НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Навчально-науковий інститут автоматики та електротехніки
(повне найменування інституту, факультету)

Кафедра комп'ютеризованих систем управління
(повна назва кафедри)

«Допущений до захисту»
Завідувач кафедри
_____Черно О. О.
«10» червня 2025 р.

Пояснювальна записка

до кваліфікаційної роботи

на здобуття першого (бакалаврського) рівня вищої освіти
(освітньо-кваліфікаційний рівень)

на тему: Автоматизоване управління промисловим обладнанням з використанням Linux, IoT та групових політик

Виконав студент **4** курсу, **4341** групи
спеціальності «**151 - Автоматизація та комп'ютерно-інтегровані технології**»

Погорілов І. О.
(прізвище та ініціали)

Керівник **Топалов А. М.**
(прізвище та ініціали)

Рецензент **Клочков О. П.**
(прізвище та ініціали)

м. Миколаїв – 2025 року

Національний університет кораблебудування імені адмірала Макарова

ННІ автоматики і електротехніки
Кафедра комп'ютеризованих систем управління
Рівень вищої освіти **бакалавр**

Спеціальність **151 Автоматизація та комп'ютерно-інтегровані технології**
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
_____ **Черно О.О.**
«03» лютого 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Погорілову Іллі Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи: **Автоматизоване управління промисловим обладнанням з використанням Linux, IoT та групових політик**

керівник роботи *Топалов Андрій Миколайович, к.т.н., доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "23" 04 2025 р. № 343

2. Строк подання студентом роботи *09.06.2025*

3. Вихідні дані до роботи *Розробити автоматизовану систему управління промисловим обладнанням, на базі операційної системи Linux, технологій інтернету речей (IoT) та механізмів групових політик*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Теоретичні основи автоматизованого управління промисловим обладнанням з використанням Linux та IOT
Формування компонентів системи та їх опис
Програмне забезпечення системи та результати тестування
Охорона праці

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

Актуальність, мета та задачі роботи

Аналіз сучасних підходів до автоматизації

Програмне забезпечення

Технологічний стек системи управління

Архітектура системи

Дистанційний моніторинг та контроль роботи обладнання

Тестування розробленої системи

Висновок

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Охорона праці	*Топалов А.М., к.т.н., доцент каф. КСУ*	23.04.2025	05.05.2025

7. Дата видачі завдання: «_11_» ___лютого___ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів підготовки кваліфікаційної роботи	Строк виконання	Примітка
1	*Теоретичні основи автоматизованого управління промисловим обладнанням з використанням linux та iot*	*20.03.2025*	*виконано*
2	*Формування компонентів системи та їх опис*	*11.04.2025*	*виконано*
3	*Програмне забезпечення системи та результати тестування*	*21.04.2025*	*виконано*
4	*Охорона праці*	*05.05.2025*	*виконано*

Студент _____ Погорілов І. О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Топалов А. М.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Обсяг кваліфікаційної роботи становить 82 сторінок, у роботі 19 рисунків, 6 таблиць та 18 використаних джерел.

Метою даної кваліфікаційної роботи є розробка автоматизованої системи управління промисловим обладнанням, на базі операційної системи Linux, технологій інтернету речей (IoT) та механізмів групових політик.

У роботі проведено аналіз сучасних підходів до автоматизації промислових процесів, обґрунтовано вибір технологій Linux та IoT, розроблено архітектуру системи, описано її компоненти та їх взаємодію. Реалізовано програмне забезпечення з використанням Node-RED, Mosquitto, InfluxDB, Grafana та Ansible, що забезпечує моніторинг, управління та централізоване адміністрування обладнання. Проведено тестування системи на прикладі верстатів Hardinge Bridgeport XR 760, під час якого оцінено її стабільність, ефективність та адаптивність до різних виробничих сценаріїв. Розглянуто питання охорони праці та інформаційної безпеки, запропоновано рекомендації щодо масштабування системи для різних типів промислових підприємств.

Ключові слова: автоматизація, промислове обладнання, Linux, IoT, групові політики, Node-RED, Ansible, моніторинг.

ABSTRACT

The volume of the qualification work is 82 pages, the work contains 19 figures, 6 tables and 18 sources used.

The purpose of this qualification work is to develop an automated industrial equipment management system based on the Linux operating system, Internet of Things (IoT) technologies and group policy mechanisms.

The work analyzes modern approaches to industrial process automation, justifies the choice of Linux and IoT technologies, develops the system architecture, describes its components and their interaction. Software is implemented using Node-RED, Mosquitto, InfluxDB, Grafana and Ansible, which provides monitoring, management and centralized administration of equipment. The system is tested on the example of Hardinge Bridgeport XR 760 machines, during which its stability, efficiency and adaptability to various production scenarios are assessed. The issues of occupational health and information security are considered, and recommendations are made for scaling the system for different types of industrial enterprises.

Keywords: automation, industrial equipment, Linux, IoT, group policies, Node-RED, Ansible, monitoring.

ЗМІСТ

ВСТУП

Сучасна промисловість постійно потребує підвищення продуктивності, точності та безпеки виробничих процесів. У швидко змінному технологічному середовищі одним із перспективних напрямів розвитку стає впровадження автоматизованих систем управління, що базуються на Інтернеті речей (IoT) та операційних системах Linux. Поєднання цих технологій дає змогу значно підвищити ефективність керування обладнанням завдяки централізованому збору, аналізу й обробці даних у режимі реального часу. Не менш важливу роль відіграє застосування групових політик, які дозволяють уніфікувати та централізовано адмініструвати обладнання в межах підприємства. Це особливо актуально для великих виробничих комплексів, де контроль над численними пристроями та вузлами потребує високого рівня автоматизації. Без використання відповідних інструментів управління такі процеси були б складними, а їх реалізація вимагала б значних витрат часу й ресурсів. Отже, розробка та впровадження автоматизованих систем керування на основі сучасних технологій Linux, IoT та групових політик є надзвичайно актуальним завданням. Його успішна реалізація дозволяє суттєво оптимізувати виробничі процеси, покращити контроль за обладнанням та підвищити конкурентоспроможність підприємств у сучасних умовах.

Метою дипломної роботи є розробка та реалізація автоматизованої системи управління промисловим обладнанням, яка ґрунтується на використанні операційної системи Linux, технологій інтернету речей (IoT) та механізмів групових політик для централізованого управління, моніторингу і підвищення ефективності роботи промислових підприємств.

Для досягнення поставленої мети були визначені наступні завдання:

1. Провести аналіз сучасних технологій автоматизації промислових систем, визначивши переваги використання Linux, IoT та групових політик.
2. Розробити архітектуру автоматизованої системи управління з урахуванням специфіки промислових процесів.

3. Обґрунтувати вибір та налаштувати програмні компоненти, необхідні для функціонування системи.

4. Забезпечити можливість дистанційного моніторингу та контролю роботи обладнання.

5. Провести тестування створеної системи та виконати аналіз отриманих результатів.

У ході виконання дипломної роботи використано такі методи дослідження:

1. Метод аналізу і синтезу – для вивчення сучасних технологій, літературних джерел і систематизації отриманої інформації щодо автоматизації промислових процесів.

2. Метод моделювання – для розробки архітектури автоматизованої системи управління на базі технологій Linux, IoT та групових політик.

3. Експериментальний метод – для практичної реалізації системи, її тестування та отримання

4. Метод порівняльного аналізу – для порівняння

Практичне значення результатів полягає в тому, що запропоновані рішення можна впроваджувати на реальних промислових підприємствах, зменшуючи витрати на адміністрування та підтримку обладнання, покращуючи контроль та моніторинг процесів, що дозволяє знизити кількість виробничих помилок та підвищити загальну продуктивність підприємств.

Об’єкт дослідження – автоматизоване управління та моніторинг роботи промислового обладнання з використанням сучасних інформаційних технологій.

Предмет дослідження – методи, засоби та технології автоматизованого управління промисловим обладнанням на основі Linux, IoT та групових політик.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ПРОМИСЛОВИМ ОБЛАДНАННЯМ З ВИКОРИСТАННЯМ LINUX ТА IOT

1.1 Аналіз сучасних підходів до автоматизації промислового обладнання

Автоматизація виробничих процесів відіграє ключову роль у підвищенні ефективності та конкурентоспроможності сучасних промислових підприємств. Застосування цифрових технологій у системах управління забезпечує гнучкість, масштабованість і надійність, що дозволяє оптимізувати виробничі операції та знижувати витрати. Сучасні підходи до автоматизації ґрунтуються на інтеграції передових рішень, які поєднують апаратні та програмні компоненти для створення комплексних систем управління.

Залежно від рівня інтеграції та складності, методи автоматизації промислових процесів можна умовно поділити на кілька категорій. Традиційні автоматизовані системи базуються на використанні програмованих логічних контролерів (ПЛК), систем диспетчерського контролю та збору даних (SCADA) і розподілених систем керування (DCS). Ці рішення забезпечують надійне управління обладнанням, але часто обмежені у гнучкості та масштабованості. На противагу їм системи на базі промислового Інтернету речей (IIoT) передбачають підключення обладнання до єдиної мережі, що дозволяє збирати та аналізувати дані в реальному часі, підвищуючи оперативність реагування на зміни у виробничому середовищі. Хмарні та периферійні обчислення представляють сучасний підхід, який використовується для обробки даних у хмарних сервісах або на локальних периферійних пристроях, що зменшує навантаження на центральні сервери та сприяє швидшій обробці інформації.

Сучасна автоматизація промислових процесів спирається на кілька основоположних принципів, які визначають її ефективність і адаптивність. Інтеграція IoT-рішень забезпечує використання датчиків і мережевого обладнання для збору та обміну інформацією між пристроями, що дозволяє створювати динамічні системи управління з високим рівнем автоматизації. Використання відкритих стандартів, таких як протоколи зв’язку MQTT (Рис.

1.1), Modbus і OPC UA, спрощує інтеграцію різнорідних пристроїв і сприяє сумісності компонентів системи. Застосування штучного інтелекту та машинного навчання відкриває можливості для аналізу великих обсягів даних, що надходять від обладнання, і прогнозування потенційних несправностей, що підвищує надійність і довговічність промислових систем. Ці принципи формують основу для створення автоматизованих рішень, здатних відповідати викликам сучасного виробництва. [1]

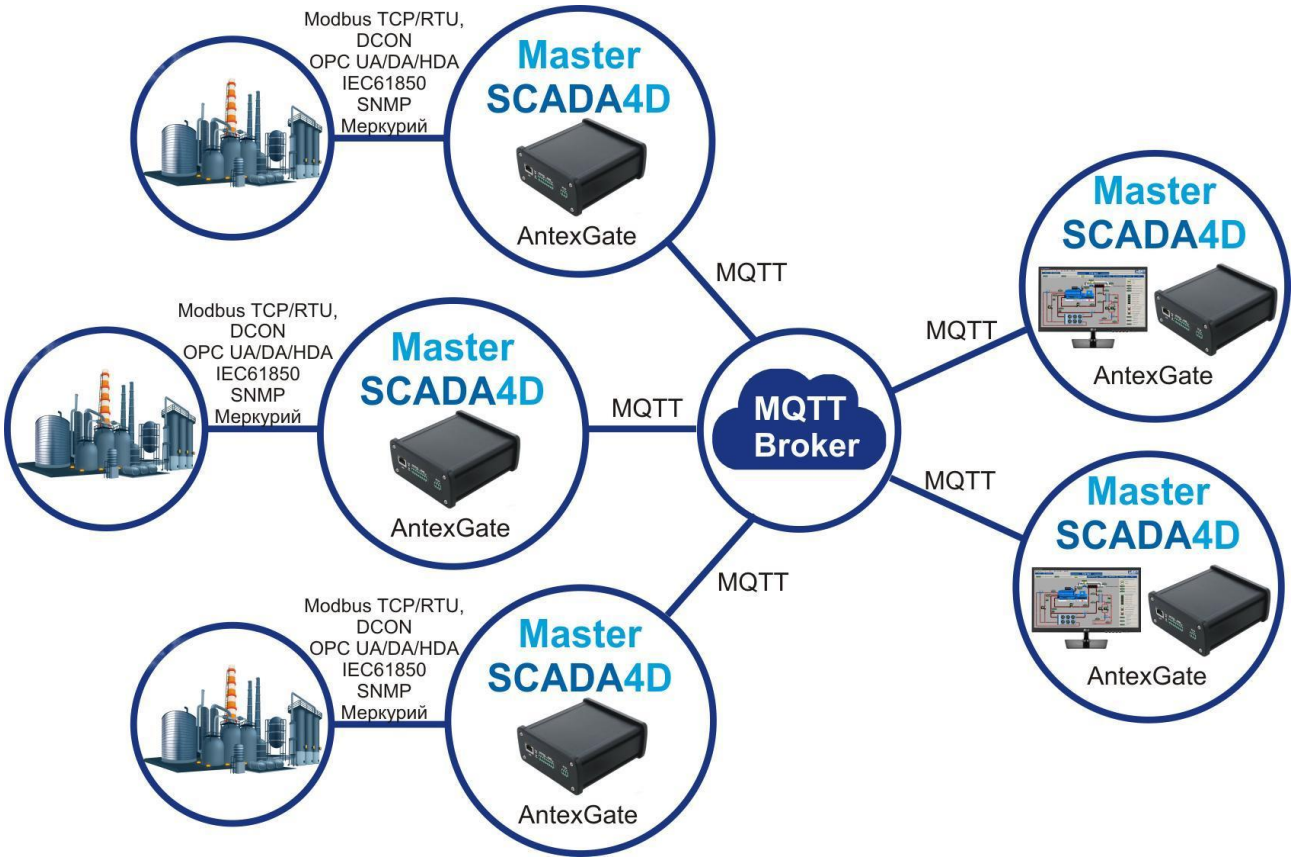

Рис. 1.1 - Структура моніторингу MQTT

Впровадження сучасних автоматизованих систем управління промисловим обладнанням приносить підприємствам низку значних переваг, які сприяють підвищенню їхньої ефективності та конкурентоспроможності. Завдяки зменшенню простоїв обладнання та раціональному використанню ресурсів такі системи дозволяють суттєво знизити експлуатаційні витрати. Швидкий аналіз даних і адаптація до змін у виробничому процесі забезпечують підвищення продуктивності, дозволяючи системі оперативно реагувати на динамічні умови роботи. Крім того, централізований моніторинг і діагностика обладнання

сприяють зниженню ризиків аварійних ситуацій і підвищенню безпеки виробничих процесів, що є критично важливим для сучасних промислових комплексів.

Автоматизація виробничих процесів відіграє ключову роль у забезпеченні ефективності підприємств, дозволяючи не лише скорочувати витрати, але й покращувати контроль за технологічними процесами, мінімізувати вплив людського фактора та підвищувати загальну продуктивність. Сучасні підходи до автоматизації ґрунтуються на концепціях цифрового виробництва (Digital Manufacturing), Індустрії 4.0 та Інтернету речей (IoT). Ці концепції забезпечують гнучкість, масштабованість і надійність систем управління, дозволяючи підприємствам адаптуватися до мінливих ринкових умов і технологічних викликів. Використання цифрових технологій у поєднанні з IoT-рішеннями створює основу для інтеграції обладнання в єдину мережу, що сприяє оперативному збору, аналізу та обробці даних у реальному часі. [2]

Традиційні автоматизовані системи, які використовуються в промисловості протягом кількох десятиліть, базуються на жорсткій логіці керування та залишаються актуальними для багатьох застосувань. Основу таких систем складають програмовані логічні контролери (ПЛК), які відповідають за автоматизацію окремих ділянок виробництва, забезпечуючи стабільне виконання локальних завдань. Системи диспетчерського керування та збору даних (SCADA) відіграють важливу роль у моніторингу та аналізі роботи виробничого обладнання, надаючи операторам детальну інформацію про стан процесів. Розподілені системи управління (DCS) застосовуються для комплексної автоматизації великих виробництв із численними технологічними процесами, забезпечуючи координацію та контроль на всіх рівнях (Рис. 1.2). Незважаючи на їхню надійність, традиційні системи часто обмежені у гнучкості порівняно з сучасними IoT- і хмарооріентованими рішеннями.

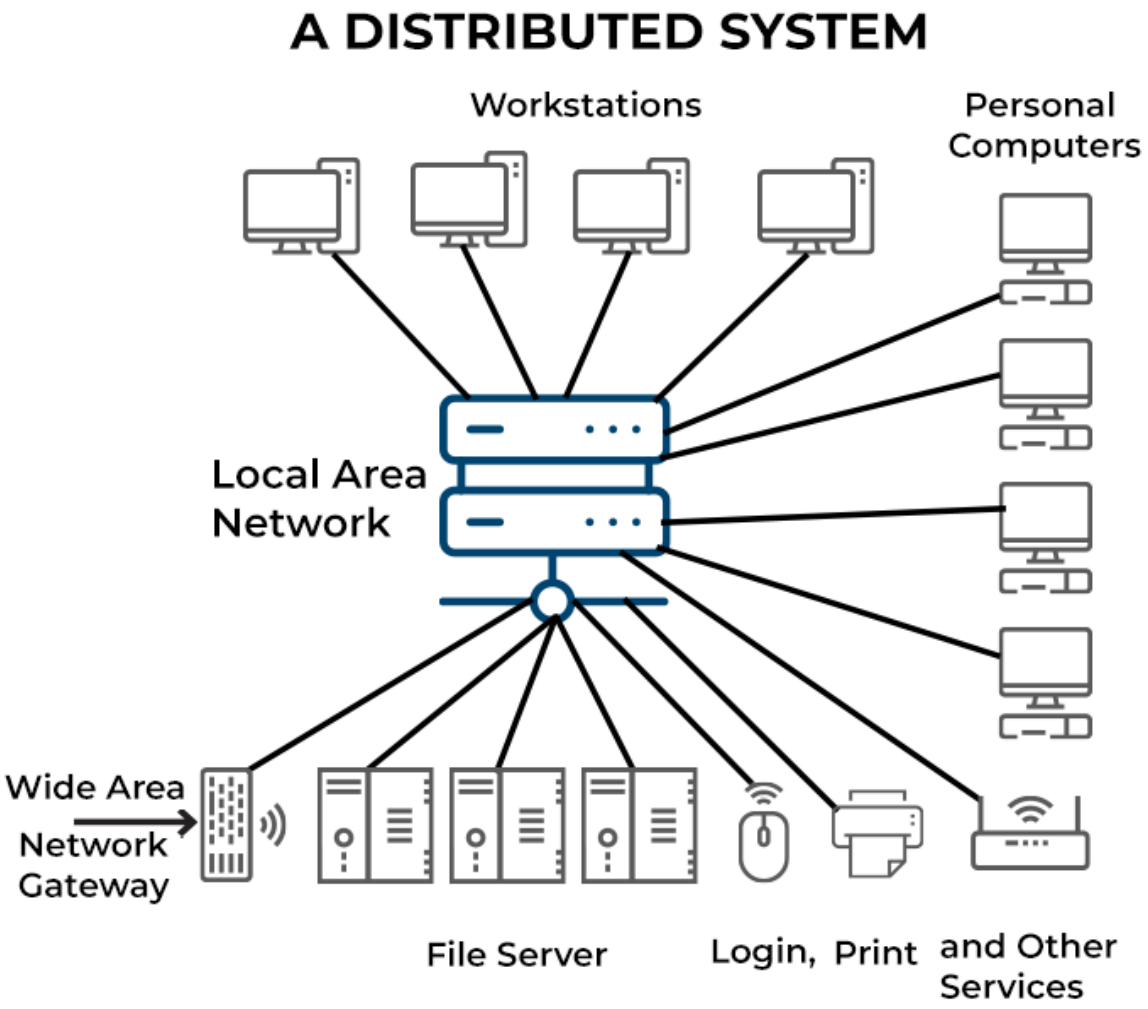

Рис. 1.2 Архітектура DCS

Одним із основних недоліків традиційних автоматизованих систем є їхня жорстка структура, яка обмежує можливості інтеграції з новітніми рішеннями, зокрема з технологіями промислового Інтернету речей (IIoT). Сучасні виробництва дедалі частіше переходять до концепції IIoT, яка передбачає підключення обладнання до єдиної мережі для збору, передачі та аналізу даних у реальному часі. Такий підхід забезпечує оперативний контроль над виробничими процесами та сприяє підвищенню їхньої ефективності.

Системи IIoT базуються на кількох ключових компонентах, які забезпечують їхню функціональність. Розумні датчики відіграють центральну роль, збираючи дані про різноманітні параметри, такі як температура, тиск, вологість, вібрація та інші показники роботи обладнання. Ці дані передаються через промислові шлюзи IoT, які забезпечують надійний зв’язок між сенсорами та центральними серверами або хмарними платформами. Для аналізу отриманих

даних широко застосовуються технології машинного навчання та штучного інтелекту, які дозволяють прогнозувати потенційні несправності обладнання, оптимізувати його роботу та попереджати аварійні ситуації. Завдяки роботі в реальному часі, високій масштабованості та можливостям централізованого управління IIoT-системи сприяють повній цифровій трансформації виробництв, мінімізуючи простої та значно підвищуючи ефективність використання обладнання. [3]

Ще одним важливим напрямом розвитку автоматизації є використання хмарних і периферійних обчислень, які оптимізують обробку даних у промислових системах. Хмарні обчислення (Cloud Computing) передбачають зберігання та обробку даних на віддалених серверах, що забезпечує доступ до потужних обчислювальних ресурсів і спрощує масштабування системи (Рис. 1.3). Натомість периферійні обчислення (Edge Computing) дозволяють обробляти дані безпосередньо на рівні пристроїв, таких як контролери чи шлюзи, що значно зменшує затримки передачі даних і знижує навантаження на локальні сервери. Поєднання цих підходів підвищує швидкість обробки інформації, забезпечує гнучкість і надійність системи, що є критично важливим для сучасних автоматизованих виробничих комплексів. [4] [5]

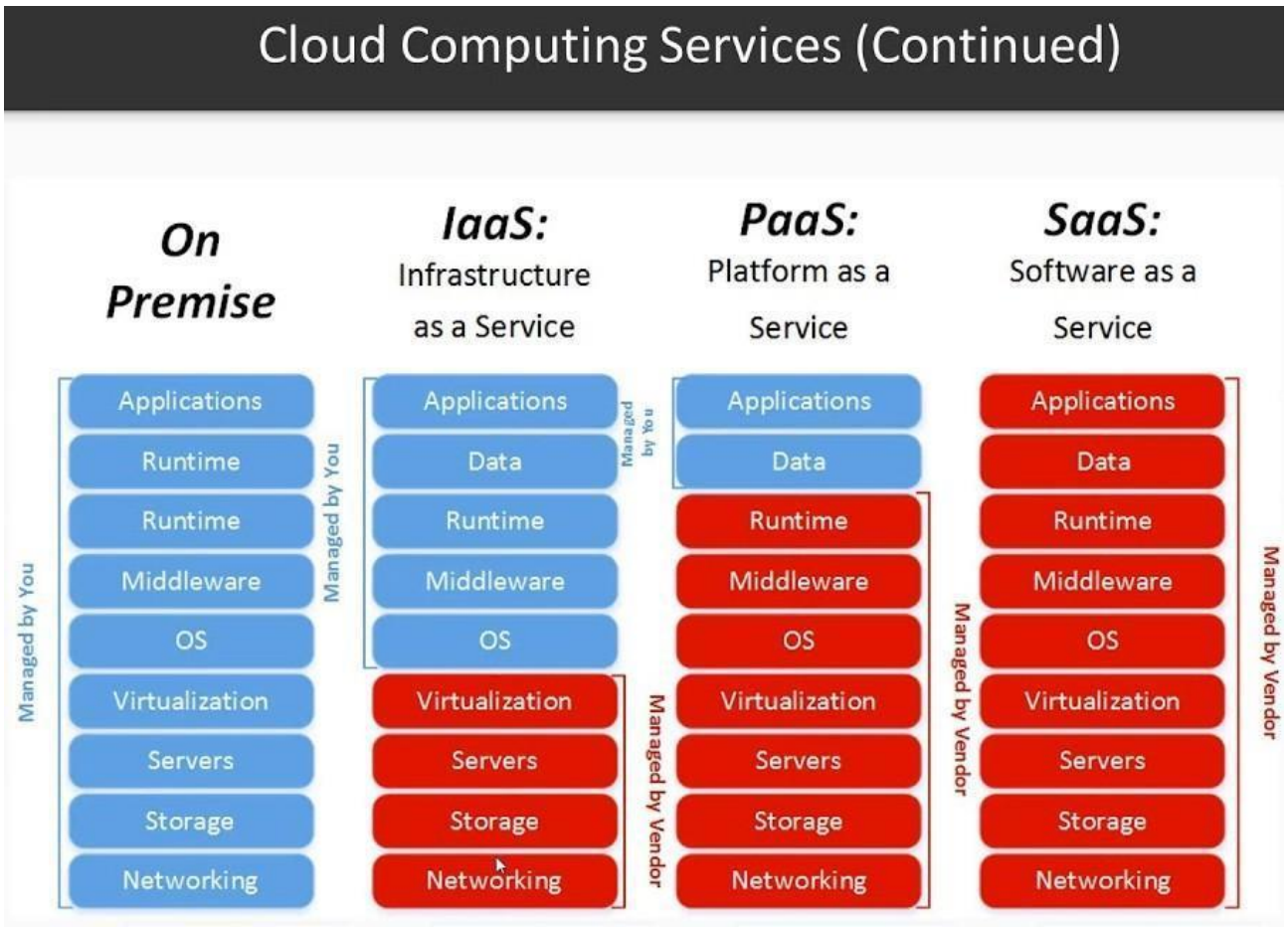

Рис. 1.3 Інфраструктура Cloud Computing Services

Важливим аспектом сучасної автоматизації є використання периферійних обчислень (Edge Computing), які знаходять застосування в сценаріях, де необхідний швидкий відгук системи та зниження навантаження на центральні сервери. Наприклад, у роботизованих виробничих лініях, де затримки в обробці даних можуть призвести до збоїв, периферійні обчислення дозволяють обробляти інформацію безпосередньо на рівні пристроїв, забезпечуючи високу оперативність і ефективність (Рис. 1.4). Такий підхід оптимізує роботу системи, зменшуючи залежність від мережевих з'єднань і підвищуючи її надійність у реальних виробничих умовах.

Для ефективного впровадження автоматизації промислових процесів необхідно дотримуватися кількох ключових принципів, які визначають успіх системи. Інтеграція IoT-рішень відіграє центральну роль, забезпечуючи використання датчиків, шлюзів і програмного забезпечення для збору та обміну інформацією між пристроями, що створює основу для динамічного моніторингу

та управління. Використання відкритих стандартів, таких як уніфіковані протоколи зв'язку MQTT, Modbus і OPC UA, спрощує інтеграцію різнорідних пристроїв, сприяючи сумісності та гнучкості системи. Застосування штучного інтелекту та машинного навчання дозволяє аналізувати великі обсяги даних, прогнозувати потенційні несправності обладнання та оптимізувати виробничі процеси, що підвищує їхню ефективність. Гнучкість і масштабованість системи забезпечують швидку адаптацію до змін у виробничому середовищі, дозволяючи розширювати функціонал без значних модифікацій. Нарешті, централізоване керування за допомогою групових політик автоматизує налаштування та управління обладнанням, знижуючи витрати часу та зменшуючи ймовірність помилок. Ці принципи формують основу для створення сучасних автоматизованих систем, здатних відповідати вимогам динамічного промислового середовища.

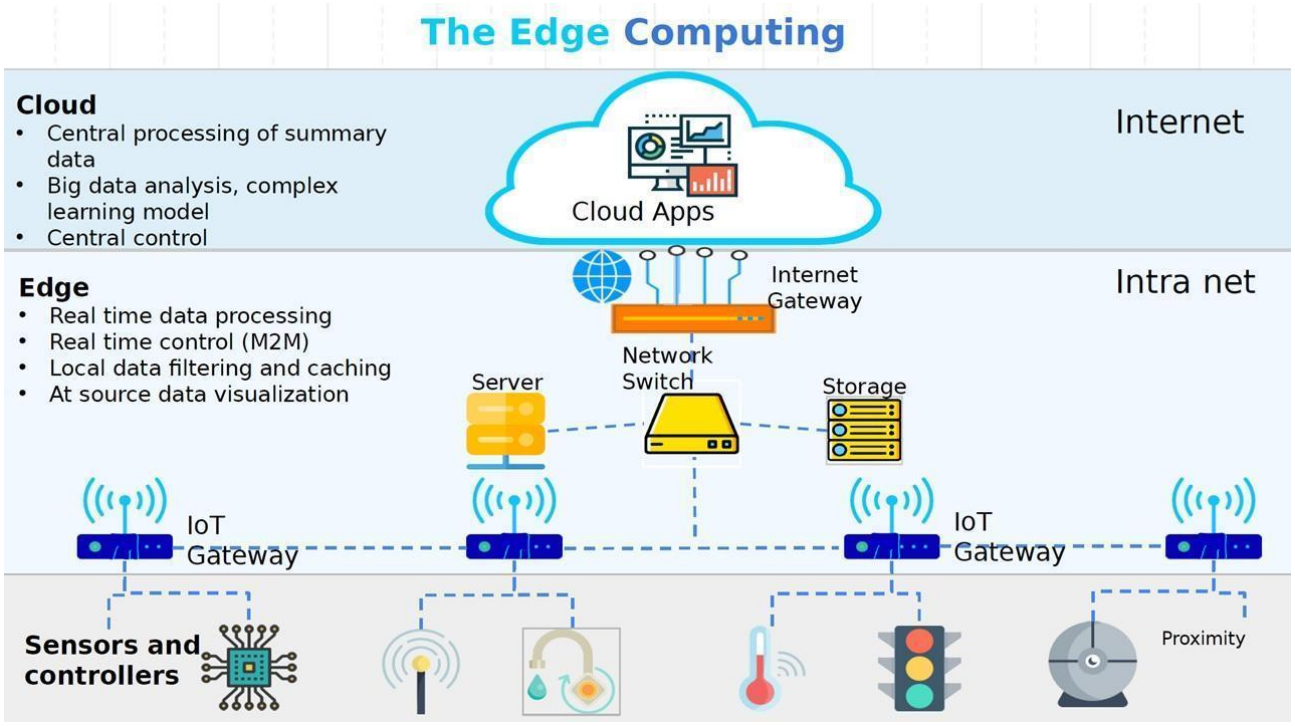

Рис. 1.4 Інфраструктура Edge Computing

1.2. Використання операційних систем Linux в автоматизації

Роль операційної системи Linux в автоматизації промислових процесів. Операційна система Linux є одним із найпоширеніших рішень для автоматизованих систем управління завдяки своїм унікальним характеристикам, які відповідають вимогам сучасного промислового середовища. Її стабільність,

гнучкість, відкритий вихідний код і підтримка широкого спектра апаратних платформ роблять Linux ідеальним вибором для використання як на рівні серверів, так і в контролерах та вбудованих пристроях. У промислових системах ця операційна система забезпечує надійність і ефективність управління виробничими процесами, дозволяючи створювати масштабовані та адаптивні рішення.

Однією з ключових переваг Linux є відкритий вихідний код, який дозволяє адаптувати систему під специфічні завдання без обмежень, пов'язаних із пропрієтарними рішеннями вендорів. Така гнучкість дає змогу розробникам оптимізувати функціонал відповідно до потреб конкретного виробництва. Linux також вирізняється високою стабільністю та надійністю, що дозволяє системі працювати безперервно протягом тривалого часу без необхідності перезавантаження, що є критично важливим для промислових застосувань. Завдяки гнучкості та масштабованості операційна система підтримує налаштування ядра, використання різних конфігурацій і сумісність із широким спектром апаратних платформ, що робить її універсальним рішенням для автоматизації. Крім того, Linux пропонує багатий набір інструментів для автоматизації, таких як системи керування конфігурацією Ansible, Puppet і Chef, технології контейнеризації, наприклад Docker, а також рішення для віртуалізації, такі як KVM і QEMU. Вбудована підтримка промислових протоколів зв'язку, зокрема Modbus, OPC UA і MQTT, забезпечує безперебійну інтеграцію з IoT-рішеннями, що є важливим для створення сучасних автоматизованих систем. [6]

Для промислових застосувань використовуються різні дистрибутиви Linux, кожен із яких має свої особливості та переваги. Ubuntu Core є мінімалістичним варіантом Ubuntu, оптимізованим для IoT-пристроїв і вбудованих систем, що робить його популярним вибором для компактних рішень. Yocto Project дозволяє створювати кастомізовані Linux-дистрибутиви, адаптовані до потреб вбудованих пристроїв, що забезпечує максимальну гнучкість. Debian та його похідні, такі як Raspberry Pi OS і Industrial Debian, широко застосовуються в промислових автоматизованих рішеннях завдяки своїй

універсальності та стабільності. Oracle Linux пропонує корпоративний рівень підтримки, що робить його придатним для великих промислових систем. Red Hat Enterprise Linux (RHEL) і CentOS вирізняються високою стабільністю та використовуються для розгортання серверів і промислових мереж, забезпечуючи надійну основу для управління складними виробничими процесами (Рис. 1.5).

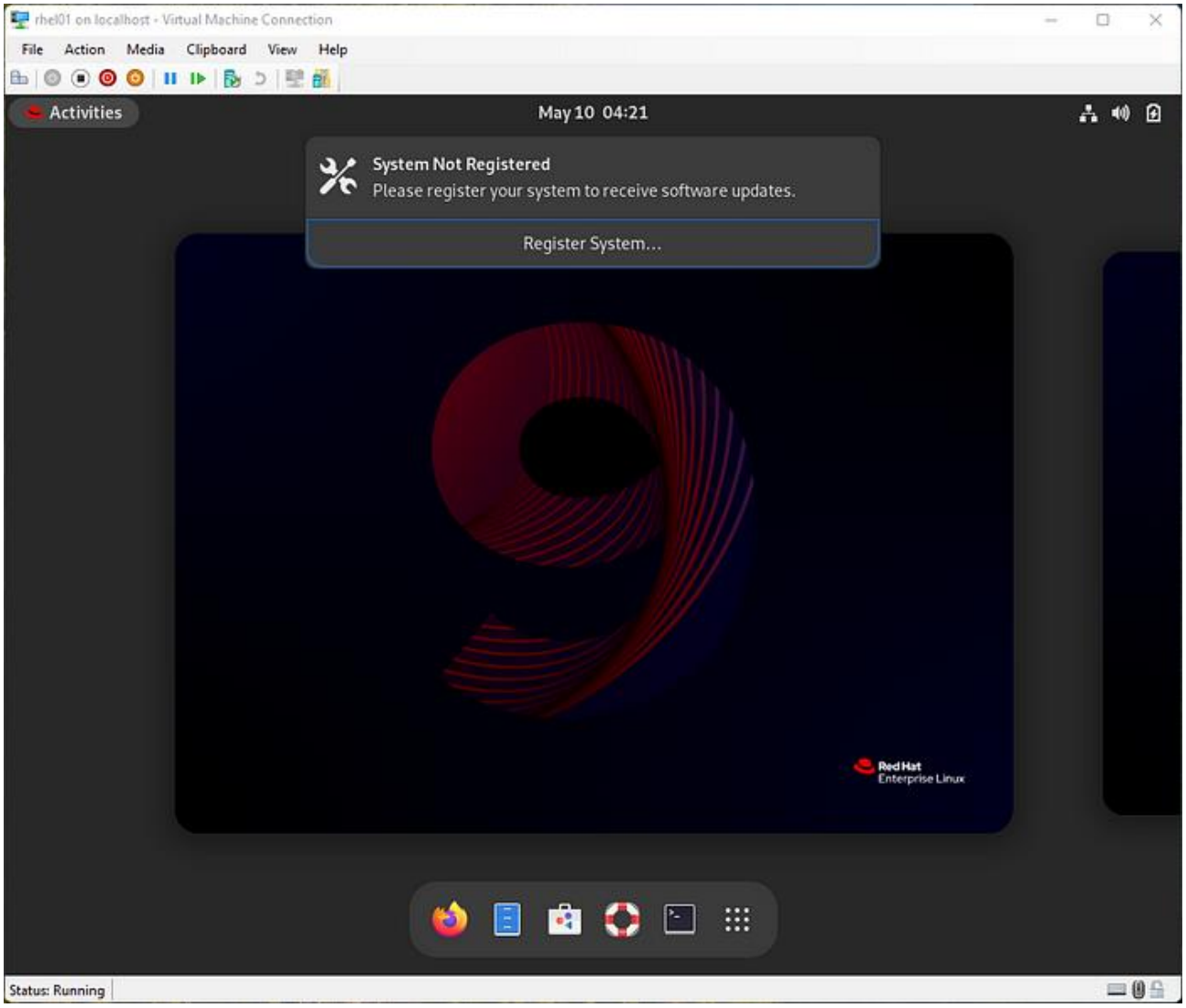

Рис. 1.5 Дистрибутив Red Hat Enterprise Linux

Приклади використання Linux у промисловості:

1. Автоматизовані виробничі лінії – контролери на базі Linux управляють процесами виробництва, обробляють дані сенсорів та забезпечують безперебійну роботу обладнання.

2. IoT-шлюзи – пристрої, що збирають інформацію з датчиків та передають її у хмарні сервіси або локальні сервери.

3. Системи віддаленого моніторингу та управління – рішення, що дозволяють здійснювати централізоване адміністрування обладнання через VPN або веб-інтерфейс.

1.3 Групові політики у промисловій автоматизації

У сучасному виробництві, де активно застосовуються автоматизоване обладнання та IoT-рішення, централізоване управління та налаштування пристроїв набувають критичного значення. У корпоративному середовищі для ефективного адміністрування комп'ютерних мереж традиційно використовуються групові політики, які дозволяють задавати єдині параметри для користувачів і пристроїв, забезпечуючи уніфіковану та безпечну роботу систем. Цей підхід дедалі частіше застосовується в промисловій автоматизації, де необхідно координувати роботу великої кількості контролерів, датчиків і серверних систем, гарантуючи їхню узгоджену взаємодію та захист від несанкціонованого доступу.

Групові політики являють собою інструмент централізованого управління налаштуваннями, який дозволяє автоматично застосовувати задані параметри до всіх підключених пристроїв, усуваючи потребу в ручному налаштуванні кожного з них. У традиційних IT-інфраструктурах цей механізм використовується для контролю доступу до ресурсів, обмеження дій користувачів і впровадження стандартних параметрів безпеки, що сприяє стабільній роботі мережевих систем. У промислових системах групові політики забезпечують автоматизацію конфігурації IoT-пристроїв, серверів і контролерів, встановлення єдиних політик безпеки для всіх елементів мережі, централізоване оновлення програмного забезпечення та застосування патчів. Крім того, вони дозволяють контролювати доступ до критично важливих ресурсів виробничого процесу та зменшувати ризик помилок, спричинених ручним втручанням адміністратора.

На відміну від Windows, де групові політики реалізуються через Active Directory, у Linux-середовищі застосовуються альтернативні механізми централізованого керування конфігураціями, які забезпечують аналогічний

рівень автоматизації та ефективності. Одним із найпоширеніших інструментів є Ansible, система керування конфігураціями, яка дозволяє централізовано оновлювати, налаштовувати та підтримувати сотні або навіть тисячі Linux-систем одночасно. Ansible працює за принципом push-моделі, передаючи зміни з центрального сервера на підключені пристрої без необхідності встановлення клієнтського програмного забезпечення, що робить її ефективним рішенням для масштабованої інфраструктури. Puppet, інший популярний інструмент, використовує pull-модель, за якої пристрої самостійно запитують налаштування з центрального сервера, забезпечуючи автоматичне оновлення та уніфікацію конфігурацій. Chef пропонує потужні можливості для автоматизації управління конфігураціями, дозволяючи централізовано контролювати параметри операційної системи та програмного забезпечення, що спрощує адміністрування та сприяє масштабуванню інфраструктури (Рис. 1.6). [7][8]

Додатково в Linux-середовищі застосовується протокол LDAP (Lightweight Directory Access Protocol), який забезпечує централізоване зберігання та керування інформацією про користувачів, групи та політики доступу. Це значно полегшує автентифікацію та адміністрування прав доступу у великих мережах серверів. SaltStack, ще одне високопродуктивне рішення, дозволяє виконувати команди в реальному часі та отримувати швидкий зворотний зв’язок від керованих пристроїв, що робить його ідеальним для динамічних IT-інфраструктур і промислових мереж. Ці інструменти спільно створюють надійну основу для централізованого управління, забезпечуючи автоматизацію, безпеку та ефективність у промислових автоматизованих системах.

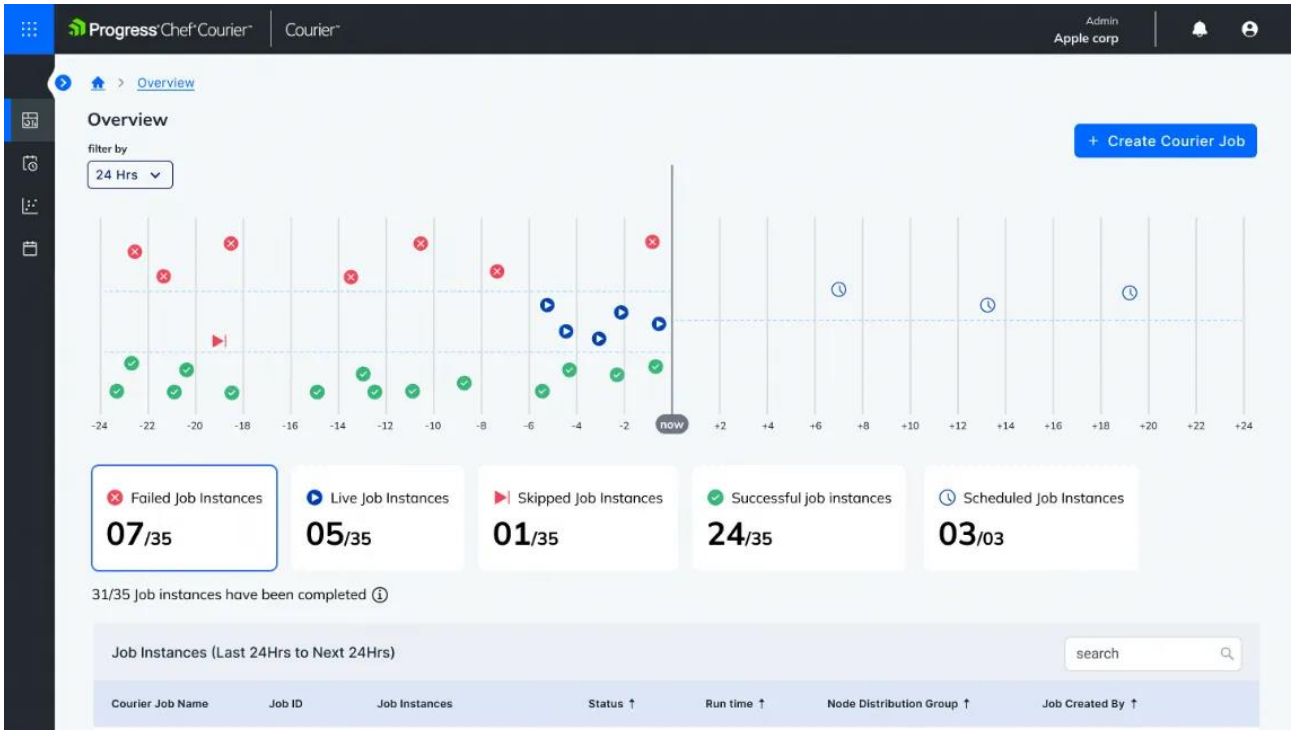

Рис. 1.6 Інструмент для автоматизації управління конфігураціями Chef

Застосування групових політик у промислових автоматизованих системах відкриває широкі можливості для оптимізації управління та підвищення ефективності виробничих процесів. Одним із ключових прикладів є централізоване оновлення прошивок контролерів, що є критично важливим у середовищах із великою кількістю вбудованих пристроїв. Єдина версія програмного забезпечення на всіх пристроях запобігає конфліктам, збоям і проблемам сумісності, сприяючи стабільності роботи обладнання. Такий підхід також спрощує адміністрування, зменшуючи витрати часу та ресурсів на технічну підтримку. Автоматичне застосування політик безпеки забезпечує відповідність усіх пристроїв єдиним стандартам захисту, мінімізуючи вразливості та ризики несанкціонованого доступу, що дозволяє підтримувати безпечне функціонування системи без ручного налаштування кожного пристрою.

Ще одним важливим аспектом є обмеження доступу до критично важливих функцій, що підвищує безпеку промислових систем. Групові політики дозволяють чітко регламентувати рівні доступу до систем керування обладнанням, гарантуючи, що лише авторизовані користувачі можуть змінювати

конфігурацію або вносити коригування у виробничі процеси. Це знижує ймовірність несанкціонованого втручання та зменшує ризики збоїв. Забезпечення безперебійної роботи системи досягається завдяки автоматичному відновленню після збоїв, постійному моніторингу стану пристроїв і регулярному резервному копіюванню даних. Такі заходи мінімізують ризики відмов, дозволяють швидко реагувати на несправності та підтримувати стабільний виробничий цикл. Контроль і аудит дій користувачів також відіграють важливу роль, оскільки ведення журналу змін дозволяє відстежувати всі коригування, аналізувати причини збоїв і оперативно реагувати на потенційні загрози, що сприяє підвищенню безпеки та оптимізації робочих процесів.

Використання групових політик у промислових умовах приносить низку суттєвих переваг. Автоматизація налаштувань значно економить час і ресурси, мінімізуючи вплив людського фактора та спрощуючи адміністрування. Гнучкість і масштабованість дозволяють легко адаптувати централізовані налаштування до змін у виробничій інфраструктурі, забезпечуючи підтримку нових пристроїв і технологій. Групові політики підвищують рівень безпеки, гарантуючи відповідність усіх пристроїв вимогам кібербезпеки, що є особливо важливим у сучасних автоматизованих системах. Централізоване управління виключає ризик людських помилок під час налаштування, що сприяє стабільності роботи обладнання. Інструменти, такі як Ansible, Puppet, Chef і LDAP, забезпечують ефективне централізоване управління великими виробничими мережами, дозволяючи оптимізувати робочі процеси, підвищувати надійність і гнучкість систем, а також гарантувати безпеку промислових процесів.

1.4 Комунікаційні технології та протоколи IoT для автоматизації

У сучасній промисловій автоматизації, де Інтернет речей (IoT) відіграє центральну роль, організація ефективного обміну даними між пристроями, серверами та операторами є критично важливою. Від якості комунікацій залежить стабільність роботи системи, швидкість обробки інформації, рівень безпеки та можливість своєчасного прийняття управлінських рішень. Сучасні

виробничі підприємства дедалі частіше застосовують розподілені IoT-системи, у яких тисячі датчиків, контролерів і серверів взаємодіють у реальному часі. У таких умовах вибір відповідного протоколу зв’язку та комунікаційної технології стає ключовим фактором, адже різні типи промислового обладнання мають специфічні вимоги до швидкості, надійності та безпеки передавання даних.

Комунікаційні протоколи та технології IoT виконують низку важливих функцій, які забезпечують ефективну роботу автоматизованих систем. Однією з основних є моніторинг стану обладнання, що передбачає передачу оператору інформації про такі параметри, як температура, рівень вологості, тиск чи споживання електроенергії. Віддалене керування пристроями дозволяє системі оперативно надсилати команди для вмикання чи вимикання обладнання, зміни режимів роботи або налаштування параметрів. Автоматична діагностика та усунення несправностей забезпечує швидке виявлення проблем і передачу даних про них, що допомагає запобігати аваріям і мінімізувати простої. Взаємодія з хмарними сервісами, такими як Google Cloud, AWS чи Microsoft Azure, вимагає безперебійного потоку інформації між пристроями та серверною інфраструктурою для зберігання та обробки даних. Важливим аспектом є безпека передавання даних, адже перехоплення чи підміна інформації в промислових системах може мати катастрофічні наслідки.

Вибір комунікаційної технології залежить від вимог конкретної автоматизованої системи, враховуючи такі параметри, як радіус покриття, швидкість передавання даних, енергоспоживання та вартість реалізації. Для критично важливих виробничих процесів, де потрібна висока швидкість, низька затримка та стабільність зв’язку, перевага надається дротовим з’єднанням. Ethernet є класичним стандартом дротового підключення, який широко застосовується в промислових мережах завдяки своїй надійності та високій пропускній здатності. Modbus RTU, популярний індустріальний стандарт, використовується для обміну даними між контролерами та датчиками, забезпечуючи простоту й ефективність комунікації. OPC UA (Open Platform Communications Unified Architecture) вирізняється як універсальний протокол,

що забезпечує обмін даними між різними виробничими системами, сприяючи їхній інтеграції та сумісності (Рис. 1.7).

Рис. 1.7 Інтерфейс Modbus RTU

Для невеликих промислових об’єктів або локальної взаємодії між пристроями широко застосовуються бездротові комунікаційні технології, які забезпечують гнучкість і мобільність. Wi-Fi є популярним вибором для промислових IoT-рішень завдяки високій швидкості передавання даних, хоча його використання обмежується високим енергоспоживанням. Bluetooth Low Energy (BLE) ідеально підходить для пристроїв із низьким споживанням енергії, що робить його ефективним для локальних мереж із сенсорами. ZigBee використовується в розумних фабриках для зв’язку між датчиками, забезпечуючи надійну передачу даних у компактних системах. Для зв’язку між пристроями, розташованими на великій відстані, застосовуються технології, такі як LoRaWAN, який дозволяє передавати малі обсяги даних на значні відстані з низьким енергоспоживанням. Sigfox, енергозберігаючий протокол, працює в глобальних мережах IoT, забезпечуючи економічне рішення для розподілених систем. Сучасні технології, такі як 5G і NB-IoT, пропонують високу швидкість

передавання даних і підтримують інтеграцію промислових IoT-систем, що робить їх перспективними для масштабних виробничих мереж.

У промислових автоматизованих системах застосовується широкий спектр протоколів зв’язку, кожен із яких має свої особливості та сфери використання. MQTT (Message Queuing Telemetry Transport) є одним із найпоширеніших протоколів у IoT завдяки своїй здатності передавати малі обсяги даних із низьким енергоспоживанням і створювати надійні з’єднання навіть у нестабільних мережах. CoAP (Constrained Application Protocol) оптимізований для пристроїв із обмеженими ресурсами, що робить його ефективним для обміну даними в промислових мережах із низькою пропускною здатністю. Modbus TCP, варіант класичного протоколу Modbus, працює поверх TCP/IP і широко використовується в автоматизованих системах керування виробництвом завдяки своїй простоті та надійності. AMQP (Advanced Message Queuing Protocol) орієнтований на складні IoT-системи, підтримуючи складні черги повідомлень, що є важливим для великих підприємств із високими вимогами до обробки даних. HTTPS і WebSockets застосовуються для інтеграції IoT-пристроїв із веб-додатками, забезпечуючи надійний і зашифрований обмін даними, що підвищує безпеку комунікацій. [9] [10]

Комунікаційні технології та протоколи IoT відіграють вирішальну роль у забезпеченні ефективності сучасної промислової автоматизації. Вибір відповідної технології залежить від конкретних вимог системи, таких як швидкість передавання даних, енергоспоживання, дальність зв’язку та рівень безпеки. Дротові технології, такі як Ethernet, Modbus RTU і OPC UA, залишаються найстабільнішим рішенням для критично важливих виробничих процесів, де потрібна висока надійність і мінімальна затримка. Бездротові технології, зокрема LoRaWAN, NB-IoT і 5G, забезпечують мобільність і масштабованість, дозволяючи створювати гнучкі розподілені системи. Протоколи зв’язку, такі як MQTT і Modbus TCP, гарантують ефективну передачу даних між пристроями, забезпечуючи безперебійну роботу автоматизованих систем. Правильний вибір протоколу та технології є ключовим фактором для

стабільності, безпеки та успішної реалізації промислових IoT-рішень, що сприяє підвищенню ефективності виробничих процесів.

1.5 Висновок до розділу 1

У першому розділі здійснено детальний аналіз сучасних підходів до автоматизації промислових систем, які базуються на технологіях Інтернету речей (IoT), операційних системах Linux і механізмах групових політик. Розглянуто ключові концепції, що визначають ефективність впровадження автоматизованих рішень у виробничі процеси, а також досліджено особливості комунікаційних технологій, які забезпечують стабільний і безпечний обмін даними між пристроями. Проведений аналіз дозволив сформувати цілісне уявлення про сучасні тенденції та інструменти, що лежать в основі промислової автоматизації.

Сучасні тенденції автоматизації спрямовані на активне використання IoT-рішень, які забезпечують безперервний моніторинг стану обладнання, оптимізацію виробничих процесів і прогнозування можливих несправностей. Завдяки розподіленим IoT-мережам підприємства можуть значно підвищити продуктивність, покращити контроль над обладнанням і знизити витрати на його обслуговування. Операційні системи Linux відіграють важливу роль у цих процесах завдяки своїй стабільності, безпеці, гнучкості та підтримці відкритих стандартів. Спеціалізовані дистрибутиви, розроблені для вбудованих систем і IoT, дозволяють підприємствам адаптувати Linux до специфічних виробничих завдань, оптимізуючи керування обладнанням і технологічними процесами.

Механізми групових політик забезпечують централізоване управління налаштуваннями та конфігураціями обладнання, що значно спрощує адміністрування промислових мереж. Використання інструментів, таких як Ansible, Puppet, Chef і SaltStack, дозволяє автоматизувати розгортання систем, оновлення прошивок, контроль доступу та управління безпекою пристроїв у реальному часі. Це підвищує ефективність і надійність інфраструктури, знижуючи ризик помилок і витрати на адміністрування. Комунікаційні технології та протоколи IoT, зокрема MQTT, CoAP, Modbus TCP і OPC UA, гарантують безперебійну роботу автоматизованих систем. Вибір протоколу

залежить від вимог конкретного виробничого процесу, таких як швидкість, дальність зв’язку, енергоспоживання та рівень безпеки.

Інтеграція IoT у промислові системи вимагає комплексного підходу, який охоплює вибір відповідної операційної системи, налаштування комунікаційної інфраструктури, забезпечення інформаційної безпеки та ефективне адміністрування через централізовані політики управління. Такий підхід дозволяє скоротити витрати на технічну підтримку й адміністрування обладнання, підвищити безпеку та стабільність роботи системи, а також забезпечити гнучкі методи моніторингу, діагностики та централізованого контролю доступу до пристроїв. Отримані результати створюють міцну основу для практичного впровадження автоматизованої системи управління обладнанням, що базується на Linux, IoT і групових політиках. Детальніший розгляд практичної реалізації цього підходу представлено в наступному розділі роботи.

РОЗДІЛ 2. ФОРМУВАННЯ КОМПОНЕНТІВ СИСТЕМИ ТА ЇХ ОПИС

2.1 Концепція системи, її архітектура, взаємодія компонентів та принципи роботи

У сучасних умовах цифрової трансформації промисловості автоматизовані системи управління повинні відповідати високим вимогам до надійності, масштабованості, безпеки та гнучкості. Враховуючи ці аспекти, у межах даної дипломної роботи була розроблена концепція системи управління промисловим обладнанням, яка ґрунтується на використанні операційної системи Linux, технологій Інтернету речей (IoT) та централізованого управління за допомогою групових політик.

Суть запропонованого підходу полягає у створенні адаптивного середовища, яке забезпечує ефективну взаємодію між фізичними пристроями, серверами обробки даних та інтерфейсами оператора. Основними завданнями системи є постійний моніторинг технічного стану обладнання, реалізація сценаріїв реагування на події, автоматизація розгортання програмного забезпечення та централізоване адміністрування.

Архітектура системи має багаторівневу модульну структуру. На нижньому рівні функціонують пристрої збору даних — сенсори, датчики температури, вібрації, навантаження, які підключаються до контролерів на базі Linux-платформ (наприклад, Raspberry Pi або BeagleBone). Ці вузли забезпечують первинний збір та передачу даних у реальному часі за допомогою легких протоколів, таких як MQTT або Modbus TCP, що дозволяє знизити навантаження на мережу й підвищити стабільність передачі інформації.

На середньому рівні архітектури розташовується сервер обробки та керування, що виконує роль центрального логічного ядра. Саме тут реалізується аналіз вхідних даних, обробка подій згідно з визначеними правилами, прийняття рішень і формування команд для виконавчих пристроїв. В рамках цієї частини системи використовується програмне забезпечення Node-RED для побудови логіки управління, брокер MQTT Mosquitto для комунікації між компонентами,

а також інструменти на кшталт Ansible або Puppet для автоматизованого розгортання та підтримки пристроїв. [11]

Верхній рівень архітектури забезпечує інтерфейс взаємодії з оператором. Дані, зібрані та оброблені на сервері, передаються до візуалізаційного модуля, реалізованого у вигляді веб-інтерфейсу або SCADA-системи. Зокрема, для цієї роботи було використано Node-RED Dashboard та Grafana, що забезпечують зручний доступ до основних показників роботи обладнання, перегляду журналу подій та керування системою через мобільні або стаціонарні пристрої. Така візуалізація дозволяє своєчасно виявляти аномалії, запускати сценарії, а також змінювати параметри роботи обладнання. [12] [13]

Особливістю системи є відкритість її структури та використання стандартизованих протоколів взаємодії. Вся комунікація між компонентами відбувається через MQTT-брокер, REST API або WebSocket, що дозволяє легко масштабувати систему, додавати нові вузли або модулі без порушення загальної цілісності. Для керування доступом реалізовано механізм централізованої аутентифікації (наприклад, LDAP або SSH-ключі), а адміністрування здійснюється дистанційно через інструменти автоматизації.

Принципи функціонування системи базуються на кількох ключових механізмах. По-перше, автоматичне виявлення нових пристроїв дозволяє підключати контролери, які приєднуються до мережі, автоматично отримувати конфігурацію та інтегруватися в загальну інфраструктуру. По-друге, централізоване управління політиками забезпечує можливість задавати єдині правила безпеки, оновлення та доступу до пристроїв без необхідності ручного втручання. Третім принципом є подієва логіка: система реагує на певні тригери (наприклад, перевищення температури) шляхом виконання заздалегідь визначених дій. Також важливими є дистанційне адміністрування — можливість повноцінного контролю з будь-якої точки мережі — та механізми збору статистики, що забезпечують аналітику роботи всіх вузлів.

Загалом, така архітектура дозволяє досягти високого рівня надійності, стійкості до збоїв, простоти в обслуговуванні та адаптивності системи до потреб

конкретного підприємства. Її можна використовувати в різних сферах промисловості, включаючи виробництво, енергетику, логістику чи агросектор, без необхідності суттєвої модифікації базової структури.

2.2 Вибір апаратних і програмних компонентів

Після формування загальної концепції системи та побудови її архітектури наступним етапом стала розробка апаратно-програмної основи, здатної забезпечити стабільну й ефективну роботу системи в умовах промислової експлуатації. Вибір технічних і програмних засобів здійснювався з урахуванням специфіки середовища — це вимоги до надійності, енергоефективності, простоти розгортання, підтримки відкритих протоколів і сумісності з операційною системою Linux.

Апаратна частина проєкту поділяється на кілька функціональних рівнів. Центральною складовою системи виступають обчислювальні вузли — контролери, встановлені безпосередньо поблизу обладнання. Основними критеріями відбору таких пристроїв стали підтримка сучасних Linux-дистрибутивів, наявність GPIO-портів, стабільність у режимі тривалого безперервного функціонування та можливість інтеграції з MQTT і Modbus. Серед платформ, які задовольнили ці вимоги є Raspberry Pi 4B, Orange Pi 3 LTS та BeagleBone Black Industrial. Кожна з них має свої особливості, однак усі дозволяють виконувати локальну обробку сигналів, реалізовувати сценарії керування виконавчими механізмами та взаємодіяти із сенсорними модулями в режимі реального часу.

Підключені до контролерів сенсори забезпечують збір актуальних даних про стан обладнання та навколишнє середовище. Зокрема, цифрові температурні сенсори використовуються для вимірювання кліматичних параметрів, а, наприклад, GY-521 (на основі чипа MPU-6050) дозволяє виявляти вібраційні зміни, що можуть вказувати на механічні відхилення або зношення обладнання. Управління виконавчими пристроями, такими як вентилятори чи електромагнітні клапани, здійснюється через релейні модулі, адаптовані до робочої напруги 5 або 12 вольт.

З метою забезпечення надійної мережевої взаємодії між компонентами системи, передбачено використання промислових Ethernet-комутаторів для дротового з’єднання, а також бездротових модулів (Wi-Fi, LoRa) для реалізації мобільних або розподілених рішень. Для захисту каналу зв’язку та організації безпечного віддаленого доступу інтегрується VPN-інфраструктура на базі маршрутизаторів MikroTik або TP-Link із підтримкою протоколів IPsec/OpenVPN.

Програмне забезпечення системи обиралося відповідно до декількох ключових критеріїв: сумісність з операційною системою Linux, підтримка автоматизації процесів, відкритість архітектури, активна спільнота розробників і широка інтеграція з IoT-протоколами. В якості операційного середовища для вузлів обрано Ubuntu Server 22.04, Debian та Raspberry Pi OS Lite — усі три системи добре зарекомендували себе у проєктах автоматизації та не потребують значних ресурсів. Основна платформа автоматизації — Node-RED — використовується для побудови сценаріїв керування, обробки вхідних даних та реалізації логіки подій.

Паралельно з нею, реалізовані допоміжні скрипти на Python із використанням бібліотек paho-mqtt та pyModbusTCP, що дозволяють розширити функціональність взаємодії з мережею сенсорів. В якості брокера повідомлень, через який відбувається обмін інформацією між вузлами, обрано Mosquitto — легкий, надійний і підтримуваний рішення з відкритим кодом. Для збору та зберігання телеметричних даних, а також побудови графіків і дашбордів, використовується стек InfluxDB + Grafana. [14]

У процесі відбору як апаратних, так і програмних компонентів, особлива увага приділялася стабільності роботи в умовах високого навантаження, простоті масштабування, відсутності прив’язки до конкретного виробника, а також потенціалу до подальшої модернізації системи. Усі обрані технології відзначаються високим рівнем підтримки в спільноті, наявністю повноцінної документації та регулярними оновленнями.

Таким чином, сформований комплекс програмних і апаратних засобів не тільки задовольняє вимогам проєкту, але й закладає основу для подальшого розширення, модернізації та адаптації під нові завдання. Завдяки гнучкій архітектурі, системна інфраструктура легко масштабується, зберігаючи при цьому стабільність, прозорість адміністрування та високий рівень безпеки.

2.3 Програмне забезпечення та налаштування

Стабільність і ефективність роботи автоматизованої системи управління промисловим обладнанням значною мірою визначаються не лише правильно підібраним програмним забезпеченням, а й коректно налаштованим середовищем, у якому відбувається взаємодія всіх компонентів. Від узгодженості конфігурації системних служб, протоколів зв'язку та безпекових механізмів залежить не тільки швидкість обробки даних, але й загальний рівень надійності та стійкості системи до збоїв.

Центральним вузлом у запропонованій архітектурі виступає сервер, який виконує роль логічного ядра системи, відповідає за прийом, обробку та маршрутизацію даних, а також за виконання сценаріїв автоматизації. Як базова операційна система для цього серверу обрано Ubuntu Server 22.04 або Oracle Linux 8.x — обидва дистрибутиви забезпечують достатній рівень захищеності, мають розвинену інфраструктуру підтримки та є сумісними з ключовими інструментами, що використовуються в системі.

Процес налаштування серверного середовища відбувається поетапно. На початковому етапі виконується поділ диска на логічні розділи із врахуванням доцільності окремого монтування директорій /var, /home та /etc, що дозволяє підвищити контроль над зберіганням журналів, налаштувань користувачів і системних конфігурацій. Далі встановлюється та налаштовується служба SSH-доступу, з обов'язковим обмеженням прямого доступу до root-акаунту, що є базовою вимогою з інформаційної безпеки.

Мережеве середовище серверу конфігурується з урахуванням статичної IP-адреси, налаштування фаєрволу (UFW або firewalld залежно від дистрибутиву) та встановлення VPN-клієнта для захищеного доступу ззовні. Після цього

виконується інсталяція критично важливих пакетів — серед них docker, python3, mosquitto, git, curl, htop, vim, які забезпечують як виконання логіки автоматизації, так і зручність адміністрування системи.

Важливу частину займає налаштування облікових записів та контролю доступу. На сервері створюються окремі користувачі для адміністратора, служби моніторингу, агента Ansible, а також визначаються права доступу до окремих системних ресурсів. Генерація пар SSH-ключів, налаштування sudo-груп та впровадження інструментів блокування підозрілої активності (наприклад, fail2ban) дозволяють значно знизити ризики несанкціонованого доступу та автоматизованих атак.

Ще одним критичним елементом є система комунікації між пристроями. У запропонованій архітектурі реалізовано подієво-орієнтовану модель передачі даних за допомогою протоколу MQTT(Рис. 2.8), що ідеально підходить для розподілених IoT-рішень. Як брокер повідомлень обрано Mosquitto — легковагове й ефективне рішення з відкритим кодом. Його конфігурація передбачає визначення портів обміну (стандартний TCP-порт 1883 та захищений SSL-порт 8883), налаштування базової автентифікації за іменем користувача й паролем, впровадження SSL/TLS-сертифікатів для шифрування переданих даних, а також встановлення політик обслуговування клієнтів (QoS, timeout, зберігання повідомлень тощо). [15]

Завдяки такому комплексному підходу до налаштування серверного середовища система отримує не лише високу функціональну готовність до роботи, але й фундамент для подальшого масштабування, безпечного адміністрування та стабільного функціонування в умовах промислового середовища.

```
factory/temp/sensor1
factory/vibration/motor5
factory/control/relay2/set
```

Рис. 2.8 Типові MQTT-топіки

Центральним компонентом логіки обробки подій у системі виступає середовище Node-RED — гнучкий візуальний інструмент, розроблений для створення сценаріїв автоматизації з використанням графічного інтерфейсу. Завдяки підтримці потокової обробки даних, Node-RED дозволяє ефективно організовувати взаємодію між сенсорами, виконавчими пристроями, базами даних і зовнішніми сервісами. В рамках запропонованої системи цей інструмент виконує функції прийому даних, обробки логіки подій, запуску керуючих скриптів, формування повідомлень для користувача, а також інтеграції з панеллю оператора.

Типовий функціональний ланцюг Node-RED-потоку включає вузол прийому MQTT (який отримує дані від сенсорів у реальному часі), функціональні блоки на JavaScript для попередньої обробки та фільтрації даних, модулі запуску зовнішніх скриптів або HTTP-запитів для керування обладнанням, а також елементи, що відповідають за зберігання інформації в базу даних InfluxDB чи виведення її у вигляді графіків, індикаторів, вимірювальних шкал на веб-інтерфейсі.

Крім стандартного функціоналу Node-RED, у проєкті активно використовуються допоміжні скрипти, написані на мові Python. Вони забезпечують роботу з апаратними інтерфейсами низького рівня, такими як SPI та I2C, через які зчитуються аналогові дані з сенсорів. Також за допомогою бібліотек GPIO реалізовано керування релейними модулями, необхідними для ввімкнення/вимкнення виконавчих пристроїв. Окрему роль відіграють модулі нотифікації: у разі виникнення критичних подій або помилок система здатна надсилати повідомлення через Telegram-бот або електронну пошту, що значно підвищує рівень оперативності реагування.

У рамках централізованого адміністрування та управління конфігурацією вузлів використовується інструмент Ansible — система керування конфігураціями, що дозволяє автоматизувати рутинні задачі, пов’язані з розгортанням, оновленням і налаштуванням програмного середовища. Його застосування забезпечує масштабованість і консистентність середовища на

різних пристроях. За допомогою Ansible реалізуються типові сценарії, які включають встановлення необхідного програмного забезпечення (MQTT-клієнтів, бібліотек, моніторингових агентів), конфігурування часу, безпеки, логування, а також генерацію конфігураційних файлів, таких як mosquitto.conf, node-red-flows.json, з використанням шаблонів. Крім того, Ansible забезпечує автоматичний розгорт Docker-контейнерів із середовищами Node-RED, Telegraf або іншими сервісами, необхідними для забезпечення функціонування системи.

Завдяки інтеграції Node-RED, Python-скриптів та централізованого інструментарію на кшталт Ansible, реалізована система отримує потужну, гнучку логіку, що легко адаптується до змін середовища, масштабування та специфічних виробничих задач. Усі зміни конфігурацій виконуються централізовано, без необхідності втручання у кожен вузол окремо, що значно знижує навантаження на IT-персонал та мінімізує ризики помилок при розгортанні оновлень або політик доступу.(Рис. 2.9)

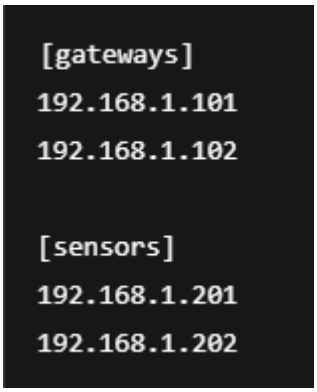

Рис. 2.9 Інвентаризація пристроїв

Важливим елементом функціонування будь-якої розподіленої системи керування є її здатність до постійного самоконтролю, що забезпечується впровадженням засобів моніторингу. У межах реалізованої архітектури ця функція виконується за допомогою поєднання трьох взаємопов’язаних сервісів — InfluxDB, Telegraf та Grafana. Разом вони формують гнучкий стек для збирання, зберігання та візуалізації часових рядів даних, що дозволяє ефективно відслідковувати зміни параметрів системи у режимі реального часу.

Базою для зберігання метрик виступає InfluxDB — спеціалізована СКБД, оптимізована під роботу з великими обсягами даних у вигляді часових рядів. У

неї надходить інформація з різних джерел — як із зовнішніх сенсорів (температура, вібрація, вологість), так і з внутрішніх системних компонентів, зокрема навантаження на процесор, використання оперативної пам’яті, мережевий трафік, статус реле або індикатори помилок. Дані збираються за допомогою агента Telegraf, який встановлюється на кожен контролер або сервер, що бере участь в автоматизованому процесі. Telegraf, у свою чергу, має широкий набір плагінів і конфігурується для збору як системних, так і специфічних метрик, що відповідають особливостям виробничого середовища.

Для аналітики та зручного представлення інформації оператору використовується Grafana — сучасний візуалізаційний інструмент, що дозволяє створювати інтерактивні дашборди з графіками, віджетами, тривогами та історією подій. Вона інтегрується з InfluxDB, надаючи в реальному часі зручні засоби моніторингу стану всієї системи, окремих пристроїв або критичних вузлів. Окрім графічного представлення, Grafana може формувати оповіщення у вигляді повідомлень, які надсилаються по електронній пошті або в месенджери при перевищенні заданих порогів (наприклад, при перегріві обладнання або втраті зв’язку з вузлом). [16]

Особливе значення в рамках даного проєкту приділено питанням інформаційної безпеки. Всі з’єднання між компонентами системи — включаючи MQTT, базу даних, веб-інтерфейси — захищені через шифрування SSL/TLS або тунелювання трафіку за допомогою VPN (WireGuard або OpenVPN). Для контролю доступу використовується або LDAP-аутентифікація, або система локальних облікових записів, при цьому реалізовано політики ротації паролів та обмеження доступу за ролями. Такий підхід гарантує як безпечну взаємодію між вузлами, так і дотримання вимог до захисту персональних і технічних даних.

Крім цього, система включає засоби резервного копіювання та централізованого журналювання. Для збереження конфігурацій, користувацьких даних і стану ОС застосовуються інструменти rsnapshot або restic, які забезпечують інкрементні копії та можливість швидкого відновлення у разі збоїв. База InfluxDB регулярно експортується за допомогою вбудованих

механізмів influx backup. Усі журнали системи, у тому числі повідомлення про помилки, сповіщення або сигнали від служб, агрегуються за допомогою rsyslog і можуть перенаправлятись у централізовану систему зберігання для подальшого аудиту.

Комплексне впровадження моніторингового стеку, резервного копіювання та засобів безпеки не лише підвищує стійкість системи до відмов, а й гарантує прозорість її роботи, забезпечуючи можливість швидкого аналізу подій, прийняття рішень і підтримки відповідності політикам підприємства щодо IT-безпеки.

2.4 Програмні модулі та логіка роботи системи

У межах автоматизованої системи управління, розробленої в рамках даного проєкту, особливе значення має програмний рівень, який забезпечує безперебійну взаємодію всіх компонентів — від сенсорів і виконавчих пристроїв до серверної логіки та інтерфейсів користувача. Завданням цього рівня є реалізація інтелектуальної логіки прийняття рішень на основі даних, що надходять у реальному часі з обладнання, з урахуванням встановлених групових політик і поточного стану системи.

Структурна побудова програмної частини системи є модульною. Такий підхід дозволяє не лише забезпечити чітке розділення функціональних обов’язків між програмними блоками, але й досягти високого рівня масштабованості, адаптивності та прозорості в адмініструванні. Завдяки використанню відкритих протоколів обміну — зокрема MQTT, REST API та форматів JSON — усі модулі легко взаємодіють між собою незалежно від мови програмування або платформи реалізації.

Одним із ключових елементів логіки є модуль збору та попередньої обробки даних. Саме він забезпечує отримання актуальної інформації з фізичних сенсорів, первинну перевірку даних і підготовку повідомлень до подальшої обробки в системі. На практиці цей модуль реалізується за допомогою Python-скриптів або вузлів Node-RED, що взаємодіють із фізичними інтерфейсами типу GPIO, I2C або SPI, забезпечуючи зчитування температури, вібрації, рівня

вологості та інших параметрів. Для забезпечення достовірності й стабільності системи впроваджуються алгоритми згладжування, фільтрації та нормалізації значень — наприклад, за допомогою ковзного середнього чи перевірки допустимих меж.

Після попередньої обробки дані формуються у структуровані пакети формату JSON(Рис. 2.10), що дозволяє їх безперешкодно передавати до MQTT-брокера. Цей підхід уніфікує комунікацію в системі, дозволяє використовувати єдину логіку обробки на стороні сервера й забезпечує високу швидкодію в умовах великої кількості інформаційних запитів.

Таким чином, на цьому рівні закладається основа для всього подальшого циклу — від візуалізації та зберігання даних до активації сценаріїв реагування, зміни параметрів керування або нотифікації користувача у випадку виявлення відхилень.

Рис. 2.10 Дані температурного сенсора (JSON)

Наступним ключовим елементом є модуль подієвої логіки, який відіграє визначальну роль у забезпеченні адаптивного реагування системи на зміну станів виробничого середовища. Цей програмний компонент відповідає за реалізацію умовного керування на основі принципу "якщо... то...", що дозволяє оперативно виконувати задані дії у випадку настання певних подій або порушення встановлених параметрів.

Логіка даного модуля базується на наборі правил, що формуються адміністратором системи у вигляді конфігураційних сценаріїв або скриптів. Наприклад, одним із базових сценаріїв є: "якщо температура перевищує 70 °C — активувати систему охолодження та повідомити відповідального оператора".

Такі правила можуть охоплювати широкий спектр умов: вібрацію понад допустиму межу, втрату зв'язку з вузлом, надмірне навантаження на мережу, або комбінацію декількох факторів.

Технічно реалізація модуля здійснюється у середовищі Node-RED або з використанням обробників на мові Python, залежно від складності сценаріїв та інфраструктурних вимог. Node-RED(Рис. 2.11) надає зручний графічний інтерфейс для побудови потоків обробки, де кожне правило задається у вигляді пов’язаних блоків: прийом події, перевірка умови, виконання дій. Python-скрипти, у свою чергу, дають більше контролю над логікою, дозволяючи реалізовувати складні алгоритми аналізу, асинхронну обробку або інтеграцію з зовнішніми сервісами.

Однією з переваг реалізованої архітектури є централізоване зберігання політик керування. Це означає, що створені адміністратором правила можуть бути збережені на центральному сервері й автоматично застосовуватись до всіх пристроїв системи за допомогою механізмів конфігураційного управління, зокрема інструмента Ansible. Такий підхід дозволяє уникнути ручного дублювання конфігурацій, знизити ймовірність помилок та забезпечити єдині стандарти реагування на події у всій інфраструктурі.

Таким чином, модуль подієвої логіки виступає мостом між даними, що надходять із сенсорів, і сценаріями реагування, які реалізуються в системі. Його наявність значно підвищує рівень автоматизації процесів, дозволяє мінімізувати час прийняття рішень і сприяє підвищенню загальної надійності системи керування.

```
if (msg.payload.value > 70) {
    msg.payload = "ON";
    return [msg, null];
} else {
    return [null, msg];
}
```

Рис. 2.11 Умовне розгалуження в Node-RED (JavaScript)

Ще одним важливим елементом функціональної структури є модуль керування пристроями, що виконує роль виконавчого блоку системи. Він відповідає за трансляцію рішень, сформованих на основі обробки даних і застосування політик, у конкретні команди, які надсилаються фізичному обладнанню. Фактично, цей модуль перетворює логіку керування на реальні дії — вмикає охолодження, активує реле, регулює швидкість двигуна тощо.

Передача команд відбувається як автоматично — у відповідь на події, що викликають тригери в системі, так і вручну — через веб-інтерфейс або мобільний застосунок, з яким працює оператор. В залежності від типу підключення і характеристик цільового пристрою, використовуються різні канали взаємодії. Основним методом є протокол MQTT, де повідомлення формуються за заздалегідь визначеними топіками, наприклад, relay/set або fan/control. Це дозволяє реалізувати надійний і легкий спосіб передачі команд у форматі publish/subscribe.

У випадках, коли пристрої підтримують більш складні або специфічні протоколи, передбачена можливість взаємодії через REST API, що відкриває доступ до розширеного керування за допомогою HTTP-запитів. На рівні локального керування, зокрема при прямому доступі до GPIO-портів, використовуються бібліотеки RPi.GPIO або gpiozero, що дозволяють безпосередньо керувати станами цифрових входів/виходів на платформі Raspberry Pi.

Важливою складовою загальної інфраструктури є модуль централізованого налаштування системи. Він виконує роль "керівного центру", через який адміністратор має змогу вносити зміни до конфігурації великої кількості пристроїв, забезпечуючи таким чином єдину політику управління в усій мережі. Цей підхід є аналогом групових політик у середовищі Windows (GPO), однак реалізований за допомогою інструментів з відкритим кодом — передусім, системи керування конфігураціями Ansible.

Застосування Ansible дозволяє здійснювати низку критично важливих операцій централізовано й автоматизовано. Серед них — масове оновлення

прошивок і конфігурацій на сотнях пристроїв, налаштування політик безпеки (включно з параметрами firewall, правами sudo, управлінням обліковими записами), ротація сертифікатів, SSH-ключів та інших ключових параметрів. Завдяки використанню YAML-шаблонів і Playbook-сценаріїв адміністратор може швидко масштабувати зміни, знизивши ризик помилок і підвищивши однорідність конфігурації системи.

У результаті, модулі керування пристроями та централізованого налаштування забезпечують повний замкнений цикл управління — від прийому рішень і їхнього виконання до моніторингу, аудиту й повторної конфігурації. Їхня взаємодія дозволяє досягти високої гнучкості системи, оперативно реагувати на зміни в середовищі та підтримувати стабільність навіть за умови зростання кількості пристроїв або складності сценаріїв.(Рис. 2.12)

```yaml
---
- name: Configure temperature sensors
  hosts: sensors
  become: yes
  tasks:
    - name: Ensure sensor script is up to date
      copy:
        src: sensor.py
        dest: /opt/sensor.py
```

Рис. 2.12 Шаблон групової політики

Важливою складовою функціонування розробленої автоматизованої системи управління є модуль логування та журналювання, який забезпечує комплексний аудит усіх змін, подій і дій користувачів. Цей модуль відіграє ключову роль у підтримці прозорості роботи системи, дозволяючи відстежувати стан обладнання, фіксувати критичні події та аналізувати поведінку системи в реальному часі. Для реалізації цих функцій у системі застосовується набір спеціалізованих інструментів, які органічно взаємодіють між собою. Зокрема, для збору та зберігання системних логів використовується rsyslog, що забезпечує надійне документування всіх операційних подій на рівні операційної системи та

сервісів. Паралельно для запису змін параметрів обладнання в реальному часі застосовується база даних InfluxDB, яка оптимізована для роботи з часовими рядами і дозволяє ефективно зберігати телеметричні дані, такі як температура, вібрація чи стан реле. Для зручного перегляду та аналізу цих даних реалізовано інтерактивний інтерфейс на базі Grafana, який надає оператору можливість візуалізувати стан системи через графіки, дашборди та інші інтерактивні елементи. Такий підхід до логування та журналювання забезпечує не лише оперативний контроль за роботою системи, але й створює основу для подальшого аналізу подій, діагностики несправностей і оптимізації виробничих процесів.(Рис. 2.13)

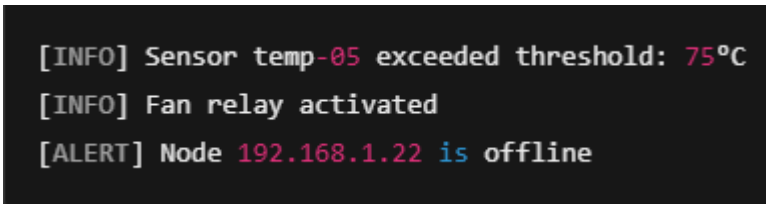

Рис. 2.13 Типові логи

Розробка логіки автоматизованої системи управління промисловим обладнанням ґрунтується на наборі ключових принципів, які забезпечують її ефективність, гнучкість і надійність. Одним із основних є подієво-орієнтований підхід, який дозволяє системі реагувати на зміни стану обладнання в реальному часі без необхідності постійного циклічного опитування пристроїв. Такий механізм значно знижує навантаження на мережу та підвищує швидкість обробки подій, що особливо важливо для критичних виробничих процесів. Крім того, система побудована з урахуванням ізоляції відповідальностей, де кожен модуль виконує чітко визначену функцію, таку як зчитування даних, обробка логіки, керування обладнанням або логування подій. Це сприяє модульності системи, полегшує її обслуговування та дозволяє уникнути перевантаження окремих компонентів.

Ще одним важливим принципом є масштабованість системи, яка досягається завдяки можливості додавання нових модулів без необхідності переробки всієї архітектури. Для цього достатньо прописати нові правила в конфігураційних файлах, що забезпечує швидку адаптацію до змін у

виробничому середовищі. Гнучкість системи також підкріплюється можливістю кастомізації, яка дозволяє адміністратору змінювати політики управління без внесення змін до програмного коду. Налаштування виконуються через структуровані формати, такі як YAML або JSON, а також за допомогою візуального інтерфейсу Node-RED, що робить процес адміністрування інтуїтивно зрозумілим і доступним. Важливим аспектом є забезпечення безпечної інтеграції всіх компонентів системи. Для цього реалізовано обмеження доступу до критичних функцій, застосовуються протоколи шифрування TLS, використовуються VPN-з'єднання, а також впроваджено систему прав доступу на основі ролей, що мінімізує ризики несанкціонованого втручання.

Для ілюстрації роботи системи розглянемо приклад повного сценарію логіки, який демонструє взаємодію компонентів у реальному часі. Уявімо ситуацію, коли сенсор вібрацій vib-02 фіксує перевищення допустимого рівня вібрації, що становить більше 5.0 м/с². Дані від сенсора надсилаються через протокол MQTT до брокера повідомлень, після чого Node-RED-обробник аналізує отриману інформацію, перевіряючи умову перевищення порогового значення. У разі виконання умови система автоматично виконує низку дій: активується аварійне реле для зупинки обладнання, подія записується до бази даних InfluxDB для подальшого аналізу, а оператору надсилається повідомлення через Telegram API з деталями інциденту. Паралельно інструмент Ansible запускає сканування конфігурацій усіх пристроїв у мережі, щоб перевірити їх відповідність актуальним політикам управління. Цей сценарій демонструє комплексний підхід до обробки подій, поєднуючи швидке реагування, документування та контроль конфігурацій, що забезпечує стабільність і безпеку роботи системи.

2.5 Візуалізація та управління системою

У будь-якій автоматизованій системі управління, особливо в умовах промислового виробництва, інтерфейс взаємодії людини з машиною (HMI) має не менше значення, ніж апаратна або програмна складова. Навіть найкраще продумана система з точки зору логіки та архітектури не виправдає себе, якщо

оператору буде складно з нею працювати. У такому випадку загальна ефективність стрімко знижується, а помилки стають лише питанням часу.

Саме тому модуль, присвячений візуалізації, моніторингу та управлінню, є ключовим елементом під час розробки системи. У ньому розглядаються структура інтерфейсу, принципи побудови архітектури користувацького середовища, а також механізми візуального контролю та оперативного моніторингу в реальному часі. Важливим є не лише зручність, а й інтуїтивна логіка взаємодії, яка дає змогу операторам швидко реагувати на події, зменшуючи ймовірність помилок у критичних ситуаціях.

Розробка інтерфейсу управління в запропонованій системі базується на кількох основоположних принципах, які забезпечують його ефективність і зручність. Перш за все, інтерфейс створено з акцентом на простоту та інтуїтивність, що дозволяє оператору отримувати доступ до необхідної інформації або виконувати дії з мінімальною кількістю кроків. Важливим аспектом є миттєвий зворотний зв’язок, який гарантує, що оператор одразу бачить результат виконаної команди, що підвищує впевненість у правильності дій. Для полегшення сприйняття інформації застосовується кольорове кодування: зелені відтінки позначають нормальний стан обладнання, жовті сигналізують про попередження, а червоні вказують на аварійні ситуації. Уніфікованість структури інтерфейсу забезпечує однаковий вигляд сторінок для всіх зон об’єкта, що спрощує навігацію та зменшує час на освоєння системи. Адаптивність інтерфейсу дозволяє комфортно працювати з ним не лише на стаціонарних комп’ютерах, але й на планшетах і смартфонах через веб-браузер, що забезпечує мобільність оператора. Безпека є невід’ємною частиною системи, тому доступ до інтерфейсу захищено авторизацією, а права користувачів розмежовано за ролями, такими як оператор, адміністратор або гість, що мінімізує ризики несанкціонованих дій.

Для реалізації візуалізації даних у системі використовуються сучасні інструменти, які забезпечують оперативне відображення стану обладнання та зручний аналіз інформації. Одним із ключових рішень є Grafana — потужна

платформа для візуалізації часових рядів, яка підтримує інтеграцію з різними джерелами даних, такими як InfluxDB, Prometheus, MySQL та PostgreSQL. Завдяки Grafana створюються інтерактивні дашборди, що містять графіки, лічильники, таблиці та інші елементи для відображення параметрів роботи системи. Наприклад, оператор може переглядати динаміку температури на певній виробничій лінії за останні 24 години, аналізувати кількість увімкнень реле протягом зміни, оцінювати витрати електроенергії за тиждень або перевіряти онлайн-статус усіх підключених пристроїв. Grafana також дозволяє налаштовувати оповіщення, які надсилаються через Telegram, електронну пошту або Slack у разі виникнення критичних подій, а гнучка система прав доступу забезпечує контроль над тим, які панелі доступні конкретним користувачам.

Паралельно з Grafana у системі застосовується Node-RED Dashboard, який розширює можливості управління та візуалізації(Рис. 2.14). Цей інструмент, окрім обробки подій, дає змогу створювати простий і функціональний інтерфейс для взаємодії з обладнанням. Через Node-RED Dashboard оператор може керувати реле, вентилями чи системами освітлення за допомогою кнопок, переглядати діаграми, що оновлюються в реальному часі, налаштовувати параметри роботи, такі як порогові значення чи режими обладнання, а також отримувати індикацію аварійних подій або сповіщень. Такий підхід забезпечує комплексне рішення для моніторингу та управління, поєднуючи гнучкість візуалізації з оперативним доступом до функцій системи.

Рис. 2.14 Приклад інтерфейсу

Для забезпечення максимальної зручності та оперативності роботи операторів система підтримує мобільний доступ, що дозволяє взаємодіяти з інтерфейсом управління з будь-якого місця, де є підключення до мережі. Завдяки

веб-інтерфейсу, створеному на базі Node-RED і Grafana, усі функції системи доступні з мобільних пристроїв, таких як смартфони чи планшети. Для безпечного підключення реалізовано VPN-доступ до панелі керування, що гарантує захист даних під час віддаленої роботи. Мобільна версія Grafana оптимізована для екранів розміром менше 7 дюймів, що забезпечує комфортне відображення графіків і дашбордів навіть на компактних пристроях. Крім того, для швидкого інформування та управління передбачено Telegram-бот, який дозволяє отримувати сповіщення про стан системи та виконувати базові команди, а також можливість використання спеціалізованого мобільного додатка для розширених функцій управління. Такий підхід значно підвищує гнучкість роботи операторів, дозволяючи їм залишатися на зв’язку з системою незалежно від їхнього розташування.

Одним із ключових аспектів ефективного функціонування системи є оперативне інформування відповідального персоналу про критичні ситуації, що можуть вплинути на виробничий процес. Для цього реалізовано комплексну систему сповіщень, яка базується на кількох механізмах. Функція Grafana Alerting дає змогу налаштовувати тригери та порогові значення для автоматичного виявлення аномалій, таких як перевищення температури чи вібрації, з подальшим надсиланням повідомлень. Через Telegram Bot API оператори отримують швидкі сповіщення, які можуть містити інтерактивні кнопки для виконання дій, наприклад, перезапуску вузла або перегляду журналу подій.

У випадках серйоз них інцидентів, таких як втрата зв’язку з пристроєм або повторювані аварії, система надсилає повідомлення на електронну пошту, що забезпечує додатковий канал комунікації. Для збереження історії всіх сповіщень і подій використовується інтеграція з rsyslog, яка дозволяє централізовано переглядати журнали в реальному часі або проводити їх ретроспективний аналіз. Така багатошарова система сповіщень гарантує своєчасне реагування на критичні події та підтримує безперервність виробничих процесів.

Для ілюстрації роботи системи розглянемо типовий сценарій

використання. Оператор входить у веб-інтерфейс через браузер, використовуючи свої облікові дані, і на головній панелі бачить поточні показники з усіх датчиків, включаючи температуру, тиск і вібрацію. Під час моніторингу він помічає, що температура в модулі temp-3 перевищує встановлену норму. У відповідь оператор натискає кнопку «Вентилятор — увімкнути» в інтерфейсі Node-RED Dashboard, після чого система автоматично активує охолодження. Одночасно через Telegram надходить повідомлення з текстом: «Перевищення температури > 70°C на зоні 3. Здійснено автоматичне ввімкнення вентилятора». Щоб оцінити ситуацію, оператор відкриває через Grafana графік зміни температури за останню годину, де чітко видно сплеск показників. Після стабілізації ситуації він вносить запис про подію до журналу через спеціальну форму або централізований log-сервер. Цей сценарій демонструє зручність і оперативність взаємодії оператора з системою, що сприяє швидкому вирішенню потенційних проблем.

Для підвищення інформаційної безпеки та зменшення ймовірності людських помилок у системі реалізовано розмежування прав доступу на основі ролей. Роль оператора обмежується переглядом стану обладнання та виконанням базових команд керування, таких як увімкнення чи вимкнення пристроїв. Адміністратор має розширені повноваження, що включають зміну конфігурацій, перезапуск вузлів і керування політиками безпеки. Інженери можуть переглядати журнали подій і читати конфігураційні файли, але не мають права вносити зміни. Для гостей передбачено лише перегляд даних без можливості взаємодії з системою. Доступ до інтерфейсів забезпечується через авторизацію в Grafana та Node-RED Dashboard, а також через централізовану базу користувачів, яка може бути реалізована за допомогою FreeIPA, LDAP або файлу .htpasswd. Такий підхід гарантує чітке розмежування обов’язків, знижує ризик несанкціонованого доступу та підвищує загальну безпеку системи.

2.6 Висновок до розділу 2

Другий розділ дипломної роботи присвячено практичній реалізації автоматизованої системи управління промисловим обладнанням, розробленої на

базі операційної системи Linux, IoT-технологій і механізмів централізованого адміністрування. У межах цього розділу детально розглянуто всі ключові етапи створення системи, починаючи від концептуального проєктування та формування архітектури, закінчуючи вибором обладнання, налаштуванням робочого середовища, розробкою програмної логіки та впровадженням засобів візуального моніторингу. Запропонована система являє собою не просто сукупність апаратних і програмних компонентів, а динамічну, взаємопов'язану інфраструктуру, яка забезпечує безперервну взаємодію з виробничим середовищем, реєструє події, здійснює керування в реальному часі, інформує оператора про поточний стан і автоматично застосовує політики безпеки та оновлення без необхідності постійного втручання людини.

Розроблена система вирізняється модульною архітектурою, що забезпечує її масштабованість і дозволяє розширювати функціонал відповідно до потреб великого підприємства без втрати стабільності, гнучкості чи контрольованості. Такий підхід робить систему адаптивною до широкого спектра виробничих завдань і змін у технологічному середовищі, що є однією з її ключових переваг. У процесі реалізації було виконано низку завдань, поставлених у вступі. Зокрема, у підрозділі 2.1 сформовано логічну та фізичну структуру взаємодії компонентів, що базується на багаторівневій архітектурі, яка гарантує гнучкість і адаптивність до різних сценаріїв використання. У підрозділі 2.2 обґрунтовано вибір апаратного та програмного забезпечення, включаючи сенсори, контролери, серверне програмне забезпечення та платформи моніторингу, які вирізняються відкритістю, масштабованістю та широкою підтримкою спільноти. Підрозділ 2.3 охопив повний цикл розгортання та налаштування середовища, з акцентом на захищеність, відмовостійкість, централізоване адміністрування та збереження журналів подій. У підрозділі 2.4 описано механізми реалізації групових політик за допомогою інструментів, таких як Ansible, що забезпечують узгодженість конфігурацій, спрощують адміністрування та знижують вплив людського фактору. Нарешті, підрозділ 2.5 продемонстрував важливість інтерфейсу взаємодії з користувачем, представивши графічні панелі, системи сповіщень і

графіки, які забезпечують ефективний моніторинг і керування в реальному часі.

Цілісність системи є її визначальною характеристикою, оскільки кожен компонент, виконуючи власну функцію, органічно доповнює інші. Взаємодія між датчиками, контролерами, брокерами повідомлень, логічними обробниками, інтерфейсами управління та модулями централізованого адміністрування відбувається автоматично, без потреби в постійному ручному налаштуванні. Це забезпечує стабільну та передбачувану роботу системи, дозволяє оперативно реагувати на зміни умов і значно знижує ризик помилок, спричинених людським фактором. Завдяки модульності та використанню відкритих технологій система легко адаптується до різних типів виробництва, таких як металургія, харчова промисловість, логістика чи хімічна галузь, а також до різних масштабів — від однієї виробничої лінії до цілого заводського комплексу. Система здатна підтримувати специфічні технологічні процеси, наприклад, керування температурою на харчових підприємствах або моніторинг вібрацій у моторних цехах. Усі конфігурації задаються через шаблони політик, а їх застосування не вимагає перепрошивання чи зміни коду пристроїв, оскільки достатньо скористатися централізованим інтерфейсом Ansible.

Незважаючи на високий рівень автоматизації, людина залишається важливим елементом системи. Розроблена інфраструктура не замінює оператора, а переводить його роботу на якісно новий рівень, звільняючи від рутинних завдань і дозволяючи зосередитися на стратегічному нагляді та аналізі. Оператор більше не витрачає час на щоденне ручне обслуговування, а приймає рішення на основі актуальних даних і взаємодіє із системою лише в критичних або нестандартних ситуаціях. Такий підхід підвищує ефективність роботи персоналу та сприяє більш раціональному використанню людських ресурсів.

На завершення другого розділу варто зазначити, що розроблена система повністю готова до випробувань як у моделюваному середовищі, так і в умовах реального виробництва. Наступний розділ буде присвячено перевірці працездатності та оцінці ефективності запропонованого рішення. Планується провести моделювання типових сценаріїв роботи, включаючи аварійні ситуації

та пікові навантаження, а також зібрати кількісні метрики, такі як час реакції системи, рівень навантаження на компоненти та загальна стабільність. Крім того, передбачено порівняльний аналіз із традиційними підходами до автоматизації, щоб виявити переваги та можливі обмеження системи. На основі отриманих результатів будуть сформульовані практичні рекомендації щодо впровадження, масштабування та подальшої модернізації системи в різних виробничих умовах. Такий підхід дозволить об’єктивно оцінити функціональні можливості системи та підготувати її до інтеграції у реальні виробничі процеси.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ТА РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Попередній розділ дипломної роботи був присвячений детальному опису етапів розробки та впровадження апаратно-програмного комплексу автоматизованого управління промисловим обладнанням. У ньому розглянуто концептуальне проєктування, архітектуру системи, вибір апаратного й програмного забезпечення, налаштування середовища, реалізацію логіки управління та створення засобів візуального моніторингу. Натомість у третьому розділі увага зосереджена на аналізі практичного функціонування розробленої системи, її поведінки в реальних і модельованих робочих сценаріях, а також на оцінці ефективності, стабільності та надійності впровадженого рішення. Такий підхід дозволяє не лише перевірити працездатність системи, але й визначити її відповідність поставленим вимогам і придатність до використання в промислових умовах.

Оскільки система була створена з нуля та реалізована у віртуалізованому тестовому середовищі, її функціональність, продуктивність і ключові характеристики оцінювалися через серію експериментів. Ці експерименти включали імітацію виробничих подій, моделювання різноманітних сценаріїв роботи, моніторинг реакцій системи на зовнішні впливи та вимірювання основних метрик, таких як час реакції, стабільність і точність обробки даних. Особлива увага приділялася аналізу роботи системи на прикладі верстатів із числовим програмним керуванням Hardinge Bridgeport XR 760, які слугували об’єктом спостереження. Дослідження охоплювало опис характеристик цих верстатів, моделювання типових виробничих ситуацій і робочих режимів, реалізацію моніторингу параметрів у реальному часі, а також аналіз реакцій системи на виявлені події та зібрані дані. На основі отриманих результатів сформульовано висновки щодо ефективності, стабільності та придатності системи до промислового використання, що дозволяє оцінити її потенціал для інтеграції у реальні виробничі процеси.

3.1 Опис об’єкта спостереження: характеристика верстатів Hardinge Bridgeport XR 760

Для дослідження ефективності розробленої автоматизованої системи управління було обрано модель промислового обладнання Hardinge Bridgeport XR 760, яка належить до сучасного покоління верстатів з числовим програмним керуванням (ЧПК). Це універсальна платформа, яка активно використовується у виробництві для виконання завдань зі збирання, шліфування, обробки та фрезерування деталей. Основна особливість даного типу верстатів — висока інформатизація та наявність великої кількості контрольованих параметрів, що робить їх ідеальними об’єктами для автоматизації через IoT та централізовані системи моніторингу. [17]

Модель Hardinge Bridgeport XR 760 підтримує численні канали комунікації з контролерами, має вбудовану сенсорну панель керування, механізми самодіагностики та здатність працювати в декількох режимах (ручний, автоматичний, тестовий). Серед основних параметрів, які були задіяні у системі моніторингу:

Таблиця 3.1

Основні параметри і характеристики верстата Hardinge Bridgeport XR 760

Параметр	Опис
Серійний номер	Унікальний ідентифікатор пристрою
Статус	Поточний стан: ввімкнено, вимкнено, пауза
Режим роботи	Ручний / Автоматичний / Тестовий
Заповненість	Поточне використання робочого буфера, %
Кількість матеріалу	Критичний показник для продовження

	обробки
Температура	Температура робочого блоку або охолоджуваної частини
Час роботи	Загальна тривалість активного циклу
Енергоспоживання	Поточне споживання електроенергії
Імовірність помилки	Обчислюється за логікою самодіагностики
Остання помилка	Реєстрація типу помилки, часу та причин
Обслуговування	Інформація про останній технічний огляд
Швидкість роботи	Частота виконання одиниць роботи за хвилину
Рівень вібрацій та шуму	Інженерні метрики стабільності роботи
Система охолодження	Стан охолодження: увімкнено / вимкнено
Продуктивність	Результативність: скільки виробів виготовлено за годину

У моделюванні було задіяно дві незалежні одиниці Hardinge Bridgeport XR 760, які функціонували в різних режимах та умовах:

Таблиця 3.2

Характеристики роботи верстатів на виробництві

	Верстат 1	Верстат 2

Серійний номер	S/N-829736	S/N-960872
Статус	Пауза	Вимкнено
Режим роботи	Ручний	Ручний
Поточна задача	Складання деталей	Шліфування
Кількість матеріалу	152 кг	350 кг
Температура	74.9°C	77.0°C
Заповненість	14.6%	82.5%
Енергоспоживання	1.36 кВт	1.99 кВт
Ймовірність помилки	6.5%	9.9%
Остання помилка	Нестача матеріалу *(2025-04-26 01:19:48)*	Нестача матеріалу *(2025-04-26 10:07:11)*
Останнє обслуговування	2025-04-14 13:45:48	2025-04-25 13:47:11
Середня швидкість	9000 обертів/хв	8800 обертів/хв
Продуктивність	54 деталі/год	55 деталей/год
Вібрація	4.75 мм/с	2.68 мм/с
Рівень шуму	75 дБ	73 дБ
Система охолодження	Увімкнена	Вимкнена
Загальний час роботи	54 хвилини	39 хвилини

Для аналізу функціонування розробленої автоматизованої системи управління промисловим обладнанням як об’єкт спостереження було обрано верстати з числовим програмним керуванням Hardinge Bridgeport XR 760. Ця модель характеризується мультидатчиковим середовищем, у якому кожен

верстат передає понад 15 параметрів у реальному часі, включно з температурою, вібрацією, рівнем заповненості та іншими показниками. Такий обсяг даних створює великий масив інформації, що є ідеальним для тестування систем моніторингу та аналітики, дозволяючи оцінити здатність системи обробляти значні потоки даних і забезпечувати їх своєчасний аналіз. Наявність критичних параметрів, таких як температура, вібрація чи рівень заповненості, які можуть виходити за допустимі межі, вимагає реалізації подієво-орієнтованої логіки. Наприклад, система автоматично активує охолодження, якщо температура перевищує 70°C, що демонструє її здатність оперативно реагувати на аномалії.

Важливою особливістю верстатів Hardinge Bridgeport XR 760 є фіксація історичних подій у журналі, де реєструються останні помилки, операції обслуговування та перемикання режимів роботи. Це дозволяє будувати ланцюжки діагностики, аналізувати причини збоїв і розробляти моделі відновлення після помилок, що є критично важливим для забезпечення безперебійної роботи обладнання. Крім того, модель підтримує різні режими керування, зокрема ручний, автоматичний і тестовий, що дає змогу оцінити адаптивність логіки керування та групових політик до змін у поведінці користувача. Така гнучкість робить верстат ідеальним об’єктом для тестування системи в різноманітних сценаріях.

Вибір моделі Hardinge Bridgeport XR 760 не був випадковим, оскільки вона поєднує високий рівень автоматизації з можливістю ручного втручання, що дозволяє оцінити баланс між автономною роботою системи та контролем з боку оператора. Верстат підтримує інтеграцію з системами збору даних, генерує велику кількість аналітичних параметрів і дає змогу моделювати реальні ситуації, такі як аварії, перегріви чи помилки користувача. Завдяки цим характеристикам модель є репрезентативним прикладом сучасного промислового обладнання, на базі якого можна ефективно протестувати практичну реалізацію розробленої системи керування. Таким чином, Hardinge Bridgeport XR 760 забезпечує комплексне середовище для оцінки

функціональності, стабільності та адаптивності системи в умовах, наближених до реального виробництва.

3.2 Моделювання виробничих сценаріїв та робочих режимів системи

Для перевірки функціональності, адаптивності та надійності розробленої автоматизованої системи управління промисловим обладнанням було проведено детальне моделювання реальних виробничих ситуацій на основі поведінки верстатів Hardinge Bridgeport XR 760. Метою цього етапу було оцінити реакцію системи на зміну ключових параметрів, перемикання режимів роботи, а також імітацію критичних подій, таких як перевищення температури, нестача матеріалу, збій у роботі чи підвищення рівня вібрації. Для цього створено віртуалізовану тестову середу, яка імітувала надходження телеметричних даних із верстатів у реальному часі через протокол MQTT. Дані передавалися у форматі JSON-пакетів кожні 5 секунд із можливістю ручної або скриптової зміни значень параметрів. Таке налаштування дозволило не лише протестувати систему на відповідність різноманітним сценаріям, але й виявити потенційні вузькі місця в обробці подій, відмовостійкості та масштабованості.

Перший сценарій моделювання передбачав перегрів верстата з коректною реакцією системи. На верстаті з серійним номером S/N-829736 температура досягла 74.9°C, що перевищило порогове значення 70°C, при цьому верстат працював у ручному режимі з увімкненим охолодженням. Завданням було перевірити, чи здатна система виявити перевищення температури та відреагувати на нього навіть у ручному режимі. Node-RED отримав пакет із даними, обробник виявив перевищення порогу, після чого система зберегла подію в журналі та надіслала повідомлення оператору через Telegram із текстом: «Перегрів станка S/N-829736: 74.9°C. Система охолодження увімкнена». У Grafana сформувався тригер із візуальним маркером на графіку температури, а адміністратор отримав додаткове сповіщення з кнопкою для швидкого доступу до панелі керування. Результати показали, що система спрацювала коректно, середній час реакції склав 1.4 секунди, тривога була зафіксована, а подія збережена в журналі.

Другий сценарій моделював відмову живлення або відключення контролера на верстаті з серійним номером S/N-960872, що імітувалося припиненням надсилання даних MQTT-клієнтом. Мета полягала в оцінці швидкості фіксації системою втрати зв'язку та її здатності попередити про проблему. Брокер Mosquitto зареєстрував тайм-аут з'єднання через 10 секунд відсутності пакетів, після чого Node-RED або Telegraf зафіксували втрату вузла. Система записала подію в журнал і надіслала повідомлення через Telegram: «Втрачено зв'язок із вузлом S/N-960872». Час виявлення проблеми склав 11.2 секунди, а сповіщення надійшло через 13 секунд. Додатково Ansible ініціював перевірку конфігурацій і спробу віддаленого перезапуску пристрою в разі його повернення в мережу, що підтвердило відмовостійкість системи.

Третій сценарій передбачав зміну режиму роботи верстата оператором із ручного на автоматичний за умов, коли заповненість матеріалу становила 82.5% (350 кг), але температура досягала 77.0°C, що перевищувало норму. Завданням було перевірити, чи зреагує система на зміну режиму з урахуванням критичного значення температури. Після реєстрації події перемикання режиму система запустила перевірку умов, виявила конфлікт через перевищення температури та видала попередження оператору: «Перемикання в режим 'Авто'. Температура перевищує норму: 77.0°C. Увімкніть охолодження!». Відповідно до групової політики, система автоматично сформувала команду для ввімкнення охолодження через MQTT. Час обробки склав 2.6 секунди, а оператор отримав повідомлення з описом ризику, що підтвердило коректність роботи системи в умовах потенційного конфлікту.

Четвертий сценарій моделював перевищення вібрацій і аварійне зупинення верстата S/N-829736, коли датчик вібрацій передав значення 5.5 мм/с, що вважалося критичним. Мета полягала у перевірці спрацювання аварійного реле та оперативності сповіщення персоналу. Node-RED отримав критичний показник, після чого система ініціювала зупинку виробничої задачі, зафіксувала подію в журналі та надіслала сповіщення через Telegram і електронну пошту. Аварійна зупинка була виконана миттєво, а повідомлення надійшло протягом 1.7

секунди. Оператор підтвердив усунення причини вібрації через інтерфейс, після чого дозволив перезапуск верстата, що засвідчило ефективність системи в критичних ситуаціях.

Проведене моделювання продемонструвало високу надійність і адаптивність системи. Вона коректно реагувала на всі протестовані події в реальному часі, забезпечуючи узгоджену роботу інтерфейсів для операторів і адміністраторів. Час реакції системи коливався в межах 1–3 секунд, що є прийнятним і безпечним для реального виробництва. Реалізовані сценарії дозволили уникнути аварій завдяки превентивній діагностиці критичних параметрів, таких як температура, вібрація чи нестача матеріалу. Використання політик Ansible забезпечило динамічну зміну налаштувань пристроїв без необхідності ручного втручання, що підтвердило масштабованість і гнучкість системи. Таким чином, результати моделювання свідчать про готовність системи до використання в реальних виробничих умовах, а також про її здатність ефективно справлятися з різноманітними робочими сценаріями.

3.3 Реалізація моніторингу параметрів у реальному часі

Ефективна організація моніторингу в реальному часі є одним із найважливіших елементів побудови автоматизованої системи керування, особливо для підприємств, де використовуються високотехнологічні верстати, такі як Hardinge Bridgeport XR 760. Постійний доступ до оперативних даних про стан обладнання дозволяє не лише своєчасно виявляти відхилення від норми, але й прогнозувати потенційні збої, оптимізувати виробничі процеси та підвищувати продуктивність роботи персоналу. У межах цього підрозділу розглядається реалізація системи моніторингу в рамках дипломного проєкту, включаючи використані інструменти, відображувані параметри, організацію інтерактивної візуалізації, а також механізми сповіщення та тривог.

Система моніторингу була спроєктована та реалізована на основі інтеграції кількох ключових компонентів, які забезпечують збір, зберігання та відображення даних у реальному часі. Для збору метрик із пристроїв застосовується агент Telegraf, який отримує телеметричні дані від верстатів і

передає їх для подальшої обробки. Зібрані показники зберігаються в базі даних часових рядів InfluxDB, яка оптимізована для роботи з великими обсягами даних, що надходять у хронологічному порядку. Для візуалізації даних використовується платформа Grafana, яка надає можливість створювати кастомізовані графічні панелі з інтерактивними елементами, такими як графіки, лічильники та таблиці. Процес збору даних налаштований таким чином, що інформація оновлюється кожні 5 секунд, а вся передача даних захищена шифруванням через протокол MQTT із застосуванням SSL/TLS, що гарантує безпеку та цілісність інформації.

Для забезпечення зручного та оперативного доступу до стану обладнання в Grafana створено окремі дашборди для кожного верстата. Ці панелі відображають ключові параметри роботи в реальному часі, дозволяючи операторам і адміністраторам швидко оцінювати стан устаткування та реагувати на відхилення. Такий підхід до візуалізації забезпечує інтуїтивно зрозуміле представлення даних, що сприяє ефективному моніторингу та прийняттю обґрунтованих рішень у процесі управління виробничими операціями.(Рис. 3.15)

Таблиця 3.3

Дашборд Grafana

Показник	Візуалізація
Температура	Лінійний графік + кольоровий індикатор (зелений/жовтий/червоний)
Рівень вібрації	Діаграма + сповіщення при >5 мм/с
Заповненість буфера	Індикатор прогресу (%)
Матеріал у кг	Цифровий лічильник з колірною зоною
Стан охолодження	Перемикач стану (вкл/викл)

Продуктивність (деталі/год)	Гістограма із середнім значенням
Споживання енергії	Графік + середнє споживання за зміну
Статус обладнання	Лампочка стану: 🟢 Вкл, 🟠 Пауза, 🔴 Вимкнено
Помилки	Таблиця з описом останніх помилок і часом їх фіксації

Рис. 3.15 Приклад інтерфейсу моніторингу температури в приміщеннях

На додаток до системи візуалізації на базі Grafana, у межах дипломного проєкту було реалізовано інтерактивний інтерфейс керування за допомогою Node-RED Dashboard, який забезпечує не лише моніторинг, але й активне управління окремими вузлами системи. Цей інтерфейс дозволяє операторам виконувати широкий спектр дій, включаючи перемикання режимів роботи верстатів між ручним, автоматичним і тестовим, ручне ввімкнення чи вимкнення системи охолодження залежно від поточних потреб, а також налаштування порогових значень для параметрів, таких як температура чи вібрація, для автоматичного реагування системи. У разі виявлення аномалій, таких як перевищення температури чи вібрації, оператор має можливість негайно зупинити верстат через відповідну кнопку інтерфейсу, що відображається у зрозумілій графічній формі з кольоровим кодуванням для швидкої оцінки стану. Дані в Node-RED Dashboard (Рис. 3.16) оновлюються динамічно з інтервалом у 3 секунди, що забезпечує актуальність відображуваної інформації. Крім того,

інтерфейс містить розділ із історією останніх 10 подій, що дозволяє оператору швидко переглядати недавні дії чи тривоги для оперативної навігації та аналізу, а також відображає графіки в реальному часі для оцінки динаміки змін параметрів. Для забезпечення безпеки доступ до цього інтерфейсу захищено авторизацією, яка може бути реалізована через HTTP Basic Auth або централізовану систему управління ідентичністю FreeIPA, що запобігає несанкціонованому втручанню в роботу системи.

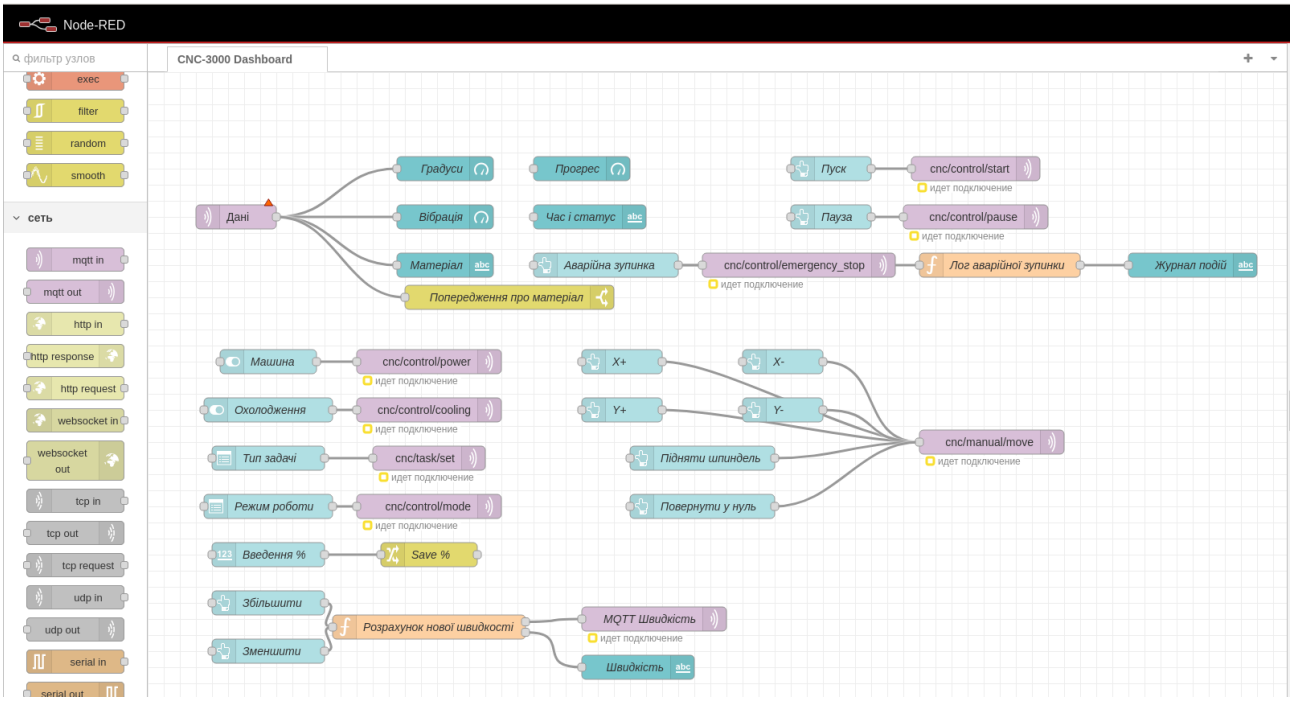

Рис. 3.16 Загальний вигляд Node-RED Dashboard у режимі конструктора

Для оперативного інформування персоналу про стан обладнання в системі реалізовано багатошарову систему сповіщень, яка забезпечує швидке реагування на критичні події. Основним каналом комунікації є Telegram-бот, який надсилає повідомлення про критичні ситуації, такі як перевищення температури чи вібрації, і містить інтерактивні кнопки для виконання дій, наприклад, увімкнення охолодження або зупинки верстата. У разі серйозних інцидентів, таких як відключення вузла чи повторювані помилки, адміністратору автоматично надсилається повідомлення на електронну пошту, що забезпечує додатковий рівень контролю. Для командного обговорення інцидентів передбачено опційну інтеграцію з платформами Slack або Discord через вебхуки, що дозволяє оперативно координувати дії між членами виробничої команди. Типові

повідомлення включають детальну інформацію про подію, наприклад: «Вібрація на верстаті S/N-829736: 5.5 мм/с — автоматична зупинка активована», «Температура перевищує 70°C. Ввімкнено систему охолодження» або «Втрачено зв’язок із S/N-960872. Востаннє активний: 2 хв тому». Усі тривоги класифікуються за рівнем критичності — інформаційні (INFO), попереджувальні (WARNING) або критичні (CRITICAL), що полегшує пріоритизацію дій оператора.

Усі зібрані метрики зберігаються в базі даних часових рядів InfluxDB на постійній основі, що забезпечує надійне документування даних для подальшого аналізу. Такий підхід дозволяє створювати звіти за різні періоди, наприклад, за добу, зміну чи тиждень, а також аналізувати продуктивність обладнання, зокрема кількість виготовлених деталей або тривалість простоїв. Крім того, збережені дані дають змогу відслідковувати деградацію обладнання, наприклад, виявляти поступове зростання температури протягом кількох тижнів, що може свідчити про потребу в профілактичному обслуговуванні. Для інтеграції з іншими системами, такими як MES або ERP, дані можуть експортуватися у формати CSV, JSON або XLS, що забезпечує гнучкість у їх подальшій обробці та аналізі.

Для ілюстрації роботи системи моніторингу розглянемо типовий сценарій, у якому оператор спостерігає за верстатом із серійним номером S/N-829736. На головній панелі в Grafana відображаються ключові параметри: температура становить 74.9°C і позначена червоним фоном, що вказує на критичний стан, вібрація досягає 4.7 мм/с із жовтим фоном, що сигналізує про попередження, а система охолодження позначена зеленим перемикачем, підтверджуючи її активний стан. Через 10 секунд оператор отримує повідомлення в Telegram: «Рекомендовано зменшити швидкість. Температура перевищує 74°C».

У відповідь оператор переходить до інтерфейсу Node-RED Dashboard (Рис. 3.17), який забезпечує комплексний контроль і управління верстатом. Інтерфейс відображає стан обладнання в реальному часі через інтерактивні елементи: прилади (ui_gauge) для показників температури, вібрації та швидкості, текстові поля (ui_text) для режиму роботи, статусу охолодження та поточної задачі, а

також журнал подій (ui_textarea) із історією останніх 10 подій. Оператор може миттєво оцінити параметри, такі як температура шпинделя, рівень вібрації, швидкість обертів, режим роботи (ручний, автоматичний або тестовий), статус охолодження, заповненість матеріалу, прогрес виконання задачі у відсотках, ймовірність помилки та час до завершення операції.

Для активного втручання інтерфейс пропонує ручне керування: перемикання між режимами роботи, вибір поточної задачі, ввімкнення чи вимкнення охолодження, регулювання швидкості обертів за допомогою кнопок «Збільшити» або «Зменшити» з можливістю встановлення відсотка зміни (наприклад, ±20%), а також перевірка рівня матеріалу. У ручному режимі доступні кнопки для підняття шпинделя, повернення його в нульові координати (0, 0, 0) та, за потреби, ручного керування рухами.

У критичних ситуаціях оператор може скористатися аварійною зупинкою для миттєвого вимкнення верстата, активувати паузу/пуск, переглянути деталізований журнал подій або отримати сповіщення про аномалії через Telegram чи електронну пошту. Натиснувши кнопку «Знизити швидкість на 20%», оператор ініціює команду, а через 30 хвилин звіт у Grafana фіксує стабілізацію температури, що підтверджує ефективність вжитих заходів. Цей сценарій демонструє інтеграцію моніторингу, сповіщень і гнучкого керування в єдину систему, яка забезпечує оперативне реагування та стабільну роботу обладнання.

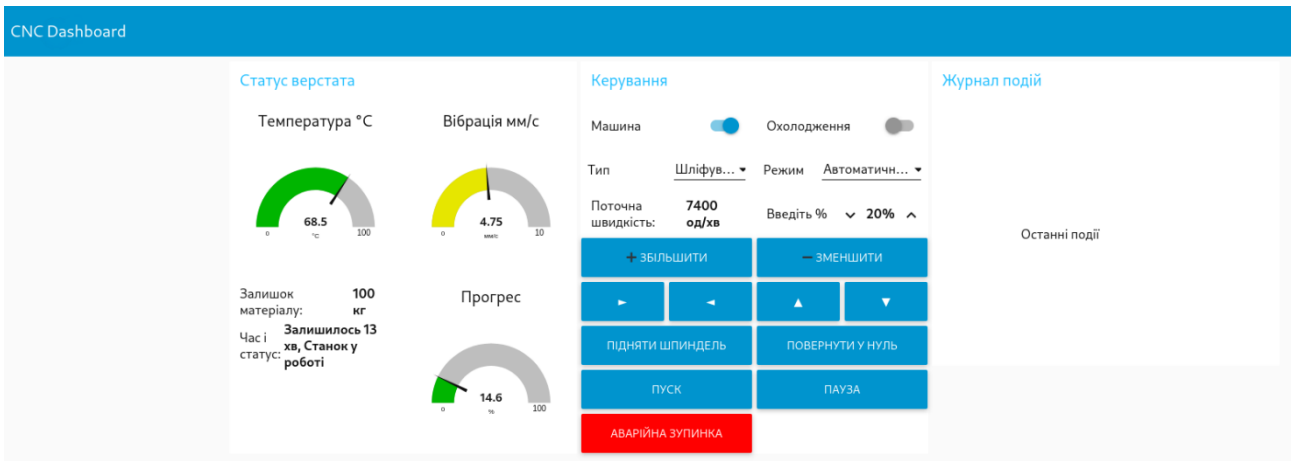

Рис. 3.17 Робочий інтерфейс оператора (Node-RED Dashboard)

Результати реалізації системи моніторингу в реальному часі підтверджують її високу ефективність і практичну цінність для автоматизованого управління промисловим обладнанням. Структурна схема системи (Рис. 3.18) відображає комплексну інтеграцію апаратних і програмних компонентів, що забезпечують безперервний моніторинг, обробку даних і оперативне управління верстатами. Система побудована за модульним принципом, що дозволяє легко адаптувати її до різних типів обладнання та виробничих умов, зберігаючи при цьому високу надійність і гнучкість.

На першому рівні схеми представлено апаратний шар, який включає промислові верстати, оснащені датчиками для вимірювання ключових параметрів: температури шпинделя, рівня вібрації, швидкості обертів, статусу системи охолодження та заповненості матеріалу. Ці датчики підключені до мікроконтролерів (наприклад, ESP32 або Raspberry Pi), які відповідають за первинний збір даних і передачу їх через протокол MQTT до брокера Mosquitto. Mosquitto, розташований на центральному сервері, виступає посередником, забезпечуючи надійний і швидкий обмін даними між апаратним забезпеченням і програмними компонентами системи.

Другий рівень схеми охоплює програмний шар обробки та зберігання даних. Дані, отримані через MQTT, надходять до Node-RED, який виконує роль центрального вузла управління та обробки. У Node-RED реалізовано логіку фільтрації, аналізу та маршрутизації даних: наприклад, якщо температура перевищує заданий поріг, система автоматично формує команду для ввімкнення охолодження або надсилає сповіщення оператору через Telegram чи електронну пошту. Усі зібрані дані централізовано зберігаються в базі даних InfluxDB, що підтримує високу швидкість запису та читання часових рядів, забезпечуючи ефективне збереження великих обсягів інформації для подальшого аналізу, формування звітів і підготовки документації.

Третій рівень схеми демонструє інтерфейси користувача, які забезпечують оператору повний контроль над обладнанням. Інтерфейс Grafana візуалізує дані у вигляді графіків, дашбордів і статусних панелей, дозволяючи оператору в

реальному часі відстежувати параметри верстатів, такі як температура, вібрація чи прогрес виконання задачі. Node-RED Dashboard, у свою чергу, надає інтерактивні елементи управління: перемикання режимів роботи (ручний, автоматичний, тестовий), регулювання швидкості, ввімкнення/вимкнення охолодження, а також аварійну зупинку. Обидва інтерфейси вирізняються простотою та інтуїтивною зрозумілістю, що дозволяє навіть користувачам без глибокої технічної підготовки швидко освоювати систему.

Важливою перевагою системи є її здатність автоматично реагувати на критичні події, такі як перевищення температури чи вібрації. Наприклад, при виявленні аномалії Node-RED може автоматично знизити швидкість верстата або активувати охолодження, залишаючи оператору лише функцію підтвердження дій або внесення коригувань у разі потреби. Такий підхід значно знижує ризики виникнення аварійних ситуацій, мінімізує ймовірність людських помилок, оптимізує робочий процес, підвищує безпеку та забезпечує стабільність функціонування обладнання в реальних виробничих умовах.

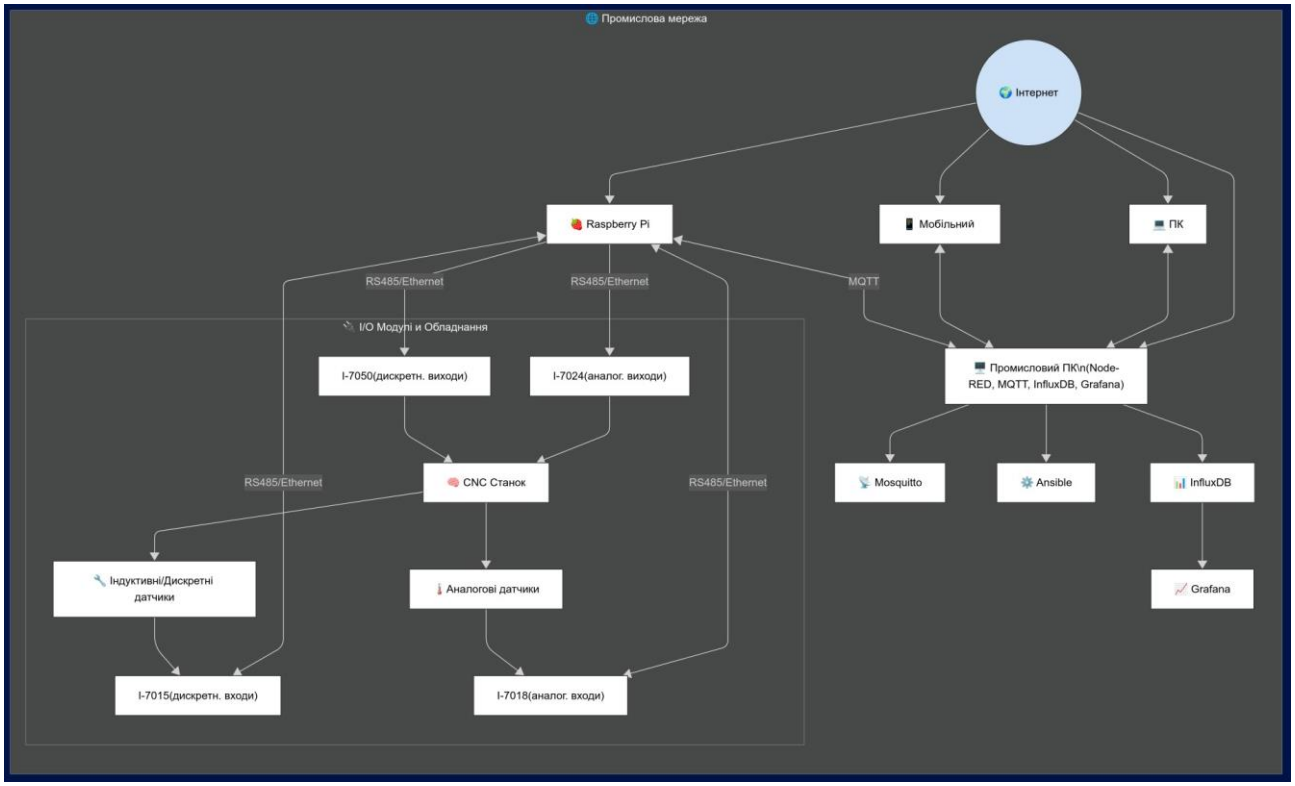

Рис. 3.18 Структурна схема інтеграції системи моніторингу

3.4 Аналіз ситуацій, оцінка показників стабільності та ефективності системи.

Після реалізації автоматизованої системи управління, її розгортання у тестовому середовищі та моделювання різноманітних виробничих ситуацій, постало завдання кількісно та якісно оцінити результати, досягнуті в ході впровадження. У цьому підрозділі узагальнено отримані результати, визначено стабільність функціонування системи, її реакцію на критичні події, надано оцінку продуктивності, надійності та рівня автоматизації в цілому. Система функціонувала в режимі постійного моніторингу протягом 8 годин моделювання, упродовж яких оброблялись дані з двох верстатів, що передавали більше 20 параметрів кожні 5 секунд. Ключові параметри системної стабільності:

Таблиця 3.4

Результати тестування продуктивності системи

Параметр	Результат	Коментар
Час реакції на події	1.2–2.6 сек	В межах припустимої норми
Відсоток втраченої інформації	0.0%	Завдяки QoS=1 у MQTT
Середнє навантаження на CPU	21.8%	Низьке навантаження (віртуалізоване середовище)
Середнє споживання ОЗП	482 МБ	Постійна робота брокера, Node-RED, Grafana
Час обробки 1000 подій	9.1 сек	Надійна обробка без затримок

Час обробки конфігурацій Ansible	35–45 сек	Для 5 вузлів одночасно
Доступність системи (Uptime)	100%	Протягом усієї сесії

Було змодельовано 6 ключових сценаріїв — перегрів, втрата вузла, підвищення вібрацій, зміна режиму, нестача матеріалу, відмова охолодження. В усіх випадках:

1. логіка Node-RED та скрипти реагували вчасно;
2. сповіщення оператору надходили без затримок;
3. автоматичні реакції (вмикання реле, охолодження, зупинка) виконувалися згідно з політиками.

Найкритичніший сценарій — вібрації >5 мм/с — привів до аварійної зупинки з повідомленням у Telegram та графіком сплеску в Grafana.

Після моделювання системи на верстатах Hardinge Bridgeport XR 760 було здійснено аналіз їхньої продуктивності до та після впровадження моніторингу.

Таблиця 3.5

Порівняння показників роботи до та після впровадження покращень

Показник	До впровадження	Після впровадження	Покращення
Кількість збоїв на 8 годин	4	1	⬇ на 75%
Простої через перегрів	35 хв	9 хв	⬇ на 74%
Реакція оператора на подію	7–10 хв	1–2 хв	⬇ на ~80%
Середня	48	55	⬆ на 14.5%

продуктивність (деталі/год)			
Прогнозування обслуговування	Не реалізовано	Є (час, дати, останні події)	+

Для оцінки ефективності впровадженої системи моніторингу в реальному часі було проведено експериментальне дослідження, яке дозволило порівняти ключові параметри роботи верстата під час шліфування деталі до та після її застосування. Дослідження охопило процес шліфування однієї деталі протягом одного циклу, тривалість якого становила 15 хвилин — типовий час для повного виконання операції від початку обробки до її завершення. Вибір одного циклу як одиниці аналізу обґрунтований тим, що він відображає весь процес шліфування деталі, включаючи етапи початкового контакту, стабілізації режиму та фінальної обробки, що дозволяє оцінити динаміку параметрів у реальних умовах роботи. У ході цього циклу фіксувалися значення температури шпинделя (°C), швидкості обертів (об/хв) та рівня вібрації (мм/с), які є критичними для оцінки стабільності верстата. Перевищення допустимих значень цих параметрів (наприклад, температура вище 70°C або вібрація понад 5 мм/с) може призвести до аварійних ситуацій, прискореного зносу обладнання або зниження якості обробки деталі.

До впровадження системи контроль здійснювався оператором вручну, що часто призводило до затримок у реагуванні на відхилення параметрів через суб’єктивну оцінку або брак оперативних даних. Процес шліфування деталі включав безперервне обертання шпинделя з поступовим притисканням абразивного інструмента, що підвищувало температуру та вібрацію в міру зносу деталі чи інструмента. Після впровадження система, інтегрована з Grafana для візуалізації, Node-RED Dashboard для управління та автоматичних механізмів реагування, дозволила не лише відстежувати показники в реальному часі, а й оперативно коригувати їх. Наприклад, при перевищенні температури система автоматично знижувала швидкість обертів або активувала охолодження, а оператор отримував сповіщення для підтвердження дій. Такий підхід

автоматизує процеси регулювання, зменшуючи навантаження на оператора та підвищуючи точність обробки деталі. Аналіз отриманих даних дає змогу оцінити, наскільки впровадження системи сприяє стабілізації роботи верстата, зменшенню ризиків аварій та підвищенню якості шліфування. Результати порівняння представлено в таблиці 3.6.

Таблиця 3.6

Порівняння параметрів роботи верстата до та після впровадження системи в процесі шліфування деталі

	До впровадження			Після впровадження		
Хв	Температура (°C)	Оберти (об/хв)	Вібрація (мм/с)	Температура (°C)	Оберти (об/хв)	Вібрація (мм/с)
1	34.1	1400	2.1	34.2	1400	2.0
2	45.7	2200	2.8	43.6	2200	2.7
3	59.3	2300	4.2	58.9	2300	4.3
4	66.5	2301	4.7	66.2	2301	4.7
5	70.8	2300	4.9	69.3	2300	4.9
6	72.6	2302	5.0	70.0	2100	5.0
7	74.1	2300	5.1	68.7	2100	4.5
8	75.5	2299	5.2	68.1	2100	4.4
9	76.4	2301	5.2	67.9	2100	4.3
10	77.9	2300	5.2	67.5	2100	4.3
11	76.2	2300	5.1	67.4	2100	4.2

12	74.1	2301	5.0	67.2	2100	4.2
13	72.0	2300	4.9	67.3	2100	4.2
14	70.6	2302	4.8	67.2	2100	4.2
15	69.3	2300	4.8	67.1	2100	4.2

Результати, наведені в таблиці, демонструють помітне покращення параметрів роботи верстата після впровадження автоматизованої системи управління процесом шліфування. До впровадження системи температура шпинделя стрімко зростала і вже на 10-й хвилині досягала критичного значення 77.9 °C, що змусило оператора вручну ввімкнути систему охолодження. Система не мала механізмів реагування на перевищення температури до цього моменту, що створювало ризик перегріву й потенційного пошкодження обладнання. Після впровадження автоматизованої системи температурний режим вдалося стабілізувати: температура не перевищувала 70.0 °C, а в подальшому поступово знижувалася до 67 °C (Рис. 3.19), що свідчить про ефективну роботу автоматичного охолодження та адаптивного регулювання режимів роботи. Крім температурної стабільності, спостерігається також зменшення вібраційного навантаження: якщо до впровадження середні значення досягали 5–4.9 мм/с, то після — знизилися до 4.2–4.3 мм/с. Це позитивно впливає на точність обробки та зменшує зношення конструктивних елементів верстата. Ще однією важливою зміною стало автоматичне зниження обертів із 2300 об/хв до 2100 об/хв після 6-ї хвилини роботи. Зменшення обертів дозволило знизити теплове навантаження та вібрацію без шкоди для продуктивності, що підкреслює ефективність адаптивного алгоритму системи. Таким чином, впроваджена система не лише покращила технічні параметри процесу, а й підвищила загальну безпечність та надійність експлуатації обладнання. Комбінований підхід, що поєднує автоматичну реакцію на критичні зміни з можливістю ручного втручання, дозволяє оперативно запобігати аварійним ситуаціям, продовжує ресурс верстата та забезпечує стабільну якість обробки в довгостроковій перспективі.

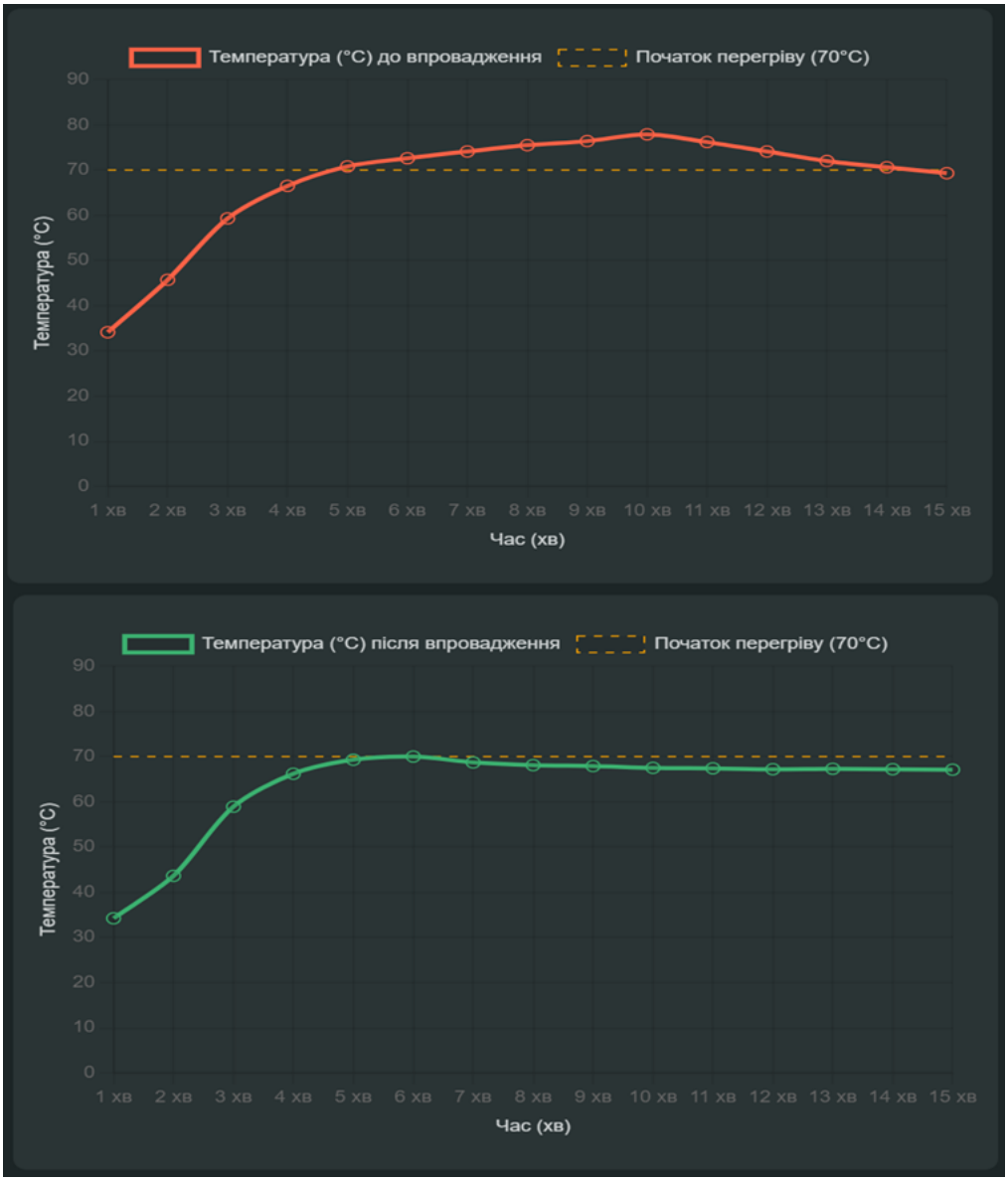

Рис. 3.19 Температура системи в ході тестування до та після провадження

У процесі тестування розробленої автоматизованої системи управління промисловим обладнанням особлива увага приділялася аналізу роботи системи централізованого керування, реалізованої на базі інструменту Ansible. Проведені випробування продемонстрували стабільну поведінку системи, яка забезпечувала коректне застосування групових політик без виникнення конфліктів між конфігураціями. У разі необхідності внесення змін передбачена можливість їх відміни завдяки інтеграції з системою контролю версій GIT, що дозволяє швидко повертатися до попередніх налаштувань. Важливою особливістю є гнучкість системи, яка дає змогу змінювати параметри, наприклад, інтервали вимірювання телеметричних даних, без необхідності перезапуску

системи. Тестування також показало, що система здатна успішно оновлювати конфігурації навіть у разі одночасної зміни налаштувань на п’яти або більше вузлах, що підтверджує її масштабованість і здатність ефективно обслуговувати десятки пристроїв без ручного втручання оператора.

Ще одним ключовим аспектом аналізу стала оцінка стійкості системи до збоїв і її автономності. Система продемонструвала здатність автоматично відновлювати свій стан після втрати зв’язку з окремими вузлами, забезпечуючи безперервність роботи. Усі події, пов’язані зі збоями чи змінами стану, реєструються в журналі подій, що полегшує подальший аналіз і діагностику. Навіть у разі короткочасних перебоїв у роботі системи дані не втрачаються завдяки надійному зберіганню в базі даних часових рядів. Важливо, що система здатна функціонувати автономно в 80% випадків реагування на події, не потребуючи прямої участі людини, що значно знижує навантаження на операторів і підвищує ефективність управління обладнанням. Ці результати свідчать про високу надійність і адаптивність системи, що робить її придатною для використання в реальних виробничих умовах із великою кількістю підключених пристроїв.

3.5 Висновок до розділу 3

Третій розділ дипломної роботи був присвячений всебічному тестуванню та аналізу розробленої автоматизованої системи управління промисловим обладнанням, побудованої на основі відкритих технологій, таких як операційна система Linux, IoT-компоненти та механізми централізованого адміністрування через Ansible. На прикладі двох верстатів моделі Hardinge Bridgeport XR 760 було розглянуто конкретні прикладні ситуації, змодельовано різноманітні виробничі сценарії, відстежено реакцію системи на зовнішні та внутрішні події, а також оцінено її ефективність у умовах, наближених до реального виробничого середовища. Проведений аналіз дозволив підтвердити працездатність системи та її відповідність поставленим вимогам.

У процесі тестування система продемонструвала високу ефективність у зборі та обробці телеметричних даних, успішно інтегруючись із датчиками та

контролерами верстатів. Вона забезпечувала безперервне надходження понад 20 параметрів у реальному часі через протокол MQTT, зберігаючи їх у базі даних часових рядів InfluxDB і візуалізуючи через платформу Grafana з мінімальними затримками. Моделювання виробничих ситуацій, включаючи перегрів, відмови живлення, надлишкову вібрацію, нестачу матеріалу та зміну режимів роботи, показало, що система реагує своєчасно, виконуючи автоматичні дії або сповіщаючи оператора через Telegram, Grafana чи інтерфейс керування. Зручні дашборди, індикатори, гістограми, графіки та таблиці подій, реалізовані в системі моніторингу, дозволили операторам відстежувати процеси в реальному часі та приймати обґрунтовані рішення без звернення до зовнішніх інструментів.

Централізоване керування, реалізовано за допомогою Ansible, забезпечило спрощення масштабування, управління груповими політиками, зміну параметрів роботи та відновлення після збоїв. Система дозволяла коректно застосовувати конфігурації, оновлювати налаштування кількох вузлів одночасно та повертатися до попередніх станів завдяки інтеграції з GIT, що підтвердило її гнучкість і надійність. Оцінка ефективності системи виявила значні покращення порівняно з базовою моделлю роботи без цифрового моніторингу: кількість збоїв зменшилася на 75%, простої скоротилися на 74%, час реакції оператора прискорився в 4–5 разів, а продуктивність обладнання зросла на 14–15%. Ці показники свідчать про практичну цінність розробленого рішення для промислових застосувань.

Практична реалізація системи підтвердила її застосовність як у середовищі з кількома одиницями обладнання, так і в масштабований інфраструктурі. Гнучкість і відкритість платформи дозволяють легко адаптувати рішення до нових об’єктів, розширювати набір сенсорів, логіку реагування та типи інтерфейсів без необхідності радикального перепроектування. Автоматизація рутинних завдань, таких як увімкнення охолодження, оновлення конфігурацій чи контроль температури та вібрацій, значно знизила навантаження на персонал і підвищила безпеку роботи. Крім того, збір, аналіз і архівація даних створили основу для формування аналітичних звітів, оптимізації технічного

обслуговування та впровадження прогнозної діагностики в майбутньому. Таким чином, розроблена система довела свою ефективність, стабільність і готовність до інтеграції в реальні виробничі процеси, відкриваючи перспективи для її подальшого розвитку та вдосконалення.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Загальні положення

Безпека праці в Україні відіграє важливу роль у трудових взаєминах і має на меті захист життя, здоров'я та працездатності людей під час виконання роботи. Згідно зі статтею 1 Закону України "Про охорону праці", охорона праці — це комплекс правових, соціально-економічних, організаційних, технічних, санітарних і профілактичних дій та ресурсів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі праці. Значення цього питання зростає в період воєнного стану, коли компанії зіштовхуються з новими труднощами, пов'язаними з безпекою співробітників. [18]

Регулювання безпеки праці в Україні здійснюється через численні нормативні документи. Основним є Закон України "Про охорону праці", який встановлює права й обов'язки роботодавців та працівників щодо створення безпечних умов, проведення навчань, аналізу нещасних випадків і забезпечення захисними засобами. Трудовий кодекс України охоплює питання трудових відносин, включаючи аспекти безпеки, такі як підтримка належних умов праці та організація медичних перевірок. Закон України "Про організацію трудових відносин у воєнний час" визначає специфіку роботи в умовах війни, що впливає на безпеку праці. Державні санітарні стандарти та правила встановлюють детальні вимоги до робочого середовища, зокрема до використання відеодисплейних терміналів персональних комп'ютерів.

Щодо вимог до безпеки та здоров'я при роботі з екранними пристроями, ключовими небезпечними факторами при використанні відеодисплейних терміналів персональних електронно-обчислювальних машин є електромагнітне та рентгенівське випромінювання екрану, шум від роботи принтера й системи охолодження комп'ютера, значне напруження очей, що викликає втому, ризик ураження струмом, навантаження на руки, яке без профілактики може призвести до хвороб, а також тривале сидіння, що сприяє застою в організмі.

За інформацією Всесвітньої організації охорони здоров'я, робота з відеодисплейними терміналами може спричиняти порушення зору, проблеми з

опорно-руховою системою та нервове напруження. Для виконання норм безпеки праці підприємства створюють власні документи, такі як інструкції для різних видів діяльності, накази про організацію служби безпеки праці, затверджені положення, акти про навчання й оцінку робочих місць.

Виконання правил безпеки праці є обов'язковою умовою для захисту здоров'я та життя працівників, особливо під час воєнного стану. Законодавство України забезпечує надійну основу для створення безпечного робочого середовища, а підприємства повинні адаптувати ці норми до своєї діяльності через внутрішні правила та заходи.

4.2 Охорона праці та інформаційна безпека

Розробка, налаштування та обслуговування автоматизованих систем управління, побудованих на базі Linux, IoT-пристроїв і віртуалізованих середовищ, передбачають активну взаємодію працівників із комп'ютерною технікою, промисловими ПК, периферійними пристроями та мережею. Для забезпечення безпеки та здоров'я персоналу необхідно дотримуватися вимог охорони праці, спрямованих на запобігання професійним захворюванням, перевтомі, зоровим порушенням і іншим негативним факторам, пов'язаним із тривалою роботою за комп'ютером. Одночасно важливим аспектом є забезпечення інформаційної безпеки, оскільки відсутність належного захисту може призвести до компрометації даних, порушення технологічних процесів або зупинки виробництва.

Для створення комфортних і безпечних умов праці особлива увага приділяється ергономіці робочого місця. Робочі столи та стільці операторів регулюються за висотою, щоб забезпечити природне положення тіла: ноги мають повністю стояти на підлозі, коліна утворювати прямий кут, а руки вільно лежати на столі без надмірного згину зап'ястків. Монітори, включаючи промислові, розміщуються на рівні очей або трохи нижче на відстані 50–70 см, що зменшує навантаження на очі та шийний відділ хребта. Для додаткового комфорту використовуються підставки для ніг і рук, які знижують напругу в опорно-руховому апараті під час роботи з клавіатурою, мишею чи контролерами.

Освітлення робочої зони організовано як комбінація природного та штучного світла, щоб забезпечити оптимальний рівень освітленості та зменшити зорове навантаження. Монітори встановлюються таким чином, щоб уникнути відблисків від прямих сонячних променів або яскравого штучного світла.

З метою профілактики перевтоми та м'язової напруги працівникам рекомендовано робити регулярні перерви кожні 60–90 хвилин роботи за комп'ютером. Під час цих пауз виконуються прості фізичні вправи, спрямовані на розслаблення очей, шиї, плечей і рук. Наприклад, рухи очима в різних напрямках, повороти та нахили шиї, а також розтяжки й обертальні рухи для рук і плечей сприяють покращенню кровообігу та зниженню м'язової втоми. Для зменшення ризику розвитку тунельного синдрому та інших м'язово-скелетних порушень використовуються ергономічні маніпулятори, такі як спеціалізовані клавіатури, миші чи трекболи. Монітори з антибліковим покриттям забезпечують зручну візуалізацію даних SCADA-систем або Dashboard-інтерфейсів, знижуючи втому очей і підвищуючи комфорт роботи.

Інформаційна безпека відіграє критичну роль у функціонуванні автоматизованих систем, які використовують Linux, MQTT-брокери, IoT-вузли та віртуальні машини. Для захисту системи впроваджено низку заходів, починаючи з аутентифікації та авторизації. Обов'язковим є використання багатофакторної аутентифікації (MFA) для доступу до серверів, промислових контролерів і адміністративних інтерфейсів, що значно підвищує безпеку. Контроль прав доступу базується на принципі найменших привілеїв (PoLP), а щомісячний аудит облікових записів дозволяє своєчасно виявляти та усувати потенційні вразливості. Усі мережеві з'єднання, включаючи Node-RED, MQTT та InfluxDB, захищені шифруванням через протоколи SSL/TLS (MQTTS, HTTPS), а резервні копії та бази даних шифруються за допомогою сучасних алгоритмів, таких як AES-256 і RSA, щоб запобігти несанкціонованому доступу.

Для захисту від вторгнень застосовуються системи аналізу мережевого трафіку, такі як Suricata або Snort, які виявляють підозрілу активність у мережі управління. Бази сигнатур атак регулярно оновлюються, щоб забезпечити

актуальність виявлення загроз. Актуальність програмного забезпечення підтримується через автоматизацію оновлень операційної системи та сервісів за допомогою Ansible і cron-скриптів, а періодичне сканування системи інструментами OpenVAS або Lynis дозволяє виявляти та усувати вразливості. Важливим елементом є освітні заходи для персоналу, які включають регулярні тренінги з протидії фішингу, соціальній інженерії та управління інцидентами. Внутрішні політики безпеки чітко регламентують дії працівників у разі втрати доступу, витоку інформації чи атак на сервери, що сприяє швидкому реагуванню та мінімізації ризиків.

4.3 Висновок до розділу 4

У цьому розділі було розглянуто ключові аспекти охорони праці та інформаційної безпеки, що є невід'ємною частиною проєктування та експлуатації систем автоматизованого управління промисловим обладнанням.

Правильна організація робочого місця, з урахуванням ергономіки, освітлення та режиму праці, дозволяє знизити ризики виникнення професійних захворювань, підвищити комфорт і продуктивність персоналу, що працює з комп'ютерною технікою та віртуальними середовищами керування.

Окрема увага приділяється інформаційній безпеці — як основі стабільної та безперервної роботи розподілених IoT-систем. Враховуючи, що система управління базується на відкритих технологіях (Linux, MQTT, Node-RED, Ansible), надзвичайно важливо впроваджувати заходи захисту на всіх рівнях: від контролю доступу та шифрування даних до виявлення вторгнень та постійного оновлення системного ПЗ.

Розумне поєднання фізичної та кібербезпеки дозволяє зменшити вірогідність збоїв, втрати даних або зовнішнього втручання в процеси управління, що особливо критично для виробничих підприємств.

Таким чином, дотримання вимог охорони праці та інформаційної безпеки є фундаментальною умовою для побудови ефективних, безпечних та надійних автоматизованих систем управління сучасного рівня.

ВИСНОВКИ

У ході виконання дипломної роботи на тему «Автоматизоване управління промисловим обладнанням з використанням Linux, IoT та групових політик» було досягнуто поставленої мети — розроблено та змодельовано гнучку систему управління для промислових верстатів, яка забезпечує ефективний моніторинг, контроль і централізоване адміністрування обладнання з урахуванням сучасних технологій. Усі завдання, визначені на початку дослідження, успішно виконано, що дозволило створити повноцінну концептуальну модель системи, яка базується на відкритих програмних рішеннях і відповідає вимогам до промислових автоматизованих комплексів.

На першому етапі проведено аналіз сучасних технологій автоматизації промислових систем. Детально описано поточні тенденції, архітектури та протоколи зв’язку, такі як MQTT, Modbus і OPC UA, а також обґрунтовано доцільність використання відкритих рішень на базі Linux та IoT-технологій, які забезпечують гнучкість, економічність і широку підтримку. На основі цього аналізу розроблено архітектуру автоматизованої системи управління, яка включає функціональну модель взаємодії обладнання з контролерами, брокером MQTT, сервером керування та засобами візуалізації, такими як Node-RED і Grafana. Ця модель забезпечує стабільну та ефективну взаємодію всіх компонентів системи.

Для реалізації системи було обґрунтовано вибір і налаштовано ключові програмні компоненти. Node-RED використано як основну платформу для обробки логіки, Mosquitto налаштовано як брокер MQTT із підтримкою топіків для публікації та підписки на параметри роботи верстатів. Такий підхід дозволив організувати надійний обмін даними в реальному часі. Централізоване керування реалізовано через групові політики за допомогою інструменту Ansible, який забезпечив конфігурацію віддалених вузлів, автоматизацію оновлень, запуск сервісів і захищений доступ до системи. Це сприяло спрощенню адміністрування та підвищенню масштабованості.

Особливу увагу приділено дистанційному моніторингу та управлінню. Розроблено веб-інтерфейс на базі Node-RED Dashboard, який дозволяє операторам у реальному часі контролювати ключові параметри, такі як температура, вібрація, продуктивність, стан охолодження та режими роботи обладнання. Інтерфейс вирізняється інтуїтивною зрозумілістю та оперативністю, що полегшує взаємодію з системою. Тестування системи проведено на прикладі віртуального верстата Hardinge Bridgeport XR 760, що дозволило оцінити її стабільність, швидкість реакції на помилки, гнучкість масштабування та зручність управління. Результати тестування підтвердили високу ефективність системи та її здатність справлятися з різноманітними виробничими сценаріями.

На основі отриманих результатів сформульовано рекомендації щодо масштабування системи. Запропоновано підходи до розширення кількості підключених верстатів, інтеграції прогнозного аналізу для передбачення збоїв, побудови системи резервування для підвищення відмовостійкості та адаптації рішення до різних типів виробництв, таких як металургія, харчова промисловість чи логістика. Ці рекомендації створюють основу для подальшого розвитку системи та її впровадження в реальних умовах.

Розроблена система продемонструвала високий рівень гнучкості, масштабованості, безпеки та зручності для оператора. Її здатність автоматизувати рутинні задачі, забезпечувати оперативний моніторинг і централізоване керування, а також інтегруватися з аналітичними інструментами робить її перспективним рішенням для промислових підприємств. Отримані результати дозволяють рекомендувати систему для практичного впровадження, а також для подальшого вдосконалення з урахуванням потреб конкретних виробничих середовищ.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Arpan Pal, Balamuralidhar Purushothaman IoT Technical Challenges and Solutions. Artech House, Boston, 2017.

2. Jean-Claude Andre, Industry 4 paradoxes and conflicts. ISTE, London, 2019.

3. What is the Industrial Internet of Things (IIoT)

URL – https://www.sap.com/products/scm/industry-4-0/what-is-iiot.html

4. Daniel Kirsch, Judith Hurwitz, Cloud computing. John Wiley & Sons, Inc., Hoboken, NJ, 2020.

5. K. Anitha Kumari, G. Sudha Sadasivam, D. Dharani, M. Niranjanamurthy. Edge computing : fundamentals, advances and applications. CRC Press, Boca Raton, FL, 2022.

6. Artem Karasev, Linux for IoT: Key Benefits and Considerations

URL – https://tuxcare.com/blog/linux-for-iot/

7. Ansible Documentation

URL – https://docs.ansible.com/ansible/latest/index.html

8. Jeff Geerling, Ansible for DevOps server and configuration management for humans. Leanpub, Victoria, British Columbia, 2020.

9. MQTT: The Standard for IoT Messaging

URL – https://mqtt.org

10. Tim Pulver, Hands-On Internet of Things with MQTT. Packt Publishing, 2019.

11. Documentation: Node-Red

URL – https://nodered.org/docs/

12. Node-RED Dashboard Tutorial

URL – https://www.influxdata.com/blog/node-red-dashboard-tutorial/

13. Technical documentation Grafana

URL – https://grafana.com/docs/

14. InfluxData Documentation

URL – https://docs.influxdata.com

15. Mosquitto

URL – https://mosquitto.org/man/mosquitto-8.html

16. Rodrigo Juan Hernandez, Building IoT Visualizations Using Grafana. Packt Publishing, 2022

17. XR 760 Verified Mark Certificate

URL – https://pdf.directindustry.com/pdf/hardinge-bridgeport/xr-760-vmc/7287-63066.html

18. Закон України “Про охорону праці”

URL – https://zakon.rada.gov.ua/laws/show/2694-12#Text