

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

РОЗРОБЛЕННЯ КОМПОНЕНТІВ SCADA СИСТЕМИ ДЛЯ ГНУЧКОГО
ВИРОБНИЦТВА НА БАЗІ AWS

(тема)

Виконав студент 2 курсу, групи КТРСм-22-2
Большаков Антон Андрійович
(прізвище, ім'я, по батькові)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва напрямку)

Тип програми Освітньо-професійна
Освітня програма Комп'ютеризовані та
робототехнічні системи
(назва)

проф. каф. КІТАР Цимбал О.М.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАР

(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2023 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

12.01.2024

АБ Большаков А.А

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет	АКТ
Кафедра	КІТАР
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми	Освітньо-професійна
Освітня програма	Комп'ютеризовані та робототехнічні системи (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри _____
(підпис)

«__» _____ 2024р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____
Большакову Антону Андрійовичу
(прізвище, ім'я, по батькові)Тема роботи: Розроблення компонентів SCADA системи для гнучкого виробництва на базі AWS

Затверджена наказом по університету від _____ №1288Ст від 03.11.2023

2. Термін подання студентом роботи до екзаменаційної комісії 19.01.2024р3. Вихідні дані до роботи: 3.1 Хмарний сервіс Amazon Web Services;3.2 Середовище розробки Arduino IDE; 3.3 Мова програмування C++;

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Аналіз існуючих технологій розробки SCADA-систем; 4.3 озробка архітектури SCADA-системи гнучкого виробництва на основі AWS; 4.4 Розробка програмного забезпечення компонентів SCADA системи для гнучкого виробництва на базі AWS; 4.5 Дослідження та аналіз отриманих результатів; 4.6 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Графічний демонстраційний матеріал в форматі PowerPoint(*.ppt) формату А4-11 сторінок.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз існуючих технологій розробки SCADA-систем	01.09.2023-16.09.2023	виконано
2	Аналіз технічного завдання	16.09.2023-19.09.2023	виконано
3	Аналіз використання хмарних-сервісів в SCADA-системах	20.09.2023-25.09.2023	виконано
4	Розробка архітектури системи	26.09.2023-08.11.2023	виконано
5	Розробка програмного забезпечення SCADA системи	09.11.2023-25.11.2023	виконано
6	Розробка макету	26.11.2023-06.12.2023	виконано
7	Оформлення пояснювальної записки	07.12.2023-31.12.2023	виконано
8	Подання роботи на перевірку Інтернет-сервісом	19.01.2023	виконано
9	Подання роботи на рецензію	20.01.2023	виконано
10	Подання роботи на підпис зав. кафедри	21.01.2023	виконано
11	Подання кваліфікаційної роботи в ЕК	22.01.2023	виконано

Дата видачі завдання 1 вересня 2023р.

Студент _____
(підпис)

Большаков А.А.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

проф. Цимбал О.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 92 с., 8 табл., 39 рис., 3 дод., 30 джерел.

SCADA-СИСТЕМИ, AMAZON WEB SERVICES, MQTT, CLOUD SERVICES, INTERNET OF THINGS, ГНУЧКЕ ВИРОБНИЦТВО.

Мета роботи – підвищення ефективності MQTT протоколу для передачі даних між компонентами SCADA системи для гнучкого виробництва.

Об'єкт роботи – процес віддаленого моніторингу та візуалізації даних у SCADA системі.

Предмет роботи – модернізація методів передачі даних між компонентами SCADA системи на базі хмарних технологій Amazon Web Services.

В даній магістерській кваліфікаційна роботі приведено аналіз існуючих технологій розробки SCADA-систем. На базі проведеного аналізу було розроблено компоненти SCADA системи для гнучкого виробництва на базі AWS.

Був створений макет, який забезпечує взаємодію апаратних засобів та хмарних сервісів. Була перевірена функціональність та працеспроможність макету. Отримані результати роботи були досліджені та описані. Після перевірки функціональності було отримано та візуалізовано дані на хмарній частині, які були зібрані за допомогою апаратних засобів.

Результати кваліфікаційної роботи описані в статті, яка знаходиться в гугл академії.

ABSTRACT

Explanatory note: 92 pages, 8 tables, 39 figures, 3 app, 30 sources.

SCADA-SYSTEMS, AMAZON WEB SERVICES, MQTT, CLOUD SERVICES, INTERNET OF THINGS, FLEXIBLE MANUFACTURE.

The purpose of the work is to develop a secure MQTT protocol for data transmission between components of the SCADA system for flexible production.

The object of development is the process of remote monitoring and visualization of data in the SCADA system.

The subject of the development is the modernization of data transfer methods between SCADA system components based on Amazon Web Services cloud technologies.

This master's qualification thesis provides an analysis of existing SCADA systems development technologies. Based on the analysis, SCADA system components were developed for flexible AWS-based production.

A layout was created that ensures the interaction of hardware and cloud services. The functionality and workability of the layout was checked. The obtained work results were investigated and described. After the functionality was tested, the cloud-based data collected by the hardware was retrieved and visualized.

The results of the qualification work are described in the article, which is located in Google Academy.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Аналіз існуючих технологій розробки SCADA-систем.....	12
1.1 Принципи організації SCADA систем	12
1.2 Програмні технології розробки SCADA систем.....	19
1.3 Використання хмарних-сервісів в SCADA системах.....	24
1.4 Постановка завдання дослідження.....	30
2 Розробка архітектури SCADA -системи гнучкого виробництва на основі AWS	32
2.1 Порівняльний аналіз можливостей хмарних сервісів	32
2.2 Архітектура SCADA-системи гнучкого виробництва	35
2.3 Розробка алгоритмічного та програмного забезпечення SCADA-системи гнучкого виробництва.....	44
2.4 Аналіз та вибір модулів моніторингу і керування гнучкого виробництва	47
2.5 Висновки до другого розділу	54
3 Розробка програмного забезпечення компонентів SCADA системи для гнучкого виробництва на базі AWS	55
3.1 Удосконалення методу передачі інформації на базі протоколу MQTT	55
3.2 Розробка програмного забезпечення SCADA-системи.....	59
3.3 Розробка макета модуля для збирання технологічних параметрів.....	63
3.4 Висновки до третього розділу	66
4 Дослідження та аналіз отриманих результатів	67
4.1 Постановка завдань експерименту	67
4.2 Проведення експерименту	67
4.3 Аналіз отриманих результатів	71

4.4 Охорона праці.....	74
4.5 Висновки до четвертого розділу.....	75
Висновки	77
Перелік джерел посилань	79
Додаток А Апробація результатів наукових досліджень.....	83
Додаток Б Основний файл прошивки	92
Додаток В Демонстраційний матеріал у вигляді презентації	102

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АРМ – автоматизовані робочі місця;

АСКОЕ – автоматизована система комерційного обліку електричної енергії;

АСУТП – автоматизовані системи управління технологічними процесами;

AWS – Amazon Web Services;

MQTT – Message Queuing Telemetry Transport;

SCADA – Supervisory Control And Data Acquisition;

ERP – enterprise resource planning;

HMI – Human Machine Interface;

Industry 4.0 – четверта промислова революція;

IoT – Internet of Things;

OPC – Open Platform Communications.

ВСТУП

На сьогоднішній день аналіз існуючих технологій розробки SCADA-систем на сучасному етапі є ключовим етапом у визначенні оптимальних рішень для гнучкого виробництва. Системи, які забезпечують контроль і автоматизацію виробничих процесів, вимагають уважного розгляду та адаптації до сучасних вимог та можливостей. Розробка компонентів SCADA-систем для гнучкого виробництва на основі Amazon Web Services (AWS) відкриває перед індустрією нові перспективи. У світлі зростаючого значення гнучкості та швидкозмінюваних виробничих умов, AWS надає ефективні інструменти для створення сучасних систем контролю, що відповідають сучасним стандартам безпеки та надійності.

Можливість швидкого масштабування та адаптації SCADA-систем для гнучкого виробництва на базі AWS до змін виробничих потреб, надійність та стійкість цих систем у реальному часі визначають актуальність даної роботи.

Такий підхід сприяє підвищенню продуктивності, забезпечуючи оптимальне використання ресурсів та високу ефективність обміну даними. Інтеграція з іншими системами стає більш зручною, а забезпечення безпеки та конфіденційності даних знаходить оптимальне рішення в екосистемі AWS.

Таким чином, розробка компонентів SCADA-систем на базі AWS не тільки відповідає потребам гнучкого виробництва, але і визначає нові стандарти у впровадженні сучасних технологій у виробничі процеси.

Мета роботи – розроблення захищеного MQTT протоколу для передачі даних між компонентами SCADA системи для гнучкого виробництва.

Об'єкт роботи – процес віддаленого моніторингу та візуалізації даних у SCADA системі.

Предмет роботи – покращення методів передачі даних між компонентами SCADA системи на базі хмарних технологій Amazon Web Services.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати принципи організації SCADA систем;
- проаналізувати технології розробки SCADA систем;
- дослідити використання хмарних-сервісів в SCADA системах;
- розробка архітектури SCADA-системи гнучкого виробництва на основі AWS;
- зборка макету компонентів SCADA-системи;
- проведення експериментальних досліджень;
- провести аналіз отриманих результатів.

Кваліфікаційна робота оформлена згідно з рекомендаціями [1], та вимогам ДСТУ 3008:2015 [2], а результати даної кваліфікаційної роботи описані в наукових статтях та їх можна побачити в гугл академії [3].

1 АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ РОЗРОБКИ SCADA-СИСТЕМ

1.1 Принципи організації SCADA систем

SCADA (Supervisory Control and Data Acquisition) представляє собою програмний комплекс, призначений для виконання функцій збирання, обробки, відображення та архівації інформації, пов'язаної із моніторингом або управлінням об'єктом в реальному часі. Програмне забезпечення цього типу може функціонувати як частина автоматизованої системи управління технологічними процесами (АСУ ТП), автоматизованої системи контролю та керування в енергетичній сфері (АСКОЕ), системи моніторингу навколишнього середовища, наукового експерименту, автоматизації будівель і т. д. SCADA-системи застосовуються в різних галузях, де важливий операторський контроль за технологічними процесами.

SCADA є складовою частиною автоматизованої системи управління технологічними процесами, а також і диспетчерської системи, яка відповідальна за моніторинг технологічних параметрів та віддалене управління обладнанням.

SCADA системи є вирішальними компонентами в промисловому управлінні та моніторингу. Вони дозволяють операторам контролювати та управляти технологічними процесами в реальному часі, використовуючи дані, які збираються з численних датчиків та приладів. Ці системи широко використовуються у багатьох галузях, таких як енергетика, виробництво, транспорт та водопостачання, забезпечуючи ефективне управління та моніторинг ключових процесів [4].

Основний принцип роботи SCADA полягає в зборі даних з різних джерел, їх передачі на центральну систему контролю, де вони аналізуються та відображаються для операторів. Це дозволяє операторам ефективно

контролювати та реагувати на будь-які зміни або проблеми в системі. Багато сучасних SCADA систем також включають можливості прогнозування, аналізу даних та автоматизованого керування, що полегшує прийняття важливих рішень.

Важливими складовими SCADA систем є сенсори та датчики, які забезпечують збір різноманітних даних про технологічні процеси. Ці дані потім передаються на центральний сервер, де вони обробляються та відображаються в зручній для аналізу формі для операторів та керівників. Важливою складовою є також можливість віддаленого керування, що дозволяє операторам змінювати параметри процесів дистанційно, забезпечуючи ефективний контроль навіть на віддалених об'єктах.

Ще однією важливою характеристикою SCADA систем є їх здатність до інтеграції з іншими системами автоматизації, такими як системи управління виробництвом (MES) та системи управління енергетичними мережами. Це дозволяє створювати єдину інфраструктуру для керування всіма аспектами виробничого процесу та забезпечує більшу ефективність та координацію між різними системами.

Також варто відзначити, що SCADA системи здатні забезпечувати безпеку даних та системи в цілому за допомогою застосування сучасних методів шифрування та захисту від несанкціонованого доступу. Це особливо важливо у сучасному кібербезпечному середовищі, де кібератаки можуть стати серйозною загрозою для промислових систем [5].

Підтримка та навчання персоналу також є ключовими аспектами в ефективному функціонуванні SCADA систем. Працівники, які використовують SCADA системи, повинні бути добре підготовлені для ефективного використання системи та ефективно реагувати на можливі проблеми. Крім того, постійна технічна підтримка є необхідною для вирішення будь-яких технічних питань чи проблем, що виникають у процесі експлуатації системи.

Для розуміння більш глибоких аспектів SCADA систем, важливо розглянути їх архітектуру та ключові компоненти. Типова SCADA система складається з декількох ключових складових, таких як робочі місця оператора, сервери збору даних, комунікаційні канали та пристрої збору даних. Класична схема побудови SCADA системи наведена на рисунку 1.1.

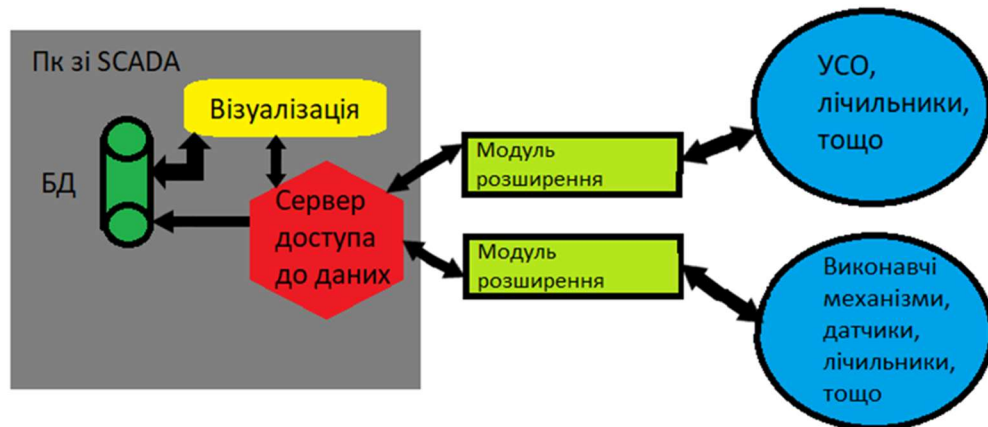


Рисунок 1.1 – Класична схема побудови SCADA системи

Робочі місця оператора є основним інтерфейсом, через який оператори спостерігають за процесами та приймають управляючі рішення. Ці місця забезпечують візуалізацію даних у вигляді графіків, діаграм, карт та інших форматів для зручного аналізу. Крім того, вони надають можливість операторам взаємодіяти з системою, змінювати параметри процесів та вживати відповідних заходів у разі виникнення проблем.

Сервери збору даних відповідають за збір, обробку та зберігання даних, які надходять з датчиків та пристроїв збору даних. Вони виконують важливі завдання зберігання історичних даних, а також забезпечують резервне копіювання для забезпечення надійності даних. Крім того, сервери забезпечують можливість аналізу даних для виявлення тенденцій та аномалій у процесах виробництва [6].

Комунікаційні канали відіграють критичну роль у передачі даних між різними компонентами SCADA системи. Ці канали можуть включати провідні

та бездротові засоби зв'язку, такі як Ethernet, Wi-Fi, а також спеціалізовані мережі, такі як MODBUS, DNP3 та інші. Ефективна комунікація є ключовим фактором для забезпечення швидкого та надійного обміну даними між різними компонентами системи.

Пристрої збору даних, такі як сенсори, датчики, контролери та вимірювальні пристрої, встановлені на різних об'єктах контролю і забезпечують збір реальних даних про технологічні процеси. Ці пристрої передають дані через комунікаційні канали до серверів збору даних для подальшої обробки та аналізу. Типова схема взаємодії компонентів SCADA системи наведена на рисунку 1.2.

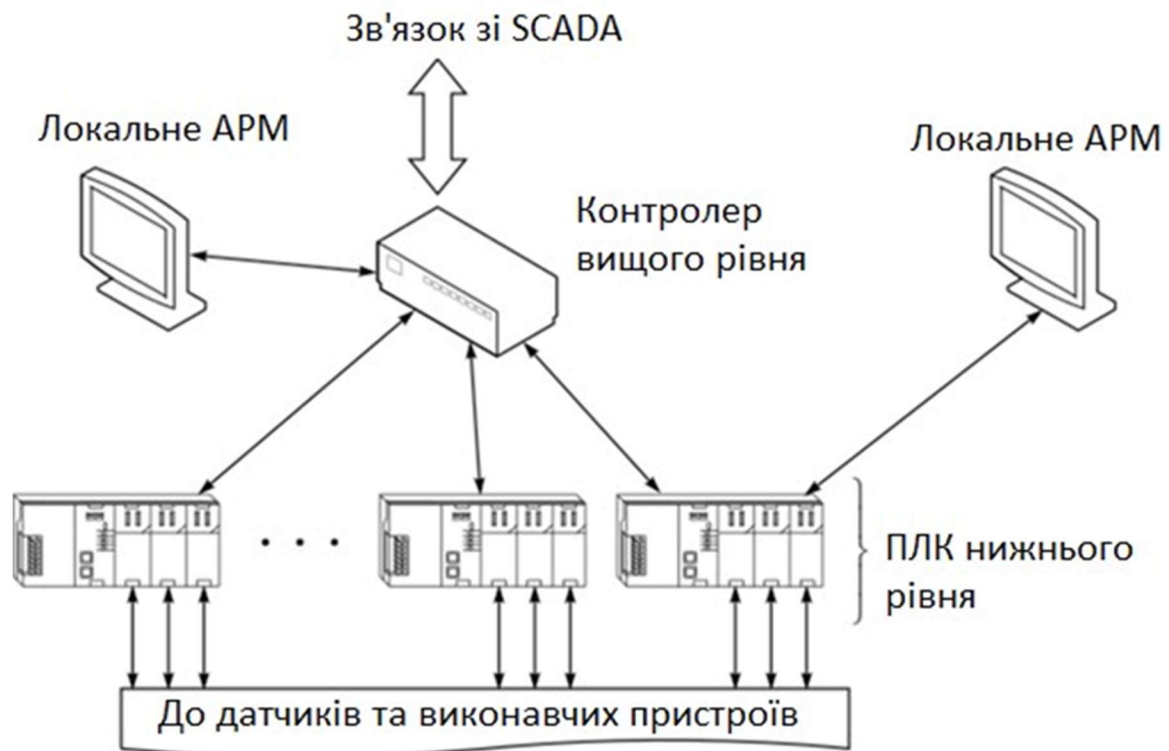


Рисунок 1.2 – Типова схема взаємодії компонентів SCADA системи

Крім того, важливо враховувати ефективність SCADA систем у вирішенні проблем енергоефективності та оптимізації виробничих процесів. Вони дозволяють виявляти потенційні джерела енергозатрат та вживати

відповідних заходів для їх усунення. Крім того, за допомогою аналізу даних SCADA можна виявити ефективність технологічних процесів та виявити можливість їх оптимізації для забезпечення економії ресурсів та зниження витрат [7].

Важливим аспектом дослідження SCADA систем є їх пристосування до сучасних тенденцій, таких як Інтернет речей (IoT) та штучний інтелект (AI). Інтеграція SCADA з IoT дозволяє підприємствам збирати ще більше даних з різних джерел, таких як датчики забруднення повітря, енергозатрати та інші, що дозволяє здійснювати більш детальний аналіз та приймати ефективніші управляючі рішення.

Щодо штучного інтелекту, його застосування у SCADA системах може полегшити автоматизацію аналізу даних, прогнозування тенденцій та аномалій у процесах виробництва. Алгоритми машинного навчання можуть допомогти виявляти недоліки та оптимізувати процеси без участі людини, забезпечуючи більш ефективну та точну роботу системи [8].

Для подальшого розвитку SCADA систем важливим є також дослідження нових технологій збору даних та їх обробки. Використання сучасних сенсорів та датчиків забезпечить більш точний та швидкий збір даних, що дозволить операторам ефективніше контролювати процеси та реагувати на можливі відхилення.

Одним з викликів, який стоїть перед дослідниками, є розробка ще більш надійних та безпечних систем SCADA. Враховуючи загрози кібербезпеки, важливо постійно вдосконалювати системи захисту, включаючи застосування передових методів шифрування, аутентифікації та моніторингу, щоб запобігти можливим кібератакам та порушенням безпеки.

Крім того, дослідження в області стандартизації та протоколів важливо для забезпечення сумісності та інтеграції між різними SCADA системами. Розробка уніфікованих стандартів дозволить забезпечити більш швидку та ефективну обміну даними між різними пристроями та системами, що сприятиме вдосконаленню індустрії в цілому [9].

У подальшому дослідженні SCADA систем важливим є також розгляд їх впливу на сталість та екологічність. Ефективне управління технологічними процесами допомагає зменшити споживання ресурсів та виробництво відходів, що сприяє збереженню навколишнього середовища та створенню більш сталої виробничої моделі.

Також, активно розвивається впровадження концепцій Industry 4.0 в SCADA системи. Це включає в себе розробку інтелектуальних фабрик, де SCADA системи взаємодіють з іншими пристроями та системами за допомогою Інтернету речей та цифрових технологій. Інтеграція цих концепцій дозволяє створювати автономні та автоматизовані виробничі процеси, що прискорює виробництво та забезпечує його більшу ефективність.

Розвиток мереж 5G також відіграє важливу роль у дослідженнях SCADA систем. Це дозволяє забезпечити більш швидку та надійну передачу даних, що важливо для оперативного контролю та управління виробничими процесами. Використання мереж 5G забезпечує низьку затримку та високу швидкість передачі даних, що дозволяє операторам більш точно реагувати на зміни в процесах та забезпечує безперебійну роботу системи.

Дослідження в області забезпечення надійності та безперебійності роботи SCADA систем також має велике значення. Розробка нових методів резервування та відновлення дозволяє забезпечити безперебійну роботу виробничих процесів навіть у випадку виникнення непередбачених ситуацій або збоїв. Це сприяє збереженню часу та ресурсів, що необхідних для відновлення нормального функціонування системи [10].

Організація SCADA систем базується на декількох ключових принципах, які сприяють їх ефективній роботі та забезпечують надійність та безпеку процесів.

У таблиці 1.1 представлені основні принципи SCADA систем.

Таблиця 1.1 – Основні принципи SCADA систем

Принцип	Опис
Централізоване керування	У SCADA-системах основний контроль технологічних процесів здійснюється централізовано з одного операторського пункту.
Збір та обробка даних	Система здійснює збір і подальшу обробку даних з об'єкта моніторингу для надання операторам зрозумілої інформації.
Реальний час	SCADA працює в режимі реального часу, що дозволяє операторам миттєво реагувати на зміни в технологічних процесах.
Візуалізація інформації	Інформація відображається у вигляді графічних елементів та інтерфейсів, що полегшує сприйняття та контроль оператора.
Автоматизація та контроль	SCADA забезпечує можливість автоматизації і контролю за різними параметрами технологічних процесів на об'єкті.
Архівування та зберігання історії	Система зберігає дані про минулі стани технологічних процесів, що дозволяє аналізувати та оптимізувати їх роботу.
Гнучкість і розширюваність	SCADA системи дозволяють легко впроваджувати нові функції та розширювати можливості відповідно до потреб користувача.

Усі ці принципи взаємодіють між собою для забезпечення ефективного та безпечного функціонування SCADA систем, що дозволяє підприємствам оптимізувати виробничі процеси та підвищувати загальну продуктивність.

1.2 Програмні технології розробки SCADA систем

Безумовно, системи SCADA інтегрують складні інструменти трендів і аналізу, які полегшують поглиблений аналіз історичних даних і оцінку продуктивності. Ці інструменти пропонують розширену візуалізацію даних, комплексний статистичний аналіз і надійні функції звітності. Вони дозволяють операторам і менеджерам розпізнавати закономірності, прогнозувати тенденції та приймати обґрунтовані рішення для оптимізації промислових процесів і підвищення загальної ефективності.

Крім того, щоб забезпечити безперервну роботу та мінімізувати час простою, системи SCADA використовують надійні механізми резервування та відновлення після відмови. Ці механізми включають дублювання критичних компонентів, таких як сервери та комунікаційні мережі, а також реалізацію автоматизованих процедур відновлення після відмови. Швидко перемикаючись на резервні системи у разі збоїв апаратного чи програмного забезпечення, ці механізми підтримують надійність системи та забезпечують безперебійний моніторинг і контроль промислових процесів.

У сфері корпоративної інтеграції системи SCADA легко інтегруються з системами планування ресурсів підприємства (ERP) і системами управління активами. Ця інтеграція спрощує потік даних в організації, сприяючи ефективному розподілу ресурсів, оптимізованому плануванню виробництва та покращенню координації між різними відділами. Забезпечуючи синхронізацію даних між системою SCADA та ширшими корпоративними системами, організації можуть досягти підвищеної операційної ефективності та покращити можливості прийняття рішень.

Системи SCADA також включають надійні функції для відповідності нормативним вимогам і звітності. Ці функції гарантують, що система дотримується галузевих норм і стандартів, збираючи та зберігаючи дані, необхідні для цілей регулятивної звітності. Дозволяючи організаціям демонструвати відповідність під час аудитів та перевірок, ці функції сприяють

розвитку культури підзвітності та дотримання нормативних вимог у промисловому середовищі, підвищуючи загальну прозорість і надійність роботи.

Виробники SCADA випускають одну основну версію та одну або дві додаткові дрібні версії щороку. Ці продукти швидко еволюціонують, щоб скористатися новими ринковими можливостями, відповідати новим вимогам своїх клієнтів та використовувати нові технології.

Як вже зазначалося, більшість продуктів SCADA, які були оцінені, розкладають процес на "атомні" параметри, до яких призначається тег-ім'я. Це є неефективним у випадку дуже великих процесів, коли потрібно налаштувати дуже великі набори тегів. Оскільки промислові застосунки зростають за обсягом, нові версії SCADA тепер розробляються таким чином, щоб обробляти пристрої навіть цілі системи як повноцінні сутності (класи), які інкапсулюють всі свої специфічні атрибути та функціональні можливості. Крім того, вони також підтримуватимуть багатокорданний розвиток.

Що стосується нових технологій, продукти SCADA тепер приймають:

- веб-технології, ActiveX, Java та інше;
- OPC як засіб внутрішнього зв'язку між клієнтськими та серверними

модулями. Таким чином, має бути можливість підключення модулів сторонніх виробників, сумісних з OPC, до цього продукту SCADA.

У сучасний час на всіх рівнях створених АСУТП використовуються програмовані технічні засоби, до яких повинне бути встановлене програмне забезпечення для вирішення функціональних завдань. Загалом, це програмне забезпечення включає в себе програмне забезпечення ПЗО, що забезпечують зв'язок з ГОУ, а також прикладне програмне забезпечення АРМів, яке вирішує завдання їх зв'язку з ПЗО та організації інтерфейсу «людина-машин» НМІ (Human-Machine Interface) для осіб, що приймають управлінські рішення: операторів, диспетчерів, управлінців [11].

У цьому розділі розглянемо сучасні принципи розробки програмного забезпечення для АРМів різних рівнів АСУТП.

Програмне забезпечення розробляється відповідно до проектної документації на етапі створення АСУТП і призначене для встановлення в АРМи, які повинні відстежувати поточний стан окремих ТхО та ТП в цілому, забезпечуючи можливості ухвалення управлінських рішень і формування управлінських впливів як з метою оперативної зміни параметрів ТП, так і з метою організаційного впливу на служби, що забезпечують виконання ТП відповідно до регламенту. Таким чином, розроблюване програмне забезпечення повинно бути спрямоване як на вирішення завдань управління обладнанням при взаємодії з ТП, так і на організацію ефективної взаємодії типу "людина (особа, що приймає управлінські рішення на певному рівні, в певному обсязі функцій) – система". При цьому принциповою особливістю програмного забезпечення, що розробляється для АСУТП, є необхідність вирішення зазначених завдань у режимі реального часу, в якому нормується час реакції на кожний збурюючий вплив як з боку ТП, так і з боку АРМ.

Під час розробки програмного забезпечення будь-якої складності завжди існує проблема: як забезпечити те, щоб розробники АСУТП і технологи, для яких створюється АСУТП, могли на зрозумілій для обох сторін мові відслідковувати повноту технічних вимог до автоматизації окремих ТхО або всього ТП і зробити це найбільш ефективно. Якщо на рівні взаємодії АРМів з ТОУ завдання програмування можуть бути чітко сформульовані на основі математичних описів законів управління, то на рівні НМІ таких законів немає. Тому виникло питання: яким чином можна описати людино-машинний інтерфейс щодо вирішення певних задач достатньо формалізовано, щоб уникнути непорозумінь щодо сформульованих вимог до автоматизації не лише різними спеціалістами, але і тими ж самими фахівцями на різних етапах створення системи [12].

Застосування SCADA-системи під час створення ПО забезпечує:

– можливість візуального відображення ТП у формі мнемосхем, зручних для сприйняття не лише операторами та диспетчерами, а й спеціалістами технологічних служб підприємства; розробка мнемосхемних

зображень здійснюється за допомогою виразних технологічних "образів" та мультиплікаторів органів управління різних типів, наприклад, органи управління у вигляді кнопок управління, вимикачів, повзункових або обертових засобів встановлення параметрів для регуляторів; екранні форми відображення параметрів процесу, такі як цифрові табло, стрілкові, смугові або цифрові багатопозиційні індикатори; графіки зміння вимірюваних величин; сигнальні табло різних форм і тематичного змісту;

- взаємодію АРМ з різними типами зовнішніх пристроїв, які взаємодіють з об'єктом, з використанням необхідних драйверів, які підтримують роботу цих пристроїв;

- підтримку функціонування ППЗ в реальному часі, з нормованим часом реакції системи на зовнішні події;

- отримання від пристроїв, що взаємодіють з ТП, інформації про вимірювані та контрольовані технологічні параметри, з фіксацією цих подій у координатах поточного часу (дата, год., хв., с);

- обробку вимірювальної інформації, представленої результатами вимірювань потрібної вимірювальної одиниці, результати вимірювань контролюються відносно межових значень допустимого діапазону або швидкості вимірювання вимірюваних величин, а також подаються у різних графічних формах;

- сповіщення (світлове, звукове) експлуатаційного та обслуговуючого персоналу про аварійні події, про неприпустимі умови функціонування технологічного обладнання або обладнання автоматики;

- обов'язкове архівне реєстрування як аварійних повідомлень, так і дій експлуатаційного персоналу в аварійних ситуаціях;

- ведення архівів подій, що відбуваються при управлінні ТП: формуються архіви часових станів системи (повне або вибіркоче збереження параметрів процесу через задані інтервали часу постійно або за умови), а

також архіви аварій, подій та поведінки змінних процесу в часі (так звані тренди);

- можливість диспетчерського або операторського управління ТП або ТхО шляхом формування команд управління з передачею їх до відповідних ПЗО; фіксація керуючих дій особи, що приймає управлінські рішення, в архівах подій;

- формування зведень та інших звітних документів на основі архівної інформації та у потрібній формі подання;

- пряме автоматичне керування ТхО за заданими алгоритмами, виконання математичних та логічних обчислень при реалізації алгоритмів формування керуючих впливів залежно від характеристик об'єкта, включаючи режими "консультанта", супервізорного управління, прямого цифрового управління, а також вибір необхідних алгоритмів із спеціальної бібліотеки;

- можливість захисту від несанкціонованого доступу до управління об'єктом;

- можливості багатовіконного графічного інтерфейсу та інших очевидних функцій, таких як імпорт зображень, створення власних бібліотек динамічних об'єктів, елементів мнемосхем тощо.

Отже, використання SCADA-технології забезпечує практично необмежені можливості для втілення функцій: збирання інформації про ТП; автоматичного управління окремими ТхО та різними методами автоматизованого управління ТП в цілому; індивідуального вирішення в кожному АРМі інтерфейсу інтерактивної взаємодії типу "людина-система" з використанням мнемонічно інтуїтивно зрозумілих зображувальних образів; збереження історії змін управління процесом та історії формування управляючих впливів; активної взаємодії з іншими засобами, що дозволяють створювати зображення, які можуть бути використані для вирішення завдань візуалізації в рамках конкретного проекту. Знову звернемо увагу на те, що SCADA-технологія лише "забезпечує можливості" розробникам програмного забезпечення, дозволяючи: скоротити терміни та значно знизити

трудомісткість розробки програмного забезпечення; забезпечити "прозорість" програмних рішень та принцип їх послідовного розвитку; підвищити надійність прийнятих у програмному забезпеченні рішень і т.д. Те, як ці можливості будуть реалізовані в рамках конкретного проекту, буде залежати вже від проектного рішення, від здібностей розробника програмного забезпечення, від можливостей обраної SCADA-системи.

Як уже зазначалося вище, надзвичайно важним у використанні технології SCADA є можливість налагодження продуктивного діалогу між розробниками програмного забезпечення та технологами, що автоматизують ТП, на зрозумілій обом сторонам мові візуальної реалізації функцій управління, що створюються для АСУТП. Суть такого діалогу полягає в тому, що розробка програмного забезпечення починається, якби "з кінця", з розробки відображення мнемосхеми технологічного процесу з урахуванням функцій автоматизованого управління, визначення форм подання вимірювальної та сигналізуючої інформації, складу та форми ведення архівних даних.

Під час діалогу на рівні узгодження вибираються бажані форми інтерфейсу "людина-система" для вирішення завдань контролю та управління на рівні кожного робочого місця. Тільки після такої діалогової узгодженості всіх питань, які фактично визначають призначення АСУТП, розробники системи можуть приступити до розробки нижньої частини АСУТП за вже проробленими та затвердженими інтерфейсами "людина-система" без обтяження, що щось з вимог замовника "не зрозуміло", чогось пропущено, а щось виявляється зайвим.

1.3 Використання хмарних-сервісів в SCADA системах

Застосування хмарних сервісів в SCADA системах розширює можливості збирання, аналізу та моніторингу даних з великих виробничих або промислових об'єктів за допомогою Інтернету. Використання цих сервісів

сприяє гнучкості, масштабованості та доступності інформації з будь-якого місця, де є підключення до Інтернету.

Основні переваги використання хмарних сервісів у системах SCADA включають масштабованість, здатність розширювати обсяг обробки і зберігання даних відповідно до потреб користувача, що особливо корисно для великих промислових об'єктів з великим обсягом інформації. Можливість уникнути великих капітальних витрат на власні сервери і пристрої для зберігання, переходячи до хмарних рішень, дозволяє зменшити витрати на обладнання. Гнучкість доступу забезпечує можливість отримання даних з будь-якого місця, де є доступ до Інтернету, сприяючи віддаленому моніторингу і управлінню об'єктами. Забезпечення безпеки полягає у використанні високого рівня захисту даних і шифрування для забезпечення конфіденційності та цілісності інформації, передаваної через мережу. Регулярні оновлення програмного забезпечення від постачальників хмарних сервісів забезпечують доступ до найновіших версій програм і додатків, що сприяє автоматичному оновленню і підтримці актуальності системи.

Однак перед використанням хмарних сервісів в SCADA системах важливо враховувати питання щодо приватності та безпеки даних. Вибір надійного постачальника та дотримання стандартів безпеки та конфіденційності є ключовими аспектами при розгляді цих рішень.

З розвитком хмарних технологій та їх можливостей трансформація SCADA систем була питанням часу. Технології Industry 4.0 суттєво розширюють спектр можливостей взаємодії зі SCADA системами та трансформують попередні підходи до їх створення. Деякі з них приведені на рисунку 1.3.

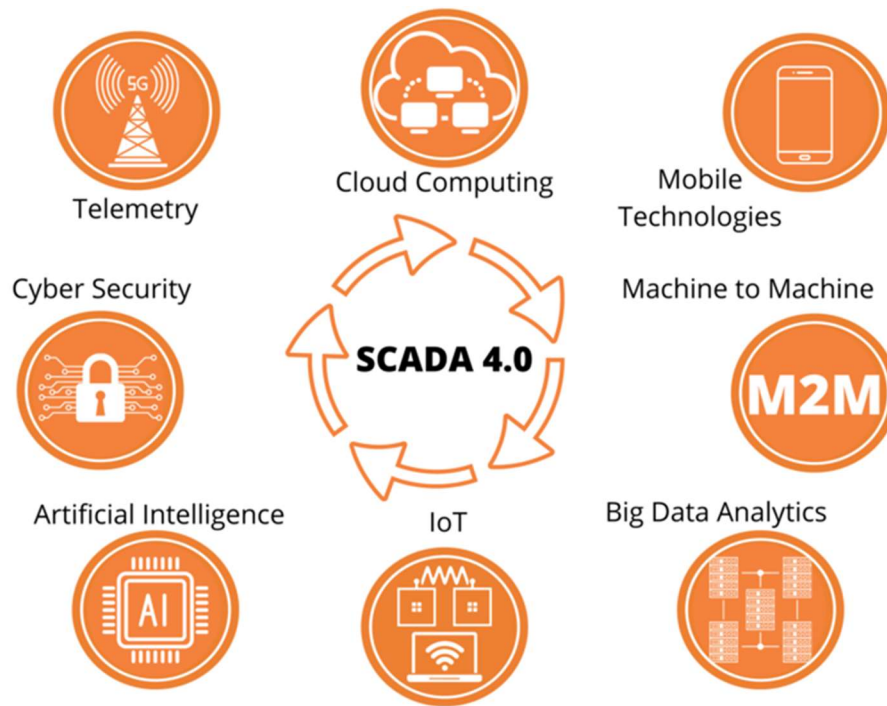


Рисунок 1.3 – Розширені можливості SCADA систем які з’явилися завдяки Industry 4.0 [13]

Нижче перераховані деякі конкретні способи, які підкреслюють, як саме хмарні сервіси змінюють сферу SCADA:

- збереження даних в хмарі; це дозволяє зберігати великі обсяги даних на хмарних серверах, уникнувши таким чином необхідності у власних серверах або зберігальних пристроях, також гарантує віддалений доступ до даних з будь-якого місця;
- аналіз даних у реальному часі; хмарні сервіси можуть забезпечувати миттєвий аналіз даних у реальному часі, дозволяючи операторам миттєво реагувати на будь-які зміни або проблеми на об'єктах;
- віддалене моніторинг та керування; використання хмарних сервісів дозволяє операторам моніторити та керувати системами SCADA навіть з віддалених місць, що робить його особливо корисним для великих або глобальних промислових об'єктів;
- зменшення витрат на IT-інфраструктуру; замість вкладання великих коштів у власні сервери та IT-інфраструктуру, компанії можуть користуватися

хмарними сервісами за оплату за використання, що дозволяє економити кошти на обслуговуванні та підтримці;

– забезпечення безпеки даних; багато постачальників хмарних сервісів пропонують широкі заходи безпеки, включаючи шифрування даних та мережеві заходи безпеки, що робить хмарні сервіси надійними для зберігання важливої інформації;

– оновлення та підтримка; постачальники хмарних сервісів часто забезпечують регулярні оновлення та технічну підтримку, що дозволяє користувачам завжди працювати з найновішими версіями програмного забезпечення та забезпечити безперебійну роботу системи.

Впровадження хмарних сервісів в системи SCADA принесло численні переваги, які суттєво змінюють підхід до управління технологічними процесами. Інтеграція Інтернету речей (IoT) з SCADA дозволяє отримувати дані з різних джерел, від датчиків до IoT-пристроїв, що забезпечує повніше та більш точне зображення стану об'єктів моніторингу.

Однією з ключових переваг є застосування машинного навчання та аналітики даних у системах SCADA. Це дає змогу передбачати відмови обладнання, виявляти аномалії в роботі та оптимізувати виробничі процеси, забезпечуючи вищу ефективність та ресурсозбереження.

Географічний розподіл об'єктів стає можливим завдяки хмарним сервісам, що дозволяють збирати дані з великої кількості об'єктів, розташованих на значній відстані один від одного. Це робить централізоване керування та моніторинг розподіленими системами більш ефективним та зручним.

Гнучкість налаштувань та масштабованість хмарних сервісів надає користувачам можливість адаптувати обсяг обробки даних та зберігання до своїх потреб. Це особливо важливо в умовах змінних виробничих обсягів та розвитку підприємства.

Нарешті, управління даними з багатьох джерел стає більш простим завдяки хмарним сервісам, які забезпечують зручний збір, аналіз та

використання інформації з сенсорів, IoT-пристроїв та SCADA-систем. Це дозволяє приймати обґрунтовані стратегічні рішення на основі вичерпної та точної інформації.

Впровадження хмарних сервісів у системах SCADA дозволяє підвищити автоматизацію, ефективність та точність виробничих процесів. Однак при впровадженні таких систем важливо враховувати вимоги до безпеки даних, вибираючи надійного постачальника хмарних послуг та використовуючи найкращі практики кібербезпеки для захисту конфіденційної інформації.

Основні галузі четвертої промислової революції включають такі аспекти, як кібербезпека, обчислювальні послуги у хмарі, Інтернет речей, великі обсяги даних і концепція взаємодії машин (M2M). Виробництво, що базується на таких підходах, зазвичай має модульну структуру і повністю контролюється програмно, що дозволяє налаштовувати середовище, збирати та аналізувати дані на протязі тривалого часу, а також швидко отримувати доступ до інформації з будь-якого місця світу. Мікроконтролери, що використовують концепцію Інтернету речей (IoT), підключаються до хмарних сервісів, які збирають дані в реальному часі та обробляють їх згідно з потребами виробництва. Тому при розробці системи обробки виробничих даних необхідно враховувати важливість захисту інфраструктури, а також створювати автоматизований потік інформації всередині системи. Також для обробки даних важливо розробляти візуалізацію інформації.

На сьогоднішній день існують сучасні системи дистанційного моніторингу та управління, які використовують різні хмарні сервіси для отримання та обробки даних. Ці системи можуть бути використані для побудови SCADA систем та їх компонентів. До числа найпопулярніших хмарних сервісів які мають відповідні рішення, належать Amazon Web Services та Azure.

За допомогою AWS (Amazon Web Services) IoT, можна підключитися до веб-сервісів та інших пристроїв, що дозволяє не лише обробляти дані з пристроїв, але й взаємодіяти з ними. Крім того, ця система забезпечує

надійний захист ваших даних та інформації. Якщо пристрій, з яким потрібно взаємодіяти, працює в автономному режимі, це не становить проблеми [14].

Шлюз пристроїв може виступати точкою входу для пристроїв Інтернету речей, які, в свою чергу, підключені до AWS. Цей шлюз управляє всіма підключеними пристроями, які є активними на даний момент. Він також надає семантику для деяких протоколів, які забезпечують безпеку та ефективність взаємодії з ядром AWS IoT [15].

Однією з ключових характеристик Інтернету речей (IoT) є набір керованих та платформових послуг, що зберігаються та виконуються у хмарних сервісах. Ці послуги дозволяють підключати, керувати та моніторити мільярди IoT пристроїв. Таке рішення може застосовуватися для аналізу даних та різноманітного управління на підприємствах, а також для розробки та впровадження рішень, використовуючи хмарні сервіси. Схема можливої реалізації взаємодії з IoT пристроями за допомогою Amazon Web Services наведена на рисунку 1.4.

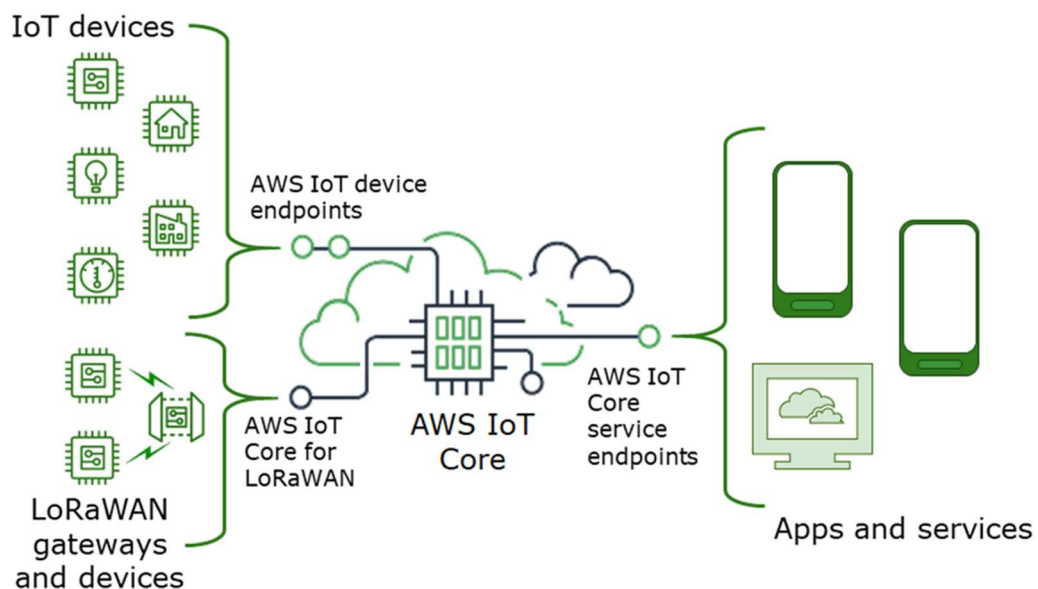


Рисунок 1.4 – Схема можливої реалізації взаємодії з IoT пристроями за допомогою Amazon Web Services [16]

При використанні термінів, пов'язаних з Інтернетом речей, зазвичай згадують такі компоненти: речі – фізичне або промислове обладнання, датчики та пристрої, що підключаються до хмарних сервісів або через них, інсайти – інформація, проаналізована та перетворена за допомогою Artificial Intelligence (AI) або людьми, дії – використання перетвореної та проаналізованої інформації у бізнес-рішеннях, а також системи та інструменти, що використовуються.

Типові пристрої Інтернету речей, що використовують для керування пристроями через підключення до Інтернету через WI-FI, зазвичай складаються з датчиків, що підключені до системи керування або Programmable Logic Controller (PLC). Прикладами можуть бути такі типові пристрої, як датчики тиску на водних насосах, датчики температури та вологості, вбудовані в кондиціонери та датчики присутності.

Зазвичай пристрої Інтернету речей використовують одностороннє з'єднання з хмарними сервісами, яке полягає у відправці стану або показів датчиків з апаратної частини до хмарного сервісу. Однак можливі і інші типи з'єднань, такі як отримання даних з хмари на апаратну частину або двостороннє з'єднання. Наприклад, датчик руху, який надсилає дані до хмари кожен раз при виявленні руху, або пристрій, який за допомогою датчиків зчитує параметри навколишнього середовища і надсилає повідомлення через канали IoT, коли дані виходять за межі заданих значень, або надсилання даних у реальному часі для візуалізації інформації з метою полегшення моніторингу систем.

1.4 Постановка завдання дослідження

Виходячи з проведеного аналізу можна зробити висновок, що аналіз існуючих технологій розробки SCADA-систем є актуальною темою на сьогоднішній день. Актуальність даної теми обумовлена тим, що SCADA-системи виявляють свою ефективність в умовах, де централізоване управління

не є пріоритетом, забезпечуючи при цьому стабільність і автономію в роботі. Крім того, аналіз SCADA-систем підтвердив їхню високу надійність та стійкість до відмов, що робить їх особливо ефективними у важливих сферах, де неперервність роботи системи є вирішальною.

Мета роботи – розроблення захищеного MQTT протокола для передачі даних між компонентами SCADA системи для гнучкого виробництва.

Об’єкт розробки – процес віддаленого моніторингу та візуалізації даних у SCADA системі.

Предмет розробки – модернізація методів передачі даних між компонентами SCADA системи на базі хмарних технологій Amazon Web Services.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати принципи організації SCADA систем;
- проаналізувати технології розробки SCADA систем;
- дослідити використання хмарних-сервісів в SCADA системах;
- розробка архітектури SCADA-системи гнучкого виробництва на основі AWS;
- зборка макету компонентів SCADA-системи;
- проведення експериментальних досліджень;
- провести аналіз отриманих результатів.

2 РОЗРОБКА АРХІТЕКТУРИ SCADA-СИСТЕМИ ГНУЧКОГО ВИРОБНИЦТВА НА ОСНОВІ AWS

2.1 Порівняльний аналіз можливостей хмарних сервісів

Інформаційні технології точно ввійшли в нашу повсякденну життя, і ми постійно стикаємося з поняттям «хмара». Хмарна інфраструктура являє собою середовище, в якому можливо видалене розвертання віртуальних машин (VM). Вона базується на фізичних серверах і гіпервізорах, програмно забезпечених, які дозволяють керувати віртуальними машинами на фізичному сервері. Гіпервізор створює віртуальне середовище, яке ізолює операційні системи та кожне додаток VM від інших, що дозволяє їм працювати незалежно. Він емулює апаратні ресурси та керує доступом віртуальної машини до фізичних ресурсів, таких як процесор, пам'ять, дисковий простір і набір.

Хмарні сервіси – це технологія, яка надає доступ до вичерпних ресурсів, зберігання даних і додатків через інтернет. Вони засновані на віртуальній інфраструктурі, яка складається з віддалених серверів і спеціального програмного забезпечення, що дозволяє керувати цією інфраструктурою. Суть хмарних сервісів полягає в наданні доступу до значущих потужностей та інфраструктури через інтернет [17].

Робота хмарних сервісів заснована на віддаленому збереженні даних і обробці інформації на віддалених серверах, що звільняє користувачів від необхідності мати фізичну інфраструктуру на своїх пристроях. Загальні сервіси дозволяють швидко масштабувати та оновлювати ресурси, забезпечуючи більшу продуктивність та ефективність роботи.

Розглянемо різні форми хмарних сервісів та їх можливості.

Порівняння декількох хмарних платформ Інтернету речей представлено у таблиці 2.1.

Таблиця 2.1 – Порівняння хмарних платформ

Платформи	Сумісність (Протоколи зв'язку)	Ключова пропозиція та її основна функція	Найбільш популярні варіанти використання
1	2	3	4
Google IoT	HTTP MQTT	Керування пристроями Google Cloud IoT Core Connectivity	Енергія Розумне паркування Транспорт та логістика
Microsoft Azure IoT	MQTT AMQP обидва через WebSockets HTTPS	Перевірка автентифікації підключення Azure IoT Hub Моніторинг пристроїв Управління пристроями IoT Edge	Охорона здоров'я Роздрібне виробництво
Amazon Web Services IoT Platform	HTTP MQTT WebSockets	AWS IoT Core Підключення Аутентифікація Двигун правил Середовище розробки	Умний город Підключений будинок Сельське господарство
Oracle IoT Intelligent Applications	HTTP MQTT	Хмарна служба Oracle IoT Підключення Моніторинг активів Аналіз даних Інтеграція даних	Розумне виробництво Прогнозне обслуговування Підключена логістика

Продовження таблиці 2.1

1	2	3	4
Salesforce	MQTT	Salesforce Service Cloud Підключення Моніторинг Управління даними Хмарна оркестровка Візуалізація	Виробництво Фінансові послуги Охорона здоров'я та науки про життя Роздрібна торгівля та споживчі товари
IBM Watson IoT	HTTP MQTT	Платформа VM Watson IoT Підключення Управління пристроєм Аналітика в реальному часі Блокчейн	Виробництво Сільське господарство Розумні будівлі
ThingsBoard Open- Source IoT Platform	MQTT CoAP HTTP	ThingsBoard Community Edition Збір даних Обробка Візуалізація Управління пристроєм	Розумний облік Розумна енергія Розумне землеробство Відстеження автопарку

Обрання AWS для реалізації проекту Інтернету речей визначається комплексом переваг, які ця хмарна платформа пропонує. Широка функціональність AWS IoT дозволяє ефективно керувати та масштабувати велику кількість підключених пристроїв, забезпечуючи високий рівень гнучкості у розробці та налаштуванні IoT-систем.

Важливим аспектом вибору AWS є високий рівень безпеки та дотримання відмінних практик у сфері інформаційної безпеки. AWS

встановив стандарти щодо захисту даних, надаючи користувачам впевненість у конфіденційності та цілісності їхніх IoT-даних.

Екосистема AWS також забезпечує широкий вибір додаткових сервісів та інструментів, які можуть бути легко інтегровані в проект. Від аналітики та машинного навчання до інших рішень, ці додаткові можливості дозволяють розширити функціональність вашого IoT-проекту та забезпечити додатковий рівень інновацій. Крім того, готовність до роботи в реальних умовах та надійність AWS дозволяють розробникам спрямовувати свої зусилля на створення високоефективних та стабільних IoT-систем, а не на вирішення проблем із інфраструктурою.

Враховуючи ці аспекти, обрання AWS для проектів Інтернету речей стає стратегічним кроком у напрямку створення інноваційних та надійних рішень у сфері підключених пристроїв.

2.2 Архітектура SCADA-системи гнучкого виробництва

Архітектура SCADA-системи гнучкого виробництва грає критичну роль сучасних виробничих системах. Вона забезпечує централізований контроль, моніторинг та інтеграцію з обладнанням, включаючи роботів, верстати та датчики. Системи SCADA гнучкого виробництва мають модульну структуру, дозволяючи легко інтегрувати нові пристрої та технології. Гнучкість досягається завдяки використанню хмарних технологій для зберігання та обробки даних. Це забезпечує високу доступність інформації в режимі реального часу та покращену масштабованість системи.

Архітектура SCADA-системи гнучкого виробництва прагне до створення інтелектуальних, адаптивних і ефективних виробничих середовищ, здатних ефективно адаптуватися до ринкових вимог, що змінюються. У центрі архітектури SCADA-системи гнучкого виробництва знаходиться централізований контроль, що забезпечує операторам та інженерам повну видимість поточних виробничих процесів. Інтеграція з промисловим

обладнанням, включаючи роботів та конвеєри, забезпечує ефективну взаємодію між різними пристроями. Модульність архітектури дозволяє легко вносити зміни та розширювати систему відповідно до потреб виробництва. Комунікаційні протоколи є важливим засобом для ефективного обміну даними між пристроями, забезпечуючи стандартизацію та сумісність у сфері промислової автоматизації. Безпека важлива для забезпечення надійності та конфіденційності даних у системі SCADA. Захист від кіберзагроз, автентифікація та авторизація є невід'ємними частинами архітектури [18].

Використання хмарних технологій вносить елемент гнучкості та масштабованості, забезпечуючи ефективне зберігання та обробку даних. Це також покращує доступність інформації у реальному часі, що є важливим аспектом сучасних виробничих систем. Архітектура SCADA-системи гнучкого виробництва прагне створення стійкого, гнучкого та інноваційного виробництва. Розвиток технологій у цій галузі продовжує сприяти більш інтегрованим, ефективним та керованим виробничим процесам, роблячи їх більш чуйними та пристосованими до динаміки сучасної промисловості.

Архітектура SCADA-системи гнучкого виробництва ефективно поєднує кілька ключових аспектів для забезпечення високої продуктивності та адаптивності. Центральний контроль та моніторинг забезпечують управління виробничими процесами та миттєвий доступ до даних для прийняття оперативних рішень.

Інтеграція з обладнанням – один із критичних елементів, що забезпечує взаємодію між різними пристроями та системами у реальному часі. Модульність архітектури надає гнучкість системі, дозволяючи легко інтегрувати нові технології та пристрої у виробничі процеси.

Комунікаційні протоколи є основою для ефективного обміну даними між пристроями, підтримуючи стандартизацію промислової автоматизації. Безпека даних відіграє важливу роль у захисті від кіберзагроз та забезпеченні конфіденційності виробничої інформації.

Використання хмарних технологій надає зручне рішення для зберігання та обробки даних, підвищуючи масштабованість та доступність інформації в реальному часі. Ця інтеграція забезпечує як високу ефективність, а й готовність до змін у виробничих вимогах.

Автономна SCADA-система має обмежену масштабованість, перш за все орієнтовану на локальне управління та моніторинг. Це часто підходить для невеликих об'єктів або підприємств, де масштаб операцій обмежений, і централізований контроль не є першорядною потребою. Надійність у такій системі оцінюється як висока, оскільки відсутність залежності від зовнішніх факторів, таких як мережеві з'єднання, може сприяти стабільності та безперебійній роботі. Тим не менш, можливості розширення та оновлення обмежені, що робить цей тип системи менш гнучким у разі зміни вимог чи збільшення масштабів операцій. Зазвичай автономні SCADA-системи знаходять застосування в лабораторіях, малих підприємствах або виробництвах з обмеженими обсягами даних. Вони можуть бути ефективними для ситуацій, де централізоване управління не є першорядною необхідністю, і головним чином, де потрібна надійна та автономна система контролю та моніторингу [19].

SCADA, що працюють в автономному режимі, відрізняються відносною незалежністю від зовнішніх факторів і мережевих з'єднань. Це забезпечує високий ступінь надійності в роботі, оскільки відсутність залежності від зовнішніх ресурсів та зв'язків знижує ймовірність збоїв через відключення мережі чи інших зовнішніх проблем. Однак, така система обмежена у своїй здатності масштабуватися та гнучко адаптуватися до змін, оскільки оновлення та розширення можуть бути ускладнені. Приклад автономної SCADA системи зображений на рисунку 2.1.

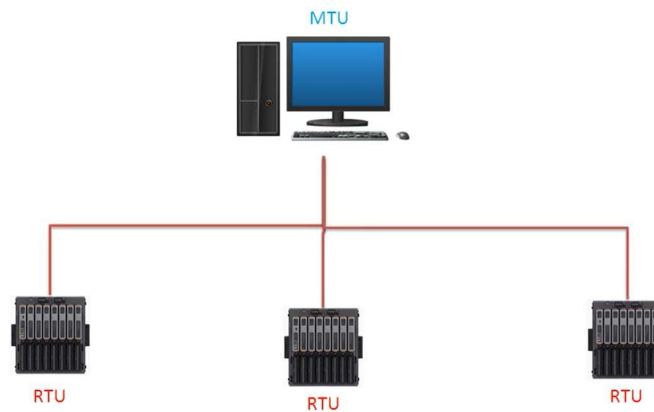


Рисунок 2.1 – Автономна SCADA система

Системи SCADA з автономним режимом знаходять застосування у сферах, де централізоване управління не потрібне насамперед, а підкреслюється стабільність та автономність у роботі.

Масштабованість в автономній SCADA-системі з резервуванням означає її здатність до локального масштабування операцій, при цьому система зберігає свою орієнтованість на локальний рівень. Ключовим елементом є впровадження резервних механізмів для забезпечення безперебійної роботи у разі збоїв, що може мати критичне значення в областях, де відсутність даних може спричинити серйозні наслідки.

Надійність автономної системи SCADA з резервуванням оцінюється як дуже висока завдяки впровадженню резервних механізмів, які роблять її стійкою до відмов у ключових компонентах.

Продуктивність в автономній SCADA-системі з резервуванням зазвичай залишається на високому рівні, оскільки додавання резервування не має значного впливу на оперативну ефективність. Приклад автономної SCADA системи з резервуванням зображений на рисунку 2.2.

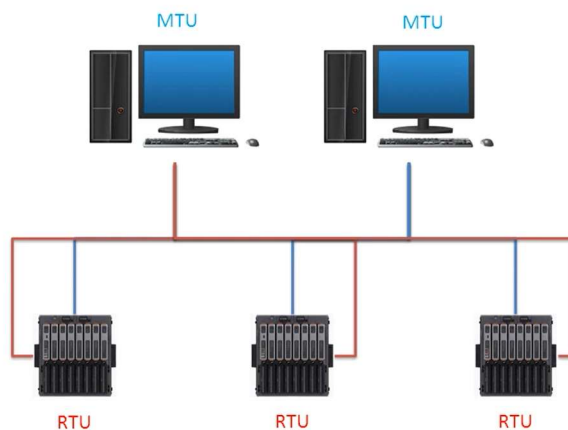


Рисунок 2.2 – Автономна SCADA система з резервуванням

Гнучкість системи збільшується за рахунок наявності резервування, що забезпечує більш гнучку адаптацію до умов і ситуацій, що змінюються, зберігаючи при цьому високий ступінь надійності.

Клієнт-серверна SCADA-система є архітектурним підходом, в якому функції системи розділені між клієнтськими та серверними компонентами. У клієнт-серверній SCADA-системі масштабованість надає високий рівень гнучкості, дозволяючи серверу обробляти значні обсяги даних, а клієнтам додавати чи видаляти з системи з легкістю. Гнучкість досягається завдяки централізованому керуванню з сервера, що полегшує внесення змін.

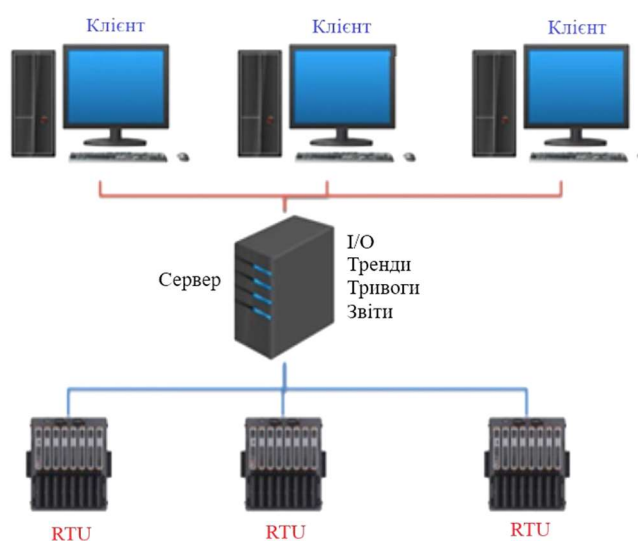


Рисунок 2.3 – Клієнт-серверна SCADA-система

Оновлення та розширення є зручнішими порівняно з автономними системами, оскільки централізований сервер може бути легко оновлений, і зміни розповсюджуються на всі клієнти. Приклад клієнт-серверної SCADA системи зображений на рисунку 2.3.

Інтеграція в клієнт-серверній системі є зручною завдяки централізованій структурі, що дозволяє легко інтегрувати інші системи із сервером SCADA. Відмовостійкість у клієнт-серверній системі залежить від реалізації заходів щодо забезпечення відмовостійкості на рівні сервера та мережі [20].

Клієнт-серверна SCADA-система з резервуванням, зображена на рисунку 2.4, представляє собою вищий рівень архітектури, включаючи заходи для забезпечення стійкості та резервування. Ця система проявляє масштабованість, що полягає у здатності ефективно адаптуватися до змін в вимогах, при цьому зберігаючи централізований контроль сервера. Надійність системи висока завдяки впровадженим механізмам резервування та стійкості до відмов у ключових компонентах.

Гнучкість архітектури клієнт-серверної системи з резервуванням досягається завдяки можливості централізованого керування та оновлення системи, полегшуючи внесення змін та оновлень.

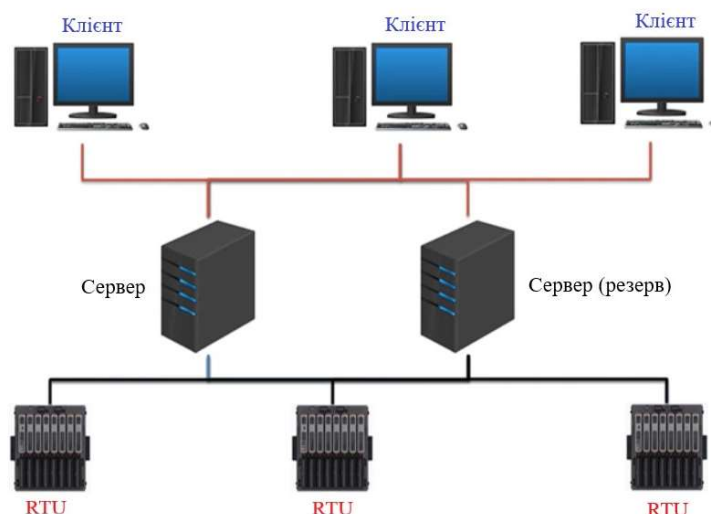


Рисунок 2.4 – Клієнт-серверна SCADA-система з резервуванням

Модульна SCADA-система забезпечує високу масштабованість, дозволяючи легко додавати і видаляти модулі в залежності від потреб, що змінюються. Надійність такої системи залежить від якості взаємодії між модулями, що потребує ретельного проектування. Модульна система має високу гнучкість, що дозволяє адаптувати модулі до змінних вимог. Однак, модулі зазвичай розробляються з урахуванням можливості інтеграції. Відмовостійкість модульної системи залежить від механізмів резервування всередині неї. Система може бути більш стійкою, якщо передбачені резервні модулі та механізми перемикавання. Приклад модульної SCADA системи зображений на рисунку 2.5.

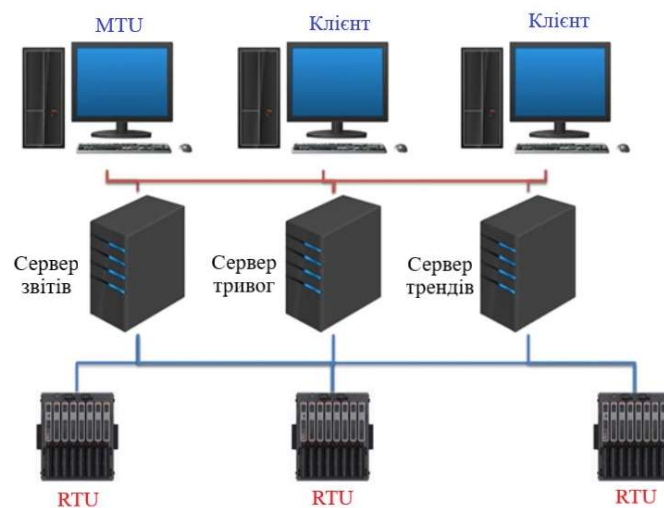


Рисунок 2.5 – Модульна SCADA-система

Мережева SCADA-система є архітектурним підходом, заснованим на взаємодії між компонентами через мережу. Важливим аспектом цієї системи є висока масштабованість, що дозволяє легко інтегрувати нові вузли та компоненти у мережу. Це робить систему ефективною для розподілених об'єктів із різними масштабами. Надійність SCADA-системи залежить від якості мережної інфраструктури. Гнучкість мережної SCADA-системи проявляється у здатності легко взаємодіяти з різними вузлами у мережі. Це

забезпечує гнучкість у підключенні нових компонентів та пристроїв до системи. Приклад системи можна побачити на рисунку 2.6.

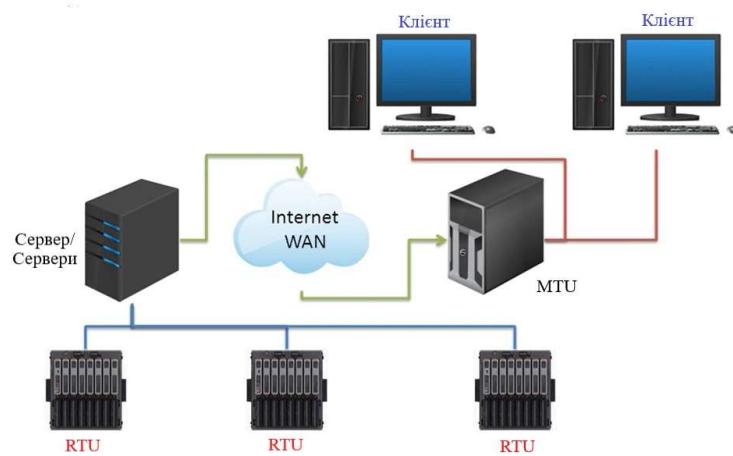


Рисунок 2.6 – Мережева SCADA-система

Архітектура SCADA-системи на основі інтернету речей (IoT) поєднує промислові системи управління та моніторингу з технологіями IoT, створюючи гнучке, масштабоване та інтегроване середовище для контролю за виробничими процесами. Ключові компоненти такої архітектури включають пристрої збору даних, такі як датчики і актуатори, які збирають інформацію про температуру, вологість, тиск і інші параметри. Протоколи зв'язку, такі як MQTT, CoAP та HTTP/HTTPS, забезпечують ефективний обмін даними між IoT-пристроями та центральним сервером SCADA/IoT. Центральний сервер, найчастіше розміщений у хмарному середовищі (AWS, Azure, Google Cloud), служить для зберігання та обробки даних від IoT-пристроїв. База даних зберігає архівні дані для подальшого аналізу. Програмне забезпечення включає SCADA-систему, що забезпечує моніторинг і управління виробничими процесами, і IoT-платформу, що управляє і моніторить IoT-пристрої і полегшує взаємодію з SCADA. Безпека системи забезпечується шифруванням даних під час передачі, автентифікацією та авторизацією.

Інтеграція з іншими системами, такими як системи управління підприємством (ERP), дозволяє ефективно взаємодіяти з бізнес-процесами та виробничими операціями. Все разом, архітектура SCADA на базі IoT створює гнучке рішення, яке в реальному часі моніторить, управляє та оптимізує промислові процеси, використовуючи сучасні технології інтернету речей. У такій архітектурі кожен пристрій IoT, такі як сенсори та актуатори, виконує роль безпосередніх збирачів даних на місцях виробництва. Ці пристрої передають дані через спеціалізовані протоколи зв'язку, забезпечуючи ефективний обмін інформацією з центральним сервером. Він, у свою чергу, обробляє, зберігає та надає доступ до цих даних, створюючи глобальне уявлення виробничих процесів. Центральний сервер, як правило, є хмарним, що забезпечує високу масштабованість та доступність. Він взаємодіє з базою даних, де зберігаються архівні дані, та забезпечує можливість аналізу та звітності. SCADA-система на цьому сервері надає інтерфейс для моніторингу та керування, дозволяючи операторам та інженерам приймати оперативні рішення [21]. Приклад системи можна побачити на рисунку 2.7.

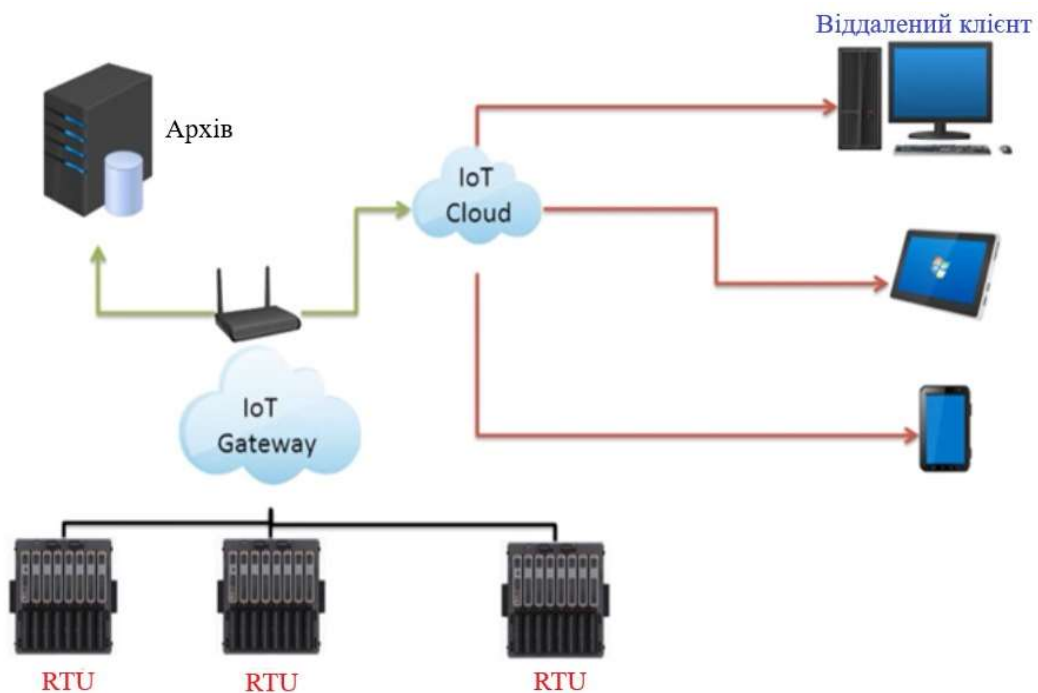


Рисунок 2.7 – SCADA-система на основі інтернету речей

Ця архітектура відкриває нові можливості для промисловості, дозволяючи не тільки стежити за поточними подіями, а й активно керувати виробничими процесами в реальному часі, оптимізувати ресурси та адаптуватися до умов, що змінюється, що стає ключовим фактором у сучасній промисловості.

Вибір SCADA-системи з урахуванням Інтернету речей обґрунтовується рядом переваг. По-перше, такі системи дозволяють централізовано керувати та контролювати різні об'єкти віддалено, що забезпечує високу гнучкість та мобільність в управлінні процесами. По-друге, завдяки інтеграції з Інтернетом речей, SCADA-системи можуть надавати точну та своєчасну інформацію для прийняття рішень на основі даних у реальному часі. Нарешті, використання IoT у SCADA сприяє підвищенню ефективності та надійності системи за рахунок автоматизації багатьох процесів та операцій. Ці переваги роблять вибір SCADA-системи на основі Інтернету речей досить перспективним для сучасних промислових та інфраструктурних завдань.

2.3 Розробка алгоритмічного та програмного забезпечення SCADA-системи гнучкого виробництва

Для досягнення поставленої мети у роботі необхідно розробити загальний алгоритм для системи моніторингу. Для правильної роботи компонентів SCADA-системи мікроконтролер має встановити з'єднання з WI-FI та приєднатися до інформаційного каналу Amazon Web Services.

Оскільки всі необхідні апаратні засоби для правильної роботи включають підключення до WI-FI мережі із доступом до глобальної мережі та джерело живлення з потужністю 5V, мікроконтролер передаватиме всі зібрані дані за допомогою протоколу MQTT.

Як було вказано раніше, MQTT – це зручний, ефективний, компактний та відкритий протокол, спеціально розроблений для обміну даними з IoT

пристроями через технології бездротового зв'язку, що дозволяє працювати з нестабільним або обмеженим з'єднанням на основі моделі з каналами підписки/публікації.

За допомогою цього протоколу, мікроконтролер встановить зв'язок з сервісом AWS IoT Core. Загальна схема реалізації зображена на рисунку 2.8. Дані будуть направлені до спеціалізованого сервісу AWS IoT Core використовуючи топіки IoT Shadow, які будуть отримувати дані з мікроконтролера.

Сервіс AWS IoT Device Shadow додає тіні до речових об'єктів AWS IoT. Тіні можуть зробити стан пристрою доступним для програм та інших служб незалежно від того, підключено пристрій до AWS IoT чи ні. Об'єкти AWS IoT можуть мати кілька іменованих тіней, щоб рішення IoT мало більше можливостей для підключення пристроїв до інших програм і служб.

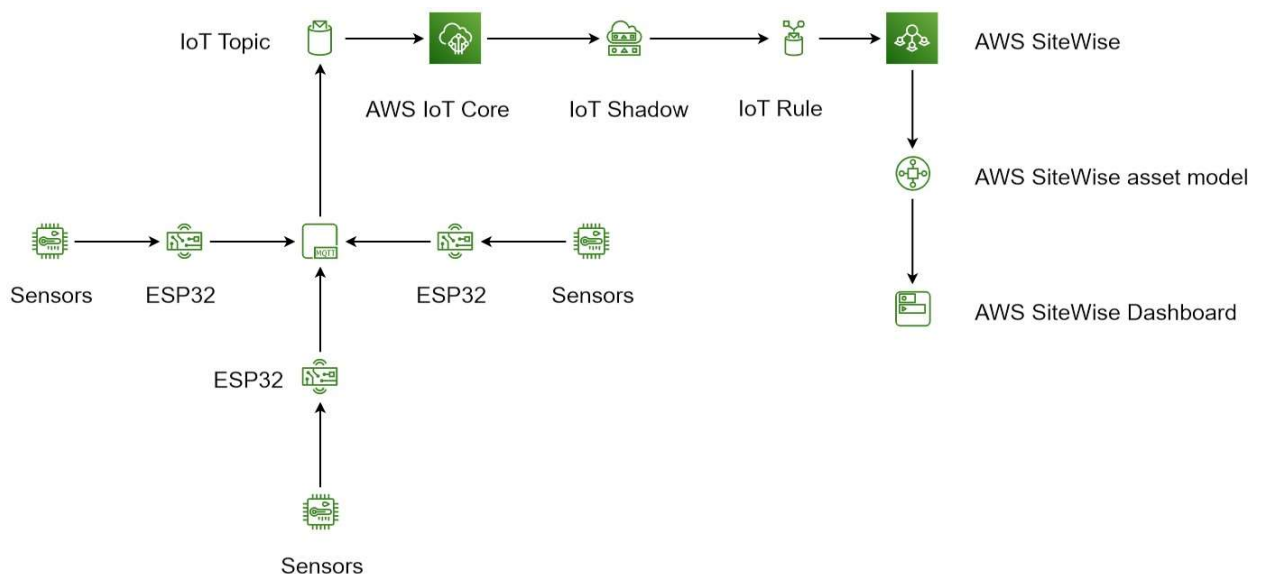


Рисунок 2.8 – Загальна схема реалізації SCADA-системи гнучкого виробництва

Сервіс AWS IoT Core Rule аналізує тіньові сповіщення пристрою та надсилає дані на активи (або асети) пристрою в AWS IoT SiteWise. Кожного разу, коли тінь пристрою оновлюється, AWS IoT надсилає повідомлення MQTT. Правило виконує дії, коли тіні пристрою змінюються на основі

повідомлення MQTT. Повідомлення про оновлення обробляється, щоб отримати значення властивостей і надіслати їх на активи пристрою в AWS IoT SiteWise. Ці дані збираються у модель активів в AWS IoT SiteWise для представлення пристроїв, які передають дані з мікроконтролера. Моделі активів забезпечують узгодженість інформації для кількох активів одного типу, аби зробити можливим обробку даних в активах, які представляють групи пристроїв. На основі даних які зберігаються в активах вже відбувається подальша візуалізація та обробка даних за допомогою AWS IoT SiteWise Dashboard та інших сервісів. На рисунку 2.9 зображений загальний алгоритм роботи SCADA-системи гнучкого виробництва.

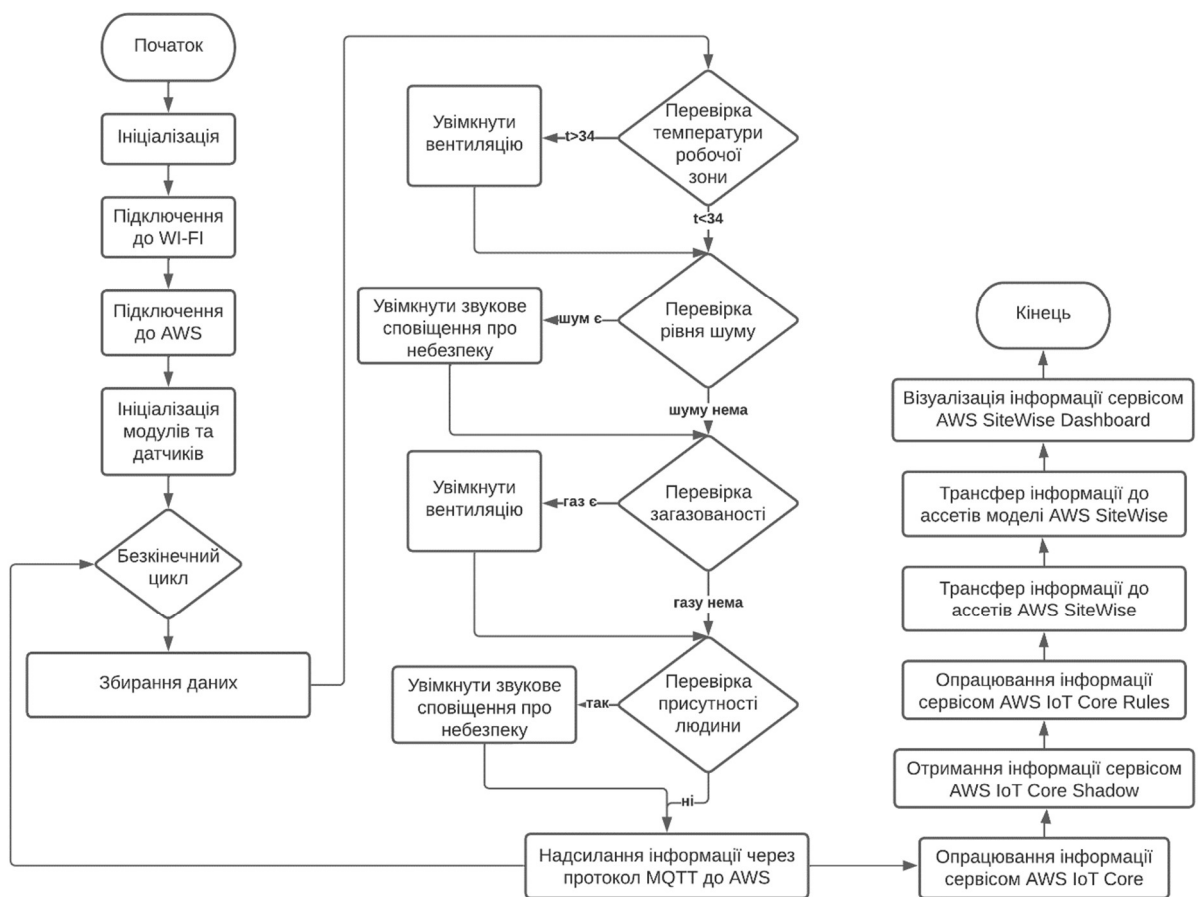


Рисунок 2.9 – Загальний алгоритм роботи SCADA-системи гнучкого виробництва

2.4 Аналіз та вибір модулів моніторингу і керування гнучкого виробництва

При розробці системи, яка збирає дані та може передавати їх дистанційно, для подальшої обробки та аналізу. Для аналізу потрібно розглянути функціональність та вимоги, необхідні для ефективного моніторингу та керування виробництвом, слід також розглянути інтеграцію з уже існуючими системами. Проведемо огляд ринку модулів, дослідивши технічні характеристики.

Розглянемо декілька найбільш вживаних і популярних варіантів основних модулів, таких як ESP32-Cam, Arduino Uno, Arduino Nano та Raspberry Pi, кожен з них має свої унікальні характеристики та застосування.

ESP32 – це мікроконтролер з можливістю підключення до WiFi та Bluetooth, за допомогою якого можна створити системи розумний дім і роботів. Цей модуль має високу потужність, гнучкість та широкі можливості.

Мікроконтролери Arduino Uno та Nano мають дуже схожі властивості між собою. Вони обладнані мікроконтролером ATmega328P і відрізняються низькою ціною та простотою використання. Arduino Uno має стандартний розмір, тоді як Arduino Nano – компактний та підходить для реалізації систем з обмеженим простором.

Також Raspberry Pi є досить розповсюдженим рішенням, це одноплатний комп'ютер, який працює на базі Broadcom BCM2837. Він відрізняється великим об'ємом оперативної пам'яті (1GB), можливістю використання MicroSD Card для зберігання даних та вбудованим USB-портом. Raspberry Pi часто використовується для створення серверів, проектів IoT, а також для вивчення програмування та комп'ютерних наук.

Визначимо найбільш підходящий пристрій для системи керування. ESP32 надає високу функціональність для проектів, пов'язаних з IoT. Arduino Uno та Nano мають значно менший об'єм даних, на відмінно від інших. Тоді як Raspberry Pi насамперед підходить для проектів більшого розміру, де

потрібно велику обчислювальну потужність та можливості, які забезпечує повноцінний одноплатний комп'ютер. Виходячи з цих параметрів буде обрана ESP32, яку можна побачити на рисунку 2.10.



Рисунок 2.10 – Wi-Fi модуль DevKit V1 з ESP-32 [22]

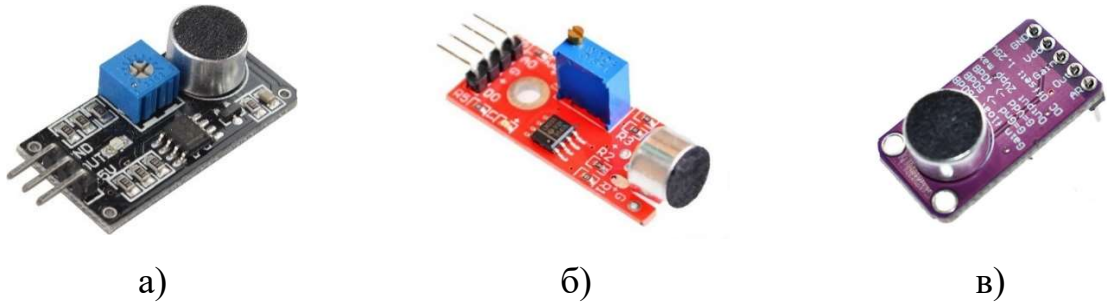
Основні характеристики модуля наведено у таблиці 2.2.

Таблиця 2.2 – Характеристики Wi-Fi модуль DevKit V1 з ESP-32

Характеристика	Значення
Частота бездротової передачі	2,4 ГГц
Стандарт Wi-Fi	802,11 b/g/n.
Стандарт Bluetooth	BLE v4,2 BR/EDR.
Тактова частота	до 240 МГц
Flash-пам'ять	448 кБ
Зовнішня Flash-пам'ять	4 МБ

Також оберемо сенсори для розроблення системи моніторингу. Використаємо модуль для реагування на різні звукові сигнали. Це допоможе для виявлення не природніх звуків у приладдя, критичних ситуацій на виробництві, а також для дотримання стандарту щодо кількості шуму на робочому місці.

Розглянемо декілька найбільш популярних і доступних моделей на ринку. Приклад різних модулів можна побачити на рисунку 2.11. Критерії їх порівняння наведені у таблиці 2.3.



а) HALJA LM393;

б) KY-037;

в) Max9814;

Рисунок 2.11 – Мікрофони

Основні критерії модулів мікрофона що порівнюються, наведено у таблиці 2.3.

Таблиця 2.3 – критерії модулів мікрофона

Критерії	KY-037	Max9814	HALJA LM393
Тип	Аналоговий мікрофон	Підсилювач мікрофону	Мікрофон з компаратором
Чутливість	Висока	Висока	Залежить від моделі
Вихід	Аналоговий сигнал	Аналоговий сигнал	Цифровий сигнал

Як бачимо виходячи з таблиці вище, KY-037 є достатньо простим у використанні, але має обмежений базовий функціонал звукового контролю. В свою чергу, Max9814 через свою потужність вимагатиме додаткове живлення,

а НАЛІА LM393 є менш гнучкою системою за рахунок цифрового виходу. Тому було обрано модуль мікрофона з високою чутливістю KY-037 [23].

Також додамо до системи датчик газу він використовуються для виявлення наявності різних газів в навколишньому повітрі та моніторингу їхніх концентрацій. Ці датчики є важливими для дотримання системи безпеки, промислових процесів та інших. Найбільш популярними є датчики MQ-серії, які мають змогу виявляти різні гази, такі як метан, пропан, дим, аміак, оксид вуглецю та інші. Вони є аналоговими та мають велику чутливість. З представлених на ринку оберемо датчик MQ-2. Приклад якого можна побачити на рисунку 2.12.



Рисунок 2.12 – MQ-2 Датчик газу [24]

Основні характеристики датчик газу наведено у таблиці 2.4.

Таблиця 2.4 – Характеристики датчик газу

Характеристика	Значення
Напруга	5 В
Ток	160 мА
Напруга живлення датчика	3,3–5 В
Розмір	32 мм x 20 мм x 22 мм

Для обмеження доступу людини до небезпечних робочих зон, з метою збереження її життя та здоров'я, встановимо на систему радарний датчик HLK-LD2410C присутності людини. Датчик зображений на рисунку 2.13.



Рисунок 2.13 – Мікрохвильовий радарний датчик HLK-LD2410C [25]

Основні характеристики мікрохвильового радарного датчик наведено у таблиці 2.5.

Таблиця 2.5 – Характеристики мікрохвильового радарного датчик

Характеристика	Значення
Частота	24 ГГц
Дальність виявлення	До 6 м
Напругення	від 5 В до 12 В
Кут виявлення	$\pm 60^\circ$

З метою більш ефективного процесу виробництва та виявлення можливих аномалій в виробничому приміщенні, додаймо барометр. Для виконання даного завдання буде використаний датчик BMP280. Приклад обраного датчика атмосферного тиску можна побачити на рисунку 2.14.

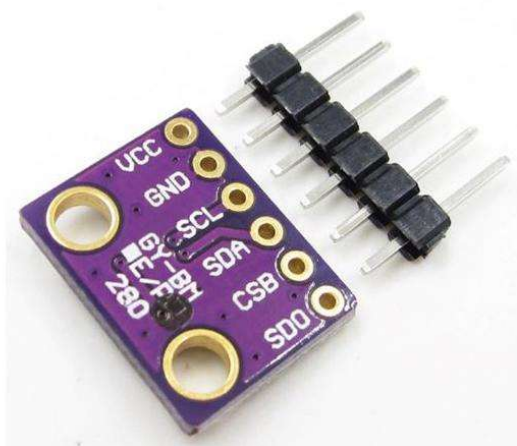


Рисунок 2.14 – Барометр BMP280 [26]

Основні характеристики барометра наведено у таблиці 2.6.

Таблиця 2.6 – Характеристики барометра

Характеристика	Значення
Діапазон тиску	300 Па – 100 ГПа
Середнє енергоспоживання	2,74 нА
Струм	2,7 мкА при частоті відліків в 1 Гц
Напруга живлення	від 1,71 В до 3,6 В
Інтерфейс	I2C, SPI (4 Провід), SPI (3 Провід)
Середній час виміру	5,5 мс

З метою попередження небезпеки на виробництві та попередження працівників додаймо баззер. З його допомогою, при виникненні небезпечної ситуації працівники будуть попереджені, що дозволить вчасно зреагувати та потенційно уникнути небезпеки. Баззер, який буде застосований має назву YNE12-01. Приклад обраного баззера можна побачити на рисунку 2.15.



Рисунок 2.15 – Баззер YHE12-01 [27]

Основні характеристики баззера наведено у таблиці 2.7.

Таблиця 2.7 – Характеристики баззера

Характеристика	Значення
Тип випромінювача	Електромагнітний
Резонансна частота	2,3 кГц
Робоча напруга	1,0 В – 2,0 В
Струм	20 мА
Гучність звуку	75 dB

Вентиляція на виробництві має вирішальне значення для забезпечення безпечних та ефективних умов праці. Основні мети включають у себе забезпечення здоров'я працівників, контроль температури та вологості, вилучення шкідливих речовин і запахів, запобігання конденсації, підтримання пожежної безпеки та оптимізацію роботи обладнання. Вентиляційні системи також впливають на загальну продуктивність та комфорт працівників, забезпечуючи безпечне та здорове робоче середовище. Для симуляції вентиляції буде використаний вентилятор для ноутбука Lenovo 5F10N82225. Приклад вентилятора наведений на рисунку 2.16.



Рисунок 2.16 – Вентилятор Lenovo 5F10N82225 [28]

2.5 Висновки до другого розділу

В цьому розділі магістерської кваліфікаційної роботи було проаналізовано та обрано хмарний сервіс для подальшої розробки SCADA-системи. Також було проаналізовано та обрано архітектуру SCADA-системи. Розроблено алгоритмічне та програмне забезпечення SCADA-системи гнучкого виробництва на базі хмарних сервісів Amazon Web Services. Було розроблено загальний алгоритм роботи SCADA-системи гнучкого виробництва. Також було проаналізовано апаратні засоби моніторингу і керування гнучкого виробництва. Були обрані апаратні модулі для подальшого використання.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПОНЕНТІВ SCADA СИСТЕМИ ДЛЯ ГНУЧКОГО ВИРОБНИЦТВА НА БАЗІ AWS

3.1 Удосконалення методу передачі інформації на базі протоколу MQTT

MQTT (Message Queuing Telemetry Transport) – це протокол передачі повідомлень, який досить часто використовується в системах Інтернету речей (IoT) для обміну даними між пристроями. Протокол працює за принципом публікації-підписки (publish-subscribe), що дозволяє пристроям обмінюватися повідомленнями через брокера. Одним з вагомих недоліків протоколу MQTT є незахищеність з'єднання. Основна версія протоколу MQTT не надає вбудованого захисту. Це означає, що важливо реалізувати додаткові заходи безпеки, такі як шифрування (SSL/TLS) та автентифікація, для захисту від несанкціонованого доступу і перехоплення даних. У цьому контексті, відсутність шифрування та механізмів автентифікації може виправдати серйозні занепокоєння щодо безпеки передачі даних. Дані, які пересилаються через мережу, залишаються вразливими перед можливістю перехоплення зловмисниками. Це може стати серйозною загрозою для конфіденційності та цілісності інформації [29].

Деталізуючи проблеми, виникає ризик несанкціонованого доступу до брокера MQTT, який є центральним об'єктом обміну повідомленнями між пристроями. Відсутність механізмів автентифікації може дозволити неавторизованим користувачам взаємодіяти з системою, порушуючи безпеку та надійність.

Етапи передачі даних основані на протоколі MQTT можна поділити на декілька основних. Першим етапом є ініціалізація клієнта, оскільки кожен пристрій яких хоче взаємодіяти через протокол, має ініціалізувати себе як клієнт. Цей процес так враховує під'єднання до брокера. Далі клієнт, який хоче

надіслати повідомлення має визначити тему свого повідомлення, оскільки в подальшому ця тема буде використовуватись як канал зв'язку для обміну інформацією. Після вже відбувається процес формування повідомлення. Повідомлення клієнта відправляється до брокера, який в свою чергу отримує повідомлення та відправляє його усім клієнтам, які підписані на вказану тему. Якщо третя особа може підписатися на тему інформація якої її цікавить. Після отримання клієнтом інформації він оброблює її згідно власних правил.

Спробуємо описати базові елементи процесу передачі інформації за допомогою математичної моделі. Функція встановлення з'єднання матиме вигляд:

$$f_{(connect)} = C \times B \rightarrow Connack, \quad (3.1)$$

де $f_{(connect)}$ – функція встановлення з'єднання;

C – множина клієнтів;

B – множина брокерів;

T – множина тем;

$Connack$ – код підтверження та додаткова інформація.

Функцію кода підтверження, можна описати як:

$$Connect Request \rightarrow Connack, \quad (3.2)$$

де $Connect Request$ – запит на встановлення з'єднання.

Припустімо, що M – множина повідомлень, тоді відправлення повідомлення має вигляд:

$$f_{(publish)} = C \times T \times M \rightarrow Publish Packet, \quad (3.3)$$

де $f_{(publish)}$ – функція відправки повідомлення;

Publish Packet – містить інформацію про повідомлення та тему.

Функція підтвердження публікації, матиме вигляд:

$$Publish\ Packet \rightarrow Puback\ Packet, \quad (3.4)$$

де *Puback Packet* – пакет підтверження.

Функція підписки на тему, матиме вигляд:

$$f_{(subscribe)} = C \times T \rightarrow Subscribe\ Packet, \quad (3.5)$$

де $f_{(subscribe)}$ – функція підписки на тему;

Subscribe Packet – функція підписки на тему.

Функція підтверження підписки, матиме вигляд:

$$Subscribe\ Packet \rightarrow Suback\ Packet, \quad (3.6)$$

де *Suback Packet* – пакет підтверження підписки;

Функція відключення, має вигляд:

$$f_{(disconnect)} = C \rightarrow Disconnect\ Packet, \quad (3.7)$$

де $f_{(disconnect)}$ – функція відключення;

Disconnect Packet – пакет відключення.

Використання SSL/TLS з'єднання у сучасному інтернет-середовищі, особливо в галузі розробки пристроїв та IoT, має визначені переваги. Зокрема, це надає шифрування даних, забезпечуючи захист конфіденційної інформації від можливого несанкціонованого доступу чи перехоплення. SSL/TLS також

важливі для уникнення атак "людина посередник", забезпечуючи надійний та захищений обмін інформацією між пристроями та серверами.

Спробуємо описати базові елементи процесу додавання сертифікатів та ключів у контексті безпечного з'єднання за допомогою математичної моделі.

Функція встановлення з'єднання матиме вигляд:

Нехай:

CA – кореневий сертифікат центра сертифікації;

$Cent$ – сертифікат сервера;

Key – приватний ключ.

Тоді ми можемо визначити контекст з'єднання Ctx як вектор із трьох компонентами:

$$Ctx = \{CA, Cent, Key\}. \quad (3.8)$$

Процес додавання сертифікатів та ключів можна виразити як функцію $AddCredentials$, яка приймає контекст Ctx та застосовує його до класу $WiFiClientSecure$:

$$AddCredentials(Ctx) \rightarrow WiFiClientSecure. \quad (3.9)$$

Таким чином, функція $AddCredentials$, встановлює кореневий сертифікат, сертифікат сервера та приватний ключ в об'єкт $WiFiClientSecure$, готовий до використання для з'єднання.

Математично це можна представити як:

$$WiFiClientSecure = AddCredentials(CA, Cent, Key), \quad (3.10)$$

де $WiFiClientSecure$ – об'єкт, готовий до використання для безпечного з'єднання, CA , $Cent$ та Key – відповідні сертифікати та ключі.

Цей процес забезпечить правильну аутентифікацію клієнта та сервера під час безпечної комунікації через TSL/SSL.

3.2 Розробка програмного забезпечення SCADA-системи

При створенні програмного керування для збору даних з ESP32 слід враховувати ключові аспекти майбутньої системи, такі як мінімізація споживання електроенергії та обсягу трафіку у WI-FI мережі, а також використання легковагих протоколів передачі даних. Для досягнення цієї мети планується використовувати енергоефективні компоненти на апаратному рівні та спеціалізований протокол MQTT.

Для реалізації з'єднання ESP32 та AWS сервісів за допомогою MQTT протокола ми будемо використовувати бібліотеку під назвою PubSubClient. Вона дозволяє надсилати дані в топіки та підписуватись на них. Таким чином, зібрані дані будуть сформовані у StaticJsonDocument та надсилатися з ESP32 до обраного топіку створеного за допомогою сервіса AWS IoT Core. На рисунку 3.1 проілюстровано створення функції формування json документа і публікації.

```
98 void publishMessage ()
99 {
100   StaticJsonDocument<200> doc;
101
102   JsonObject state_reported = doc["state"].createNestedObject("reported");
103   state_reported["pressure"] = pressure;
104   state_reported["temperature"] = temperature;
105   state_reported["approx_altitude"] = altutude;
106   state_reported["timestamp"] = seconds;
107   char jsonBuffer[512];
108   serializeJson(doc, jsonBuffer); // print to client
109
110   client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer);
111 }
```

Рисунок 3.1 – Функція формування json документа і публікації


```

119 Serial.println(F("BMP280 test"));
120 unsigned status;
121 //status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
122 status = bmp.begin(0x76);
123 if (!status) {
124     Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
125         "try a different address!"));
126     Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(), 16);
127     Serial.print("      ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");
128     Serial.print("      ID of 0x56-0x58 represents a BMP 280,\n");
129     Serial.print("      ID of 0x60 represents a BME 280.\n");
130     Serial.print("      ID of 0x61 represents a BME 680.\n");
131     while (1) delay(10);
132 }
133
134 /* Default settings from datasheet. */
135 bmp.setSampling(Adafruit_BMP280::MODE_NORMAL, /* Operating Mode. */
136     Adafruit_BMP280::SAMPLING_X2, /* Temp. oversampling */
137     Adafruit_BMP280::SAMPLING_X16, /* Pressure oversampling */
138     Adafruit_BMP280::FILTER_X16, /* Filtering. */
139     Adafruit_BMP280::STANDBY_MS_500); /* Standby time. */
140

```

Рисунок 3.3 – Ініціалізація датчика BMP280

Подальше використання датчика буде відбуватися як проілюстровано на рисунку 3.4.

```

Serial.print(F("Temperature = "));
Serial.print(bmp.readTemperature());
Serial.println(" *C");
temperature = bmp.readTemperature();
Serial.print(F("Pressure = "));
Serial.print(bmp.readPressure());
Serial.println(" Pa");
pressure = bmp.readPressure();
Serial.print(F("Approx altitude = "));
Serial.print(bmp.readAltitude(1013.25)); /* Adjusted to local forecast! */
Serial.println(" m");
altitude = bmp.readAltitude(1013.25);
Serial.println();

```

Рисунок 3.4 – Використання датчика BMP280

Для розпізнавання небезпеки пов'язаної з горючими газами, був використаний датчик MQ2. Він відправляє логічну одиницю на підключений цифровий вихід у випадку, якщо виявляє газ. Нам необхідно реалізувати

зчитування цього піна та перевірку його стану. У випадку виявлення газу вмикається вентиляція. Реалізація зображена на рисунку 3.5

```

Serial.println();
int gasState = digitalRead(DO_PIN);

if (gasState == HIGH) {
    //    digitalWrite(soundinp, HIGH); //потом вернуть
    digitalWrite(soundinp, LOW);
    ledcWrite(1, 0);
    Serial.println("The gas is NOT present");
}
else {
    Serial.println("The gas is present");
    ledcWrite(1, MAX_DC);
}

```

Рисунок 3.5 – Зчитування показань датчика MQ2

Датчики HLK-LD2410C та KY-037, баззер та вентилятор працюють за тим самим принципом. Код реалізації виявлення людини у робочій зоні зображений на рисунку 3.6.

```

Serial.println();
detectState = digitalRead(detectin);

if (detectState == HIGH) {
    digitalWrite(soundinp, HIGH);
//    digitalWrite(soundinp, LOW);
    Serial.printf("Someone detected in sight");
} else {
    digitalWrite(soundinp, LOW) ;
    detectState = 0;
    digitalWrite(detectState, LOW) ;
    Serial.printf("No one detected in sight");
}

```

Рисунок 3.6 – Реалізація виявлення людини у робочій зоні

3.3 Розробка макета модуля для збирання технологічних параметрів

Схема підключення до мікроконтролера є важливою, оскільки вона визначає правильний спосіб інтеграції компонентів у систему. Цей підхід дозволяє забезпечити ефективну взаємодію між різними елементами, забезпечуючи стабільну та надійну роботу. Схема підключення визначає порядок взаємодії, електричні параметри та передачу сигналів, що є важливим для досягнення оптимального функціонування системи в цілому.

Крім того, схема підключення до мікроконтролера сприяє ефективній управлінській стратегії, оскільки вона визначає, як кожен компонент взаємодіє з мікроконтролером, які дані передаються та як вони обробляються. Це є ключовим аспектом в розробці електронних систем, де правильна організація взаємозв'язків між компонентами може суттєво впливати на функціональність, надійність та продуктивність системи в цілому.

Додатково, схема підключення допомагає в ідентифікації можливих проблем та ускладнень на етапі розробки, дозволяючи заздалегідь вирішити питання з сумісністю, електричними параметрами та іншими аспектами інтеграції. Це важливо для забезпечення ефективної роботи системи в реальних умовах експлуатації.

Під час складання макету були використанні всі модулі та деталі наведенні вище. Схема підключення наведена на рисунку 3.7.

На схемі підключення можна побачити схему підключення датчиків та інших модулів до мікроконтролера ESP32. Живлення мікроконтролера відбувається шляхом підключення microUSB розташованого на платі ESP32 до зовнішнього джерела живлення 5 В. Датчики та інші модулі окрім BMP280 беруть живлення зі спільної лінії живлення, що під'єднанні до виходів 5 В та GND. BMP280 використовує лінію живлення 3,3 В. Також важливо правильно обрати цифрові піни для коректної роботи.

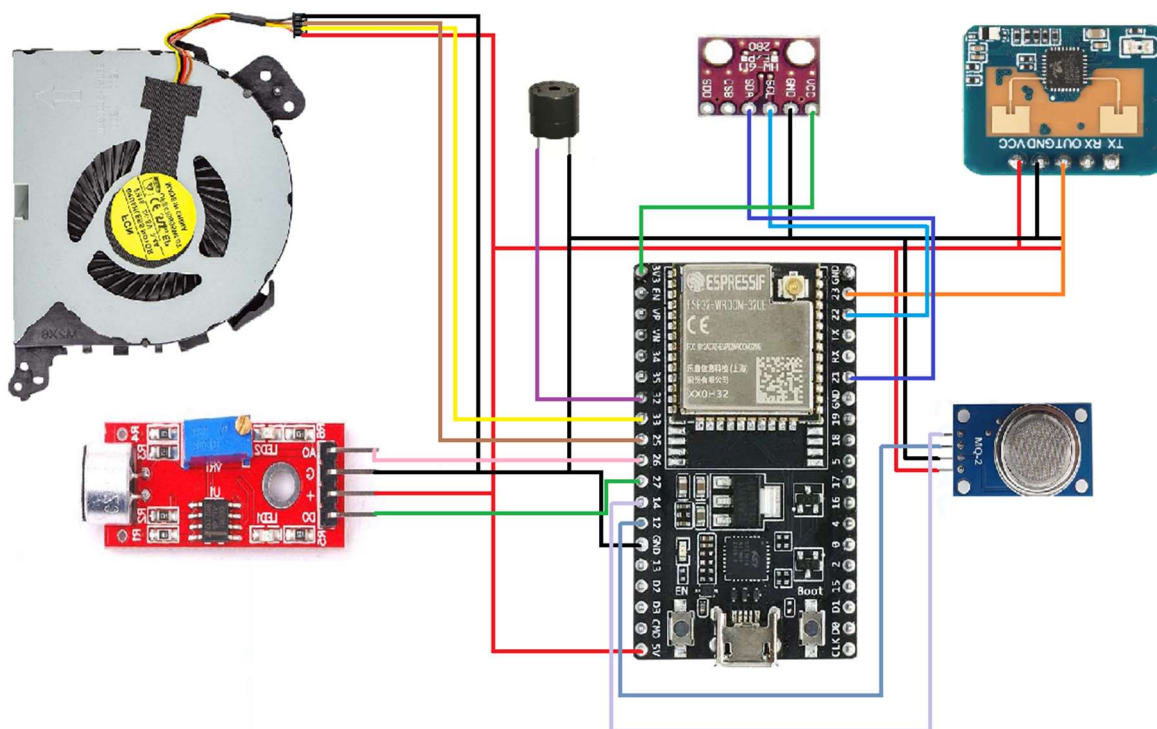


Рисунок 3.7 – Схема підключення

За основу був взятий мікроконтролер ESP32. Для зручності збірки була використана макетна плата, зображена на рисунку 3.8.

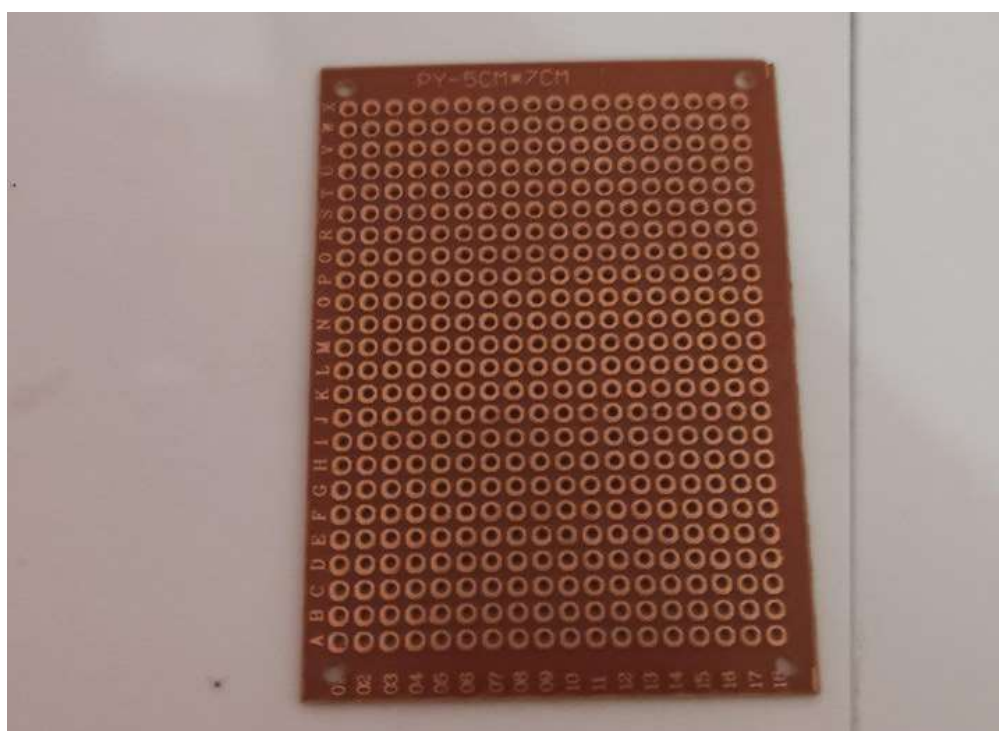


Рисунок 3.8 – Макетна плата

Фінальний вигляд макета зображений на рисунку 3.9.

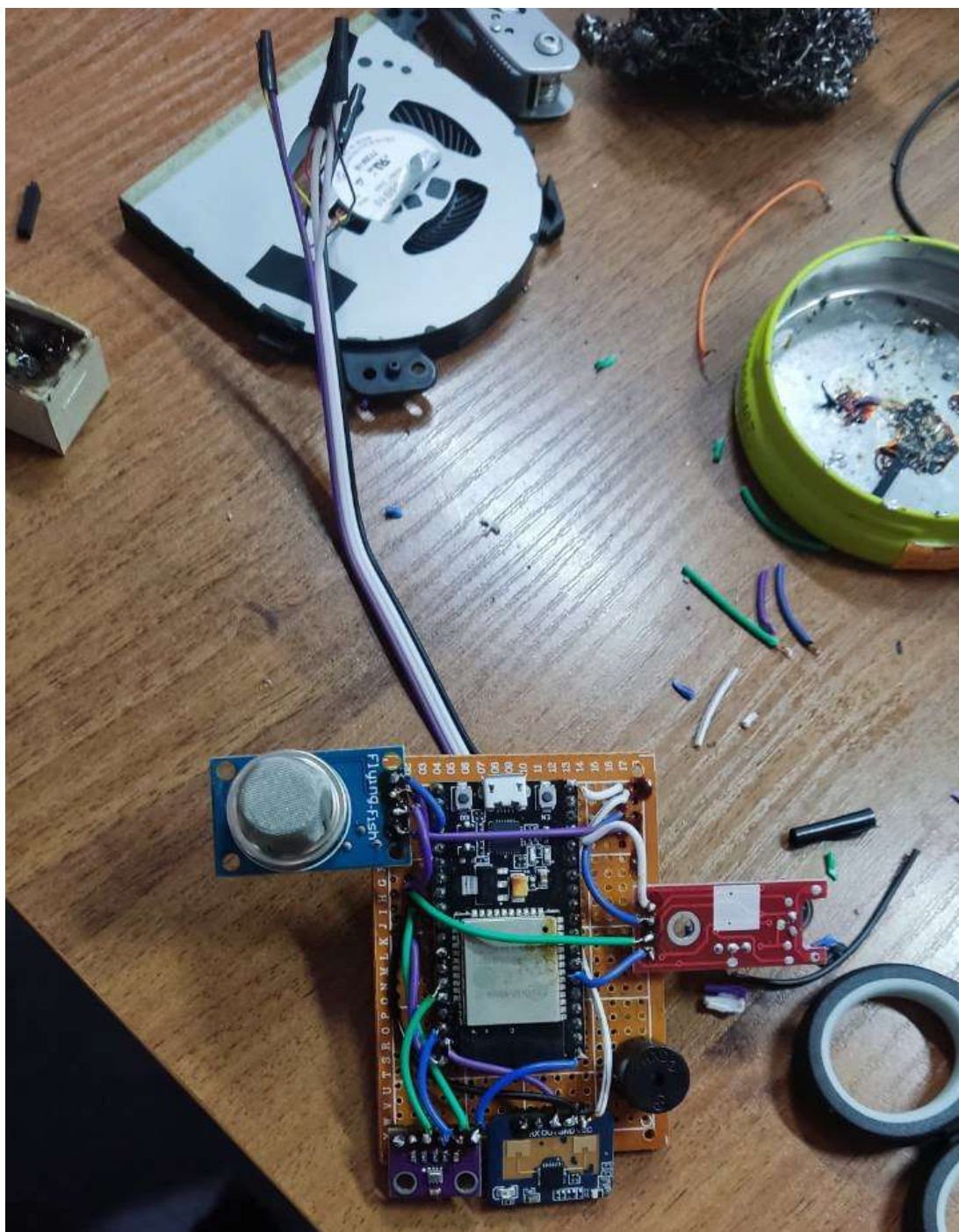


Рисунок 3.9 – Фінальний вигляд макета

Як ми можемо бачити, на рисунку 3.9 до мікроконтролера ESP32 підключено декілька датчиків та модулів, а саме: вентилятор Lenovo 5F10N82225, барометр BMP280, датчик присутності людини HLK-LD2410C, датчик газу MQ2, мікрофон KY-037.

3.4 Висновки до третього розділу

В цьому розділі магістерської кваліфікаційної роботи було удосконалено метод передачі інформації на базі протоколу MQTT. Також було розроблено програмне забезпечення SCADA-системи для взаємодії з хмарними сервісами та апаратними модулями. Було розроблено схему підключення та зібрано макет модуля для збирання технологічних параметрів з модулів обраних раніше.

4 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Постановка завдань експерименту

Постановка задач експерименту перевірки працеспроможності макету компонентів SCADA системи для гнучкого виробництва на базі AWS. Методи проведення експерименту: практичний експеримент на базі розробленого макету та методи порівняльного аналізу.

Очікуванні результати практичних досліджень – це візуалізація даних у хмарних компонентах SCADA системи які були отримані за допомогою апаратних датчиків.

Для досягнення мети експерименту необхідно виконати наступні завдання:

- розробити алгоритмічне та програмне забезпечення SCADA-системи гнучкого виробництва;
- удосконалити метод передачі інформації на базі протоколу MQTT;
- розробити макет модуля для збирання технологічних параметрів;
- забезпечити живлення 5 В;
- забезпечити відеофіксацію проведення процесу досліджень.

4.2 Проведення експерименту

Під час розробки лабораторного макету відповідно до технічного завдання виникла потреба в перевірці його працездатності. Для цього було виконано серію експериментів з метою глобальної перевірки функціональності. З метою експерименту лабораторний макет було підключено до Wi-Fi мережі та живлення 5 В. Під час проведення експериментів було важливо здійснювати перевірку працездатності макету на кожному етапі роботи.

Після приєднання макету до джерела живлення 5 В, через серійний порт макет повідомляє про успішне підключення до WI-FI та сервісів AWS IoT Core. Крім того, через серійний порт він надає текстове відображення даних, зібраних з датчиків BMP280, MQ2, KY-037 та HLK-LD2410C в режимі реального часу. Це ілюструється на рисунку 4.1. Датчик BMP280 збирає дані про температуру, тиск та вологість. Датчик MQ2 перевіряє приміщення на наявність горючих газів. Датчик KY-037 перевіряє приміщення на рівень шуму. Датчик HLK-LD2410C перевіряє приміщення на присутність людини.

```

COM8
08:20:01.484 -> BMP280 test
08:20:01.716 -> [D][WiFiGeneric.cpp:374] _eventCallback(): Event: 0 - WIFI_READY
08:20:01.716 -> [D][WiFiGeneric.cpp:374] _eventCallback(): Event: 2 - STA_START
08:20:01.763 -> Connecting to Wi-Fi
08:20:02.224 -> ....[D][WiFiGeneric.cpp:374] _eventCallback(): Event: 4 - STA_CONNECTED
08:20:03.986 -> [D][WiFiGeneric.cpp:374] _eventCallback(): Event: 7 - STA_GOT_IP
08:20:03.986 -> [D][WiFiGeneric.cpp:419] _eventCallback(): STA IP: 192.168.50.239, MASK:
08:20:04.216 -> .Connecting to AWS IOT
08:20:06.863 -> [D][ssl_client.cpp:225] start_ssl_client(): Protocol is TLSv1.2 Ciphersui
08:20:06.863 -> [D][ssl_client.cpp:227] start_ssl_client(): Record expansion is 29
08:20:07.097 -> AWS IoT Connected!
08:20:07.097 -> Waiting for NTP time sync: .
08:20:07.563 -> Current time: Mon Jan 22 06:20:07 2024
08:20:07.623 -> Sound is fine
08:20:07.623 -> Someone detected in sight
08:20:07.623 -> The gas is NOT present
08:20:07.623 -> Timestamp 1705904407
08:20:07.623 -> Temperature = 27.75 *C
08:20:07.623 -> Pressure = 101796.23 Pa
08:20:07.623 -> Approx altitude = -39.16 m
08:20:07.623 ->
Autoscroll Show timestamp Newline 115200 baud Clear output

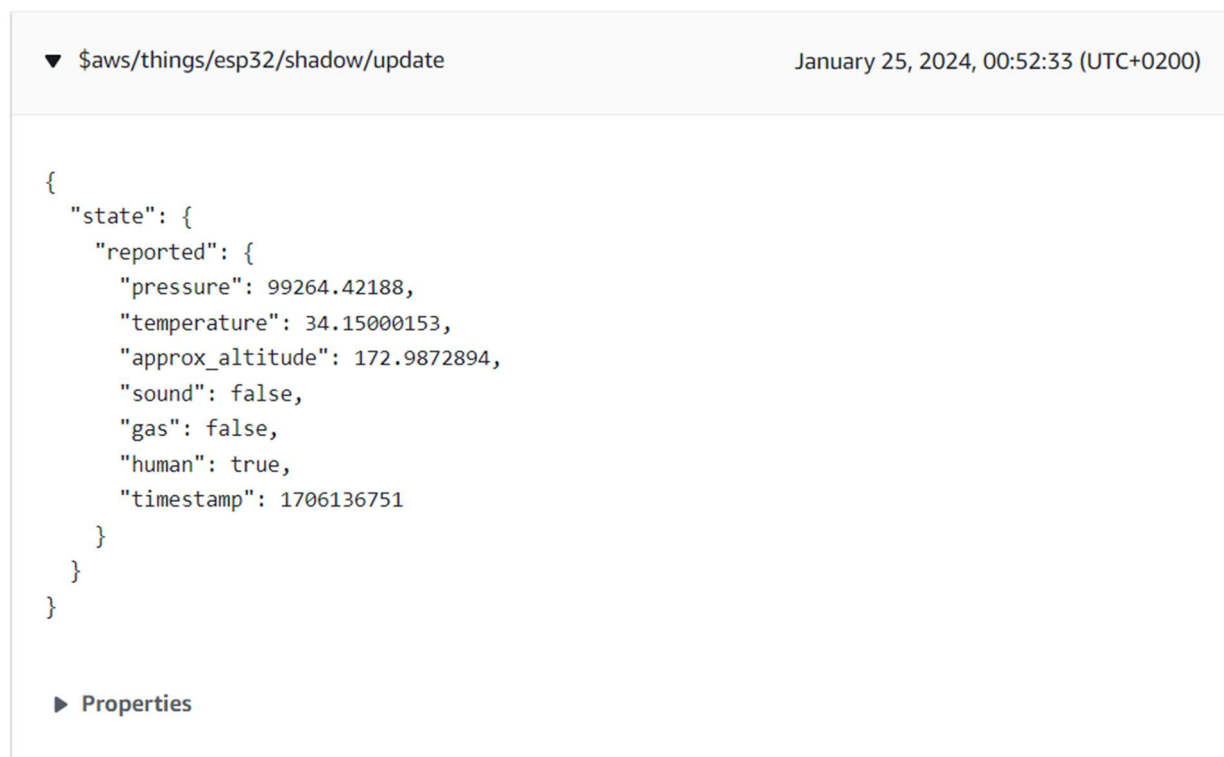
```

Рисунок 4.1 – Дані з серійного монітору макета

Як ми можемо побачити на рисунку 4.1, рівень шуму не перевищений, людина помічена у полі зору, рівень загазованості не перевищений, температура приміщення складає 27,78 °C, атмосферний тиск складає 101796,23 Па, приблизна висота від рівня моря -39,16 м.

Також необхідно перевірити коректність відправки даних до хмарних сервісів. Для цього необхідно увійти у використовуваний акаунт AWS та

перевірити тестовий MQTT клієнт у сервісі AWS IoT Core. Повідомлення отримані у тестовому клієнті AWS IoT Core зображені на рисунку 4.2.



```
▼ $aws/things/esp32/shadow/update January 25, 2024, 00:52:33 (UTC+0200)

{
  "state": {
    "reported": {
      "pressure": 99264.42188,
      "temperature": 34.15000153,
      "approx_altitude": 172.9872894,
      "sound": false,
      "gas": false,
      "human": true,
      "timestamp": 1706136751
    }
  }
}
```

► Properties

Рисунок 4.2 – Дані з тестового MQTT клієнта у сервісі AWS IoT Core

Дані, отримані в сервісі AWS IoT Core передаються до сервісу AWS SiteWise за допомогою AWS IoT Core Rule. Базова конфігурація правила для трансферу даних зображена на рисунку 4.3. Як можна побачити з рисунку 4.3, джерелом даних є топик «\$aws/things/esp32/shadow/update/accepted». Дані передаються до моделі асетів сервісу AWS SiteWise де сортуються за відповідністю до параметрів. Параметри моделі асетів, за якими сортуються дані приведено на рисунку 4.4.

The screenshot displays the configuration page for a rule named 'esp32_to_sitewise' in the AWS IoT SiteWise console. The page is divided into several sections:

- Details:** Shows the rule's description (empty), ARN (arn:aws:iot:us-east-1:382866676528:rule/esp32_to_sitewise), Topic (empty), Status (Active), Basic ingest topic (\$aws/rules/esp32_to_sitewise), and Created date (January 14, 2024, 22:38:37 (UTC+02:00)).
- SQL statement:** Shows the SQL query: `SELECT * FROM '$aws/things/+ /shadow/update/accepted' WHERE startswith(topic(3), 'esp32')` with SQL version 2016-03-23.
- Actions:** Shows one action: 'Send a message data to asset properties in AWS IoT SiteWise'.

Рисунок 4.3 – Базова конфігурація правила для трансферу даних

The screenshot displays the 'Measurements' section of an asset model in the AWS IoT SiteWise console. The left sidebar shows navigation options: Attributes (0), Measurements (6) (selected), Transforms (0), Metrics (2), and Components (0). The main area shows a table of measurements:

Name	ID	External ID	Alias	Unit	MQTT Notification status
Approx altitude	c8a15d93-f63f-45f8-9814-5795b5f2c73a	-	/test/device/esp32/altitude	meters	Inactive
Gas is present	8be1a09d-b1f4-4b47-80a2-2881088a83a2	-	/test/device/esp32/gas	Bool	Inactive
Human detected	019af3c0-f127-499f-ad40-c3e238723620	-	/test/device/esp32/human	Bool	Inactive
Pressure	207fb480-9404-4d14-a4c5-3e07616cf9ad	-	/test/device/esp32/pressure	Pascale	Inactive
	0cfdcb16-				

Рисунок 4.4 – Параметри моделі асетів, за якими сортуються дані

На основі цих параметрів сконфігурований AWS IoT SiteWise Dashboard, який візуалізує дані у вигляді графіків, приведених на рисунку 4.5. Дані можуть бути відфільтровані за проміжками часу.

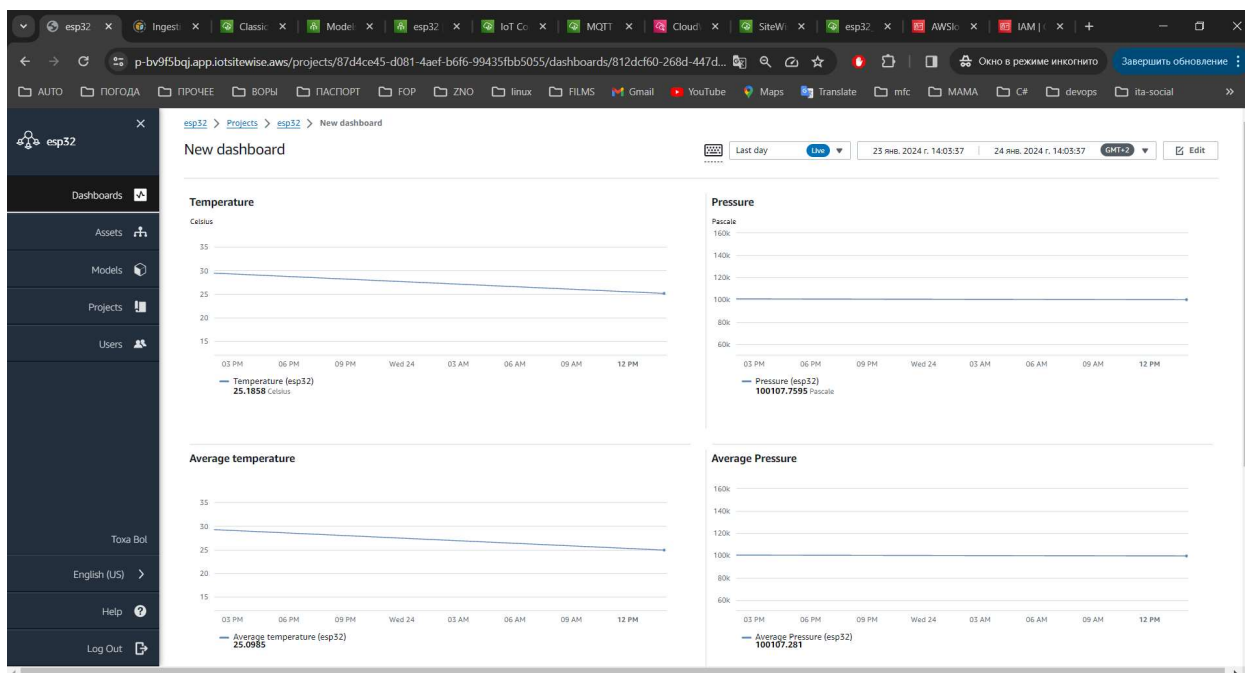


Рисунок 4.5 – AWS IoT SiteWise Dashboard

4.3 Аналіз отриманих результатів

Як ми можемо побачити з рисунка 4.6, дані надіслані апаратною частиною макета були успішно трансформовані та візуалізовані хмарною частиною за допомогою таких сервісів як AWS IoT Core та AWS IoT SiteWise.

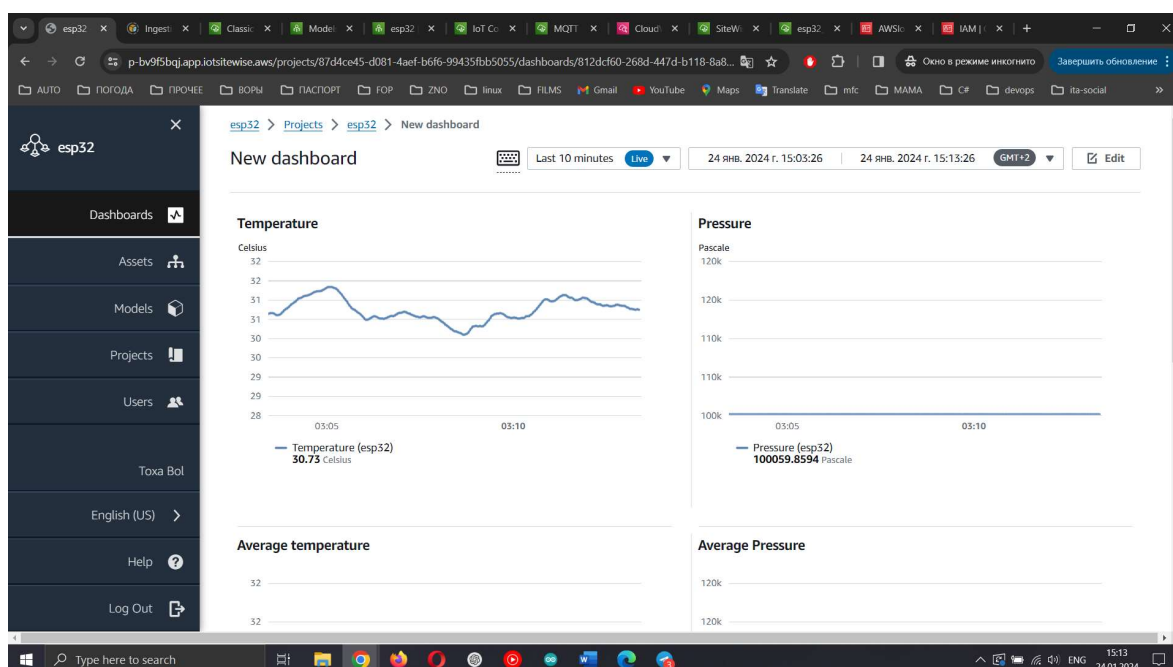


Рисунок 4.6 – Фінальний вигляд макета

Дані можна представити як за абсолютними проміжками часу, так і за відносними. Відносне представлення даних за останні 10 хвилин наведено на рисунку 4.7. Представлення даних за абсолютний проміжок часу з 24.01.2024 14:00 по 24.01.2024 13:00 наведено на рисунку 4.8.

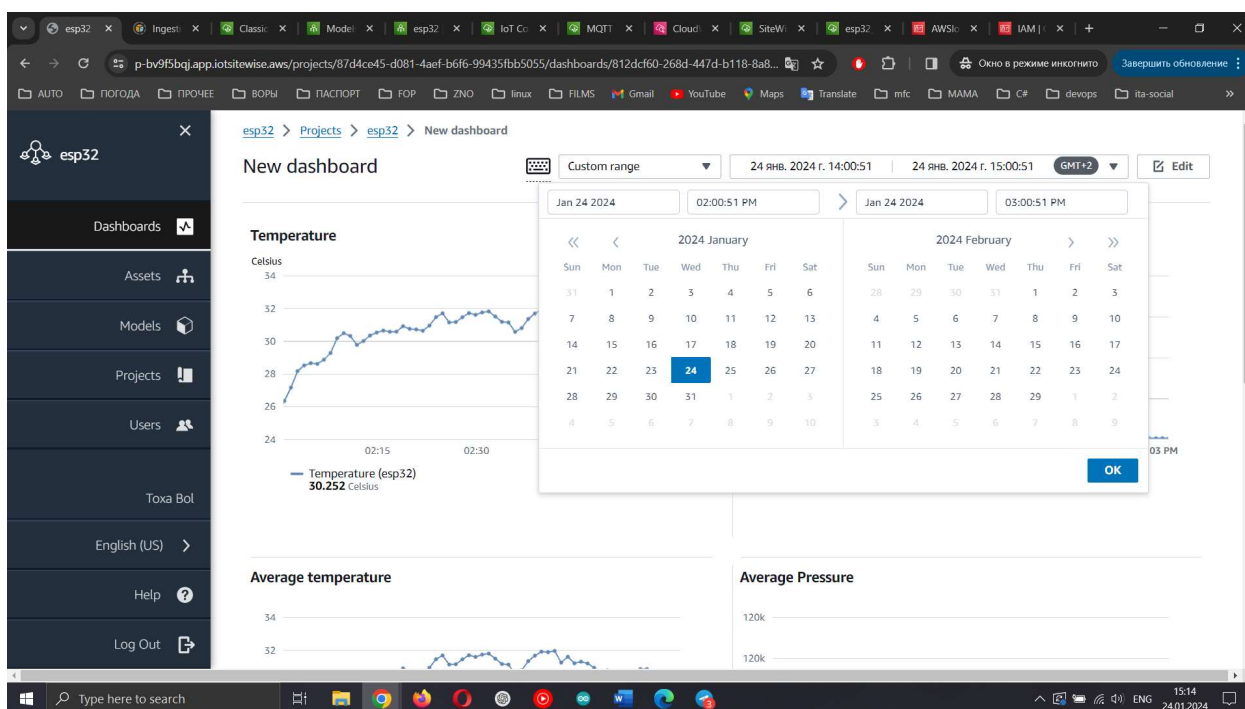


Рисунок 4.7 – Відносне представлення даних за останні 10 хвилин



Рисунок 4.8 – Представлення даних за абсолютний проміжок часу

Усі метрики які були задані у асетах моделі присутні на Dashboard та отримують, візуалізують та оновлюють дані.

Також, під час експерименту було перевірено, що усі датчики які використані для контролю безпеки працюють у штатному режимі. Як наведено на рисунку 4.9, кожен датчик виводить свій поточний стан у Serial Monitor. Дані з датчиків про небезпеку у робочій зоні також передаються до AWS IoT SiteWise, як наведено на рисунку 4.10.

```

15:34:24.564 -> Sound is fine
15:34:24.564 -> Someone detected in sight
15:34:24.564 -> The gas is NOT present
15:34:24.564 -> Timestamp 1706103264
15:34:24.564 -> Temperature = 30.89 *C
15:34:24.564 -> Pressure = 100032.91 Pa
15:34:24.564 -> Approx altitude = 108.13 m
15:34:24.564 ->

```

Рисунок 4.9 – Стан датчиків у Serial Monitor

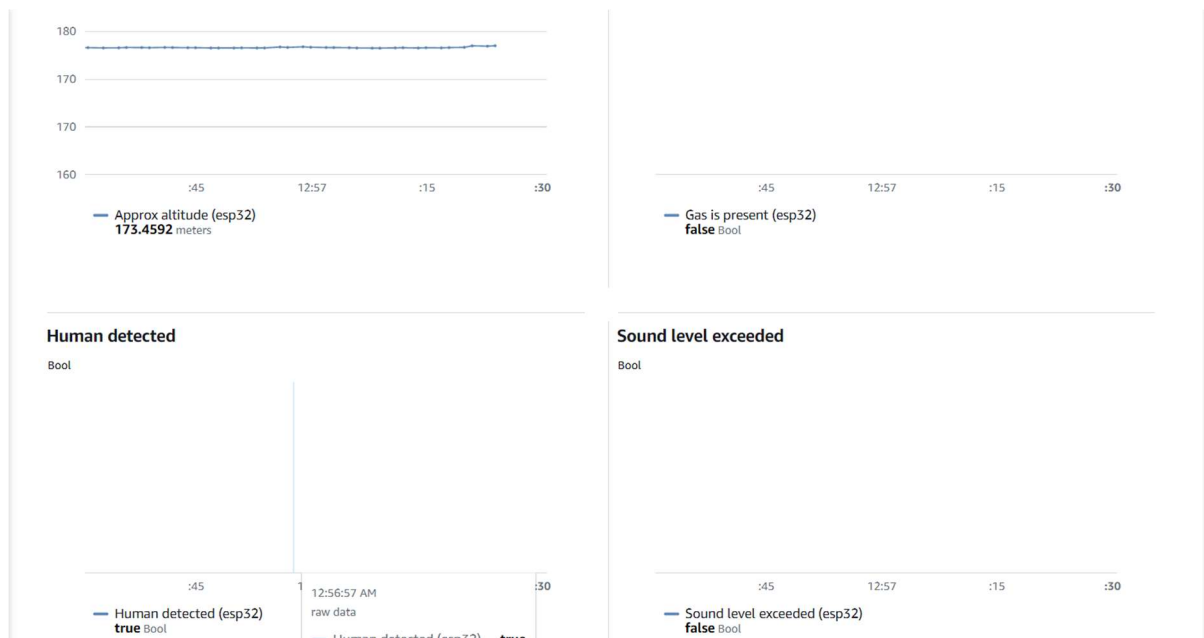


Рисунок 4.10 – Дані з датчиків про небезпеку у робочій зоні

4.4 Охорона праці

В даний час експлуатація переважної більшості технологічного обладнання, енергетичних установок неминуче пов'язана з виникненням шумів та вібрацією різної частоти та інтенсивності, що надають несприятливий вплив на організм людини. Тривалий вплив шуму та вібрації знижує працездатність, може призвести до розвитку професійних захворювань.

Шум, як гігієнічний фактор, являє собою сукупність звуків, що несприятливо впливають на організм людини, що заважають його роботі та відпочинку. Шум являє собою хвилеподібні коливальні рухи частинок пружного (газового, рідкого або твердого) середовища. Зазвичай шум є поєднанням звуків різної частоти та інтенсивності.

Інтенсивний шум при щоденному впливі призводить до виникнення професійного захворювання – приглухуватості, основним симптомом якого є поступова втрата слуху на обидва вуха, що спочатку лежить в області високих частот (4000 Гц), з подальшим поширенням на нижчі частоти, що визначають здатність сприймати мовлення. При дуже великому звуковому тиску може статися розрив барабанної перетинки [30].

Сумарний рівень звукового тиску від декількох джерел шуму визначається за формулою:

$$L = 10 \lg \sum_{i=1}^n 10^{0,1L_i}, \quad (4.1)$$

де L_i – октавний рівень звукового тиску розглядуваного джерела, дБ;

i – номер джерела;

n – загальна кількість джерел в приміщенні.

В разі n однакових джерел шуму формула має вигляд:

$$L = L_i + 10 \lg n, \quad (4.2)$$

де L_i – октавний рівень звукового тиску одного джерела;

n – кількість джерел.

Якщо кількість джерел $n > 2$, то користуючись таблицею визначення рівня шуму необхідно послідовно додавати рівні, починаючи, починаючи із максимального. Спочатку визначають різницю двох додаваних рівнів, потім – добавку до більш високого з додаваних рівнів. Після цього добавку слід додати до більшого з додаваних рівнів.

Знайдемо сумарний рівень звукового тиску від даних п'яти джерел за формулою:

$$L = 10 \lg(10^{0,181} + 10^{0,183} + 10^{0,185} + 10^{0,187} + 10^{0,190} + 10^{0,191}) = 95,3 \text{ дБ}. \quad (4.3)$$

Отримані результати, після порівнянням зі стандартом рівня шуму є задовільними та відповідають умовам праці.

4.5 Висновки до четвертого розділу

В четвертому розділі було проведено експеримент якій підтверджував працеспроможності макета модуля для збирання технологічних параметрів та компонентів SCADA-системи гнучкого виробництва. Було перевірено наявність даних з апаратних модулів за допомогою серійного монітора. Дані були успішно передані до хмарного сервісу AWS IoT Core, успішно трансформовані та надіслані до сервісу AWS IoT SiteWise. За допомогою сервісу AWS IoT SiteWise вони були оброблені та візуалізовані у вигляді графіків. Виходячи з експерименту, апаратний макет модуля для збирання технологічних параметрів та компоненти SCADA-системи гнучкого виробництва працюють у штатному режимі.

ВИСНОВКИ

В магістерській кваліфікаційній роботі було проведено аналіз існуючих технологій розробки SCADA-систем, а саме проаналізовано принципи організації SCADA-систем, програмні технології розробки SCADA-систем, використання хмарних-сервісів в SCADA-системах. Базуючись на цьому аналізі було обрано деякі компоненти SCADA-системи, визначено актуальність даної теми, мету роботи, об'єкт розробки, предмет розробки та поставлені задачі, які необхідно виконати для досягнення поставленої мети.

В другому розділі була розроблена архітектура SCADA-системи гнучкого виробництва на основі AWS. Було розроблено алгоритмічне та програмне забезпечення SCADA-системи гнучкого виробництва на базі хмарних сервісів Amazon Web Services. Було розроблено загальний алгоритм роботи SCADA-системи гнучкого виробництва. Також було проаналізовано апаратні засоби моніторингу і керування гнучкого виробництва. Були обрані апаратні модулі для подальшого використання.

У третьому розділі було удосконалено метод передачі інформації на базі протоколу MQTT. Також було розроблено програмне забезпечення SCADA-системи для взаємодії з хмарними сервісами та апаратними модулями. Було розроблено схему підключення та зібрано макет модуля для збирання технологічних параметрів з модулів обраних раніше.

Наступним етапом було проведення експерименту, для перевірки працеспроможності макета модуля для збирання технологічних параметрів та компонентів SCADA-системи гнучкого виробництва. Було перевірено наявність даних з апаратних модулів за допомогою серійного монітора. Дані були успішно передані до хмарного сервісу AWS IoT Core, успішно трансформовані та надіслані до сервісу AWS IoT SiteWise. За допомогою сервісу AWS IoT SiteWise вони були оброблені та візуалізовані у вигляді

графіків. Виходячи з експерименту, апаратний макет модуля для збирання технологічних параметрів та компоненти SCADA-системи гнучкого виробництва працюють у штатному режимі. Дані були передані за допомогою протокола MQTT із додаванням технології SSL/TLS для забезпечення безпечного з'єднання та цілісності даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. – Харків: ХНУРЕ, 2021. – 55 с
2. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
3. Большаков А.А. Розроблення архітектури SCADA-системи гнучкого виробництва та вибір апаратних засобів / А.А. Большаков// «Development of flexible manufacturing scada system architecture and selection of hardware» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National University of Radio Electronics [electronic edition], 2023. – Part 2. – P. 239-243.
4. Daneels, A.; Salter, W. What is SCADA?// In Proceedings of the International Conference of Accelerator and Large Experimental Physics Control Systems. – 1999. – P. 339–343.
5. Pliatsios D. et al. A survey on SCADA systems: secure protocols, incidents, threats and tactics //IEEE Communications Surveys & Tutorials. – 2020. – Т. 22. – №. 3. – С. 1942-1976.
6. Yadav G., Paul K. Architecture and security of SCADA systems: A review //International Journal of Critical Infrastructure Protection. – 2021. – Т. 34. – С. 100433.

7. Qays M. O. et al. Monitoring of renewable energy systems by IoT-aided SCADA system //Energy Science & Engineering. – 2022. – Т. 10. – №. 6. – С. 1874-1885.
8. Babayigit B., Abubaker M. Industrial Internet of Things: A Review of Improvements Over Traditional SCADA Systems for Industrial Automation //IEEE Systems Journal. – 2023.
9. Osman F. A., Hashem M. Y. M., Eltokhy M. A. R. Secured cloud SCADA system implementation for industrial applications //Multimedia Tools and Applications. – 2022. – Т. 81. – №. 7. – С. 9989-10005.
10. Sazid M. M. et al. Developing a Low-cost SCADA System for Industrial Application //2022 6th International Conference on Trends in Electronics and Informatics (ICOEI). – IEEE, 2022. – С. 227-232.
11. DUYMAZLAR O., ENGIN D. Design, application and analysis of an OPC-based SCADA system //Politeknik Dergisi. – 2023. – С. 1-1.
12. Sarvaiya H. B. et al. Recent Developments in SCADA System for Remote Industry and It's Experimental Implementation //International Journal of Engineering Technology and Management Sciences Website: ijetms. in. – 2022. – Т. 6. – №. 3.
13. What is SCADA 4.0 // Сайт Teamuis. URL: <https://teamuis.com/services/what-is-scada-4-0/> (дата звернення: 23.10.2023).
14. Fazlollahtabar H. Internet of Things-based SCADA system for configuring/reconfiguring an autonomous assembly process //Robotica. – 2022. – Т. 40. – №. 3. – С. 672-689.
15. Bolshakov A. USE OF MICROCONTROLLERS WITH CLOUD SERVICES FOR PRODUCTION NEEDS / Yevsieiev V., Bolshakov A. // «Automation and Development of Electronic Devices» ADED-2022 : Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National University of Radio Electronics [electronic edition], 2022. – Part 2. – P. 89-90.

16. Connecting to AWS IoT Core // Сайт Amazon Web Services.
URL: <https://docs.aws.amazon.com/iot/latest/developerguide/connect-to-iot.html>
(дата звернення: 23.10.2023).
17. Bello S. A. et al. Cloud computing in construction industry: Use cases, benefits and challenges //Automation in Construction. – 2021. – Т. 122. – С. 103441.
18. Sverko M., Grbac T. G., Mikuc M. Scada systems with focus on continuous manufacturing and steel industry: A survey on architectures, standards, challenges and industry 5.0 //IEEE Access. – 2022. – Т. 10. – С. 109395-109430.
19. Nazir S., Patel S., Patel D. Autonomic computing architecture for SCADA cyber security //Cyber Warfare and Terrorism: Concepts, Methodologies, Tools, and Applications. – IGI Global, 2020. – С. 543-557.
20. DUYMAZLAR O., ENGIN D. Design, application and analysis of an OPC-based SCADA system //Politeknik Dergisi. – 2023. – С. 1-1.
21. Myint A. K. et al. IoT-Based SCADA System Design and Generation Forecasting for Hydropower Station //International Journal of Electrical and Electronic Engineering & Telecommunications. – 2021. – Т. 10. – №. 4. – С. 251-260.
22. Wi-Fi модуль DevKit V1 з ESP-32 // Сайт arduino.ua.
URL: <https://arduino.ua/prod3990-wi-fi-modyl-devkit-v1-s-esp-32> / (дата звернення: 20.01.2024).
23. KY-037 Мікрофон з високою чутливістю // Сайт myproject.com.ua.
URL: <https://myproject.com.ua/ky-037-mikrofon-z-visokoju-chutlivistju-ua.html> / (дата звернення: 20.01.2024).
24. Модуль датчика диму MQ-2 // Сайт arduino.ua.
URL: <https://arduino.ua/prod298-modyl-datchika-dima-mq-2> / (дата звернення: 20.01.2024).
25. HLK-LD2410C 24Ghz Human Presence Radar Module // Сайт hlktech.
URL: <https://www.hlktech.net/index.php?id=1095> / (дата звернення: 20.01.2024).

26. Барометр BMP280 3.3В (датчик атмосферного тиску) // Сайт arduino.ua. URL: <https://arduino.ua/prod1758-barometr-datchik-atmosferного-davleniya-na-bmp280> / (дата звернення: 20.01.2024).

27. Буззер Електромагнітний YHE12-01 // Сайт Voron. URL: <http://new.voron.ua/catalog/042030--buzzer-elektromagnitnyy-yhe12-01-s-generatorom-hyt-huayu-electronics> / (дата звернення: 20.01.2024).

28. Вентилятор для ноутбука Lenovo IdeaPad 5F10N82225 // Сайт 4laptop. URL: <https://4laptop.kiev.ua/ventilyatory-dlya-noutbukov/ventiljator-kuler-dlja-lenovo-ideapad-320-14-320-15-320-17-330-14-330-15-330-17-v320-17-series---oem-.html> / (дата звернення: 20.01.2024).

29. Mishra B., Kertesz A. The use of MQTT in M2M and IoT systems: A survey //IEEE Access. – 2020. – Т. 8. – С. 201071-201086.

30. Нормування шумів // Сайт norma.org.ua. URL: <http://norma.org.ua/knigi/2/2-8-4.php> / (дата звернення: 20.01.2024).