

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

Бакалаврська кваліфікаційна робота

на тему:

«АВТОМАТИЗОВАНА СИСТЕМА МОДЕЛЮВАННЯ ГНУЧКОГО
ВИРОБНИЦТВА»

Виконав: студент 4 курсу, групи 2АКІТ-
21б спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології

 Ілля БОРЩ

Керівник: д.т.н., проф., професор каф. КСУ

 В'ячеслав КОВТУН

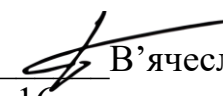
«13» червня 2025 р.

Рецензент: к.т.н., доцент каф. АІТ

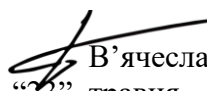
 Юрій ІВАНОВ

«13» червня 2025 р.

Допущено до захисту
Зав. кафедри КСУ

 В'ячеслав Ковтун
«16» червня 2025р.

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 15 – Автоматизація та приладобудування
Спеціальність – 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітньо - професійна програма – Інтелектуальні комп'ютерні системи управління

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
Д.т.н., проф.
 **В'ячеслав КОВТУН**
“23” травня 2025 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
студенту Борщу Іллі Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизована система моделювання гнучкого виробництва

керівник роботи Ковтун В'ячеслав Васильович

затверджені наказом ВНТУ № 97 від 20.03.2025 р.

2. Термін подання студентом роботи 10 червня 2025 року

3. Вихідні дані до роботи:

1. Сидоренко, Л. М. (2017). Промислова автоматизація та SCADA-системи: огляд сучасних рішень. Технічні науки, 4(22), 55–60.

2. Федоренко, І. С. (2020). Моделювання гнучких виробничих систем з використанням SCADA-технологій. Вісник Дніпровського національного університету, 2(58), 60–66.

4. Зміст текстової частини: Аналіз сучасних підходів до SCADA-систем та застосування хмарної інфраструктури. Формулювання технічних вимог до інформаційного каналу обміну даними в хмарному середовищі. Розробка структури та алгоритмів системи моніторингу фізичних параметрів. Вибір апаратних модулів.. Розробка програмної логіки для інтеграції апаратного стенду з хмарною інфраструктурою. Побудова електричної схеми та фізичного макета модуля збору параметрів. Проведення експерименту та перевірка працездатності рішення.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна та принципова схеми SCADA-системи в хмарі. Схема підключення сенсорів. Алгоритм функціонування системи та схема інформаційного потоку AWS. Приклад візуалізації даних.

1.Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв

1. Дата видачі завдання 27.05.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Вибір, узгодження та затвердження теми БКР	08.03.2025	11.03.2025	
2	Визначення технічних характеристик	12.05.2025	16.05.2025	
3	Постановка задачі та розробка ТЗ	16.05.2025	20.05.2025	
4	Розробка структурної, функціональної та принципової схем системи	21.05.2025	02.06.2025	
5	Розробка алгоритму роботи	01.06.2025	03.06.2025	
6	Розробка програмного забезпечення системи	03.06.2025	04.06.2025	
7	Перевірка ПЗ на відповідність вимогам	05.06.2025	06.06.2025	
8	Попередній захист БКР, доопрацювання, рецензування	06.06.2025	07.06.2025	
9	Захист БКР	18.06.2025	18.06.2025	

Студент


(підпис)

Ілля БОРЩ

(ім'я і ПРІЗВИЩЕ)

Керівник роботи


(підпис)

В'ячеслав КОВТУН

(ім'я і ПРІЗВИЩЕ)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 96 сторінок формату А4, на яких є 30 рисунків, 2 таблиці, список використаних джерел містить 30 найменувань. Метою роботи є розробка та впровадження автоматизованої системи моніторингу фізичних параметрів технологічного процесу, що функціонує у хмарному середовищі, для забезпечення оперативного контролю та підвищення ефективності виробництва.

У теоретичній частині розглянуто сучасні SCADA-системи, застосування AWS для інтелектуальних рішень та сформульовано технічні вимоги. У практичній частині розроблено структуру та алгоритми системи моніторингу, обрано апаратні модулі. Реалізовано MQTT-комунікацію з SSL/TLS, розроблено ПЗ для інтеграції з хмарною інфраструктурою. Побудовано електричну схему та макет модуля збору параметрів. Проведено експеримент та перевірено працездатність рішення, включаючи збір даних сенсорами, передачу до AWS IoT Core та IoT SiteWise, обробку та візуалізацію.

Ключові слова: SCADA-система, хмарні технології, Amazon Web Services (AWS), MQTT, моніторинг параметрів, IoT Core, IoT SiteWise, автоматизована система, промисловий інтернет речей.

ABSTRACT

The Bachelor's qualification thesis consists of 96 pages of A4 format, including 30 figures, 2 tables, and a list of 30 references. The purpose of this work is the development and implementation of an automated system for monitoring physical parameters of a technological process, operating in a cloud environment, to ensure operational control and increase production efficiency.

The theoretical part examines modern SCADA systems, the application of AWS for intelligent solutions, and formulates technical requirements. The practical part involved the development of the monitoring system's structure and algorithms, and the selection of hardware modules. MQTT communication with SSL/TLS was implemented, and software was developed for integration with the cloud infrastructure. An electrical circuit diagram and a layout of the parameter collection module were created. An experiment was conducted to verify the solution's functionality, including sensor data collection, transmission to AWS IoT Core and IoT SiteWise, processing, and visualization.

Keywords: SCADA system, cloud technologies, Amazon Web Services (AWS), MQTT, parameter monitoring, IoT Core, IoT SiteWise, automated system, industrial Internet of Things.

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ СУЧАСНИЙ ТЕНДЕНЦІЙ В РОЗРОБЦІ SCADA-СИСТЕМ.....	5
1.1 Концепція структурної організації SCADA систем	5
1.2 Принципи структурної організації SCADA систем	12
1.3 Інтеграція SCSDS-систем з хмарними сервісами.....	20
1.4 Постановка задач	27
2 РОЗРОБЛЕННЯ АРХІТЕКТУРИ SCADA-СИСТЕМИ ГНУЙЧКОГО ВИРОБНИЦТВА З ІНТЕГРАЦІЄЮ ХМАРНОЇ ПЛАТФОРМИ	30
2.1 Аналітичний огляд функціональних можливостей хмарних платформ	30
2.2 Структура SCADA-системи гнучкого виробництва	33
2.3 Проєктування алгоритмічної та програмної складової SCADA-системи гнучкого виробництва	44
3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПОНЕНТІВ SCADA-СИСТЕМИ ОРІЄНТОВАНОЇ НА ГНУЧКЕ ВИРОБНИЦТВО З ІНТЕГРАЦІЄЮ ХМАРНОЇ ІНФРАСТРУКТУРИ.....	48
3.1 Розроблення програмного забезпечення SCADA-системи	48
3.2 Розроблення макету модуля збору технологічних параметрів гнучкого виробництва.....	54
4 ТЕСТУВАННЯ МАКЕТУ СИСТЕМИ ЗБОРУ ТЕХНОЛОГІЧНИХ ПАРАМЕТРІВ ГНУЧКОГО ВИРОБНИЦТВА	59
4.1 Формулювання завдань дослідницького експерименту	59
4.2 Реалізація експериментального дослідження	59
4.3 Інтерпретація результатів дослідження.....	65
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
Додаток А (обов'язковий) Протокол перевірки на наявність запозичень	75
Додаток Б (обов'язковий) Технічне завдання	76
Додаток В (довідковий) Лістинг	79
Додаток Г (обов'язковий) Ілюстративна частина.....	83

ВСТУП

В умовах стрімкого розвитку цифрових технологій аналіз сучасних підходів до створення SCADA-систем набуває особливого значення для ефективного впровадження концепції гнучкого виробництва. Питання автоматизованого контролю та управління технологічними процесами потребує високого рівня технічної реалізації. Особливої уваги заслуговує застосування хмарної інфраструктури, зокрема платформ типу Amazon Web Services, яка дозволяє реалізувати інтелектуальні рішення, орієнтовані на оперативну адаптацію до динамічних змін у виробничому середовищі.

Завдяки використанню сервісів AWS можливо розгорнути масштабовану SCADA-архітектуру, що поєднує високий рівень відмовостійкості, оптимізовану обробку даних у реальному часі та відповідність сучасним вимогам до інформаційної безпеки. Такий технологічний підхід дозволяє досягти високої ефективності управління та оперативного реагування на виробничі зміни, що є критично важливим для підприємств, які працюють у режимі гнучкого виробництва.

Актуальність. Обрання теми зумовлена необхідністю побудови надійного інформаційного каналу для обміну даними між різними функціональними модулями SCADA-системи, що діють у хмарному середовищі. Використання протоколу MQTT, доповненого елементами захисту, відкриває нові можливості для побудови стійких систем моніторингу, в яких безпека, стабільність та інтегративність поєднуються з вимогами до швидкодії та гнучкості.

Запропонована в межах цієї роботи методологія дозволяє досягнути більш ефективного використання ресурсів, мінімізувати затримки при передачі інформації, а також забезпечити стабільну взаємодію з суміжними інформаційними системами. Конструкція SCADA-системи з урахуванням

стандартів, реалізованих у середовищі AWS, формує основу для технологічного прориву в галузі цифровізації виробництва.

Мета і завдання дослідження. Метою цієї бакалаврської кваліфікаційної роботи є проектування і реалізація захищеного протоколу передачі даних MQTT між модулями SCADA-системи, орієнтованої на підтримку принципів гнучкого виробництва.

Об'єктом дослідження є механізми віддаленого збору та візуалізації виробничих даних у хмарно-орієнтованій SCADA-системі.

Предметом дослідження виступають технології організації безпечного обміну даними між компонентами SCADA-системи з використанням обчислювальних потужностей Amazon Web Services.

Методи дослідження. У роботі використовувалися методи системного аналізу, порівняльного аналізу технічних рішень, алгоритмічного моделювання, програмної реалізації вбудованих систем та інтеграції з хмарними сервісами. Для візуалізації та аналізу даних застосовувалися засоби хмарних платформ.

Новизна дослідження. У ході дослідження передбачається реалізувати такі етапи: розгляд архітектурних особливостей SCADA-систем, вивчення інструментарію їх створення, проектування архітектури SCADA-системи з елементами адаптації до AWS, проведення моделювання та оцінка експериментальних результатів у контексті продуктивності та надійності.

Практичне значення. Розроблене рішення може слугувати основою для прототипування та впровадження реальних SCADA-систем на підприємствах з гнучким виробництвом, дозволяючи оптимізувати операційні процеси, підвищити ефективність використання ресурсів та зменшити час простоїв. Також система може бути використана як приклад для вивчення принципів побудови промислових IoT-систем, інтеграції з хмарними платформами та розробки систем моніторингу виробничих процесів.

1 АНАЛІЗ СУЧАСНИЙ ТЕНДЕНЦІЙ В РОЗРОБЦІ SCADA-СИСТЕМ

1.1 Концепція структурної організації SCADA систем

SCADA (Supervisory Control and Data Acquisition) становить собою програмно-апаратний комплекс, який реалізує функції дистанційного нагляду, збору, обробки, відображення та архівації даних у режимі близькому до реального часу. Цей тип програмного забезпечення інтегрується в широкий спектр автоматизованих систем керування, зокрема в індустріальні АСУ ТП, системи контролю енергоресурсів, екологічного моніторингу, автоматизації об'єктів цивільної інфраструктури, а також науково-дослідних установок.

Функціональна сутність SCADA-систем полягає у побудові ефективної інформаційної платформи, що забезпечує зв'язок між фізичними процесами на об'єкті і центрами прийняття рішень. Їхнє впровадження дозволяє суттєво підвищити оперативність реагування на зміну технологічних параметрів, а також досягти цілеспрямованого керування системами у великомасштабних і розподілених середовищах.

SCADA-системи знаходять широке застосування у таких сферах, як енергетичне виробництво та розподіл, металургія, харчова промисловість, логістика, управління водними ресурсами та транспортними мережами. Вони дозволяють забезпечити безперервний контроль за критичними параметрами процесу, своєчасне виявлення відхилень і виконання адаптивних дій без затримок, що є визначальним фактором у середовищах з підвищеною відповідальністю [4].

Концептуальною основою функціонування SCADA виступає багаторівнева модель обробки даних: від збору первинної інформації з периферійних пристроїв до її аналітичної обробки та відображення у вигляді, придатному для оперативного аналізу оператором. Таким чином, кожна зміна стану або параметрів у системі миттєво фіксується, передається та візуалізується з можливістю швидкого втручання.

До ключових функціональних елементів SCADA-систем належать сенсорні пристрої та датчики, які виступають джерелом первинних даних про стан і параметри технологічного середовища. Ці компоненти здійснюють безперервний контроль температури, тиску, витрати, рівня, напруги та інших фізичних величин, критично важливих для стабільного функціонування виробничого процесу. Отримана інформація передається до центрального вузла обробки, де вона систематизується, аналізується та візуалізується за допомогою операторських інтерфейсів. Такий підхід забезпечує своєчасне прийняття рішень та мінімізує ризики технічних збоїв.

Не менш важливою характеристикою сучасних SCADA-рішень є можливість дистанційного керування. Це дозволяє фахівцям оперативно вносити корективи у параметри процесів, незалежно від їхньої фізичної присутності на об'єкті. У випадках, коли об'єкти розташовані у важкодоступних зонах або розподілені територіально, дистанційне управління набуває стратегічного значення – зменшує витрати часу та ресурсів, а також підвищує ефективність управлінських дій.

У контексті сучасних загроз інформаційній безпеці SCADA-системи дедалі частіше оснащуються засобами криптографічного захисту даних. Застосування алгоритмів шифрування, аутентифікації доступу та контрольної перевірки команд дозволяє протистояти спробам зовнішнього втручання, забезпечуючи безперервність виробничого циклу навіть в умовах загроз кібернетичного характеру [5]. Такі засоби захисту стали невід'ємною частиною архітектури SCADA-систем, з огляду на їхню критичну роль у промисловій інфраструктурі.

Необхідною умовою стабільної роботи таких систем є також якісна підготовка персоналу. Спеціалісти, які здійснюють моніторинг та керування через SCADA-інтерфейси, мають володіти відповідними навичками для інтерпретації показників, діагностики збоїв і реалізації заходів із усунення аномалій. Безперервне підвищення кваліфікації операторів, технічне консультування та підтримка з боку розробників є обов'язковими

елементами життєвого циклу будь-якої SCADA-системи, що працює в динамічному виробничому середовищі.

Для всебічного розуміння функціонування SCADA-систем доцільно звернутися до аналізу їхньої архітектурної побудови та структурних елементів, що формують основу таких систем. Стандартна структура SCADA реалізується у вигляді багаторівневої моделі, яка включає операторські станції, серверні вузли, засоби передачі даних та обладнання збору інформації. На рисунку 1.1 представлено традиційну конфігурацію взаємодії компонентів SCADA-системи.

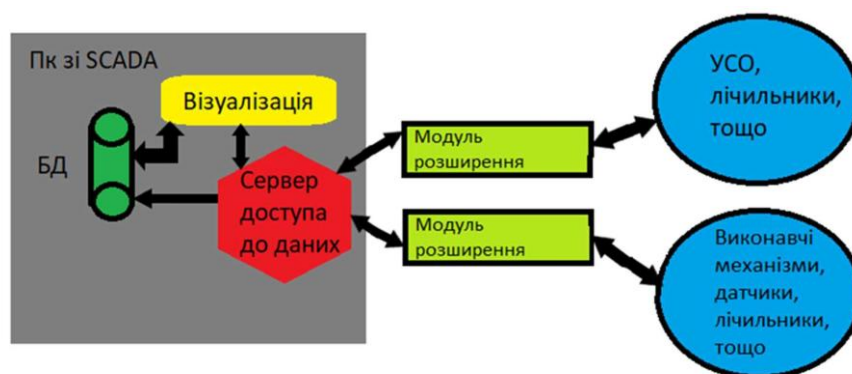


Рисунок 1.1 – Класична схема побудови SCADA системи

Операторські станції виступають основною точкою взаємодії людини з системою. Через них забезпечується візуальний контроль за технологічним середовищем, оперативна обробка сповіщень та реалізація управлінських рішень. Інтерфейси робочих місць надають засоби для відображення даних у вигляді зручних візуальних образів – трендів, мімік-схем, аналітичних панелей. Завдяки цьому оператор отримує повну картину стану системи та може здійснювати точкове налаштування параметрів у разі виявлення відхилень або порушень у технологічному циклі.

Вузли серверного рівня реалізують функціональність зі збору та агрегації даних, що надходять від сенсорної інфраструктури та контролерів. Ці сервери здійснюють попередню обробку сигналів, зберігають історичні

вибірки для подальшого аналізу, а також забезпечують доступ до аналітичних інструментів, необхідних для виявлення закономірностей у зміні параметрів виробничого процесу. Крім того, у структурі серверів передбачені засоби з резервного копіювання та відновлення даних, що підвищує надійність роботи всієї системи [6].

Комунікаційні підсистеми виступають сполучною ланкою, що забезпечує синхронізований обмін даними між функціональними модулями SCADA. Для цих цілей застосовуються як кабельні технології на базі локальних мереж (наприклад, з використанням стандартів Ethernet), так і бездротові рішення, серед яких Wi-Fi або мобільні технології передачі даних. У випадках, коли важливим є забезпечення високої точності та мінімальної затримки, впроваджуються індустріальні протоколи типу MODBUS, CAN або DNP3, які дозволяють встановити надійний обмін інформацією між польовими пристроями та диспетчерським рівнем.

До основної технічної бази SCADA-систем належать пристрої збору даних, серед яких особливе місце займають сенсори, виконавчі механізми, контролери реального часу та прилади для точних вимірювань. Встановлювані безпосередньо в середовищі технологічного процесу, ці пристрої забезпечують безперервне отримання актуальної інформації про параметри виробництва – температурні умови, витрати матеріалів, тиск у трубопроводах, рівень заповнення резервуарів та інші ключові показники. За допомогою відповідних протоколів зв'язку ця інформація передається в режимі низької затримки на центральні обчислювальні вузли системи для її подальшої обробки, візуалізації та прийняття рішень.

На рисунку 1.2 зображено типову модель взаємодії функціональних компонентів SCADA-системи, що ілюструє напрямки потоків даних та принципи логічного зв'язку між структурними елементами.

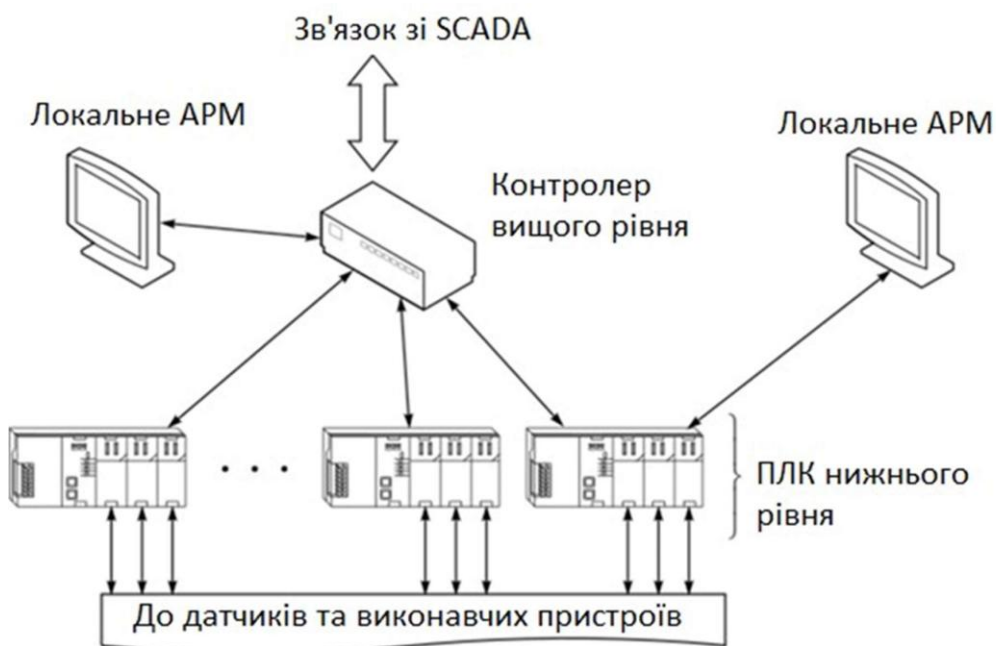


Рисунок 1.2 – Типова схема взаємодії компонентів SCADA системи

Одним із ключових напрямів підвищення ефективності функціонування SCADA-рішень є їхнє використання у сфері енергоефективності. Система моніторингу, зібрана на основі SCADA, дозволяє виявляти ділянки підвищених енергетичних витрат, здійснювати порівняльний аналіз споживання ресурсів у різних часових або виробничих інтервалах, а також автоматизовано реагувати на відхилення, які призводять до нерационального витрачання енергії. На основі накопичених даних можливе впровадження коригувальних стратегій, спрямованих на зменшення навантаження на енергетичну інфраструктуру та підвищення загальної продуктивності виробничої системи [7].

Застосування методів штучного інтелекту в архітектурі SCADA-систем відкриває нові горизонти у сфері автоматизованого аналізу технологічних даних. Інструменти машинного навчання, які впроваджуються у програмні компоненти SCADA, дозволяють не лише автоматично виявляти відхилення в процесах, а й прогнозувати ймовірні збої на основі накопиченої історичної інформації. Алгоритмічна обробка великих обсягів даних створює

можливості для самонавчальних моделей, які поступово вдосконалюють точність діагностики та оптимізаційних рішень без потреби в постійній участі оператора [8].

Подальша еволюція SCADA-рішень безпосередньо пов'язана з розвитком засобів збору інформації. Новітні сенсорні технології забезпечують високу швидкість реакції, підвищену точність та енергоефективність, що є вирішальним для розгортання систем у середовищах з високою динамікою змін. Забезпечення достовірності й своєчасності отриманих даних дозволяє більш точно керувати параметрами технологічного процесу та знижує ризики втрат.

Значним викликом для фахівців, які працюють над удосконаленням SCADA-систем, залишається підвищення рівня їхньої інформаційної безпеки. Враховуючи сучасні реалії кіберзагроз, критично важливо інтегрувати у SCADA-архітектуру багаторівневі засоби захисту. Це охоплює як використання прогресивних методів шифрування трафіку та автентифікації користувачів, так і побудову постійно діючих систем моніторингу аномальної активності. Такий підхід дозволяє ефективно протидіяти як зовнішнім атакам, так і внутрішнім спробам несанкціонованого доступу.

Важливою складовою розвитку SCADA-середовищ є уніфікація протоколів обміну даними. Стандартизація взаємодії між різними системами дозволяє спростити інтеграційні процеси, підвищити інтероперабельність та зменшити витрати на адаптацію обладнання. Завдяки впровадженню загальноприйнятих технічних регламентів можливим стає побудова більш відкритої та гнучкої екосистеми автоматизації, що сприяє підвищенню рівня цифрової трансформації виробництва [9].

У цьому контексті значну роль відіграє інтеграція принципів Індустрії 4.0, які включають глибоку взаємодію між фізичними та цифровими складовими виробництва. У рамках інтелектуального підприємства SCADA-системи взаємодіють із об'єктами через IoT, використовують хмарні обчислення, моделі цифрових двійників та інструменти автоматизованої

аналітики. Це дозволяє реалізовувати самокеровані виробничі ланцюги з високим ступенем автономності, що забезпечує оперативне прийняття рішень, підвищену ефективність і адаптивність виробничих систем до умов ринку.

Одним із ключових напрямів сучасних досліджень у сфері SCADA-технологій є вивчення потенціалу бездротових мереж п'ятого покоління. Інфраструктура 5G істотно розширює можливості систем моніторингу та керування завдяки високій швидкості обміну даними, надзвичайно низькій затримці сигналу та підвищеній стабільності з'єднання. Це забезпечує умови для надійної передачі критичних повідомлень у реальному часі, що є вкрай важливим для оперативного керування технологічними процесами з підвищеним рівнем чутливості до затримок. Застосування 5G у SCADA-системах дозволяє реалізовувати більш точну реакцію на змінні параметри виробництва, забезпечуючи стабільність функціонування навіть в умовах значного навантаження на інформаційні канали.

Поряд із цим, вагоме місце в структурі досліджень посідає забезпечення безперервної роботи SCADA-систем у ситуаціях порушення стандартного режиму експлуатації. Упровадження методів динамічного резервування, відновлення функцій після збоїв та попереджувального діагностування з використанням предиктивної аналітики дозволяє гарантувати стійкість системи навіть за умов аварійного припинення роботи окремих її елементів. Такий підхід сприяє скороченню часу простою, зниженню ризиків технологічних втрат та покращенню загального балансу продуктивності [10].

Побудова ефективної SCADA-системи неможлива без дотримання низки базових організаційних принципів, які забезпечують її функціональну цілісність та здатність до масштабування. Умовно ці принципи можна згрупувати за основними архітектурними характеристиками, що формують цілісну логіку взаємодії системних компонентів.

У таблиці 1.1 подано модифікований перелік ключових принципів, що лежать в основі організації SCADA-систем, із зазначенням їхнього функціонального призначення.

Таблиця 1.1 – Основоположні принципи архітектури SCADA-систем

Принцип	Функціональна характеристика
Обробка подій у реальному часі	Забезпечує негайну реакцію системи на зміну технологічних параметрів або появу аварійних ситуацій
Централізована диспетчеризація	Сприяє концентрації управлінських функцій у єдиному операторському середовищі
Візуально-орієнтоване середовище	Формує зручну для оператора інтерпретацію даних у вигляді інфографіки, панелей та схем
Автоматизоване керування	Дозволяє системі самостійно регулювати задані параметри процесу без втручання людини
Історичне логування	Передбачає збереження хронології подій та параметрів з метою подальшого аналізу
Модульна масштабованість	Створює умови для розширення або модифікації системи без необхідності повної перебудови
Інтегративність	Забезпечує сумісність з іншими промисловими системами через відкриті протоколи

Згадані принципи взаємодіють у рамках спільної архітектурної моделі, що дозволяє SCADA-системам адаптуватися до змін, підвищувати ефективність виробничих процесів та мінімізувати ризики, пов'язані з технічними або інформаційними відмовами.

1.2 Принципи структурної організації SCADA систем

Сучасне програмне забезпечення, що реалізує функціональність SCADA-систем, базується на широкому наборі високотехнологічних рішень, які забезпечують не лише базовий моніторинг, а й поглиблений аналіз параметрів технологічних процесів. У межах цих систем реалізовано інтегровані механізми обробки історичних даних, що дозволяють здійснювати часовий аналіз, виявляти приховані закономірності та моделювати прогнозні сценарії на основі минулих показників. Засоби візуалізації в SCADA-системах забезпечують інтуїтивно зрозуміле представлення аналітичної інформації через графіки, інтерактивні панелі,

діаграми та інші форми аналітичного інтерфейсу, що значно полегшує прийняття оперативних рішень менеджментом виробництва.

Надійність експлуатації SCADA значною мірою залежить від наявності ефективних засобів захисту від збоїв, що реалізується через механізми резервування. Зокрема, дублювання основних апаратно-програмних компонентів – таких як серверні вузли, канали зв'язку та програмні модулі – дозволяє автоматизовано перемикатися на резервні елементи інфраструктури у разі виявлення критичних порушень. Це дозволяє уникати значних затримок у роботі системи, а також підтримувати постійний контроль за технологічним середовищем навіть у надзвичайних обставинах.

У рамках інтеграційних підходів SCADA-системи здатні взаємодіяти з іншими корпоративними платформами, зокрема з комплексами управління ресурсами підприємства (ERP) та системами обліку та керування виробничими фондами. Такий тип взаємодії забезпечує наскрізну передачу інформації між виробничим рівнем і рівнем стратегічного планування, дозволяючи здійснювати точне узгодження між фактичними виробничими параметрами та логікою розподілу ресурсів. Це сприяє створенню єдиного інформаційного середовища, що охоплює як технічну, так і адміністративну частину підприємства.

У сфері розробки SCADA-платформ спостерігається постійна динаміка оновлення продуктів. Провідні розробники програмного забезпечення щорічно випускають основні стабільні релізи та додаткові оновлення, які включають покращення інтерфейсу, розширення функціоналу або інтеграцію з новими технологічними протоколами. Це зумовлено потребою в адаптації до змін ринку, появою нових вимог користувачів, а також технологічним прогресом у сфері хмарних обчислень, IoT та аналітики великих даних. Усе це забезпечує постійну актуальність SCADA-рішень у контексті цифрової трансформації виробничих систем.

У межах еволюції SCADA-систем одним із ключових напрямів удосконалення є зміна принципів структуризації об'єктів управління.

Традиційні реалізації SCADA, як правило, оперували концепцією індивідуальних параметрів, кожен з яких маркувався унікальним тегом. Такий підхід, заснований на атомарному розбитті процесу на окремі змінні, виявляється неефективним у великих за масштабом виробничих середовищах. При масштабуванні систем, коли кількість параметрів обчислюється тисячами, ручне створення і підтримка тегів значно ускладнює проєктування, супровід і розвиток SCADA-інфраструктури.

З урахуванням цих обмежень, сучасні програмні платформи SCADA орієнтуються на об'єктно-орієнтований підхід, за якого окремі пристрої або цілі функціональні підсистеми трактуються як повноцінні об'єкти (класи), що інкапсулюють у собі всі властиві їм параметри та методи. Це дозволяє описувати технологічні вузли у вигляді структурованих сутностей, де логіка, змінні та інтерфейси поєднані в єдину модель. Такий підхід підтримує повторне використання компонентів, значно спрощує налаштування системи та полегшує командну розробку, зокрема у складних проєктах із паралельною участю кількох інженерів.

У контексті автоматизованих систем управління технологічними процесами (АСУ ТП), практично всі рівні такої ієрархії сьогодні базуються на програмованих технічних засобах, які потребують відповідного програмного забезпечення. До складу цього ПЗ входять програмні модулі контролю та зв'язку з технологічним обладнанням (ТОУ), а також клієнтські додатки, розгорнуті на робочих станціях операторів або диспетчерів. Зазвичай ці модулі відповідають за забезпечення взаємодії між польовими рівнями і людино-машинним інтерфейсом HMI (Human-Machine Interface), через який здійснюється візуалізація та керування процесом. Ефективна побудова HMI забезпечує зручність користування, зменшує ймовірність помилок персоналу й підвищує загальну оперативність реагування на виробничі ситуації [11].

У цьому розділі детально розглядаються актуальні концепції створення програмного забезпечення для автоматизованих робочих місць (АРМів), які є

складовими частинами різнорівневих структур автоматизованих систем керування технологічними процесами (АСУ ТП).

Програмне забезпечення для АРМів розробляється згідно з вимогами проєктної документації ще на етапі проєктування АСУ ТП. Основне призначення цього програмного забезпечення полягає в забезпеченні безперервного контролю за динамікою стану окремих технологічних об'єктів (ТхО), а також за поведінкою технологічного процесу (ТП) у цілому. Воно повинно надавати засоби для підтримки процесів прийняття управлінських рішень, як у межах оперативного втручання для зміни параметрів функціонування, так і в аспектах координації дій виконавчих підрозділів підприємства відповідно до затверджених регламентів. Таким чином, створюване програмне забезпечення повинно одночасно відповідати вимогам до інтеграції з фізичним обладнанням і підтримувати інтерфейси, орієнтовані на ефективну взаємодію з користувачем, який приймає рішення на відповідному функціональному рівні.

Відмінною особливістю програмного забезпечення для АРМів у складі АСУ ТП є необхідність забезпечення дій у реальному часі. Це передбачає жорстке нормування часу реагування системи на будь-який збурюючий вплив, що може походити як з боку технологічного середовища, так і з боку користувача, який працює через інтерфейс АРМу. У контексті цього підходу особливого значення набуває гарантоване дотримання вимог до часового бюджету обробки подій, включно з аналізом вхідних даних, формуванням рішень та передачею відповідних керуючих дій у контур управління.

Окрему проблему при розробці програмного забезпечення АСУ ТП становить забезпечення прозорості взаємодії між фахівцями різного профілю – інженерами, відповідальними за розробку систем автоматизації, і технологами, які формують технічне завдання на основі знань про виробничі процеси. Проблематика формалізації взаєморозуміння між цими групами є особливо актуальною, оскільки ефективна автоматизація потребує узгодженого трактування кожного елемента функціональної специфікації.

Якщо для рівня взаємодії з технічними об'єктами опис алгоритмів керування може базуватися на чітко визначених математичних моделях, то у випадку з інтерфейсами НМІ (Human-Machine Interface) подібна формалізація значно ускладнена через відсутність суворих законів, що регламентували б структуру взаємодії людини з цифровою системою.

Ця проблема піднімає питання про пошук таких методів опису НМІ, які дозволяли б досягати достатнього рівня формалізації у постановці завдань автоматизації з боку різних учасників розробки. Формалізований підхід до проєктування інтерфейсів повинен не лише зменшити кількість помилок на етапі специфікації вимог, але й забезпечити цілісність бачення системи протягом усього життєвого циклу її створення, від концептуального проєктування до дослідної експлуатації [12].

Інтеграція SCADA-систем у процес розробки програмного забезпечення для автоматизованих систем керування технологічними процесами забезпечує низку функціональних переваг, які охоплюють як взаємодію з технічними об'єктами, так і підтримку користувацького інтерфейсу для осіб, що приймають управлінські рішення.

Однією з ключових особливостей є можливість візуального представлення технологічного процесу у формі мнемосхем, які розробляються з використанням уніфікованих візуальних образів – графічних елементів, що відображають поточний стан технологічного обладнання, а також органів керування, стилізованих під кнопки, перемикачі, повзунки або аналогові регулятори. Інформація про параметри процесу може бути представлена у вигляді цифрових індикаторів, шкальних стрілочних показників, діаграм тренду або кольорових сигнальних табло, що забезпечує зручне сприйняття навіть у динамічному режимі зміни стану об'єкта.

Високий рівень інтеграції досягається завдяки підтримці роботи з широким спектром зовнішніх пристроїв, зокрема контролерами, сенсорними модулями, виконавчими механізмами тощо, за рахунок підключення через відповідні драйвери, які забезпечують коректний обмін даними незалежно

від виробника пристрою. Програмне забезпечення SCADA реалізовано як подієво-орієнтоване середовище, що працює у режимі реального часу. При цьому нормується час реакції на зовнішній вплив, а всі отримані параметри фіксуються з прив'язкою до календарно-часової координати.

Система дозволяє здійснювати аналітичну обробку даних, що надходять з технологічного середовища, перевіряючи їх на відповідність допустимим межам або швидкісним характеристикам зміни. Інформація може бути виведена у графічному форматі або використана для подальшого реагування системи через сигнали тривоги. Сповіщення персоналу про критичні події реалізуються через світлові індикатори, акустичні сигнали або повідомлення, що з'являються в інтерфейсі користувача, з одночасною реєстрацією цих подій у відповідному архіві.

Система також забезпечує збереження логів, у яких фіксуються аварійні стани, дії персоналу в період їхнього виникнення, а також архівуються ключові параметри функціонування об'єкта з можливістю зберігання у формі часових вибірок – повних або часткових, що створюються періодично або за умов досягнення певних порогів. Такі архіви дозволяють не лише відстежити динаміку зміни змінних, а й побудувати прогнозні моделі або виявити закономірності за допомогою функції трендів.

Значна увага приділяється і засобам безпосереднього управління технологічними об'єктами з диспетчерського або операторського рівня. Система забезпечує формування керуючих команд і передає їх до виконавчих пристроїв через програмно-зв'язані інтерфейси, з одночасною реєстрацією кожного акту управління у системному журналі. Архівна інформація використовується як основа для створення звітів, аналітичних довідок або зведень, що формуються у заданому форматі й можуть включати як числові значення, так і візуальні матеріали.

Програмна реалізація SCADA передбачає можливість автоматичного управління об'єктами за наперед заданими алгоритмами. У таких алгоритмах можуть застосовуватись математичні або логічні залежності, що адаптуються

до режимів роботи технологічного середовища. Підтримуються також спеціалізовані режими, як-от консультативне керування, супервізорний контроль або пряме цифрове регулювання. Обирати необхідні алгоритми дозволяє підключення до бібліотеки шаблонів, яка містить типові логічні схеми або математичні моделі керування.

Безпека функціонування реалізується через впровадження механізмів захисту від несанкціонованого доступу, включаючи розмежування прав користувачів, автентифікацію, реєстрацію сеансів і контроль змін параметрів. Удосконалений графічний інтерфейс підтримує багатовіконний режим, імпорт користувацьких зображень, створення бібліотек динамічних об'єктів, а також розширення функціональності через додаткові графічні редактори, що дає змогу адаптувати систему до конкретних вимог замовника.

Застосування SCADA-технологій у середовищі створення програмного забезпечення для автоматизованих систем управління відкриває широкі перспективи щодо реалізації складних функцій моніторингу, регулювання та взаємодії між людиною та технічною системою. Ця технологічна платформа забезпечує можливість реалізації комплексного контролю за технологічними процесами на всіх рівнях, охоплюючи збір інформації з різних точок технологічного об'єкта, автоматизоване керування окремими вузлами та агрегатами, а також синтез узагальнених алгоритмів регулювання для всього процесу в цілому. Однією з найсильніших сторін SCADA є підтримка графічного інтерфейсу, орієнтованого на інтуїтивно зрозумілу візуалізацію, який дозволяє будувати ефективну модель інтерактивної взаємодії типу «людина – система» в межах кожного окремого АРМу.

Важливою є й здатність SCADA-технології до створення вбудованих механізмів фіксації історії змін у параметрах управління, а також архівування інформації про всі дії, що здійснювалися як з боку системи, так і користувачем. Така історичність є основою для подальшого аналізу ефективності системи, виявлення вузьких місць у процесі, а також аудиту виконаних дій в рамках технічного чи адміністративного нагляду. Крім того,

SCADA-системи підтримують інтеграцію з зовнішніми інструментами для обробки візуальних матеріалів, що дозволяє створювати користувацькі мнемосхеми, адаптовані до вимог конкретного проекту.

Слід підкреслити, що SCADA сама по собі не реалізує управлінські функції, а надає розробникам інструментарій, який дозволяє побудувати цю функціональність із урахуванням специфіки об'єкта автоматизації. Це означає, що технологія лише задає рамкові можливості – а ефективність їх реалізації повністю залежить від технічного завдання, обраної програмної архітектури, рівня підготовки розробника, а також функціонального потенціалу тієї SCADA-платформи, яка використовується у проєкті. Водночас ця відкритість платформи дозволяє значно зменшити витрати часу на розробку, оптимізувати трудомісткість написання коду, створити прозору й документовану логіку керування, а також досягти високої повторюваності та надійності в реалізованих рішеннях.

Як уже було зауважено раніше, одним із ключових факторів успішного впровадження SCADA є здатність забезпечити якісну комунікацію між фахівцями різних профілів – зокрема, між розробниками ПЗ та інженерами-технологами, які формують вимоги до автоматизації. Цей діалог відбувається через спільне формування візуального відображення логіки технологічного процесу: мнемосхема, яка стає початковим етапом створення програмного забезпечення, уже містить в собі концепцію керуючих дій, форми подання вимірюваних параметрів, способи індикації аварій або порушень, а також структуру збереження архівних даних. Таким чином, процес проєктування починається з визначення очікуваного результату на рівні візуальної взаємодії, що дозволяє побудувати функціональну модель системи, яка відповідає реальним потребам виробництва і зрозуміла всім учасникам проєкту ще до етапу реалізації програмної логіки.

На етапі функціонального узгодження, яке передуює безпосередній реалізації програмно-технічної частини автоматизованої системи керування технологічним процесом, ключовим елементом є формування погодженого

бачення інтерфейсу взаємодії користувача із системою. Саме під час діалогу між замовником і розробником визначаються пріоритетні форми представлення інформації, зокрема способи відображення змінних параметрів, типи елементів керування, логіка візуального супроводу процесів, принципи подання сигнальних повідомлень та структура інтерфейсу кожного конкретного АРМу.

Лише після того як досягнуто консенсус щодо концепції інтерфейсу «людина – система», який має забезпечувати контроль і регулювання процесів на кожному функціональному рівні, розробники можуть переходити до етапу створення нижнього рівня системи – апаратного забезпечення, драйверів взаємодії з технічними об'єктами та логіки обробки подій у реальному часі. Такий підхід гарантує цілісність технічного рішення та дозволяє уникнути ситуацій, коли в процесі реалізації виявляються неузгодженості, пропуски у функціональних вимогах або неактуальні компоненти, які не відповідають реальним запитам користувача.

Визначення взаємодії людини з системою ще на ранньому етапі дозволяє не лише уникнути трудомістких доопрацювань у майбутньому, а й формує стабільне методологічне підґрунтя для всієї системи. Результатом такого погодження стає документація, яка визначає функціональне призначення системи й слугує орієнтиром для розробників при реалізації нижнього рівня, без потреби постійного повернення до вихідних вимог або уточнень, які гальмують процес проєктування.

1.3 Інтеграція SCSDS-систем з хмарними сервісами

Інтеграція хмарних технологій у структуру SCADA-систем стала важливим етапом у переосмисленні підходів до організації промислового моніторингу, збирання даних і централізованого управління. Завдяки хмарним сервісам істотно розширюється функціональний спектр таких систем, зокрема за рахунок можливості забезпечення віддаленого доступу до

інформаційних потоків у реальному часі, незалежно від географічного розташування користувача або об'єкта автоматизації. Це дозволяє оперативно реагувати на зміну технологічних параметрів та підтримувати стабільне управління навіть у багаторівневих або географічно розподілених виробничих структурах.

Переваги, які забезпечує використання хмарних обчислювальних платформ у SCADA-середовищі, охоплюють декілька ключових напрямів. По-перше, масштабованість інфраструктури дозволяє динамічно адаптувати обсяг оброблюваних і збережених даних до поточних потреб підприємства, без необхідності інвестування у фізичне розширення локальних серверів. По-друге, хмарна модель значно знижує витрати на апаратну інфраструктуру, оскільки користувачеві не потрібно розгортати та обслуговувати власні серверні рішення. По-третє, системи, розміщені у хмарному середовищі, забезпечують безперервний доступ до інформації з будь-якої точки з доступом до Інтернету, що сприяє реалізації концепції віддаленого управління виробничими потужностями.

Особливу увагу в контексті використання хмарних SCADA-рішень слід приділити питанням інформаційної безпеки. Передача даних через відкриті канали потребує впровадження криптографічних засобів захисту, зокрема шифрування даних, механізмів автентифікації користувачів і контролю доступу. Вибір хмарного провайдера повинен ґрунтуватися на відповідності його політик актуальним стандартам у сфері кібербезпеки, включаючи сертифікацію відповідно до вимог міжнародних протоколів з безпеки й конфіденційності.

Хмарна інфраструктура також забезпечує регулярне оновлення програмного забезпечення на стороні сервіс-провайдера. Це дозволяє знизити ризики, пов'язані з використанням застарілих версій ПЗ, а також гарантує своєчасну адаптацію системи до нових функціональних вимог без потреби у втручанні зі сторони користувача. Таким чином, система залишається

актуальною, стійкою до уразливостей і здатною швидко реагувати на зміни в середовищі експлуатації.

Разом із розвитком хмарних технологій, що стали основою для формування концепції Industry 4.0, змінюється сама парадигма побудови SCADA-систем. Підключення до IoT-пристроїв, використання хмарних аналітичних сервісів, автоматизоване оновлення структури взаємодії між підсистемами – усе це сприяє формуванню нової архітектури, в якій SCADA виступає не лише засобом збору даних, а й повноцінною платформою для інтелектуального керування процесами на основі обчислювальної аналітики та штучного інтелекту.

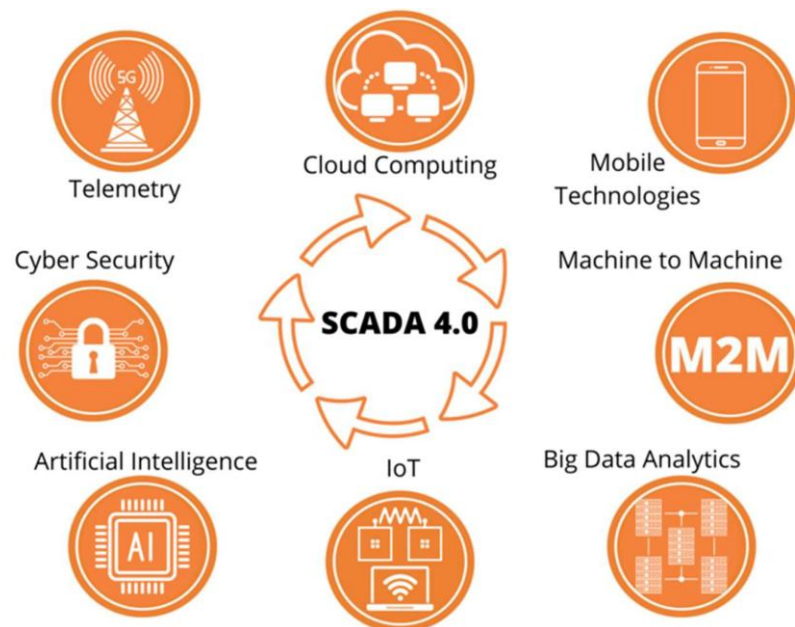


Рисунок 1.3 – Розширені можливості SCADA-систем, які з’явилися завдяки Industry 4.0 [13]

Розширення функціональності SCADA-систем шляхом інтеграції з хмарними сервісами суттєво змінює традиційні підходи до зберігання, обробки й управління даними в промисловому середовищі. Одним із найважливіших напрямів такої трансформації є перенесення інфраструктури зберігання інформації до хмарного середовища, що дозволяє уникнути витрат на обслуговування локальних серверів та зберігальних пристроїв. Хмарне

зберігання гарантує можливість доступу до накопичених масивів даних з будь-якого пристрою, підключеного до Інтернету, що створює передумови для повної децентралізації управління та моніторингу.

Крім того, платформи хмарного типу надають інструменти для аналізу даних у режимі реального часу. Це забезпечує оперативну реакцію на критичні події без потреби залучення проміжних пристроїв або додаткових програмно-апаратних засобів. Автоматизована обробка сигналів від польових пристроїв і сенсорів дозволяє сформувати динамічну модель процесу з високою точністю відображення поточних параметрів, що сприяє прийняттю своєчасних управлінських рішень.

Перехід до хмарної моделі також дозволяє скоротити загальні витрати на побудову та утримання IT-інфраструктури. Замість одноразових великих інвестицій у локальні сервери, користувачі отримують доступ до обчислювальних потужностей у рамках моделі оплати за використання, що робить систему гнучкою в плані масштабування та економічно ефективною на тривалий період експлуатації.

У контексті забезпечення безпеки хмарні платформи пропонують розширені засоби захисту інформації. Шифрування даних, автентифікація на основі сертифікатів, сегментація мережевого трафіку та постійний моніторинг підозрілої активності є стандартними елементами таких сервісів. Вибір сертифікованого постачальника дозволяє гарантувати збереження конфіденційності й цілісності критично важливих технологічних даних навіть при роботі в загальнодоступному мережевому середовищі.

Суттєвим фактором змін є також поєднання хмарних SCADA-систем з концепцією Інтернету речей. Така інтеграція забезпечує збирання даних не лише з традиційних сенсорів, а й з IoT-пристроїв, що працюють у розподілених умовах і передають велику кількість параметрів у централізоване аналітичне середовище. Завдяки цьому створюється більш повна картина стану контрольованих об'єктів, що дає змогу не лише моніторити поточні параметри, а й формувати довгострокові прогнози щодо

зміни їхнього стану. Це суттєво підвищує адаптивність, надійність та ефективність усієї системи управління.

Одним із визначальних напрямів трансформації SCADA-систем в умовах цифрової індустрії є впровадження інструментів машинного навчання та аналітики великих даних. Завдяки цим технологіям з'являється можливість прогнозувати потенційні збої в роботі обладнання, виявляти нетипові відхилення в динаміці процесів і здійснювати превентивні заходи ще до настання критичних ситуацій. Це не лише підвищує надійність технологічних ланцюгів, а й оптимізує витрати на технічне обслуговування, дозволяючи перейти до концепції обслуговування на основі фактичного стану об'єкта.

Хмарні сервіси, які лежать в основі новітніх SCADA-архітектур, забезпечують ефективну підтримку географічно розподілених об'єктів, даючи змогу централізовано збирати та аналізувати дані з численних віддалених локацій. Такий підхід значно підвищує ефективність управління інфраструктурою, дозволяє зменшити кількість локального персоналу на об'єктах і забезпечити координацію дій в реальному часі навіть у масштабах транснаціонального виробництва.

Хмарна модель також значно спрощує управління потоками даних з різномірних джерел – від промислових сенсорів і контролерів до інтелектуальних IoT-модулів, які фіксують широкий спектр параметрів у режимі реального часу. Завдяки централізованій обробці та уніфікованому зберігання даних користувачі мають змогу отримати узгоджену, повну та достовірну інформаційну картину, на основі якої приймаються стратегічні рішення у сфері управління виробничими ресурсами.

Інтеграція хмарних технологій у SCADA-середовище сприяє підвищенню автоматизації, точності та продуктивності всього виробничого процесу. Водночас впровадження таких систем вимагає ретельного дотримання вимог до інформаційної безпеки. Вибір хмарного провайдера має ґрунтуватися на оцінці його здатності гарантувати цілісність даних,

використанні сучасних механізмів кіберзахисту та дотриманні міжнародних стандартів інформаційної безпеки.

Станом на сьогодні спостерігається стрімкий розвиток систем дистанційного моніторингу та керування, які базуються на хмарних обчислювальних сервісах і можуть бути ефективно використані як основа для створення архітектур SCADA-систем нового покоління. Завдяки використанню таких платформ, як Amazon Web Services (AWS) та Microsoft Azure, стало можливим впровадження комплексних рішень для віддаленого збору, обробки, аналізу й зберігання даних у промислових умовах. Ці сервіси надають набір готових інструментів, що значно спрощують інтеграцію SCADA з Інтернетом речей (IoT), формуючи цілісну інформаційно-керуючу екосистему.

Зокрема, використання AWS IoT відкриває широкі можливості для підключення пристроїв до мережевої інфраструктури через веб-сервіси, що дозволяє не лише отримувати дані від сенсорів, але й організовувати зворотний зв'язок – надсилання керуючих команд до виконавчих модулів. Однією з важливих переваг платформи є її здатність забезпечувати безперервну роботу з пристроями, навіть коли ті функціонують в автономному режимі. Це дозволяє підтримувати зв'язок з віддаленими об'єктами у складних або нестабільних умовах підключення [14].

Шлюз пристроїв у системі AWS виконує функцію координатора для IoT-модулів, що взаємодіють із хмарною інфраструктурою. Через цей шлюз здійснюється управління активними пристроями, а також реалізується переклад протоколів комунікації в уніфіковану форму для взаємодії з ядром обчислювальної платформи AWS IoT. Крім того, шлюз реалізує захисні функції, зокрема автентифікацію, контроль доступу та шифрування, що дозволяє забезпечити безпечну передачу даних навіть у публічному мережевому середовищі [15].

Ключовим елементом концепції Інтернету речей є набір керованих сервісів, що функціонують безпосередньо в хмарі, включаючи компоненти

зберігання, логіки керування, моніторингу й обміну даними. Ці сервіси дозволяють підтримувати підключення величезної кількості пристроїв, виконувати масштабований контроль, а також реалізовувати аналітичну обробку даних в реальному часі. Подібні технологічні рішення є базовими при побудові інтелектуальних виробничих середовищ, де необхідне інтегроване управління технологічними об'єктами на базі розподілених обчислень.

Крім того, хмарні сервіси дають змогу розробляти спеціалізовані програмні компоненти, призначені для моделювання, адаптації та розширення функціональних можливостей систем керування. Це дозволяє створювати масштабовані SCADA-рішення, які можна гнучко налаштовувати під конкретні завдання підприємства.

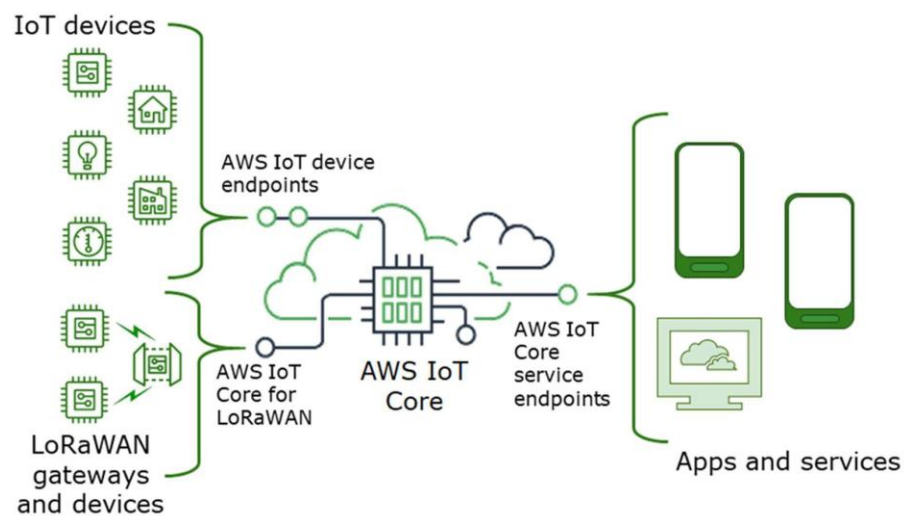


Рисунок 1.4 – Схема можливої реалізації взаємодії з IoT-пристроями за допомогою Amazon Web Services [16]

У контексті побудови сучасних SCADA-систем, які інтегрують концепцію Інтернету речей, особливе значення мають базові структурні елементи IoT-архітектури. Серед них виокремлюють три ключові категорії: «речі», тобто фізичні або промислові пристрої – зокрема сенсори, виконавчі механізми, мікроконтролери, які генерують дані та взаємодіють із зовнішнім

середовищем; «інсайти» – результати обробки зібраної інформації, що можуть бути сформовані як штучними інтелектуальними системами, так і користувачами на основі алгоритмічного аналізу; а також «дії» – тобто реалізовані на основі цих інсайтів рішення, що впливають на керування технічними об'єктами або стратегічне планування.

Більшість IoT-пристроїв функціонують у режимі односторонньої передачі даних, що передбачає надсилання показників з апаратного рівня до хмарного середовища. У цьому випадку хмара виконує функції зберігання, агрегації, попередньої обробки й аналітики інформації. Проте технічно можливі й інші режими взаємодії – зокрема зворотна передача даних із хмари до пристрою або повноцінне двостороннє з'єднання. Наприклад, сенсор руху може кожного разу надсилати дані до хмарної платформи при фіксації об'єкта в зоні дії, а багатофункціональні пристрої можуть не лише інформувати про перевищення порогових значень, але й отримувати зворотні команди для зміни конфігурації або активації виконавчих механізмів.

У деяких випадках передача даних здійснюється у реальному часі з метою підтримки візуалізації на диспетчерському рівні або HMI, що дозволяє операторам спостерігати за станом системи з мінімальною затримкою. Це особливо важливо для забезпечення безперервного контролю та миттєвого реагування у середовищах з підвищеними вимогами до безпеки або надійності. Таким чином, використання IoT-пристроїв у хмарно-орієнтованих SCADA-системах формує гнучке, адаптивне й масштабоване середовище цифрового моніторингу й управління виробничими процесами.

1.4 Постановка задач

На основі проведеного огляду сучасних підходів до побудови SCADA-систем можна констатувати, що дана тематика зберігає свою актуальність у контексті розвитку автоматизованих засобів керування та цифровізації виробничих процесів. Попри те, що класичні SCADA-рішення часто

асоціюються з централізованими моделями управління, аналіз сучасних реалізацій демонструє високу адаптивність цих систем до децентралізованих умов, де стабільність, автономність і стійкість до відмов виступають визначальними факторами. Саме ці характеристики роблять SCADA-системи незамінними у критичних сферах, що вимагають гарантованої безперервності функціонування та підвищеної надійності.

Ураховуючи тенденції розвитку індустріальних IT-рішень і швидке впровадження хмарних технологій, виникає необхідність у переосмисленні механізмів обміну даними між функціональними модулями SCADA-систем. У зв'язку з цим доцільним є проектування механізмів захищеного зв'язку, орієнтованих на специфіку гнучкого виробництва, що передбачає постійну змінність умов та структури керованих об'єктів.

Метою цієї кваліфікаційної роботи є розробка криптографічно захищеного протоколу обміну повідомленнями на основі MQTT, призначеного для забезпечення надійної комунікації між компонентами SCADA-системи, реалізованої в середовищі гнучкого виробництва на основі хмарної інфраструктури Amazon Web Services.

Об'єктом дослідження виступає процес дистанційного нагляду за технологічним процесом із можливістю візуалізації його параметрів у реальному часі в межах SCADA-системи.

Предметом розробки є удосконалення процедур передачі інформації між модулями SCADA-архітектури з використанням сервісів AWS та засобів MQTT, які забезпечують баланс між ефективністю передавання, масштабованістю системи та безпекою даних.

Для реалізації поставленої мети необхідно послідовно вирішити низку дослідницько-прикладних завдань. Насамперед, потрібно здійснити систематичний аналіз концептуальних засад побудови SCADA-систем, включно з принципами їхньої архітектури та логікою функціонування. Далі – провести огляд сучасних інструментів програмної реалізації та моделей розгортання SCADA-інфраструктур, орієнтованих на гнучке виробництво.

Окрему увагу слід приділити дослідженню потенціалу хмарних сервісів, зокрема AWS, в контексті реалізації віддаленого моніторингу, зберігання й обробки даних.

Наступним етапом є розробка архітектурної моделі SCADA-системи, що інтегрує обрані технічні рішення, та її реалізація у вигляді макету, на якому можна здійснити серію експериментальних досліджень. Завершальним кроком стане аналіз отриманих результатів, спрямований на оцінку ефективності запропонованого рішення з точки зору швидкодії, надійності та стійкості до інформаційних загроз.

2 РОЗРОБЛЕННЯ АРХІТЕКТУРИ SCADA-СИСТЕМИ ГНУЙЧКОГО ВИРОБНИЦТВА З ІНТЕГРАЦІЄЮ ХМАРНОЇ ПЛАТФОРМИ

2.1 Аналітичний огляд функціональних можливостей хмарних платформ

У сучасному цифровому середовищі технології, що ґрунтуються на хмарних обчисленнях, поступово трансформуються зі сфери побутового використання в ключову інфраструктуру для реалізації автоматизованих систем керування виробництвом. Віртуалізоване середовище, відоме як хмарна інфраструктура, базується на взаємодії апаратних серверів та спеціалізованого програмного забезпечення, яке дозволяє реалізовувати розподілене розгортання віртуальних машин у середовищі з підтримкою гіпервізора. Такий програмний інтерфейс забезпечує автономну ізоляцію операційних систем і прикладних модулів, емулюючи апаратні ресурси та керуючи їх розподілом між компонентами віртуалізованої системи. До базових ресурсів, доступ до яких контролюється гіпервізором, належать обчислювальні потужності, оперативна пам'ять, обсяг внутрішнього сховища та пристрої вводу/виводу.

Хмарні обчислення формують основу для динамічного надання ресурсів і сервісів за допомогою віддаленого доступу через глобальну мережу. Вони орієнтовані на підтримку масштабованих програмних рішень, інтеграцію з великими обсягами даних і оптимізацію витрат на фізичну інфраструктуру. Такий підхід дозволяє зменшити залежність підприємств від локальних обчислювальних потужностей, сприяючи реалізації більш гнучких моделей автоматизованого виробництва [17].

Функціонування хмарних сервісів передбачає збереження даних та їх обробку на спеціалізованих віддалених платформах. Це забезпечує можливість динамічного масштабування обчислювальних потужностей, підвищення продуктивності та зменшення експлуатаційних витрат. У межах

побудови SCADA-систем гнучкого виробництва важливим є вибір відповідної хмарної платформи з урахуванням її сумісності з протоколами зв'язку, спеціалізованими інструментами, підтримкою розподіленої аналітики й інтерфейсів візуалізації.

Аналітичне порівняння низки хмарних платформ для реалізації Internet of Things представлено в оновленому вигляді в таблиці 2.1. У ній враховано характерні протоколи комунікації, домінуючі можливості кожної платформи та типові галузі застосування.

Таблиця 2.1 – Огляд функціоналу хмарних платформ у контексті IoT

Платформа	Протоколи взаємодії	Основне призначення та ключові інструменти	Сфери типового застосування
AWS IoT (Amazon)	HTTP, MQTT, WebSockets	Платформа AWS IoT Core. Аутентифікація, розширене середовище правил	Розумні міста, енергоефективні оселі, агросектор
Google Cloud IoT	MQTT, HTTP	Модуль IoT Core. Підключення пристроїв, віддалений контроль	Транспорт, системи паркування, енергетичні проекти
Azure IoT (Microsoft)	AMQP, MQTT, HTTPS через WebSocket	Інтерфейс IoT Hub. Перевірка доступу, моніторинг стану пристроїв	Сектор охорони здоров'я, роздрібні мережі
Oracle IoT Cloud	MQTT, HTTP	Інтеграція даних, прогнозна аналітика, контроль активів	Промислове виробництво, логістичні системи
IBM Watson IoT	MQTT, HTTP	Система підключення з підтримкою блокчейну та аналітики в реальному часі	Сільське господарство, будівлі з інтелектуальним керуванням
Salesforce IoT	MQTT	Хмарна аналітика, сервісний моніторинг, візуалізація даних	Банківський сектор, фармацевтика, споживчий ринок
ThingsBoard (відкритий код)	MQTT, CoAP, HTTP	Збір та обробка показників, диспетчеризація пристроїв, графічні інтерфейси	Вимірювальні системи, енергоменеджмент, аграрні рішення

Вибір хмарного середовища AWS як базової платформи для реалізації архітектури системи Інтернету речей у рамках моделі гнучкого виробництва зумовлений сукупністю технічних, функціональних і безпекових переваг.

Платформа AWS IoT характеризується високим рівнем масштабованості та адаптивності, що дає змогу централізовано координувати велику кількість гетерогенних пристроїв і забезпечувати оперативне реагування на зміни у виробничому середовищі. Розгалужені можливості налаштування конфігурацій і сценаріїв взаємодії з підключеними об'єктами дозволяють будувати індивідуалізовані рішення, що відповідають специфічним потребам конкретного виробничого процесу.

Суттєвим чинником при виборі AWS виступає дотримання провідних міжнародних стандартів у сфері кібербезпеки. Платформа забезпечує реалізацію багаторівневих механізмів контролю доступу, криптографічного захисту даних, їх цілісності та автентичності, що особливо важливо для розподілених індустріальних систем із критичними вимогами до надійності обміну інформацією.

Додатковою перевагою AWS є розвинена екосистема сервісів, серед яких зокрема варто відзначити інструменти для обробки даних у реальному часі, компоненти для побудови моделей штучного інтелекту, сервіси машинного навчання, інтерфейси візуалізації й модулі прогнозової аналітики. Завдяки безшовній інтеграції між цими компонентами проєктувальники мають можливість створювати комплексні рішення, що поєднують виробничу логіку з інтелектуальною обробкою потоків даних. Це значно знижує часові витрати на розробку та підвищує стабільність і продуктивність кінцевої системи.

Таким чином, опора на AWS як основу для розгортання IoT-інфраструктури в контексті автоматизації гнучкого виробництва є не лише виправданим, а й стратегічно доцільним рішенням, що відкриває шлях до впровадження інноваційних цифрових технологій з гарантованим рівнем надійності, безпеки та функціонального зростання.

2.2 Структура SCADA-системи гнучкого виробництва

Структура SCADA-системи, яка обслуговує гнучке виробництво, є ключовим елементом побудови цифрової фабрики нового покоління. Така архітектура виконує роль інтеграційної платформи, забезпечуючи централізовану координацію виробничих процесів, оперативний моніторинг стану систем, а також взаємодію із широким спектром промислових компонентів – від робототехнічних вузлів до сенсорних мереж і виконавчих механізмів. З огляду на постійно зростаючу складність виробничих систем, архітектура SCADA-системи проектується на засадах модульності, що дає змогу реалізувати поступове нарощування функціоналу шляхом додавання нових підсистем і пристроїв без порушення цілісності загального середовища керування.

Гнучкість архітектури досягається, зокрема, завдяки залученню хмарних обчислень, які дозволяють перенести частину обробки та зберігання даних у віддалене середовище, мінімізуючи навантаження на локальні ресурси. Це забезпечує як підвищену масштабованість, так і гарантований доступ до актуальних даних у режимі реального часу, що є критично важливим для прийняття оперативних управлінських рішень. Центральний диспетчерський вузол системи виконує роль логічного ядра SCADA-інфраструктури, надаючи інтерфейс для візуалізації даних, виявлення відхилень і миттєвого втручання у виробничий цикл.

Інтеграційна взаємодія з промисловими пристроями реалізується за допомогою сумісних протоколів обміну даними, які уніфікують процес комунікації між фізичним обладнанням та програмними компонентами. Зокрема, широко застосовуються протоколи з підтримкою публікації-підписки, що забезпечують асинхронну передачу даних з мінімальною затримкою, зберігаючи структурованість і цілісність інформації на всіх рівнях системи. Це дає змогу впроваджувати розподілену логіку обробки, коли попередня фільтрація та агрегування здійснюються безпосередньо на

периферійних вузлах, а централізоване ядро зосереджується на аналітиці й стратегічному управлінні.

Безпекові механізми архітектури включають багаторівневу автентифікацію користувачів, контроль доступу до окремих сегментів даних та реалізацію алгоритмів криптографічного захисту каналів зв'язку. Такий підхід мінімізує ризики несанкціонованого втручання та забезпечує стабільність функціонування SCADA-системи навіть за умов зовнішнього впливу чи кібератак [18].

Можливість динамічної адаптації архітектури до змін у конфігурації виробництва – одна з її головних переваг. Систему легко доповнити новими програмно-апаратними модулями або пристроями без потреби її повної перебудови, що дає змогу підприємствам оперативно реагувати на зміни попиту або технологічні інновації. У поєднанні з можливостями хмарної платформи така архітектура підтримує стратегії сталого цифрового розвитку, орієнтованого на автоматизацію, автономізацію та максимальну адаптивність виробничого середовища.

Впровадження хмароорієнтованих рішень у SCADA-системи суттєво підвищує їхню гнучкість і адаптивність до змін середовища. Завдяки використанню віддалених обчислювальних ресурсів забезпечується не лише оперативна обробка значних обсягів виробничих даних, а й можливість їх масштабованого зберігання з підтримкою синхронного доступу в режимі реального часу. Це дозволяє адаптувати виробничу інфраструктуру до змін у попиті, логістиці чи технологічних параметрах, зберігаючи при цьому високу ефективність функціонування автоматизованих систем керування.

Натомість автономні SCADA-системи, побудовані за принципом ізольованого функціонування, залишаються актуальними для об'єктів, де централізоване управління не є критично важливим. Такі системи орієнтовані на локальну обробку сигналів, виконання керуючих команд і моніторинг у межах одного виробничого середовища без використання віддалених або хмарних сервісів. Їх ключовою перевагою виступає стабільність

функціонування за умов відсутності доступу до зовнішніх мереж, що знижує вірогідність порушень, пов'язаних із перебоями у підключенні чи кіберзагрозами. Рівень надійності в таких системах традиційно оцінюється як високий через мінімізацію зовнішніх залежностей.

Проте функціональні можливості автономних SCADA-платформ мають природні обмеження. Вони менш пристосовані до ситуацій, які вимагають динамічного масштабування або інтеграції з розширеними інтелектуальними інструментами, такими як хмарна аналітика чи машинне навчання. Такі системи зазвичай використовуються у виробничих середовищах з невисокою інтенсивністю процесів, де обсяг даних, що підлягає обробці, є відносно незначним. Це може бути характерним, наприклад, для лабораторій, дослідницьких майданчиків або малих підприємств з фіксованою топологією автоматизації [19].

Функціональну модель автономної SCADA-системи, що ілюструє її структурну побудову без інтеграції з хмарними платформами, наведено на рисунку 2.1.

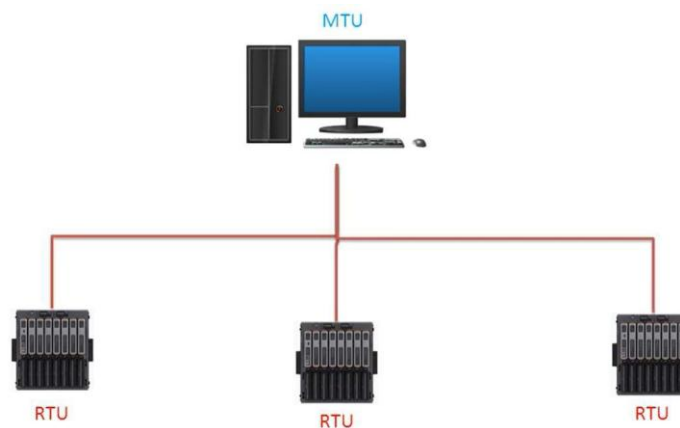


Рисунок 2.1 – Структура автономної SCADA-системи з локалізованим керуванням та моніторингом

Системи SCADA, що функціонують в автономному режимі, знаходять практичне застосування в контекстах, де ключовим чинником виступає гарантована стабільність функціонування, а централізоване управління не

становить пріоритетної потреби. Такі рішення широко використовуються в інфраструктурі малих об'єктів, виробництв із фіксованими топологіями або технологічних комплексів, де критично важливим є забезпечення

Іншою архітектурною моделлю, що забезпечує високий рівень гнучкості, є клієнт-серверна структура SCADA-системи. У межах такого підходу відбувається розподіл функціональних обов'язків між серверними й клієнтськими модулями, де сервер виступає ядром, відповідальним за централізовану обробку, зберігання та агрегацію даних, а клієнти – за локалізований інтерфейс взаємодії з оператором або користувачем. Це дозволяє легко масштабувати систему за рахунок додавання нових клієнтських вузлів без потреби переглядати архітектуру всієї інфраструктури.

Ключова перевага клієнт-серверної SCADA-моделі полягає у централізованому адмініструванні, що дає змогу ефективно координувати інформаційні потоки, централізовано керувати доступом, реалізовувати політики безпеки та оперативно оновлювати системні компоненти. Сервер виконує роль координуючого центру, що обробляє великі обсяги даних у реальному часі, забезпечуючи узгодженість інформації для всіх клієнтських терміналів.

Гнучкість клієнт-серверної моделі також полягає в її здатності швидко адаптуватися до змін у конфігурації виробництва або до потреб конкретного підприємства. Наприклад, у випадку розширення виробничих потужностей достатньо додати нові клієнтські інтерфейси, що взаємодіють з центральним сервером, не порушуючи загальної архітектури. Аналогічно, видалення окремих клієнтських вузлів або переадресація потоків даних реалізується безперервно, що підвищує гнучкість реагування системи на зовнішні та внутрішні зміни.

Архітектурну модель SCADA-системи, побудованої на клієнт-серверному принципі, наведено на рисунку 2.2.

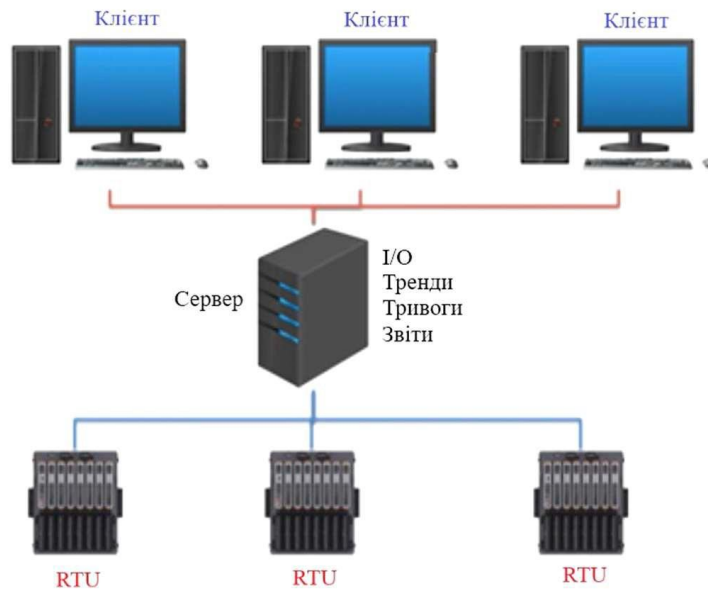


Рисунок 2.2 – Схематична побудова SCADA-системи на основі клієнт-серверної архітектури

Перевагою клієнт-серверної архітектури SCADA-систем є спрощена процедура оновлення та розширення функціоналу, що значно відрізняє її від автономних рішень. У такій структурі всі ключові зміни впроваджуються на рівні центрального сервера, після чого автоматично розповсюджуються на клієнтські вузли без необхідності індивідуального втручання до кожного компонента. Це забезпечує швидку адаптацію системи до нових технологічних вимог або змін у виробничих процесах.

Централізований характер клієнт-серверної архітектури сприяє зручній інтеграції з іншими цифровими системами підприємства, зокрема з модулями ERP, MES або базами даних історичних даних. Сервер у такій моделі виступає як єдина точка входу, через яку реалізується двосторонній обмін даними, що спрощує впровадження функціональності міжсистемної взаємодії, зменшуючи часові й ресурсоємні витрати на сумісність.

Стійкість до відмов у клієнт-серверних системах залежить передусім від того, наскільки ефективно реалізовані інженерні механізми резервування на рівні обчислювальних потужностей сервера та мережевої інфраструктури. У разі втрати зв'язку або відмови основного сервера, система має

перемикатися на резервний вузол без зупинки процесів моніторингу чи управління [20]. Такий підхід дозволяє забезпечити безперервність функціонування навіть за умов зовнішніх збоїв або апаратних несправностей.

Вищим ступенем реалізації цієї моделі є клієнт-серверна SCADA-система з повноцінною схемою резервування, в якій паралельно функціонують основний і дублюючий сервери. Це дає змогу забезпечити як горизонтальну масштабованість системи – через додавання нових клієнтських терміналів або виробничих вузлів, так і вертикальну – шляхом посилення обчислювальних потужностей або підвищення аналітичної складності системи. У межах такої архітектури зберігається централізований контроль, однак система отримує здатність до самовідновлення та адаптації до зовнішніх і внутрішніх факторів без зниження продуктивності.

Гнучкість клієнт-серверної SCADA-системи з резервуванням проявляється в здатності оперативно вносити зміни, оновлювати конфігурацію, адаптувати сценарії керування та реагувати на зростаючі навантаження без переривання поточних технологічних процесів. Таке поєднання централізованого адміністрування з високим рівнем відмовостійкості формує архітектурну основу для промислових систем нового покоління.

Функціональна схема такої SCADA-системи наведена на рисунку 2.3.

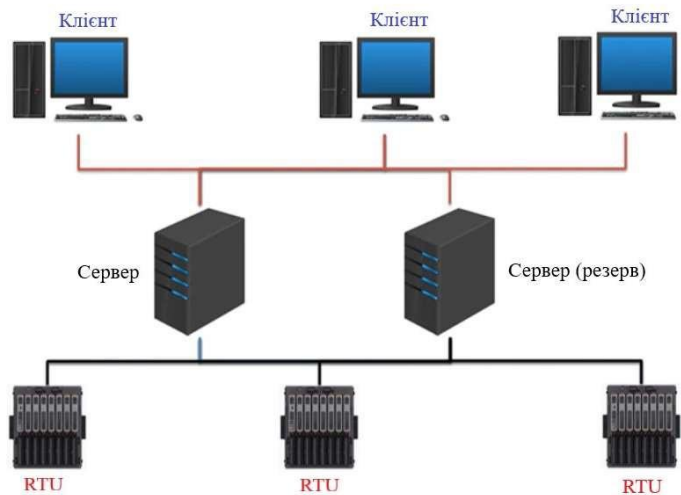


Рисунок 2.3 – Архітектура клієнт-серверної SCADA-системи з реалізованими механізмами резервування

Модульно орієнтована SCADA-система є прикладом архітектури, що побудована за принципами масштабованості, адаптивності та структурної гнучкості. Такий підхід передбачає розподіл функціональності системи на окремі, логічно незалежні блоки – модулі, кожен із яких відповідає за певний набір завдань, таких як моніторинг, обробка даних, візуалізація, управління або безпека. Основною перевагою модульної архітектури є можливість оперативного розширення або скорочення функціонального складу SCADA-системи без потреби в повній перебудові її структури. Це дозволяє адаптувати систему до змін у виробничих процесах або впровадження нових технологічних засобів.

Масштабованість модульної SCADA-системи реалізується шляхом додавання нових функціональних компонентів, які взаємодіють із базовим середовищем за узгодженими інтерфейсами. Такий підхід дозволяє не лише нарощувати обчислювальні ресурси чи розширювати діапазон керованих пристроїв, але й забезпечувати поступову модернізацію, знижуючи ризики, пов'язані з технічним або фінансовим перевантаженням.

Надійність цієї архітектури багато в чому залежить від правильності реалізації комунікаційних зв'язків між модулями. Висока якість взаємодії

досягається шляхом ретельного проектування інтерфейсів обміну даними, уніфікації протоколів та використання механізмів синхронізації. Особливої уваги потребує впровадження механізмів взаємного контролю між модулями, що дозволяє вчасно виявляти збої й автоматично їх компенсувати шляхом перемикавання на резервні компоненти.

Гнучкість модульної архітектури полягає в її здатності оперативно адаптуватися до нових вимог без порушення цілісності системи. Кожен модуль може бути вдосконалений або замінений без суттєвого впливу на роботу всієї інфраструктури. Водночас, з метою забезпечення стабільності, модулі проектуються з урахуванням принципів взаємної сумісності та можливості інтеграції до загальної системної платформи.

Відмовостійкість модульної SCADA-системи прямо пов'язана з наявністю внутрішніх резервів: дублюванням критичних модулів, реалізацією алгоритмів гарячого перемикавання, а також застосуванням моніторингових засобів для виявлення та локалізації збоїв. За умови належного проектування така архітектура здатна забезпечити високу стійкість до відмов, зберігаючи безперервність обробки даних та управління технологічними процесами.

Структурну схему модульної SCADA-системи наведено на рисунку 2.4.

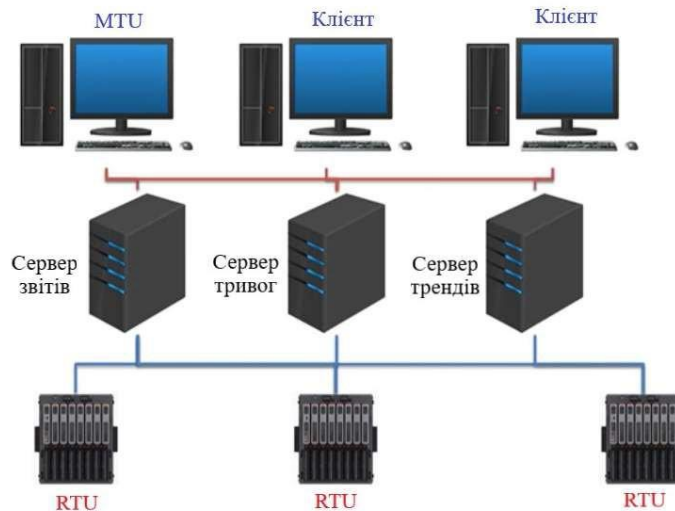


Рисунок 2.4 – Архітектура модульної SCADA-системи з функціональною сегментацією та резервуванням

Архітектурна модель SCADA-системи, реалізованої на основі технологій Інтернету речей (IoT), становить сучасний підхід до побудови інтегрованих промислових платформ керування. Вона поєднує традиційні засоби автоматизації з цифровими IoT-компонентами, формуючи розподілене середовище з високим рівнем масштабованості, адаптивності та доступності. Такий тип архітектури дозволяє створити замкнений цикл: від збору даних на фізичному рівні до аналітичної обробки та прийняття рішень у реальному часі.

Ключовими елементами SCADA-інфраструктури в цьому випадку виступають сенсорні пристрої та актуатори, які виконують роль первинних джерел інформації безпосередньо у виробничому середовищі. Вони фіксують параметри навколишнього середовища та технологічного процесу – наприклад, рівень вологості, тиск, температуру, швидкість обертання тощо – і передають ці дані у форматі, сумісному з протоколами IoT-комунікації. До таких протоколів, які забезпечують ефективний і стандартизований обмін інформацією, належать MQTT, CoAP та HTTP/HTTPS, що дозволяють оптимізувати передачу даних навіть у умовах обмеженої пропускної здатності мережі.

Центральний обчислювальний вузол у такій архітектурі здебільшого реалізується на базі хмарної інфраструктури (AWS, Google Cloud, Azure тощо). Він виконує функції централізованої обробки, зберігання й візуалізації даних, отриманих від польових пристроїв. Серверна частина також відповідає за реалізацію інтерфейсу SCADA-системи, забезпечуючи засоби моніторингу в реальному часі, диспетчеризації, аналітики й управління. Для збереження історичних даних і підтримки функцій довготривалого аналізу використовується окрема база даних, синхронізована з основним сервером.

Інтеграція SCADA/IoT-системи з корпоративними інформаційними платформами, такими як ERP, дозволяє реалізувати наскрізну логіку управління – від оперативного контролю за станом обладнання до формування управлінських звітів і прийняття бізнес-рішень на рівні підприємства. Це забезпечує міжрівневу взаємодію між технічним і адміністративним сегментами, сприяючи ефективному функціонуванню виробничої екосистеми.

Безпека такої архітектури досягається за рахунок впровадження комплексного підходу: шифрування даних під час передачі, автентифікації IoT-пристроїв, контролю доступу та верифікації запитів. Це дозволяє захистити критичні виробничі дані від несанкціонованого доступу чи втручання ззовні [21].

Таким чином, SCADA-система, побудована на основі IoT, є гнучким і стійким рішенням для управління складними розподіленими виробництвами. Вона забезпечує повноцінну інформаційну прозорість процесів, оперативне реагування на зміни та стратегічну готовність до цифрової трансформації.

Схему взаємодії компонентів такої системи представлено на рисунку 2.5.

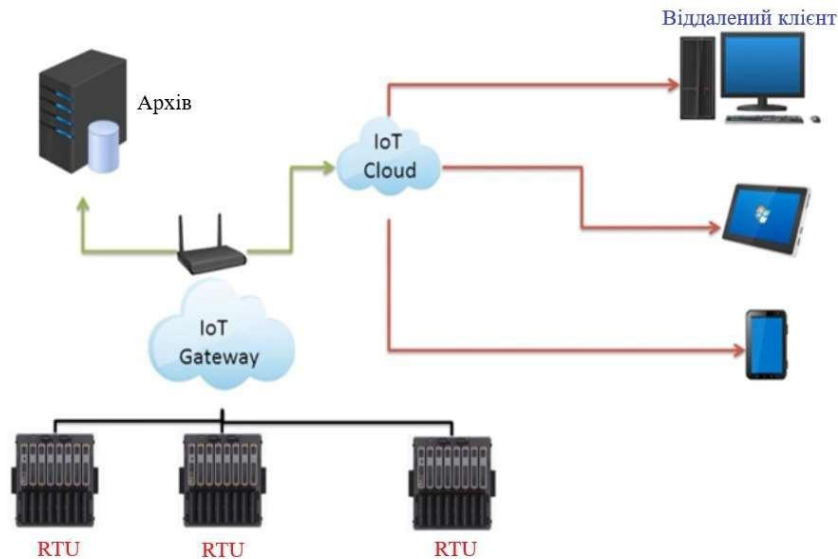


Рисунок 2.5 – Архітектура SCADA-системи з підтримкою технологій Інтернету речей

Застосування архітектури SCADA-системи, побудованої з урахуванням технологій Інтернету речей, відкриває суттєво розширені можливості для промислового управління та цифрової трансформації виробничого середовища. Така система дозволяє не лише фіксувати стан технологічних процесів у реальному часі, а й активно керувати ними, забезпечуючи високий ступінь реактивності на змінні умови, оперативно адаптуючи стратегії керування відповідно до змін у навантаженні, параметрах виробничого циклу чи зовнішніх впливах. Завдяки здатності до глибокої оптимізації ресурсів, ця архітектура формує інтелектуальну основу для розвитку адаптивних і самоналаштовуваних систем керування, що є критично важливим у динамічних умовах сучасної промисловості.

Обґрунтування вибору SCADA-рішень із вбудованими можливостями IoT базується на кількох принципових перевагах. Насамперед, така система забезпечує централізований контроль над розподіленими технологічними об'єктами з можливістю дистанційного доступу, що істотно розширює межі керованості, знижує залежність від фізичної присутності персоналу на об'єктах і підвищує рівень мобільності операційного управління.

Іншою важливою перевагою є здатність таких систем надавати вичерпну аналітику в реальному часі. Завдяки постійному обміну інформацією між сенсорами, виконавчими механізмами та обчислювальними вузлами, створюється суцільне інформаційне поле, яке дозволяє здійснювати аналіз на основі актуальних даних, формувати прогностичні моделі та приймати управлінські рішення на основі об'єктивної аналітики.

Крім того, інтеграція IoT-компонентів у SCADA-середовище сприяє підвищенню загальної ефективності системи за рахунок автоматизації рутинних операцій, самодіагностики компонентів і мінімізації часу реагування на критичні події. Це також покращує показники надійності, оскільки зменшується людський фактор, а контроль здійснюється за чітко визначеними алгоритмами.

Таким чином, вибір SCADA-системи, що поєднує принципи традиційної автоматизації та можливості Інтернету речей, виступає обґрунтованим рішенням для реалізації складних виробничих і інфраструктурних завдань, де критично важливими є оперативність, гнучкість і здатність до масштабованого розвитку.

2.3 Проєктування алгоритмічної та програмної складової SCADA-системи гнучкого виробництва

З метою реалізації ефективної SCADA-системи, що відповідає вимогам гнучкого виробництва, необхідно сформулювати узагальнений алгоритм функціонування модуля моніторингу. Основою такого алгоритму є послідовна взаємодія мікроконтролера з мережевою інфраструктурою та хмарними сервісами, зокрема з платформою Amazon Web Services (AWS), яка використовується як обчислювальне та комунікаційне середовище.

На початковому етапі мікроконтролер повинен ініціалізувати з'єднання з мережею WI-FI, що має стабільний вихід до глобальної мережі. Всі необхідні технічні передумови – зокрема наявність джерела живлення з

напругою 5В та сумісного мережевого адаптера – вважаються виконаними. Після встановлення з'єднання мікроконтролер здійснює передачу зібраних сенсорних даних за допомогою протоколу MQTT, який є оптимізованим для середовищ з обмеженою пропускну здатністю.

MQTT (Message Queuing Telemetry Transport) застосовується як легкий транспортний протокол у системах IoT, що функціонують у нестабільних мережах або з низьким енергоспоживанням. Його модель публікації/підписки дозволяє створити ефективну архітектуру, у якій пристрої можуть передавати дані у вигляді повідомлень, асоційованих із певними топіками, а інші компоненти системи – підписуватися на ці топіки для отримання відповідної інформації.

У запропонованій реалізації мікроконтролер налагоджує зв'язок із AWS IoT Core – спеціалізованим сервісом платформи AWS, орієнтованим на приймання, обробку та маршрутизацію даних від IoT-пристроїв. Передача даних здійснюється через механізм так званих «тіней» (shadow topics), що дозволяє зберігати поточний стан пристрою навіть у разі тимчасової втрати з'єднання або недоступності вузла.

Сервіс AWS IoT Device Shadow реалізує концепцію віртуального подання пристрою у вигляді JSON-структури, що синхронізується з фізичним об'єктом. Це дозволяє стороннім застосункам отримувати або оновлювати стан пристрою незалежно від його наявності в мережі. Кожен пристрій може мати кілька ізольованих тіней, що підвищує гнучкість у сценаріях взаємодії з різними сервісами або програмними інтерфейсами.

Загальна архітектурна схема реалізації взаємодії між мікроконтролером, протоколом MQTT, сервісом AWS IoT Core та його підсистемою Device Shadow представлена на рисунку 2.6.

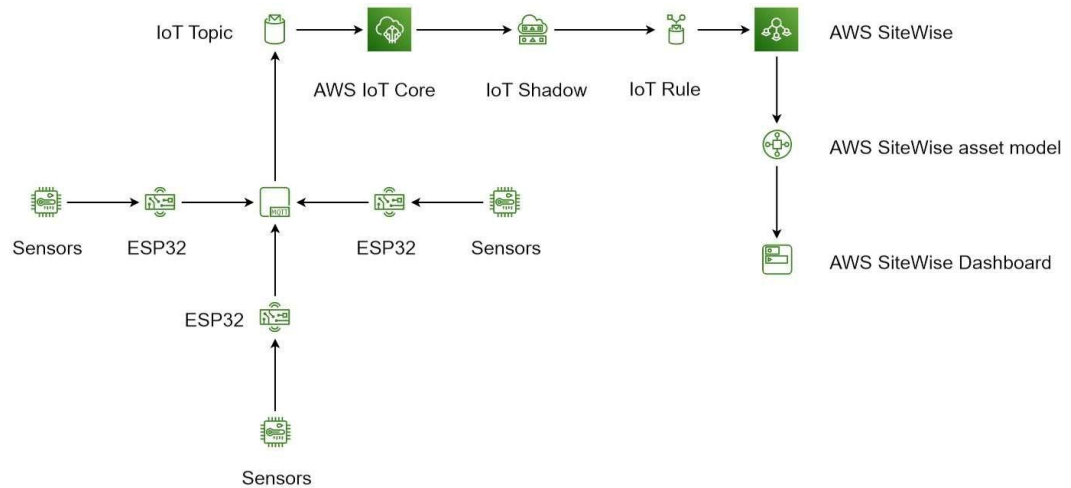


Рисунок 2.6 – Інтегрована схема функціонування SCADA-системи адаптивного виробництва з використанням AWS IoT

Компонент AWS IoT Core Rule відіграє ключову роль у формуванні логіки обробки даних у SCADA-системі, орієнтованій на гнучке виробництво, забезпечуючи автоматичну реакцію на зміни стану пристроїв, що передають дані через MQTT-протокол. У разі оновлення тіньового відображення (device shadow) будь-якого пристрою, система AWS генерує MQTT-повідомлення, яке миттєво обробляється правилом, що налаштоване на виявлення відповідних подій.

Коли відбувається зміна в структурі тіні, сервіс AWS IoT Core Rule ініціює процедуру вилучення ключових значень параметрів – зокрема, температури, вібрації, тиску, напруги або інших показників, – і спрямовує ці значення до віртуальних моделей пристроїв, реалізованих у середовищі AWS IoT SiteWise. Кожен пристрій, з якого надходять дані, представлений у SiteWise як актив, що належить до певного типу. Таке структурування дозволяє забезпечити семантичну узгодженість між даними, що надходять з різних джерел, але мають однакову функціональну природу.

Моделі активів, побудовані в AWS IoT SiteWise, служать основою для інтегрованої обробки інформації та побудови шаблонів для агрегування й порівняльного аналізу. Завдяки цьому можливо не лише спостерігати за змінами параметрів у межах одного пристрою, а й отримувати аналітику

щодо сукупностей пристроїв одного типу. Зібрані в такий спосіб дані надходять до інтерфейсу візуалізації SiteWise Dashboard, що дозволяє формувати інформативні панелі контролю та аналітичні графіки в реальному часі. Крім того, вони можуть бути інтегровані з іншими сервісами AWS для реалізації розширених функцій – від прогнозної діагностики до автоматизованого прийняття рішень.

Алгоритм взаємодії між мікроконтролером, тіньовими структурами, MQTT-комунікацією, правилами обробки та візуалізацією у SCADA-системі, орієнтованій на гнучке виробництво, зображено на рисунку 2.7.

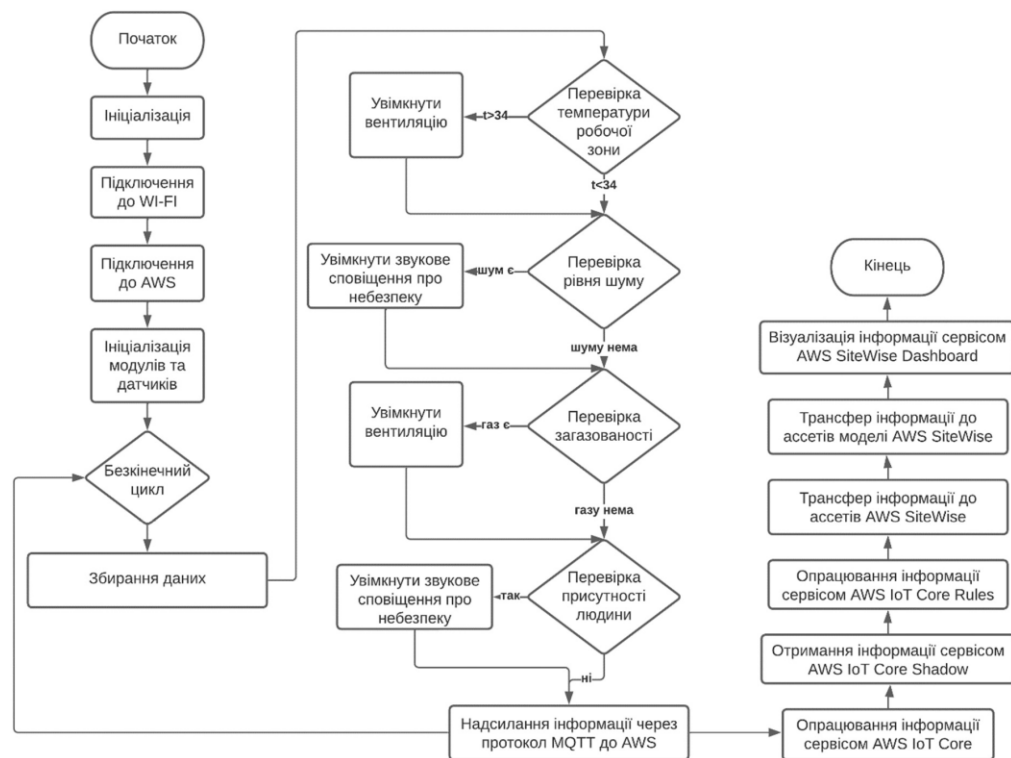


Рисунок 2.7 – Алгоритмічна схема функціонування SCADA-системи орієнтованої на гнучке виробництво з використанням AWS IoT Core і SiteWise

3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПОНЕНТІВ SCADA-СИСТЕМИ ОРІЄНТОВАНОЇ НА ГНУЧКЕ ВИРОБНИЦТВО З ІНТЕГРАЦІЄЮ ХМАРНОЇ ІНФРАСТРУКТУРИ

3.1 Розроблення програмного забезпечення SCADA-системи

У процесі проєктування програмного забезпечення для збору і передавання даних з мікроконтролера ESP32 у SCADA-системі, орієнтованій на гнучке виробництво, особливу увагу слід приділити оптимізації витрат енергії та мережевого трафіку. Оскільки комунікація здійснюється через бездротову мережу Wi-Fi, важливо мінімізувати обсяг переданих даних і забезпечити використання енергоощадних режимів роботи пристрою. Для досягнення цих цілей доцільно обирати легковагі протоколи взаємодії та компоненти з низьким енергоспоживанням на апаратному рівні.

В якості основного транспортного протоколу обрано MQTT – ефективний механізм публікації-підписки, що добре зарекомендував себе в середовищах із обмеженими ресурсами. Для реалізації передачі телеметричних даних між ESP32 і хмарними сервісами Amazon Web Services (AWS) у межах даного проєкту використовується бібліотека PubSubClient, яка надає інтерфейс для публікації повідомлень у топіки та підписки на них.

У рамках реалізації системи дані, зібрані з периферійних сенсорів, формуються в об'єкті типу `'StaticJsonDocument'`. Структурований JSON-документ передається до заданої теми через модуль AWS IoT Core, де він обробляється для подальшого збереження чи аналізу. На рисунку 3.1 наведено приклад програмної реалізації функції, що виконує формування повідомлення у форматі JSON і здійснює його публікацію у відповідний MQTT-топик.

```

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 та його публікації

Із міркувань інформаційної безпеки та зручності супроводу коду доцільно винести всі конфіденційні параметри, необхідні для встановлення захищеного з'єднання, в окремий заголовковий файл. До таких параметрів належать: реквізити доступу до бездротової мережі Wi-Fi, доменне ім'я кінцевої точки AWS IoT Core, а також усі криптографічні компоненти, які забезпечують автентифікацію та шифрування з'єднання за протоколом SSL/TLS. Зокрема, це кореневий сертифікат Amazon Trust Services, унікальний сертифікат, що прив'язаний до конкретного пристрою, і приватний ключ, необхідний для генерації цифрового підпису під час встановлення сеансу зв'язку.

Винесення цієї інформації до окремого `.h` файлу дозволяє ізолювати змінні, що містять критичні облікові дані, від основної логіки програми, тим самим спрощуючи оновлення параметрів та підвищуючи безпеку. У рамках структури SCADA-системи такий підхід відповідає принципам інкапсуляції конфігураційних даних, забезпечуючи надійність взаємодії з хмарними сервісами.

також передбачає обробку помилок підключення, що дозволяє гарантувати працездатність системи навіть у випадку відмови сенсора.

```

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 – Процедура ініціалізації сенсора BMP280 у середовищі програмного контролю

Після успішної ініціалізації та налаштування сенсора BMP280 у середовищі програмного забезпечення SCADA-системи, його подальше використання зводиться до періодичного зчитування показників температури та тиску. Отримані значення можуть бути інтегровані в загальний оброблюваний набір даних, який надалі передається через MQTT до відповідної теми у хмарному сервісі.

У типовій реалізації сенсор опитується у циклі, або у відповідь на певну подію (наприклад, у задані інтервали часу чи при запуску переривання). Зчитані значення, після базової обробки, можуть бути записані в структуру JSON, з якої формується MQTT-повідомлення. Такий підхід дозволяє забезпечити ефективний збір телеметричної інформації та її подальше використання для прийняття рішень у межах автоматизованої системи виробничого моніторингу.

```

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 – Приклад виклику функції зчитування даних з сенсора BMP280 та обробки результату

Для виявлення потенційної загрози, пов'язаної з наявністю горючих газів у середовищі гнучкого виробництва, у SCADA-системі реалізовано підсистему моніторингу з використанням сенсора MQ2. Цей датчик є газоаналізатором, що працює за принципом цифрового виводу логічного сигналу: при перевищенні порогового значення концентрації горючих газів (наприклад, пропану, метану або водню) він подає логічну одиницю на відповідний цифровий пін мікроконтролера.

Для інтеграції MQ2 у програмну архітектуру передбачено механізм опитування його цифрового виходу. У разі, якщо за результатами зчитування пін має високий логічний рівень, це інтерпретується як сигнал тривоги, що вказує на наявність газу у навколишньому середовищі. У відповідь на це активується система примусової вентиляції – зокрема, здійснюється керування відповідним виконавчим елементом (наприклад, реле або транзистором), який вмикає вентилятор для зниження концентрації небезпечних речовин.

```

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 – Реалізація логіки зчитування сигналу з датчика MQ2 та керування вентиляцією у разі виявлення газу

У контексті побудови інтелектуальної системи безпеки в межах SCADA-системи, орієнтованої на гнучке виробництво, доцільним є застосування сенсорів, здатних реагувати на присутність людини у контрольованій зоні. Зокрема, сенсори HLK-LD2410C (радарного типу) та KY-037 (акустичний детектор звуку), а також виконавчі пристрої – баззер та вентилятор – інтегруються у загальну архітектуру аналогічно до вже розглянутих рішень.

Принцип дії зазначених компонентів полягає в тому, що кожен із сенсорів передає логічний сигнал про виявлення активності на окремий цифровий пін. Цей сигнал далі опрацьовується у програмному забезпеченні. У випадку фіксації присутності – будь-то рух, шум або інший прояв активності – здійснюється відповідна реакція з боку виконавчих механізмів. Наприклад, може вмикатися світлова або звукова індикація (через баззер), запускатися система вентиляції або надсилатися повідомлення у хмару для подальшої обробки.

Алгоритм взаємодії з сенсором HLK-LD2410C, представлений на рисунку 3.6, передбачає циклічне зчитування стану цифрового входу та виконання логіки реагування у разі підтвердженого виявлення присутності людини в зоні обслуговування. Це дозволяє підвищити рівень безпеки

працівників, а також автоматизувати відповідні дії в рамках виробничого процесу.

```
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 присутності людини у робочій зоні за допомогою сенсорів та керування виконавчими пристроями

3.2 Розроблення макету модуля збору технологічних параметрів гнучкого виробництва

На етапі розроблення апаратного компонента SCADA-системи, орієнтованої на гнучке виробництво, критично важливим є формування коректної топології підключення периферійних пристроїв до мікроконтролера. Вибір схеми з'єднань безпосередньо впливає на стабільність функціонування пристроїв, сумісність між модулями та точність обміну даними в системі. Грамотно спроектована схема дозволяє забезпечити узгодженість електричних характеристик, оптимізувати розведення ліній живлення, мінімізувати перешкоди та запобігти перевантаженням за струмом або напругою.

Схема підключення визначає логіку та пріоритет взаємодії між сенсорами, виконавчими механізмами та контролером, а також окреслює ключові зв'язки, які зумовлюють конфігурацію каналів обробки сигналів. Це забезпечує формування чіткої керувальної стратегії, де кожен компонент

інтегровано відповідно до своєї функціональної ролі: вимірювання параметрів, передача даних або реакція на події.

Окрім забезпечення ефективної передачі сигналів, правильна схема підключення дозволяє проводити структурну діагностику на етапі розробки: з її допомогою легко виявити й усунути потенційні конфлікти, проблеми сумісності або помилки підключення. Це підвищує надійність функціонування системи в реальних умовах експлуатації та забезпечує можливість швидкого масштабування або модифікації макета.

Для реалізації апаратного прототипу було використано мікроконтролер ESP32, до якого підключено всі компоненти, розглянуті у попередніх підрозділах. Візуалізація схеми підключення наведена на рисунку 3.7. Із неї видно, що подача живлення до контролера здійснюється через microUSB-роз'єм, приєднаний до джерела постійного струму напругою 5 В. Більшість сенсорів та модулів отримують живлення з лінії 5 В, тоді як BMP280 працює від окремої шини з напругою 3,3 В. Важливим є також правильне призначення цифрових вхідно-вихідних контактів для забезпечення стабільної взаємодії з периферією.

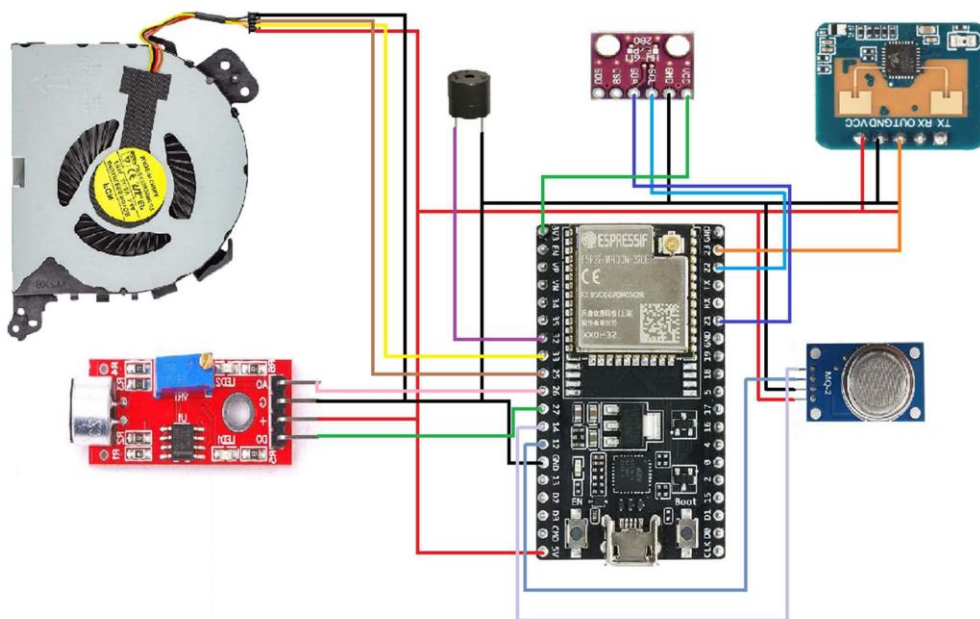


Рисунок 3.7 – Схема електричних з'єднань між мікроконтролером ESP32 та зовнішніми компонентами

Для спрощення збірки була використана універсальна макетна плата, що дозволила швидко організувати з'єднання без пайки. Цей підхід знижує ризик пошкодження компонентів на етапі тестування та дозволяє вносити гнучкі зміни у конфігурацію. Зовнішній вигляд такої плати представлено на рисунку 3.8.

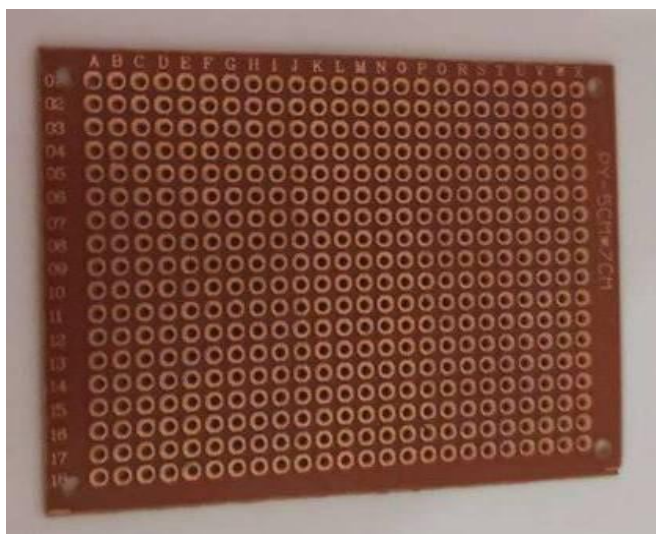


Рисунок 3.8 – Макетна плата, використана для збирання електронних компонентів

Фінальне компонування усіх елементів системи на макеті, з дотриманням електричних з'єднань і логіки функціонального розміщення, наведене на рисунку 3.9.

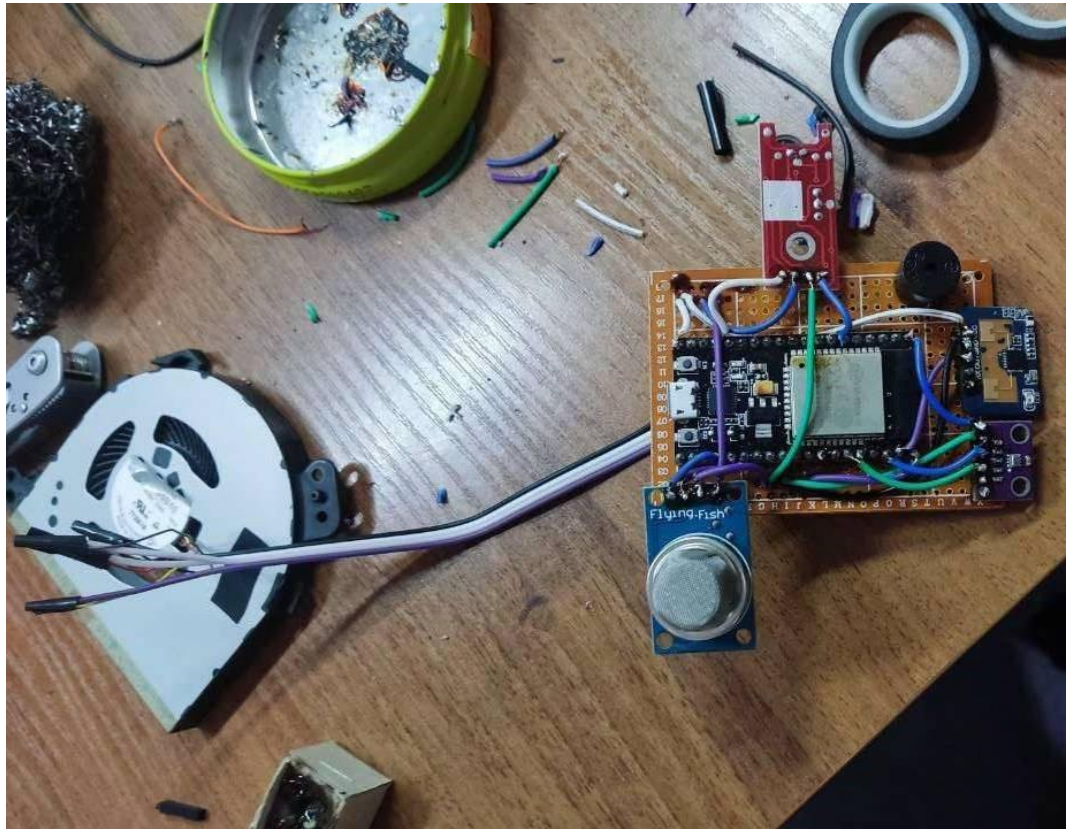


Рисунок 3.9 – Готовий макет апаратного модуля для збору технологічних параметрів у SCADA-системі

Як видно з ілюстрації, наведеної на рисунку 3.9, до мікроконтролера ESP32 підключено низку сенсорних та виконавчих компонентів, які забезпечують комплексний збір і обробку технологічної інформації у межах SCADA-системи орієнтованої на гнучке виробництво. Зокрема, у складі апаратного макета використано вентилятор Lenovo 5F10N82225, що виконує функцію реактивного елемента системи керування у разі виявлення небезпеки, барометр BMP280 для моніторингу атмосферного тиску й температури, сенсор присутності HLK-LD2410C, здатний виявляти об'єкти у зоні дії за допомогою радарної технології, газоаналізатор MQ2 для

визначення концентрації горючих газів у повітрі, а також мікрофонний модуль КУ-037, що фіксує акустичні коливання в середовищі.

Кожен з перелічених компонентів виконує специфічну функцію в рамках єдиної інформаційно-керуючої структури, забезпечуючи не лише вимірювання відповідних параметрів, але й активацію виконавчих механізмів на основі визначених умов чи порогів. Така модульна організація дозволяє масштабувати систему залежно від цілей конкретного виробничого процесу та забезпечує адаптивність до змін у середовищі функціонування.

4 ТЕСТУВАННЯ МАКЕТУ СИСТЕМИ ЗБОРУ ТЕХНОЛОГІЧНИХ ПАРАМЕТРІВ ГНУЧКОГО ВИРОБНИЦТВА

4.1 Формулювання завдань дослідницького експерименту

У межах цього етапу проєкту сформульовано завдання щодо перевірки функціональної спроможності створеного стенду, що відтворює компоненти SCADA-системи, орієнтованої на гнучке виробництво, яка розгортається з використанням хмарної інфраструктури AWS. Верифікація функціоналу здійснюється з використанням методології практичного тестування, основаної на застосуванні прототипу, а також із залученням інструментарію компаративного аналізу для зіставлення результатів.

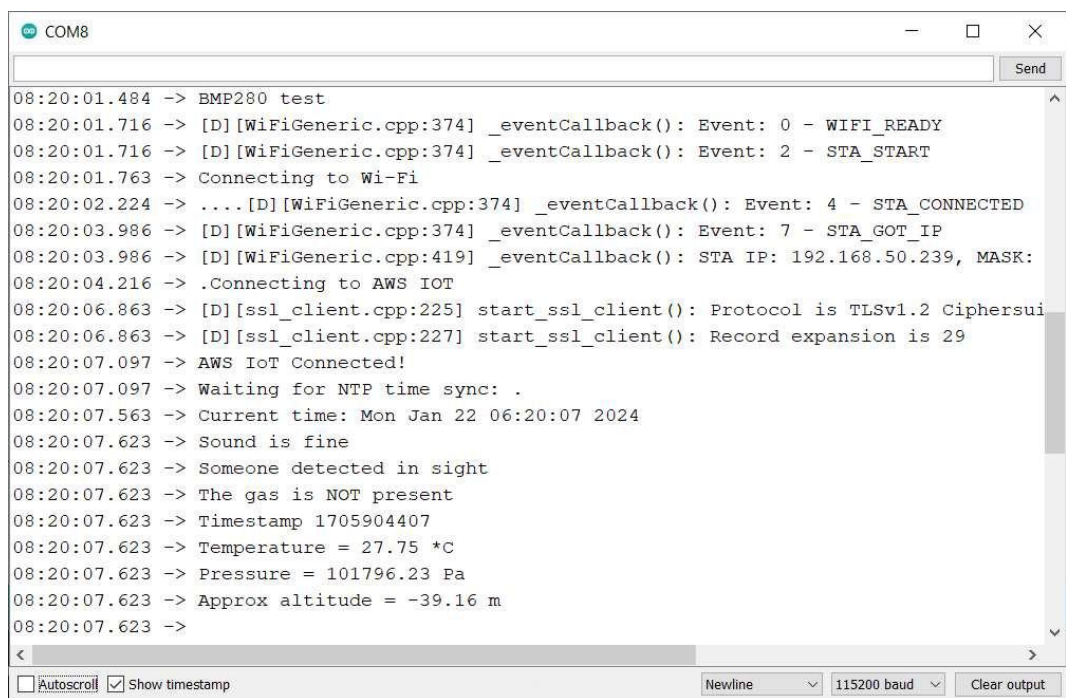
Очікуваним результатом проведеного випробування є підтвердження здатності хмарних модулів до коректної обробки та відображення інформації, отриманої безпосередньо з фізичних сенсорів, інтегрованих у стенд. У контексті реалізації експерименту передбачено досягнення таких прикладних цілей: створення архітектури та розробка програмної логіки SCADA-рішення, адаптованого до умов змінного виробничого навантаження; оптимізація способу передачі даних з опорою на комунікаційний протокол MQTT; конструювання модуля збору параметрів виробничого середовища; реалізація системи електроживлення з вихідною напругою 5 В; документування всього процесу випробувань за допомогою засобів відеофіксації.

4.2 Реалізація експериментального дослідження

У процесі створення лабораторного стенду відповідно до технічного опису виникла необхідність у багатоступеневій перевірці його ефективності в умовах реального функціонування. Для досягнення поставленої мети було проведено низку експериментальних дій, які дозволили оцінити стабільність роботи окремих функціональних вузлів. Зокрема, дослідницький стенд було

під'єднано до локальної бездротової мережі Wi-Fi, а також забезпечено автономним джерелом живлення з номінальною напругою 5 В. Усі етапи експерименту супроводжувалися проміжною перевіркою працездатності системи.

Після подачі електроживлення макет ініціює з'єднання з хмарним середовищем AWS IoT Core, про що надходить відповідне повідомлення через інтерфейс серійного порту. У подальшому каналом цього ж порту здійснюється передача зчитаних сенсорних даних у форматі текстового відображення. Серед задіяних сенсорів – модуль BMP280, який вимірює значення температури, атмосферного тиску та вологості повітря; модуль MQ2, що реагує на присутність легкозаймистих речовин у навколишньому середовищі; сенсор KY-037, відповідальний за аналіз інтенсивності звукового фону; а також пристрій HLK-LD2410C, здатний виявляти присутність людини в контрольованій зоні. Результати зчитування проілюстровані на рисунку 4.1.



```
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 – Виведення показників сенсорів у середовищі серійного монітора

Як видно на рисунку, зафіксовано виявлення присутності людини, рівень звукового тиску перебуває в межах допустимого фону, шкідливі гази виявлені не були. Температурне значення становить приблизно 28°C, атмосферний тиск наближається до 101800 Па, а розрахункова висота над рівнем моря вказує на зниження – близько 40 м. Усі ці дані підтверджують правильне функціонування сенсорних компонентів та їхню здатність до передачі інформації в режимі реального часу.

Однією з ключових умов успішної реалізації SCADA-системи, орієнтованої на гнучке виробництво, є перевірка стабільності та достовірності передачі інформації до хмарної інфраструктури. З цією метою було здійснено доступ до облікового запису у середовищі Amazon Web Services, де у межах сервісу AWS IoT Core проведено верифікацію обміну повідомленнями між сенсорними модулями та хмарним брокером за допомогою вбудованого MQTT-клієнта.

Для підтвердження правильності передавання даних було використано механізм тестування публікацій та підписок у реальному часі, доступний безпосередньо через інтерфейс AWS. Отримані повідомлення, сформовані внаслідок роботи сенсорних елементів і передані на відповідний топік, наочно відображаються у середовищі тестового клієнта. Візуалізація цих повідомлень подана на рисунку 4.2.



Рисунок 4.2 – Приклад вхідних повідомлень у тестовому MQTT-клієнті середовища AWS IoT Core

Аналіз отриманих записів підтвердив, що структура переданої інформації відповідає передбаченому формату, а сама передача відбувається у режимі реального часу без втрат чи затримок. Це свідчить про коректне функціонування системи взаємодії між пристроєм збору даних і хмарними компонентами інфраструктури, що є критично важливим для динамічних умов гнучкого виробництва.

Передані в середовище AWS IoT Core дані далі автоматизовано спрямовуються до сервісу AWS SiteWise через спеціально налаштоване правило обробки подій, реалізоване на рівні AWS IoT Core Rule. Механізм такого трансферу ґрунтується на прив'язці до певного MQTT-топіка, зокрема «\\$aws/things/esp32/shadow/update/accepted», з якого відбувається вибірка інформації, що надійшла від пристрою. Базову конфігурацію правила, що визначає логіку цієї маршрутизації, відображено на рисунку 4.3.

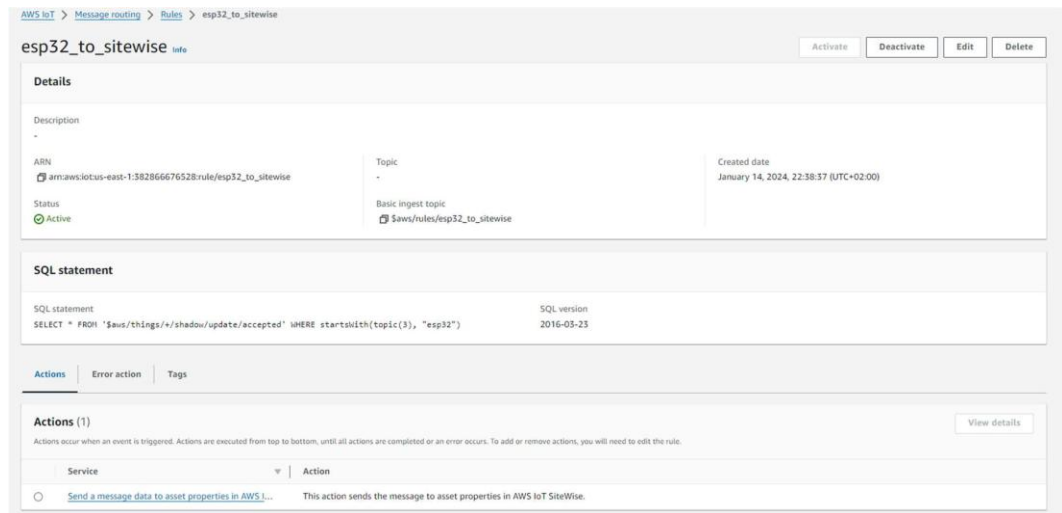


Рисунок 4.3 – Конфігураційне правило для переспрямування даних між сервісами AWS

На етапі надходження даних до AWS SiteWise застосовується попередньо створена модель асетів, яка виконує структурування переданої інформації відповідно до набору визначених параметрів. Ці параметри виступають як атрибути агрегування, що дозволяють ефективно впорядкувати потік даних за критеріями технологічної доцільності. Конфігурацію моделі асетів, яка виступає шаблоном класифікації, представлено на рисунку 4.4.

Attributes (0)

Measurements (6)

Transforms (0)

Metrics (2)

Components (0)

Measurements (6)

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 – Атрибутивна структура асет-моделі у системі AWS SiteWise

На основі цієї структури формується візуальне уявлення про зміну контрольованих параметрів у вигляді інтерактивних графічних компонентів, що реалізуються за допомогою інструмента AWS IoT SiteWise Dashboard. Даний інтерфейс дозволяє не лише моніторити поточні значення параметрів у реальному часі, а й здійснювати їхній фільтр за часовими інтервалами, що критично важливо для ретроспективного аналізу виробничих процесів. Візуалізовані дані представлено на рисунку 4.5.

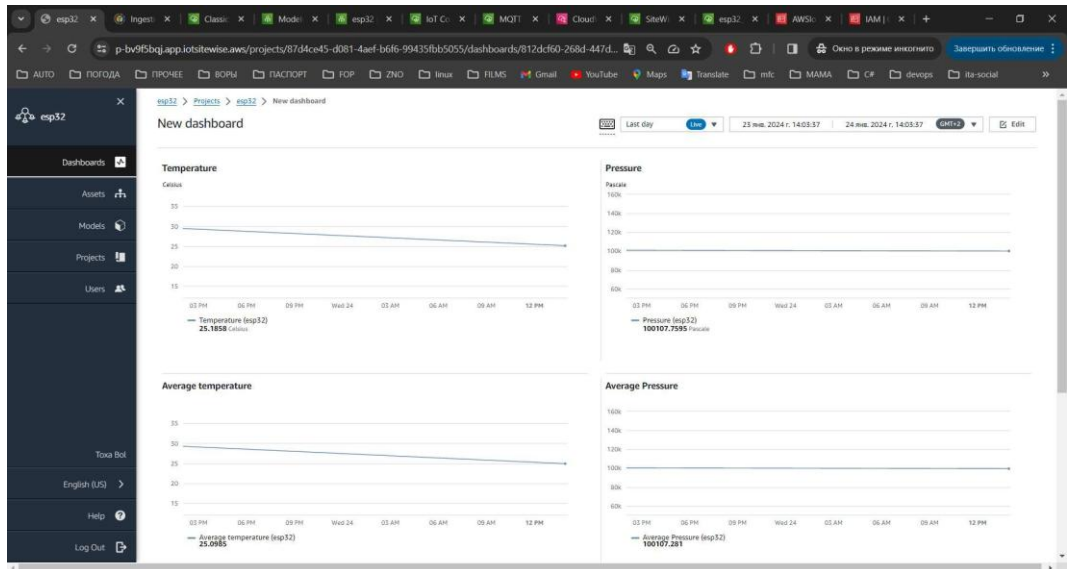


Рисунок 4.5 – Графічне представлення параметрів у середовищі SiteWise Dashboard

4.3 Інтерпретація результатів дослідження

Аналіз функціонування розробленої SCADA-системи, орієнтованої на гнучке виробництво, показав, що апаратна частина експериментального стенду ефективно взаємодіє з хмарними модулями AWS. Як видно з рисунка 4.6, передані з сенсорів дані було успішно опрацьовано та відображено в інтерфейсі користувача з використанням інструментів AWS IoT Core і AWS IoT SiteWise. Це свідчить про надійність розгорнутої архітектури та узгодженість між фізичним і цифровим рівнями системи.

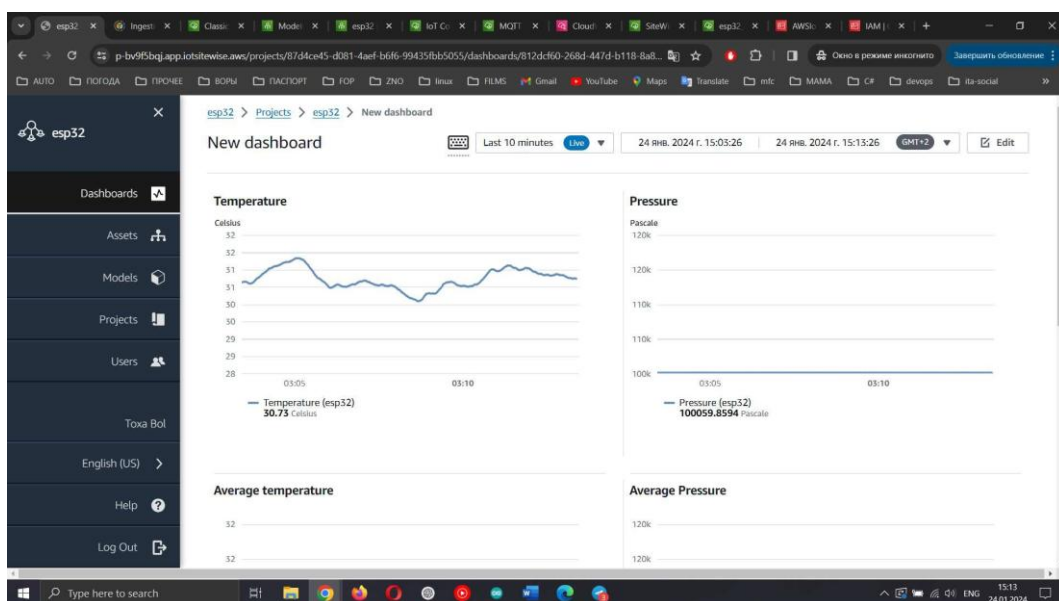


Рисунок 4.6 – Зовнішній вигляд прототипу після завершення конфігурації

Отримані телеметричні дані можуть бути проаналізовані як у контексті абсолютних часових діапазонів, так і на основі відносних відрізків часу. Наприклад, на рисунку 4.7 показано виведення інформації, зібраної за останні 10 хвилин функціонування системи, що дає змогу оперативно реагувати на зміни в умовах навколишнього середовища.

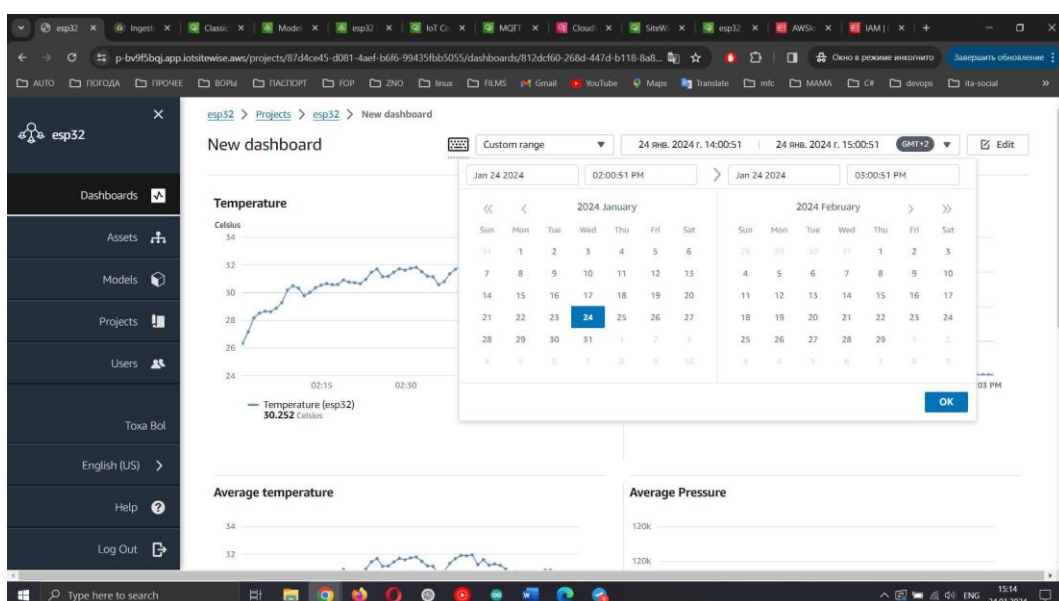


Рисунок 4.7 – Візуалізація даних за останній 10-хвилинний інтервал

Окрім цього, доступне формування аналітичних зрізів на основі фіксованих періодів. Зокрема, рисунок 4.8 демонструє відображення показників, отриманих у межах визначеного відрізка часу з 14:00 до 13:00 24 січня 2024 року. Це дозволяє здійснювати повноцінний ретроспективний аналіз на основі історичних даних.

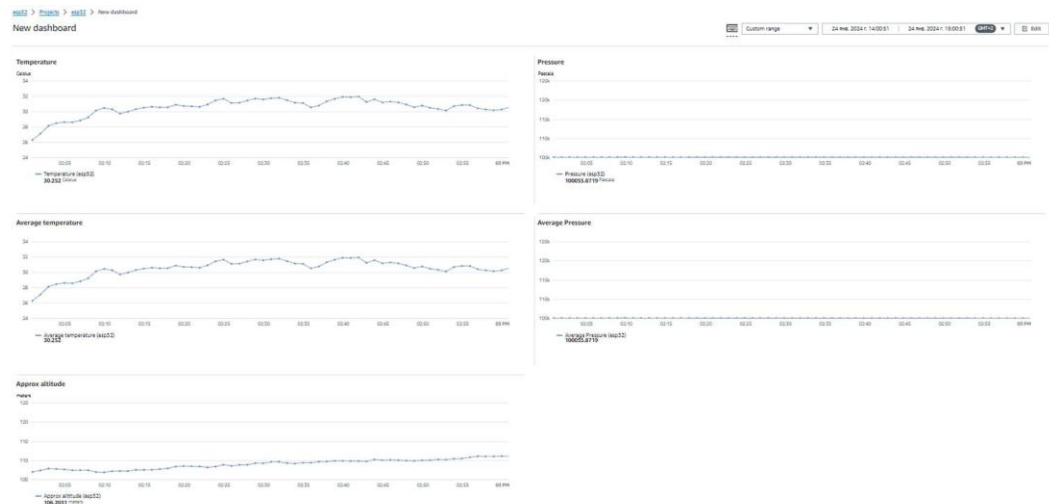


Рисунок 4.8 – Графічне представлення значень за заданим періодом часу

Усі показники, що були заздалегідь включені до конфігурації моделі асетів, коректно синхронізуються з панеллю моніторингу, де відбувається їхнє періодичне оновлення та відображення. Це забезпечує як стабільність передачі даних, так і наочність для кінцевого користувача у процесі прийняття рішень.

Під час реалізації експериментальної частини дослідження особливу увагу було приділено перевірці стабільності функціонування сенсорів, відповідальних за контроль умов безпеки у виробничій зоні. Результати підтвердили, що всі використовувані сенсори працюють у штатному режимі без збоїв або критичних відхилень. Як ілюструє рисунок 4.9, кожен із сенсорів у реальному часі виводить інформацію про свій поточний стан за

допомогою інтерфейсу серійного монітора, що дозволяє оперативно виявляти можливі аномалії.

```
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 – Поточні показники сенсорів у вікні серійного монітора

Уся інформація, яка стосується потенційно небезпечних змін у параметрах середовища (наприклад, виявлення горючих газів, підвищеного шумового фону або присутності сторонніх осіб), автоматично транслюється в хмарну інфраструктуру AWS, зокрема в систему AWS IoT SiteWise. Це дозволяє здійснювати централізований моніторинг техногенних ризиків у виробничій зоні. Приклад відображення таких критичних даних навідено на рисунку 4.10.

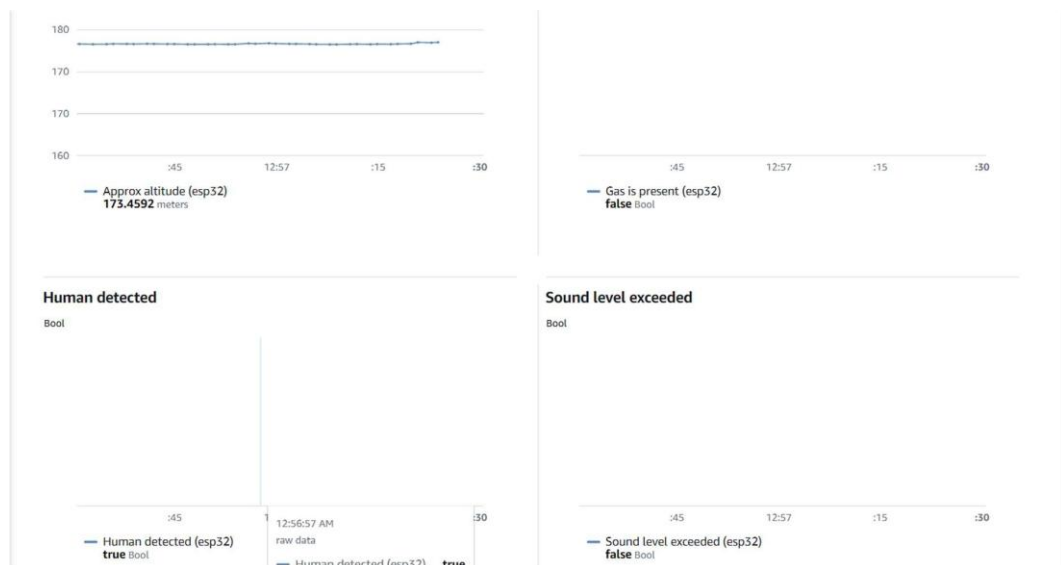


Рисунок 4.10 – Виведення повідомлень про ризики у середовищі SiteWise Dashboard

Фіксація подій безпеки в хмарному середовищі забезпечує основу для впровадження систем раннього попередження, а також відкриває можливості для подальшої автоматизації алгоритмів реагування на надзвичайні ситуації в умовах гнучкого виробництва.

ВИСНОВКИ

У межах бакалаврської кваліфікаційної роботи було здійснено комплексне дослідження принципів проектування, реалізації та функціонування SCADA-системи, орієнтованої на гнучке виробництво з використанням хмарної платформи Amazon Web Services. В аналітичному блоці розглянуто фундаментальні підходи до архітектури SCADA-рішень, проведено порівняльний аналіз технологій розробки та класифіковано можливості інтеграції хмарних сервісів у контексті індустріального застосування. На підставі огляду сформульовано мотивацію дослідження, окреслено мету, визначено об'єкт і предмет розробки, а також сформульовано перелік завдань, необхідних для досягнення цілей роботи.

У другому розділі побудовано логічну архітектуру SCADA-системи, адаптовану до умов динамічного виробництва. Здійснено розробку програмного середовища системи з акцентом на використання сервісів AWS IoT Core, SiteWise та супутніх інструментів візуалізації. Запропоновано загальний алгоритм функціонування SCADA-рішення, у якому враховано як сенсорну частину, так і логіку обробки даних. Визначено набір апаратних модулів, здатних забезпечити моніторинг фізичних параметрів у виробничому середовищі, та обґрунтовано їх вибір з погляду сумісності та ефективності.

У межах третього розділу було реалізовано комунікаційний механізм на основі протоколу MQTT, адаптованого до вимог хмарного середовища з додатковим впровадженням захисного шару SSL/TLS. Розроблено програмну логіку для інтеграції апаратного стенду з хмарною інфраструктурою. Побудовано електричну схему підключення обраних сенсорів і реалізовано фізичний макет модуля збору технологічних параметрів.

Проведений експеримент дозволив здійснити повну перевірку працездатності розробленого рішення. За допомогою серійного монітора підтверджено надходження даних з кожного сенсора, після чого інформація

була успішно передана до сервісу AWS IoT Core, трансформована згідно з визначеним правилом та передана до AWS IoT SiteWise. Там дані були оброблені, агреговані відповідно до створеної моделі асетів та візуалізовані у вигляді динамічних графічних компонентів.

Загальні результати засвідчують стабільну роботу системи: апаратна частина демонструє надійність у зборі телеметричних параметрів, хмарна – здатність до обробки та збереження інформації в реальному часі. Отже, запропоноване рішення відповідає вимогам до сучасних SCADA-систем для гнучкого виробництва, а його архітектурна модель може слугувати основою для масштабованого впровадження в умовах індустрії 4.0.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bauer, M., et al. (2017). Industrial IoT: Challenges, Design Principles, Applications, and Security. Springer.
2. Boyes, H., et al. (2018). The Industrial Internet of Things (IIoT): An Analysis Framework. Computers in Industry, 101, 1–12. URL: <https://www.sciencedirect.com/science/article/pii/S0166361517307285?via%3Dihub>
3. Chen, X., et al. (2019). Cloud-Based SCADA Systems: A Review of Recent Developments. Journal of Industrial Information Integration, 15, 1–10.
4. Gubbi, J., et al. (2017). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future Generation Computer Systems, 29(7), 1645–1660. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0167739X13000241>
5. Huang, Y., et al. (2020). Design and Implementation of a Cloud-Based SCADA System for Smart Manufacturing. IEEE Access, 8, 123456–123465.
6. Kovalenko, I. (2019). SCADA Systems in Modern Industry: Trends and Challenges. Automation and Remote Control, 80(5), 789–798.
7. Lee, J., et al. (2018). Industrial AI: Applications with Sustainable Performance. Springer.
8. Li, S., et al. (2017). Edge Computing for Industrial Applications: A Review. IEEE Industrial Electronics Magazine, 11(4), 6–16. URL: https://www.researchgate.net/publication/369628636_Edge_Computing_Applications_in_Industrial_IoT_A_Literature_Review
9. Liu, Y., et al. (2020). Digital Twin-Based Smart Manufacturing: A Review. Journal of Manufacturing Systems, 58, 144–156. URL: <https://www.sciencedirect.com/science/article/pii/S2667344423000099?via%3Dihub>
10. Lu, Y., et al. (2019). Cloud Manufacturing: A New Service-Oriented Networked Manufacturing Model. Journal of Manufacturing Systems, 54, 1–12.

11. Mourtzis, D., et al. (2018). The Role of Additive Manufacturing and Cloud Computing in Industrial Product-Service Systems. *Procedia CIRP*, 73, 3–8.
12. Nguyen, T., et al. (2021). A Survey on SCADA Systems: Architectures, Security Threats, and Solutions. *IEEE Communications Surveys & Tutorials*, 23(2), 1–25. URL: https://www.researchgate.net/publication/340453361_A_Survey_on_SCADA_Systems_Secure_Protocols_Incidents_Threats_and_Tactics/fulltext/5e96773c299bf130799acac2/A-Survey-on-SCADA-Systems-Secure-Protocols-Incidents-Threats-and-Tactics.pdf
13. Qiu, T., et al. (2019). A Survey on SCADA Systems: Architectures, Security Threats, and Solutions. *Future Generation Computer Systems*, 98, 667–681.
14. Rojas, E., et al. (2017). Cloud-Based SCADA Systems: A Review of Recent Developments. *Journal of Industrial Information Integration*, 6, 1–10.
15. Sadeghi, A., et al. (2018). Security and Privacy Challenges in Industrial Internet of Things. *Proceedings of the 52nd Annual Design Automation Conference*, 1–6. URL: <https://doi.org/10.1145/2744769.2747942>
16. Shrouf, F., et al. (2017). Smart Factories in Industry 4.0: A Review of the Concept and Its Applications. *Computers in Industry*, 81, 11–25. URL: https://www.researchgate.net/publication/309725967_Smart_factories_in_industry_40_a_review_of_the_concept_and_of_energy_management_approached_in_production_based_on_the_Internet_of_things_paradigm
17. Tan, L., et al. (2019). A Review of SCADA Systems: Architectures, Security Threats, and Solutions. *Future Generation Computer Systems*, 98, 667–681.
18. Wang, S., et al. (2020). Cloud-Based SCADA Systems: A Review of Recent Developments. *Journal of Industrial Information Integration*, 15, 1–10.
19. Xu, X., et al. (2018). Cloud Manufacturing: A New Service-Oriented Networked Manufacturing Model. *Journal of Manufacturing Systems*, 54, 1–12.
20. Yang, C., et al. (2017). Cloud-Based SCADA Systems: A Review of Recent Developments. *Journal of Industrial Information Integration*, 6, 1–10.

21. Zhang, Y., et al. (2019). A Survey on SCADA Systems: Architectures, Security Threats, and Solutions. *Future Generation Computer Systems*, 98, 667–681.
22. Zhou, K., et al. (2018). Cloud-Based SCADA Systems: A Review of Recent Developments. *Journal of Industrial Information Integration*, 6, 1–10.
23. Іваненко, О. М. (2019). Хмарні технології в системах автоматизації виробництва. *Вісник Національного технічного університету України «КПІ»*, 4(12), 45–52.
24. Коваленко, І. В. (2020). Використання протоколу MQTT в системах промислової автоматизації. *Автоматизація виробничих процесів*, 3(27), 33–39.
25. Мельник, С. П. (2018). SCADA-системи: сучасні тенденції та перспективи розвитку. *Системи управління, навігації та зв'язку*, 2(50), 12–18.
26. Петренко, А. В. (2021). Інтеграція хмарних сервісів в SCADA-системи для гнучкого виробництва. *Вісник Харківського національного університету радіоелектроніки*, 1(65), 25–30.
27. Сидоренко, Л. М. (2017). Промислова автоматизація та SCADA-системи: огляд сучасних рішень. *Технічні науки*, 4(22), 55–60.
28. Ткаченко, В. О. (2019). Використання хмарних технологій в системах управління виробництвом. *Інформаційні технології та комп'ютерна інженерія*, 3(45), 40–46.
29. Федоренко, І. С. (2020). Моделювання гнучких виробничих систем з використанням SCADA-технологій. *Вісник Дніпровського національного університету*, 2(58), 60–66.
30. Шевченко, Н. П. (2018). Хмарні обчислення в системах автоматизації: переваги та недоліки. *Сучасні інформаційні технології*, 1(31), 70–75.

Додаток А
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Автоматизована система моделювання гнучкого виробництва

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра КСУ
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 0.1 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ✱ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

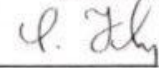
Експертна комісія:

Завідувач кафедри КСУ
(прізвище, ініціали, посада)


(підпис)

В'ячеслав КОВТУН

Гарант ОПП
(прізвище, ініціали, посада)


(підпис)

Олег КОВАЛЮК

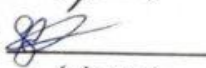
Особа, відповідальна за перевірку 
(підпис)

Володимир ДУБОВОЙ
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
(підпис)

В'ячеслав КОВТУН
(прізвище, ініціали, посада)

Здобувач 
(підпис)

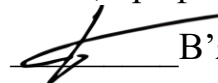
Ілля БОРІЦ
(прізвище, ініціали)

Додаток Б
(обов'язковий)
ВНТУ

ЗАТВЕРДЖЕНО

Зав. кафедри КСУ ВНТУ,

д.т.н., проф.

 В'ячеслав КОВТУН

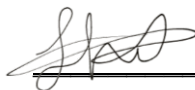
“ 23 ” травня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

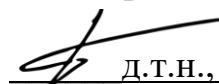
на виконання бакалаврської кваліфікаційної роботи

«АВТОМАТИЗОВАНА СИСТЕМА МОДЕЛЮВАННЯ ГНУЧКОГО
ВИРОБНИЦТВА»

Студент групи

 2АКІТ-216 Ілля БОРЩ

Керівник

 д.т.н., проф. В'ячеслав КОВТУН

Вінниця 2025

1. Назва та галузь застосування

1.1. Назва – Автоматизована система моделювання гнучкого виробництва.

1.2. Галузь застосування – Промислова автоматизація, IoT, Хмарні технології, Системи SCADA.

2. Підстава для проведення розробки.

Тема бакалаврської кваліфікаційної роботи затверджена наказом по ВНТУ № 97 від 20.03.2025 р.

3. Мета та призначення розробки.

Метою дипломної роботи є розробка прототипу хмарної SCADA-системи для автоматизованого моніторингу та управління технологічними процесами та попереджувати небезпечні ситуації оптимізуючи виробництво шляхом постійного контролю параметрів, інтеграції з AWS та створення масштабованої архітектури для гнучкого виробництва.

4. Джерела розробки.

Бакалаврська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Сидоренко, Л. М. (2017). Промислова автоматизація та SCADA-системи: огляд сучасних рішень. Технічні науки, 4(22), 55–60.
2. Федоренко, І. С. (2020). Моделювання гнучких виробничих систем з використанням SCADA-технологій. Вісник Дніпровського національного університету, 2(58), 60–66.
3. Іваненко, О. М. (2019). Хмарні технології в системах автоматизації виробництва. Вісник Національного технічного університету України «КПІ», 4(12), 45–52.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- збір технологічних параметрів з виробничого середовища за допомогою сенсорів
- передача зібраних даних до хмарної інфраструктури
- візуалізація технологічних параметрів у реальному часі

- автоматизований контроль технологічними процесами

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- Python, C++;
- Linux, Window, MacOS

5.2.2. Умови експлуатації системи:

- Стабільний доступ до мережі інтернет;
- Мінімізація впливу зовнішніх чинників;
- Постійне живлення мікроконтролерів та сенсорів.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми дипломної роботи. Постановка задач дослідження «15» травня 2025 р.
2. Визначення технічних характеристик системи «19» травня 2025 р.
3. Розробка програмного забезпечення системи «23» травня 2025 р.

6.2 Графічні матеріали:

1. Розробка моделі системи «24» травня 2025 р.
2. Тестування програмного забезпечення «10» червня 2025 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «4» червня 2025 р.
- 7.2. Атестація роботи здійснюється на попередньому захисті. Попередній захист бакалаврської кваліфікаційної роботи провести до «10» червня 2025 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист бакалаврської кваліфікаційної роботи провести до «18» червня 2025 р.

Додаток В

(довідковий)

Лістинг програмного коду

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Adafruit_BMP280.h>
#include <Wire.h>

// Піни сенсорів
#define MQ2_PIN 34
#define SOUND_PIN 35
#define HLK_RX 16
#define HLK_TX 17

Adafruit_BMP280 bmp;
HardwareSerial HLKSerial(1);
WiFiClientSecure secureClient;
PubSubClient client(secureClient);

// Wi-Fi доступ
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";

// AWS MQTT параметри
const char* mqtt_server = "your-aws-iot-endpoint.amazonaws.com";
const int mqtt_port = 8883;
const char* mqtt_topic = "$aws/things/esp32/shadow/update";

// Сертифікати (змінити на свої)
const char* ca_cert = \
"-----BEGIN CERTIFICATE-----\n" \
"...CA CERTIFICATE...\n" \
"-----END CERTIFICATE-----\n";

const char* client_cert = \
"-----BEGIN CERTIFICATE-----\n" \
"...DEVICE CERTIFICATE...\n" \
```

```
"-----END CERTIFICATE-----\n";
```

```
const char* client_key = \
"-----BEGIN RSA PRIVATE KEY-----\n" \
"...PRIVATE KEY...\n" \
"-----END RSA PRIVATE KEY-----\n";
```

```
void setup_wifi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  int attempts = 0;
  while (WiFi.status() != WL_CONNECTED && attempts < 20) {
    delay(500);
    Serial.print(".");
    attempts++;
  }
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi connection failed. Restarting...");
    ESP.restart();
  }
}
```

```
void reconnect() {
  while (!client.connected()) {
    Serial.print("Connecting to AWS...");
    if (client.connect("ESP32ClientSecure")) {
      Serial.println("Connected.");
    } else {
      Serial.print("Failed, rc=");
      Serial.println(client.state());
      delay(3000);
    }
  }
}
```

```
void setup() {
  Serial.begin(115200);
  HLKSerial.begin(256000, SERIAL_8N1, HLK_RX, HLK_TX);
  setup_wifi();
```

```

secureClient.setCACert(ca_cert);
secureClient.setCertificate(client_cert);
secureClient.setPrivateKey(client_key);
client.setServer(mqtt_server, mqtt_port);

```

```

if (!bmp.begin(0x76)) {
    Serial.println("BMP280 not detected. System halted.");
    while (1);
}
}

```

```

String sanitize(String input) {
    input.replace("\\", "");
    input.replace("\\", "/");
    if (input.length() > 128) input = input.substring(0, 128); // ліміт довжини
    return input;
}

```

```

void loop() {
    if (!client.connected()) reconnect();
    client.loop();
}

```

```

float temp = bmp.readTemperature();
float press = bmp.readPressure();
float alt = bmp.readAltitude(1013.25);
int gas = analogRead(MQ2_PIN);
int sound = analogRead(SOUND_PIN);

```

```

String presence = "";
while (HLKSerial.available()) {
    presence += (char)HLKSerial.read();
    delay(2);
}
presence = sanitize(presence);

```

```

String payload = "{";
payload += "\"state\": { \"reported\": {";
payload += "\"temperature\": " + String(temp, 2) + ",";

```

```

payload += "\"pressure\": " + String(press / 100.0, 2) + ",";
payload += "\"altitude\": " + String(alt, 2) + ",";
payload += "\"gas_level\": " + String(gas) + ",";
payload += "\"sound_level\": " + String(sound) + ",";
payload += "\"presence_raw\": \"\" + presence + "\"";
payload += "} }";

if (payload.length() < 512) {
  client.publish(mqtt_topic, payload.c_str(), true); // QoS 1, retained
  Serial.println("Published:");
  Serial.println(payload);
} else {
  Serial.println("Payload too large. Not sent.");
}

delay(10000);
}

```

Додаток Г
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

АВТОМАТИЗОВАНА СИСТЕМА МОДЕЛЮВАННЯ ГНУЧКОГО ВИРОБНИЦТВА

Студент групи 2АКІТ-216

Ілля БОРЩ

Керівник д.т.н., проф.

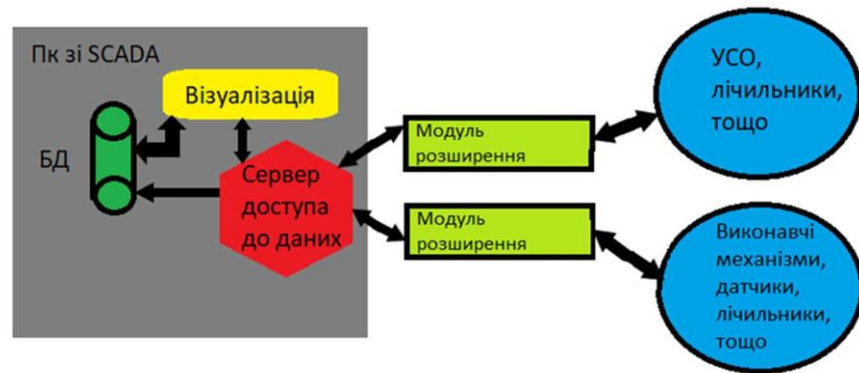
В'ячеслав КОВТУН

Вінниця 2025

Перелік ілюстративних матеріалів:

1. Класична схема побудови SCADA системи.
2. Типова схема взаємодії компонентів SCADA системи.
3. Схема можливої реалізації взаємодії з IoT-пристроями за допомогою Amazon Web Services.
4. Інтегрована схема функціонування SCADA-системи адаптивного виробництва з використанням AWS IoT.
5. Алгоритмічна схема функціонування SCADA-системи орієнтованої на гнучке виробництво з використанням AWS IoT Core і SiteWise.
6. Схема електричних з'єднань між мікроконтролером ESP32 та зовнішніми компонентами.
7. Готовий макет апаратного модуля для збору технологічних параметрів у SCADA-системі.
8. Виведення показників сенсорів у середовищі серійного монітора.
9. Приклад вхідних повідомлень у тестовому MQTT-клієнті середовища AWS IoT Core.
10. Атрибутивна структура ассет-моделі у системі AWS SiteWise.
11. Графічне представлення параметрів у середовищі SiteWise Dashboard.
12. Візуалізація даних за останній 10-хвилинний інтервал.
13. Графічне представлення значень за заданим періодом часу.
14. Поточні показники сенсорів у вікні серійного монітора.
15. Виведення повідомлень про ризики у середовищі SiteWise Dashboard.

Класична схема побудови SCADA системи.



Типова схема взаємодії компонентів SCADA системи.

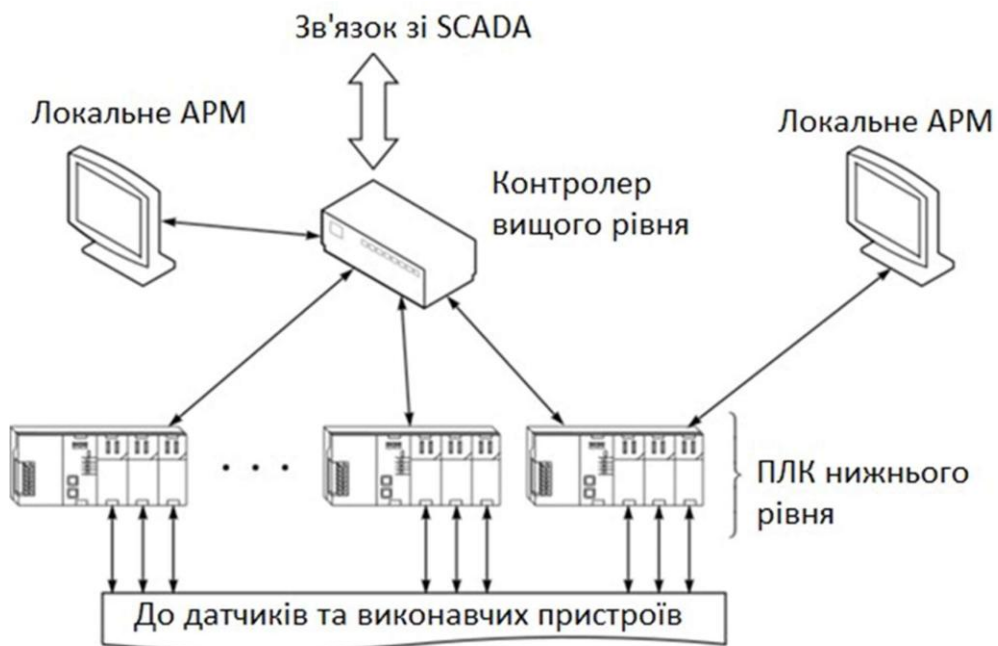
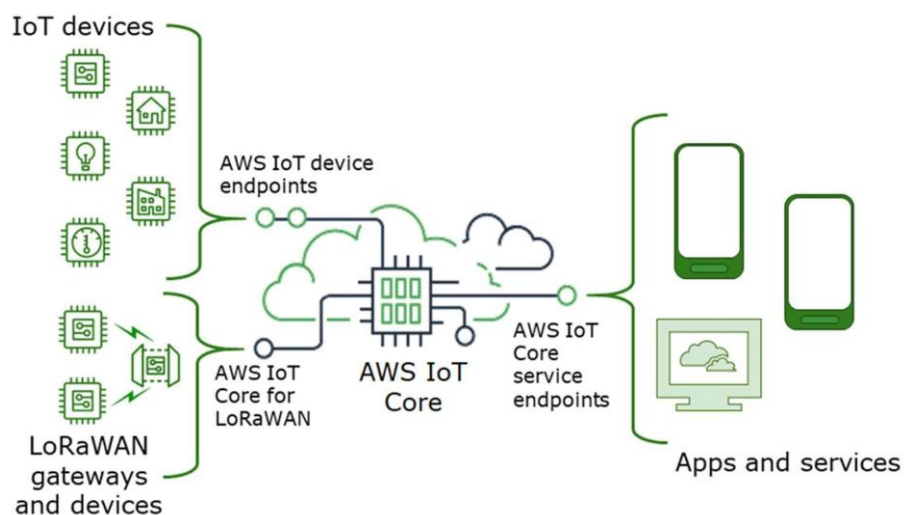
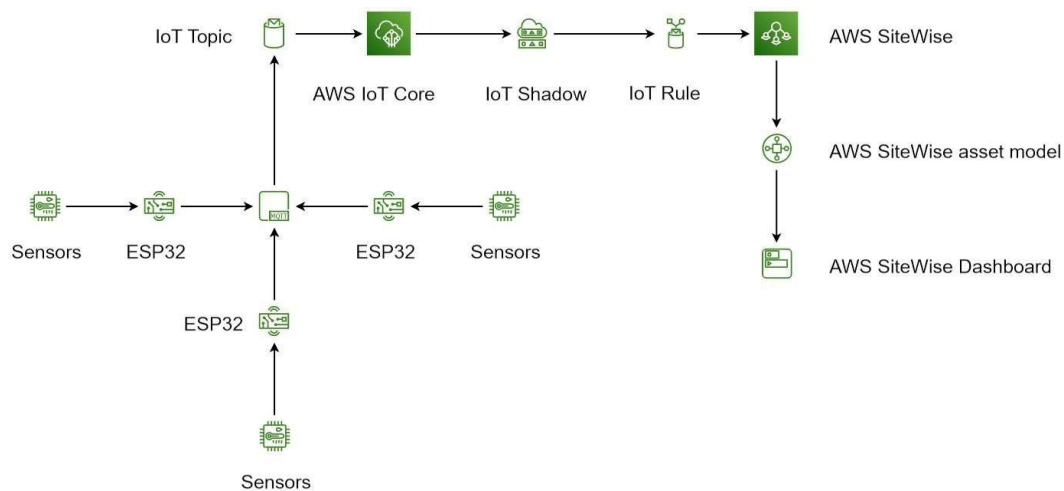


Схема можливої реалізації взаємодії з IoT-пристроями за допомогою Amazon Web Services.



Інтегрована схема функціонування SCADA-системи адаптивного виробництва з використанням AWS IoT.



Алгоритмічна схема функціонування SCADA-системи орієнтованої на гнучке виробництво з використанням AWS IoT Core і SiteWise.

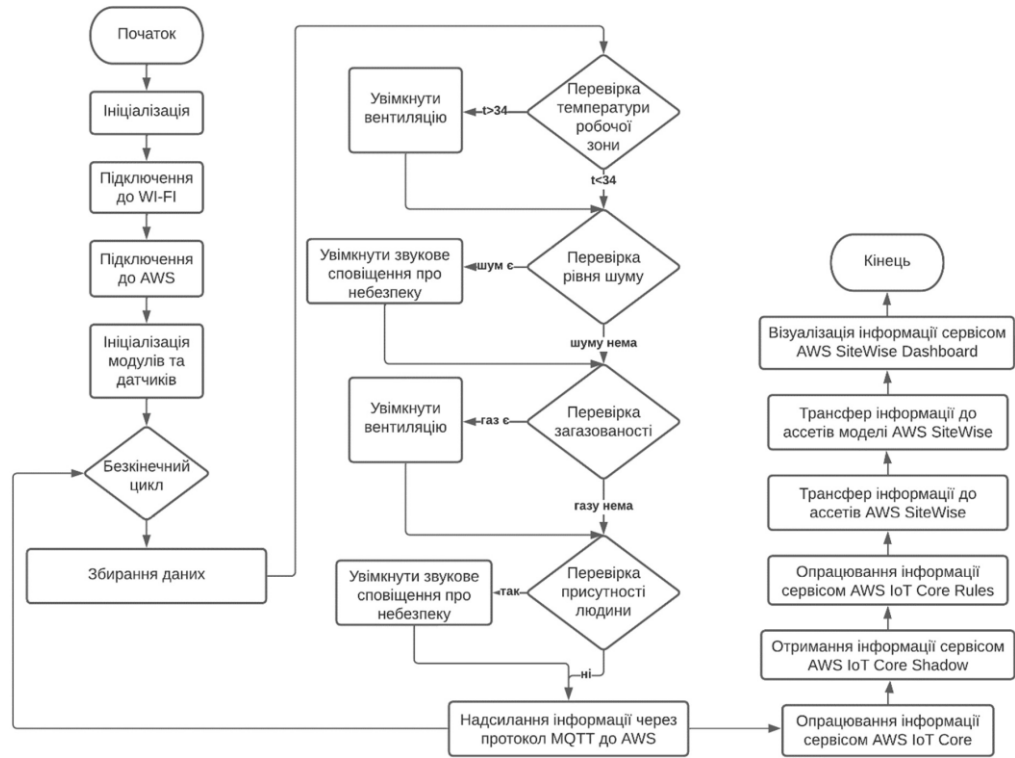
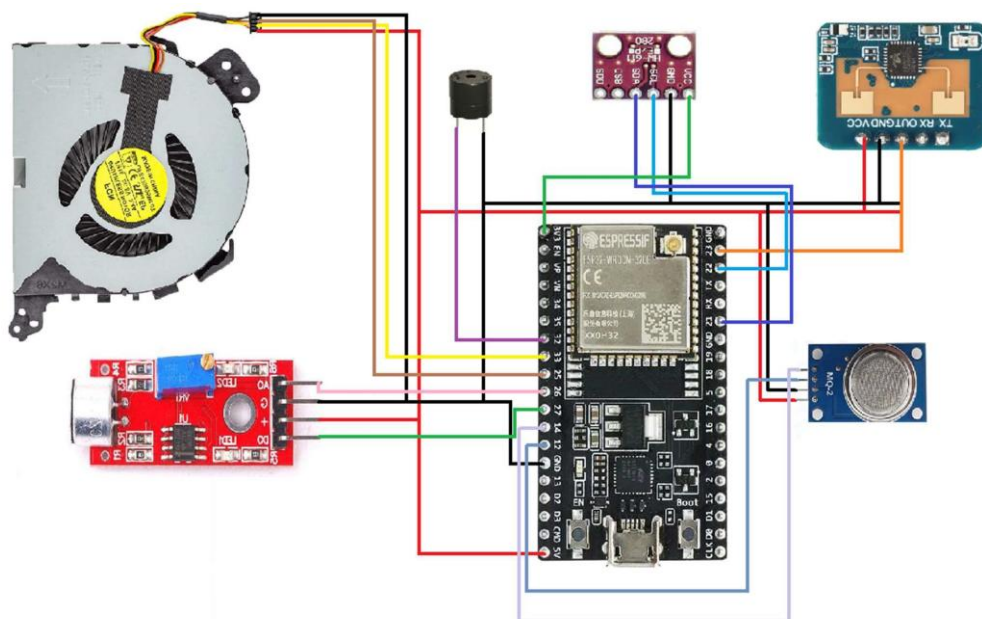
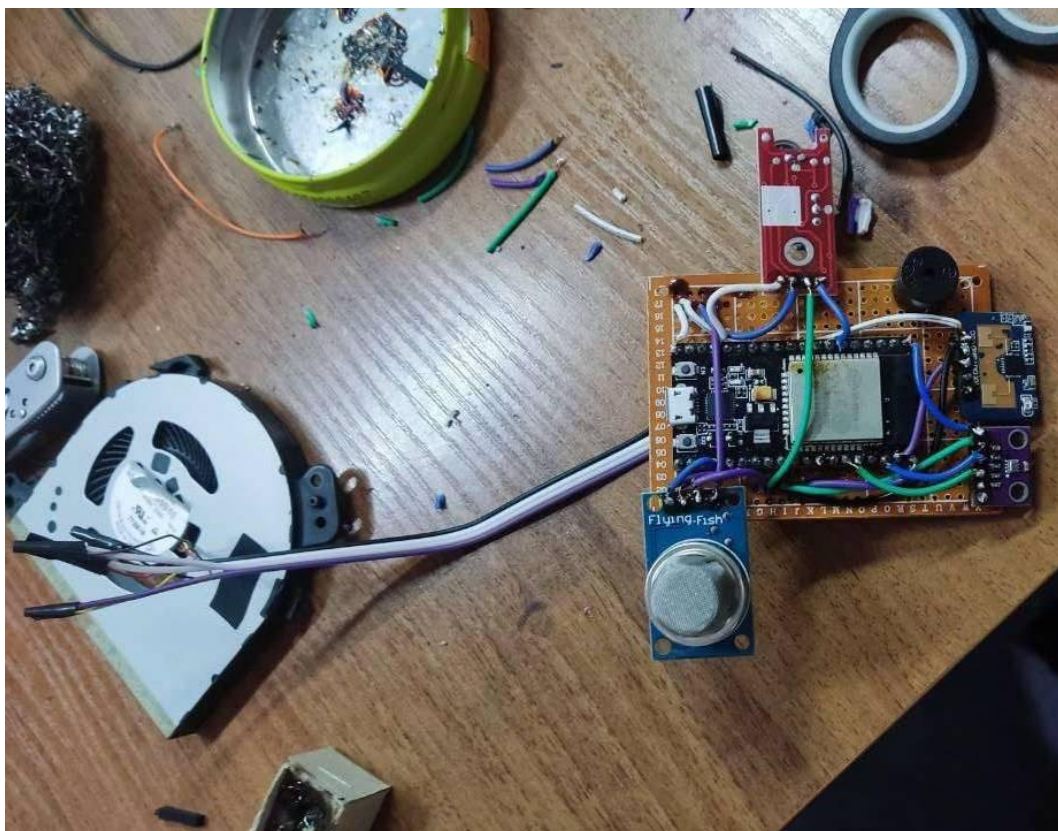


Схема електричних з'єднань між мікроконтролером ESP32 та зовнішніми компонентами.



Готовий макет апаратного модуля для збору технологічних параметрів у SCADA-системі.



Виведення показників сенсорів у середовищі серійного монітора.

```

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

Приклад вхідних повідомлень у тестовому MQTT-клієнті середовища AWS IoT Core.

```

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

```

► Properties

Атрибутивна структура ассет-моделі у системі AWS SiteWise.

Attributes (0)

Measurements (6)

Transforms (0)

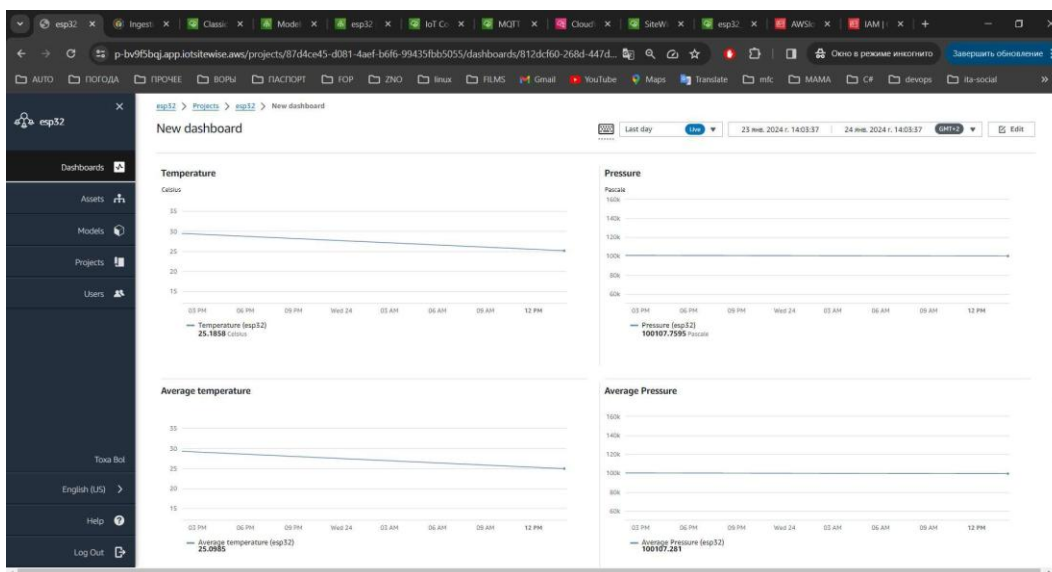
Metrics (2)

Components (0)

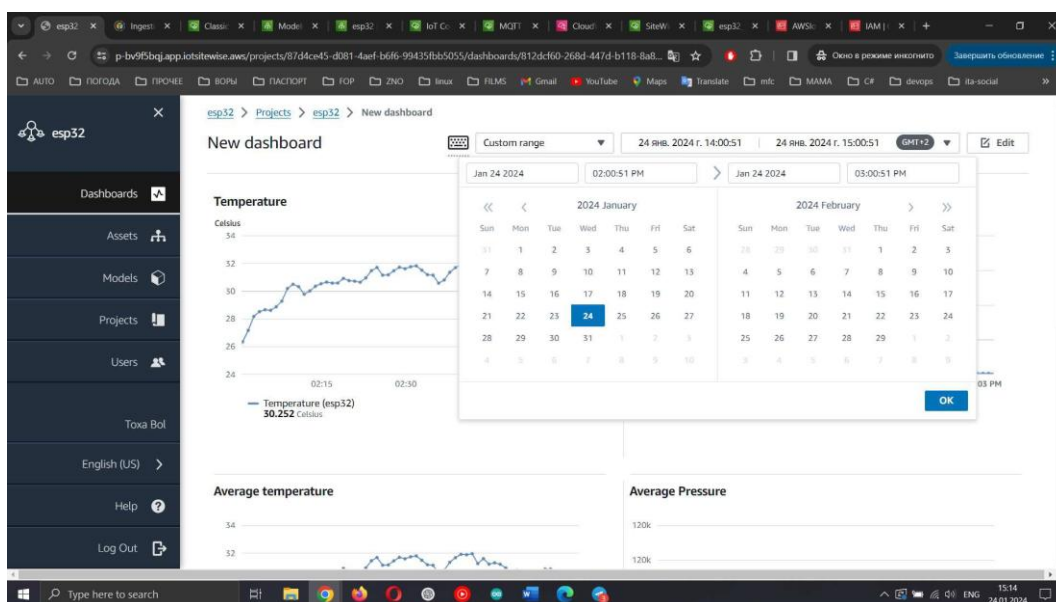
Measurements (6)

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-				

Графічне представлення параметрів у середовищі SiteWise Dashboard.



Візуалізація даних за останній 10-хвилинний інтервал.



Графічне представлення значень за заданим періодом часу.



Поточні показники сенсорів у вікні серійного монітора.

```

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 ->

```

Виведення повідомлень про ризики у середовищі SiteWise Dashboard.

