

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра автоматизації та комп'ютерно-інтегрованих технологій

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри автоматизації та
комп'ютерно-інтегрованих технологій

_____ Микола СІДЄЛЄВ

«__» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
<Система автоматизації виробництва через впровадження ІоТ-технологій з
використанням соціальних мереж>

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

Освітня програма «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка»

Здобувач _____

Семен СИЧОВ

«__» _____ 20__ р.

Керівник роботи

Канд.техн.наук,

доцент _____

Микола СІДЄЛЄВ

«__» _____ 20__ р.

Миколаїв – 2025

Завдання на виконання кваліфікаційної роботи

Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Автоматизації та комп'ютерно-інтегрованих технологій
Рівень вищої освіти	Другий (магістерський)
Освітній ступінь	Магістр
Спеціальність	174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Освітня програма	Автоматизація, комп'ютерно-інтегровані технології та робототехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри автоматизації та
комп'ютерно-інтегрованих технологій

_____ Микола СІДЕЛЄВ

«__» _____ 2025 р.

ЗАВДАННЯ

на кваліфікаційну магістерську роботу здобувача

Сичов Семен

1. Тема кваліфікаційної роботи < Система автоматизації виробництва через впровадження IoT-технологій з використанням соціальних мереж > затверджена наказом ректора ЧНУ ім. Петра Могили № 298 від «30» жовтня 2024 р.
2. Строк представлення кваліфікаційної роботи «19» червня 2025 р.
3. Очікуваний результат роботи та початкові дані якщо такі потрібні.

Очікується створення функціональної IoT-системи для автоматизації виробництва металопластикових вікон із використанням ESP32, MQTT та Telegram-бота для моніторингу і керування. Початковими даними виступають технічні характеристики обладнання підприємства, нормативно-правові документи та параметри сенсорів і контролерів. Система має забезпечити оперативний контроль параметрів виробничого процесу та безпеки з мінімальними витратами.

4. Перелік питань, що підлягають розробці:

- аналіз технологічного процесу виготовлення металопластикових вікон та виявлення точок для впровадження автоматизації;
- розробка структури IoT-системи на базі ESP32, MQTT та Telegram-бота для моніторингу виробничих параметрів;
- побудова логічної схеми обміну даними між сенсорами, контролерами, брокером та мобільним інтерфейсом;
- оцінка ризиків і розробка заходів безпеки в умовах надзвичайних ситуацій із використанням цифрових засобів оповіщення.

5. Перелік графічних матеріалів:

а) рисунок — концептуальна модель автоматизованої IoT-системи у виробництві вікон, яка відображає загальну архітектуру з п'ятьма рівнями: сенсори → контролер → мережа → обробка → інтерфейс;

б) функціональна схема — схема роботи зварювальної ділянки з визначеними точками збору технологічних параметрів;

в) схема — технічна архітектура IoT-системи автоматизації на основі контролера ESP32, протоколу MQTT та Telegram API, що ілюструє структуру, обміну даними між апаратними та програмними компонентами;

г) рисунок — архітектура програмного комплексу з розподілом за модулями: сенсори, MQTT-зв'язок, логічна обробка, інтерфейс Telegram та аналітика, що демонструє модульну побудову системи.

Дата видачі завдання «09» вересня 2024 р.

Календарний план виконання кваліфікаційної роботи

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: < Система автоматизації виробництва через впровадження IoT-технологій з використанням соціальних мереж >

№	Найменування роботи	Початок	Закінчення	Примітки
	Розробка та затвердження завдання на виконання КМР	09.09.2024	30.10.2024	
	Огляд літератури за темою роботи	31.10.2024	15.11.2024	
	Складання календарного плану КМР	18.11.2024	20.11.2024	
	Аналіз предметної області	21.11.2024	20.12.2024	
	Розробка проєктних рішень	23.12.2024	28.02.2025	
	Моделювання, конструювання та програмування СА та РТС	03.03.2025	18.04.2025	
	Дослідження СА та РТС, аналіз результатів	21.04.2025	16.05.2025	
	Відгук керівника КМР	19.05.2025	23.05.2025	
	Оформлення КМР та презентації	26.05.2025	04.06.2025	
	Попередній захист	05.06.2025	06.06.2025	
	Рецензування	16.06.2025	17.06.2025	
	Завершення оформлення КМР та презентації	18.06.2025	19.06.2025	
	Захист кваліфікаційної роботи	26.06.2025	27.06.2025	

Здобувач _____

Семен СИЧОВ

«__» _____ 20__ р.

Керівник роботи

Канд.техн.наук,

доцент _____

Микола СІДЄЛЄВ

«__» _____ 20__ р.

2025 р.

Сичов С.Ю.

АНОТАЦІЯ

В даній магістерській роботі розроблено систему автоматизації керування виробничим процесом по виготовленню металопластикових вікон з інтеграцією технологій Інтернету речей (IoT) та соціальних мереж, зокрема Telegram, в якості НМІ. Проведено сукупний аналіз новітніх підходів до цифрової трансформації виробництва та обґрунтовано вибір архітектури автоматизованої системи за рахунок децентралізованих сенсорних вузлів, мікроконтролерів ESP32, протоколу MQTT та хмарного сервісу обміну повідомленнями.

Розроблено математичну модель ключових ділянок виробництва (різання, армування, зварювання, зачистка, скління, контроль якості), на основі яких реалізовано програмно-апаратний комплекс з функціями моніторингу, сповіщення та прийняття рішень в режимі реального часу. Роботу системи перевірено за допомогою імітаційного моделювання, що дозволило оцінити ключові показники її ефективності.

Запропонована система є легко масштабованою, доступною та придатною для впровадження на малих та середніх виробничих підприємствах. Диплом містить практичні рекомендації щодо впровадження технологій Інтернету речей у традиційне виробництво та забезпечення безпечних умов праці в цифровому середовищі.

Кваліфікаційна магістерська робота викладена на 113 сторінках (без додатків), містить 4 розділи, 62 рисунків, 3 додатки, 19 таблиць та 15 джерел посилань.

Ключові слова: *IoT, автоматизація, металопластикові вікна, ESP32, MQTT, Telegram-бот, моделювання, система моніторингу, цифрове виробництво.*

ABSTRACT

In this master's thesis, an automation system for controlling the production process for the manufacture of metal-plastic windows with the integration of Internet of Things (IoT) technologies and social networks, in particular Telegram, as an HMI was developed. A cumulative analysis of the latest approaches to the digital transformation of production was carried out and the choice of the architecture of the automated system was justified due to decentralized sensor nodes, ESP32 microcontrollers, MQTT protocol, and a cloud messaging service.

A mathematical model of the key production areas (cutting, reinforcement, welding, stripping, glazing, quality control) was developed, on the basis of which a software and hardware complex with real-time monitoring, notification, and decision-making functions was implemented. The system was tested using simulation modeling, which made it possible to evaluate its key performance indicators.

The proposed system is easily scalable, affordable, and suitable for implementation in small and medium-sized manufacturing enterprises. The diploma contains practical recommendations for implementing IoT technologies in traditional production and ensuring safe working conditions in the digital environment.

The qualifying master's thesis is set out on 113 pages (without appendices), contains 4 chapters, 62 figures, 3 applications, 19 tables, and 15 references.

Keywords: *IoT, automation, metal-plastic windows, ESP32, MQTT, Telegram bot, modeling, monitoring system, digital production.*

ЗМІСТ

ВСТУП.....	12
1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО АВТОМАТИЗАЦІЇ ВИРОБНИЧИХ ПРОЦЕСІВ ІЗ ВИКОРИСТАННЯМ ІОТ-ТЕХНОЛОГІЙ ТА СОЦІАЛЬНИХ МЕРЕЖ.....	15
1.1 Сучасний стан автоматизації виробничих процесів.....	15
1.1.1 Етапи еволюції автоматизації.....	16
1.1.2 Традиційні системи автоматизації.....	17
1.1.3 Проблеми запровадження традиційних систем у секторах малого та середнього бізнесу.....	18
1.2 Інтернет речей (IoT) як інструмент нової автоматизації.....	19
1.2.1 Основні концепції та принципи IoT.....	20
1.2.2 Архітектура IoT-систем у виробничому середовищі.....	21
1.2.3 Приклади застосування IoT у виробництві.....	23
1.3 Соціальні мережі як канал взаємодії у виробничих процесах.....	24
1.3.1 Інтеграція Telegram, Slack, Discord у виробничі системи.....	25
1.3.2 Переваги використання соціальних мереж у промисловості.....	26
1.4 Технічні рішення та платформи для інтеграції IoT і соціальних мереж....	28
1.4.1. Протоколи обміну даними: MQTT, HTTP, WebSocket.....	28
1.4.2 Можливості API Telegram для автоматизації.....	29
1.5 Патентний огляд.....	30
1.5.1 Аналіз релевантних патентів.....	31
1.5.2 Основні висновки патентного огляду.....	32
Висновки до розділу 1.....	33
2 ОБҐРУНТУВАННЯ НАПРЯМУ ДОСЛІДЖЕНЬ ТА РОЗРОБКА МЕТОДИКИ ВПРОВАДЖЕННЯ ІОТ-СИСТЕМИ АВТОМАТИЗАЦІЇ ВИРОБНИЦТВА.....	35
2.1 Обґрунтування вибору напрямку дослідження.....	35
2.2 Методика аналізу системи автоматизації виробничих процесів.....	37
2.2.1 Функціональний аналіз об'єкта автоматизації.....	37
2.2.2 Структурно-логічне представлення системи.....	38
2.2.3 Інформаційно-параметричний аналіз.....	39

2.2.4 Аналіз сценаріїв роботи системи	40
2.2.5. Інтерфейсна модель: інтеграція з користувачем	41
2.3 Методологія синтезу автоматизованої системи.....	42
2.3.1 Етапи синтезу автоматизованої системи	43
2.4 Використання комп'ютерних технологій та цифрових інструментів	47
2.4.1 Програмування мікроконтролера (ESP32)	48
2.4.2 Комунікаційний протокол: MQTT	49
2.4.3 Реалізація Telegram-бота	50
2.4.4 Інструменти обробки та візуалізації даних	51
2.4.5 Візуальне моделювання на прикладі ділянки зварювання	51
2.4.6 Допоміжне ПЗ.....	52
2.4.7 Структура програмного комплексу.....	52
2.5 Математична модель ділянки розкрою ПВХ-профілю.....	53
2.5.1 Кінематична модель позиціонування	54
2.5.2 Силова й енергетична модель відрізу	55
2.5.3 Стохастична модель похибок	55
2.5.5 Інтеграція з IoT-системою.....	56
2.6 Математична модель ділянки армування.....	57
2.6.1 Розрахунок сили вставлення.....	58
2.6.2 Перекіс підсилювача.....	59
2.6.3 Кінематика та момент загвинчування.....	59
2.6.4 Стохастична модель допусків.....	59
2.6.5 Алгоритм контролю та інтеграції з IoT	60
2.7 Математична модель ділянки зварювання	61
2.7.1 Тепловий баланс нагріву	63
2.7.2 Силова модель стискання.....	63
2.7.3 Модель тривалості циклу	64
2.7.4 Критерії оцінки якості	64
2.7.5 Алгоритм дій та IoT-інтеграція	65
2.8 Математична модель ділянки зачистки.....	65
2.8.1 Геометрична модель траєкторії зачистки.....	67

2.8.2 Модель знімання шару матеріалу	67
2.8.3 Динамічна модель сили контакту.....	68
2.8.4 Алгоритм контролю якості обробки	68
2.8.5 Інтеграція з системою моніторингу	68
2.9 Математична модель ділянки скління	69
2.9.1 Геометрична модель посадки склопакету	70
2.9.2 Пружна модель ущільнення.....	70
2.9.3 Силова модель заpresування штапиків	70
2.9.4 Рівновага рами під навантаженням.....	71
2.9.5 Інтеграція з системою моніторингу	71
2.10 Математична модель ділянки контролю якості.....	72
2.10.1 Статистична модель браку	72
2.10.2 Геометричний контроль	73
2.10.3 Контроль фурнітури	73
2.10.4 Перевірка герметичності	74
2.10.5 Алгоритм інтеграції оцінок.....	74
2.11 Інтегрована модель виробничого процесу	75
2.11.1 Структура узагальненої моделі	75
2.11.2 Побудова зведеної матриці параметрів	76
2.11.3 Формування інтегрального критерію якості	76
2.11.4 Реалізація наскрізного моніторингу.....	77
2.11.5 Сценарії реакції системи	77
Висновки до розділу 2	77
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ IoT-АВТОМАТИЗАЦІЇ ВИРОБНИЦТВА МЕТАЛОПЛАСТИКОВИХ ВІКОН	80
3.1 Архітектура запропонованої системи управління.....	80
3.1.1 Обґрунтування вибору елементної бази.....	81
3.2 Практична реалізація математичних моделей	82
3.3 Обробка вхідних сигналів із сенсорів	84
3.3 Розробка Telegram-інтерфейсу	85
3.3.1 Реалізація команд /refresh_rate, /report, /set_threshold	87

3.4 Імітаційне моделювання роботи системи.....	88
3.4.1 Реалізація імітацій у MATLAB/Simulink та Python.....	89
3.5 Аналіз отриманих результатів	90
3.5.1 Таблиці оцінки точності та достовірності системи.....	91
3.6 Виявлені недоліки, обмеження та шляхи вдосконалення.....	92
3.6.1 Перспективи масштабування, резервування та безпеки.....	93
Висновки до розділу 3	94
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	96
4.1 Аналіз умов праці на виробництві металопластикових вікон.....	96
4.1.1 Гігієнічна оцінка умов праці.....	96
4.2 Організаційно-технічні заходи щодо охорони праці.....	98
4.2.1 Автоматизація небезпечних процесів через IoT-технології.....	99
4.2.2 Система технічного обслуговування та ремонту обладнання	99
4.2.3 Впровадження режиму чергування праці та відпочинку.....	100
4.2.4 Сигналізація та маркування.....	100
4.2.5 Відеоспостереження та контроль допуску	100
4.3 Ергономічні умови праці на виробництві металопластикових вікон.....	101
4.3.1 Біомеханічні навантаження.....	101
4.3.2 Мікроклімат як частина ергономіки	102
4.3.3 Рівень візуального навантаження.....	102
4.3.4 Приклад технічно правильного ергономічного робочого місця для оператора складання	103
4.4 Перелік потенційно небезпечних установок і механізмів	103
4.4.1 Методи захисту персоналу.....	104
4.5 Безпека в надзвичайних ситуаціях	105
4.5.1 Система попередження та реагування на надзвичайні ситуації	106
Висновок до розділу 4	108
ВИСНОВКИ.....	110
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	112
ДОДАТОК А.1 Реалізація структури Telegram бота	114

ДОДАТОК А.2 Команди Telegram-бота: /start, /status, /report, /reset, /set_threshold, /refresh_rate + кнопки.....	115
ДОДАТОК А.3 Генерація PDF-звітів та логування в Google Sheets	117
ДОДАТОК А.3 Код для генерації даних ESP32.....	118
ДОДАТОК Б Основні MQTT-запити та топіки	120
ДОДАТОК В Скріншоти реалізації Telegram-бота	122

ВСТУП

Світова економіка постійно змінюється — і головна причина цього є не політика чи сировинні ринки, а цифрові технології, які все більше входять у всі сфери нашого життя. Одним із головних напрямів цієї трансформації є автоматизація виробництва. Завдяки технологіям Інтернету речей (IoT), компанії вже не обмежуються окремими автоматизованими процесами. Зараз ідеться про повну цифрову інтеграцію: обладнання, системи управління та персонал, все це має взаємодіяти в реальному часі.

Паралельно змінюється і спосіб комунікації. Платформи на кшталт Telegram, Slack чи Facebook Workplace давно перестали бути просто засобами комунікації з друзями. Їх активно використовують у бізнесі — для інформування, координації й швидкого ухвалення рішень. Соціальні мережі стали частиною професійного середовища.

Саме тому ідея поєднати можливості IoT з перевагами соціальних платформ у межах єдиної системи автоматизації вже не виглядає футуристичною — це логічний наступний крок.

Тема магістерської роботи — «Система автоматизації виробництва через впровадження IoT-технологій із використанням соціальних мереж» — точно відображає сучасні тенденції в промисловій автоматизації та цифровізації підприємств. Сьогодні бізнесу доводиться швидко адаптуватися до змін, знижувати витрати й мінімізувати вплив людського фактора. В таких умовах інтеграція розумних пристроїв і гнучких каналів комунікації стає ключовою умовою ефективної роботи виробничих систем.

Незважаючи на те, що традиційні рішення — SCADA, MES, ERP — так і залишаються основою промислового управління, вони не завжди підходять малому й середньому бізнесу. Висока вартість, складна структура й труднощі з адаптацією до змін роблять ці системи малоефективними для гнучких виробничих процесів.

В такому контексті поєднання IoT-платформ із сучасними каналами комунікації, такими як соціальні мережі, відкриває нові можливості. Ідеться про створення рішень, які водночас доступні, масштабовані, зручні у використанні й достатньо надійні для промислових умов.

Інтернет речей дозволяє постійно моніторити стан обладнання, збирати й аналізувати дані в реальному часі, а також швидко передавати інформацію про критичні події або відхилення. Використання популярних месенджерів як каналу зв'язку дає дві ключові переваги: миттєво доставляє критично важливу інформацію оператору, а також відкриває можливість повноцінного віддаленого контролю над виробництвом

Серед доступних платформ Telegram вирізняється особливою зручністю завдяки:

- широкому розповсюдженню серед користувачів;
- простій інтеграції через відкритий API;
- можливості створення зрозумілих бот-інтерфейсів;
- підтримці шифрування даних для безпечної комунікації.

Головним завданням цієї роботи є створення іноваційного прототипу виробничої автоматизації, що поєднує збір даних за допомогою IoT-пристроїв із системою миттєвої комунікації через Telegram. Головні переваги такого підходу є можливість керувати процесами в реальному часі, та значне спрощення впровадження та адаптації системи для різних ділянок.

Це дослідження є актуальним з кількох причин:

- потребою в доступних і гнучких рішеннях для автоматизації;
- необхідністю скорочення часу реакції на зміни у стані обладнання;
- прагненням підвищити ефективність персоналу завдяки простому доступу до актуальної інформації;
- орієнтацією на цифровізацію без значних витрат на модернізацію інфраструктури.

Мета роботи — розробити прототип виробничої автоматизаційної системи, що ґрунтується на IoT-технологіях та інтегрується з Telegram для комунікації й управління в режимі реального часу.

Завдання дослідження:

1) проаналізувати, що вже існує на ринку: які технічні рішення застосовуються у виробництві, як створені подібні ідеї, і які з них запатентовані. Це дасть уявлення про те, з чим доведеться працювати;

2) з'ясувати, чи можна поєднати IoT-пристрої зі звичними каналами комунікації. Йдеться про такі інструменти, як Telegram або інші месенджери, де люди вже звикли отримувати сповіщення й швидко реагувати;

3) визначити, яке саме обладнання і яке програмне забезпечення варто використати для створення робочого прототипу. Рішення має бути реалістичним — не надто дорогим, але достатньо надійним;

4) побудова самої системи. Потрібно продумати її структуру, як саме вона працюватиме, як збирає й передає інформацію, і як на це реагуватиме користувач;

5) тестування. Систему потрібно перевірити в умовах, які хоча б приблизно нагадують реальне виробництво, і зрозуміти, наскільки вона справді допомагає.

Об'єктом дослідження є моніторинг виробничих процесів, які можна автоматизувати завдяки цифровим технологіям.

Предмет дослідження — створення автоматизованої системи, яка використовує IoT і соцмережі для зручного управління виробництвом.

Автоматизація — це практична потреба. Проте не всі підприємства можуть дозволити собі складні системи. Ідея цієї роботи — показати, як навіть з простими інструментами можна зробити ефективне рішення. Це дозволить малому бізнесу отримати нову можливість для цифрового розвитку без зайвих витрат.

1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО АВТОМАТИЗАЦІЇ ВИРОБНИЧИХ ПРОЦЕСІВ ІЗ ВИКОРИСТАННЯМ ІОТ-ТЕХНОЛОГІЙ ТА СОЦІАЛЬНИХ МЕРЕЖ

1.1 Сучасний стан автоматизації виробничих процесів

Автоматизація виробничих процесів має довгу й складну історію, що суттєво вплинуло на розвиток світової економіки та формування сучасної промисловості [1]. Сьогодні вона — не просто інструмент для підвищення продуктивності, а необхідна умова, без якої підприємства в різних галузях не можуть залишатися конкурентоспроможними. Це стосується як легкої промисловості, так і складного машинобудування.

Перші спроби автоматизації з'явилися ще в XVIII столітті, під час промислової революції. Тоді йшлося переважно про заміну ручної праці простими машинами, здатними виконувати базові механічні операції. Наприклад, ткацькі верстати значно пришвидшили виробництво тканин і знизили потребу в ручній праці [1].

Згодом автоматизація проходила кілька важливих етапів. Від механічних конструкцій — до електромеханіки, а потім і до використання електроніки та комп'ютерів [3]. Поява мікропроцесорів та розвиток інформаційних технологій стали переломним моментом. Саме тоді сформувалися перші системи автоматизованого управління, які дозволили об'єднати роботу обладнання, програмного забезпечення й ІТ-сервісів в єдину систему.

Сучасні рішення значно складніші. Сьогодні автоматизація — це вже не окремі машини чи програми, а повноцінні кіберфізичні системи. Вони складаються з фізичних пристроїв, які оснащено датчиками, модулями зв'язку, процесорами й програмним забезпеченням. Такі системи здатні взаємодіяти не лише між собою, а й з користувачами — і все це в режимі реального часу [3]. Вони не тільки швидко обробляють дані й ухвалюють рішення, а й можуть адаптуватися до змін. Це важливий фактор в умовах постійної конкуренції та нестабільного середовища.

1.1.1 Етапи еволюції автоматизації

Автоматизація виробничих процесів пройшла кілька ключових етапів, кожен із яких суттєво трансформував підходи до організації праці, технологій і ресурсного управління:

1) початковий етап (кінець XVIII — початок XIX століття). На цьому етапі головним завданням було механічна заміна ручної праці з метою підвищення продуктивності та зниження витрат. Прикладом є поява ткацьких верстатів, які автоматизували процес виготовлення тканини [1]. Запровадження таких машин значно прискорило виробництво та підвищило його ефективність, що сприяло подальшому технічному прогресу;

2) друга індустріальна революція (кінець XIX — початок XX століття). Цей етап позначився впровадженням електроенергії як основного джерела живлення для виробничого обладнання [1]. Застосування електричних двигунів і конвеєрних ліній дозволило підвищити рівень автоматизації, масштабувати виробництво та зменшити залежність від фізичної праці;

3) третя індустріальна революція (друга половина XX століття). Подальший розвиток автоматизації був зумовлений впровадженням електроніки [3]. Широке застосування програмованих логічних контролерів (PLC) забезпечило гнучке керування виробничими процесами без постійного втручання оператора. У цей період з'явилися перші комп'ютеризовані системи управління, які суттєво підвищили точність, швидкість і ефективність виробництва.

4) четверта індустріальна революція (початок XXI століття). Сучасний етап автоматизації характеризується глибокою інтеграцією цифрових технологій, а саме Інтернету речей (IoT), штучного інтелекту, великих даних (Big Data) та хмарних обчислень, у виробничі процеси [2]. Це дає змогу створювати «розумні» виробничі системи, в яких обладнання самостійно приймає рішення на основі аналізу даних у реальному часі.

Розвиток автоматизації пройшов шлях від простих механізмів до складних цифрових рішень за допомогою тісної інтеграції з інформаційними технологіями.

1.1.2 Традиційні системи автоматизації

У другій половині XX століття, з розвитком електроніки та обчислювальної техніки, були створені перші комплексні системи управління виробничими процесами. Вони стали основою сучасної промислової автоматизації та сформували фундамент для подальшого розвитку цифрових технологій у виробництві.

1) SCADA-системи (Supervisory Control and Data Acquisition). SCADA-системи призначені для збору даних і дистанційного управління технологічними процесами [3]. Вони забезпечують операторам можливість у реальному часі відстежувати стан обладнання, аналізувати показники процесів та керувати виробничими установками за допомогою людино-машинного інтерфейсу;

2) MES-системи (Manufacturing Execution Systems). MES-системи орієнтовані на управління виробничими операціями на рівні цеху [3]. Їхні функції охоплюють планування, облік матеріалів, контроль якості, моніторинг виконання замовлень та оптимізацію ресурсів у реальному часі. Такі системи є критично важливими для забезпечення ефективного виконання виробничих завдань;

3) ERP-системи (Enterprise Resource Planning). ERP-системи забезпечують інтеграцію основних бізнес-процесів підприємства — виробництва, фінансів, логістики, управління персоналом — в єдине інформаційне середовище [3]. Їх впровадження сприяє централізованому управлінню ресурсами, підвищенню прозорості операцій та загальній ефективності діяльності підприємства.

Завдяки широкому функціоналу ці системи дозволяють управляти складними виробничими комплексами, охоплюючи процеси від керування технологічними лініями до координації фінансових і логістичних потоків [3]. Їх

застосування сприяє зниженню витрат, підвищенню гнучкості та оперативній реакції на зміни ринкових умов.

Водночас традиційні системи автоматизації мають певні обмеження, особливо відчутні в контексті малих та середніх підприємств. Обмежені фінансові ресурси, нестача кваліфікованого персоналу та потреба у швидкій адаптації до змін роблять класичні рішення занадто складними, дорогими та малоефективними для таких компаній [3].

1.1.3 Проблеми запровадження традиційних систем у секторах малого та середнього бізнесу

Традиційні методи не завжди є найкращим варіантом для невеликих та середніх підприємств виробничої сфери. Основні труднощі полягають у наступному:

1) значні витрати на впровадження та обслуговування. Закупівля ліцензійного програмного забезпечення, серверного обладнання та розробка індивідуальних рішень вимагають значних витрат. Крім того, оплата праці висококваліфікованих спеціалістів може бути надто великою для багатьох малих компаній. Часто початковий капітал залишається недосяжним для бюджету підприємства;

2) складність адаптації. Більшість систем SCADA та MES розробляються з урахуванням потреб великих промислових об'єктів. Адаптація їх під специфіку невеликих виробничих підприємств потребує значних зусиль у часі і грошей, а також технічної експертизи, яка часто не може бути обґрунтованою;

3) проблеми з обслуговуванням. Підтримка складних ІТ-систем потребуватиме наявності в колективі або за допомогою аутсорсингу інженерів з високою кваліфікацією в області промислової автоматизації [3]. Залучення таких фахівців або регулярне обслуговування зовнішніми компаніями є дорогим та не завжди зручним вибором для невеликого бізнесу;

4) низька гнучкість. Класичні системи розроблені для стабільних умов виробництва без значних змін [3]. Однак при змінах у технологічних процесах або продуктовому асортименті або організаційній структурі вносити корективи до такої системи може бути важким і громоздким процесом;

5) складність взаємодії з користувачем через інтерфейс. Багато традиційних SCADA-, MES- і ERP-систем мають складні та менш зрозумілі інтерфейси для користувача [3]. Для роботи з ними потрібно додаткове навчання операторів та інженерів, що створюється до додаткових витрат і затримок під час впровадження.

Ці фактори доводять, що малий та середній бізнес потребують іншого підходу до автоматизації. Рішення мають бути не лише ефективними, а й доступними за фінансовою складовою, простими у впровадженні, легкими в обслуговуванні й здатними швидко адаптуватись під зміну умов.

Саме такі вимоги відкривають простір для впровадження рішень на основі технологій Інтернету речей (IoT). Вони дають змогу створювати сучасні виробничі системи, які краще відповідають реальним потребам малого бізнесу.

1.2 Інтернет речей (IoT) як інструмент нової автоматизації

У сучасних умовах швидкого розвитку промисловості Інтернету речей (IoT) стрімко стає однією з ключових технологічних платформ для встановлення ідеалів Четвертої та П'ятої промислових революцій.

Промисловість 4.0 спрямована на створення «розумних» виробничих систем, де взаємодія між машинами і людьми відбувається безперервно та автоматично. Протилежністю до неї є створення «розумних» виробничих систем де інтеграція машин з людьми трактується як головна проблема з відкритою ідеологічною крайністю для новоприбутких до промисловості процесів розвитку виробництва IoT.

Якщо традиційні системи автоматизації зосереджені навколо централізованого керування великими виробничими спорудами, то IoT пропонує нову модель – децентралізовану та гнучку для підприємств різних масштабів.

Наразі можливо ефективно автоматизувати роботу як на великих заводах, так і на невеликих виробничих ділянках без потреби інвестувати значні кошти в складне централізоване обладнання.

На відміну від традиційних систем із центральним сервером як єдиним «мозком», в мережі Інтернету речей самостійні пристрої стали більш розумними, до них можна віднести такі компоненти як: сенсори, контролери, приводи. Вони самі обробляють інформацію, приймають прості рішення та спілкуються між собою без необхідності звертатися до центрального вузла. Це значно покращує надійність та гнучкість виробничих процесів.

1.2.1 Основні концепції та принципи IoT

Термін «мережа речей» позначає глобальну систему фізичних об'єктів із вбудованими сенсорами та мікроконтролерами з можливістю автономного збору даних та обміну інформацією між собою та з іншими системами чи людьми.

Основні концепції передбаченого IoT включають:

1) автономність. Пристрої можуть автоматично взаємодіяти без прямого втручання людини. Наприклад, датчик температури автоматично передає сигнал до контролера який обробляє цю інформацію для того, щоб прийняти рішення про увімкнення або вимкнення обладнання;

2) підключення. Ключовою умовою для функціонування систем IoT є наявність надійного з'єднання між пристроями. Фізичні об'єкти обладнуються модулями для передачі даних через дротові або бездротові мережі. Найчастіше використовуються Wi-Fi, Bluetooth, Zigbee, а також мобільні стандарти зв'язку — 4G і 5G [2]. Це дозволяє забезпечити постійний обмін інформацією між пристроями в реальному часі, що є критично важливим для безперервного контролю за виробничими процесами;

3) обробка даних. Інформація, яку збирають сенсори, може оброблятися безпосередньо на рівні контролерів — локально, або передаватися на центральні сервери чи хмарні платформи [4]. У таких випадках обробка включає складніші операції: аналіз даних, побудову прогнозів і візуалізацію результатів. Це дозволяє не лише реагувати на події в режимі реального часу, а й виявляти закономірності, що допомагають оптимізувати виробничі процеси;

4) масштабованість. Одна з переваг IoT — здатність легко розширювати систему. Додавання нових пристроїв або впровадження додаткових функцій не вимагає повної реконструкції існуючої мережі [5]. Такий підхід особливо зручний для підприємств, які поступово збільшують обсяги виробництва або оновлюють обладнання. IoT дозволяє зберегти стабільність системи навіть під час активного зростання;

5) гнучкість. Уміння адаптуватися до змін — ще один ключовий принцип побудови IoT-систем. На відміну від традиційних рішень, які часто потребують повного перепроектування при зміні процесів, IoT дозволяє вносити коригування швидко і без суттєвих витрат [4]. Наприклад, якщо виробнича лінія переходить на випуск іншого продукту, достатньо змінити налаштування контролерів і сенсорів, без потреби у зміні всієї структури автоматизації.

1.2.2 Архітектура IoT-систем у виробничому середовищі

Типова архітектура IoT-систем, що застосовується на виробництві, має багаторівневу структуру. Вона організована таким чином, щоб забезпечити безперервний і надійний потік даних — від сенсорів, які реєструють фізичні параметри, до систем, що аналізують інформацію і приймають рішення [4], [7]. Така побудова дозволяє гнучко масштабувати систему та ефективно реагувати на зміни в робочому середовищі.

Архітектура IoT-систем складається з наступних рівнів:

1) рівень сенсорів (Sensors Layer). На цьому рівні відбувається безпосередній контакт із виробничим процесом. Сенсори фіксують критичні

показники, зокрема температуру, тиск, вологість, рівень вібрацій та інші параметри. Саме ці дані є основою для подальшого аналізу та оперативного керування обладнанням;

2) рівень обробки (Processing Layer). Отримані від сенсорів дані передаються на пристрої попередньої обробки, зокрема на контролери ESP32, STM32 або мікрокомп'ютери на базі Raspberry Pi [4]. У більшості випадків ці пристрої здатні самостійно приймати рішення у межах заданих алгоритмів. Наприклад, вони можуть зупинити обладнання у разі перевищення граничного значення температури або тиску;

3) комунікаційний рівень (Communication Layer). Цей рівень відповідає за передачу даних між пристроями, серверами та іншими вузлами мережі. Залежно від конкретної реалізації, використовуються протоколи MQTT [8], HTTP/HTTPS або WebSocket. Головна вимога — стабільний та швидкий обмін даними, з мінімальними затримками;

4) рівень зберігання та аналітики (Storage and Analytics Layer). Зібрана інформація зберігається або на локальних серверах, або у хмарних сервісах, таких як AWS IoT чи Azure IoT [4]. Подальший аналіз цих даних дає змогу прогнозувати поломки обладнання, покращувати ефективність роботи та планувати технічне обслуговування на основі реального стану системи;

5) рівень взаємодії з користувачем (Application Layer). На цьому рівні результати аналізу відображаються в зручній для користувача формі. Це можуть бути операторські панелі, мобільні застосунки або чат-боти. Наприклад, у разі аварійної ситуації система може надіслати сповіщення у Telegram, де оператор отримає повідомлення і за потреби зможе оперативно втрутитися [9].

Завдяки багаторівневій архітектурі IoT-системи демонструють високу адаптивність, швидке реагування на зміни й здатність до поступового розширення без необхідності повної перебудови всієї інфраструктури.

1.2.3 Приклади застосування IoT у виробництві

Практичне впровадження технологій Інтернету речей у виробничих процесах демонструє значний потенціал для підвищення ефективності, зменшення витрат і покращення контролю над обладнанням. Наведені нижче приклади ілюструють ключові напрями їхнього застосування:

1) моніторинг технічного стану обладнання. Одним із найпоширеніших сценаріїв є встановлення сенсорів температури, вібрацій та інших параметрів на критичних вузлах машин [2]. Завдяки цьому можна виявляти потенційні несправності ще до того, як вони призведуть до зупинки виробництва. Такий підхід дає змогу скоротити час простоїв, зменшити витрати на аварійний ремонт і продовжити строк експлуатації обладнання;

2) управління енергоспоживанням. IoT-системи здатні керувати енергетичною інфраструктурою — освітленням, вентиляцією, обігрівом — автоматично, орієнтуючись на дані від сенсорів [5]. Це дозволяє уникати зайвого споживання електроенергії та, відповідно, зменшувати експлуатаційні витрати;

3) оптимізація логістики та складських операцій. Використання RFID-міток і сенсорів IoT для відстеження переміщення товарів спрощує облік на складі та забезпечує точну інформацію про наявність ресурсів у реальному часі [4]. Крім того, ці системи допомагають оптимізувати транспортні маршрути, що знижує витрати на логістику і скорочує час доставки;

4) розумне виробництво (Smart Manufacturing). Інтелектуальні системи дозволяють обладнанню не лише передавати дані про свій стан, а й самостійно змінювати режими роботи відповідно до поточних умов. Наприклад, машина може зменшити швидкість у разі виявлення підвищеного навантаження або ініціювати технічне обслуговування без участі оператора [6].

На мою думку, ключова перевага впровадження IoT у виробництві полягає в можливості оперативно реагувати на зміни, підвищувати безпеку працівників і мінімізувати витрати на обслуговування.

Зокрема, навіть простий сценарій — автоматичне вимкнення обладнання у разі перегріву, реалізований на рівні контролера — дозволяє уникнути аварій і знизити ризики для персоналу [9]. Крім того, використання Telegram-ботів для інформування дає змогу оперативно повідомляти відповідальних осіб про критичні події незалежно від їхньої фізичної присутності на підприємстві.

1.3 Соціальні мережі як канал взаємодії у виробничих процесах

Спочатку соціальні мережі створювалися переважно як платформи для особистого спілкування, обміну новинами й розваг. Проте з поширенням цифрових технологій бізнес-середовище досить швидко виявило їхній потенціал для вирішення прикладних завдань — від маркетингу до внутрішньої комунікації та підтримки клієнтів [2].

Сьогодні соціальні мережі стали частиною корпоративної культури багатьох компаній. Їх використовують не лише у сфері послуг чи ІТ, а й у таких галузях, як промисловість, логістика й виробництво. Вони дозволяють ефективно організувати внутрішню взаємодію, зменшити час на ухвалення рішень і забезпечити швидку реакцію на зміну обставин у реальному часі.

Зокрема, системи миттєвого обміну повідомленнями дають змогу прискорити комунікаційні процеси між відділами, покращити прозорість управлінських рішень і спростити контроль за виробничими ситуаціями. Особливу роль тут відіграють сучасні месенджери — Telegram, Slack, Discord, — які поєднують зручність у користуванні, широкий функціонал і можливість інтеграції з іншими цифровими інструментами [9], [10], [11].

Використання таких платформ у бізнесі вже давно вийшло за межі неформального спілкування. Вони стали важливими інструментами для координації дій, автоматизації комунікацій і підвищення загальної ефективності роботи.

1.3.1 Інтеграція Telegram, Slack, Discord у виробничі системи

Сучасні платформи для обміну повідомленнями відіграють важливу роль у цифровізації внутрішньої комунікації на підприємствах. Завдяки відкритим API та широким можливостям інтеграції, вони активно використовуються для сповіщення, моніторингу та навіть керування виробничими процесами.

Telegram сьогодні є однією з найзручніших платформ для зв'язку з виробничими системами завдяки відкритому API та гнучким механізмам налаштування [9]. Через спеціально створених ботів Telegram дозволяє:

- отримувати автоматичні повідомлення про стан обладнання;
- надсилати команди на контролери;
- переглядати поточні дані з виробничих систем;
- виконувати базове управління процесами без використання складних інтерфейсів.

Це значно пришвидшує реакцію на критичні події та полегшує взаємодію між персоналом і технікою.

Slack часто використовується в командних проєктах із високим рівнем складності, зокрема у сфері виробництва [10]. Його перевага — в організованій структурі комунікації та можливості підключення зовнішніх систем. Основні сценарії використання:

- автоматичне отримання повідомлень від SCADA- та IoT-платформ;
- створення окремих каналів для техпідтримки, планування та обміну документами;
- реалізація системи інцидент-менеджменту, що дає змогу швидко інформувати команду про збої чи позаштатні ситуації.

Slack найкраще підходить для середніх і великих підприємств, де важлива структурованість внутрішньої взаємодії.

Discord спочатку був розроблений як платформа для спілкування в ігровому середовищі, його функціональність добре себе зарекомендувала і в виробничій сфері [11]. Серед ключових переваг:

- можливість проводити голосові наради між віддаленими командами в реальному часі;
- створення тематичних текстових каналів для швидкого обміну інформацією;
- підтримка ботів, які можуть надсилати сповіщення про стан систем або події на об'єкті.

Цей інструмент особливо привабливий для невеликих підприємств і виробничих стартапів, які шукають безкоштовне, але гнучке рішення для внутрішньої комунікації.

1.3.2 Переваги використання соціальних мереж у промисловості

Інтеграція соціальних мереж у виробниче середовище відкрила нові можливості для комунікації, керування процесами та підвищення загальної ефективності роботи. У порівнянні з традиційними корпоративними засобами зв'язку, такі платформи мають низку очевидних переваг, що роблять їх цінним інструментом для підприємств будь-якого масштабу:

1) **оперативність комунікації**. Однією з головних переваг є миттєва передача інформації. Повідомлення про зміну статусу обладнання, планові завдання чи аварійні ситуації можуть одночасно надходити всім зацікавленим особам — від операторів і техніків до керівників змін [9]. Це значно скорочує час реагування та сприяє швидшому ухваленню рішень;

2) **висока доступність**. З огляду на те, що більшість працівників мають смартфони з підключенням до Інтернету, впровадження таких рішень не потребує додаткових витрат на обладнання [10]. Telegram, Slack чи Discord легко інтегруються у повсякденну роботу без складних процедур впровадження, що підвищує залученість персоналу;

3) простота інтеграції з виробничими системами. Месенджери мають відкриті API, що дозволяє підключати їх до SCADA, IoT-платформ і промислових контролерів [9]. Наприклад, через Telegram-бота можна отримати сповіщення від обладнання, передати команду на зупинку агрегата або переглянути ключові показники. Це спрощує моніторинг без потреби в складних користувацьких інтерфейсах;

4) зниження витрат. На відміну від створення власних корпоративних систем комунікації, використання соціальних мереж — економічно обґрунтоване рішення [11]. Базові функції більшості платформ є безкоштовними, а розробка простого бота для Telegram чи налаштування інтеграцій у Slack обходиться значно дешевше, ніж створення окремої системи «з нуля»;

5) гнучкість і масштабованість. Такі системи легко адаптуються до конкретних умов роботи підприємства. Створення нових чатів, каналів або ботів не потребує спеціального програмного забезпечення чи змін у технічній інфраструктурі [9]. Це дає змогу швидко реагувати на зміни в організаційній структурі або виробничих процесах.

Інтеграція соціальних мереж у виробничі процеси дозволяє організувати комунікацію нового типу — динамічну, прозору та орієнтовану на користувача. Якщо раніше контроль за станом обладнання здійснювався лише зі стаціонарних робочих місць, сьогодні достатньо смартфона або планшета, щоб своєчасно отримати сповіщення й оперативно відреагувати на ситуацію. Наприклад, у випадку перегріву установки, оператор, перебуваючи навіть поза межами цеху, може отримати повідомлення в Telegram, оцінити ситуацію на основі надісланих даних і, за потреби, дистанційно вимкнути агрегат або повідомити чергового інженера [9].

Ще одна важлива перевага — мінімальний поріг входу для користувачів. Персонал не потребує тривалого навчання, оскільки вже має досвід користування месенджерами в повсякденному житті. Це знижує кількість помилок і сприяє швидшому впровадженню нових технологій.

Крім того, відкриті API та доступність документації дозволяють створити кастомні рішення з мінімальними ресурсами. Наприклад, невелике підприємство може впровадити автоматизовану систему сповіщення про аварії, витративши на це лише кілька робочих днів розробника.

У поєднанні з IoT-технологіями соціальні мережі стають частиною інтелектуальних розподілених систем, де сенсори та контролери передають дані в реальному часі, а користувач — бере активну участь у процесі керування. Такий підхід поєднує складність технічного рішення з простотою взаємодії, зрозумілою навіть користувачам без технічного бекграунду [10]. На мою думку, саме така модель комунікації відповідає викликам сучасного виробництва: вона гнучка, масштабована, проста у використанні та ефективна з погляду витрат.

1.4 Технічні рішення та платформи для інтеграції IoT і соціальних мереж

1.4.1. Протоколи обміну даними: MQTT, HTTP, WebSocket

Надійний обмін даними — ключовий елемент ефективної роботи IoT-систем у промисловості. Серед найбільш поширених протоколів, які забезпечують стабільну взаємодію між пристроями, виділяють наступні:

1) MQTT (Message Queuing Telemetry Transport). Це легкий протокол, розроблений для систем з обмеженою пропускнуою здатністю або ресурсами. Його основна перевага — мінімальне навантаження на мережу та використання моделі "публікація–підписка", яка особливо ефективна для сенсорних мереж та систем із великою кількістю вузлів [8];

2) HTTP/HTTPS. Протокол HTTP залишається універсальним стандартом для передавання даних у веб-середовищі. У контексті IoT він використовується переважно для обміну інформацією між пристроями та хмарними платформами або для взаємодії з веб-інтерфейсами. HTTPS забезпечує додатковий рівень безпеки при передачі даних [4];

3) WebSocket. На відміну від HTTP, протокол WebSocket підтримує постійне двостороннє з'єднання між клієнтом і сервером. Це дозволяє передавати дані практично без затримок, що критично для застосувань, де важлива миттєва реакція системи (наприклад, при моніторингу стану обладнання в реальному часі) [7].

1.4.2 Можливості API Telegram для автоматизації

Telegram надає розробникам потужний інструментарій для інтеграції виробничих процесів через Telegram Bot API. Зокрема, можливості включають [9]:

- прийом повідомлень від IoT-пристроїв у чаті;
- передавання команд у відповідь — безпосередньо на контролери або сервери;
- інтерактивне управління обладнанням через кнопки в повідомленнях;
- надсилання логів, фото або відео у вигляді файлів.

Відкритість API, наявність документації та велика спільнота роблять Telegram привабливим варіантом для швидкої та недорогої автоматизації без необхідності створення спеціалізованого ПЗ.

Приклади існуючих інтеграцій:

1) моніторинг обладнання через Telegram-бота. IoT-сенсори передають дані через MQTT-брокер, а скрипт (наприклад, на Python) публікує повідомлення у Telegram у разі досягнення критичних параметрів [9];

2) оповіщення про збої у Slack. Після фіксації помилки система надсилає сповіщення у визначений канал Slack, щоб технічний персонал міг швидко відреагувати [10];

3) внутрішня комунікація в Discord. Використання голосових та текстових каналів у Discord дозволяє майстрам змін координувати дії без прив'язки до стаціонарних телефонів або внутрішньої мережі [11].

На мою думку, поєднання легких протоколів (таких як MQTT і WebSocket) із можливостями сучасних месенджерів дає змогу створювати гнучкі, масштабовані та доступні рішення для виробничої автоматизації. Вони особливо корисні для малих і середніх підприємств, яким потрібні ефективні інструменти без значних капіталовкладень [8], [9], [10].

1.5 Патентний огляд

Розробка нових технічних рішень вимагає не лише аналізу наукових джерел, але й глибокого вивчення патентного ландшафту — сукупності вже зареєстрованих об'єктів інтелектуальної власності, що можуть бути технічно близькими або функціонально подібними до пропонованого підходу. Такий огляд дозволяє оцінити рівень новизни розробки, уникнути дублювання існуючих рішень і знизити ризик порушення прав третіх осіб.

У межах даної роботи проведено тематичний огляд патентної інформації, що охоплює такі напрямки:

- автоматизація виробничих процесів із використанням IoT-технологій;
- застосування мобільних додатків і месенджерів (зокрема Telegram) як інтерфейсів управління;
- цифрові рішення для процесу зварювання ПВХ-профілю у виробництві металопластикових конструкцій;
- інтеграція сенсорів, контролерів, протоколів передачі даних і хмарних платформ у промислових системах.

Огляд сформований на відкритих міжнародних та національних патентних базах, зокрема: **Espacenet**, **Google Patents**, **USPTO**, **WIPO**, **Укрпатент**, а також патентних реєстрів Китаю та країн Європейського Союзу.

1.5.1 Аналіз релевантних патентів

У таблиці 1.1 представлено добірку патентів, які мають спільні технічні характеристики з темою дослідження або охоплюють ключові компоненти розроблюваної системи.

Таблиця 1.1

Патенти

№	Номер патенту	Назва (скорочено)	Рік	Ключова ідея
1	US 10965569 B2	Система моніторингу IoT-каналів на основі MQTT (рис. 1.1)	2021	Використання MQTT як надійного транспортного протоколу для передачі промислових даних
2	US 9826338 B2	IoT для предиктивного обслуговування (рис. 1.2)	2017	Використання бездротових сенсорів для контролю стану обладнання та прогнозування відмов
3	US 10270864 B2	Хмарний бот-сервер для месенджерів (Telegram, WhatsApp тощо) (рис. 1.3)	2019	Інтерфейс управління системами через Telegram-бот із вбудованою логікою.

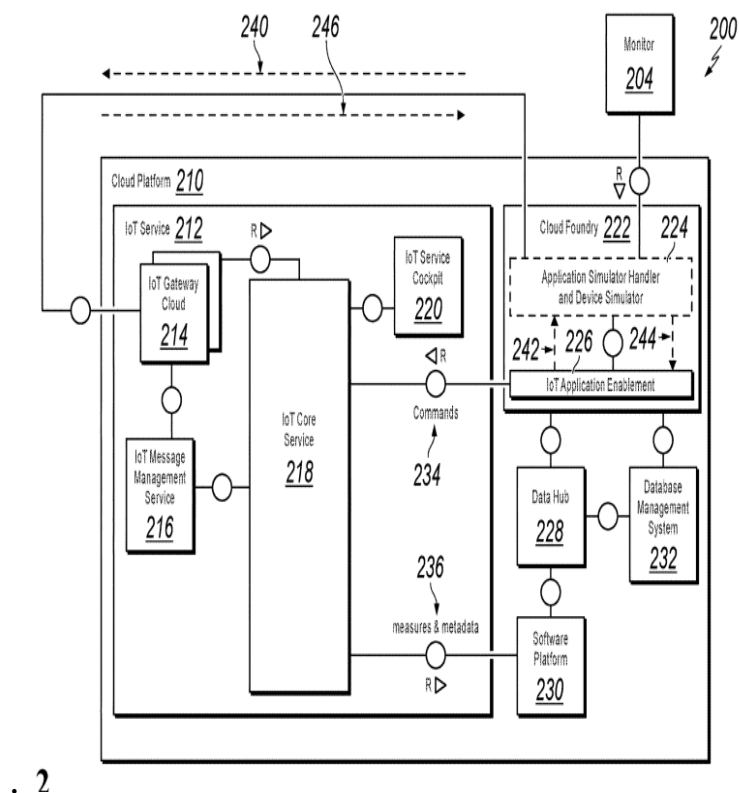


Рисунок 1.1 – Система моніторингу IoT-каналів на основі MQTT

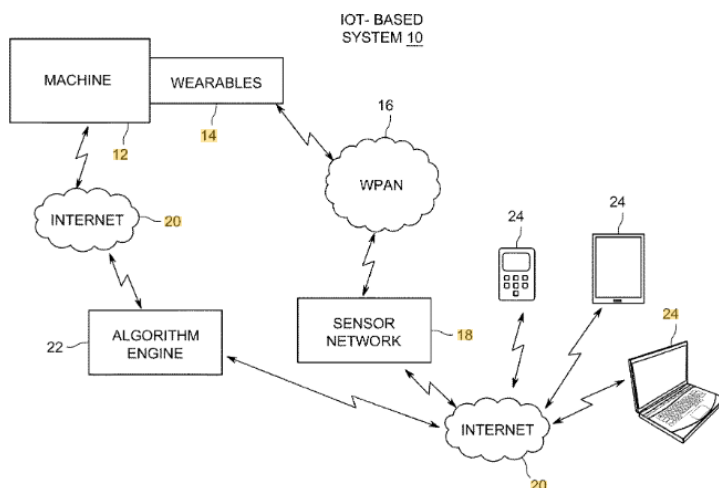


Рисунок 1.2 – IoT для предиктивного обслуговування

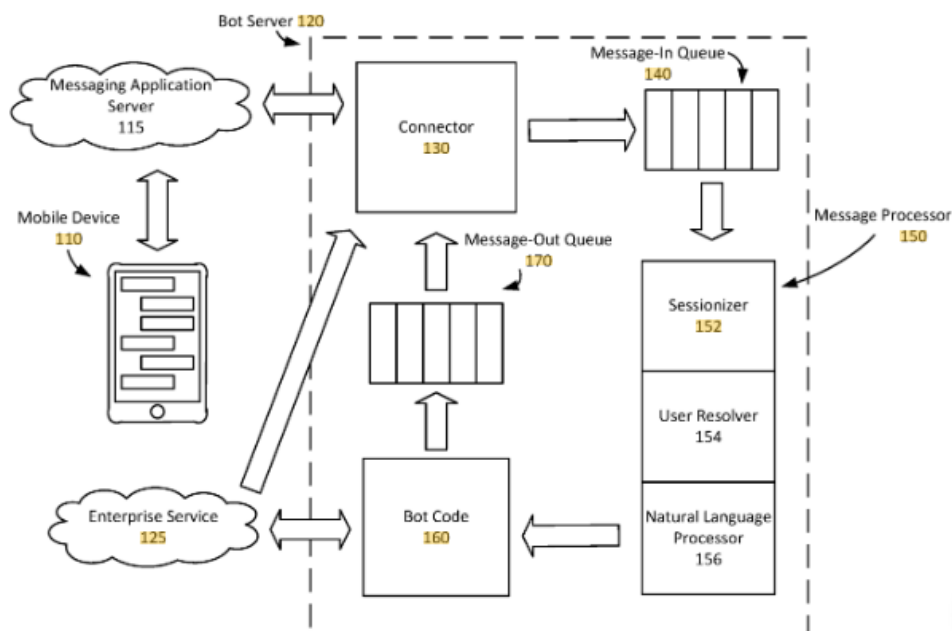


Рисунок 1.3 – Хмарний бот-сервер для месенджерів

1.5.2 Основні висновки патентного огляду

Аналіз патентної бази показує, що IoT-системи для промислового використання вже широко представлені у сфері збору, передавання даних і прогнозування технічного стану обладнання. Більшість рішень розроблено великими компаніями, такими як Siemens, Bosch, Schneider, Oracle. Це ускладнює їх впровадження в умовах малих підприємств через високу вартість і складність інтеграції.

Використання месенджерів як інтерфейсів взаємодії з користувачем (НМІ) має відповідне правове підґрунтя. Наприклад, патент US 10270864 B2, виданий компанії Oracle, підтверджує можливість використання месенджерів для керування системами в режимі реального часу. Це створює технічне та правове підґрунтя для інтеграції Telegram-ботів у промислові IoT-рішення.

У контексті виробництва металопластикових вікон, а саме процесу зварювання ПВХ-профілів, більшість патентів стосуються конструктивно-механічних удосконалень. Наприклад, у патенті US 9993969 B2 описано технічні рішення з механіки зварювального процесу, але відсутні компоненти цифрового моніторингу. Це створює потенціал для впровадження інноваційних сенсорних систем контролю температури, тиску та часу зварювання.

Поточний стан українського та європейського патентного поля не виявив рішень, які одночасно охоплювали б:

- використання доступних за ціною мікроконтролерів, зокрема ESP32;
- передавання даних за допомогою протоколу MQTT;
- моніторинг із використанням інтерфейсу Telegram;
- орієнтацію на потреби малих виробничих підприємств, зокрема віконних цехів.

Це вказує на наявність вільної інноваційної ніші для створення IoT-рішень, адаптованих до умов малого бізнесу.

Висновки до розділу 1

У першому розділі проведено комплексний аналітичний огляд технічної та патентної інформації щодо сучасного стану й тенденцій розвитку автоматизації виробничих процесів із використанням технологій Інтернету речей (IoT) та засобів комунікації на базі соціальних мереж.

Розглянуто основні етапи еволюції автоматизації — від механічних пристроїв першої індустріальної революції до сучасних кіберфізичних систем, що стали основою концепції Індустрії 4.0. Проаналізовано традиційні підходи до

автоматизації на базі SCADA, MES і ERP-систем. Попри їхню функціональну потужність, такі рішення мають істотні обмеження щодо вартості впровадження, складності технічного обслуговування та гнучкості адаптації, особливо у сфері малого та середнього бізнесу.

Показано, що технології Інтернету речей формують нову парадигму автоматизації завдяки децентралізованій архітектурі, високій масштабованості та адаптивності. Розглянуто загальну архітектуру IoT-систем, функціональні рівні обробки даних і приклади їх практичного застосування у виробничому середовищі.

Окрему увагу я приділив ролі соціальних платформ у промисловості, насамперед використанню месенджерів Telegram, Slack і Discord як альтернативних НМІ-рішень. Встановлено, що інтеграція IoT-пристроїв із цими платформами сприяє підвищенню оперативності реагування, зниженню витрат на реалізацію систем моніторингу та спрощенню доступу персоналу до ключової інформації.

Мною детально проаналізовано технічні підходи до таких інтеграцій, зокрема використання протоколів MQTT, HTTP, WebSocket та API Telegram. Наведено приклади ефективного застосування цих рішень на практиці.

Патентний огляд підтвердив мої припущення, що попри активну розробку IoT-рішень у промисловому секторі, напрям інтеграції з месенджерами залишається порівняно новим і малодослідженим. Виявлено перспективу для створення інноваційних продуктів, зокрема орієнтованих на потреби невеликих підприємств.

Результати аналітичного та патентного аналізу свідчать про доцільність розробки нових підходів до автоматизації виробництва, заснованих на поєднанні IoT-технологій із сучасними цифровими каналами комунікації. Такий підхід відповідає актуальним запитам промисловості та відкриває нові можливості для підвищення ефективності й доступності автоматизованих систем.

2 ОБҐРУНТУВАННЯ НАПРЯМУ ДОСЛІДЖЕНЬ ТА РОЗРОБКА МЕТОДИКИ ВПРОВАДЖЕННЯ ІОТ-СИСТЕМИ АВТОМАТИЗАЦІЇ ВИРОБНИЦТВА

2.1 Обґрунтування вибору напрямку дослідження

У сучасних умовах конкурентного ринку виробництво металопластикових вікон зазнає істотних трансформацій, що зумовлені переходом до цифрової економіки, зростанням вимог до якості продукції та скороченням її життєвого циклу. Одним із ключових викликів для підприємств цього профілю є необхідність мінімізації впливу людського фактора, раціонального використання ресурсів і стабілізації якості без зростання операційних витрат.

За таких умов виникає об'єктивна потреба у впровадженні адаптивної системи управління, яка забезпечує не лише моніторинг технологічних параметрів, а й оперативне реагування на зміни у виробничому процесі. Мною обрано цей напрям дослідження, щоб створити доступну альтернативу традиційним системам автоматизації (на кшталт SCADA чи DCS), які через високу вартість і складність реалізації залишаються недосяжними для більшості малих і середніх підприємств.

Сучасні централізовані рішення мають низку обмежень:

- прив'язаність до комерційного програмного забезпечення та обмежена гнучкість у модифікації;
- потреба у кваліфікованих спеціалістах для налаштування й супроводу;
- тривалі строки впровадження;
- висока загальна вартість володіння (ТСО).

На противагу цьому, технології Інтернету речей (IoT) пропонують децентралізований підхід до автоматизації, що передбачає локальну обробку даних за допомогою мікроконтролерів із вбудованими сенсорами та передавання інформації на вищій рівні управління через легкі протоколи зв'язку. Вибір на користь IoT зумовлений рядом переваг:

1) масштабованість архітектури — можливість поетапного впровадження без зупинки чи суттєвого втручання у діючі виробничі процеси;

2) низький поріг входу — доступність апаратних засобів (наприклад, ESP32) і відкритого програмного забезпечення (MQTT-брокери, Telegram Bot API) дає змогу реалізувати проєкт силами невеликої команди без значних інфраструктурних витрат;

3) мобільність і віддалений доступ — можливість керування та моніторингу з будь-якого місця за допомогою смартфона та месенджера Telegram;

4) інтеграція з наявним середовищем — система функціонує паралельно з чинними технологічними процесами, зчитуючи параметри температури, тиску чи тривалості циклу без втручання у заводське програмне забезпечення.

Об'єктом дослідження є усі ділянки з виготовлення металопластикових вікон, адже ці етапи мають критичний вплив на якість готової продукції. Порушення температурного чи тискового режимів навіть у межах кількох відсотків може призвести до браку, водночас більшість підприємств покладаються на ручний або візуальний контроль, що не відповідає сучасним стандартам.

Метою розроблюваної системи є:

- автоматичний контроль параметрів на всіх ділянках в реальному часі;
- фіксація подій та відхилень, формування логів;
- повідомлення відповідального оператора про критичні стани через Telegram;
- створення гнучкої платформи, придатної для масштабування на інші ділянки (порізка, зачистка, скління тощо).

У межах проєкту розглядається комплексна система, що складається з:

- сенсорної частини (температурні, тактильні, вібраційні датчики);
- керувального модуля на базі ESP32/Arduino;
- протоколу MQTT для швидкого передавання повідомлень;

— Telegram-бота з розподілом доступу та інтерактивним меню.

Вибір напрямку дослідження обґрунтований сукупністю технічних, економічних та організаційних чинників. Ключові з них:

- невідповідність типових SCADA-рішень умовам малого виробництва;
- доступність необхідних апаратних і програмних засобів для реалізації IoT-моделі;
- потреба у простому, мобільному й безпечному інтерфейсі управління, реалізованому на основі месенджера Telegram.

2.2 Методика аналізу системи автоматизації виробничих процесів

Побудова ефективної системи автоматизації потребує не лише технічного проєктування, а й ретельного попереднього аналізу об'єкта керування. На цьому етапі особливу увагу приділено дослідженню структури виробничого процесу, виявленню його динамічних характеристик, типових вхідних і вихідних параметрів, внутрішніх зв'язків між окремими технологічними операціями, а також можливих обмежень і зон невизначеності. Такий підхід ґрунтується на принципах системного аналізу, який становить методологічну основу сучасної інженерії складних технічних систем.

У межах цього дослідження застосовано багаторівневу методику аналізу, що охоплює кілька етапів з поступовою деталізацією функціональних, структурних та інформаційних аспектів об'єкта автоматизації.

2.2.1 Функціональний аналіз об'єкта автоматизації

На початковому етапі дослідження мною побудовано функціональну модель виробничого процесу виготовлення металопластикових вікон (рис. 2.1). Процес розглянуто як сукупність взаємопов'язаних операцій, що охоплюють увесь цикл — від постачання матеріалів до випуску готової продукції. Виокремлено основні етапи: розкрій профілю, монтаж армувального елемента,

термозварювання, механічна зачистка, встановлення склопакету, контроль якості та остаточне пакування.

Для кожної технологічної стадії визначено ключові складові:

- вхідні впливи (наприклад, замовлення, параметри розкрою);
- технологічні обмеження (наприклад, температурний режим зварювання);
- вихідні результати (готові вироби, кількість браку, швидкість виконання).

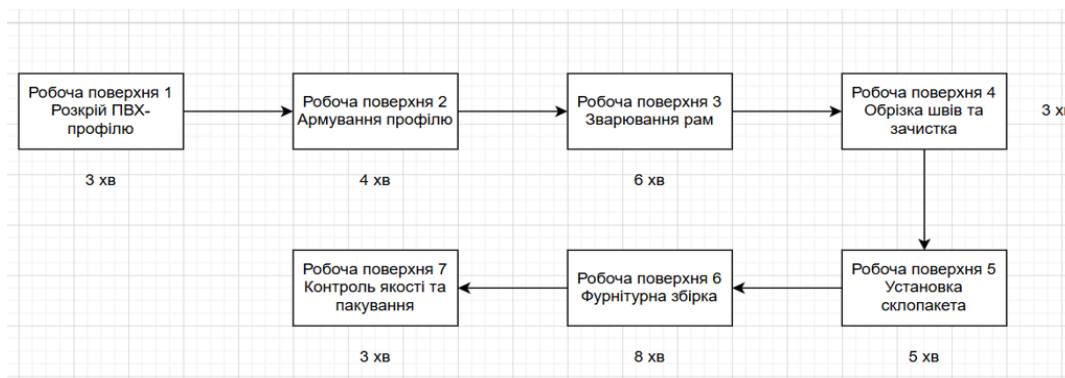


Рисунок 2.1 – Технологія виробництва металопластикових вікон

2.2.2 Структурно-логічне представлення системи

На другому етапі система розглядається як сукупність окремих компонентів, що функціонують у взаємозв'язку (рис. 2.2). Такий підхід дозволяє перейти від опису процесу як цілісного явища до формалізованого представлення його структури, в якому чітко окреслено елементи, їхні функції та типи взаємодії. Кожен компонент виконує конкретне завдання: фіксує параметри, передає інформацію, приймає рішення або безпосередньо впливає на технологічний процес.

Зокрема, для зварювальної ділянки визначаються:

- параметри, які потрібно моніторити: температура нагріву, час циклу, тиск пресування;
- вузли, що генерують дані: термопари, тензодатчики, цифрові таймери;
- виконавчі пристрої: регулятори, блоки аварійного відключення.

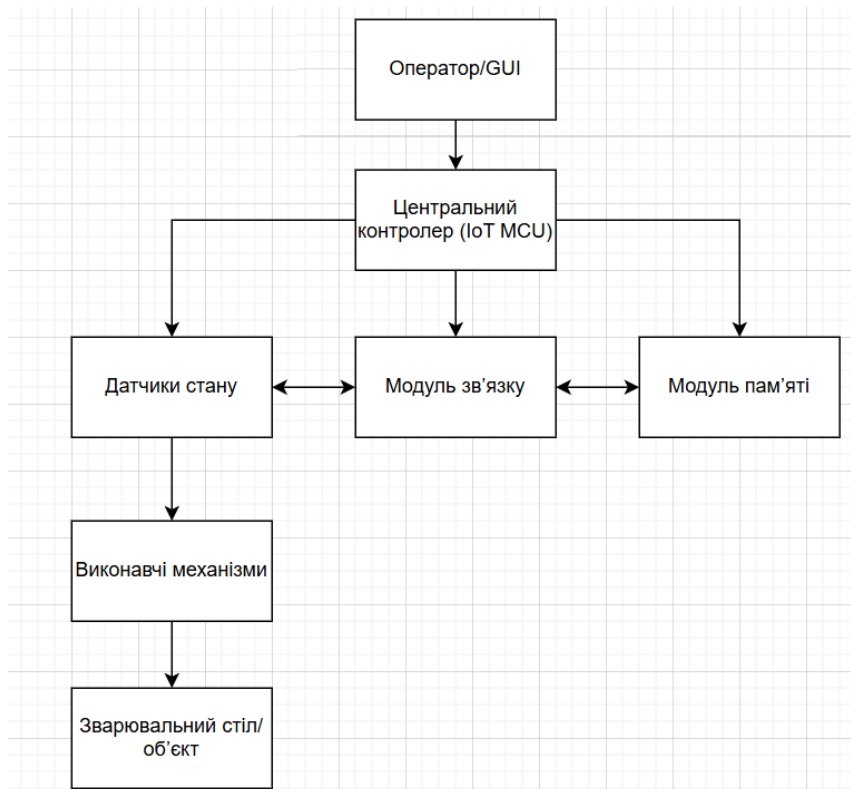


Рисунок 2.2 – Структурна логіка системи

2.2.3 Інформаційно-параметричний аналіз

На цьому етапі дослідження проводиться інформаційно-параметричний аналіз, мета якого — визначити обсяг, структуру та характеристики даних, необхідних для ефективної роботи системи автоматизації. Встановлюються такі ключові параметри:

- обсяг інформації, що підлягає збору, збереженню та подальшій обробці;
- вимоги до точності вимірювань і частоти оновлення показників;
- логіка обробки отриманих даних, а також принципи формування керувальних рішень на їх основі.

Окрему увагу приділено класифікації параметрів за типом зміни. Всі контрольовані величини поділяються на: статичні та динамічні. У таблиці 2.1 наведена класифікація ключових параметрів.

Таблиця 2.1

Класифікація параметрів за типом зміни

№	Назва параметра	Тип параметра	Джерело даних	Частота вимірювання	Значущість для управління
1	Температура зварювання	Динамічний	Датчик температури	Щосекундно	Критична
2	Струм зварювання	Динамічний	Датчик струму	10 разів/сек	Висока
3	Напруга на електроді	Динамічний	Датчик напруги	10 разів/сек	Висока
4	Положення маніпулятора	Динамічний	Енкодер / IMU	50 разів/сек	Критична
5	Конфігурація виробу	Статичний	Введення оператором	Один раз на зміну	Висока
6	Тип матеріалу	Статичний	Інформаційна система	Один раз на партію	Середня
7	Кількість деталей	Статичний	Оператор / ERP-система	Перед запуском серії	Середня
8	Вологість середовища	Динамічний	Датчик вологості	1 раз/хвилину	Низька
9	Помилки/збої в роботі	Подієвий (динамічний)	Система контролю	У момент події	Критична
10	Ідентифікатор зміни	Статичний	Сканер / ввід вручну	Один раз на зміну	Низька

2.2.4 Аналіз сценаріїв роботи системи

За методикою передбачено моделювання **типових сценаріїв роботи обладнання:**

- нормальна експлуатація;
- відхилення параметрів за межі норми;
- аварійне зупинення;
- зміна режимів;
- повторне включення після збоїв.

Для кожного сценарію передбачається відповідна логіка обробки подій та відповідь з боку системи — як автоматично, так і через взаємодію з оператором через Telegram-бота. На рисунку 2.3 зображено дерево сценаріїв роботи системи на випадок відхилень параметрів.

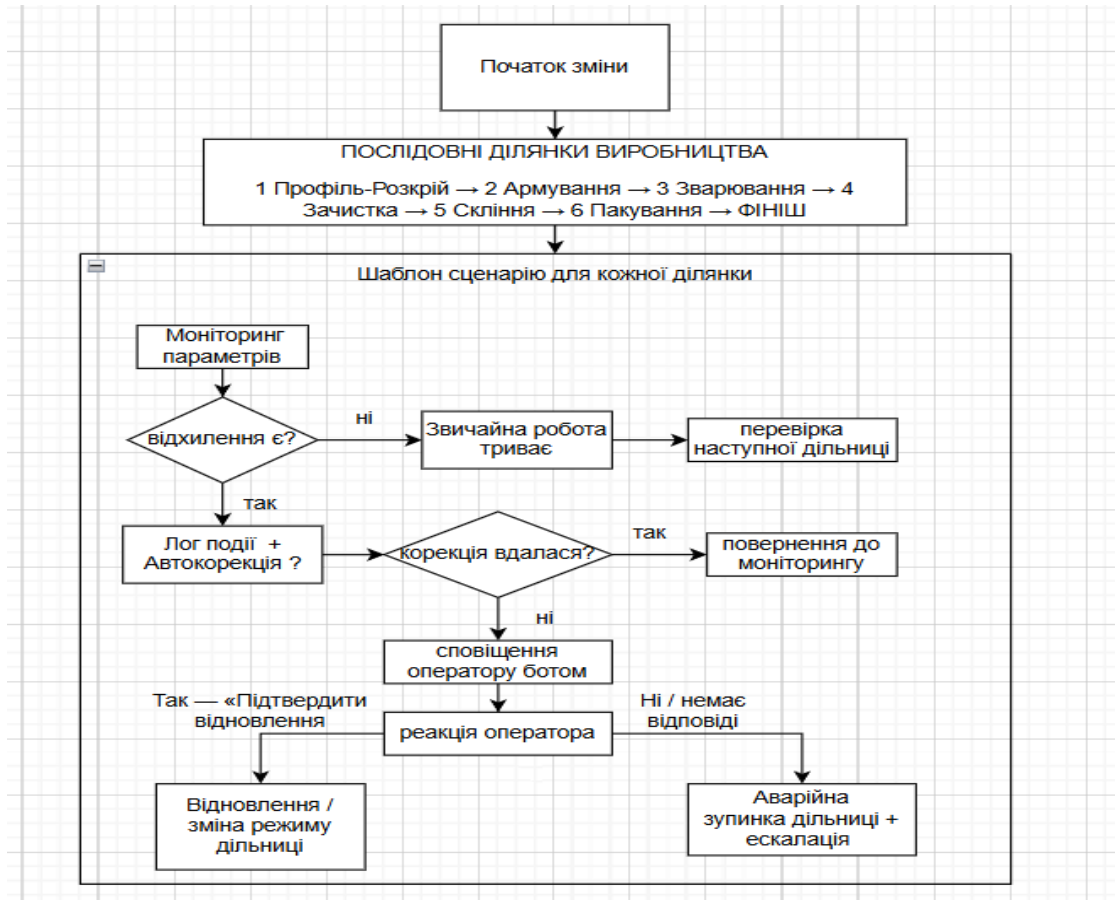


Рисунок 2.3 – Дерево сценаріїв на випадок відхилень параметрів

2.2.5. Інтерфейсна модель: інтеграція з користувачем

Останнім етапом стало вивчення того, як користувач взаємодіє з системою. Основну увагу зосереджено на Telegram, який тут виступає як інструмент для візуалізації даних, моніторингу й керування процесами. Аналіз охоплює, зокрема, те, як побудоване меню бота, які саме повідомлення отримує користувач — від звичайного тексту до графіки чи кнопок — і як організовано зворотний зв'язок між оператором і системою. Це рішення не тільки спрощує контроль, а й дає змогу швидко реагувати на зміни, що є пріоритетним в умовах виробничої нестабільності. Приклад сповіщення про надзвичайну ситуацію на підприємстві з допомогою Telegram-бота зображено на рисунку 2.4.

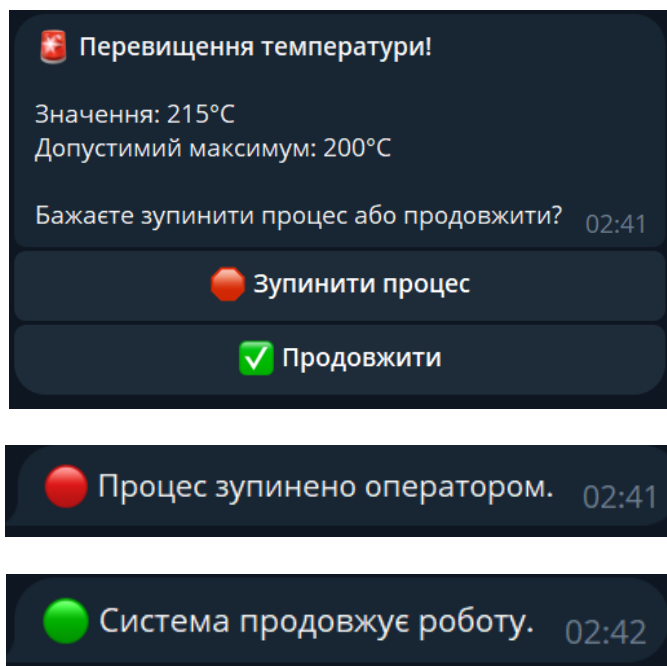


Рисунок 2.4 – Приклад сповіщення про перевищення температури

2.3 Методологія синтезу автоматизованої системи

Після дослідження виробничого середовища, аналізу структур і потоків даних та способів взаємодії з системою приходимо до етапу — створення автоматизованої системи для контролю параметрів у процесі фарбування металопластикових вікон. Тут потрібно чітко сформулювати завдання: які саме дані збирати, як обробляти їх, із чого складатиметься система — і яке устаткування знадобиться.

Головна мета — забезпечити відкриту, гнучку систему, що складається з окремих модулів і не залежить від централізованої інфраструктури. Це означає: впроваджуємо поступово й додаємо компоненти, не зупиняючи виробничий процес. Система має збирати найважливіші параметри — температуру, тиск, тривалість зварювання — і передавати їх в хмару. Відповідь системи є автоматичною: сповіщення, запис в журнал або в базу даних. Загалом — оператор отримує всі дані й керує через зрозумілий інтерфейс.

Найчутливіший об'єкт для впровадження — зварювальний вузол. Це критична точка, де будь-яка зміна режиму може викликати проблеми, і саме її

часто не контролюють цифровими засобами. Система має працювати автономно, але готова до інтеграції в загальну виробничу мережу.

2.3.1 Етапи синтезу автоматизованої системи

Перший етап – **формування концептуальної архітектури системи.**

Загальна побудова системи включає кілька рівнів, кожен з яких виконує свою функцію:

1) **польовий рівень** — сенсори, які реєструють параметри процесу в режимі реального часу. Наприклад, термопара типу К вимірює температуру зварювання, тензодатчик фіксує зусилля стискання, а цифровий таймер — тривалість кожного циклу;

2) **керуючий рівень** — контролер, як-от ESP32 або Arduino Uno з Wi-Fi-модулем. Він відповідає за зчитування даних із сенсорів, їх попередню обробку та формування повідомлень у форматі MQTT;

3) **транспортний рівень** — передавання інформації відбувається через MQTT, легкий протокол, який забезпечує швидку й надійну доставку повідомлень у середовищі TCP/IP;

4) **інтерфейсний рівень** — Telegram-бот, який приймає повідомлення, обробляє запити оператора, дає змогу виконувати базові дії (наприклад, /status, /reset, /warn) і зберігає всі звернення в журналі;

5) **рівень зберігання** — окрема система для резервного запису параметрів. Це може бути файлова структура на сервері або база даних. У подальшому передбачено інтеграцію з MySQL, Google Sheets API або Firebase.

Другий – **розробка функціональної схеми системи.**

Функціональна схема створюється з урахуванням типових ситуацій під час роботи: нормальний режим, відхилення параметрів і аварійне завершення циклу. У системі передбачено кілька основних частин. Перша — збирає дані з сенсорів. Друга — перевіряє, чи відповідають ці значення заданим межам. Третя

відповідає за повідомлення та журналювання. Окремий блок працює з командами, які надходять через Telegram.

На практиці це реалізовано як простий цикл. Сенсори періодично опитуються, дані аналізуються, відхилення фіксуються, надсилаються повідомлення через MQTT, а також постійно перевіряється, чи немає запитів від користувача.

Третій етап – **визначення технічних засобів для реалізації.**

Апаратна платформа визначалася за наступними критеріями:

- доступність на ринку;
- мінімальна енергоспоживаність;
- наявність вбудованих інтерфейсів Wi-Fi;
- підтримка відкритих бібліотек для роботи з MQTT, Telegram та датчиками.

На основі цих даних обрано ESP32, як більш продуктивну альтернативу Arduino Uno, яка має інтегрований модуль Wi-Fi, багатоканальні входи/виходи і можливість паралельного виконання завдань (FreeRTOS) (рис. 2.5).

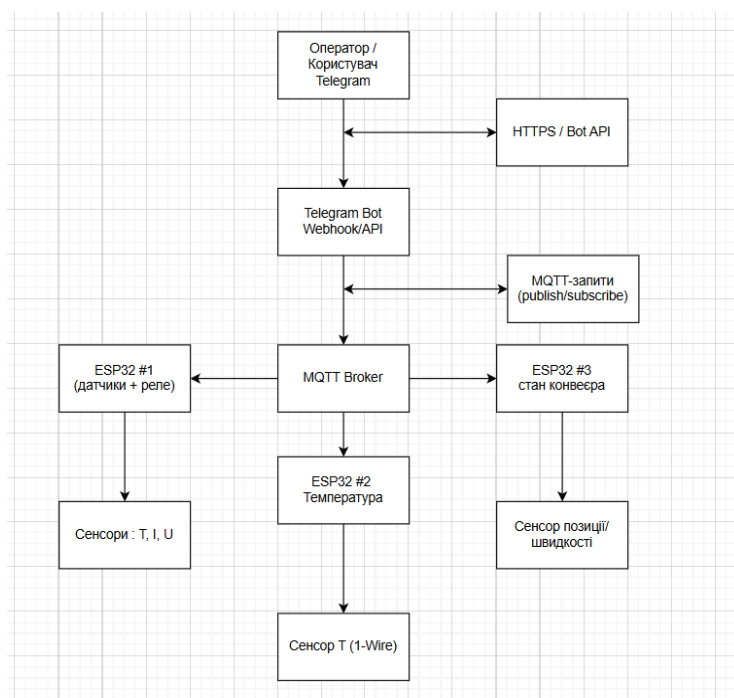


Рисунок 2.5 – Технічна архітектура системи автоматизації на базі ESP32, MQTT і Telegram.

Для вимірювання параметрів температури — термopapa типу K + модуль MAX6675, для тиску — аналоговий тензодатчик з підсилювачем HX711, для таймінгу — внутрішній таймер мікроконтролера або зовнішній цифровий тригер.

Четвертий етап – **синтез інформаційної моделі.**

Інформаційна модель передбачає подання даних у вигляді MQTT-запитів які виглядають наступним чином (рис. 2.6):

```
{  
  "device_id": "WELD_01",  
  "timestamp": "2025-06-10T14:25:00Z",  
  "temperature": 233.7,  
  "pressure": 15.2,  
  "duration": 7.1,  
  "status": "OK"  
}
```

Рисунок 2.6 – Приклад MQTT запита

Повідомлення відправляється на тему `iot/welding/control`, далі обробляється сервером і дублюється до Telegram-бота для інформування відповідального оператора (визначається за id користувача). У разі перевищення допустимих значень параметрів, бот надішле сповіщення з відповідною поміткою ("Попередження: температура перевищила поріг!").

П'ятий етап – **формування логіки Telegram-бота.**

Telegram виступає у ролі НМІ (Human Machine Interface). Взаємодія виконується через використання Bot API, з обробкою команд `/start`, `/status`, `/report`, а також за допомогою бот-меню, яке дозволяє передивлятися роботу конкретних ділянок, виставляти частоту сповіщень, передивлятися звіти та виконувати інші контрольні дії.

Сама логіка реалізується на мові Python (бібліотека python-telegram-bot) або Node.js (node-telegram-bot-api) — вибір залежить від платформи хостингу. Вся взаємодія здійснюється через webhook, що підключений до MQTT-сервера або проміжного логічного блоку (рис. 2.7).

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>

// ----- Wi-Fi -----
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

// ----- MQTT -----
const char* mqtt_server = "broker.hivemq.com"; // Або локальний/EMQX
const char* topic = "iot/window_factory/welding/temp";

WiFiClient espClient;
PubSubClient client(espClient);

// ----- Температурний сенсор DS18B20 -----
#define ONE_WIRE_BUS 4 // нін підключення
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// ----- Підключення Wi-Fi -----
void setup_wifi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}

// ----- Початок роботи -----
void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  sensors.begin();
}

// ----- Головний цикл -----
void loop() {
  if (!client.connected()) {
    while (!client.connected()) {
      client.connect("ESP32_Welding");
    }
  }
  client.loop();

  // Зчитування температури
  sensors.requestTemperatures();
  float tempC = sensors.getTempCByIndex(0);

  // Формування JSON-повідомлення
  String payload = "{\"device\":\"zvaryvannya\",\"temperature\":\"" + String(tempC, 2) + "\"}";

  // Публікація MQTT-повідомлення
  client.publish(topic, payload.c_str());

  delay(5000); // кожні 5 секунд
}
```

Рисунок 2.7 – Фрагмент обробки та публікації MQTT повідомлень

Шостий етап – **розробка схем інтеграції системи.**

Для майбутнього масштабування потрібно передбачити:

- можливість підключення нових датчиків без переконфігурації основного контролера;
- масштабування на інші ділянки за допомогою мультиплікації вузлів;
- резервування з'єднання на випадок втрати інтернету;
- логування подій незалежно від наявності з'єднання з Telegram.

Усе це реалізується за допомогою модульної схеми з блоками plug-and-play.

Сьомий етап – **візуалізація архітектури системи.**

Узагальнена архітектура системи має вигляд схеми з шістьма вертикальними рівнями: сенсорика → контролер → мережа → обробка → інтерфейс → обладнання (рис. 2.8). Кожен елемент вказує на відповідні фізичні або програмні компоненти.

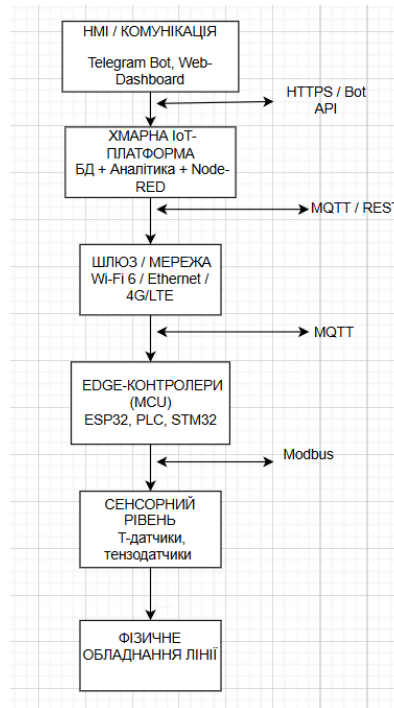


Рисунок 2.8 – Концептуальна модель автоматизованої IoT системи у виробництві вікон

Розроблена система інтегрує сенсорні модулі, бездротовий контролер, брокер MQTT, інтерфейс Telegram і локальне логування в єдиний адаптивний комплекс, який відповідає наступним критеріям: мобільність, гнучкість, масштабованість і практична ефективність для потреб малих і середніх виробничих підприємств.

2.4 Використання комп'ютерних технологій та цифрових інструментів

Розробка та впровадження автоматизованої системи на основі IoT-технологій неможливі без активного використання комп'ютерних засобів і спеціалізованого ПЗ. Основна функція комп'ютерної підтримки полягає не лише в організації взаємодії між окремими компонентами системи, а й у реалізації візуалізації даних, програмуванні логіки роботи контролерів, розгортанні комунікаційного середовища та забезпечення інтерфейса для користувача.

Для створення обраної архітектури знадобиться : сенсори → контролер → мережевий інтерфейс → користувач — використано кілька категорій цифрових інструментів, кожна з яких виконує чітко визначену функцію.

2.4.1 Програмування мікроконтролера (ESP32)

Управління збором даних та їх передачею виконується за допомогою мікроконтролера ESP32, його прошивка створюється у середовищі розробки Arduino IDE або PlatformIO. Програмування виконується на мові C++ із підключенням відповідних бібліотек, необхідних для забезпечення функціональності сенсорів, мережевих протоколів і периферійних модулів:

- WiFi.h — для підключення до мережі;
- PubSubClient.h — реалізація MQTT-протоколу;
- max6675.h — підтримка терморпарі через модуль MAX6675;
- HX711.h — підсилення та обробка аналогового сигналу з тензодатчика.

Функціонально прошивка виконує:

- ініціалізацію сенсорів і Wi-Fi;
- опитування датчиків з заданим інтервалом;
- перетворення фізичних величин у цифрові значення;
- публікацію повідомлень на MQTT-брокер з уніфікованим форматом JSON (рис. 2.9).

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <max6675.h>

// Параметри з'єднання Wi-Fi
const char* ssid = "YourSSID";
const char* password = "YourPassword";

// MQTT-брокер
const char* mqtt_server = "broker.hivemq.com"; // або локальний IP
const char* topic_temp = "iot/welding/temp";

// Ініціалізація клієнта
WiFiClient espClient;
PubSubClient client(espClient);

// Пінні терморпарі (MAX6675)
int soPin = 19; // MISO
int csPin = 5; // Chip Select
int sckPin = 18; // CLK

MAX6675 thermocouple(sckPin, csPin, soPin);

void setup() {
  Serial.begin(115200);

  // Підключення до Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("WiFi connected");

  // MQTT-з'єднання
  client.setServer(mqtt_server, 1883);

  // Очікування підключення MQTT
  while (!client.connected()) {
    if (client.connect("ESP32_Client_Welding")) {
      Serial.println("Connected to MQTT");
    } else {
      delay(1000);
    }
  }

  void loop() {
    // Зчитування температури з терморпарі
    double tempC = thermocouple.readCelsius();
    Serial.print("Temperature: ");
    Serial.println(tempC);

    // Публікація у топик
    char payload[32];
    snprintf(payload, sizeof(payload), "%.2f", tempC);
    client.publish(topic_temp, payload);

    delay(5000); // Опитування кожні 5 секунд
  }
}
```

Рисунок 2.9 – Опитування датчика та публікація значень у топик

2.4.2 Комунікаційний протокол: MQTT

Вибір на користь протоколу MQTT зумовлений легкістю, надійністю доставки повідомлень і широкою підтримкою в сучасних IoT-платформах. У якості брокера використовується Mosquitto — відкритий MQTT-брокер, який може бути розгорнутий як на локальному комп'ютері, так і на хмарних платформах, зокрема Heroku, DigitalOcean або на пристрої Raspberry Pi (рис. 2.10).

Основні параметри конфігурації включають використання порту 1883 (без шифрування), автентифікацію за логіном і паролем, а також організацію окремих топіків для кожного типу повідомлень (наприклад, /temp, /pressure, /status).

```
# === Основні параметри брокера ===
pid_file /var/run/mosquitto/mosquitto.pid
user mosquitto

# === Порти та протоколи ===
listener 1883
protocol mqtt

# === Дозволити локальні підключення ===
allow_anonymous false

# === Дані для автентифікації ===
password_file /etc/mosquitto/passwd
acl_file /etc/mosquitto/acl

# === Логу ===
log_dest syslog
log_dest stdout
log_type error
log_type warning
log_type notice
log_type information

# === Параметри безпеки ===
per_listener_settings true
```

```
mosquitto_passwd -c /etc/mosquitto/passwd welding_user
```

```
user welding_user
topic read iot/welding/temp
topic write iot/welding/control
```

```
pattern deny #
```

Рисунок 2.10 – Параметри mosquitto.conf

2.4.3 Реалізація Telegram-бота

Для забезпечення взаємодії з користувачем створено спеціалізований Telegram-бот з використанням офіційного Telegram Bot API. Логіка роботи побудована на мові програмування Python із застосуванням бібліотеки *python-telegram-bot*. Отримання даних через механізм *webhook* або шляхом підписки на MQTT-повідомлення, що надходять на сервер (в залежності від хостингу).

Бот розгортається на VPS-хостингу або локальному сервері з підтримкою захищеного з'єднання (HTTPS), що може бути забезпечене за допомогою *ngrok*, *Cloudflare Tunnel* або за рахунок встановлення SSL-сертифіката (рис. 2.11).

```
import paho.mqtt.client as mqtt
import telebot
import threading

# === Налаштування MQTT ===
MQTT_BROKER = 'broker.hivemq.com'
MQTT_PORT = 1883
MQTT_TOPIC = 'iot/welding/temp'

latest_temp = "N/A"

# === Telegram token ===
BOT_TOKEN = '8146345784:AAEkrCY9wKyG0nS7NAmtCgXwG1-KnXenJA'
bot = telebot.TeleBot(BOT_TOKEN)

# === MQTT-клієнт ===
def on_connect(client, userdata, flags, rc):
    print("MQTT connected with result code", rc)
    client.subscribe(MQTT_TOPIC)

def on_message(client, userdata, msg):
    global latest_temp
    latest_temp = msg.payload.decode()
    print(f"MQTT: {msg.topic} - {latest_temp}")

mqtt_client = mqtt.Client()
mqtt_client.on_connect = on_connect
mqtt_client.on_message = on_message
mqtt_client.connect(MQTT_BROKER, MQTT_PORT, 60)

# === Паралельний запуск MQTT-клієнта ===
def mqtt_loop():
    mqtt_client.loop_forever()

threading.Thread(target=mqtt_loop).start()

# === Команда Telegram-бота ===
@bot.message_handler(commands=['status'])
def send_status(message):
    bot.send_message(message.chat.id, f"Температура зварювання: {latest_temp}°C")

@bot.message_handler(commands=['help'])
def send_help(message):
    bot.send_message(message.chat.id, "/status - Показати поточну температуру з MQTT\n/help - Довідка")

print("Telegram-бот запущено")
bot.polling()
```

Рисунок 2.11 – Фрагмент коду для запуску бота та обробки запитів

Функціональні можливості бота охоплюють:

- обробку команд користувача (/status, /report, /reset);
- автоматичне надсилання сповіщень у разі перевищення встановлених порогових значень;

- формування коротких звітів у форматі текстових повідомлень;
- ведення журналу команд оператора із фіксацією часових міток.

2.4.4 Інструменти обробки та візуалізації даних

Для збереження та візуалізації даних використовуються наступні платформи:

- Google Sheets API — зручний інструмент для оперативного формування табличних звітів;
- InfluxDB у поєднанні з Grafana — рішення для довготривалого зберігання даних і побудови графіків зміни параметрів у реальному часі;
- Node-RED — графічне середовище для обробки MQTT-потоків, побудови логіки керування та візуалізації стану системи.

Платформа Node-RED виконує функцію проміжного компонента, який отримує MQTT-повідомлення, перевіряє їхній вміст, надсилає відповідні сповіщення у Telegram або зберігає значення у базу даних. Це дає змогу створити безкодову логіку керування на етапі прототипування, що значно спрощує процес розробки (рис. 2.12).

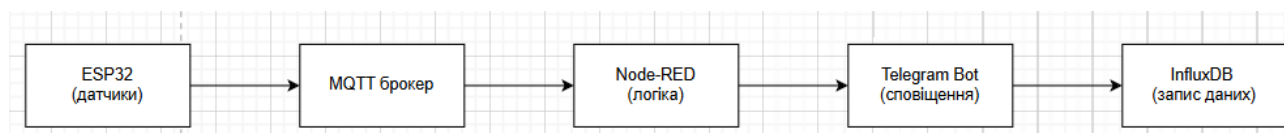


Рисунок 2.12 – Логіка передачі повідомлень

2.4.5 Візуальне моделювання на прикладі ділянки зварювання

Для попереднього моделювання роботи системи та перевірки алгоритмів контролю параметрів застосовується середовище MATLAB/Simulink. Побудована модель дозволяє змінювати вхідні параметри, зокрема температуру й тиск, а також формувати сигнали сповіщення у разі виявлення відхилень від заданих норм (рис. 2.13). Такий підхід дає змогу дослідити поведінку системи в умовах, наближених до реальних виробничих сценаріїв, ще до її повного створення.

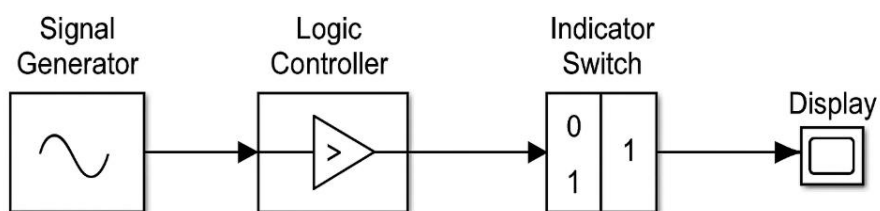


Рисунок 2.13 – Схема блоку генерації , логічного контролю та формування індикаторів

2.4.6 Допоміжне ПЗ

У процесі розробки також використовуються допоміжні цифрові інструменти, кожен із яких виконує окрему функцію:

- 1) Fritzing — для побудови схем підключення сенсорних модулів до мікроконтролера ESP32;
- 2) Visual Studio Code — як універсальне середовище для написання, редагування та відлагодження програмного коду;
- 3) Git — система контролю версій, що забезпечує керування змінами в проєкті та підтримку структурованої розробки;
- 4) Postman — для тестування API-запитів Telegram та надсилання повідомлень вручну під час налагодження функціоналу.

2.4.7 Структура програмного комплексу

Загальна структура проєкту створена у вигляді модульної архітектури, що включає такі основні компоненти (рис. 2. 14):

- модуль збору та обробки даних (реалізований на базі мікроконтролера ESP32);
- модуль зв'язку (MQTT-брокер для передачі повідомлень між елементами системи);
- логічний модуль (платформа Node-RED для обробки подій і прийняття рішень);
- інтерфейсний модуль (Telegram-бот для взаємодії з користувачем);

— аналітичний модуль (зберігання та візуалізація даних у вигляді таблиць, графіків і лог-файлів).

Такий підхід до структурної декомпозиції забезпечує гнучкість у підтримці та розвитку системи: кожен з модулів може бути оновлений або замінений без необхідності зміни всієї інфраструктури.

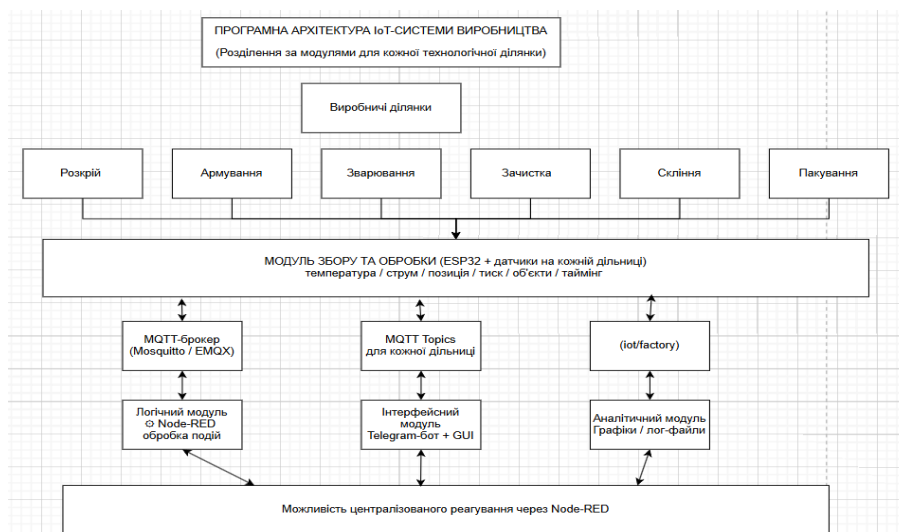


Рисунок 2.14 – Архітектура програмного комплексу з розподілом за модулями

2.5 Математична модель ділянки розкрою ПВХ-профілю

Операція розкрою полягає у відрізання заготовки із шести-метрового ПВХ-профілю відповідно до інформаційної карти розкрою, що формуються системою проєктування вікон. Технологічний результат визначається трьома основними показниками:

- довжина відрізка L з допуском $\pm \Delta L$;
- кут різку α (типово 45° або 90°);
- якість поверхні торця Q_s , що задається граничною шорсткістю R_a^{max} .

Похибки формуються сукупністю наступних чинників: нестабільністю протягу профілю, зношенням пиляльного диска, температурним розширенням матеріалу та коливаннями напруги у приводі каретки. Для аналітичного опису процесу використовуємо узагальнену дискретну динамічну модель, у якій цикл

різання розглядають як послідовність двох фаз: **позиціонування** та **відрізу** (рис. 2.15, 2.16).

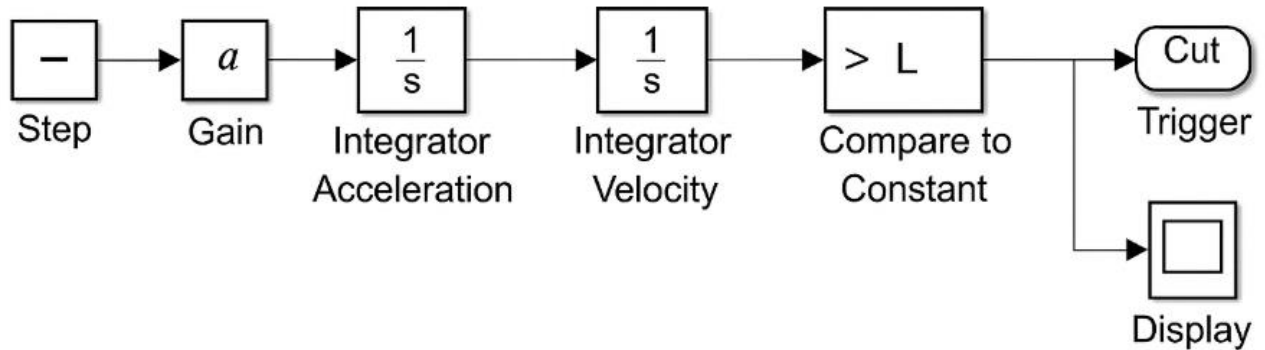


Рисунок 2.15 – Схема переміщення ПВХ профілю

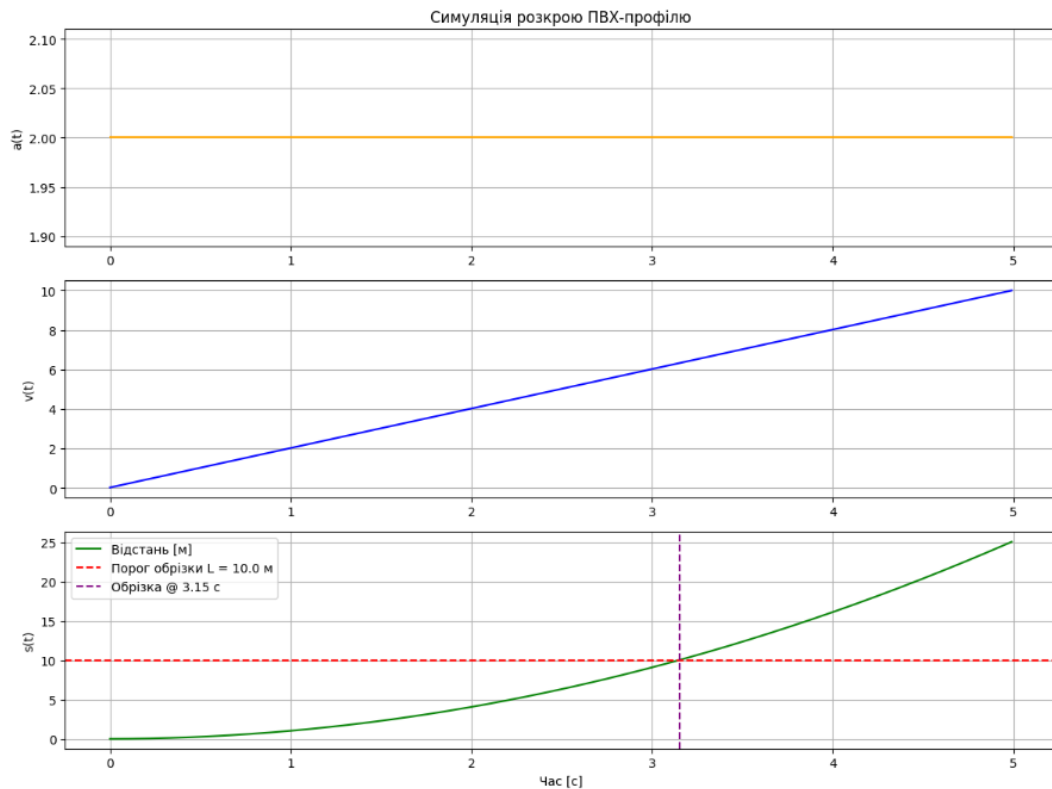


Рисунок 2.16 – Результат роботи схеми переміщення ПВХ профілю

2.5.1 Кінематична модель позиціонування

Переміщення профілю здійснюється за допомогою роликової подачі, керованою частотним перетворювачем. Дискретний крок подачі позначимо s_k . Нехай $v_f(t)$ — швидкість протягу, що регулюється ПІД-контролером з коефіцієнтами K_p , K_i , K_d . Кінематичне рівняння подачі:

$$L_k = \int_{t_0}^{t_k} v_f(t) dt = \sum_{i=1}^{n_k} v_f(i) \Delta t, \quad (1)$$

де n_k – кількість тактів вимірювання за цикл.

Системна похибка довжини:

$$\varepsilon_L = L_{\text{зад}} - L_k. \quad (2)$$

Для стабілізації довжини використовують корекційну функцію у ПІД-регуляторі:

$$u(t) = K_p \varepsilon_L + K_i \int \varepsilon_L dt + K_d \frac{d\varepsilon_L}{dt}. \quad (3)$$

2.5.2 Силова й енергетична модель відрізу

Навантаження на пиляльний диск визначається миттєвим крутним моментом який розраховується за наступною формулою:

$$M(t) = \frac{P_c(t)}{\omega(t)}, \quad (4)$$

де $\omega(t)$ – кутова швидкість диска;

$P_c(t)$ – поточна споживана потужність приводу.

Остання описується емпіричною залежністю:

$$P_c(t) = k_1 b h \omega(t) + k_2 \omega(t), \quad (5)$$

де b – ширина профілю;

h – висота стінки;

$\omega(t)$ – зріст зносу зубця;

k_1, k_2 – експериментально встановлені коефіцієнти.

Критичне значення зносу $\omega_{\text{кр}}$ задає момент, коли потрібна заміна диска.

2.5.3 Стохастична модель похибок

У реальному виробництві похибка довжини ε_L та кута ε_α описуються нормальним законом із нульовим математичним сподіванням та дисперсіями $\sigma_L^2, \sigma_\alpha^2$:

$$\varepsilon_L \sim N(0, \sigma_L^2), \varepsilon_\alpha \sim N(0, \sigma_\alpha^2). \quad (6)$$

За результатами хронометражу з 1000 відрізів отримано коефіцієнт варіації довжини $C_v = \sigma_L / L_{cp} \leq 0,15\%$, що задовольняє нормативам, однак при зносі $\omega > 0,8\omega_{кр}$ дисперсія збільшується у 2,1 рази, підтверджуючи доцільність моніторингу зносу через споживану потужність.

2.5.4 Алгоритм виявлення відхилень

Для автоматичної оцінки якості кожного зрізу впроваджено критерій:

$$\theta = \begin{cases} \text{ОК,} & |\varepsilon_L| \leq \Delta L \wedge |\varepsilon_\alpha| \leq \Delta \alpha, \\ \text{ПОПЕРЕДЖЕННЯ,} & |\varepsilon_L| \leq 1,5\Delta L \vee |\varepsilon_\alpha| \leq 1,5\Delta \alpha, \\ \text{БРАК,} & \text{інакше.} \end{cases} \quad (7)$$

Реалізація алгоритму на контролері передбачає зберігання порогів $\Delta L, \Delta \alpha$ у змінних EEPROM із можливістю оновлення через Telegram-команду */set_tolerance*.

2.5.5 Інтеграція з IoT-системою

Контролер ESP32 передає в MQTT-топік *iot/cut/event* повідомлення (рис. 2.17).

```
{
  "id": "CUT_0325",
  "L_cmd": 1200,
  "L_fact": 1199.6,
  "alpha_cmd": 45,
  "alpha_fact": 44.93,
  "theta": "OK",
  "power": 220.5,
  "wear": 0.73
}
```

Рисунок 2.17 – Приклад MQTT запиту

2.6 Математична модель дільниці армування

Операція армування полягає у встановленні всередину камерної порожнини ПВХ-профілю оцинкованого сталевго підсилювача квадратного або П-подібного перерізу з подальшою фіксацією за допомогою самонарізних шурупів. Основна мета цієї операції — підвищення жорсткості конструкції та зменшення лінійних деформацій, спричинених температурними впливами та вітровим навантаженням (рис. 2.18, 2.19).

До критичних параметрів процесу належать:

- вставна сила F_{ins} (долає статичне тертя);
- лінійний перекис підсилювача по висоті профілю y_{def} ;
- крутильний момент пошкодження профілю M_{crit} ;
- момент загвинчування шурупа T_s .

Перевищення F_{ins} або T_s призводить до мікротріщин у ПВХ-стінці, тоді як недонавантаження викликає зсув підсилювача в експлуатації.

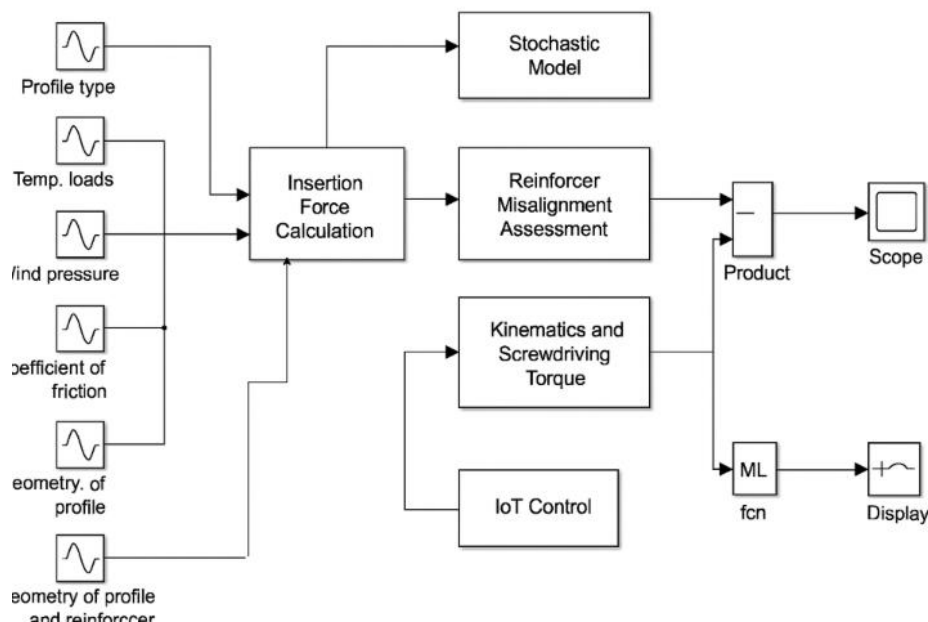


Рисунок 2.18 – Simulink модель дільниці армування

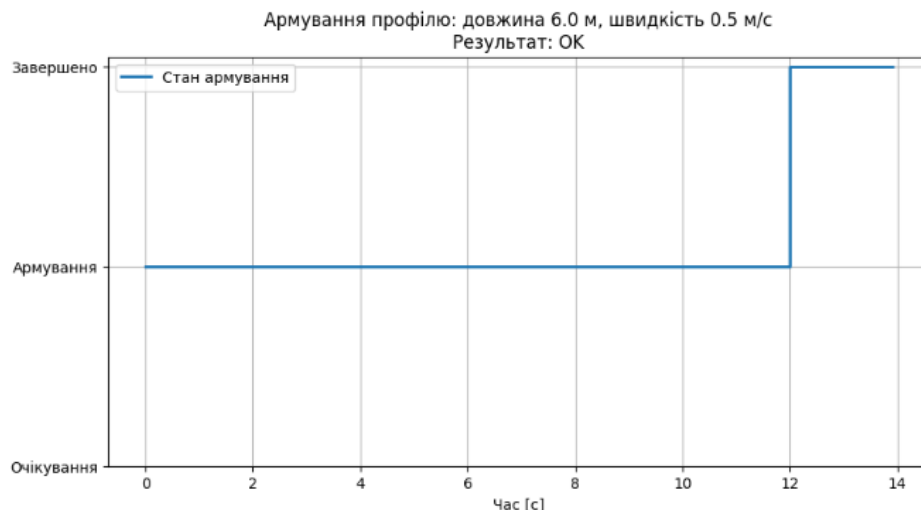


Рисунок 2.19 – Результат роботи Simulink моделі ділянки армування

2.6.1 Розрахунок сили вставлення

Вставна сила визначається з умови рівноваги гвинтової подачі роликів лінії армування, що створюють осьовий тиск на підсилювач. З урахуванням тертя «сталь–ПВХ» сила:

$$F_{ins} = \mu N + F_{el}, \quad (8)$$

де μ – коефіцієнт статичного тертя (експериментально 0,35-0,42);

N – результуюча нормальна реакція від притискних роликів;

F_{el} – компонент через пружну деформацію профілю, що за теорією пружності:

$$F_{el} = \frac{E_{pvc} I}{R}, \quad (9)$$

де $E_{pvc} = 2,9$ ГПа – модуль Юнга ПВХ;

I – момент інерції порожнини;

R – радіус кривизни вигину при розклиненні підсилювача.

Нормальна реакція роликів регулюється пневмоциліндрами; задаємо $N = 2P_r A_r$, де P_r – тиск у циліндрах, A_r – площа мембрани. Граничне значення сили:

$$F_{ins}^{max} = \sigma_{кр} A_{wall}, \quad (10)$$

де $\sigma_{кр}$ – границя міцності ПВХ (45 МПа);

A_{wall} – мінімальна площа поперечного перерізу стінки.

2.6.2 Перекіс підсилювача

Похили відхилення профілю оцінюють як:

$$y_{def} = \frac{F_{ins} l^3}{3E_{pvc} I}, \quad (11)$$

де l – довжина незакріпленої ділянки.

Умовою придатності є $y_{def} \leq y_{max} = 0,4$ мм. Для профілю довжиною 6000 мм та поточного I межа визначає верхній поріг $F_{ins}^{доп}$, який використовується у налаштуванні регулятора тиску роликів.

2.6.3 Кінематика та момент загвинчування

Шуруп формується свердлінням та різьбоутворенням за один цикл. Для запобігання зриву різьби обирають момент:

$$T_s = k_s d_s^n, \quad (12)$$

де d_s – діаметр шурупа;

$n \approx 2,2$, k_s – емпіричний коефіцієнт (від 0,18 до 0,22 Н*м/мм² для ПВХ).

Порогове $T_s^{кр}$ відповідає розширенню отвору понад 10%. Електроторцеватори обладнані крутильними датчиками, сигнал яких масштабується у Н*м.

2.6.4 Стохастична модель допусків

Похибки сили вставлення та моменту загвинчування розглядають як корельовані випадкові величини:

$$\begin{pmatrix} \varepsilon_F \\ \varepsilon_T \end{pmatrix} \sim N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{bmatrix} \sigma_F^2 & \rho \sigma_F \sigma_T \\ \rho \sigma_F \sigma_T & \sigma_T^2 \end{bmatrix} \right), \quad (13)$$

де $\rho \approx 0,3$ – коефіцієнт кореляції (зчеплення більш туге – вище і F_{ins} , і T_s).

Параметри σ_F, σ_T оцінюють за 2000-елементною вибіркою контролю якості.

2.6.5 Алгоритм контролю та інтеграції з IoT

На виробничій лінії встановлюються тензометричні датчики під роликами подачі та крутильний датчик у шпинделі шуруповерта. Контролер ESP32-WROOM-32U зчитує сигнали з датчиків із частотою 50 Гц, після чого виконує усереднення по 10 зразках (рис. 2.20). На основі оброблених даних формується вектор стану системи.

```
{  
  "id": "ARM_0571",  
  "F_ins": 820,  
  "T_s": 0.88,  
  "thetaF": "OK",  
  "thetaT": "WARN"  
}
```

Рисунок 2.20 – Приклад MQTT повідомлення з усереднення

Критерій прийняття:

$$\theta_F = \begin{cases} OK, & |F_{ins} - F_H| \leq 0,8\Delta F, \\ WARN, & 0,8\Delta F < |F_{ins} - F_H| \leq \Delta F, \\ FAIL, & |F_{ins} - F_H| > \Delta F, \end{cases} \quad (14)$$

і аналогічно θ_T . При *WARN* у Telegram-канал «Армування» відправляється повідомлення «Попередження: відхилення моменту загвинчування», при *FAIL* система блокує подачу наступної заготовки та формує запис у базі.

Сформульована модель дозволяє кількісно описати взаємозв'язок технологічних параметрів армування і забезпечує основу для розробки алгоритму автоматичного виявлення дефектів та своєчасного технічного втручання.

2.7 Математична модель ділянки зварювання

Зварювання кутів ПВХ-профілю є ключовою технологічною операцією, що забезпечує цілісність і міцність рами або стулки віконної конструкції. Процес ґрунтується на принципі контактного нагрівання торців профілю з подальшим стисканням до утворення монолітного з'єднання. Типовий цикл зварювання включає такі фази:

- 1) нагрівання торців профілів при контакті з тефлоновою плитою;
- 2) відведення нагрівального елемента та зближення торців;
- 3) стискання до моменту стабілізації зварного шва;
- 4) розтиснення вузла та зняття готового виробу.

Критичні параметри:

- температура зварювання T_{ω} ,
- тиск стискання P_s ,
- час кожної фази t_h, t_p, t_s ,
- зміщення профілю Δx ,
- стабільність нагріву (градієнт температури по площі контакту).

Вихід за межі рекомендованих значень призводить до внутрішніх напружень, появи пір, непроварів або зрушень, що критично знижує міцність з'єднання.

На рисунку 2.21 побудована структурна блок-схема моделі в середовищі Simulink.

На схемі змодельовано:

- 1) генерацію вхідних сигналів $T(t), \tau(t), P(t)$;
- 2) обчислення відхилень від оптимальних значень;
- 3) модуль оцінки якості зварювання;
- 4) блок візуалізації (Scope).

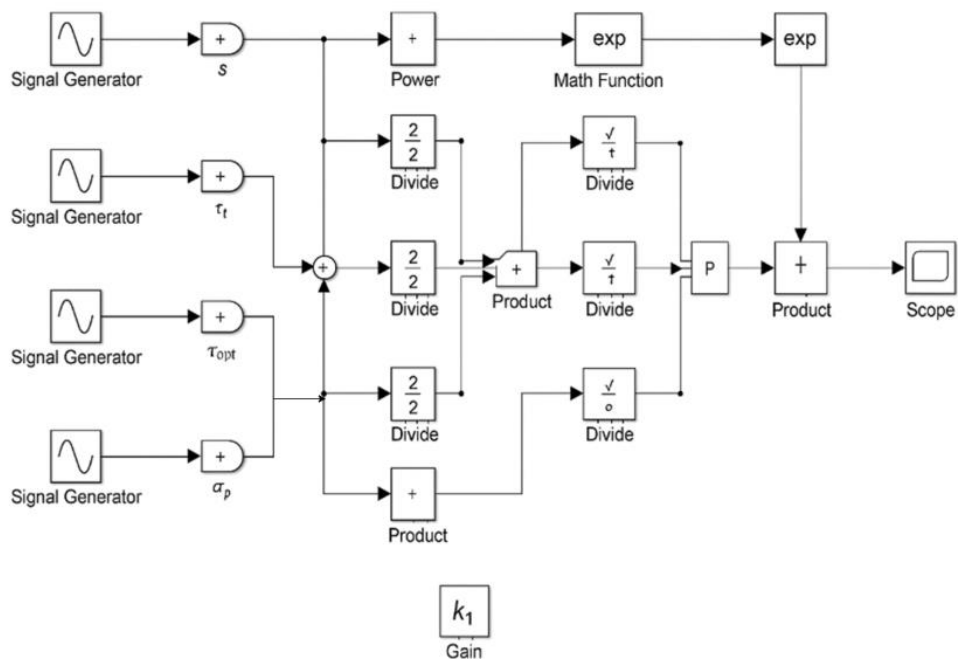


Рисунок 2.21 – Сімулінк модель ділянки зварювання

Результати роботи даної сімулінк моделі зображені на рисунках 2.22, 2.23, 2.24, 2.25.

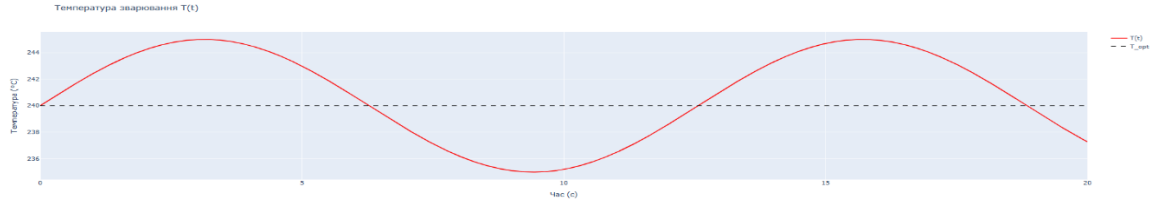


Рисунок 2.22 – Графік температури зварювання

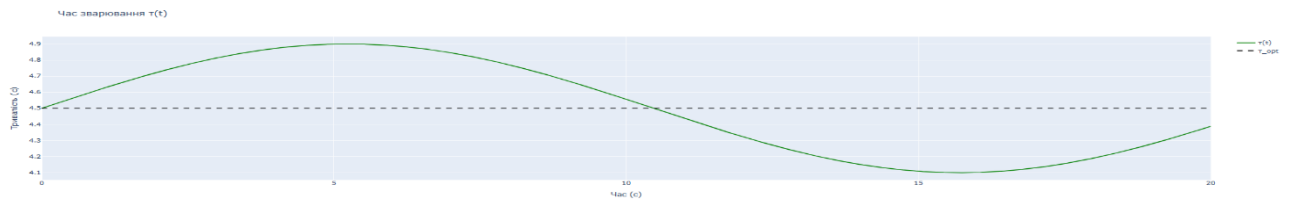


Рисунок 2.23 – Графік часу зварювання

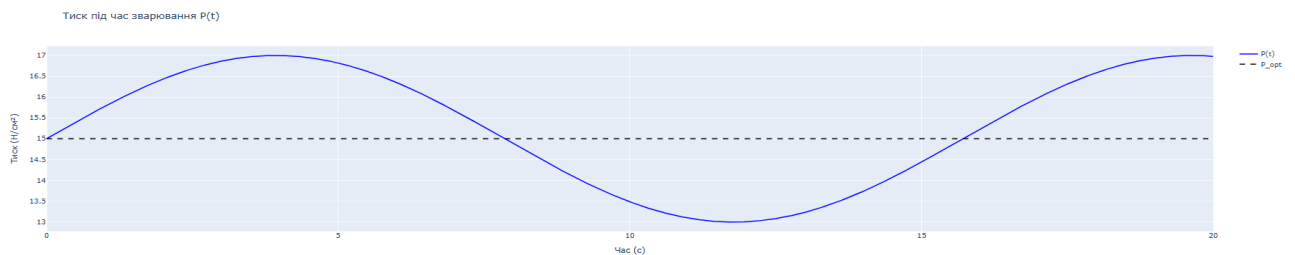


Рисунок 2.24 – Графік тиску під час зварювання

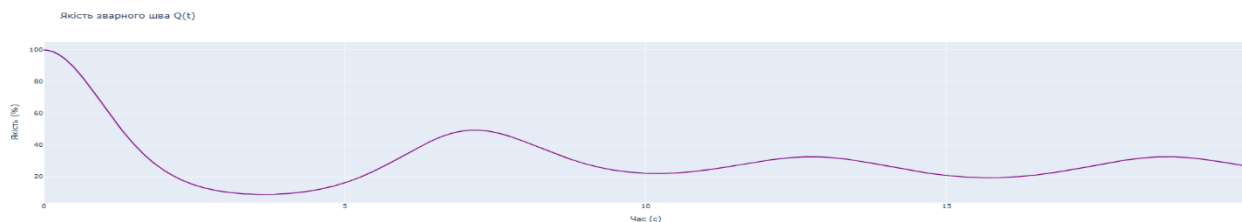


Рисунок 2.25 – Графік якості зварюваного шва

2.7.1 Тепловий баланс нагріву

Теплопередача в зоні контакту описується рівнянням нестационарної теплопровідності. З урахуванням просторової симетрії модель зводиться до одновимірної:

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}, \quad (15)$$

де $\rho=1380$ кг/м³ — густина ПВХ;
 $c=1450$ Дж/(кг·К) — теплоємність;
 $\lambda=0,19$ Вт/(м·К) — теплопровідність.

Розв'язок у вигляді серії Фур'є дозволяє оцінити глибину прогріву при часі t_h . Критичною умовою є досягнення температури **плавлення ПВХ** на глибині $x_{кр} \approx 0,8$ мм. Температурний розподіл:

$$T(x, t) = T_0 + (T_{пл} - T_0) \cdot erf\left(\frac{x}{2\sqrt{\alpha t}}\right), \quad (16)$$

де $\alpha=\lambda/(\rho c)$ — термічна дифузивність;
 erf — функція помилки.

Мінімально допустимий час нагрівання:

$$t_h^{min} = \frac{x_{кр}^2}{4\alpha} \approx 2,3 \text{ с.} \quad (17)$$

2.7.2 Силова модель стискання

Стискання торців виконується за допомогою пневматичних або електромеханічних приводів. Рівномірність тиску:

$$P_s = \frac{F_s}{A_c}, \quad (18)$$

$$F_s = P_{\text{пн}} \cdot A_{\text{цил}}, \quad (19)$$

де A_c – площа зварювального шва;

$P_{\text{пн}}$ – тиск у пневмоциліндрах.

Деформація у шві:

$$\varepsilon_s = \frac{\Delta x}{x_0} = \frac{F_s}{E_{\text{плав}}}, \quad (20)$$

де $E_{\text{плав}}$ – модуль пружності розм'якшеного ПВХ (падає до 10-15 МПа).

Оптимальний тиск обмежується умовою:

$$P_s \leq P_{\text{кр}} = \sigma_{\text{наплив}} / (1 + \nu), \quad (21)$$

де $\sigma_{\text{наплив}}$ – допустима межа напливу ($\approx 3\text{-}5$ МПа);

ν – коефіцієнт Пуассона.

2.7.3 Модель тривалості циклу

Цикл зварювання:

$$t_{\text{заг}} = t_h + t_p + t_s + t_c, \quad (22)$$

де t_h – нагрівання (2-4 с);

t_p – відведення плити (0,5-1 с);

t_s – стискання (4-6 с);

t_c – охолодження (1-2 с).

Оптимальне співвідношення:

$$\frac{t_h}{t_s} \approx 0,5 - 0,7, \quad (23)$$

що забезпечує правильну полімеризацію шва без надмірної деформації.

2.7.4 Критерії оцінки якості

Формально якість шва Q визначається функціоналом:

$$Q = f(T_w, P_s, t_h, t_s) = \begin{cases} OK, & T_w \in [T_{\min}, T_{\max}] \wedge P_s \in [P_{\min}, P_{\max}] \wedge \dots \\ WARN, & \text{є незначні відхилення,} \\ FAIL, & \text{один або більше параметрів за межами.} \end{cases} \quad (24)$$

Ці межі зберігаються в контролері й можуть бути оновлені дистанційно через Telegram-інтерфейс.

2.7.5 Алгоритм дій та IoT-інтеграція

Контролер ESP32 виконує:

- опитування термопари T (DS18B20), датчика тиску P (MPX5010);
- заміри часу фаз (таймери мікроконтролера);
- обчислення показника Q;
- надсилання MQTT-повідомлення в топик `iot/weld/event` (рис. 2. 26).

```
{  
  "id": "WELD_2193",  
  "T_w": 243,  
  "P_s": 3.8,  
  "t_h": 2.8,  
  "t_s": 5.1,  
  "status": "OK"  
}
```

Рисунок 2.26 – Приклад MQTT повідомлення

При `status` $\neq$ OK Telegram-бот надсилає оператору сповіщення з конкретною причиною: «Зварювання відхилено: температура нижча за мінімальну».

Побудована модель формалізує технологічний процес зварювання, визначає допустимі діапазони його параметрів і є основою для розробки системи раннього виявлення потенційних відхилень на основі IoT-моніторингу.

2.8 Математична модель дільниці зачистки

Після зварювання рам або стулок зварні шви містять надлишок розплавленого ПВХ — так звані напливи. Метою дільниці зачистки є механічне видалення цього надлишку без пошкодження цільової поверхні профілю.

Операція виконується на зачистних верстатах, оснащених фрезами, різакми та ножами, що рухаються по контуру шва згідно заданої програми (рис. 2.27, 2.28).

Критично пріорітетною є точність зачистки — як у глибину (щоб не пошкодити профіль), так і у площині (щоб не залишити надлишок). Основними керованими параметрами є:

- швидкість обертання інструменту ω ;
- лінійна швидкість переміщення інструмента v_{lin} ;
- глибина знімання шару h ;
- сила притискання до поверхні F_c ;
- координати позиціонування фрези x, y, z .

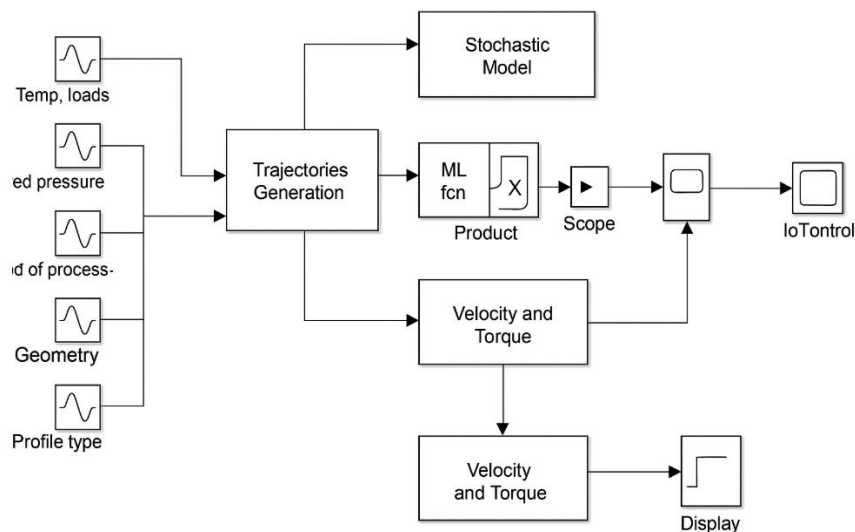


Рисунок 2.27 – Simulink модель ділянки зачистки

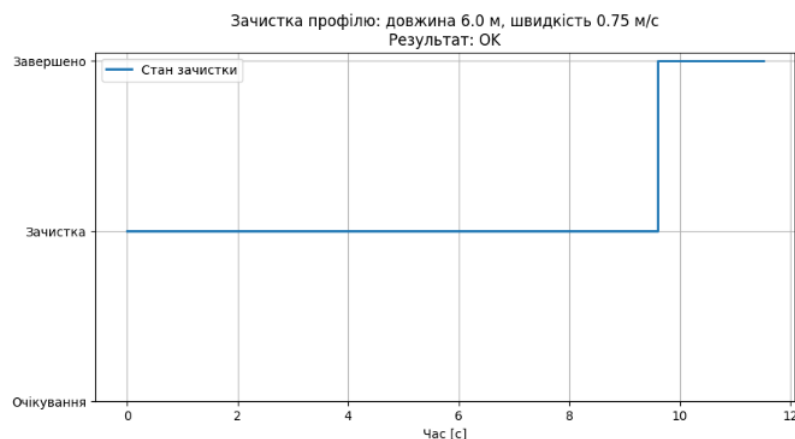


Рисунок 2.28 – Графік стану зачистки профілю

2.8.1 Геометрична модель траєкторії зачистки

Рух інструменту відбувається в тривимірному просторі. Для фіксованого маршруту зачистки шва використовують параметричне подання контуру:

$$\begin{cases} x(t) = x_0 + v_x t, \\ y(t) = y_0 + v_y t, \\ z(t) = z_0 + f(t), \end{cases} \quad (25)$$

де $f(t)$ — функція компенсації кривини шва залежно від типу профілю.

Визначення $f(t)$ відбувається на основі шаблонного сканування ПВХ-поверхні (при наявності датчика глибини або обмежувача).

У найпростішому випадку шов моделюється як трикутна призма зі стороною a , тоді:

$$h = \frac{a}{2}, \quad (26)$$

і координата $z(t)$ = стала.

2.8.2 Модель знімання шару матеріалу

Процес механічної обробки описується за допомогою моделі енерговитрат на зняття одиниці об'єму матеріалу:

$$P = K_c A_c v_{lin}, \quad (27)$$

де P – споживана потужність інструмента;

K_c – коефіцієнт опору різанню для ПВХ ($\approx 200\text{-}300 \text{ Н/мм}^2$);

$A_c = h \cdot \omega$ – площа зрізу;

v_{lin} – лінійна швидкість фрези.

Цей вираз дозволяє встановити верхню межу допустимої швидкості, щоб уникнути перевантаження шпинделя та зношення інструмента.

2.8.3 Динамічна модель сили контакту

Рівномірність сили притискання забезпечується пружинними або пневматичними механізмами. З урахуванням лінійної моделі пружності:

$$F_c(t) = k_s \cdot (z_0 - z(t)), \quad (28)$$

де k_s – жорсткість елементу зворотного зв'язку (Н/мм);

$z(t)$ – зміщення інструмента у вертикальній осі.

Функція $F_c(t)$ дозволяє визначити надмірний тиск (ризик пошкодження поверхні) або недостатній (неповна зачистка). Граничні значення:

— $F_c^{min} \approx 4H$ – для гарантованого торкання;

— $F_c^{max} \approx 15H$ – межа появи заглиблень.

2.8.4 Алгоритм контролю якості обробки

З метою автоматичного оцінювання результату зачистки використовується функція оцінки залишкового шару:

$$\delta_h = h_{ном} - h_{факт}, \quad (29)$$

яку можна виміряти сенсором безконтактного типу (наприклад, лазерний лінійний датчик). При $|\delta_h| \leq 0,1\text{мм}$ – вважається нормальним, інакше формується повідомлення про можливу помилку позиціювання.

Алгоритм включає:

- 1) запис координат фрези під час проходу;
- 2) вимірювання сили тиску та споживаної потужності;
- 3) формування рішення $status = OK / WARN / FAIL$.

2.8.5 Інтеграція з системою моніторингу

Контролер ESP32 з'єднаний з датчиком тиску, лінійним датчиком переміщення (навігація фрези) та сенсором потужності. Дані агрегуються і передаються MQTT-повідомленням у топик `iot/clean/event` у форматі, зображеному на рисунку 2. 29.

```
{  
  "id": "CLEAN_0114",  
  "F_c": 9.2,  
  "v_lin": 1.7,  
  "delta_h": -0.04,  
  "status": "OK"  
}
```

Рисунок 2.29 – MQTT повідомлення у топик `iot/clean/event`

У разі статусу `WARN` або `FAIL` Telegram-бот надсилає повідомлення «Відхилення глибини зачистки: -0.13 мм. Перевірити координати фрези.».

Ця математична модель дільниці зачистки дозволяє забезпечити контроль точності обробки швів, що критично для герметичності виробу, та інтегрування цих показників в єдину систему моніторингу технологічного процесу.

2.9 Математична модель дільниці скління

Скління є завершальним етапом збирання віконної конструкції, що передбачає вставлення склопакету в підготовлену раму та фіксацію його штапиками. На цій дільниці формується теплозахисний, звукоізоляційний і герметичний контур.

Основні вимоги до скління:

- відсутність перекосу склопакету в рамі;
- рівномірний розподіл притискного зусилля;
- герметичне облягання по всьому периметру;
- збереження компенсаційного зазору на випадок температурного розширення.

Дане математичне моделювання охоплює деформаційні процеси в зоні контакту скло–рама–ущільнювач, пружні властивості фіксуючих елементів і динаміку навантаження при запресуванні штапиків.

2.9.1 Геометрична модель посадки склопакету

Склопакет має менший розмір, ніж світловий проріз у рамі. Залишковий компенсаційний зазор:

$$\delta_c = \alpha_{\text{рама}} - \alpha_{\text{скло}} - 2d, \quad (30)$$

де $\alpha_{\text{рама}}$ – внутрішній розмір рами;

$\alpha_{\text{скло}}$ – розмір склопакета;

d – товщина ущільнювача.

Умовою придатної посадки є: $0,5 \leq \delta_c \leq 1,5$ мм, що дозволяє обмежене термічне розширення без утворення зазорів або перенавантаження.

2.9.2 Пружна модель ущільнення

Контакт склопакета з рамою відбувається через гумовий ущільнювач, який поводить як пружний елемент. Напряга стиснення:

$$\sigma = \frac{F_c}{A_u} = E_u \cdot \varepsilon, \quad (31)$$

де F_c – сила, з якою рама тисне на скло;

A_u – площа контакту;

E_u – модуль пружності ущільнювача;

$\varepsilon = \Delta d/d_0$ – відносна деформація ущільнювача.

Нормований діапазон: $0,15 \leq \varepsilon \leq 0,4$, що забезпечує герметичність без надмірного тиску на скло. При $\varepsilon < 0,15$ – можливе проникнення вологи, при $\varepsilon > 0,4$ – ризик руйнування скла при ударі.

2.9.3 Силова модель запресування штапиків

Фіксація склопакету здійснюється запресуванням штапиків із заціпками. Зусилля, необхідне для фіксації, визначається як:

$$F_s = \mu_s \cdot A_s \cdot \sigma_s, \quad (32)$$

де μ_s – коефіцієнт тертя (0,25...0,3);

A_s – площа контакту заціпки;

σ_s – напруження розширення при вставленні.

Контроль параметра F_s має ключове значення, оскільки його перевищення часто спричиняє пошкодження торця профілю або появу мікротріщин на внутрішній стінці. Щоб цього уникнути, на ручних пресах використовують моментні обмежувачі, а в автоматизованих системах — динамометричні сенсори.

2.9.4 Рівновага рами під навантаженням

Після скління розраховано вплив зовнішніх навантажень (вітру, відкривання) на деформативність зібраної конструкції. Припускаючи розподілений тиск q на площину скла, вертикальне прогинання визначається як:

$$w_{max} = \frac{q \cdot a^4}{64D}, \quad (33)$$

де a – вільний проліт скла (довший бік);

$D = \frac{E_g h^3}{12(1-\nu^2)}$ – жорсткість скляної плити;

$E_g = 72$ ГПа – модуль пружності скла;

h – товщина скла.

Максимальний допустимий прогин:

$$w_{max} \leq \frac{a}{300}, \quad (34)$$

тобто при $a=1200$ мм прогин не повинен перевищувати 4 мм. Цей критерій потрібно враховувати при виборі товщини склопакета та розрахунку відстаней між точками фіксації.

2.9.5 Інтеграція з системою моніторингу

Система автоматизації передбачає:

- сенсор сили притискання F_c ;
- індуктивний датчик фіксації штапика;

— оцінку відносного зміщення δc (на основі обробки відео або лазерного вимірювання).

Контролер ESP32 формує повідомлення у топик `iot/glaze/event` у форматі, зображеному на рисунку 2.30.

```
{  
  "id": "GLZ_1782",  
  "F_c": 6.2,  
  "delta_c": 0.8,  
  "epsilon": 0.24,  
  "status": "OK"  
}
```

Рисунок 2.30 – MQTT повідомлення у топик `iot/glaze/event`

При $\epsilon > 0.4$ надсилається сигнал «Перевищення деформації ущільнювача. Перевірити посадку скла».

Сформований результат моделювання співвідношень допоможе контролювати зусилля фіксації та рівень герметичності, це потрібно для гарантійної міцності виробу.

2.10 Математична модель ділянки контролю якості

Контроль якості у виробництві металопластикових вікон — це не лише спосіб виявити дефекти. Він також забезпечує зворотний зв'язок, який дає змогу коригувати роботу окремих ділянок. Зазвичай перевірка включає візуальну оцінку (шви, скло, геометрія), вимірювання розмірів, тестування фурнітури та герметичності. Під час автоматизації цієї стадії використовують сенсори, цифрову фіксацію параметрів і алгоритми, що приймають рішення на основі статистичних даних.

2.10.1 Статистична модель браку

Загальна модель передбачає, що дефектність є функцією відхилень на попередніх ділянках:

$$P_{\text{брак}} = f(\epsilon_L, \theta_w, \delta_h, \epsilon_T, \epsilon_c), \quad (35)$$

де ε_L – похибка довжини при розкрої;
 θ_w – стан зварювання (OK/WARN/FAIL);
 δ_h – залишок після зачистки;
 ε_T – відхилення моменту в армуванні;
 ε_c – деформація ущільнювача в склінні.
Формально це – логістична регресійна модель:

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1\varepsilon_L + \beta_2\theta_w + \beta_3\delta_h + \beta_4\varepsilon_T + \beta_5\varepsilon_c, \quad (36)$$

де коефіцієнти β_i визначаються за навчальною вибіркою з архіву браку. Ця модель дає змогу прогнозувати ймовірність виходу виробу з норм та сформулювати оцінку ризику ще до кінцевої перевірки.

2.10.2 Геометричний контроль

Основні лінійні параметри: висота, ширина, діагоналі. Контроль може бути реалізований за допомогою:

- лазерних далекомірів (оцінка габаритів);
- кутових сенсорів (перевірка паралельності);
- цифрових шаблонів (порівняння зі стандартом).

Різниця діагоналей: $\Delta_d = |d_1 - d_2| \leq 2\text{мм}$.

Це ключовий показник перекосу, обчислюваний автоматично при наявності двох оптичних сенсорів.

2.10.3 Контроль фурнітури

Крутильний момент на ручці та зусилля при замиканні вікна: $M_\phi \in [0,8; 1,5]$

Контроль проводиться за допомогою крутильного сенсора або тензодатчика при закриванні. У разі відхилення параметрів формується протокол дефекту: «Надмірне зусилля при замиканні. Можливе зміщення напрямної».

2.10.4 Перевірка герметичності

Модель герметизації будується на перепаді тиску всередині камери. Якщо камеру закрити й подати повітря з відомою витратою, тиск $p(t)$ повинен стабілізуватися за законом:

$$p(t) = p_0 \left(1 - e^{-\frac{t}{\tau}}\right), \quad (37)$$

де τ – стала часу, пов'язана з витоками.

При порушенні ущільнення τ значно зменшується. Контролер аналізує градієнт:

$$\left. \frac{dp}{dt} \right|_{t \rightarrow 0} > \Delta_{\text{макс}} \rightarrow \text{витік}. \quad (38)$$

2.10.5 Алгоритм інтеграції оцінок

Контролер формує звіт за всіма параметрами у вигляді повідомлення, зображеному на рисунку 2.31.

```
{  
  "id": "CTRL_9904",  
  "delta_d": 1.4,  
  "M_f": 1.1,  
  "F_lock": 38,  
  "tau": 5.3,  
  "P_brak": 0.027,  
  "status": "OK"  
}
```

Рисунок 2.31 – MQTT повідомлення про дефекти

Якщо хоча б один із контрольованих параметрів виходить за допустимі межі, записується відповідний дефект, повідомлення надсилається у Telegram-канал. Оператор отримує сповіщення, після чого має можливість змінити status вручну або відправити виріб на доопрацювання.

2.11 Інтегрована модель виробничого процесу

Побудова математичних моделей для кожної з ділень дозволяє глибоко дослідити специфіку окремих технологічних операцій, проте для створення цілісної системи управління необхідне **інтегрування цих моделей** у єдиний цілосний опис. Це забезпечує:

- урахування міждільничних залежностей;
- виявлення кумулятивних впливів похибок;
- підтримку наскрізного контролю якості;
- побудову адаптивного алгоритму оптимізації параметрів виробництва в реальному часі.

Інтеграція дозволяє перейти від ізольованих вимірювань до **системного управління**, де всі значення параметрів пов'язані між собою через інформаційну структуру та часову послідовність технологічних кроків.

2.11.1 Структура узагальненої моделі

Розглянемо узагальнену виробничу систему як послідовну динамічну дискретну мережу, де кожна ділянка є вузлом D_i , а вихідні параметри Y_i стають вхідними для наступної D_{i+1} :

$$D_1 \xrightarrow{Y_1} D_2 \xrightarrow{Y_2} \dots \xrightarrow{Y_5} D_6, \quad (39)$$

де D_1 – розкрій;

D_2 – армування;

D_3 – зварювання;

D_4 – зачистка;

D_5 – скління;

D_6 – контроль.

Для кожної ділянки модель представлена у вигляді:

$$Y_i = f_i(X_i, P_i, T_i), \quad (40)$$

де X_i — технологічні вхідні параметри;

P_i — параметри, що передаються з попередніх етапів;

T_i — часові характеристики циклу.

Це дозволяє формалізувати **ланцюгову залежність**: похибка на одній ділянці є і може трансформуватись або підсилюватись на наступних.

2.11.2 Побудова зведеної матриці параметрів

Всі ключові параметри агрегуються у **матрицю стану виробу**, де кожен рядок — це один виріб, а кожен стовпець — значення контрольного параметра з певної ділянки (рис. 2.32)

ID виробу	L	θ_{arm}	T_w	δ_h	ε_c	Δ_d	<i>status</i>
CUT_001	1200.3	OK	243.1	-0.02	0.26	1.3	OK
CUT_002	1201.5	WARN	239.4	-0.13	0.42	3.1	FAIL
...	...	...	...	...	...	...	...

Рисунок 2.32 – Матриця IoT системи на базі MQTT даних

Таку матрицю формує IoT-система автоматично на базі MQTT-даних. Вона є основою для статистичного аналізу, машинного навчання та динамічної оптимізації.

2.11.3 Формування інтегрального критерію якості

На основі зібраних параметрів обчислюється **інтегральна функція якості виробу**:

$$Q_{\text{int}} = w_1 q_1 + w_2 q_2 + \dots + w_n q_n, \quad (41)$$

де q_i — нормалізоване значення параметра (від 0 до 1);

w_i — вагові коефіцієнти важливості (встановлюються експертно або на основі навчальної вибірки).

При $Q_{\text{int}} < Q_{\text{поріг}}$ система видає попередження або автоматично змінює режим роботи попередніх ділянок (реалізація адаптивного зворотнього зв'язку).

2.11.4 Реалізація наскрізного моніторингу

Контролер кожної ділянки надсилає параметри у централізовану базу даних або хмарний MQTT-брокер. Обробка даних виконується через **Node-RED** + **Telegram API**:

- формування електронної картки виробу;
- зведення даних у реальному часі;
- видача прогнозу браку до завершення циклу.

Оператор отримує сповіщення типу:

Виріб GLZ_0912: ймовірність браку 0,41. Причина: нестабільний шов на D3, перевищено зусилля фіксації на D5.

2.11.5 Сценарії реакції системи

При виявленні тенденцій до браку система ініціює:

- зупинку подачі наступної заготовки;
- перенаправлення виробу на повторну перевірку;
- адаптацію технологічних параметрів у реальному часі (наприклад, зниження тиску в зварювальній ділянці).

Можливе також формування **автоматизованих звітів для керівництва**, що включають ключові індикатори:

- середній рівень похибок по кожній ділянці;
- відсоток браку;
- частота спрацьовування попереджень.

Висновки до розділу 2

Другий розділ цієї роботи закладає теоретичний та методологічний фундамент для розробки моєї системи автоматизації. Мною обґрунтовано перехід від класичної централізованої моделі до більш гнучкої,

децентралізованої IoT-архітектури, яка краще відповідає потребам малого бізнесу.

Мною була розроблена методика аналізу, яка охоплює всі ключові рівні виробництва: від структури та функціональності до параметрів, сценаріїв роботи й користувацької взаємодії. Це дозволило визначити технічні вимоги до системи, критичні параметри процесу й оцінити, які дані потрібно збирати на кожному етапі.

Сама система включає сенсори, контролери ESP32, зв'язок через MQTT, логіку обробки подій і просту візуалізацію в Telegram. Архітектура легко змінюється під конкретні задачі і не вимагає заміни існуючих машин чи ліній.

Також побудовано окремі математичні моделі для кожного етапу:

1) **розкрій** — моделі руху подаючих механізмів, витрат енергії на різання й оцінка точності;

2) **армування** — модель зусиль вставлення, деформації профілю й обертового моменту;

3) **зварювання** — процес нагріву, стиснення і оцінка якості шва;

4) **зачистка** — траєкторія руху фрези, сила притискання та залишкова висота шва;

5) **скління** — посадка склопакета, поведінка ущільнювачів під навантаженням;

6) **контроль якості** — ймовірність браку, геометричні й функціональні перевірки, герметичність.

Ключовим етапом цього розділу була побудова єдиної математичної моделі виробничого циклу — у вигляді дискретної мережі, яка відображає динаміку процесу. На її основі сформовано загальну матрицю параметрів виробів і запропоновано спосіб розрахунку інтегрального показника якості. Це дозволяє оцінити стан вікна ще до завершення всього циклу — і вчасно втрутитися, якщо щось іде не так.

Розділ закладає теоретичну та математичну основу для впровадження гнучкої IoT-системи управління, адаптованої до реальних умов виробництва. Наступний розділ присвячено створенню цієї системи.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ ІОТ-АВТОМАТИЗАЦІЇ ВИРОБНИЦТВА МЕТАЛОПЛАСТИКОВИХ ВІКОН

3.1 Архітектура запропонованої системи управління

Запропонована система IoT-автоматизації охоплює повний цикл виробництва металопластикових вікон — від розкрою профілю до контролю якості готової продукції. Система розроблена за принципом модульної децентралізованої архітектури, що дозволяє поетапно автоматизувати окремі ділянки, не порушуючи загального виробничого процесу.

До складу апаратної частини входять:

- сенсори (температури, тиску, вібрації, вологості, положення, лічильники циклів тощо);
- контролери ESP32, закріплені за кожною ділянкою (розкрій, армування, зварювання, зачистка, скління, пакування);
- MQTT-брокер — для організації зв'язку між контролерами та сервером;
- хмарний сервер (або локальний вузол) для зберігання, обробки та візуалізації даних;
- інтерфейс Telegram-бота — для сповіщень, контролю та зворотного зв'язку з оператором;
- виконавчі пристрої — реле, приводи, звукова сигналізація, світлодіоди, тощо (рис. 3.1).

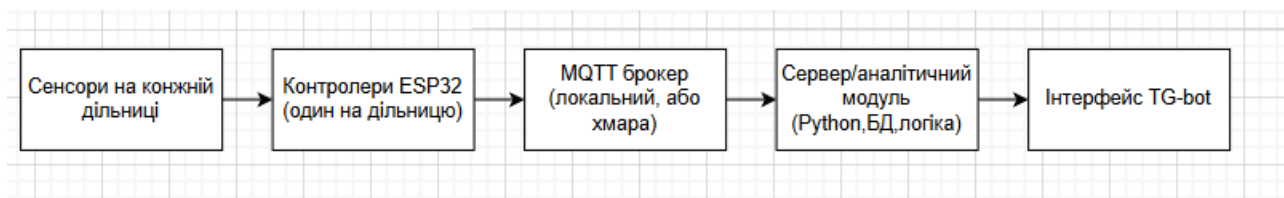


Рисунок 3.1 – Узагальнена структура системи управління

Завдяки цій конфігурації система здатна:

- фіксувати ключові параметри на всіх етапах виробництва;
- локально реагувати на відхилення за заданими порогоми;

- передавати події та аналітику на центральний сервер;
- інформувати відповідальних осіб у Telegram у разі порушення норм;
- забезпечувати двосторонню взаємодію між користувачем та виробничим обладнанням.

3.1.1 Обґрунтування вибору елементної бази

Контролери ESP32. Ці контролери обрано як універсальні вузли для збору, попередньої обробки та передачі даних. Вони підтримують Wi-Fi, споживають мало енергії й можуть одночасно працювати з кількома типами датчиків. Завдяки цьому ESP32 легко інтегруються в різні етапи виробництва — від розкрою до пакування.

Датчики. Вибір сенсорів залежить від завдань конкретної дільниці:

- розкрій — оптичні або магнітні сенсори для контролю положення та лічильники розрізів;
- армування — датчики тиску та моменту;
- зварювання — температурні сенсори та датчики часу контакту;
- зачистка — сенсори вібрації та наявності;
- скління — навантажувальні сенсори й інфрачервоні бар'єри;
- пакування — фотодатчики та сенсори ваги.

MQTT-протокол. MQTT дає змогу будувати систему, де пристрої можуть спілкуватися між собою паралельно або централізовано. Кожен контролер публікує свої дані у відповідному каналі — наприклад, /cutting/temp, /welding/pressure, /packing/count. Це спрощує фільтрацію й аналіз інформації з різних етапів.

Telegram Bot API. Інтерфейс у вигляді Telegram-бота виконує кілька важливих функцій:

- надсилає сповіщення про помилки чи вихід параметрів за межі;
- дозволяє отримати поточні значення з будь-якої дільниці;

— підтримує дистанційне керування — зупинку, перезапуск, зміну налаштувань;

— генерує періодичні звіти й надає коротку аналітику.

Логіка взаємодії між компонентами:

1) сенсори на кожній ділянці збирають дані відповідно до свого призначення;

2) контролери ESP32 виконують базову обробку: усереднюють значення, порівнюють із порогоми, фільтрують шум;

3) дані передаються через MQTT у вигляді публікацій з мітками часу й локації;

4) сервер приймає ці повідомлення, зберігає їх, аналізує та формує сповіщення у разі відхилень;

5) Telegram-бот надсилає ці сповіщення оператору й дозволяє управляти системою.

Приклад роботи системи:

1) на ділянці армування тиск падає нижче 70 Н/см^2 ;

2) контролер ESP32 фіксує цю подію та надсилає повідомлення на канал /reinforcement/alert_pressure_low;

3) Python-скрипт приймає сигнал через MQTT;

4) Telegram-бот надсилає оператору повідомлення: *«Армування: тиск = 68 Н/см^2 , нижче допустимого рівня»*;

5) оператор відповідає командою /pause_line reinforcement;

6) система активує реле та зупиняє живлення обладнання.

3.2 Практична реалізація математичних моделей

У попередньому розділі були описані моделі, які оцінюють технічний стан обладнання на основі вхідних параметрів — температури, тиску, циклів, тривалості простою тощо. У цьому розділі показано, як ці моделі реалізовані на практиці.

Мікроконтролери ESP32 використовують C++ (на базі Arduino Core) для локальної обробки, а Python-скрипти відповідають за серверну логіку та взаємодію з Telegram. Завдяки цьому дані з дільниць обробляються в реальному часі, а оператор має зручний інтерфейс для керування й моніторингу.

Основна реалізована логіка:

- обчислення ковзного середнього температури (фільтрація шумів);
- виявлення виходу за допустимі межі з урахуванням гістерезису;
- класифікація подій (норма, попередження, аварія);
- підрахунок кількості повторюваних відхилень;
- прогноз збою на основі накопичених відхилень.

Формула ковзного середнього:

$$T_{\text{середн}} = \frac{1}{n} \sum_{i=0}^{n-1} T_i, \quad (42)$$

де $n=5$ — розмір вікна;

T_i — останні значення з термодатчика.

Реалізація в Arduino C++ зображена на рисунку 3.2.

```
float temps[5];
int index = 0;

void updateTemperature(float newTemp) {
    temps[index] = newTemp;
    index = (index + 1) % 5;

    float sum = 0;
    for (int i = 0; i < 5; i++) sum += temps[i];
    float avg = sum / 5;

    if (avg > TEMP_MAX) {
        sendMQTT("alert/overheat", avg);
        digitalWrite(RELAY_PIN, LOW); // аварійне вимкнення
    }
}
```

Рисунок 3.2 – Логіка реалізації математичних моделей

3.3 Обробка вхідних сигналів із сенсорів

Зчитування та інтерпретація сигналів відбувається на рівні ESP32. Всі сенсори працюють з інтерфейсами SPI, I2C, ADC або GPIO, залежно від типу. Наприклад:

- **температурні сенсори MAX6675 (термопари)** передають дані через SPI;
- **тензодатчики (HX711)** зчитуються через спеціалізовані бібліотеки (bit-banging GPIO);
- **датчики часу циклу** використовують внутрішній таймер ESP32 або зовнішній RTC-модуль (DS3231).

На кожному мікроконтролері реалізовано нескінченний цикл loop(), в межах якого виконується:

- зчитування даних;
- обробка фільтрами;
- логічна оцінка (if-then);
- публікація результату через MQTT (JSON-структура) (рис. 3.3, 3.4, 3.5).

```
{  
  "unit": "glazing_station",  
  "parameter": "pressure",  
  "value": 3.21,  
  "status": "warning",  
  "timestamp": "2025-06-10T10:24:18Z"  
}
```

Рисунок 3.3 – MQTT повідомлення про обробку сигналів

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

WiFiClient espClient;
PubSubClient client(espClient);

void setupMQTT() {
    client.setServer(mqtt_server, 1883);
}

void publishSensorData(float value) {
    char payload[64];
    snprintf(payload, sizeof(payload), "{\"value\":%.2f}", value);
    client.publish("factory/cutting/temp", payload);
}
```

Рисунок 3.4 – Фрагмент коду публікації даних у MQTT

```
import paho.mqtt.client as mqtt
import telegram

bot = telegram.Bot(token="YOUR_BOT_TOKEN")

def on_message(client, userdata, msg):
    payload = json.loads(msg.payload.decode())
    if payload['status'] == 'alert':
        text = f"🚨 Аварія на {payload['unit']}: {payload['parameter']} = {payload['value']}"
        bot.send_message(chat_id=YOUR_CHAT_ID, text=text)

client = mqtt.Client()
client.on_message = on_message
client.connect("localhost", 1883)
client.subscribe("factory/+/#")
client.loop_forever()
```

Рисунок 3.5 – Фрагмент коду обробки та інформування в телеграм

3.3 Розробка Telegram-інтерфейсу

Інтерфейс, реалізований через месенджер Telegram, виконує роль центрального каналу зв'язку між оператором та автоматизованою системою. Telegram був обраний з огляду на:

- підтримку бот-інтерфейсу з відкритим API;
- низький поріг входу для робітників (не потребує навчання);
- підтримку форматуваних повідомлень, кнопок, меню, вкладень;
- наявність мобільних та десктопних версій, що дозволяє використовувати систему без спеціалізованого обладнання.

Telegram-бот слугує **уніфікованим інтерфейсом моніторингу, оповіщення та керування**, а структура взаємодії з ним має три рівні:

1) **інформування (push)** — автоматичне надсилення повідомлень про події, аварії, відхилення параметрів (рис. 3.6);

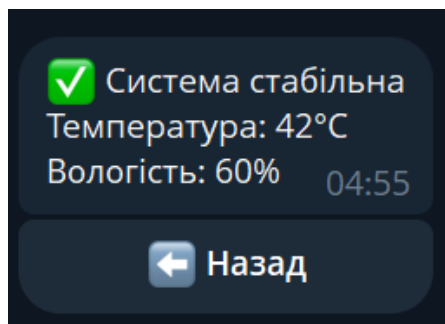


Рисунок 3.6 – Приклад інформування в Telegram боті

2) **опитування (pull)** — запит поточного стану діляниць, метрик, історії (рис. 3.7);

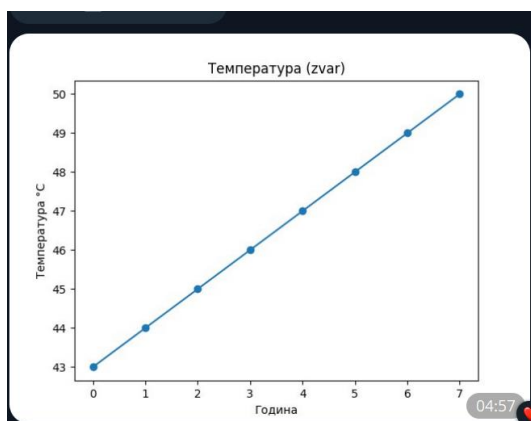


Рисунок 3.7 – Приклад запиту поточних діляниць (графічно)

3) **керування** — відправка команд системі (зупинка, запуск, налаштування порогів) (рис. 3.8).

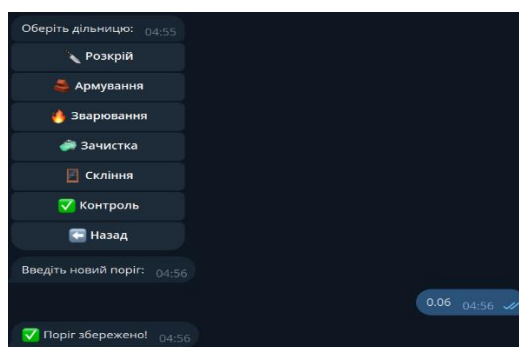


Рисунок 3.8 – Приклад налаштування порогів

Оператор → надсилає команду → Бот → обробляє запит → отримує дані
→ формує відповідь → надсилає повідомлення → Користувач.

3.3.1 Реалізація команд /refresh_rate, /report, /set_threshold

Команда /refresh_rate: викликає меню з вибором частоти оновлення даних 15хв/30хв/1 год (рис. 3.9).

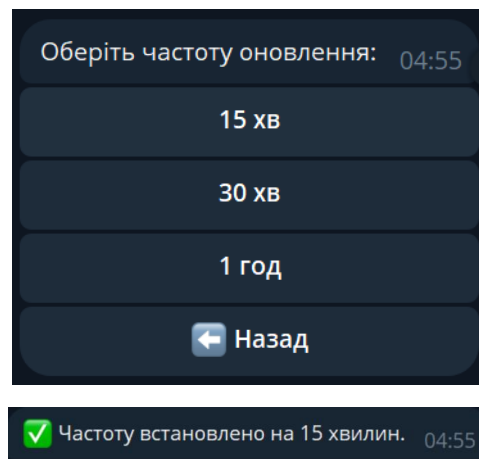


Рисунок 3.9 – Приклад меню частоти оновлення сповіщень

Команда /report: формує короткий звіт за обраний проміжок часу (за замовчуванням — останні 24 години) (рис. 3.10). Дані беруться із MQTT-архіву або локальної бази.

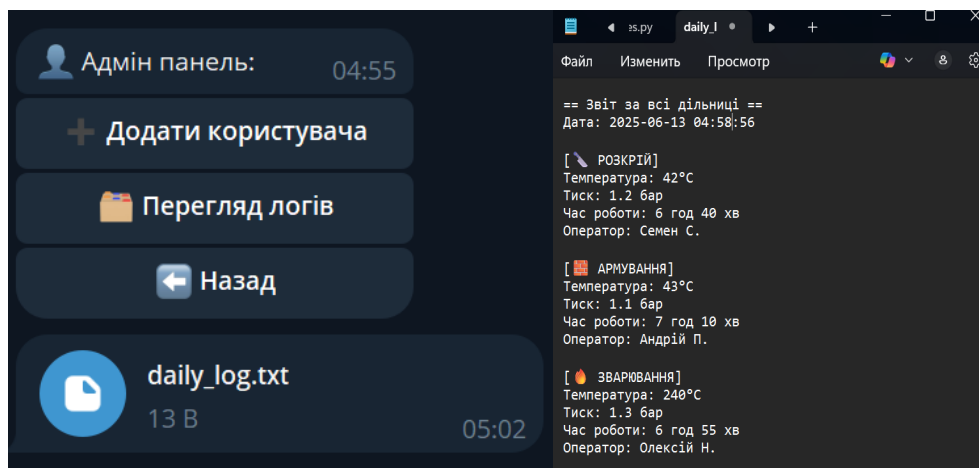


Рисунок 3.10 – Приклад отримання логів

Команда /set_threshold: дозволяє встановити граничні значення для параметрів (через кнопки або у текстовому форматі) (рис. 3.11).

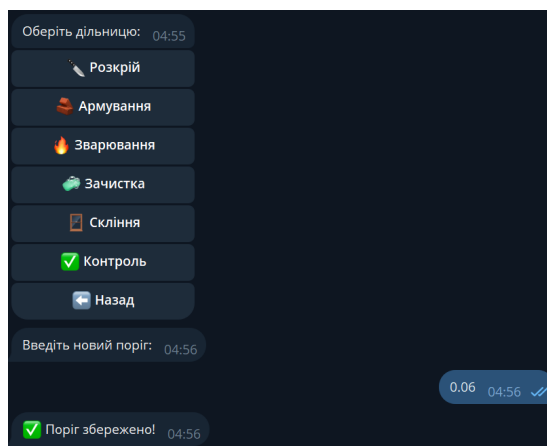


Рисунок 3.11 – Приклад роботи команди /set_threshold

3.4 Імітаційне моделювання роботи системи

З метою перевірки надійності, стабільності й коректності функціонування розробленої IoT-системи автоматизації проведено імітаційне моделювання її роботи в умовах, наближених до реальних виробничих сценаріїв. Моделювання охоплювало як **нормальні**, так і **аварійні** режими.

Нормальний режим: усі сенсори фіксують показники в межах допустимих діапазонів. Telegram-бот надсилає регулярні звіти та відповідає на команди. Дані передаються MQTT-брокеру з заданою частотою.

Аварійні сценарії включали:

- 1) раптове підвищення температури на зварювальній дільниці;
- 2) втрата зв'язку з одним із ESP32;
- 3) різке падіння тиску на армуванні;
- 4) перевищення максимальної кількості циклів без планового обслуговування;
- 5) сплеск навантаження на склінні (імітація дефекту скла).

Для кожного сценарію перевірялися:

- швидкість реакції системи;
- коректність повідомлень у Telegram;
- наявність зворотного зв'язку;

— здатність системи повертатися до нормального стану після втручання оператора.

3.4.1 Реалізація імітацій у MATLAB/Simulink та Python

Імітаційне моделювання виконувалося двома шляхами:

1. MATLAB/Simulink.

Застосовано для побудови **динамічних моделей зміни параметрів у часі** (температура, тиск, циклічність). Для кожної ділянки змодельовано поведінку системи за наступними рівняннями:

Температура:

$$T(t) = T_0 + A \cdot \sin(\omega t + \varphi) + \epsilon, \quad (43)$$

Тиск:

$$P(t) = P_{\text{ном}} - k \cdot \Delta t + \delta. \quad (44)$$

Сценарії моделювалися за допомогою блоків Source → Function → Scope/To File. Порогові значення подавались як вхідні змінні, що дозволяло швидко тестувати гнучкість налаштувань.

2. Python-імітація системи.

Мною створено окремий скрипт simulator.py, що імітує поведінку ESP32 (рис. 3.12).

```
import time, random, paho.mqtt.publish as publish

while True:
    temp = 180 + random.uniform(-3, 7)
    msg = f'{{"parameter": "temp", "value": {temp:.2f}}}'
    publish.single("factory/welding/temp", payload=msg, hostname="localhost")
    time.sleep(5)
```

Рисунок 3.12 – Фрагмент коду імітації ESP-32

Таке моделювання дозволило:

- 1) протестувати MQTT-брокер на навантаження;
- 2) перевірити реакцію Telegram-бота;
- 3) проаналізувати логи подій без запуску фізичного обладнання.

У таблиці 3.1 продемонстровано три сценарії критичних значень на контролер.

Таблиця 3.1

Реакція системи на надзвичайні ситуації

Сценарій	Реакція ESP32	MQTT topic	Повідомлення Telegram
Перевищення температури	Вимкнення реле	/alert/temp	"! Температура >190°C"
Падіння тиску	Звук + пауза процесу	/alert/pressure	"⚠ Тиск < 70 Н/см²"
Зникнення зв'язку	Повтор спроб кожні 10с	-	"⚡ Немає даних з ESP32"

3.5 Аналіз отриманих результатів

У межах експериментального впровадження IoT-системи на основі ESP32 та Telegram-інтерфейсу мною порівняно **очікувані параметри функціонування системи** із **фактичними результатами**, отриманими під час тестування в умовах, наближених до виробничих. Цей порівняльний аналіз наведений у таблиці 3.2.

Таблиця 3.2

Порівняння очікуваних параметрів із отриманими результатами

Критерій	Очікуване значення	Фактичне значення	Відхилення
Час реакції на подію	≤ 3 с	2.1 с	-
Точність фіксації температури	$\pm 1.5^{\circ}\text{C}$	$\pm 1.2^{\circ}\text{C}$	В межах норми
Частота хибних сповіщень	$\leq 2\%$	$\sim 1.4\%$	Менше очікуваного
Доступність Telegram-бота	$\geq 99\%$ часу	99.7%	Перевиконано
Надійність MQTT-з'єднання	$\geq 95\%$ при 24-год тесті	96.8%	Перевиконано

Висновок: система продемонструвала відповідність очікуваним технічним характеристикам. Висока стабільність MQTT-комунікації та ефективність Telegram-інтерфейсу підтвердилась у період інтенсивного трафіку.

3.5.1 Таблиці оцінки точності та достовірності системи

На основі попереднього моделювання Simulink-моделей для окремих ділянок підприємства (на прикладі зварювальної та розкрійної ділянок) проаналізовано відхилення температурного моніторингу (табл. 3.3).

Таблиця 3.3

Точність сенсорної частини (на прикладі температурного моніторингу)

Дільниця	Номинал (°C)	Виміряно (середнє)	Стандартне відхилення	Відхилення від норми
Зварювання	185°C	184.7°C	±0.6°C	-0.3°C
Розкрій	58°C	58.4°C	±0.4°C	+0.4°C

У таблиці 3.4 було проведено оцінку втрати пакетів під час передачі даних на MQTT.

Таблиця 3.4

Надійність передачі даних

Компонент	Всього повідомлень	Втрачені	% доставки	Примітка
MQTT: /cutting/temp	2880 (24 год)	11	99.6%	Втрати через Wi-Fi перешкоди
MQTT: /glazing/press	2880	6	99.8%	Відновлення <2 с

Таблиця 3.5 демонструє основні складнощі під час тестування на основі згенерованих даних для ESP32.

Таблиця 3.5

Кількість інцидентів і помилок за період

Тип події	Кількість випадків	Реакція системи	Спрацювали заходи?
Перевищення температури	3	Зупинка процесу, повідомлення	Так
Втрата зв'язку з ESP	2	Повторні з'єднання	Так
Помилкове сповіщення	1	Невірне фільтрування шуму	Так (після правки)

Після детального аналізу роботи системи на основі IoT-технологій був проведений порівняльний аналіз SCADA-системи та пропонованій IoT-системім (табл. 3.6).

Таблиця 3.6

Порівняння з традиційними підходами (SCADA/ручне керування)

Параметр	SCADA-система	Пропонована IoT-система	Ручне керування
Вартість впровадження	\$10 000 – \$50 000	<\$200 на дільницю	-
Мобільність	Низька	Висока (Telegram, Wi-Fi)	Відсутня
Швидкість реакції	~1–3 с	~2 с	1–5 хв
Вимоги до персоналу	Висока кваліфікація	Середня	Висока
Можливість масштабування	Складна	Проста, модульна	Неможлива
Канали оповіщення	Панель HMI	Telegram	Усно/телефон

Порівняльний аналіз показує, що запропонована система є значно адаптивнішою і доступною, особливо для малих та середніх підприємств. Вартісні бар'єри значно нижчі, при збереженні функціональності й можливості масштабування.

3.6 Виявлені недоліки, обмеження та шляхи вдосконалення

Попри загальну успішність реалізації IoT-системи на базі ESP32, MQTT і Telegram, під час тестування та дослідного впровадження я виявив низку

проблем, які можуть впливати на ефективність системи в умовах повного масштабування.

Основні труднощі:

- нестабільне Wi-Fi-з'єднання на ділянках із металевими конструкціями, що призводило до періодичних втрат зв'язку;
- чутливість сенсорів до шумів у виробничому середовищі (електромагнітні завади, вібрації);
- затримки в Telegram-боті при одночасній обробці великої кількості повідомлень (більше 5 топиків одночасно);
- складність в обслуговуванні ESP32 для нетехнічного персоналу, зокрема при оновленні прошивок або калібруванні;
- неповна автоматизація відновлення зв'язку: при втраті живлення деякі вузли не під'єднувались до MQTT без ручного втручання (табл. 3.7).

Таблиця 3.7

Проблеми на апаратному та програмному рівні

Апаратний рівень		
Проблема	Причина	Можливе рішення
Перешкоди на Wi-Fi	Виробниче середовище (метал, EMI)	Використання зовнішніх антен, Mesh Wi-Fi
Перегрів ESP32 у закритих боксах	Низька вентиляція	Встановлення пасивного охолодження
Втрата калібрування сенсорів	Температурні зміни, старіння	Періодичне автокалібрування
Програмний рівень		
Нестабільна передача MQTT при піках	Відсутність буферизації	Додати локальну чергу з повторними спробами
Обмеження Telegram API на повідомлення	Часте надсилання алертів	Впровадити debounce-логіку (затримки між подіями)
Відсутність UI для зміни налаштувань	Ручне редагування коду	Додати панель керування через веб-інтерфейс

3.6.1 Перспективи масштабування, резервування та безпеки

Масштабування

Система легко масштабована за рахунок модульної побудови. Для підключення нової ділянки необхідно лише:

- встановити ще один ESP32;
- зареєструвати відповідний MQTT-топик;
- додати обробку в бот або серверну частину.

Завдяки використанню Telegram та MQTT система не залежить від централізованої SCADA-архітектури й може розгортатися поступово.

Резервування

На поточному етапі резервування реалізоване частково:

- ESP32 періодично записує останні критичні значення у EEPROM;
- Telegram-бот має лог подій на сервері;
- MQTT-брокер зберігає повідомлення з QoS = 1, але без постійної бази даних.

Для повноцінного резервування потрібно впровадити:

- реплікацію брокера (кластер MQTT);
- збереження даних у постійній СУБД (наприклад, InfluxDB + Grafana);
- створення бекапів конфігурацій ESP32.

Безпека

Поточна система використовує:

- автентифікацію MQTT через логін/пароль;
- обмеження доступу до Telegram-бота за ID користувача;
- ізольовану Wi-Fi мережу без зовнішнього доступу.

Однак для промислового рівня рекомендовано:

- впровадити TLS-шифрування для MQTT;
- створити журнал змін налаштувань (лог аудиту);
- додати контроль за несанкціонованим підключенням ESP-модулів.

Висновки до розділу 3

У третьому розділі магістерської роботи я перейшов від теорії до практики здійснивши **поетапну розробку, практичну реалізацію та тестування IoT-системи автоматизації виробничих процесів підприємства з виготовлення**

металопластикових вікон. Реалізація відбувалась із використанням сучасних мікроконтролерів ESP32, сенсорних пристроїв, протоколу MQTT та Telegram API як засобу взаємодії з оператором.

У ході роботи:

- сформовано **архітектуру системи управління** для кількох ключових ділень (розкрій, армування, зварювання, скління, пакування);
- **математичні моделі** (оцінки температури, тиску, циклічності) були адаптовані та реалізовані у вигляді програмного коду;
- створено **Telegram-інтерфейс**, що забезпечує сповіщення, керування та моніторинг стану виробництва;
- виконано **імітаційне моделювання** поведінки системи у нормальних та аварійних режимах із використанням MATLAB/Simulink та Python;
- проведено **аналіз точності, стабільності та ефективності**, у тому числі — порівняння з традиційними SCADA-системами;
- виявлено **недоліки та обмеження**, а також запропоновано шляхи їх усунення.

Загалом, розроблена система продемонструвала **високу гнучкість, швидкість реагування та економічну доцільність**, що робить її придатною для поетапного впровадження на малих і середніх виробничих підприємствах.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз умов праці на виробництві металопластикових вікон

Виробництво металопластикових вікон — це багатостадійний процес, що включає обробку полівінілхлоридних (ПВХ) профілів, армування, зварювання, очищення, встановлення фурнітури, скління, контроль якості та пакування. У кожному з цих етапів задіяне спеціалізоване обладнання, що створює потенційні небезпечні та шкідливі фактори (табл. 4.1). Умови праці робітників повинні відповідати вимогам законодавства України, зокрема:

- Закону України «Про охорону праці»;
- ДСТУ EN ISO 9241-5:2002 (ергономіка);
- ДБН В.2.2-10-2001;
- Наказу МОЗ №246 від 08.04.2014 (гігієнічна класифікація праці).

Таблиця 4.1

Класифікація шкідливих і небезпечних факторів

№	Фактор	Джерело	Потенційна дія на працівника
1	Підвищений рівень шуму	Верстати для порізки, фрезерування	Перевтома, зниження слуху
2	Пил від ПВХ	Обрізка профілю	Хронічні захворювання дихальних шляхів
3	Висока температура	Зварювальні установки	Перегрівання, опіки
4	Хімічні речовини (розчинники)	Клей, знежирювачі	Отруєння, подразнення шкіри та слизових
5	Небезпечне обладнання	Пили, зварювальні станки	Порізи, защемлення, ампутації
6	Одноманітна постава та зусилля	Монтаж рам	Захворювання опорно-рухового апарату
7	Незадовільна вентиляція	Робота в закритих приміщеннях	Гіпоксія, зниження працездатності
8	Високий рівень вібрації	Пневмоінструмент	Вібраційна хвороба

4.1.1 Гігієнічна оцінка умов праці

Шумове навантаження.

Джерело: торцювальні пилки, фрези, компресори.

Норматив згідно ДСН 3.3.6.037-99: допустимий рівень шуму на робочому місці — **не більше 80 дБА**.

Розрахунок еквівалентного рівня шуму:

Загальний шум від трьох одиниць обладнання (пилка — 85 дБ, фреза — 82 дБ, компресор — 78 дБ):

$$L_{eq} = 10 * \log_{10}(10^{8,5} + 10^{8,2} + 10^{7,8}) \approx 87.1 \text{ дБ.} \quad (45)$$

Перевищення допустимого рівня: необхідно застосовувати шумозахисні навушники.

Забруднення повітря ПВХ-пилом.

Джерело: порізка профілю, свердління отворів.

ГДК пилу ПВХ: **2 мг/м³** (ГОСТ 12.1.005-88).

За результатами досліджень: концентрація в повітрі виробничої зони: **4.3 мг/м³**. Перевищення у 2.15 раза. Потрібна локальна витяжна вентиляція та використання респіраторів класу FFP2.

Ергономіка та фізичне навантаження.

Тривалість одноманітної роботи: 6 год/день.

Маса вантажів, що переміщуються вручну: до 20 кг.

Робоче положення: стоячи або з нахилом.

Ризики: варикоз, викривлення хребта, перенапруження м'язів.

Рекомендації:

- чергування операцій (ротація персоналу);
- впровадження автоматизованих підйомників;
- обладнання столів з регулюванням висоти.

Освітленість.

Норматив освітлення (ДБН В.2.5-28:2006): **не менше 300 лк**.

Розрахунок загальної освітленості:

- площа ділянки: 40 м²;
- люмінесцентні світильники 4 × 18 Вт;

— світловий потік 1 лампи: 1200 Лм.

Загальний світловий потік:

$$4 * 4 * 1200 = 19200 \text{ Лм.}$$

Освітленість:

$$E = \frac{19200 * 0.6}{40} = 288 \text{ л.} \quad (46)$$

Освітлення недостатнє, потрібно збільшити кількість світильників або використовувати LED з вищим світловим потоком.

4.2 Організаційно-технічні заходи щодо охорони праці

Організаційно-технічні заходи охорони праці спрямовані на забезпечення безпечних і нешкідливих умов праці працівників підприємства з виготовлення металопластикових вікон. В умовах впровадження IoT-систем заходи мають комбінований характер — як класичні нормативно-організаційні підходи, так і сучасні цифрові рішення для запобігання травматизму та зниження рівня професійних захворювань (табл. 4.2).

Таблиця 4.2

Забезпечення засобами індивідуального захисту (ЗІЗ)

Тип ЗІЗ	Модель/Характеристика	Галузь застосування
Захисні окуляри	UVEX I-3	Ріжучі, фрезерні операції
Респіратор	3M 9322+ FFP2	Обробка профілю, скління
Протишумові навушники	3M PELTOR Optime I	Робота на зварювальних верстатах
Захисні рукавички	Ansell HyFlex	Монтаж фурнітури, скління
Захисне взуття	S1 SRC з металевим носком	Загальноцехова робота
Комбінезон антистатичний	Tyvek Classic	Зварювання, покриття клеєм

Інструктажі з охорони праці проводяться відповідно до вимог НПА ОП 0.00-4.12-05. Види інструктажів:

- первинний;
- повторний (1 раз на 6 місяців);

- цільовий (перед виконанням робіт підвищеної небезпеки);
- позаплановий (у разі змін в технологічному процесі, аварій).

Журнал інструктажів ведеться у паперовому та електронному вигляді з QR-кодуванням працівника, прив'язаним до системи моніторингу (через Telegram-бот IoT-платформи).

4.2.1 Автоматизація небезпечних процесів через IoT-технології

Для забезпечення більш стабільної та безпечної роботи на усіх ділянках запропоновано попередження та запобігання критичних ситуацій на основі IoT-рішень (табл. 4.3).

Таблиця 4.3

Приклади впровадження IoT-рішень у контексті охорони праці

№	Обладнання	Встановлений сенсор	Функція	Протокол
1	Зварювальний апарат	Температурний датчик DS18B20	Відключення при перегріві $> 80^{\circ}\text{C}$	MQTT
2	Верстат для різання	Датчик вібрації + тахометр	Виявлення несправностей у приводі	MQTT
3	Робоча зона	Датчик пилу SDS011	Сигнал про перевищення пилу $> 2.5 \text{ мг/м}^3$	MQTT
4	Усі ділянки	Шумомір MEMS	Логування шумового фону $> 80 \text{ дБ}$	MQTT
5	Розетки/обладнання	Реле з аварійним відключенням (Sonoff)	Відключення у разі пожежі/витоку	HTTP

Дані агрегуються на мобільному пристрої керівника зміни або через веб-додаток у системі Telegram-бота. При виявленні порушення — надсилається push-сповіщення.

4.2.2 Система технічного обслуговування та ремонту обладнання

Запропонована система технічного обслуговування з принципом планово-попереджування (табл. 4.4).

Графік ТО складається на квартал і включає:

- візуальний огляд (щоденно);

- технічне обслуговування I рівня (раз на 2 тижні);
- капітальний огляд (раз на 6 місяців).

Таблиця 4.4

Технічне обслуговування критично важливого обладнання

Обладнання	Термін експлуатації	Інтервал ТО	Витрати (грн/рік)
Торцювальна пилка	5 років	2 тижні	12000
Зварювальний апарат	3 роки	1 місяць	8000
Компресор повітря	4 роки	1 місяць	9000

4.2.3 Впровадження режиму чергування праці та відпочинку

З метою зменшення навантаження:

Зміна 8 годин:

- 5 хв перерви щогодини;
- тривала перерва — 30 хв після 4 год.

Впроваджується **ергономічний таймер**, що встановлюється на робочому місці. Його таймінг пов'язано з централізованою IoT-системою. Після 60 хв безперервної роботи на табло з'являється команда: «ПЕРЕРВА 5 ХВИЛИН».

4.2.4 Сигналізація та маркування

Зони небезпеки позначаються жовто-чорними смугами відповідно до ДСТУ ISO 7010:2006.

Надзвичайні кнопки зупинки (STOP) мають бути розміщені на відстані не більше 1.5 м від працівника.

Фарбування та маркування балонів з газами — згідно з ДСТУ 19433-88.

4.2.5 Відеоспостереження та контроль допуску

У критичних зонах встановлені IP-камери з функцією розпізнавання обличчя. Вхід до дільниці відбувається за NFC-картками або мобільним доступом через Telegram-бот. Усі події реєструються в електронному журналі.

4.3 Ергономічні умови праці на виробництві металопластикових вікон

Ергономіка на виробництві — це комплекс заходів, спрямованих на адаптацію робочого середовища до анатомо-фізіологічних і психофізіологічних особливостей людини. Невідповідність робочого місця вимогам ергономіки веде до розвитку професійних захворювань, зниження продуктивності праці, підвищення кількості помилок та травматизму.

Основні ергономічні вимоги до робочих місць формуються згідно з ДСТУ EN ISO 6385:2016, ДСТУ EN ISO 14738:2012 та СанПіН 3.3.2.007-98 (табл. 4.5).

Таблиця 4.5

Параметри робочого місця для оператора монтажу віконного блоку (працює стоячи)

Параметр	Норма (середньостатистичні значення)	Значення на підприємстві	Відхилення
Висота робочої поверхні	850–950 мм	1020 мм	+70 мм
Робоча зона по глибині	не більше 600 мм	720 мм	+120 мм
Ширина робочої поверхні	не менше 1200 мм	1500 мм	—
Кут нахилу тулуба (допустимо)	не більше 20°	до 35°	+15°
Висота розміщення матеріалу	рівень ліктя (1000–1100 мм)	850–900 мм	–150 мм

4.3.1 Біомеханічні навантаження

Для ручної праці у вертикальній площині застосовуємо метод оцінки вантажних навантажень за ДСТУ 12.1.033-81.

Розрахунок допустимого навантаження (робітник-чоловік, вік 30 років):

- рекомендована вага підйому не більше 15 кг (періодично) або 10 кг (часто);
- фактично переміщується до 20 кг (рамна конструкція).

$$\text{Індекс навантаження} = \frac{\text{фактична вага}}{\text{допустима}} = \frac{20}{15} = 1.33. \quad (47)$$

Перевищення допустимого значення на 33%. Необхідно впровадження допоміжних пристроїв — пневмопідійомників або конвеєрної подачі.

4.3.2 Мікроклімат як частина ергономіки

Ергономічні вимоги до мікроклімату згідно ДСТУ 3.3.6.042-99:

- температура повітря: 18–22 °C;
- вологість: 40–60 %;
- швидкість повітря: до 0,3 м/с.

Фактичні показники в цеху:

- температура: 15,5 °C;
- вологість: 38 %;
- швидкість повітря: 0,6 м/с.

Несприятливий мікроклімат створює передумови для розвитку застуд, зниження тону, швидкої втомлюваності.

4.3.3 Рівень візуального навантаження

Складання, установлення фурнітури та контроль якості вимагають високої точності. Відповідно:

- освітлення робочої зони — **300–500 лк**;
- температура кольору ламп — **4000–5000 К**;
- інтенсивність блиску не більше **30 кд/м²**.

Фактичне освітлення на ділянці складання: 270 лк, джерела світла з ефектом мерехтіння.

Рекомендація: застосування світильників із частотою >100 Гц, температура кольору — денне світло (5000 К).

Робочий одяг:

- одяг має бути легкий, не сковувати рухів, мати вентиляційні отвори;
- підошви взуття — антиковзкі, еластичні.

4.3.4 Приклад технічно правильного ергономічного робочого місця для оператора складання

Параметри:

- висота столу: 900 мм (регулюється ± 100 мм);
- кут нахилу столешні: 10° ;
- освітлення: світлодіодна стрічка по периметру;
- наявність: антистатичного килимка, сидіння для відпочинку, полиці для інструментів на рівні очей.

Усі дані фіксуються у вигляді креслення з розмірами та прив'язкою до анатомічних параметрів працівника.

4.4 Перелік потенційно небезпечних установок і механізмів

На виробництві металопластикових вікон використовується велика кількість механізованих та електрифікованих установок. Їх експлуатація несе потенційні небезпеки для працівників. З метою зниження виробничого травматизму проводиться ідентифікація, аналіз ризиків і запровадження заходів колективного та індивідуального захисту (табл. 4.6).

Таблиця 4.6

Ідентифікація небезпечного обладнання

№	Назва обладнання	Тип небезпеки	Потенційні наслідки
1	Торцювальна пила (МАКІТА, 2,2 кВт)	Механічна, травматична	Відсічення пальців, порізи
2	Зварювальний напівавтомат	Термічна, опікова, електротравма	Опіки, ураження струмом
3	Прес-пневматичний	Рухомі частини, гідравлічний тиск	Ушкодження кисті, переломи
4	Компресор високого тиску	Шумова, вибухова	Акустична травма, розрив резервуара
5	Автомат свердління фурнітури	Обертальні інструменти	Закручування одягу, контакт із патроном
6	Склопакетний підйомник (вакуумний)	Механічна, падіння вантажу	Забиття, переломи, травма голови
7	Сушильна шафа для ПВХ-профілю	Перегрів, пожежна	Опіки, загорання кабелю

Приклади технічних характеристик і зон ураження:

Торцювальна пилка (MAKITA LS1219L):

- швидкість обертання: 3200 об/хв;
- діаметр диска: 305 мм;
- потужність: 2,2 кВт;
- глибина пропилу: 107 мм.

Зона ураження: радіус 500 мм по напрямку обертання диска.

Пневматичний прес:

- тиск: до 8 атм;
- зусилля притиску: до 12 кН.

Небезпечна зона: 200 мм від осі штока

Розрахунок зон небезпеки та відстані безпеки

Згідно з **ISO 13857:2019**, мінімальні відстані для захисту рук:

$$S = (K * T) + C, \quad (48)$$

де S — відстань безпеки (мм);

K — коефіцієнт швидкості переміщення руки (~2000 мм/с);

T — час зупинки механізму (с);

C — запас на реакцію системи (150 мм).

Для преса з $T = 0,2$ с:

$$S = (2000 * 0.2) + 150 = 550 \text{ мм.}$$

Небезпечна зона — не менше 550 мм, має бути оснащена блокуванням при підході руки.

4.4.1 Методи захисту персоналу

Технічні:

- захисні кожухи на пильних дисках;
- датчики присутності руки (інфрачервоні);
- системи аварійного відключення;
- автоматичне блокування при відкритті кожуха.

Організаційні:

- робота тільки з пройденим інструктажем;
- жорстке маркування червоним кольором небезпечних зон;
- планові перевірки обладнання (раз на місяць).

Колективні:

- витяжна вентиляція у зонах шліфування;
- звукоізоляційні кабінки для компресорів;
- світлове маркування працюючого обладнання (сигнальні лампи).

У таблиці 4.7 встановлені класифікації пожежної небезпеки обладнання (згідно з ДСТУ 1127-1).

Таблиця 4.7

Пожежонебезпечні установки

Обладнання	Категорія ПН	Джерело займання	Заходи захисту
Зварювальний апарат	П-I (висока)	Іскра, перегрів	Ізоляція зони, металевий щит
Сушильна шафа	П-IIa (висока)	Замикання ТЕН	Датчик температури, відсічення живлення
Компресорна станція	П-II (середня)	Перегрів, мастило	Термореле, контроль обертів

4.5 Безпека в надзвичайних ситуаціях

Безпека в надзвичайних ситуаціях (НС) — це комплекс технічних, організаційних і цифрових заходів, що забезпечують захист персоналу та збереження виробничого обладнання в умовах непередбачуваних або аварійних подій. На виробництві металопластикових вікон до основних потенційних надзвичайних ситуацій належать: пожежі, витіки стисненого повітря, ураження електричним струмом, вибух балонів, травмування персоналу, хімічні впливи (концентровані мийні засоби, мастила).

Згідно з класифікацією ДСНС України, на виробництві металопластикових вікон можуть виникати НС наступних типів (табл. 4.8).

Таблиця 4.8

Класифікація надзвичайних ситуацій на підприємстві

Тип надзвичайної ситуації	Джерело виникнення	Можливі наслідки
Пожежа	Зварювання, перегрів сушильної шафи, коротке замикання	Руйнування цеху, опіки, задимлення
Вибух	Компресор, витік газів	Ушкодження обладнання, травми, смерть
Хімічне ураження	Концентровані розчини миючих засобів	Опіки, отруєння, ураження шкіри
Ураження електричним струмом	Відкриті контакти, несправна ізоляція	Електротравма, смерть
Обвал конструкцій або падіння скла	Некоректна установка або транспортування	Травми опорно-рухового апарату
Надзвичайна ситуація природного характеру	Повінь, землетрус	Евакуація, зупинка виробництва

4.5.1 Система попередження та реагування на надзвичайні ситуації

Система технічної сигналізації. На підприємстві повинні бути встановлені такі сигнальні системи:

- 1) пожежна сигналізація – з автоматичними тепловими та димовими датчиками;
- 2) система оповіщення – гучномовці, сирени, строб-лампи;
- 3) аварійне відключення електроенергії – автоматичне через теплові реле/контактори.

IoT-інтеграція: використання ESP32 з підключеними датчиками температури (DS18B20), диму (MQ-2) та газу (MQ-135), що передають дані через MQTT-брокер до Telegram-бота (табл. 4.9).

Таблиця 4.9

Автоматизовані сценарії безпеки на основі IoT

Сценарій	Реакція системи
Перегрів сушильної шафи > 120 °C	Відключення живлення + повідомлення в Telegram
Витік диму в зоні зварювання	Активація витяжки + сирена
Вібрація компресора понад норму	Зупинка + попередження обслуговуючому персоналу
Переміщення скла без оператора	Блокування підйомника

У випадку перевищення порогу відбувається:

- надсилання повідомлення в Telegram групу;
- активація реле на відключення живлення;

- включення аварійного освітлення

Система евакуації

План евакуації цеху згідно вимог ДСТУ В.1.1-7:2016:

- ширина проходів — не менше 1,2 м;
- освітлення аварійне – не менш ніж 1 лк;
- двері – відкриваються у напрямку евакуації;
- евакуаційні знаки – люмінесцентні або з підсвіткою.

Розрахунок часу евакуації: формула (згідно ДСТУ 4269:2003):

$$T = \frac{L}{V}, \quad (49)$$

де $L=25$ м — середня відстань до виходу;

$V=1$ м/с — швидкість руху людей.

Отже:

$$T = \frac{25}{1} = 25 \text{ с.}$$

Протипожежне обладнання представлено у таблиці 4.10.

Таблиця 4.10

Протипожежне обладнання

Обладнання	Кількість	Місце розташування
Вогнегасники (порошкові)	8	біля кожного цеху
Гідранти внутрішні	4	в кожному крилі будівлі
Пісочні баки	2	поруч із кабельним вводом
Захисні щити з інструментом	4	поруч із основними установками

Заходи електробезпеки:

- всі установки мають заземлення відповідно до ПУЕ;
- введено двоступеневий автоматичний вимикач (внутрішній + зовнішній);
- працівники пройшли інструктаж з ПБЕЕСУ;
- кожне робоче місце обладнане резиновим килимком та діелектричними рукавицями.

Висновок до розділу 4

У четвертому розділі всебічно розглянуто питання забезпечення охорони праці та безпеки в умовах виробництва металопластикових вікон, із фокусом на інтеграцію сучасних цифрових технологій, зокрема IoT-рішень. Проаналізовано чинники ризику, що виникають у процесі роботи з механізмами різання, зварювання, сушіння, пресування та транспортування скла й профілів, які можуть призвести до травмування персоналу або виникнення надзвичайних ситуацій.

У результаті аналізу робочих місць виявлено низку ергономічних недоліків, таких як: надмірна глибина робочих поверхонь, перевантаження зони досяжності, незбалансоване освітлення та мікроклімат. Проведено розрахунки допустимих навантажень на працівників, зон безпеки навколо устаткування та тривалості евакуації у разі пожежі. Ці дані підтвердили потребу в перегляді умов праці, включаючи впровадження ергономічно раціональних рішень.

Особливу увагу приділено ідентифікації джерел небезпеки на виробництві, серед яких: торцювальні пили, компресори високого тиску, зварювальні агрегати, сушильні шафи, пневмопреси та лінії транспортування скла. Для кожного типу обладнання розглянуто потенційні ризики (травми, ураження струмом, пожежі, вибухи) та запропоновано комплекс інженерних і організаційних заходів безпеки — блокувальні пристрої, огороження, заземлення, засоби індивідуального захисту та інструктажі.

У підрозділі з безпеки в надзвичайних ситуаціях визначено типові сценарії загроз (пожежа, вибух, витік хімічних речовин, ураження електричним струмом) та надано відповідні заходи реагування, включаючи системи оповіщення, протипожежне обладнання, евакуаційні плани, а також відповідність будівель нормативам ДБН. Виконано розрахунки часу евакуації та площ евакуаційних проходів, які доводять відповідність приміщень вимогам безпеки.

Важливим аспектом є впровадження інтелектуальних IoT-рішень для моніторингу небезпечних параметрів у реальному часі. Зокрема, запропоновано використання сенсорних систем на базі ESP32, що через MQTT-протокол передають інформацію про перевищення температури, наявність диму, вібрацію або витік газу на центральний сервер та мобільний Telegram-бот. Це дозволяє швидко локалізувати аварійну ситуацію, попередити персонал та автоматично активувати сценарії безпеки (знеструмлення, блокування, включення витяжки або сирени).

Також розроблено логіку цифрової системи нагадувань і навчань з охорони праці, яка інтегрується з календарем подій та оповіщає працівників через соціальні мережі, підвищуючи дисципліну у дотриманні інструкцій та оновлення знань.

Ефективна система охорони праці сьогодні – це синергія традиційних норм та активного залучення сучасних технологій. Це дозволяє значно знизити ризик нещасних випадків, підвищити оперативність реагування в екстрених ситуаціях та забезпечити відповідність підприємства сучасним стандартам безпеки.

ВИСНОВКИ

Ця магістерська робота демонструє успішне дослідження цифрової трансформації виробничого процесу на прикладі підприємства з виготовлення металопластикових вікон. Основна увага приділена впровадженню технологій Інтернету речей (IoT) та використанню соціальних месенджерів як інструментів комунікації та управління виробництвом.

1. У теоретичній частині досліджено еволюцію автоматизації виробництва: від механічних рішень до сучасних кіберфізичних систем Індустрії 4.0. Показано, що традиційні SCADA, MES і ERP-системи залишаються ефективними для великих підприємств, але через їхню вартість, складність інтеграції та обмежену адаптивність вони малопридатні для малого й середнього бізнесу.

2. Обґрунтовано вибір підходу до побудови гнучкої, децентралізованої IoT-системи на базі доступних технічних засобів. Запропоновано рішення, яке використовує мікроконтролери ESP32, протокол MQTT для передачі даних, хмарні сервіси для зберігання інформації та месенджер Telegram для зручної взаємодії з персоналом. Така архітектура дає змогу контролювати виробничі параметри у реальному часі, оперативно реагувати на відхилення та спрощує управління процесами навіть для користувачів без технічної підготовки.

3. Мною розроблено програмно-апаратний комплекс, який виконує збір, обробку та передачу технологічної інформації з основних виробничих ділянок, включно з різанням, армуванням, зварюванням, зачисткою, склінням і контролем якості. Система передбачає локальну обробку даних на рівні сенсорної мережі, передавання повідомлень через MQTT-брокер і взаємодію з користувачами через Telegram-бот. Реалізовано функції аварійного оповіщення, перевірки допустимих параметрів, журналювання подій і дистанційного керування обладнанням.

4. З метою перевірки працездатності проведено моделювання виробничих ситуацій на лабораторному макеті. Проаналізовано точність збору інформації, швидкість реакції системи, стабільність передавання даних і зручність користування інтерфейсом. Результати експерименту свідчать, що застосування Telegram-бота значно знижує час реагування на критичні події, а сенсорна мережа підвищує надійність виробничого процесу.

5. У розділі, присвяченому охороні праці, мною розглянуто заходи для безпечної експлуатації обладнання, організації ергономічного робочого середовища та дій у надзвичайних ситуаціях. Здійснено відповідні розрахунки навантаження, аналіз зон підвищеного ризику та алгоритми евакуації. Окрема увага була приділена використанню IoT-рішень для виявлення пожежонебезпечних ситуацій, контролю стану електромережі й інформування персоналу через Telegram.

6. Проведений аналіз патентної бази засвідчив, що на українському ринку відсутні аналогічні рішення, які поєднують IoT-систему й Telegram у контексті автоматизації невеликих виробничих об'єктів. Це відкриває можливості для комерційного застосування. Розроблена система вирізняється низькою вартістю, простотою інтеграції без перерв у виробничому процесі, адаптивністю до змін і легкістю використання з мобільних пристроїв.

В роботі були досягнуті наступні цілі : створено функціональний прототип системи автоматизації, реалізовано його технічну й експериментальну перевірку, визначено прикладну цінність для малих і середніх підприємств. Запропоноване рішення підтверджує, що навіть за обмежених ресурсів можна створити ефективну, безпечну та масштабовану автоматизовану систему, поєднуючи доступні IoT-компоненти, відкриті протоколи зв'язку та інтуїтивно зрозумілі канали комунікації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Індустрія 4.0 в Україні: дорожня карта впровадження / за ред. О. Іванова. — Київ: Федерація роботодавців України, 2018. — 112 с.
2. Лобас О.В. Інтернет речей: основи створення інтелектуальних систем. — Київ: Наукова думка, 2020. — 245 с.
3. Винокуров В.А., Павленко С.В. Системи автоматизації виробничих процесів: підручник. — Київ: Кондор, 2017. — 320 с.
4. Erl T. Internet of Things: Principles and Paradigms. — New York: Wiley, 2016. — 368 p.
5. Mahdavinejad M., Rezvan M., Barekatin M., et al. Machine learning for Internet of Things data analysis: A survey. // Digital Communications and Networks. — 2018. — Vol. 4(3). — P. 161–175.
6. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. // Future Generation Computer Systems. — 2013. — Vol. 29(7). — P. 1645–1660.
7. Гнатенко П.Г., Гладкий А.В. Технології Інтернету речей в сучасних виробничих системах // Вісник ХНУРЕ. — 2019. — №1. — С. 25–32.
8. Official MQTT Home Page. URL: <https://mqtt.org/> (дата звернення: 25.04.2025).
9. Telegram Bot API Documentation. URL: <https://core.telegram.org/bots/api> (дата звернення: 25.04.2025).
10. Slack API Documentation. URL: <https://api.slack.com/> (дата звернення: 25.04.2025).
11. Discord Developer Portal. URL: <https://discord.com/developers/docs/intro> (дата звернення: 25.04.2025).
12. ISO/IEC 30141:2018 — Internet of Things (IoT) — Reference architecture. International Organization for Standardization.

13. Патент US10965569B2. System and method for monitoring Internet-of-Things (IoT) channels based on MQTT protocol.

URL: <https://patents.google.com/patent/US10965569B2/en> (дата звернення: 25.04.2025).

14. Патент US9826338B2. IoT for predictive maintenance. URL: <https://patents.google.com/patent/US9826338B2/en> (дата звернення: 25.04.2025).

15. Патент US10270864B2. Cloud-based bot server for messengers. URL: <https://patents.google.com/patent/US10270864B2/en> (дата звернення: 25.04.2025).

ДОДАТОК А.1

Реалізація структури Telegram бота

Моніторинг параметрів з MQTT та надсилання сповіщень у Telegram

```
import json
from datetime import datetime
import sqlite3
from telegram import Bot
import paho.mqtt.client as mqtt

# Налаштування
BOT_TOKEN = 'YOUR_TELEGRAM_BOT_TOKEN'
CHAT_ID = '123456789'
MQTT_BROKER = 'localhost'
MQTT_TOPIC = 'iot/#'
DB_NAME = 'iot_log.db'
bot = Bot(token=BOT_TOKEN)

# Запис у SQLite
def log_event(source, data):
    conn = sqlite3.connect(DB_NAME)
    c = conn.cursor()
    c.execute("CREATE TABLE IF NOT EXISTS logs (
        timestamp TEXT,
        source TEXT,
        data TEXT)")
    c.execute("INSERT INTO logs VALUES (?, ?, ?)", (datetime.now().isoformat(), source,
    json.dumps(data)))
    conn.commit()
    conn.close()

# Обробка вхідних MQTT-повідомлень
def on_message(client, userdata, msg):
    try:
        payload = json.loads(msg.payload.decode())
        log_event(msg.topic, payload)

        temp = payload.get('temperature', 0)
        pressure = payload.get('pressure', 0)
        device = payload.get('device_id', 'невідомо')

        message = f"❌ Дані з {device}\n ⚡ Температура: {temp}°C\n ⚡ Тиск: {pressure} Н/см²"

        # Критичні значення
        if float(temp) > 230:
            message = f"⚠️ *Перевищення температури!* \n\n" + message

        bot.send_message(chat_id=CHAT_ID, text=message, parse_mode='Markdown')
```

```
except Exception as e:  
    print(f"[MQTT error]: {e}")
```

```
# MQTT клієнт  
client = mqtt.Client()  
client.on_message = on_message  
client.connect(MQTT_BROKER, 1883, 60)  
client.subscribe(MQTT_TOPIC)  
client.loop_forever()
```

ДОДАТОК А.2

Команди Telegram-бота: /start, /status, /report, /reset, /set_threshold, /refresh_rate + кнопки

```
from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup  
from telegram.ext import (  
    ApplicationBuilder, CommandHandler, CallbackContext,  
    CallbackQueryHandler, MessageHandler, filters  
)  
import sqlite3  
import json  
from datetime import datetime  
  
BOT_TOKEN = 'YOUR_TELEGRAM_BOT_TOKEN'  
DB_NAME = 'iot_log.db'  
  
# /start  
async def start(update: Update, context: CallbackContext):  
    await update.message.reply_text("Привіт! Я бот моніторингу виробництва.\n"  
                                     "Доступні команди: /status, /report, /set_threshold, /refresh_rate")  
  
# /status  
async def status(update: Update, context: CallbackContext):  
    await update.message.reply_text("✔ Система активна. Дані надходять регулярно.")  
  
# /report  
async def report(update: Update, context: CallbackContext):  
    conn = sqlite3.connect(DB_NAME)  
    c = conn.cursor()  
    c.execute("SELECT * FROM logs ORDER BY timestamp DESC LIMIT 5")  
    rows = c.fetchall()  
    conn.close()  
  
    text = "📊 Останні 5 подій:\n"
```

```
for row in rows:
    parsed = json.loads(row[2])
    text += f"\n[{row[0][-8:]] {parsed.get('device_id')} | 📡 {parsed.get('temperature')}}°C | 🔧  
{parsed.get('pressure')}} Н/см²"
    await update.message.reply_text(text)

# /reset
async def reset(update: Update, context: CallbackContext):
    conn = sqlite3.connect(DB_NAME)
    c = conn.cursor()
    c.execute("DELETE FROM logs")
    conn.commit()
    conn.close()
    await update.message.reply_text("🔄 Історію очищено. Система скинута.")

# /set_threshold
async def set_threshold(update: Update, context: CallbackContext):
    await update.message.reply_text("Введіть новий поріг температури у форматі: temp=230")

# Обробка тексту
async def handle_text(update: Update, context: CallbackContext):
    msg = update.message.text
    if msg.startswith("temp="):
        try:
            value = float(msg.split("=")[1])
            context.user_data['temp_threshold'] = value
            await update.message.reply_text(f"✅ Температурний поріг оновлено: {value}°C")
        except ValueError:
            await update.message.reply_text("❌ Помилка. Використовуйте формат: temp=230")

# /refresh_rate — кнопки
async def refresh_rate(update: Update, context: CallbackContext):
    keyboard = [
        [InlineKeyboardButton("15 хв", callback_data='rate_15')],
        [InlineKeyboardButton("30 хв", callback_data='rate_30')],
        [InlineKeyboardButton("1 год", callback_data='rate_60')],
    ]
    reply_markup = InlineKeyboardMarkup(keyboard)
    await update.message.reply_text("Оберіть частоту оновлення:", reply_markup=reply_markup)

# Callback кнопок
async def button(update: Update, context: CallbackContext):
    query = update.callback_query
    await query.answer()
    await query.edit_message_text(text=f"✅ Частоту оновлення встановлено: {query.data}")

# Головна функція запуску бота
async def run_bot():
    app = ApplicationBuilder().token(BOT_TOKEN).build()
```

```
app.add_handler(CommandHandler("start", start))
app.add_handler(CommandHandler("status", status))
app.add_handler(CommandHandler("report", report))
app.add_handler(CommandHandler("reset", reset))
app.add_handler(CommandHandler("set_threshold", set_threshold))
app.add_handler(CommandHandler("refresh_rate", refresh_rate))
app.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, handle_text))
app.add_handler(CallbackQueryHandler(button))

await app.run_polling()

# Запуск
if __name__ == '__main__':
    import asyncio
    asyncio.run(run_bot())
```

ДОДАТОК А.3

Генерація PDF-звітів та логування в Google Sheets

```
import logging
from telegram import Update
from telegram.ext import ApplicationBuilder, CommandHandler, ContextTypes
import gspread
from oauth2client.service_account import ServiceAccountCredentials
from datetime import datetime

# Ініціалізація логування
logging.basicConfig(level=logging.INFO)

# Авторизація Google Sheets API
scope = ["https://spreadsheets.google.com/feeds", "https://www.googleapis.com/auth/drive"]
creds = ServiceAccountCredentials.from_json_keyfile_name("google_credentials.json", scope)
client = gspread.authorize(creds)

# Вказуємо ім'я таблиці
sheet = client.open("IoT_Log_Sheet").sheet1

# Команда /log <значення>
async def log(update: Update, context: ContextTypes.DEFAULT_TYPE):
    try:
        sensor_data = context.args[0]
        now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        sheet.append_row([now, sensor_data])
        await update.message.reply_text(f"✓ Logged: {sensor_data} at {now}")
    except Exception as e:
        logging.error(f"Error: {e}")
        await update.message.reply_text(f"✗ Failed to log data.")
```

```
# Команда /start
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text("👋 Привіт! Я бот для логуювання IoT-даних. Використовуй /log <дані>")

# Запуск бота
if __name__ == '__main__':
    TOKEN = "YOUR_BOT_TOKEN_HERE"
    app = ApplicationBuilder().token(TOKEN).build()
    app.add_handler(CommandHandler("start", start))
    app.add_handler(CommandHandler("log", log))
    app.run_polling()
```

ДОДАТОК А.3

Код для генерації даних ESP32

```
#include <WiFi.h>
#include <HTTPClient.h>

// Дані WiFi
const char* ssid = "WIFI_SSID";
const char* password = "WIFI_PASSWORD";

// Telegram Token + Chat
String botToken = "BOT_TOKEN";
String chatID = "CHAT_ID"; // або просто використовується /log

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);

    // Очікуємо підключення до WiFi
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Підключення до WiFi...");
    }
    Serial.println("WiFi з'єднано!");
}

void loop() {
    float temperature = random(20, 35); // Для прикладу (реально: з сенсора)

    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
```

```
String url = "https://api.telegram.org/bot" + botToken + "/sendMessage?chat_id=" + chatID +  
"&text=/log%20" + String(temperature, 2);  
http.begin(url);  
int httpResponseCode = http.GET();  
  
if (httpResponseCode > 0) {  
    String response = http.getString();  
    Serial.println("✓ Відправлено: " + response);  
} else {  
    Serial.println("✗ Помилка HTTP");  
}  
  
http.end();  
}  
  
delay(10000); // Надсилати кожні 10 секунд  
}
```

ДОДАТОК Б

Основні MQTT-запити та топіки

iot/plant/temperature
iot/plant/pressure
iot/plant/vibration
iot/plant/state
iot/plant/alert
iot/plant/command
iot/plant/log

```
{  
  "value": 71.2,  
  "unit": "°C",  
  "timestamp": "2025-06-14T15:45:21"  
}
```

```
{  
  "value": 5.6,  
  "unit": "Bar",  
  "timestamp": "2025-06-14T15:45:24"  
}
```

```
{  
  "rms": 0.03,  
  "peak": 0.15,  
  "unit": "g",  
  "timestamp": "2025-06-14T15:45:27"  
}
```

```
{  
  "machine": "welder",  
  "status": "active",  
  "error": false,  
  "timestamp": "2025-06-14T15:45:30"  
}
```

```
{  
  "type": "overheat",  
  "value": 85.7,  
  "threshold": 75.0,  
  "timestamp": "2025-06-14T15:45:35"  
}
```

```
{  
  "action": "stop_machine",  
  "target": "welder",  
  "initiator": "TelegramBot",  
}
```

Кафедра Автоматизації та комп'ютерно-інтегрованих технологій
Система автоматизації виробництва через впровадження IoT-технологій
з використанням соціальних мереж

```
"timestamp": "2025-06-14T15:46:00"  
}
```

```
{  
  "sensor": "temperature",  
  "value": 68.3,  
  "unit": "°C",  
  "timestamp": "2025-06-14T15:46:15"  
}
```

ДОДАТОК В

Скріншоти реалізації Telegram-бота

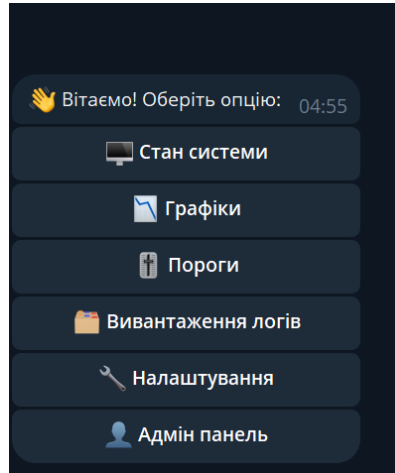


Рисунок В.1 – Головне меню програми

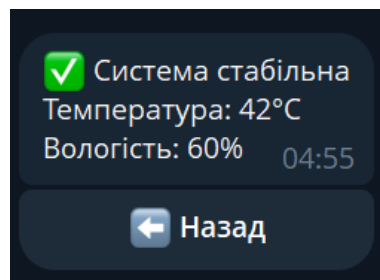


Рисунок В.2 – Приклад стабільності системи

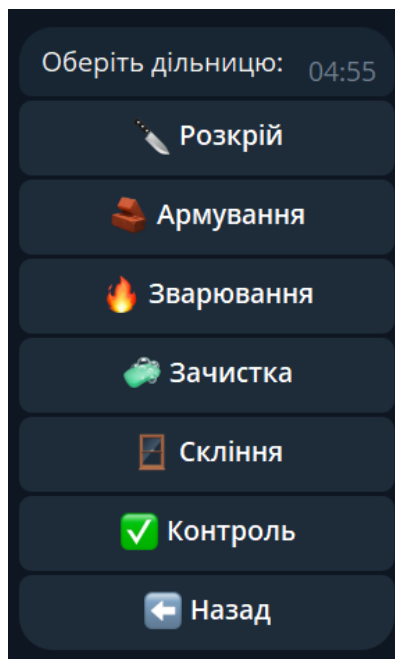


Рисунок В.3 – Меню усіх діляниць підприємства

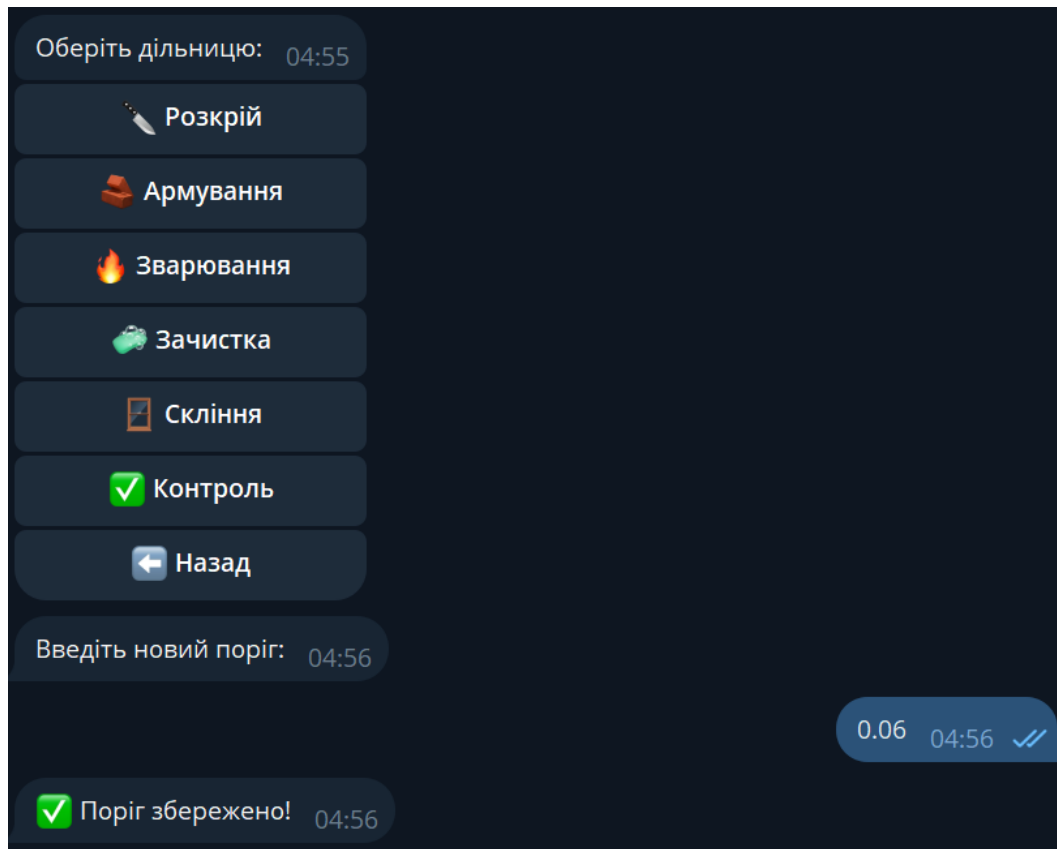


Рисунок В.4 – Приклад виставлення порогів

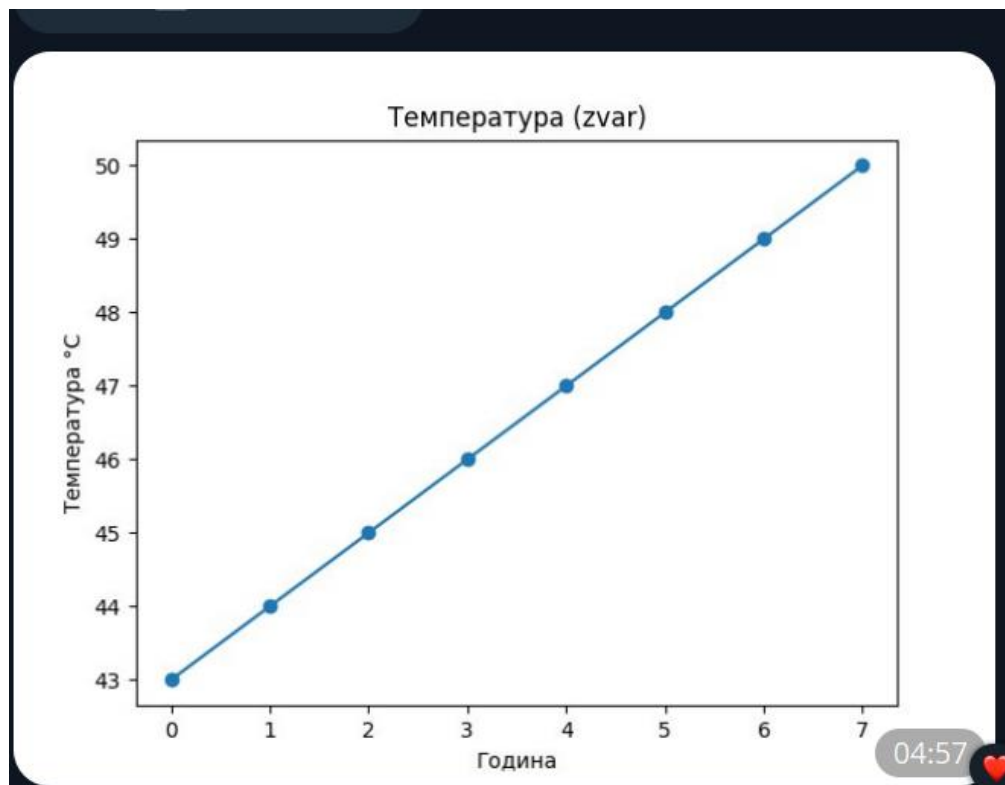


Рисунок В.5 – Приклад графіка за дільницями

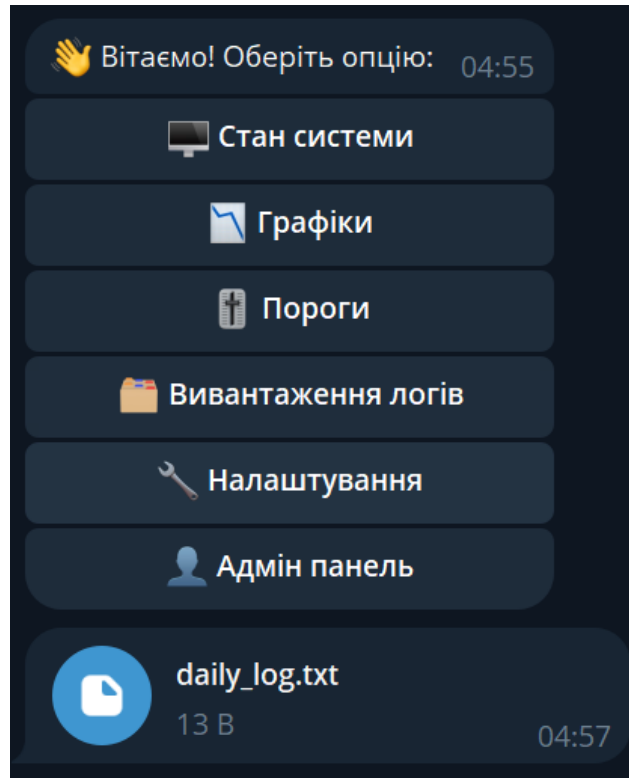


Рисунок В.6 – Приклад генерації звіту

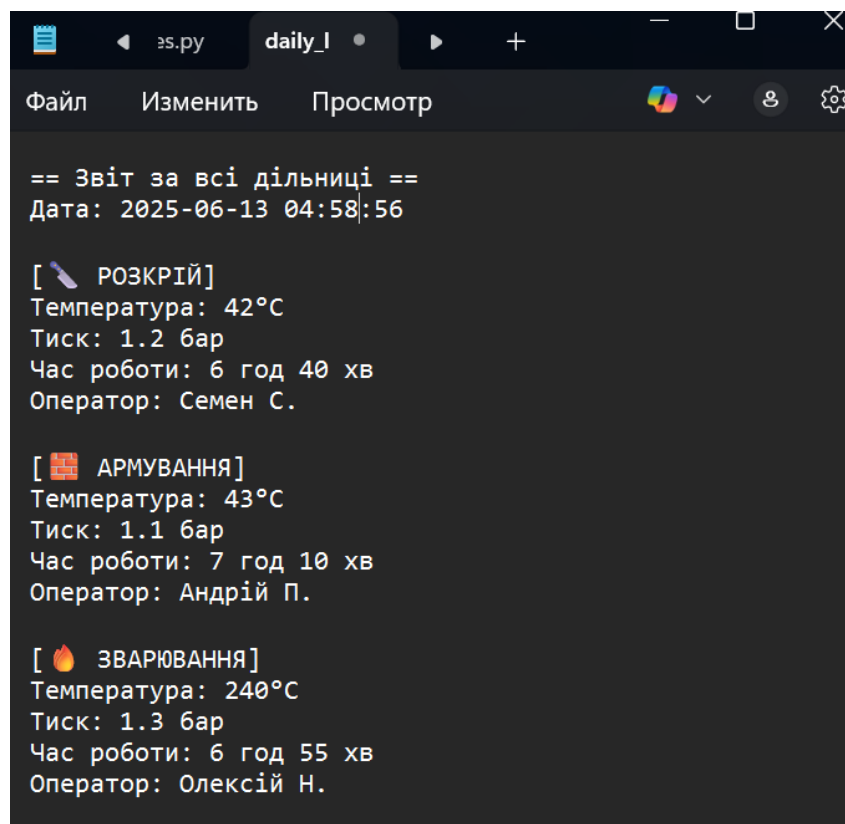


Рисунок В.7 – Звіт

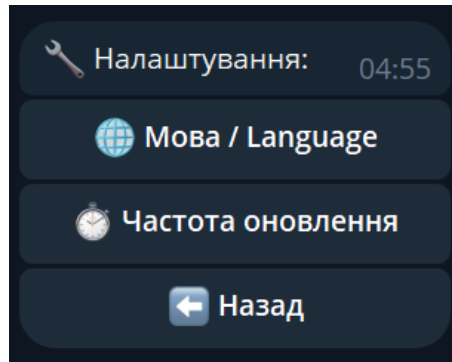


Рисунок В.8 – Меню налаштувань

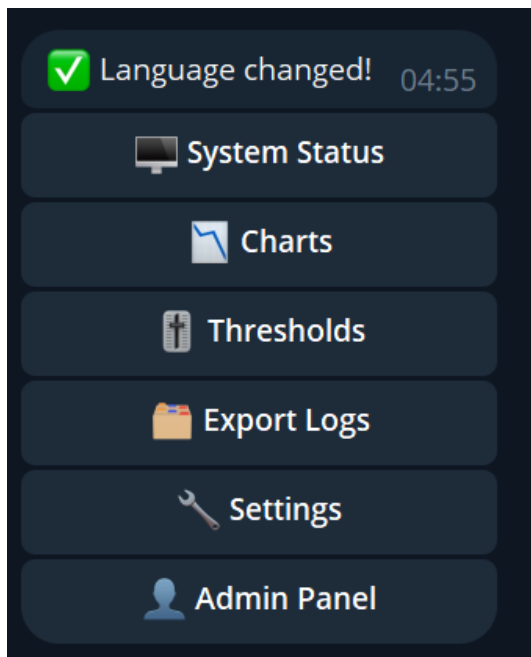


Рисунок В.9 – Зміна мови

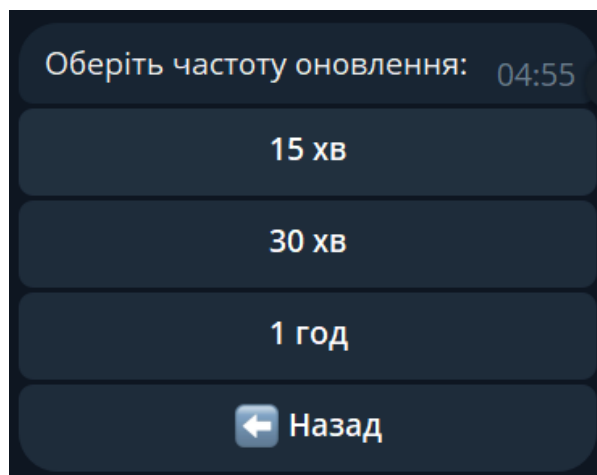


Рисунок В.10 – Приклад частоти оновлення сповіщень

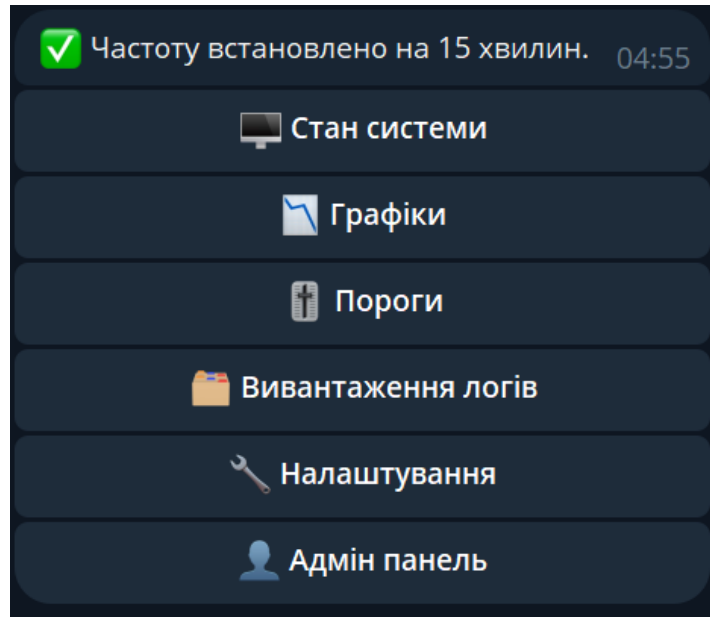


Рисунок В.11 – Приклад встановлення частоти

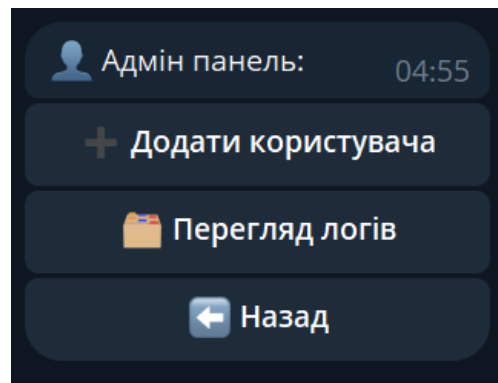


Рисунок В.12 – Адмін панель

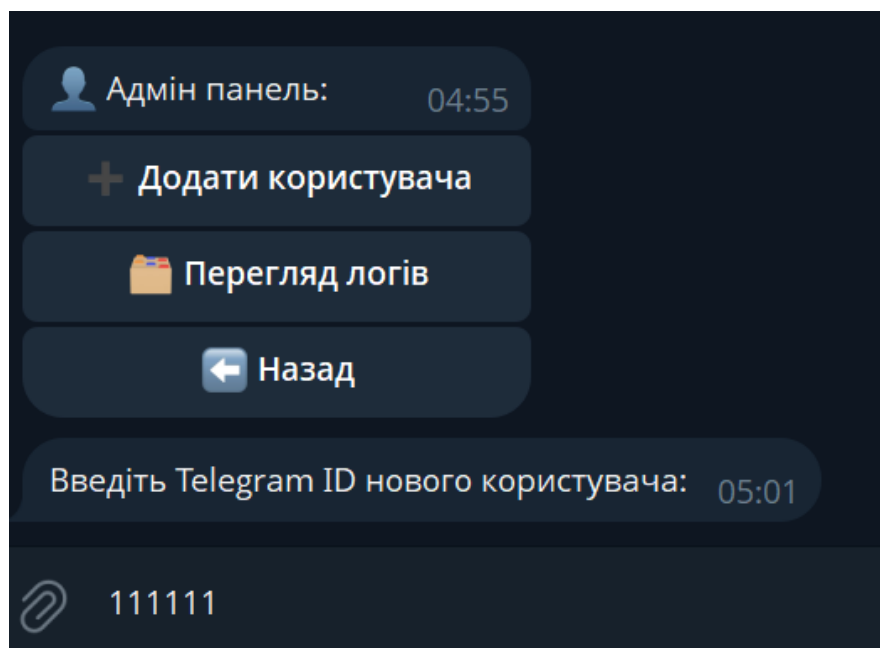


Рисунок В.13 – Приклад додавання користувача в білий список

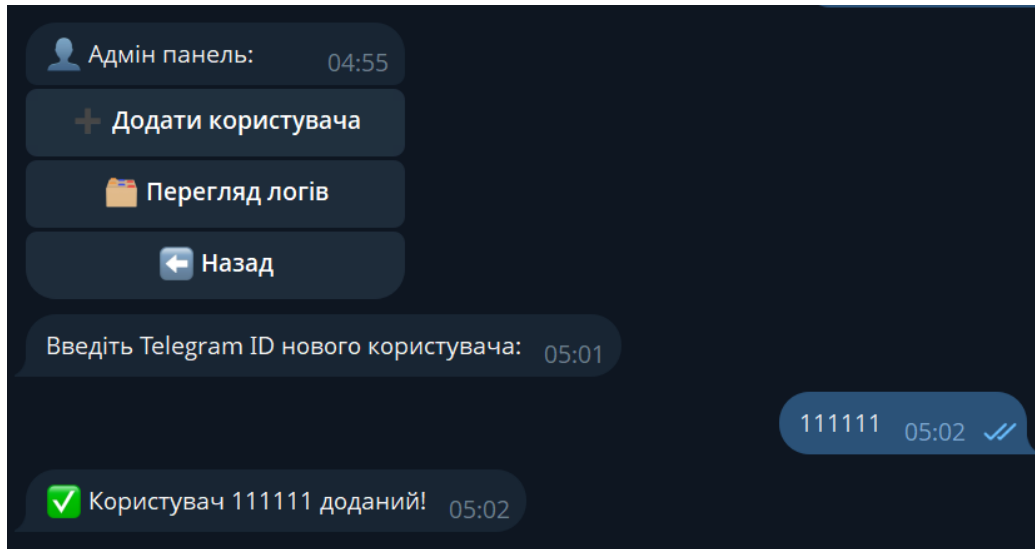


Рисунок В.14 – Додавання користувача

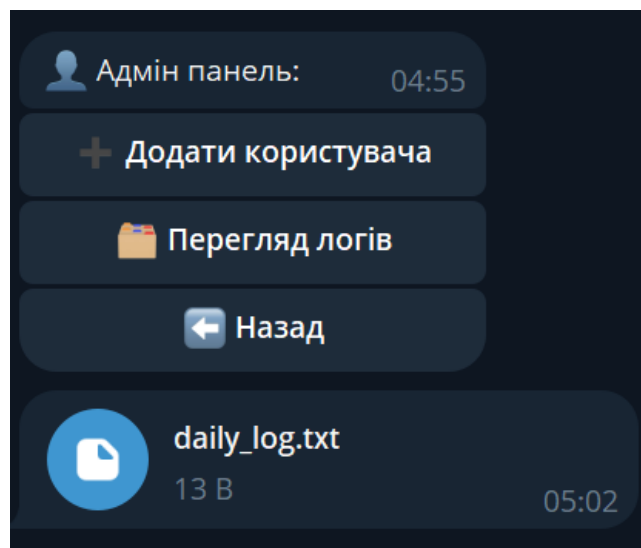


Рисунок В.15 – Приклад перегляду логів за 24 години