних моделей, що встановлюють взаємозв’язок між реальним технічним станом об’єкта та характеристиками, отриманими в результаті обробки вимірювальної інформації. У такому підході математична модель розглядається як випадковий процес або випадкове поле певного класу з притаманними йому властивостями, які дають змогу виконувати оцінювання за допомогою статистичних або інших методів на основі аналізу однієї чи кількох реалізацій процесу.

Під час побудови ймовірнісних моделей фізичних процесів у вузлах теплоенергетичного обладнання доцільно використовувати класи лінійних випадкових процесів. Такі моделі є конструктивними, оскільки кожен їхній параметр має чіткий фізичний зміст. Крім того, властивості лінійних випадкових процесів дозволяють отримувати аналітичні вирази для моментів розподілу досліджуваного сигналу, формувати функції розподілу або характеристичні функції.

КБПЗ_2026

Аналіз зазначених моделей дає можливість теоретично обґрунтувати вибір діагностичних ознак, визначити допустимі межі їх зміни (так звані діагностичні простори) та сформувати ефективні вирішальні правила для виявлення й класифікації дефектів. Окрім цього, такі моделі створюють передумови для застосування методів імітаційного комп'ютерного моделювання, що дозволяє значно зменшити витрати часу й ресурсів на фізичне моделювання процесів або проведення експериментів на реальному обладнанні.

Датчики збору зовнішніх показників і модулі інтеграції

У промисловості та комерційному секторі застосовується широкий спектр датчиків, які використовуються практично в усіх галузях - від побутових систем до складних спеціалізованих виробничих комплексів [5]. Кількість впроваджених сенсорних пристроїв у виробництві постійно зростає, що зумовлено підвищенням поінформованості щодо їхніх можливостей і поступовим зниженням вартості.

Датчик призначений для вимірювання певного фізичного параметра. Отримане значення передається на обробку, після чого формується вихідний сигнал, який ініціює відповідну дію або зміну стану системи. Наприклад, у термостаті температурний датчик безперервно контролює температуру. Коли виміряне значення досягає встановленого рівня, система обробляє сигнал і подає команду на вимкнення котла.

У промислових умовах датчики контролюють параметри технологічних процесів і стан обладнання, формуючи базові значення нормальної роботи та фіксуючи навіть незначні відхилення. На основі зібраних даних реалізується концепція прогнозного технічного обслуговування, що дозволяє своєчасно виявляти потенційні несправності, зменшувати простої та підтримувати обладнання у працездатному стані.

Важливим елементом сучасних сенсорних систем є забезпечення зв'язку та передачі даних у режимі реального часу. Завдяки використанню високошвидкісних мережевих технологій та Інтернету забезпечується оперативний моніторинг і обробка інформації.

КПІ – 2026

Існує велика кількість типів виробничих датчиків і засобів керування процесами, кожен з яких призначений для контролю конкретних параметрів і показників ефективності. У системах промислової автоматизації кожен тип сенсора виконує визначену функцію, тому їх вибір має ґрунтуватися на особливостях обладнання та цілях стратегії прогнозного обслуговування. Не всі сенсори є необхідними для кожного застосування - їх впровадження повинно бути технічно й економічно обґрунтованим.

На рисунку 2.1 представлено приклади інформаційних моніторингових панелей, розгорнутих на різних пристроях.



Рисунок 2.1 – Структурна схема інформаційних моніторингових панелей

Триосьові акселерометри застосовуються для контролю вібрацій у обертовому обладнанні, виконуючи вимірювання швидкості, прискорення та переміщення. Отримані дані дозволяють аналізувати характер змін вібрації, зокрема узгодженість пікових і мінімальних значень. Якщо вібраційні параметри змінюються стабільно та передбачувано, це свідчить про нормальну роботу обладнання. Натомість поява нестабільності або відхилень від звичного рівня сигналізує про можливі несправності та необхідність додаткової діагностики і

технічного обслуговування.

Температура є одним із найбільш надійних показників технічного стану обладнання. Зазвичай механізми працюють у відносно стабільному температурному режимі, який підтримується за допомогою систем охолодження, таких як повітряний потік, рідина або вода. Тому раптові або неконтрольовані температурні зміни часто свідчать про зниження ефективності роботи або виникнення несправностей у певному вузлі.

Датчики крутного моменту також широко застосовуються в обертових механізмах [5]. У поєднанні з вібраційними сенсорами вони надають технічному персоналу детальну інформацію для раннього виявлення потенційних проблем, що дозволяє підвищити надійність обладнання та мінімізувати незаплановані простої. Особливу роль у цьому відіграє моніторинг стану електродвигунів, який є одним із ключових напрямів застосування промислових сенсорів. Системи контролю стану двигуна [6] дозволяють оцінювати загальний технічний стан обладнання та виявляти початкові ознаки можливих відмов. Завдяки цьому спрощується діагностика несправностей ще до того, як вони почнуть негативно впливати на продуктивність виробництва.

Апаратно-програмні платформи та мікроконтролери

Платформа Arduino

Плата мікроконтролера фактично є компактним комп’ютером, реалізованим на основі напівпровідникової мікросхеми. Незалежно від типу, усі мікроконтролери мають спільну архітектурну основу: центральний процесор (CPU), модулі введення/виведення (I/O), пам’ять та програмовані периферійні пристрої. Поєднання цих компонентів в одному корпусі забезпечує автономну обробку даних і керування зовнішніми пристроями. Завдяки своїй універсальності та компактності мікроконтролери є надзвичайно важливими елементами сучасної електроніки [7].

Сьогодні мікроконтролери широко застосовуються практично в усіх автоматизованих системах: у побутових пристроях, автомобільних електронних

КПЗ_2026

блоках керування, пультах дистанційного керування, вимірювальних приладах, електроінструментах, іграшках, медичних імплантах та інших вбудованих системах. Вони є популярним вибором серед інженерів, розробників і ентузіастів, а також активно використовуються в навчальних і DIY-проєктах.

Мікропроцесор, своєю чергою, - це мініатюрний електронний пристрій, що виконує функції центрального процесора цифрової обчислювальної системи. Він інтегрує арифметико-логічні та керуючі блоки в одній мікросхемі, яка може містити мільйони транзисторів, резисторів та інших компонентів. Саме мікропроцесор забезпечує виконання обчислювальних операцій у комп'ютерних системах.

Для реалізації задач обробки сигналів із датчиків та передавання даних до вебзастосунку для подальшої візуалізації й аналізу доцільно використовувати мікроконтролерні платформи, такі як Arduino та Raspberry Pi, які є доступними та зручними рішеннями для створення прототипів.

КОПЗ_2026

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Проєкт Arduino було започатковано у 2005 році в Італії з метою створення недорогого та простого інструменту для навчання студентів і розробки інтерактивних електронних пристроїв. Arduino відомий як платформа з відкритим апаратним і програмним забезпеченням, що дозволяє вільно використовувати схеми та програмний код.

Arduino застосовується для реалізації широкого спектра задач: зчитування сигналів із кнопок і датчиків, керування світлодіодами та двигунами, роботи з дисплеями тощо. Однією з найпопулярніших плат є Arduino UNO.

Arduino UNO базується на мікроконтролері ATmega328P і має:

– 14 цифрових входів/виходів (6 з них підтримують ШІМ);
– 6 аналогових входів;

– кварцовий резонатор 16 МГц;
– USB-інтерфейс;
– роз’єм живлення;
– ICSP-інтерфейс;
– кнопку скидання.

Плата оснащена стандартними роз’ємами з кроком 2,54 мм для підключення зовнішніх модулів, вбудованим світлодіодом та зручним USB-підключенням до ПК. Для початку роботи достатньо підключити її через USB або подати живлення від адаптера чи акумулятора [9].

Відкритість апаратної та програмної частини означає, що будь-який розробник може використовувати схеми та бібліотеки як основу для створення власних модифікацій. На відміну від попередніх версій, сучасна плата Arduino UNO використовує мікроконтролер ATmega8U2 (у новіших ревізіях — ATmega16U2) для реалізації USB-інтерфейсу замість окремого FTDI-чіпа.

Таким чином, платформи Arduino та подібні їм мікроконтролерні рішення є зручним і економічно доцільним інструментом для створення систем збору та обробки даних, зокрема в задачах промислового моніторингу та IoT-застосувань.

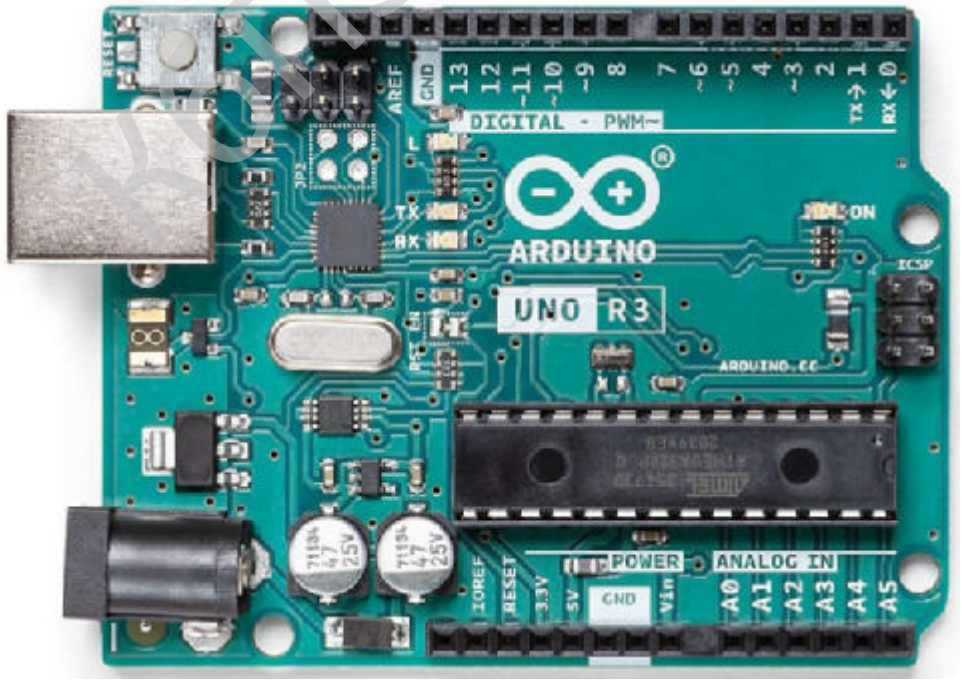


Рисунок 2.2 – Arduino UNO REV3

ВКРБ – 2026

Платформа Raspberry Pi

Raspberry Pi - це відома серія одноплатних комп'ютерів, створених британською благодійною організацією Raspberry Pi Foundation. Основною метою розробки було поширення комп'ютерної освіти, спрощення доступу до програмування та електроніки, а також популяризація інженерних знань у всьому світі. Важливо розрізняти, що Arduino є мікроконтролерною платою, тоді як Raspberry Pi — це повноцінна мікропроцесорна система (зазвичай на базі ARM Cortex-A), яка функціонує як мінікомп'ютер [10].

Raspberry Pi за своїми можливостями подібний до настільного комп'ютера та підтримує підключення різноманітних периферійних пристроїв, зокрема:

– монітора через HDMI або AV-інтерфейс;
– клавіатури та миші через USB-порти;
– камери через спеціалізований CSI-інтерфейс;
– мережі Інтернет через Ethernet або Wi-Fi.

Завдяки таким можливостям Raspberry Pi може використовуватися як компактний комп'ютер для навчання, програмування, автоматизації, IoT-проєктів, обробки даних і створення вбудованих систем. Він поєднує доступність, універсальність та достатню обчислювальну потужність, що робить його популярним серед студентів, інженерів і розробників.



Рисунок 2.3 – Raspberry Pi 4

КбПЗ_2026

Плати Raspberry Pi та подібні до них пристрої відносять до класу одноплатних комп'ютерів (SBC - Single Board Computer), оскільки всі основні компоненти комп'ютера - процесор, оперативна пам'ять, накопичувач, графічна підсистема та інтерфейси підключення - інтегровані на одній друкованій платі. Завдяки цьому Raspberry Pi фактично є повноцінним мінікомп'ютером, здатним працювати під керуванням операційної системи. Розробкою апаратної частини займається організація Raspberry Pi Foundation, яка також підтримує власний дистрибутив Linux на базі Debian - Raspberry Pi OS (раніше відомий як Raspbian).

Raspberry Pi працює під керуванням операційних систем сімейства Linux, що дозволяє розробляти програмне забезпечення різними мовами програмування, зокрема C, C++, Python, Java, HTML та іншими. Початково платформа створювалася для популяризації Python і Scratch у навчальних закладах, однак з часом стала широко використовуватися розробниками, ентузіастами та інженерами для створення:

- метеостанцій;
- робототехнічних систем;
- систем відеоспостереження;
- IoT-пристроїв;
- автоматизованих систем керування.

Однією з найдоступніших моделей є Raspberry Pi Zero W, яка забезпечує повноцінний Linux-комп'ютер із підтримкою Wi-Fi та Bluetooth за мінімальної вартості. Водночас більш продуктивні моделі, такі як Raspberry Pi 4 (8GB), мають значно вищу обчислювальну потужність, але потребують додаткових аксесуарів і плат розширення (HAT).

Порівняння Arduino та Raspberry Pi

Серед великої кількості контролерних платформ найпопулярнішими залишаються Arduino і Raspberry Pi. Обидві платформи мають процесор, пам'ять, таймери та контакти введення/виведення, однак їх призначення різне.

КБПЗ_2026

Основні відмінності:

– Arduino базується на мікроконтролерах сімейства ATmega, має просту архітектуру та орієнтований на пряме керування апаратурою (датчики, реле, двигуни). Відзначається низькою вартістю та простотою використання.

– Raspberry Pi є повноцінним одноплатним комп'ютером із операційною системою, вищою обчислювальною потужністю та ширшими можливостями обробки даних, проте його GPIO менш потужні й часто потребують додаткових схем (наприклад, транзисторів або драйверів) для керування зовнішнім обладнанням.

Arduino зазвичай виграє за критерієм мінімальної вартості плати, однак для програмування йому потрібен зовнішній комп'ютер і додаткові компоненти. Raspberry Pi, у свою чергу, забезпечує більшу функціональність і може працювати як автономна обчислювальна система.

Таким чином, вибір між Arduino та Raspberry Pi залежить від конкретного завдання: для керування апаратурою та вбудованих систем доцільніше використовувати Arduino, тоді як для обробки даних, мережевих сервісів і складніших програмних рішень більш придатним є Raspberry Pi.

Програмне забезпечення обробки інформації

Платформу Arduino можна програмувати мовами C або C++, оскільки вона використовує власну мову, побудовану на основі синтаксису C++. Водночас Arduino може взаємодіяти й з мовами високого рівня, зокрема Python. Такі поєднання особливо ефективні у проєктах, де необхідна інтеграція з датчиками, виконавчими пристроями та іншими фізичними компонентами.

Загалом використання Arduino разом із Python створює зручне навчальне середовище, що сприяє освоєнню електроніки та програмування. За наявності базових знань Python можна організувати керування платою Arduino через відповідні бібліотеки та інтерфейси, що спрощує розробку і тестування систем [12].

Платформа Arduino поєднує апаратні та програмні засоби, тому в межах

КПЗ_2026

даної роботи застосовується апаратна частина Arduino разом із програмним середовищем Python, використовуючи цифрові й аналогові входи/виходи та відповідні електричні схеми підключення. Мова програмування Python є зручною для створення автоматизованих систем моніторингу, зокрема завдяки своїй простоті, гнучкості та великій кількості бібліотек.

Крім того, Python у поєднанні з фреймворком Flask дозволяє реалізувати серверну частину системи, організувати обробку даних і взаємодію з хмарними сервісами, такими як AWS IoT, що відповідає сучасним підходам до побудови масштабованих IoT-архітектур.

Розгорнення системи у мережі Інтернет

Хмарні технології забезпечують швидкий доступ до широкого спектра інструментів і сервісів для побудови сучасних інфраструктурних рішень. Користувач може масштабувати ресурси відповідно до поточних потреб - від базових сервісів (обчислення, зберігання даних, бази даних) до рішень у сфері Інтернету речей (IoT), машинного навчання та аналітики. Використання можливостей Amazon Web Services (AWS) значно пришвидшує реалізацію проєктів, дозволяє експериментувати, тестувати нові ідеї та впроваджувати інновації без значних початкових інвестицій.

AWS позиціонується як одна з найбільш гнучких і безпечних хмарних платформ [13]. Її інфраструктура відповідає вимогам організацій із підвищеними стандартами безпеки, зокрема фінансових установ і державних структур. Платформа підтримує численні стандарти безпеки та сертифікації, надає інструменти шифрування даних, керування доступом і моніторингу, що підвищує рівень захищеності розроблюваної системи моніторингу.

Однією з актуальних проблем безпеки в хмарних середовищах є питання ротації ключів доступу. Хоча провайдери рекомендують регулярну зміну ключів, це потребує впровадження додаткових механізмів контролю відповідно до політик безпеки конкретної організації [14]. Водночас застосування таких механізмів підвищує загальний рівень захисту системи.

КПЗ-2026

Перевагою хмарних обчислень є відсутність необхідності резервувати надлишкові ресурси наперед. Замість цього інфраструктура масштабується динамічно - потужності можна швидко збільшувати або зменшувати залежно від навантаження. Контейнерні та оркестраційні сервіси AWS спрощують управління як локальною, так і хмарною інфраструктурою, дозволяючи зосередитися на розвитку функціональності системи та бізнес-процесах.

Практичним прикладом довіри до інфраструктури AWS є підписання меморандумів про співпрацю з державними структурами, що підтверджує її відповідність вимогам масштабованості, безпеки та надійності.

У межах розроблюваної автоматизованої системи моніторингу використання AWS дозволяє організувати централізований збір та аналіз даних щодо освітлення, температури, вологості, рівня шуму та вібрації. Контроль цих параметрів сприяє підтриманню стабільної та відмовостійкої роботи виробництва. Система здійснює обробку показників за допомогою сенсорів і сучасних інформаційних технологій, що забезпечує підвищення ефективності виробничих процесів.

Застосування хмарної інфраструктури AWS додатково підвищує рівень безпеки завдяки використанню механізмів шифрування, захисту мережевих з’єднань та управління доступом. Крім того, хмарна архітектура забезпечує гнучкість у подальшому розширенні функціоналу та впровадженні нових можливостей [15].

Таким чином, автоматизація контролю виробничих показників із використанням хмарних технологій дозволяє оперативно реагувати на відхилення, здійснювати глибокий аналіз стану обладнання та запобігати критичним ситуаціям, підвищуючи загальну ефективність підприємства.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну роботу, реалізації

КОПЗ – 2026

розробка програмного забезпечення для управління мікрокліматом. В процесі розробки кваліфікаційної роботи необхідно виконати наступний обсяг проекту:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику розробки відповідної системи та програмного забезпечення. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран результатів аналізу та тестування;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) сформувати висновки про виконаний обсяг робіт та одержані результати.

Висновки

У розділі було здійснено комплексний аналіз процесу розробки автоматизованої системи моніторингу та управління виробничими процесами на підприємстві. Зокрема, обґрунтовано актуальність створення такої системи, наведено приклади існуючих рішень та проаналізовано технології, що застосовуються у сфері промислового моніторингу.

Проведено дослідження сучасних систем контролю та їхніх компонентів, зокрема датчиків збору зовнішніх показників і модулів інтеграції. Також виконано порівняльний аналіз апаратно-програмних платформ, у результаті якого для реалізації проєкту обрано платформу Arduino як оптимальне рішення з точки зору вартості, функціональності та простоти впровадження.

Окрему увагу приділено аналізу можливостей хмарної інфраструктури

КОПЗ_2026

Amazon Web Services, що дозволяє забезпечити масштабованість, безпеку та надійність обробки даних у мережі Інтернет.

На основі зібраної та систематизованої інформації було розроблено схему розгортання програмного забезпечення системи обробки даних у мережевому середовищі, що стало підґрунтям для подальшого проєктування архітектури автоматизованої системи моніторингу.

КБПЗ_2026

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

Проєктування інфраструктури системи моніторингу

Під час проєктування інфраструктури системи моніторингу основною метою було створення автоматизованої системи контролю ключових параметрів виробничих процесів із використанням сенсорів та сучасних інформаційно-технічних технологій. Під робочими процесами розуміється створення матеріального продукту, що забезпечує функціонування й розвиток підприємства, а моніторинг виступає важливим інструментом підвищення ефективності цього процесу.

Як приклад можна розглянути виробниче підприємство, для якого критично важливим є дотримання заданого температурного режиму під час обробки матеріалів, зокрема в хімічній або електротехнічній галузі. Порушення кліматичних умов може призвести до зниження якості продукції або збільшення відсотка браку. Для запобігання таким ситуаціям доцільно впровадити систему моніторингу температури та вологості на виробничих ділянках. Кількість датчиків визначається параметрами середовища, що впливають на якість кінцевого продукту.

Актуальним завданням є створення автоматизованої системи контролю освітлення, температури, вологості, рівня шуму та вібрації - ключових параметрів виробничого середовища. Для реалізації такої системи передбачається використання сенсорів і сучасних інформаційних технологій із залученням хмарної інфраструктури Amazon Web Services, де здійснюватиметься розгортання програмного застосунку та обробка даних.

Хмарні технології забезпечують швидкий доступ до широкого спектра інструментів і сервісів, що дозволяє оперативно впроваджувати інновації та масштабувати ресурси відповідно до потреб системи - від обчислювальних потужностей і баз даних до сервісів IoT, аналітики та машинного навчання.

КПЗ_2026

Інфраструктура AWS відповідає високим вимогам безпеки, підтримує численні стандарти та механізми шифрування, що підвищує рівень захисту розроблюваної системи моніторингу [17].

Для підтримання технологічних норм виробництва необхідно контролювати такі параметри середовища:

- освітленість;
- температуру та вологість;
- рівень шуму;
- рівень вібрації.

3.1 Опис функціонування системи

Запропонована система передбачає взаємодію мікроконтролера з датчиками, збір та передачу даних через бездротові канали зв’язку до хмарного сервісу, де інформація обробляється, зберігається та відображається у програмному забезпеченні. У подальших розділах роботи описується реалізація такої автоматизованої системи моніторингу виробничих процесів, структурну схему якої наведено на рисунку 3.1.

КОПЗ_2026

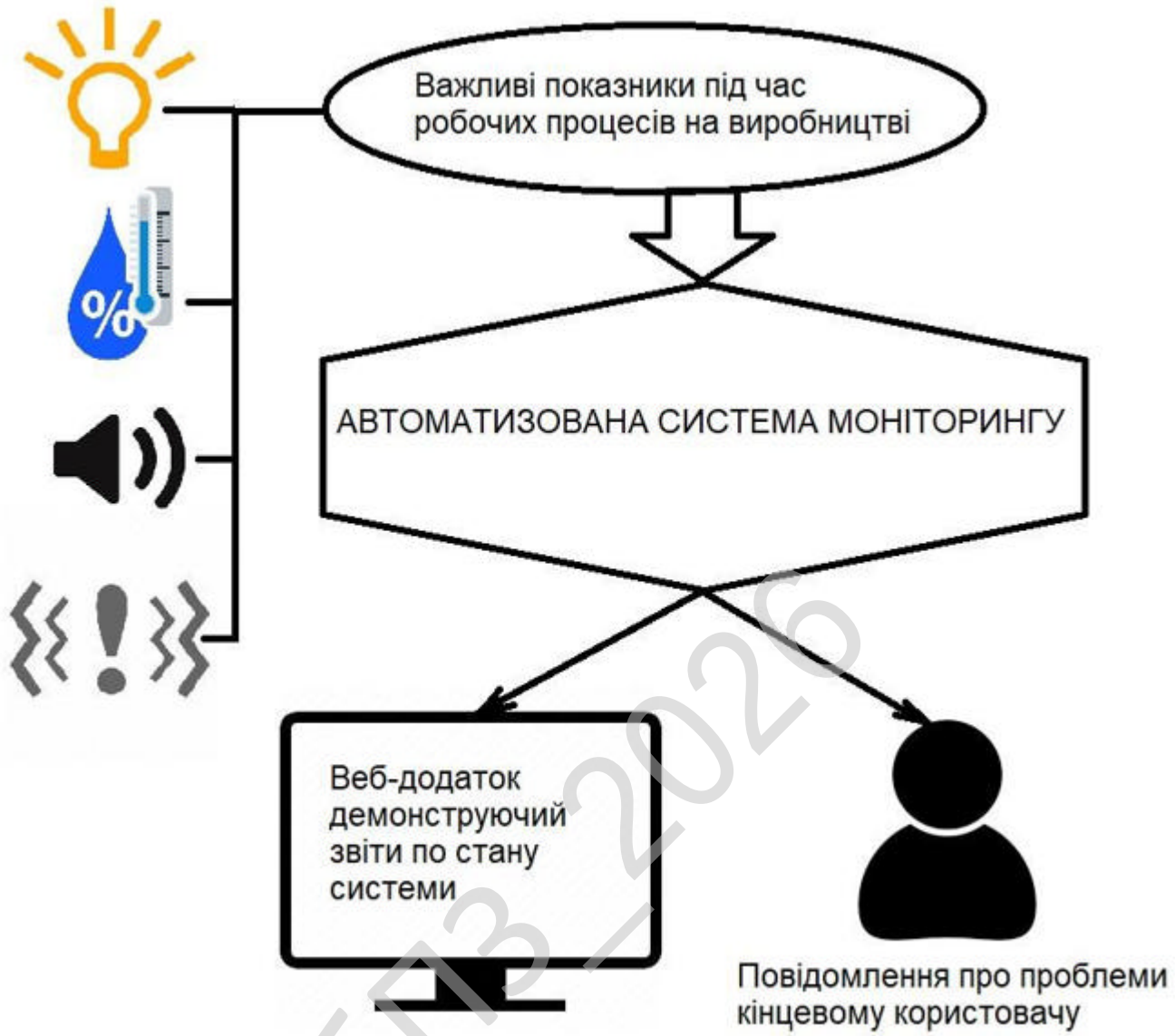


Рисунок 3.1 – Функції автоматизованої системи моніторингу

Послідовність і принцип дії автоматизованої системи

Для автоматизованої обробки даних із датчиків шуму, освітлення, температури, вологості та вібрації пропонується створення системи, яка враховує недоліки сучасних аналогів і підвищує ефективність за рахунок використання перевірених компонентів та оптимізованої архітектури [18].

Після аналізу технічної документації, практичних відгуків щодо роботи датчиків у виробничих умовах, а також існуючих мікроконтролерних і апаратно-програмних рішень було визначено склад запропонованої системи

КБПЗ_2026

автоматизованого контролю та обліку показників (рисунок 3.2).

До складу системи входять

Мікроконтролерний модуль

Центральним елементом системи є мікроконтролер (наприклад, ESP32 або Arduino), який виконує:

– зчитування сигналів із датчиків;

– попередню обробку та фільтрацію даних;

– передачу інформації до сервера або хмарної платформи;

– керування допоміжними модулями системи.

Датчики контролю середовища:

– датчик температури та вологості - для контролю мікроклімату;

– датчик освітленості - для контролю умов освітлення;

– датчик шуму - для визначення рівня акустичних навантажень;

– датчик вібрації - для діагностики роботи обладнання та виявлення відхилень.

Комунікаційний модуль - забезпечує передачу даних через Wi-Fi, Ethernet або інші бездротові технології до серверної частини або хмарної інфраструктури.

Сервер або хмарна платформа

Виконує:

– прийом та збереження даних;

– аналіз і візуалізацію показників;

– формування попереджень при перевищенні порогових значень;

– довгострокове зберігання статистики.

Програмне забезпечення:

– модуль збору та обробки даних;

– інтерфейс моніторингу (веб-панель або SCADA-подібна система);

– система сповіщень і журналювання подій.

Запропонована структура системи дозволяє здійснювати безперервний моніторинг виробничих параметрів, оперативно реагувати на відхилення від

КбПЗ_2026

норми та підвищувати ефективність і надійність виробничих процесів. У подальшому (рисунок 3.2) наведено структурну схему взаємодії зазначених компонентів системи.

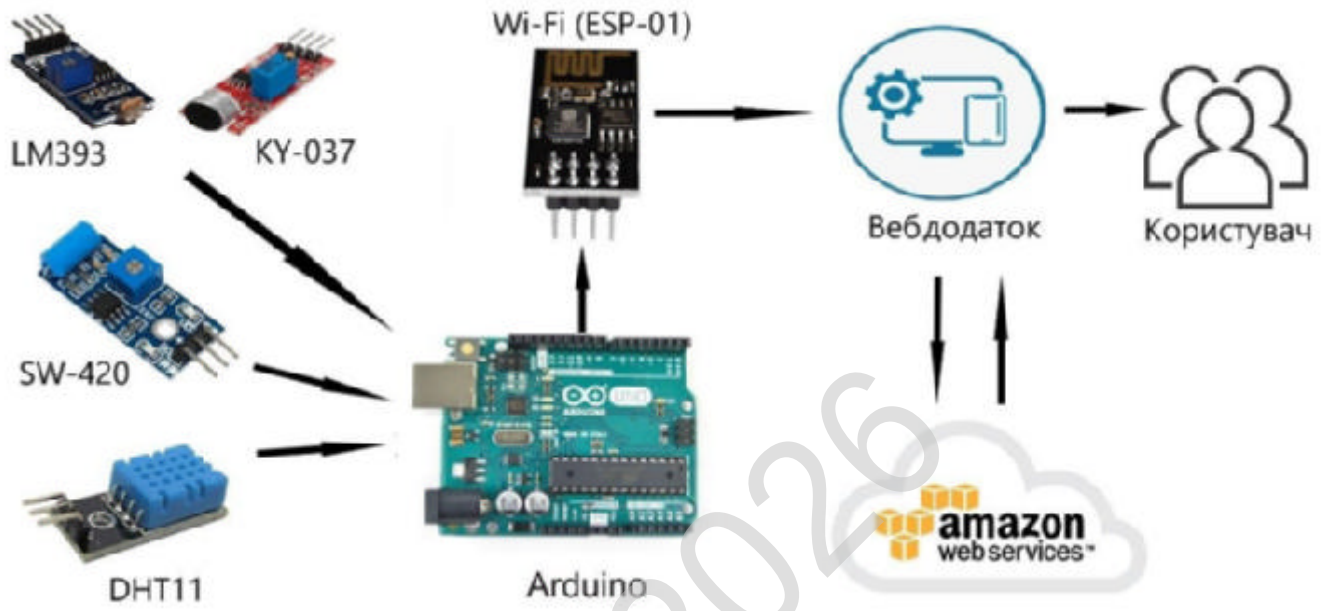


Рисунок 3.2 - Структурна схема системи

3.3 Розробка функціональної схеми

Wi-Fi модуль ESP-01 є одним із найпоширеніших представників серії ESP8266. Керуючий пристрій взаємодіє з ним через інтерфейс UART (Serial-порт), використовуючи набір АТ-команд. Обмін даними за принципом роботи подібний до взаємодії з TCP-сокетом або послідовним портом комп'ютера. Прошивку модуля можна завантажувати через Arduino IDE [19], аналогічно до плат Arduino. Стандартна реакція на АТ-команди реалізована у заводській прошивці. Завдяки наявності двох портів введення/виведення після перепрошивки модуль може працювати без окремого мікроконтролера, безпосередньо керуючи підключеною периферією. Для прошивки достатньо звичайного програматора. Вбудована РСВ-

КОПЗ - 2026

антена забезпечує дальність зв’язку до 400 м на відкритому просторі. Модуль широко використовується у стаціонарних і мобільних пристроях, системах «розумного будинку», носимій електроніці та роботизованих комплексах. Підтримка Wi-Fi та енергозбереження робить його придатним для автономних систем із живленням від акумуляторів або сонячних батарей. У даному проєкті ESP-01 застосовується для передавання даних від мікроконтролера до вебзастосунку, розгорнутого у хмарному середовищі AWS.

Датчик LM393 призначений для визначення відстані до об’єктів і рівня освітленості в люксах. До складу модуля входять інфрачервоний світлодіод із керованим драйвером, фотодіоди для вимірювання загальної та інфрачервоної освітленості, підсилювачі, мікроконтролер, АЦП, пам’ять та контролер шини I2C (SDA, SCL). У проєктах Arduino він може застосовуватися для автоматичного регулювання яскравості екранів або визначення наближення об’єктів.

DHT11 - цифровий датчик температури та вологості, що містить термістор, ємнісний сенсор вологості та вбудований АЦП. Хоча датчик не відзначається високою точністю і швидкодією, він простий, недорогий і придатний для базового контролю мікроклімату.

Датчик звуку KY-037 характеризується низьким енергоспоживанням (3,1–6 мА) і автоматичним регулюванням шуму: слабкі сигнали підсилюються, а надмірно сильні приглушуються. Вбудований мікрофон сприймає звуки у діапазоні 22 Гц – 22 кГц. Підключення здійснюється трьома контактами: живлення, земля та вихід. Контакт GAIN дозволяє змінювати коефіцієнт підсилення, а AR - налаштовувати час спрацьовування.

Датчик вібрації SW-420 використовується для виявлення ударів і механічних коливань. Чутливість регулюється підстроювальним резистором. Модуль має два індикатори: один сигналізує про наявність живлення, інший - про спрацювання. На відміну від багатьох аналогів, датчик реагує навіть на одиничний поштовх, що робить його ефективним для систем безпеки та моніторингу обладнання.

КПІЗ_2026

На основі перелічених датчиків реалізується система автоматизованого збору та обробки показників із подальшою побудовою графіків у реальному часі [20]. Дані передаються до програмного забезпечення, розгорнутого у хмарному середовищі AWS, де відображаються у вебінтерфейсі. Такий підхід дозволяє користуватися системою без встановлення додаткового ПЗ - достатньо доступу до браузера.

Хмарні обчислення забезпечують гнучке масштабування ресурсів відповідно до потреб підприємства без попереднього резервування потужностей. Контейнерні та оркестраційні сервіси AWS спрощують керування інфраструктурою та дозволяють зосередитися на розвитку функціоналу системи.

Для роботи з пристроями Інтернету речей сервіс AWS IoT Core підтримує такі протоколи:

– MQTT;
– MQTT over WSS;
– HTTPS;
– LoRaWAN.

Схема взаємодії мікроконтролера Arduino з хмарною інфраструктурою AWS та системою візуалізації наведена на функціональній схемі рисунок 3.3. Для відображення та аналізу метрик використовується Grafana, яка дозволяє будувати інформативні графіки, оцінювати стан системи та інтегрується з сервісами AWS [21]. Це забезпечує зручний, надійний і масштабований інструмент моніторингу виробничих показників у реальному часі.

КПЗ – 2026

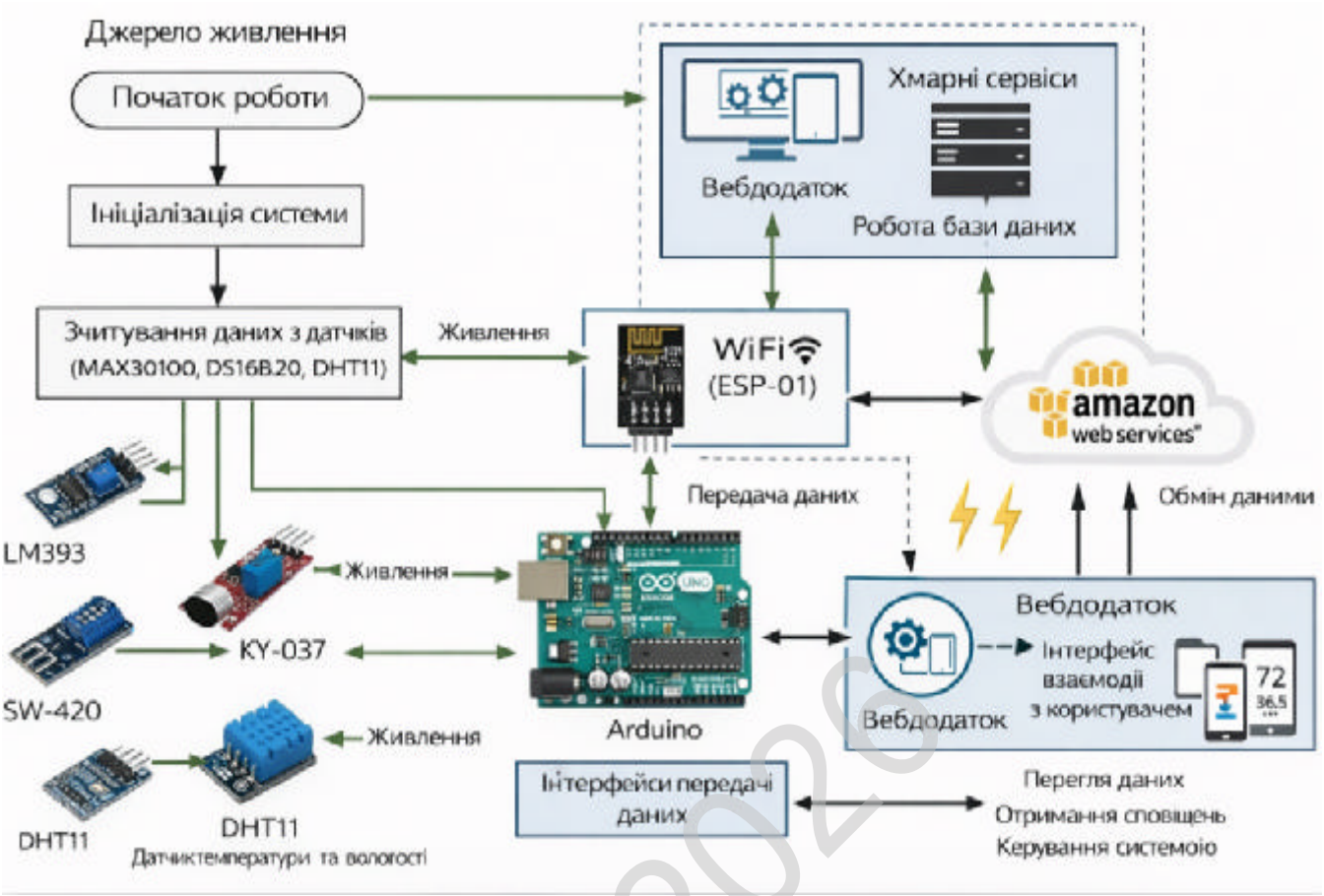


Рисунок 3.3 - Функціональна схема системи керування

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів автоматизованої системи моніторингу та управління виробничими процесами на підприємстві відображає логіку функціонування програмного забезпечення та взаємозв'язки між основними етапами обробки даних, починаючи від моменту запуску системи і завершуючи отриманням інформації користувачем та можливістю впливу на систему. У центрі всієї системи знаходиться мікроконтролер, який виконує роль основного керуючого елемента, забезпечуючи обробку інформації, координацію роботи датчиків та організацію передачі даних до зовнішніх сервісів.

Робота системи починається з моменту її запуску, після чого відбувається ініціалізація, під час якої налаштовуються всі необхідні апаратні та програмні компоненти. На цьому етапі здійснюється конфігурація портів вводу-виводу,

КОПЗ_2026

ініціалізація модулів зв’язку, а також підготовка датчиків до роботи. Це дозволяє забезпечити коректне функціонування всієї системи в подальшому та уникнути помилок при зчитуванні даних.

Після завершення ініціалізації система переходить до етапу зчитування даних з датчиків. У даній системі використовуються різні типи сенсорів, які відповідають за контроль параметрів виробничого середовища. Зокрема, датчики освітленості дозволяють визначати рівень освітлення, датчики звуку фіксують рівень шуму, датчики вібрації контролюють механічні коливання обладнання, а датчики температури та вологості забезпечують вимірювання кліматичних параметрів. Усі ці дані передаються до мікроконтролера для подальшої обробки.

Зібрані дані надходять до блоку збору інформації, де вони акумулюються та структуруються. Далі відбувається обробка даних мікроконтролером, яка включає фільтрацію шумів, усунення похибок та перетворення сигналів у формат, придатний для подальшого аналізу. На цьому етапі система також може виконувати базові обчислення, що дозволяють отримати більш інформативні показники на основі первинних даних.

Після обробки даних система переходить до етапу їх передачі. Для цього використовується модуль бездротового зв’язку, який забезпечує передачу інформації до вебдодатку або хмарних сервісів. Передача даних здійснюється у реальному часі або з певною періодичністю, залежно від налаштувань системи. Це дозволяє забезпечити постійний моніторинг стану виробничих процесів та оперативно реагувати на зміни.

Отримані дані надходять до вебдодатку, який виступає інтерфейсом взаємодії між системою та користувачем. Вебдодаток забезпечує відображення інформації у зручному вигляді, що може включати графіки, таблиці та інші форми представлення даних. Крім того, через вебінтерфейс користувач має можливість здійснювати керування системою, змінювати параметри роботи та отримувати сповіщення про виникнення аварійних або критичних ситуацій.

КБПЗ – 2026

Паралельно з цим відбувається взаємодія з хмарними сервісами, які забезпечують збереження даних, їх обробку та можливість доступу до інформації з будь-якого пристрою. Хмарна інфраструктура дозволяє накопичувати історичні дані, виконувати їх аналіз та використовувати для прийняття управлінських рішень. Крім того, вона забезпечує масштабованість системи та підвищує її надійність.

Важливою складовою системи є процес контролю та формування сповіщень. На основі отриманих даних система може визначати відхилення від нормальних параметрів та формувати відповідні сигнали. Це можуть бути попередження про перевищення допустимих значень або критичні повідомлення, що потребують негайного втручання. Такі сповіщення передаються користувачу через вебдодаток або інші засоби комунікації.

Користувач, у свою чергу, отримує доступ до всієї інформації через персональні пристрої, такі як комп'ютер, смартфон або планшет. Він може переглядати поточні дані, аналізувати їх динаміку та здійснювати керування системою. Таким чином, забезпечується двостороння взаємодія між користувачем і системою, що є важливою умовою ефективного функціонування автоматизованого комплексу.

Усі процеси в системі організовані у вигляді безперервного циклу, що включає зчитування даних, їх обробку, передачу та аналіз. Це дозволяє забезпечити постійний контроль за станом виробничих процесів та своєчасне реагування на будь-які зміни. Така організація роботи підвищує ефективність управління підприємством, знижує ризики виникнення аварійних ситуацій та сприяє оптимізації виробничих процесів.

Таким чином, діаграма взаємодії процесів відображає комплексну структуру роботи системи, демонструючи взаємозв'язки між її компонентами та послідовність виконання основних функцій. Вона дозволяє наочно представити логіку роботи програмного забезпечення, що є важливим етапом при розробці,

КПЗ – 2026

аналізі та впровадженні автоматизованих систем моніторингу та управління на підприємстві.

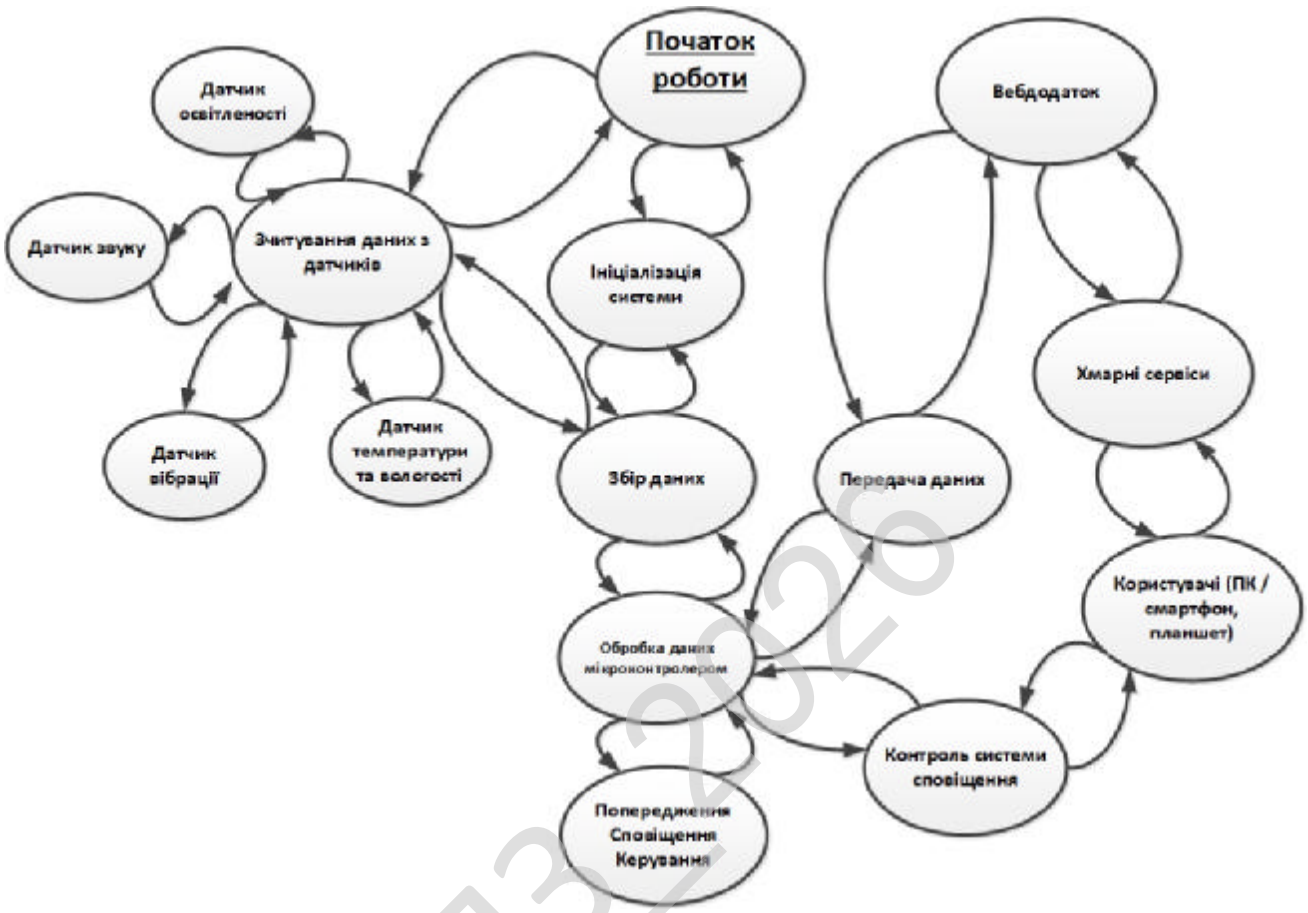


Рисунок 3.4 – Діаграма взаємодії процесів

КОПЗ 2026

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

За останні роки Python став однією з найпопулярніших мов програмування у світі. Його застосовують у багатьох галузях - від машинного навчання та аналізу даних до веброзробки, автоматизації процесів і тестування програмного забезпечення. Python використовується як професійними розробниками, так і користувачами без глибокої технічної підготовки завдяки простому синтаксису та широкій екосистемі бібліотек. На основі Python створено численні відомі системи - від алгоритмів рекомендацій потокових сервісів до програмного забезпечення для автономних транспортних засобів. Як мова загального призначення, Python придатний для вирішення широкого спектра задач, включаючи обробку даних, вебзастосунки, автоматизацію та наукові обчислення.

У веброзробці Python часто використовується для створення серверної частини (backend) вебсайтів і додатків. Його функції можуть включати обмін даними між клієнтом і сервером, обробку інформації, роботу з базами даних, маршрутизацію URL-запитів і реалізацію механізмів безпеки. Для спрощення розробки вебзастосунків Python пропонує спеціалізовані фреймворки, серед яких найпоширенішими є Django та Flask. Вебфреймворк являє собою набір бібліотек і модулів, що дозволяють створювати вебдодатки без необхідності реалізовувати низькорівневі механізми, такі як обробка протоколів або керування потоками [22].

Фреймворк Flask є легким мікрофреймворком, який дозволяє швидко створювати вебзастосунки. Його ядро мінімальне, але розширюване, що дає змогу підключати додаткові модулі відповідно до потреб проєкту. Flask не містить вбудованої ORM або складних компонентів, надаючи розробнику свободу вибору технологій - бази даних, засобів взаємодії з нею та архітектури застосунку. До ключових можливостей Flask належать маршрутизація URL-запитів, система

КПЗ – 2026

шаблонів для формування вебсторінок і гнучкість конфігурації [23].

Попри те, що Flask є мікрофреймворком, це не обмежує розмір або складність проєкту. Застосунок може складатися з багатьох модулів і масштабуватися до повноцінних вебсистем. Термін micro означає простоту та мінімалізм базового ядра, а не обмеження функціональності. Flask широко використовується в сучасних вебпроєктах завдяки своїй гнучкості, актуальності та можливості розширення, що робить його ефективним інструментом для розробки серверної частини автоматизованих систем моніторингу та обробки даних.

Аналіз особливостей розгортання додатку у AWS

Звичайно, можна було б обмежитися використанням простих вимірювальних приладів, які безпосередньо відображають значення параметрів, а отримані дані заносити до паперового журналу вручну. Такий підхід є простішим у реалізації та потенційно дешевшим, однак він має низку суттєвих недоліків, пов’язаних насамперед із впливом людського фактору. Ручне ведення журналу супроводжується ризиком помилок під час вимірювання, пропусків записів, неточностей обчислень і суб’єктивності оцінки показників. Крім того, людина фізично не завжди може здійснювати контроль у важкодоступних або небезпечних зонах, де електронні сенсори можуть працювати безперервно та безпечно.

Окремою проблемою є організація зберігання та обміну інформацією. Паперові журнали ускладнюють передачу даних, їх аналіз і резервне копіювання, що знижує оперативність реагування на критичні ситуації та підвищує ризик втрати важливої інформації. В умовах сучасних виробничих і інформаційних систем такі обмеження є суттєвими.

Таким чином, доцільним є впровадження автоматизованої системи моніторингу, яка забезпечує безперервний контроль параметрів без постійної участі людини. Усі дані збираються у цифровому вигляді, що значно спрощує їх передачу, обробку, зберігання та аналіз. Інтеграція з хмарними вебсервісами

КОПЗ_2026

забезпечує централізований доступ до інформації, можливість резервного копіювання та ефективну взаємодію з даними в реальному часі, що підвищує надійність і ефективність функціонування системи.

WS IoT

AWS IoT Core - це хмарна платформа, призначена для підключення різноманітних пристроїв до сервісів хмарної інфраструктури AWS. Вона забезпечує спеціальний інтерфейс, який дозволяє пристроям здійснювати безпечний і надійний двосторонній обмін даними з сервісами AWS навіть у випадках тимчасової відсутності підключення до мережі.

AWS надає широкий спектр рішень і сервісів Інтернету речей (IoT), що забезпечують підключення, моніторинг і керування мільярдами пристроїв. Хмарні служби AWS IoT дозволяють інтегрувати IoT-пристрої між собою та з іншими сервісами AWS. Для цього доступне спеціалізоване програмне забезпечення, яке спрощує інтеграцію пристроїв у хмарні рішення AWS IoT. Після підключення пристроїв до платформи AWS IoT вони можуть взаємодіяти з такими сервісами AWS, як S3 Bucket, Lambda, DynamoDB та іншими (рис. 4.1).

Amazon DynamoDB - це повністю керована безсерверна NoSQL база даних типу «ключ-значення», створена для роботи з високонавантаженими застосунками незалежно від масштабу системи. DynamoDB підтримує обробку як структурованих, так і напівструктурованих даних, зокрема у форматі JSON. Водночас SQL залишається стандартною мовою для збереження та отримання інформації в реляційних базах даних, які забезпечують широкий набір інструментів для розробки програмних систем, що працюють із базами даних.

КПЗ-2026

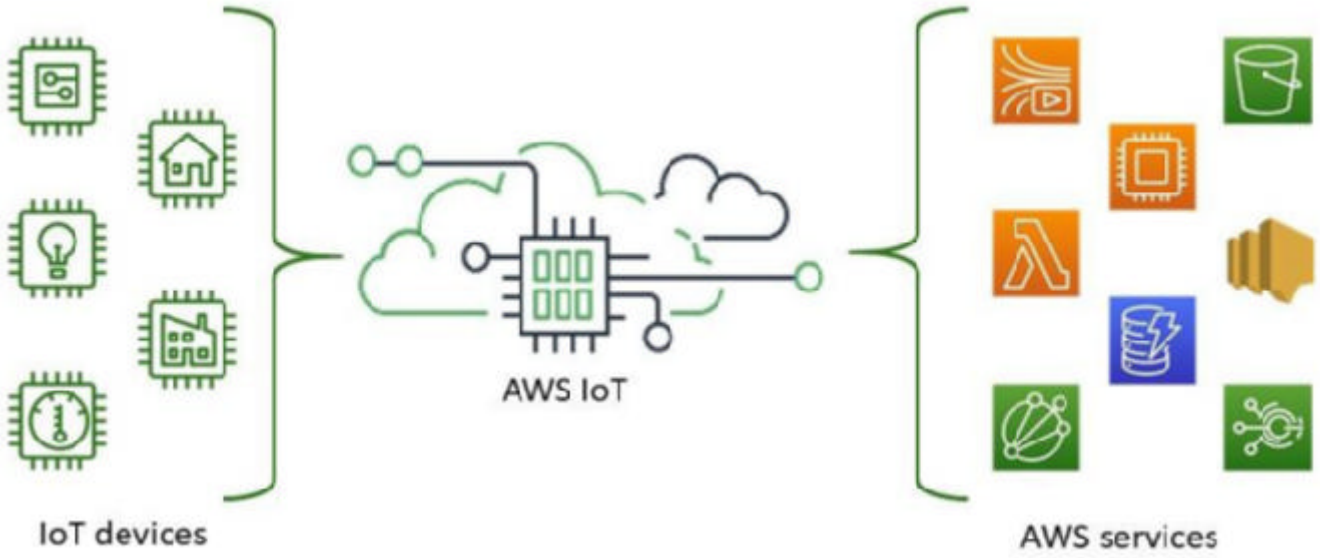


Рисунок 4.1 – Схема роботи AWS IoT

AWS IoT надає можливість обирати найбільш оптимальні та сучасні технології відповідно до потреб конкретного рішення, що значно спрощує процес керування IoT-пристроями та їх подальшу підтримку.

Брокер повідомлень AWS IoT Core забезпечує взаємодію пристроїв і клієнтських застосунків, які використовують протоколи MQTT та MQTT через WSS для обміну повідомленнями за принципом публікації та підписки. Крім того, підтримується передавання повідомлень за допомогою протоколу HTTPS.

Функціональність AWS IoT Core для LoRaWAN дозволяє підключати й керувати бездротовими пристроями стандарту LoRaWAN - енергоефективної мережі великого радіусу дії. Використання цього сервісу усуває необхідність самостійного створення та обслуговування мережевого сервера LoRaWAN (LNS).

У випадку, якщо підприємству не потрібні можливості AWS IoT, пов’язані зі зв’язком пристроїв, правилами обробки або керуванням завданнями, можна скористатися сервісами AWS Messaging, які пропонують альтернативні рішення для обміну повідомленнями відповідно до потреб організації.

Сервіси AWS IoT Core забезпечують підключення IoT-пристроїв як до служб AWS IoT, так і до інших сервісів AWS. До складу AWS IoT Core входять шлюз пристроїв і брокер повідомлень, які відповідають за встановлення з’єднання

КОПЗ_2026

та обробку даних між IoT-пристроями і хмарною інфраструктурою (рисунок 4.2).

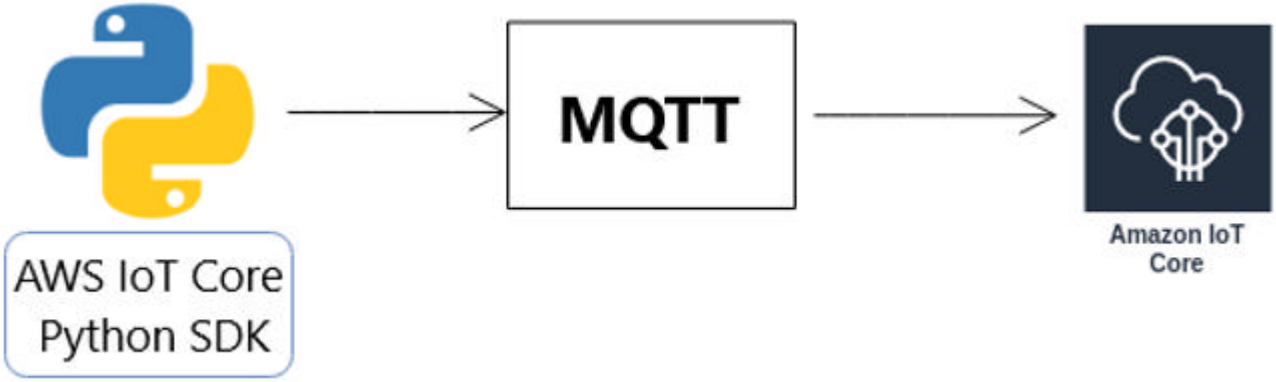


Рисунок 4.2 – Схема публікування MQTT повідомлень в AWS IoT CORE за допомогою Python

Переваги розробки та впровадження рішень із використанням технологій AWS IoT пояснюються тим, що AWS IoT Core є керованим хмарним сервісом, який забезпечує просту та безпечну взаємодію підключених пристроїв із хмарними застосунками та іншими пристроями. Платформа здатна обслуговувати мільярди пристроїв і обробляти трильйони повідомлень, гарантуючи їх надійну передачу та маршрутизацію до сервісів AWS або кінцевих пристроїв.

MQTT (Message Queuing Telemetry Transport) - це легкий і поширений протокол обміну повідомленнями, створений спеціально для пристроїв із обмеженими ресурсами [24]. Підтримка MQTT у AWS IoT Core реалізована відповідно до специфікацій MQTT версій 3.1.1 та 5.0 із певними особливостями реалізації. Остання версія стандарту MQTT 5 містить низку важливих удосконалень, що підвищують надійність системи, зокрема покращену масштабованість, розширені механізми обробки помилок із кодами причин, можливість встановлення часу життя повідомлень і сеансів, а також використання користувацьких заголовків. AWS IoT Core також підтримує сумісність між різними версіями протоколу MQTT, що дозволяє пристроям із MQTT 3 взаємодіяти з клієнтами MQTT 5 і навпаки.

AWS IoT Core забезпечує підключення пристроїв, які використовують

КОПЗ_2026

протоколи MQTT або MQTT через WebSocket Secure (WSS), із застосуванням унікального ідентифікатора клієнта. Для спрощення інтеграції рекомендується використовувати AWS IoT Device SDK, який підтримує необхідні функції підключення, обміну даними та доступу до сервісів AWS IoT. SDK також реалізує механізми автентифікації та вимоги до встановлення з'єднання, передбачені протоколами MQTT і WSS.

Хоча використання AWS IoT Device SDK є рекомендованим, воно не є обов'язковим. У разі його відсутності розробник повинен самостійно забезпечити належний рівень безпеки з'єднання та передачі даних. Зокрема, клієнтські пристрої повинні передавати розширення TLS Server Name Indication (SNI) під час встановлення з'єднання, інакше підключення буде відхилено. Крім того, клієнти, що використовують користувачів IAM та облікові дані AWS для автентифікації, мають застосовувати механізм підпису запитів Signature Version 4 (рис. 4.3).

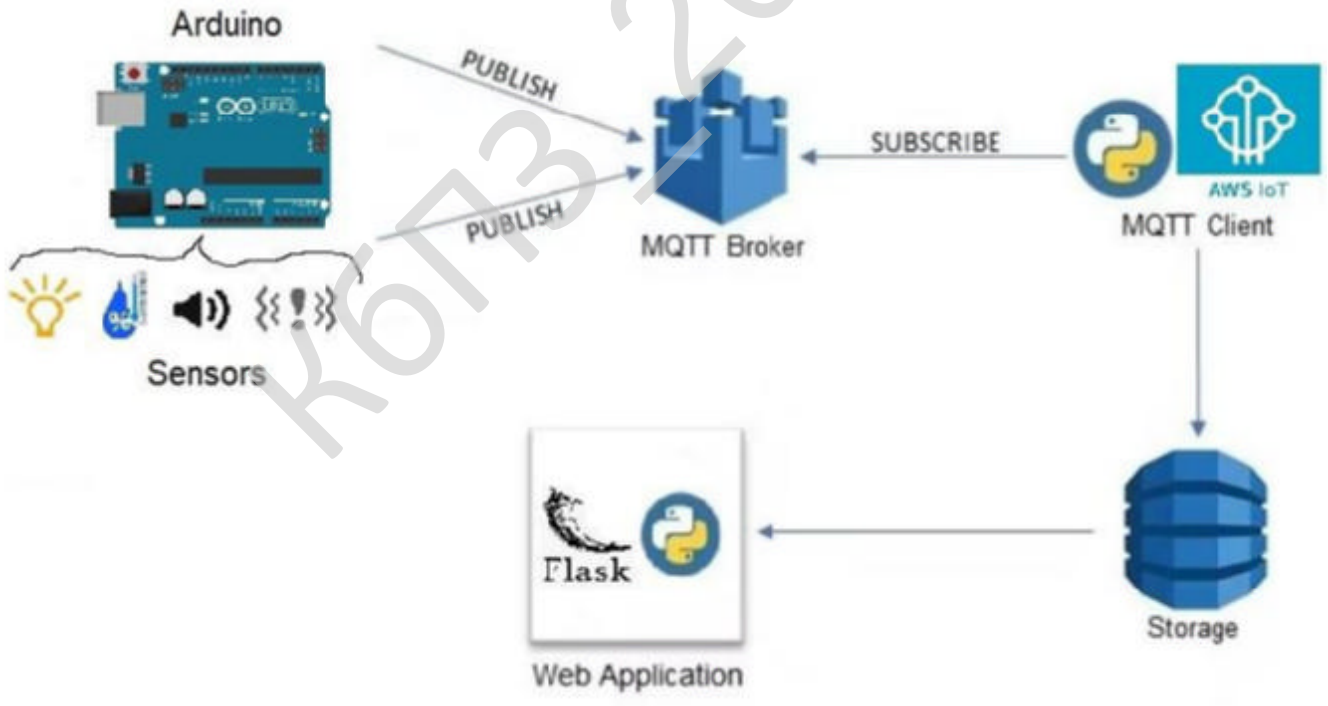


Рисунок 4.3 – Схема роботи системи IoT за допомогою технології MQQT

КбПЗ 2026

4.1 Створення загальної архітектури автоматизованої системи моніторингу

Було проаналізовано різні типи систем Інтернету речей, що застосовуються або впроваджуються у середовищах моніторингу в режимі реального часу. Особливу увагу приділено рішенням, побудованим на основі хмарних обчислень, а також визначено їхні спільні характеристики. Описано основні компоненти IoT-системи та розглянуто принцип їх взаємодії під час здійснення моніторингу параметрів навколишнього середовища в реальному часі.

Для побудови цілісної структури системи виконано формалізацію її складових на різних рівнях, що дозволило сформувати узагальнену архітектуру IoT-системи моніторингу. Запропонована архітектура включає чотири основні рівні: рівень сприйняття даних, рівень шлюзів і зв'язку, рівень хмарних обчислень та сервісного обслуговування, а також рівень прикладних застосунків.

Запропонована модель архітектури може використовуватися як універсальне рішення для систем моніторингу критично важливих параметрів у виробничому середовищі.

На схемі архітектури автоматизованої системи моніторингу відображено загальний принцип її функціонування та структуру інфраструктури. Система містить апаратний рівень, на якому використовуються датчики разом з апаратно-програмною платформою Arduino. Завдяки середовищу розробки Arduino IDE здійснюється програмування сенсорів і передавання зібраних даних через Wi-Fi модуль.

Передані дані надходять до брокера повідомлень MQTT, який за допомогою однойменного протоколу виконує їх обробку та передає інформацію з апаратного комплексу до хмарного середовища AWS із використанням сервісу AWS IoT Core.

У хмарній інфраструктурі реалізується логіка роботи вебзастосунку, що приймає дані із сенсорів, забезпечує їх обробку, візуалізацію у вигляді графіків та

КПЗ_2026

функції керування системою. Основою роботи є програмне забезпечення, розроблене мовою Python із застосуванням вебфреймворку Flask, інтегрованого з AWS IoT Core. Для підвищення стабільності та швидкодії системи використовується база даних DynamoDB, яка забезпечує мінімальний час обробки запитів і швидкий доступ до збережених показників, що підвищує загальну ефективність роботи системи.

Розробка комплексу апаратного забезпечення на платформі Arduino

Під'єднання компонентів до Arduino

Основою апаратно-програмного комплексу обробки даних обрано мікроконтролер ATmega328P, реалізований на платформі Arduino UNO [26]. Дана плата оснащена наборами цифрових та аналогових входів/виходів (I/O), що дозволяє підключати різноманітні модулі розширення (shield) та зовнішні електронні компоненти. Arduino UNO має 14 цифрових контактів вводу/виводу, з яких шість підтримують широтно-імпульсну модуляцію (ШІМ), а також 6 аналогових входів. Програмування плати здійснюється через інтегроване середовище розробки Arduino IDE за допомогою USB-кабелю типу B. Живлення може подаватися як через USB-порт, так і від зовнішнього джерела напруги 9 В, при цьому допустимий діапазон живлення становить від 7 до 20 В. За функціональними можливостями плата є подібною до Arduino Nano та Arduino Leonardo. Еталонна апаратна схема поширюється за ліцензією Creative Commons Attribution Share-Alike 3.3 і доступна на офіційному вебресурсі Arduino.

До плати Arduino UNO підключаються датчики, що забезпечують отримання необхідних параметрів навколишнього середовища для системи моніторингу. Зокрема, використано датчик освітленості на базі компаратора LM393, датчик температури та вологості DHT11 (другої версії), а також датчик вібрації SW-420 із нормально замкнутим контактом. Варто зазначити, що датчик шуму (модуль мікрофонного підсилювача KY-037), який планувався до використання, не був включений до системи через його відсутність. Для забезпечення бездротового підключення застосовується Wi-Fi модуль ESP8266

КПЗ_2026

(ESP-01), що працює через послідовний інтерфейс.

Оскільки апаратний комплекс містить значну кількість компонентів, подальше представлення виконано у вигляді окремих схем підключення для кожного елемента.

Контакти VCC і GND [26] використовуються для подачі живлення на модуль. Для коректної роботи модуль потребує напруги живлення 3,3 В. Вивід RX модуля підключається до цифрового контакту 0 плати Arduino, а вивід TX - до контакту 1. Перед подачею живлення необхідно обов’язково перевірити правильність підключення ліній живлення та заземлення. Після успішного підключення та подачі напруги на модулі має засвітитися індикатор живлення червоного кольору.

Для підключення датчика необхідно використати резистор номіналом 10 кОм і з’єднувальні провідники. У даному випадку застосовується чотириконтактна версія DHT11. Сигнальний контакт датчика (жовтий провід) підключається до входу живлення 3 В плати Arduino UNO. Резистор номіналом 10 кОм встановлюється між першим і другим контактами датчика DHT11. Третій контакт датчика не використовується, а контакт GND (чорний провід) підключається до відповідного виводу заземлення Arduino UNO.

Наступним етапом є підключення модуля SW-420, який використовується як датчик вібрації у складі апаратного комплексу системи моніторингу. Модуль SW-420 містить сенсор вібрації, компаратор напруги LM393, потенціометр для регулювання чутливості, струмообмежувальні резистори, що виконують функцію подільника напруги, а також конденсатори, призначені для зміщення сигналу та фільтрації шумів.

Для налаштування чутливості датчика на модулі передбачено потенціометр номіналом 10 кОм. Рівень чутливості можна змінювати відповідно до поставлених вимог шляхом встановлення необхідного порогового значення, яке використовується компаратором LM393 як еталонне під час порівняння сигналів.

КПЗ_2026

Вібраційний перемикач SW-420 визначає рівень механічних коливань у навколишньому середовищі, реагуючи на них замиканням або розмиканням електричного контакту. Тригерний елемент може бути реалізований у вигляді електромеханічного, релейного або напівпровідникового компонента.

Модуль оснащений двома світлодіодними індикаторами: перший сигналізує про подачу живлення, а другий відображає стан цифрового виходу датчика. Для наочного підтвердження працездатності апаратного комплексу додатково було підключено стандартний світлодіод, який використовується як індикатор роботи системи.

Даний модуль оснащений високочутливим сенсором вібрації та вбудованим компаратором напруги, який формує цифровий вихідний сигнал. У разі, коли вібраційний перемикач перебуває у замкненому стані, на виході встановлюється низький логічний рівень, при цьому світиться зелений індикатор. Якщо ж перемикач розмикається, вихідний сигнал переходить у високий логічний рівень, а зелений світлодіод гасне.

Модуль може бути безпосередньо підключений до мікроконтролера, що дає змогу зчитувати стан виходу та швидко визначати наявність або відсутність вібрації у навколишньому середовищі. Таким чином, вихідний сигнал датчика відображає факт виявлення механічних коливань.

Загальний вигляд апаратного комплексу з усіма компонентами, підключеними до плати Arduino UNO відповідно до наведених схем та описаних інструкцій, представлено на рисунку 4.4.

КПЗ 2026

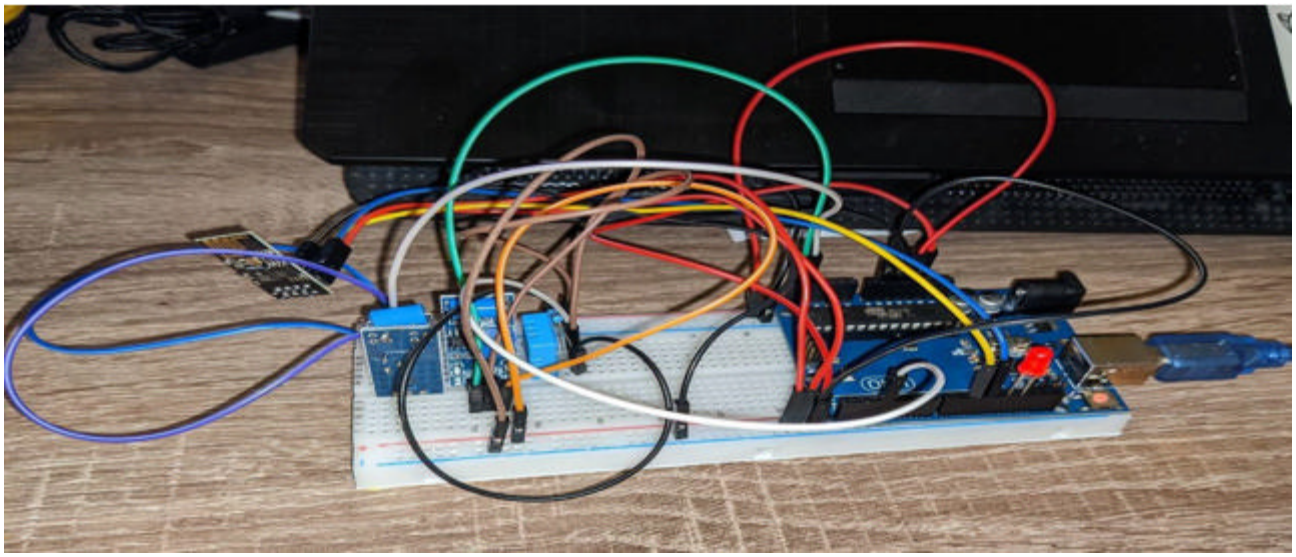


Рисунок 4.4 – Демонстрація під’єднання всіх елементів Arduino

4.2 Розробка блок-схем та опис алгоритмів функціонування системи

Алгоритм роботи програмного забезпечення описує послідовність дій, які виконує система для збору, обробки, аналізу та передачі даних, а також взаємодії з користувачем. Робота системи організована за циклічним принципом, що забезпечує безперервний моніторинг виробничих параметрів.

Після подачі живлення відбувається запуск системи та ініціалізація програмного середовища мікроконтролера. На цьому етапі здійснюється налаштування всіх необхідних ресурсів, зокрема конфігурація портів вводу-виводу, запуск інтерфейсів зв’язку, ініціалізація датчиків та підключення до бездротової мережі Wi-Fi. Також виконується перевірка працездатності всіх підключених модулів. У випадку виникнення помилки система може повторити ініціалізацію або перейти в режим очікування.

Після успішної ініціалізації система переходить до основного циклу роботи. На першому етапі циклу здійснюється зчитування даних з датчиків. Мікроконтролер послідовно опитує всі підключені сенсори, серед яких датчик освітленості, датчик звуку, датчик вібрації та датчик температури і вологості.

КОПЗ_2026

Отримані значення зчитуються у вигляді аналогових або цифрових сигналів і зберігаються у внутрішніх змінних системи.

Далі виконується обробка отриманих даних. На цьому етапі здійснюється фільтрація шумів, усунення випадкових коливань сигналів та нормалізація значень. При необхідності застосовуються алгоритми усереднення або згладжування для підвищення точності вимірювань. Це дозволяє отримати більш стабільні та достовірні дані для подальшого аналізу.

Після обробки виконується аналіз даних. Система порівнює отримані значення з встановленими пороговими значеннями, які відповідають нормальному режиму роботи обладнання. У разі перевищення допустимих меж визначається характер відхилення та формується відповідний стан системи. Це може бути нормальний режим, попередження або аварійний стан. Такий підхід дозволяє оперативно виявляти небезпечні ситуації та реагувати на них.

На основі результатів аналізу система переходить до формування керуючих впливів та інформаційних повідомлень. Якщо параметри знаходяться в межах норми, система продовжує роботу без змін. У разі виявлення відхилень формуються сповіщення для користувача, а також можуть генеруватися керуючі сигнали для виконавчих механізмів, якщо вони передбачені системою.

Наступним етапом є підготовка даних до передачі. Система формує структурований пакет даних, який може включати значення параметрів, часові мітки, стан системи та додаткову службову інформацію. Дані перетворюються у формат, зручний для передачі через мережу, наприклад JSON.

Після цього виконується передача даних через модуль Wi-Fi. Мікроконтролер встановлює або використовує існуюче з’єднання з сервером або хмарним сервісом і надсилає сформований пакет даних. У випадку помилки передачі система може виконати повторну спробу або зберегти дані для подальшої відправки.

Паралельно система обробляє вхідні запити від користувача. Через вебдодаток або інший інтерфейс користувач може надсилати команди, змінювати

КПЗ_2026

параметри роботи системи або отримувати актуальну інформацію. Отримані команди аналізуються та виконуються мікроконтролером відповідно до заданої логіки.

Після передачі та обробки даних система повертається до початку основного циклу. Такий безперервний цикл забезпечує постійний моніторинг параметрів та актуальність інформації.

У випадку виникнення критичних помилок або завершення роботи система може виконати процедуру коректного завершення, що включає збереження даних, відключення модулів та переведення пристрою у безпечний стан.

Таким чином, алгоритм роботи програмного забезпечення забезпечує повний цикл функціонування системи - від збору даних до їх аналізу, передачі та взаємодії з користувачем. Його реалізація дозволяє створити ефективну, надійну та масштабовану систему моніторингу та управління виробничими процесами.

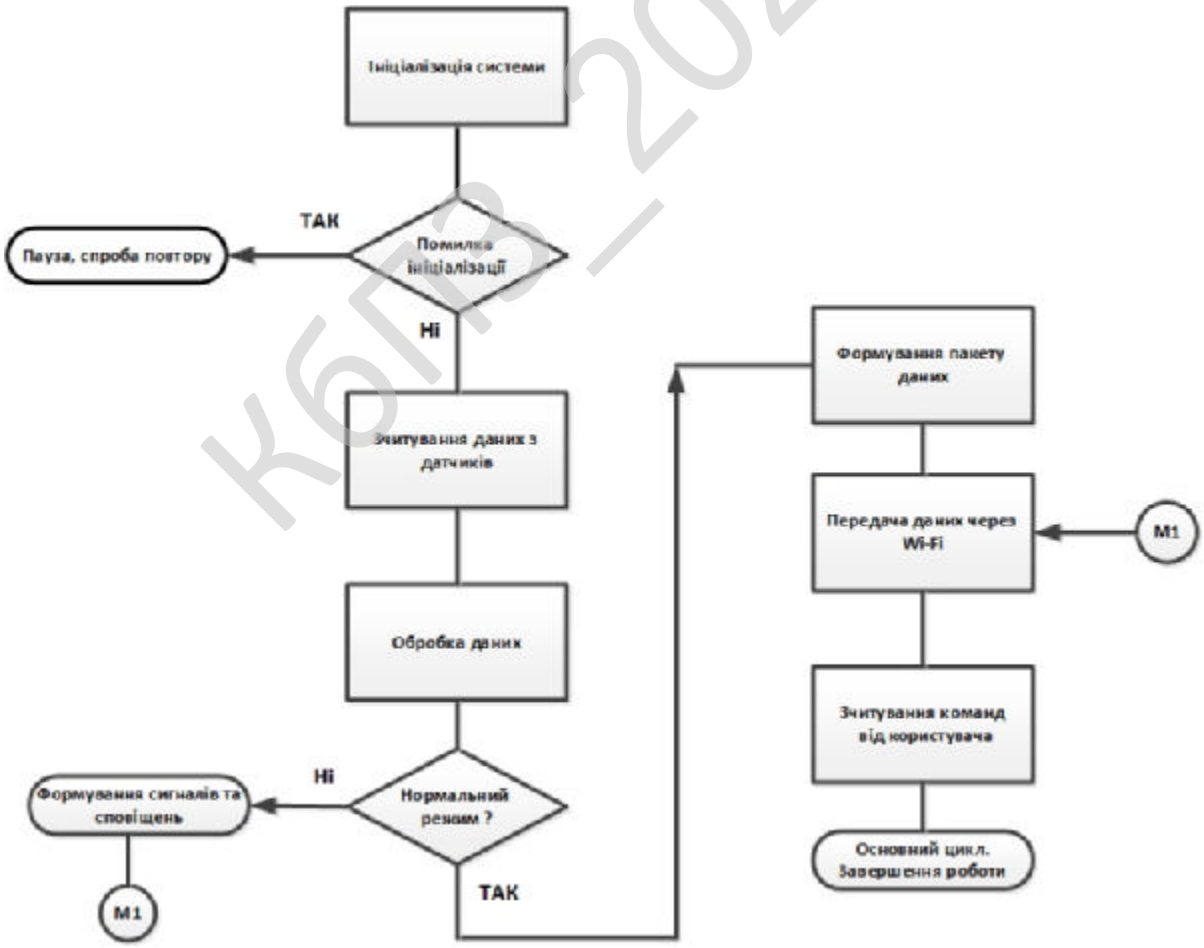


Рисунок 4.5 – Алгорим роботи автоматизованої системи моніторингу та управління

КБПЗ_2026

Алгоритм завершення роботи системи описує послідовність дій, які виконуються для коректного припинення функціонування програмного забезпечення автоматизованої системи моніторингу та управління виробничими процесами. Його основною метою є забезпечення збереження даних, уникнення втрати інформації та переведення всіх компонентів системи у безпечний стан.

Процес завершення роботи починається з отримання команди на завершення. Така команда може надходити від користувача через вебдодаток, автоматично при виникненні критичної помилки або у випадку планового вимкнення системи. Після отримання відповідного сигналу система переходить до виконання процедур завершення.

На першому етапі здійснюється збереження незавершених або тимчасових даних. Це можуть бути результати останніх вимірювань, буферизовані пакети даних або службова інформація, яка ще не була передана до серверу чи хмарного сховища. Збереження цих даних є важливим для забезпечення цілісності інформації та можливості її подальшого використання після повторного запуску системи.

Далі виконується зупинка роботи датчиків та допоміжних модулів. Мікроконтролер припиняє опитування сенсорів, вимикає або переводить у неактивний стан периферійні пристрої та завершує всі активні процеси збору даних. Це дозволяє уникнути некоректної роботи обладнання та зменшити енергоспоживання.

Наступним етапом є відключення мережевих з’єднань, зокрема модуля Wi-Fi. Система коректно завершує всі активні сеанси зв’язку, передає залишкові дані (якщо це можливо) та розриває з’єднання з сервером або хмарними сервісами. Це запобігає втраті даних та некоректному завершенню мережевих операцій.

Після цього система переходить у режим очікування або низького енергоспоживання. У цьому режимі всі основні функції припинені, але система може залишатися готовою до повторного запуску або отримання нової команди.

КБПЗ_2026

Такий підхід дозволяє оптимізувати використання ресурсів та забезпечити швидке відновлення роботи.

Завершальним етапом є повне завершення роботи системи. Усі процеси зупиняються, апаратні компоненти переводяться у безпечний стан, а система готова до вимкнення живлення або повторного запуску. Таким чином, алгоритм завершення роботи забезпечує коректне та безпечне завершення функціонування програмного забезпечення, зберігаючи цілісність даних і стабільність системи.



КбПЗ-2026

Рисунок 4.6 – Алгоритм завершення роботи

Алгоритм реєстрації

Алгоритм реєстрації в системі описує послідовність дій, необхідних для створення нового облікового запису користувача та його інтеграції в автоматизовану систему моніторингу та управління виробничими процесами. Основною метою даного алгоритму є забезпечення коректного введення, перевірки та збереження облікових даних, а також надання користувачу доступу до функціоналу системи.

Процес реєстрації починається з ініціації відповідної дії користувачем, який формує запит на створення нового облікового запису через вебдодаток або інший інтерфейс взаємодії. Після цього система переходить до етапу введення облікових даних, де користувач заповнює необхідні поля, такі як ім’я користувача, пароль, електронна адреса та інші параметри, що можуть бути потрібні для ідентифікації та авторизації.

Далі система виконує перевірку введених даних. На цьому етапі здійснюється контроль правильності заповнення полів, перевірка відповідності формату даних встановленим вимогам, а також перевірка унікальності користувача в базі даних. Крім того, можуть застосовуватися додаткові механізми безпеки, такі як перевірка складності пароля або підтвердження електронної пошти. Якщо виявляються помилки або невідповідності, система формує повідомлення про помилку та пропонує користувачу повторно ввести або виправити дані.

КПЗ_2026

У випадку успішного проходження перевірки система переходить до збереження облікових даних у базі даних. На цьому етапі створюється новий запис користувача, який містить усю необхідну інформацію для подальшої ідентифікації та взаємодії з системою. Збереження даних здійснюється з урахуванням вимог безпеки, зокрема може використовуватися шифрування паролів та інших конфіденційних даних.

Після збереження облікового запису виконується синхронізація з хмарними сервісами або серверною частиною системи. Це дозволяє забезпечити

доступність облікового запису з різних пристроїв та гарантує збереження даних у централізованому сховищі. На цьому ж етапі можуть бути налаштовані початкові параметри користувача, права доступу та інші індивідуальні налаштування.

Завершальним етапом алгоритму є підтвердження успішної реєстрації. Користувач отримує відповідне повідомлення про створення облікового запису та може бути автоматично авторизований у системі або перенаправлений до сторінки входу. У деяких випадках додатково може виконуватися підтвердження через електронну пошту або інші засоби автентифікації.

Таким чином, алгоритм реєстрації забезпечує надійний та послідовний процес створення облікового запису, включаючи введення, перевірку, збереження та синхронізацію даних, що дозволяє інтегрувати користувача в систему та надати йому доступ до її функціональних можливостей.

КбПЗ_2026

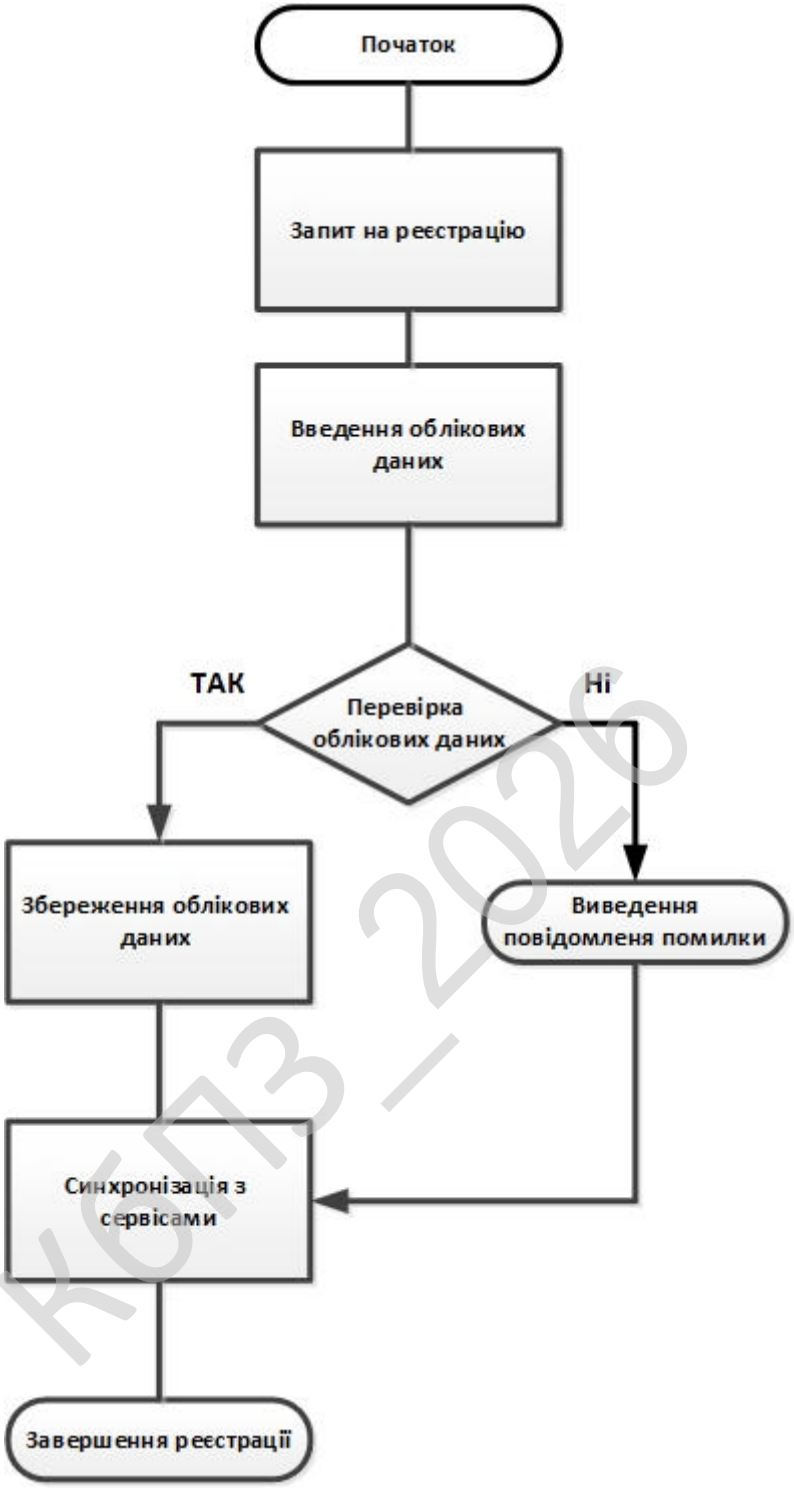


Рисунок 4.7 – Алгоритм реєстрації в системі

КПЗ – 2026

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Середовище розробки апаратного забезпечення і поєднання з AWS IoT

Після завершення підключення всіх компонентів та з'єднання мікроконтролера з персональним комп'ютером необхідно виконати завантаження програмного коду до плати Arduino. Даний процес було здійснено за допомогою офіційного інтегрованого середовища розробки Arduino IDE, що представлено на рисунку 5.1.

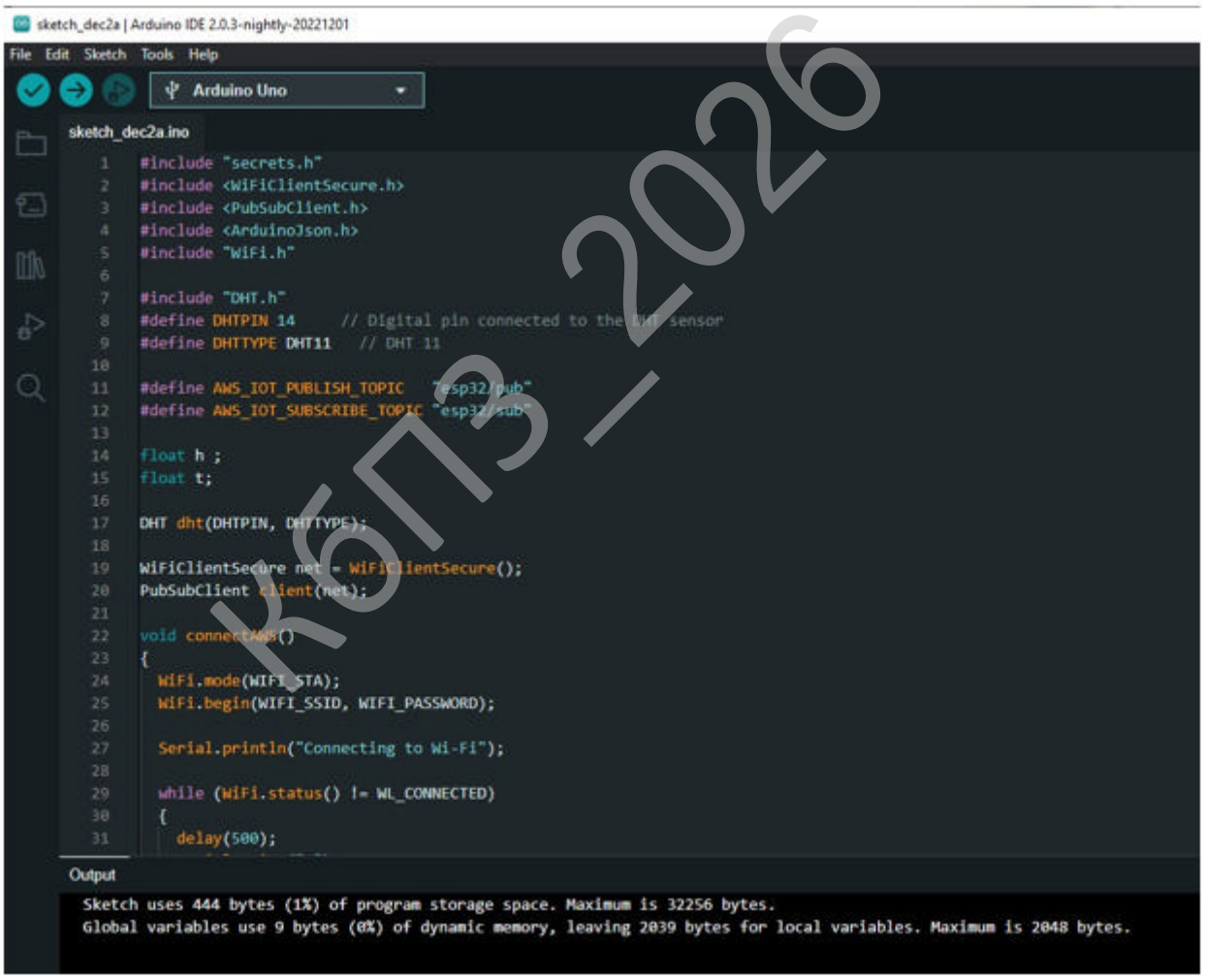


Рисунок 5.1 – Робоча програма у IDE Arduino

Для отримання даних із датчика температури та вологості була

КбПЗ_2026

використана бібліотека Adafruit DHT, яка була обрана відповідно до вимог розроблюваної системи моніторингу. Інші підключені модулі функціонували коректно та не потребували додаткового програмного налаштування.

Наступним етапом стало налаштування передавання та обробки повідомлень через MQTT-брокер. Для цього необхідно виконати конфігурацію пристроїв у сервісі IoT Core, дотримуючись рекомендацій та покрокових інструкцій, доступних на інформаційній панелі AWS IoT, що наведено на рисунку 5.2.

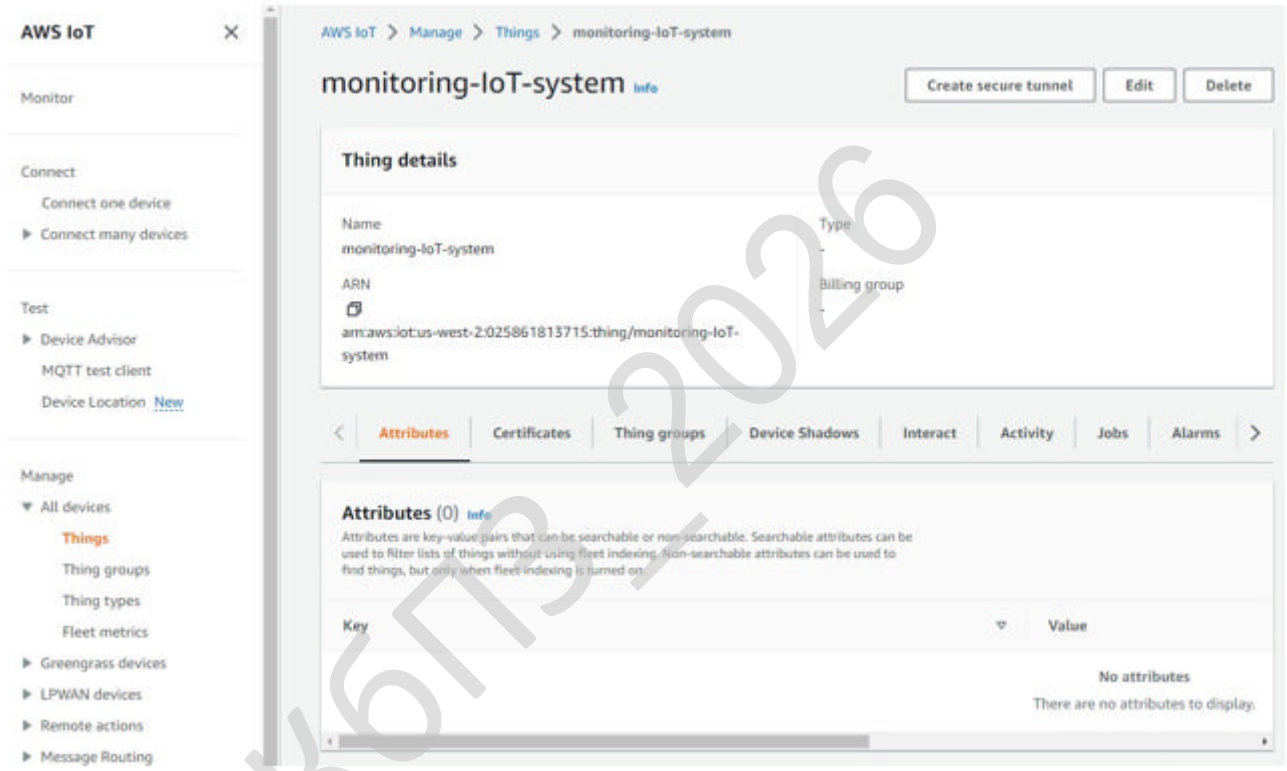


Рисунок 5.2 – Інформаційна панель AWS IoT Core

Для забезпечення взаємодії з пристроями необхідно створити нову політику доступу, перейшовши до розділу Secure → Policies → Create new policy. Створена політика повинна надавати відповідні дозволи для роботи з сервісами AWS IoT та містити посилання на ARN (Amazon Resource Name).

ARN використовується для унікальної ідентифікації ресурсів у середовищі AWS. Він застосовується у випадках, коли необхідно однозначно визначити певний ресурс, зокрема під час налаштування політик IAM, призначення тегів або

КБТЗ-2026

виконання API-запитів.

На рисунку 5.3 наведено приклад створення базової політики, яка забезпечує взаємодію між підключеними IoT-пристроями.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:*",
      "Resource": "*"
    }
  ]
}
```

Рисунок 5.3 – Політика взаємодії AWS IoT Core

Для підвищення рівня безпеки доцільно обмежити перелік доступних ресурсів і дозволених дій, наприклад шляхом зазначення Thing ARN у полі ресурсів політики доступу. Такий підхід дозволяє надавати права лише конкретному пристрою, мінімізуючи можливі ризики несанкціонованого доступу.

Наступним кроком є завантаження приватного та публічного ключів. Після цього, для застосування створеної політики, необхідно згенерувати сертифікат і прив’язати його як до політики доступу, так і до відповідного пристрою в сервісі AWS IoT. Сертифікат обов’язково повинен мати активний статус, оскільки лише в цьому випадку пристрій зможе встановити з’єднання з AWS IoT.

Отримані файли сертифікатів використовуються для безпечного підключення до MQTT-брокера AWS із застосуванням протоколу TLS. Поле для завантаження необхідних сертифікатів наведено на рисунку 5.4.

КОПЗ_2026

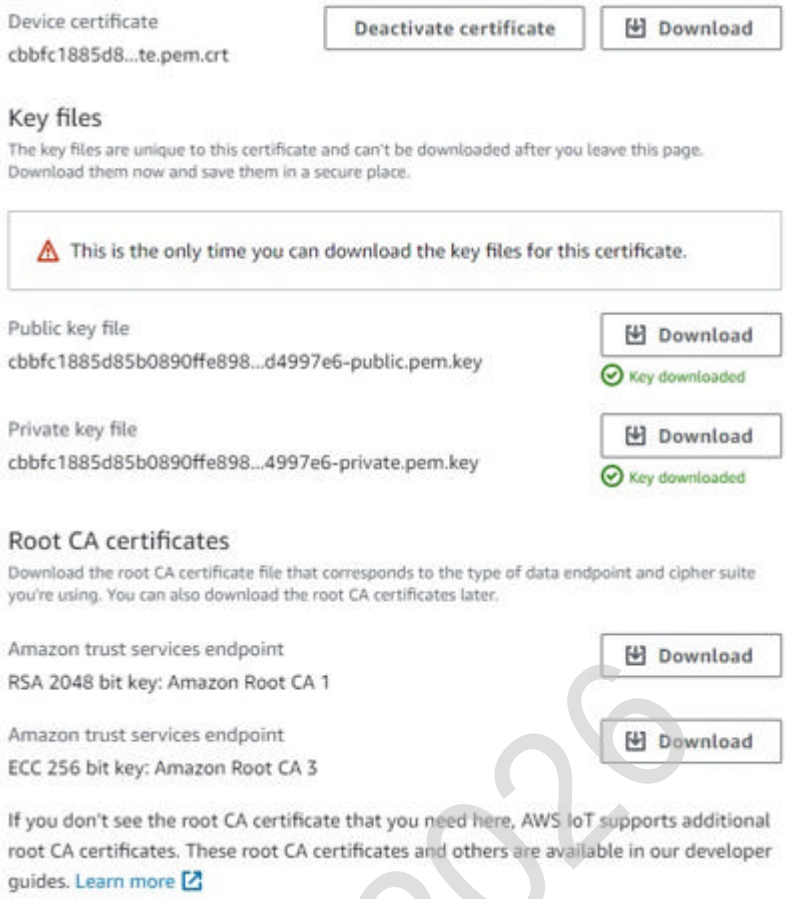


Рисунок 5.4 – Поле завантаження сертифікатів і ключів для AWS IoT

Наступним етапом стало налаштування підключення до мережі Wi-Fi та реалізація взаємодії з сервісами AWS. Оскільки підтримка модуля ESP8266 у середовищі Arduino та інтеграція AWS для Arduino є обмеженими, існувала ймовірність виникнення проблем із нестабільними або застарілими бібліотеками.

У зв'язку з цим було прийнято рішення здійснити ручне підключення до MQTT-брокера із використанням клієнтських сертифікатів.

Для реалізації безпечного з'єднання було інтегровано фрагмент коду з відкритої бібліотеки GitHub, призначеної для встановлення захищеного MQTT-з'єднання на базі ESP8266 з автентифікацією за сертифікатом клієнта. Сертифікати було завантажено у флешпам'ять модуля за допомогою плагіна Arduino ESP8266 Filesystem Uploader, доступного у репозиторіях GitHub [27].

Важливим елементом безпеки є жорстке збереження відбитка TLS-сертифіката сервера безпосередньо в пам'яті ESP-модуля. Такий підхід дозволяє запобігти

КПЗ_2026

атакам типу Man-in-the-Middle (MITM), оскільки пристрій перевіряє відповідність сертифіката під час кожного наступного підключення. Для ефективного захисту від подібних атак також рекомендується заборонити автоматичне підключення пристроїв до невідомих Wi-Fi мереж і використовувати лише перевірені точки доступу.

Наступним кроком стало налаштування та використання нереляційної бази даних DynamoDB (рис. 5.5). З метою підвищення надійності системи було застосовано вбудовані можливості сервісів Amazon, зокрема механізм створення тригерів для обробки отриманих повідомлень. Такі тригери забезпечують автоматичну взаємодію з іншими службами AWS, серед яких використовується DynamoDB для збереження отриманих даних.

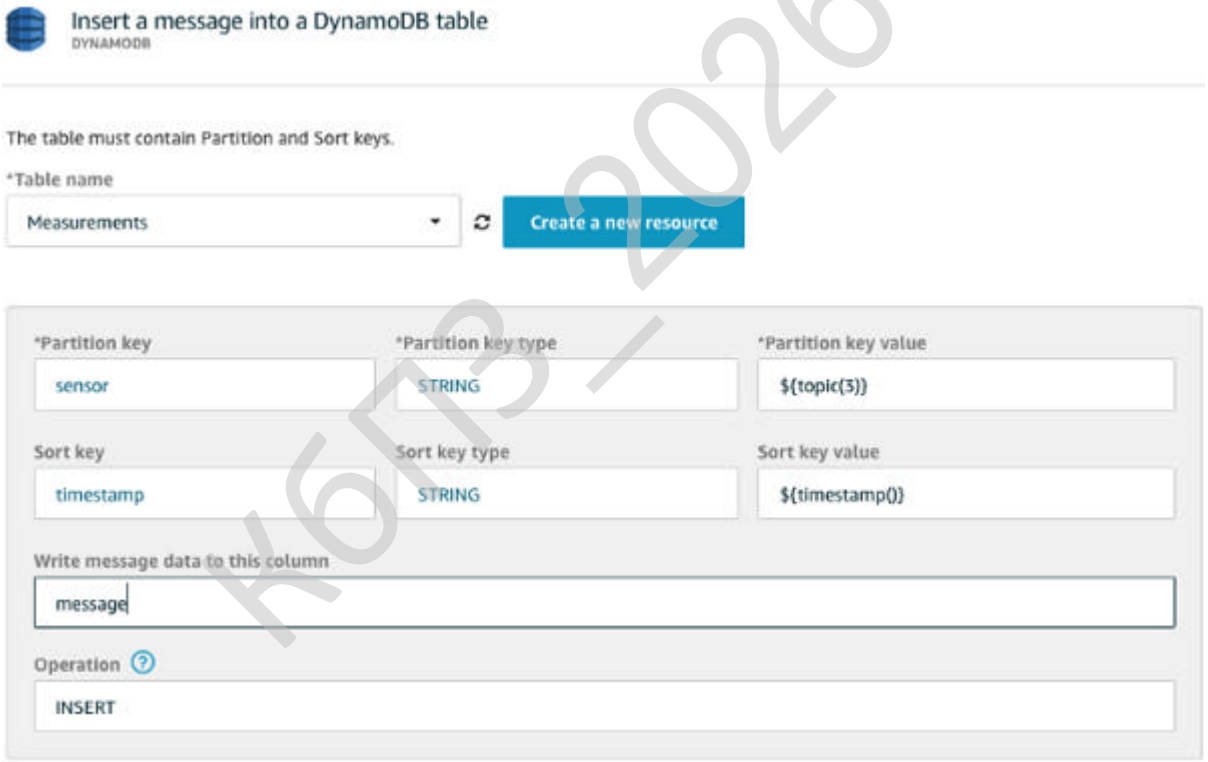


Рисунок 5.5 – Створення таблиці DynamoDB у AWS

На початковому етапі було виконано налаштування сервісу DynamoDB [28], зокрема створено нову таблицю з основним індексом (hash key) під назвою «сенсор» та додатковим сортувальним індексом - «мітка часу». Така структура

КБПЗ – 2026

дозволяє ефективно організувати зберігання та подальший пошук отриманих даних.

Наступним кроком стало налаштування правила обробки повідомлень, що є окремим і досить відповідальним етапом, оскільки більшість помилок системи фіксуються автоматично, а їх діагностика значною мірою залежить від досвіду розробника. Завдяки використанню бібліотек AWS вдалося реалізувати спроєктовану архітектуру системи моніторингу.

Amazon AWS надає SQL-подібну мову запитів для аналізу JSON-даних, отриманих у вхідних MQTT-повідомленнях. У даному проєкті було використано бета-версію SQL-діалекту AWS IoT, оскільки окремі функції, зокрема topic(3), що дозволяє отримувати третій сегмент URL-адреси MQTT-теми (наприклад, назву датчика), були відсутні або працювали нестабільно у попередніх версіях.

Amazon DynamoDB є повністю керованою NoSQL базою даних, яка підтримує моделі зберігання типу «ключ–значення» та документно-орієнтовані структури. Сервіс входить до складу платформи Amazon Web Services і базується на концепціях системи Dynamo, хоча має власну реалізацію та розширені можливості масштабування.

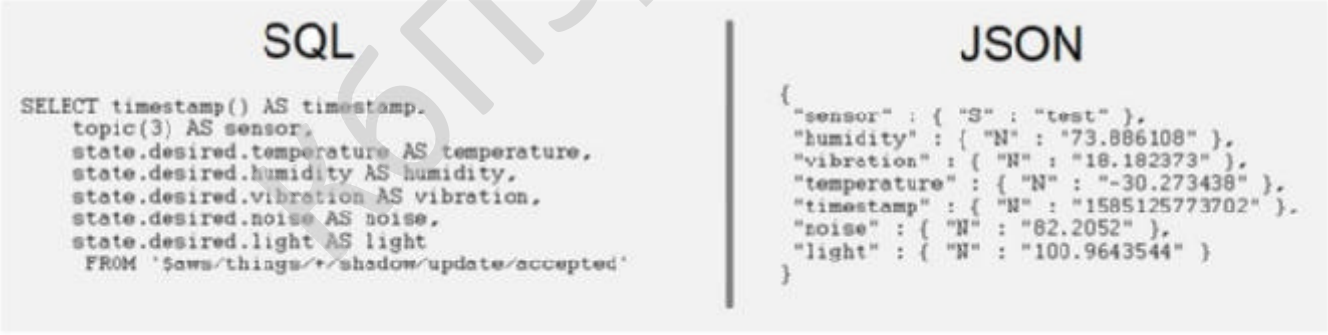


Рисунок 5.6 – Опис SQL-запиту та залежностей JSON

Середовище та розробка вебдодатку на Flask та Python

Для реалізації користувацького інтерфейсу було обрано вебфреймворк Flask, призначений для розробки вебзастосунків мовою Python. На початковому

БПЗ_2026

етапі створення серверної частини було реалізовано окремий клас для взаємодії з базою даних DynamoDB [28], використовуючи бібліотеку Boto3 (рис. 3.13).

Однією з ключових переваг Boto3 є автоматичний пошук облікових даних AWS, створених за допомогою AWS CLI, що усуває необхідність ручного введення токенів доступу у програмному коді. Для цього достатньо встановити інтерфейс командного рядка Amazon, виконати команду aws configure та вказати ідентифікатор доступу, секретний ключ і регіон використання сервісів.

AWS SDK для Python (Boto3) застосовується для створення, налаштування та керування різними сервісами AWS, зокрема AWS IoT і DynamoDB. Бібліотека надає як об’єктно-орієнтований інтерфейс програмування, так і низькорівневий доступ до функцій хмарних сервісів. Відповідно до офіційної документації Boto3 було реалізовано використання ресурсів DynamoDB.ServiceResource та DynamoDB.Table, що дозволяє створювати таблиці, додавати нові записи, змінювати існуючі дані, отримувати елементи та виконувати запити або фільтрацію інформації.

Крім того, Boto3 містить механізм так званих *очікувачів* (waiters), які автоматично перевіряють зміну стану ресурсів AWS. Наприклад, за їх допомогою можна очікувати запуск екземпляра Amazon EC2 або завершення створення таблиці DynamoDB до моменту, коли вона стане доступною для використання. Такі механізми доступні як для клієнтських, так і для ресурсних API бібліотеки Boto3.

КБПЗ-2026

```
class DB:
    # Query db for the last 20 measures of a sensor
    def retrieve_values(sensor, metric=None):
        table = dbm.Table('Measurements')
        response = table.query(
            KeyConditionExpression=keyword('sensor').eq(sensor),
            ScanIndexForward=False,
            Limit=2)
        items = []

        for i in response[u'Items']:
            # Transforming timestamp to readable hour and minutes
            # in one line
            i['message']['timestamp'] = datetime.fromtimestamp(int(i['timestamp'][:-3])).strftime('%H:%M')

            if _MetricsDict is not None:
                #if metric is set return only that value
                i = i['message'][metric]
            else:
                # otherwise return all metrics for this sensor
                i = i['message']

            items.append(i)
        return ItemsView
```

Рисунок 5.7 – Таблиця Boto3 для взаємодії з DynamoDB

Бібліотека Boto3 підтримує два основні рівні програмного інтерфейсу. Клієнтський (низькорівневий) API забезпечує пряме відображення базових HTTP-операцій сервісів AWS. Натомість ресурсний API приховує мережеві виклики та надає зручну об’єктну модель доступу до ресурсів і їх атрибутів, що дозволяє виконувати необхідні дії через відповідні об’єкти та колекції.

Після впорядкування ключів з’являється можливість отримувати останні повідомлення шляхом виконання запиту до таблиці за первинним ключем із використанням параметра:

KeyConditionExpression = Key('sensor').eq(sensor).

Параметр ScanIndexForward = False застосовується для зміни порядку сортування записів, а значення Limit = 20 дозволяє отримати дані приблизно за останню годину роботи системи, враховуючи, що кожен датчик надсилає повідомлення з інтервалом у три хвилини.

Завершальним етапом розробки автоматизованої системи моніторингу стало використання вебфреймворку Flask для Python. Даний фреймворк забезпечує обробку HTTP та AJAX-запитів, а також реалізацію серверної та клієнтської

КПЗ–2026

частин вебзастосунку. Через Flask здійснюється передавання та отримання даних між модулем ESP-01 і вебдодатком. Обробка повідомлень виконується у середовищі AWS IoT Core із застосуванням протоколу MQTT [29]. Таким чином, датчики передають інформацію, яка відображається у застосунку Flask, а сам вебдодаток може надсилати керуючі повідомлення назад до середовища IoT [30] (рис. 5.8).

На початковому етапі було реалізовано сторінку авторизації користувачів для обмеження доступу до інформації системи. Для цього використовується стандартна бібліотека Flask, яка представляє собою клас Python, що виступає основою створення екземплярів вебзастосунків [31] (рис. 5.9).

Для структуризації проєкту застосовано механізм Flask Blueprints, який дозволяє інкапсулювати окремі компоненти застосунку, такі як представлення, шаблони та статичні ресурси. Кожен Blueprint функціонує подібно до окремого модуля програми Flask і може містити власні маршрути, шаблони та файли ресурсів. Під час створення Blueprint реєструються необхідні операції, які виконуються після його підключення до основного застосунку. Наприклад, прив’язка представлення до маршруту зберігається та активується після реєстрації відповідного Blueprint у головному додатку.

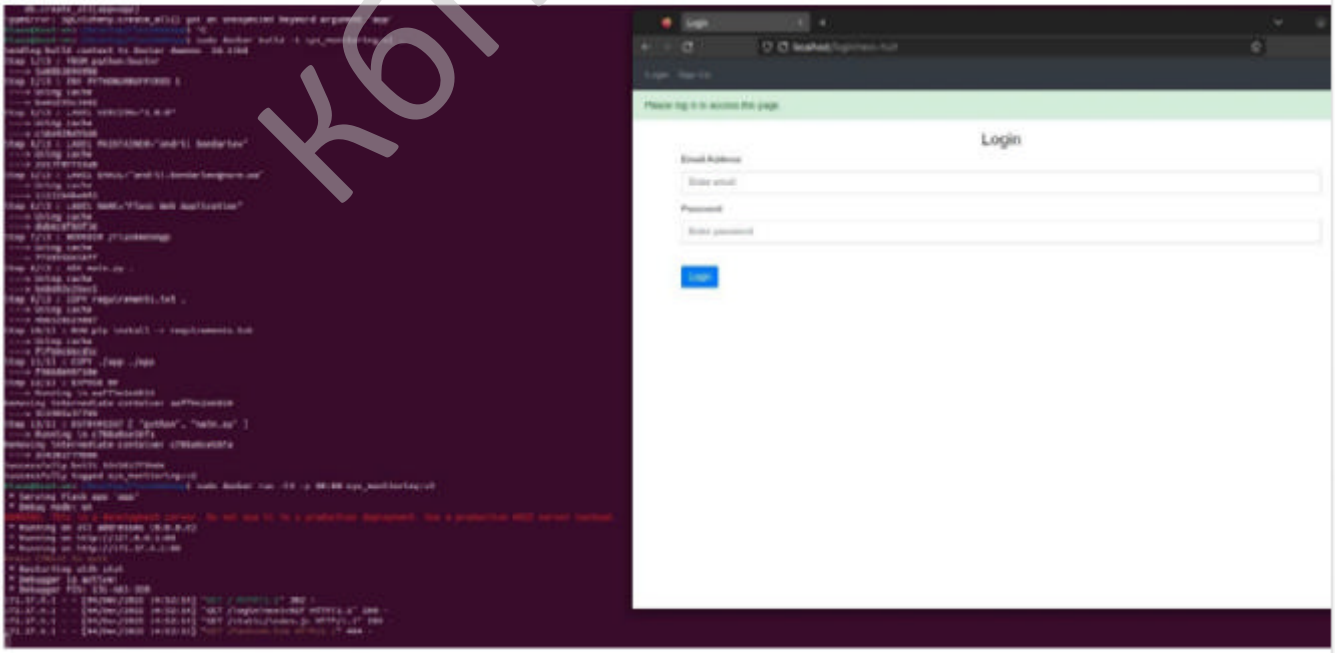


Рисунок 5.8 – Демонстрація роботи сторінки авторизації

КБПЗ-2026

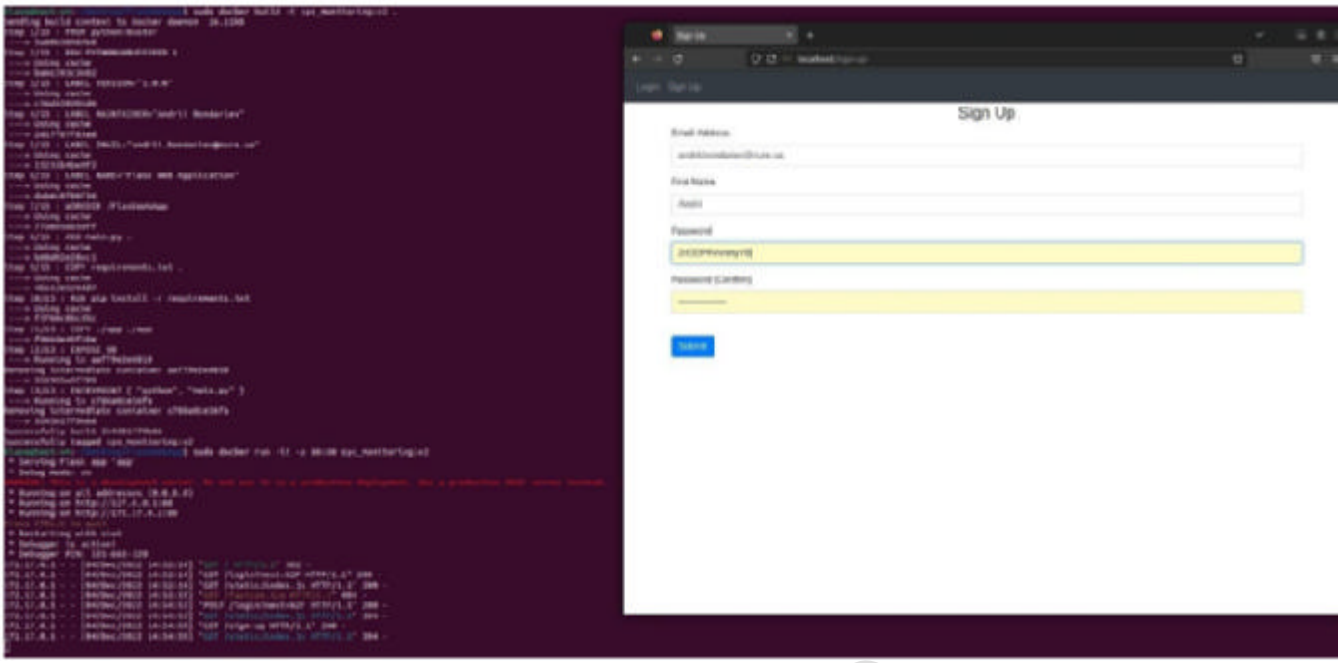


Рисунок 5.9 – Демонстрація роботи сторінки реєстрації

Надання користувачам можливості авторизованого доступу до вебзастосунку є однією з ключових функцій автоматизованої системи моніторингу підприємства. З цією метою в застосунок Flask було інтегровано механізм автентифікації користувачів із використанням пакета **Flask-Login**, який забезпечує керування сесіями, перевірку доступу та контроль авторизованих підключень. Окрім цього, у процесі розробки було застосовано додаткові технології, спрямовані на оптимізацію структури та організації роботи вебзастосунку. Зокрема, використано контейнеризацію за допомогою сервісу Docker та систему контролю версій Git. Дані інструменти виконують допоміжну роль, спрощуючи процес розроблення, тестування та розгортання програмного забезпечення, а також підвищуючи ефективність створюваного продукту. Основна розробка виконувалась у середовищі редактора коду Visual Studio Code (рис. 5.10).

Docker являє собою платформу контейнеризації, що дозволяє упаковувати застосунок разом із необхідним програмним середовищем і залежностями в ізольований контейнер. Такий підхід забезпечує незалежність роботи програми від операційної системи та інших встановлених застосунків, зменшує

КПІЗ_2026

навантаження на сервер і сприяє підвищенню продуктивності вебзастосунку системи моніторингу.

Git, у свою чергу, є розподіленою системою контролю версій, яка дозволяє відстежувати зміни у програмному коді, керувати історією розробки та забезпечує можливість спільної роботи кількох розробників над одним проєктом.

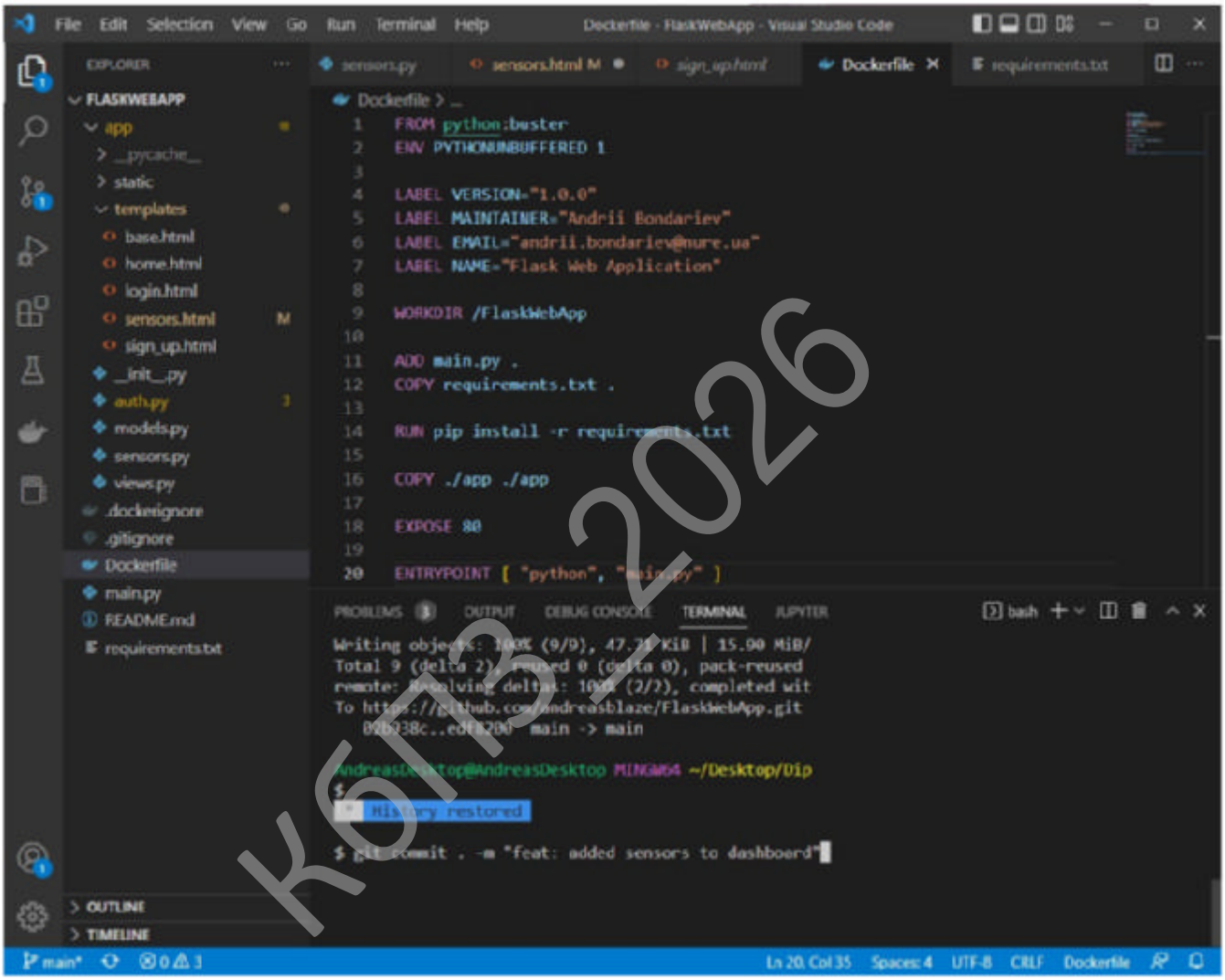


Рисунок 5.10 – Демонстрація процесу розробки у Visual Studio Code

Вебзастосунок Flask запускається у локальному Docker-контейнері з використанням середовища AWS IoT Core для обробки даних, що надходять із сенсорів та апаратно-програмного комплексу, побудованого на платформі Arduino. На рівні серверної логіки Flask було реалізовано обробники даних, які приймають інформацію із сенсорів, аналізують отримані показники та визначають їх

КБПЗ-2026

відповідність встановленим нормативним значенням. У результаті система відображає стан датчиків і підтверджує, що їхні параметри перебувають у допустимих межах.

Для створення користувацького інтерфейсу застосовано шаблони вебверстки на основі HTML із використанням бібліотеки jQuery та CSS-фреймворку Bulma, який надає готові інтерфейсні компоненти та дозволяє швидко формувати адаптивні вебінтерфейси. Підсумковий вигляд розробленого вебзастосунку наведено на рисунку 5.11.

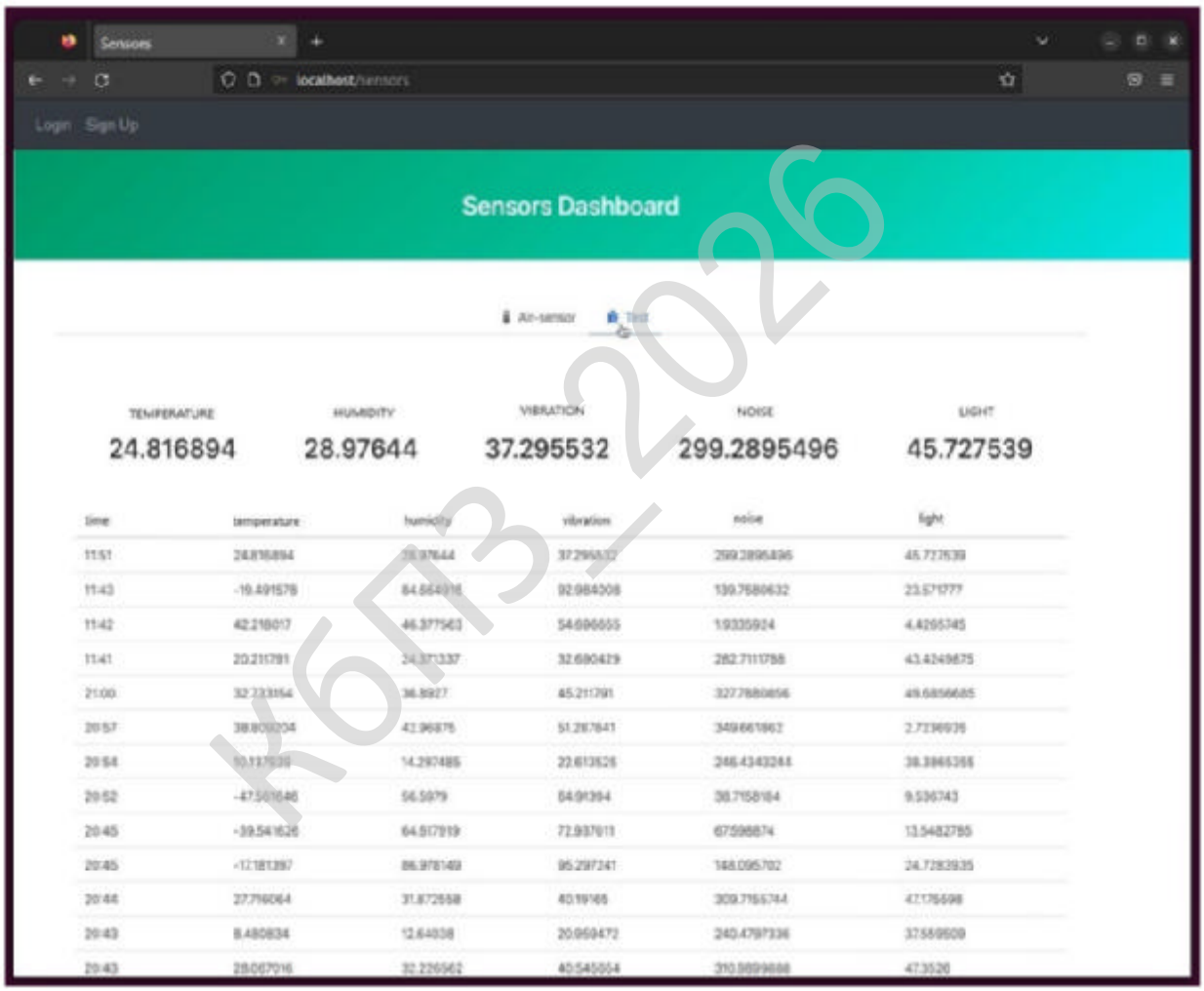


Рисунок 5.11 – Результат розгортки Flask додатку

Таким чином було розроблео загальну архітектуру системи моніторингу, а також наведено опис використаних технологій і принципів їх взаємодії. Проведено підключення всіх апаратних компонентів до платформи Arduino UNO,

КОПЗ_2026

у результаті чого реалізовано відповідний апаратно-програмний комплекс системи.

Крім того, виконано налаштування середовища розробки апаратної частини та забезпечено її інтеграцію з хмарною платформою AWS IoT. Окрему увагу приділено створенню програмного забезпечення вебзастосунку на основі фреймворку Flask мовою програмування Python.

У підсумку було продемонстровано працездатність розробленої автоматизованої системи моніторингу та керування виробничими процесами підприємства, що підтверджує ефективність запропонованого рішення.

КБПЗ_2026

6 ОСНОВНІ ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено аналіз актуальності створення систем моніторингу виробничих процесів на підприємствах, а також досліджено сучасні технології, що застосовуються у даній сфері. Виконано аналіз технічного завдання, здійснено вибір і обґрунтування програмних та апаратних засобів реалізації системи. Описано структуру, функціональні можливості та принцип роботи розроблюваної системи. Проведено огляд існуючих систем моніторингу, зокрема рішень на основі сенсорних технологій, призначених для збору даних про критично важливі параметри виробничих процесів. Також досліджено типи датчиків, що можуть застосовуватися на підприємствах, їх значущість та актуальність збору відповідних показників, а також розглянуто мікроконтролерні платформи як основу апаратного забезпечення.

На основі отриманих результатів аналізу було обрано та спроєктовано структуру автоматизованої системи моніторингу, розроблено схеми взаємодії її компонентів і технологій обробки інформації. Побудовано інфраструктуру збору, обробки та зберігання даних із використанням хмарної платформи Amazon Web Services, мови програмування Python та вебфреймворку Flask. Продемонстровано роботу окремих модулів системи та структуру проєкту, розгорнутого у середовищі AWS IoT.

У результаті проєктування було визначено середовища розробки, сформовано алгоритм функціонування програмного забезпечення, виконано його налаштування та підбір необхідних бібліотек. Реалізовано апаратну частину системи на базі платформи Arduino UNO, здійснено її інтеграцію з хмарним середовищем AWS IoT, а також розроблено вебзастосунок мовою Python із використанням фреймворку Flask. Забезпечено взаємодію апаратно-програмного комплексу з інфраструктурою Інтернету речей AWS за допомогою брокера

КПЗ – 2026

повідомлень MQTT із застосуванням мови Arduino C++. Обробку та збереження даних реалізовано із використанням нереляційної бази даних AWS DynamoDB.

На основі впровадження розробленої системи на виробничому об’єкті, зокрема на контрольно-пропускному пункті підприємства, було також підготовлено документацію з охорони праці.

У межах кваліфікаційної роботи продемонстровано функціонування:

– апаратно-програмного комплексу датчиків і Wi-Fi модуля на базі Arduino UNO;

– інтеграції компонентів Arduino із сервісами AWS IoT за допомогою MQTT-брокера повідомлень;

– нереляційної бази даних AWS DynamoDB;

– програмного забезпечення у вигляді вебзастосунку для обробки даних сенсорів, реалізованого мовою Python із використанням фреймворку Flask.

КбПЗ_2026

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.

2. Yasin, J.N., Mohamed, S.A.S., Haghbayan, MH. et al. Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. Int. j. inf. tecnol. 13, 97–107 (2021). https://doi.org/10.1007/s41870-020-00513-w.

3. Sabry F. Self Driving Car: Solving Full Self-driving Need Solving Real-world artificial Intelligence. One Billion Knowledgeable. 2022 . 293 p.

4. Shapiro D. G. Three Anecdotes from the DARPA Autonomous Land Vehicle Project. AI Magazine, 2008. 29(2), 40. https://doi.org/10.1609/aimag.v29i2.2108.

5. Young R. Critical Analysis of Prototype Autonomous Vehicle Crash Rates Six Scientific Studies from 2015-2018 SAE International. 2021. 252 p.

6. Рудик А. В. Наукові основи та принципи побудови приладової системи вимірювання прискорення мобільного робота : дис. ... д-ра техн. наук : 05.11.01– Прилади та методи вимірювання механічних величин. Київ, 2018. – 460 с.

7. Поліщук М. М., Ткач М.М. Робототехнічні системи: проектування і моделювання: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.

8. Мигаль В. Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.

9. Staron M. Automotive Software Architectures: An Introduction. Second Edition Springer International Publishing A&G. 2022. 274 p.

10. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397.

11. Wevolver 2020 Autonomous Vehicle Technology Report. URL:https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehicechnologyReportpdf/ (accessed 2023 June 1).

КПІ – 2026

12. React.js [Електронний ресурс] – Режим доступу до ресурсу: https://https://uk.reactjs.org/

13. AngularJS MVC [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm.

14. Vue.js [Електронний ресурс] – Режим доступу до ресурсу: https://vuejs.org/

15. A Look Inside ADAS Modules. URL: https://amkor.com/semiconductor-story/alook.

16. AV and ADAS Sensors. URL https://community.sw.siemens.com/s/article/AVand- ADAS-Sensors (accessed 2023 June 1).

17. Гуржій А. М. Основи автоматики та робототехніки: Навчальний посібник/ А.М. Гуржій, А. Т. Нельга, В. М. Співак, О. С. Ітякін:–Дніпро:«Гарант СВ», 2021.- 243с.

18. Lim, B.S. et al. Autonomous Vehicle Ultrasonic Sensor Vulnerability and Impact Assessment. In Proceedings of the IEEE World Forum on Internet of Things (WFIoT), Singapore, 5–8 February 2018; pp. 231–236.

19. Саліхов М. М. Arduino – перспективний інструмент протитопування у робототехніці // Концептуальні шляхи розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції м. Львів, 9-10 червня 2023 року.– Львів: Львівський науковий форум, 2023. – с. 73-79.

20. Bräunl T. Embedded Robotics: From Mobile Robots to Autonomous Vehicles with RaspberryPi and Arduino Springer Nature, 2022. 519 p.

21. Навігаційні системи [Електронний ресурс] : навч. посіб. для студ. Спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології» / С.Л. Лакоза; КПІ ім. Ігоря Сікорського, 2021. — 80 с.

22. Imad M. et al. Navigation system for autonomous vehicle: A survey. JCSTS., vol. 2, no. 2, pp. 20–35, 2020.

КОПЗ 2026

23. Ковалець І. В. та ін. Технологія планування траєкторій руху мобільних об’єктів з урахуванням перешкод на складній місцевості. Енергетика і автоматика. - 2017. - № 1. - С. 110-122.

24. HTML [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/uk/docs/Web/HTML.

25. Aguilo I. et al. Artificial Intelligence Research and Development Hardcover by IOS Press. 2003. 500 p.

26. Гребенюк Б. А. Розробка підсистеми управлення інтелектуальним роботом / Б. А. Гребенюк // «Automation and Development of Electronic Devices» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National Universitety of Radio Elektronics [electronic edition], 2023. – Part 1. –336p. P. 263-269.

27. MongoDB [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/MongoDB.

28. Becker M. et al Obstacle avoidance procedure for mobile robots. ABCM Symposium series in Mechatronics. Vol. 2, 2006. P. 250-257.

29. CORS, XSS [Електронний ресурс] – Режим доступу до ресурсу: https://dev.to/maleta/cors-xss-and-csrf-with-examples-in-10-minutes-35k3.

30. AJAX [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/AJAX.

31. Fetch API [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Web/API/Fetch_API.

32. Звенігородський О.С. Інтелектуальна система планування тактики руху автономного робота в квазістаціонарному середовищі: Дис... канд. техн. наук: 05.13.23. — Д., 2002. — 127с.

33. Адамів, О.П. Моделі та інтелектуальні засоби адаптивного керування автономним мобільним роботом. Дис. канд.техн. наук. Одеса, 2007. 124 с.

КБПЗ–2026

34. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.

35. npm [Електронний ресурс] – Режим доступу до ресурсу: https://www.npmjs.com/

36. Install MongoDB [Електронний ресурс] – Режим доступу до ресурсу: https://docs.mongodb.com/manual/administration/install-on-linux/

37. Node.js [Електронний ресурс] – Режим доступу до ресурсу: https://www.digitalocean.com/community/tutorials/node-js-ubuntu-18-04-ua.

38. Linux [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Linux.

39. npm-audit [Електронний ресурс] – Режим доступу до ресурсу: https://docs.npmjs.com/cli/audit.

40. Kojima T. et al. Intelligent Technology for More Advanced Autonomous Driving. 2018. Hitachi Review Vol. 67, No. 1. p. 58–63.

41. SQL [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/SQL.

42. MongoDB Compass [Електронний ресурс] – Режим доступу до ресурсу: https://www.mongodb.com/products/compass.

43. Postman [Електронний ресурс] – Режим доступу до ресурсу: https://www.postman.com/

44. Саліхов М. М. Від автоматизованих до автономних транспортних засобів // Наукові досягнення та інновації: шлях до успіху. X Всеукраїнська мультидисциплінарна науково-практична Інтернет-конференція, 31 травня 2023, Україна, Київ : зб. матеріалів — Електрон. дан. — Київ : Ярочéнко Я. В., 2023. — с. 136-142.

45. SPA (Single-page application) [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Glossary/SPA.

КПІ_2026

46. Саліхов М.М. Самокеровані автомобілі та системи їх навігації// Практичні та теоретичні питання розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції: м. Львів, 19-20 червня 2023 року. – Львів: Львівський науковий форум, 2023. – С.53-59.

47. SPA (Single-page application) [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Glossary/SPA.

48. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their

49. Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397.

50.https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehicec hnologyReportpdf/ (accessed 2024 June 1).

КБПЗ_2026