

Я, Одноралов Денис Юрійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчестності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

“11” червня 2025 р.

A photograph of a handwritten signature in black ink on a light-colored surface. The signature is stylized and appears to be 'D. Odnoralo'.

Денис ОДНОРАЛОВ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
Рівень вищої освіти перший (бакалаврський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)
Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____

(підпис)

«30» квітня 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Одноралову Денису Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення модуля автоматизованого управління
поліетиленових труб

Затверджена наказом по університету від 19.05.2025 р. № 390 Ст

2. Термін подання здобувачем роботи до екзаменаційної 25.06.2025 р.

3. Вхідні дані до роботи

3.1 Апаратне забезпечення

3.2 База даних SQLite

3.3 Мова програмування Python

3.4 Оформлення документації за ДСТУ 3008-2015

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз предметної області та вимог до модуля, побудова структурно-
функціональної схеми, розробка алгоритму сортування, вибір програмного
середовища для реалізації, програмування логіки сортування, тестування,
питання охорони праці.

5. Перелік графічного матеріалу із закінченням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.pptx) – 8 с.

6. Коснультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
		етапів роботи	
1	Аналіз технічного завдання	30.04.2025	Виконано
2	Аналіз літератури за темою	07.05.2025	Виконано
3	Вибір та обґрунтування програмних засобів	14.05.2025	Виконано
4	Перевірка роботи модуля в умовах симуляції	17.05.2025	Виконано
5	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	01.06.2025	Виконано
6	Оформлення пояснювальної записки	11.06.2025	Виконано
7	Подання роботи на рецензію	12.06.2025	Виконано
8	Подання роботи на підпис зав.	21.06.2025	Виконано
9	Подання кваліфікаційної роботи в ЕК	25.06.2025	Виконано

Дата видачі завдання 30.04.2025

Здобувач _____
(підпис)

Денис ОДНОРАЛОВ _____
(посада, власне ім'я, прізвище)

Керівник роботи _____
(підпис)

Олена ЧАЛА _____
(власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка містить: 80 с., 1 табл., 17 рис., 2 дод., 23 джерела.

АВТОМАТИЗАЦІЯ, МОДУЛЬ, БАЗА ДАНИХ, ЛОГІСТИЧНЕ СОРТУВАННЯ, СОРТУВАЛЬНО-ЛОГІСТИЧНІ СИСТЕМИ.

Об'єкт розробки – автоматизоване управління сортуванням в межах виробництва та зберігання на складах.

Предмет розробки – модуль автоматизованого управління сортуванням поліетиленових труб.

Мета роботи – підвищення якості автоматизованого процесу сортування в технологічних операціях виробництва та зберігання поліетиленових труб шляхом розробки програмного модуля на основі баз даних.

Методи розробки – моделювання процесу автоматизованого сортування у внутрішньоскладських логістичних системах; метод впровадження роботи з базою даних підприємства.

Було проаналізовано предметну область, досліджено особливості сортувально-логістичних процесів на виробництві поліетиленових труб і сучасні автоматизовані системи сортування. Обрано та розроблено базу даних і програмну логіку алгоритма сортування. Проведено тестування в симуляційному середовищі для оцінки точності сортування, обробки браку та управління складськими зонами.

Упровадження розробленого модуля сприятиме успішній реалізації роботи систем автоматизованого сортування поліетиленових труб у складі внутрішньоскладських логістичних системах, а при проведенні лабораторних та практичних занять буде корисним для здобувачів першого (бакалаврського) рівня вищої освіти за освітньо-професійною програмою: «Автоматизація та комп'ютерно-інтегровані технології» (АКТАКІТ).

ABSTRACT

Explanatory note contains: 80 pp., 1 table, 17 figures, 2 appendices, 23 sources.

AUTOMATION, MODULE, DATABASE, LOGISTIC SORTING, SORTING-LOGISTIC SYSTEMS.

Object of development – automated management of sorting within production and warehouse storage.

Subject of development – a module for automated management of polyethylene pipe sorting.

Objective of the work – enhancing the quality of the automated sorting process in technological operations of polyethylene pipe production and storage by developing a software module based on a database.

Development methods – modeling the automated sorting process in internal warehouse logistics systems; implementing enterprise database operations.

The subject area was analyzed, and the specifics of sorting-logistics processes in polyethylene pipe production and modern automated sorting systems were investigated. A database and the software logic of the sorting algorithm were selected and developed. Testing was conducted in a simulation environment to evaluate the accuracy of sorting, defect handling, and warehouse zone management.

The implementation of the developed module will facilitate the successful operation of automated polyethylene pipe sorting systems within internal warehouse logistics systems. It will also be beneficial for students at the first (bachelor's) level of higher education in the educational-professional program "Automation and Computer-Integrated Technologies" (AKTAKIT) during laboratory and practical classes.

ЗМІСТ

Перелік скорочень.....	6
Вступ.....	7
1. Аналіз предметної області та вимог до модуля сортування.....	9
1.1 Опис сортувально-логістичних процесів на виробництві.....	9
1.2 Завдання на роботу.....	11
1.3 Вимоги до модуля.....	12
1.4 Огляд існуючих рішень.....	14
2. Розробка архітектури та алгоритму роботи модуля.....	19
2.1 Структура та принципи роботи модуля.....	19
2.2 Алгоритми сортування.....	22
3. Реалізація, тестування та симуляція роботи модуля.....	25
3.1 Проектування та розробка бази даних.....	25
3.2 Створення вебресурсу для тестування.....	33
3.3 Програмна реалізація модуля з допомогою Python.....	38
3.4 Тестування роботи модуля.....	43
3.5 Тестування системи на стійкість.....	48
3.6 Охорона праці.....	52
Висновки.....	55
Перелік джерел посилання.....	57
Додаток А Лістинг коду.....	61
Додаток Б Демонстраційний матеріал.....	72

ПЕРЕЛІК СКОРОЧЕНЬ

АКТАКІТ – автоматизація та комп'ютерно-інтегровані технології;

КІТАР – комп'ютерно-інтегрованих технологій, автоматизації та робототехніки;

ПЗ – програмне забезпечення;

ТАУ – теорія автоматичного управління

ХНУРЕ – Харківський національний університет радіоелектроніки;

AES-256 (Advanced Encryption Standard with 256-bit key) – стандарт шифрування з 256-бітним ключем;

ERP (Enterprise Resource Planning) – планування ресурсів підприємства;

Ethernet/IP (Ethernet Industrial Protocol) – промисловий протокол Ethernet;

IP54 (Ingress Protection 54) – захист від проникнення, клас 54;

MES (Manufacturing Execution System) – система управління виробництвом;

OPC UA (Open Platform Communications Unified Architecture) – уніфікована архітектура відкритих платформних комунікацій;

PLC(ПЛК) – програмований логічний контролер;

SCADA (Supervisory Control and Data Acquisition) – система диспетчерського управління та збору даних;

ВСТУП

У сфері виробництва поліетиленових труб, проблема ефективного сортування стає ще більш актуальною через різноманіття типів та розмірів продукції та високих обсягів виготовлення. Ручне або частково автоматизоване сортування не забезпечує достатньої продуктивності, а також ускладнює подальше транспортування та зберігання.

Не зважаючи на велику кількість досліджень та розробок за цією тематикою, виникає проблема в необхідності вдосконалення процесу сортування, яке є важливим у багатьох галузях, де обмежується недостатня гнучкість сучасних систем і застаріле обладнання, яке потребує модернізації.

Також частково стає питання упорядкування виробничими й складськими процесами. В будь якій виробничій системі потрібна інтеграція автоматизації сортування, та внутрішньоскладського переміщення продукції. Завдяки автоматизації цих процесів, підвищується швидкість виконання операцій, зменшується кількість помилок, знижується потреба в ручній праці, що дозволяє економити ресурси та забезпечити стабільну якість продукції.

Спираючись на вище наведене, можна вважати, що обрана тема є нагальною, своєчасною та актуальною.

Об'єкт розробки – автоматизоване управління сортуванням в межах виробництва та зберігання на складах.

Предмет розробки – модуль автоматизованого управління сортуванням поліетиленових труб.

Мета роботи – підвищення якості автоматизованого процесу сортування, в технологічних операціях виробництва та зберігання поліетиленових труб шляхом розробки програмного модуля на основі бази даних.

Упровадження розробленого модуля сприятиме успішній реалізації систем автоматизованого керування сортуванням поліетиленових труб у складі внутрішньоскладських логістичних систем, а при проведенні лабораторних та

практичних занять буде корисним для здобувачів першого (бакалаврського) рівня вищої освіти за освітньо-професійною програмою: «Автоматизація та комп'ютерно-інтегровані технології» (АКТАКІТ).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО МОДУЛЯ СОРТУВАННЯ

1.1 Опис сортувально-логістичних процесів на виробництві

Сортувально-логістичні процеси на підприємстві з виготовлення поліетиленових труб становлять ключовий елемент ефективної роботи автоматизованого модуля, який забезпечує повний цикл транспортування, класифікації та розподілу труб від моменту їх виробництва на екструзійній лінії до розміщення у складських зонах або підготовки до відвантаження [1-2]. Ці процеси є основою для забезпечення безперебійного функціонування виробництва, мінімізуючи ручні операції та підвищуючи точність і швидкість обробки продукції.

Після завершення процесу формування труб за допомогою технології екструзії кожна одиниця проходить ретельну перевірку якості, що є критично важливим етапом для забезпечення відповідності продукції встановленим стандартам. Для цього використовуються два високотехнологічні інструменти: система комп'ютерного зору з точністю виявлення поверхневих дефектів на рівні 99% та ультразвуковий дефектоскоп, який забезпечує високу точність вимірювання товщини стінок труб.

Система комп'ютерного зору аналізує зовнішній вигляд труб, виявляючи такі недоліки, як тріщини, подряпини, деформації чи забруднення поверхні [3].

У той же час ультразвуковий контроль спрямований на виявлення внутрішніх аномалій, таких як порожнини, відхилення у товщині стінок або неоднорідність матеріалу, що дозволяє гарантувати високу якість готової продукції.

На кожную трубу наноситься унікальний штрих-код формату Code 128 розміром 2×5 см, який містить ключову інформацію про виріб. Цей штрих-код зчитується спеціалізованим сканером на різних етапах логістичного процесу.

На основі даних, отриманих від системи комп'ютерного зору та ультразвукового дефектоскопа, автоматизований модуль оновлює статус

бракованості кожної труби в базі даних, що забезпечує чітке відстеження якості продукції на всіх етапах.

База даних відіграє центральну роль у процесі управління, зберігаючи детальну інформацію про кожну трубу, включаючи її унікальний ідентифікатор, ідентифікатор замовлення, тип труби, зону призначення, час зміни статусу та поточний статус.

Автоматизований модуль попередньо визначає оптимальну зону призначення для кожної труби, враховуючи пріоритети замовлень, технічні характеристики труб і поточний стан складських зон. Зчитування штрих-коду на різних етапах підтверджує правильність маршруту та оновлює статус бракованості після проходження перевірок якості, що забезпечує точність і надійність логістичного процесу.

Після проходження перевірок труби спрямовуються до відповідних зон за допомогою конвеєрної системи, яка оснащена розподільними вузлами. Ці вузли керуються програмованим логічним контролером (PLC), який використовує інформацію з бази даних для точного направлення труб до визначених зон. Конвеєрна система працює з високою точністю, забезпечуючи безперебійне транспортування та мінімізуючи ризик помилок у маршрутизації [4].

На етапах переміщення труб по конвеєрним лініям та перед їх остаточним розміщенням у кінцеву зону штрих-код зчитується повторно. Це дозволяє підтвердити правильність маршруту, оновити дані в базі та забезпечити повну відповідність логістичних операцій заданій логіці. Такий підхід гарантує, що кожна труба потрапляє саме туди, куди було заплановано, без відхилень.

Система управління виробництвом (MES) відіграє важливу роль у забезпеченні зворотного зв'язку. Вона дозволяє коригувати чергу виробництва, наприклад, додаючи нові труби для заміни бракованих або зупиняючи виробничі процеси у разі переповнення складських зон. Усі значущі події, такі як запуск виробництва, виявлення дефектів, завершення замовлення чи відвантаження, фіксуються в таблиці подій бази даних. Це забезпечує повну прозорість процесів, можливість їх подальшого аналізу та оптимізації.

Після завершення всіх етапів обробки труби розміщуються у відповідних складських зонах або спрямовуються до зони відвантаження для подальшої доставки клієнтам. Цей процес завершує цикл сортувально-логістичних операцій, забезпечуючи високу ефективність і точність виконання замовлень.

1.2 Завдання на роботу

Розробка модуля автоматизованого керування сортуванням поліетиленових труб спрямована на створення високоефективної системи, яка забезпечує повну автоматизацію логістичних і сортувальних операцій на виробництві, знижуючи залежність від ручної праці та підвищуючи продуктивність. Основним завданням є проектування модуля, який інтегрує апаратні та програмні компоненти для точного транспортування, класифікації та розподілу труб від виробничої лінії до складських зон або зони відвантаження, з урахуванням пріоритетів замовлень і автоматичної обробки бракованих одиниць.

Модуль має забезпечувати безперебійну взаємодію з системою управління виробництвом (MES) через сучасний протокол OPC UA, який гарантує швидку передачу даних у реальному часі із затримкою до 100 мс і надійним шифруванням AES-256 [5]. Це дозволяє синхронізувати логістичні процеси з виробничим циклом, оперативно реагувати на виявлення браку та коригувати плани виробництва. Крім того, модуль повинен підтримувати інтеграцію з базою даних, яка зберігає детальну інформацію про кожну трубу, включаючи її ідентифікатор, тип, статус, зону призначення та пріоритет замовлення, забезпечуючи прозорість і можливість аналізу даних.

Серед ключових завдань також є розробка гнучкої архітектури модуля, яка дозволяє легко масштабувати систему, додаючи нові конвеєрні лінії, сенсори чи типи труб без значних змін у апаратному чи програмному забезпеченні.

Модуль має бути стійким до промислових умов, захищеним від пилу, вологи та вібрацій, а також оснащеним механізмом аварійної зупинки для забезпечення безпеки. Важливим аспектом є автоматизація обробки бракованих труб, що

включає їх автоматичне перенаправлення до зони «Defect» і формування запитів до MES для виготовлення заміни, що мінімізує простої та підвищує ефективність.

Ще одним завданням є створення алгоритмів сортування, які базуються на логіці пріоритизації замовлень за дедлайнами та враховують технічні характеристики труб.

Ці алгоритми повинні забезпечувати точне визначення маршрутів, своєчасне оновлення статусів у базі даних і контроль місткості складських зон для запобігання переповненню.

Таким чином, завдання на розробку модуля охоплюють створення комплексної системи, яка поєднує апаратні компоненти, програмне забезпечення, базу даних і алгоритми для забезпечення високої продуктивності, гнучкості та надійності в автоматизації сортувальних і логістичних процесів на виробництві.

1.3 Вимоги до модуля

Модуль автоматизації сортування розроблено з метою значного підвищення продуктивності логістичних процесів на підприємстві шляхом повної автоматизації класифікації труб і максимального скорочення ручних операцій [6].

Цей модуль є ключовим інструментом для оптимізації виробничих і логістичних операцій, що дозволяє підприємству працювати більш ефективно та економічно.

Модуль обробляє інформацію з бази даних, використовуючи дані про замовлення і технічні характеристики для визначення їх пріоритетності, та встановлення оптимальних маршрутів, статусів і черговості виробництва.

Формування маршрутів здійснюється з урахуванням пріоритетів замовлень і параметрів труб, з подальшим коригуванням на основі результатів зчитування штрих-кодів і перевірок якості. Такий підхід забезпечує гнучкість і точність у розподілі труб.

У разі виявлення бракованих труб модуль автоматично спрямовує їх до спеціально відведеної зони «Defect». Одночасно формується запит до системи

управління виробництвом (MES) на виготовлення заміни для бракованих одиниць, що дозволяє оперативно реагувати на проблеми якості та підтримувати безперебійність виконання замовлень.

Контроль місткості складських зон здійснюється через базу даних, яка відстежує доступний простір у реальному часі. У разі переповнення основних та тимчасових складських зон модуль автоматично зупиняє виробництво, щоб уникнути накопичення надлишкової продукції, та забезпечити оптимальне використання складських ресурсів.

Усі ключові події, такі як початок чи завершення замовлення, виявлення браку або відвантаження, ретельно фіксуються в базі даних для подальшого аналізу та вдосконалення процесів.

Технічні вимоги до модуля включають інтеграцію з системою управління виробництвом (MES) через протокол OPC UA, який забезпечує передачу даних у реальному часі із затримкою не більше 100 мс. Це дозволяє синхронізувати всі процеси та забезпечити їх безперебійну роботу.

Сканери штрих-кодів зчитують інформацію на вході, під час переміщення труб по конвеєрних лініях і перед їх розміщенням у кінцеву зону, що гарантує точність маршрутизації.

Архітектура модуля спроектована з урахуванням гнучкості, що дозволяє легко додавати нові конвеєрні лінії, сенсори чи складські зони без необхідності значних змін у програмному чи апаратному забезпеченні.

Модуль забезпечує стабільну роботу в промислових умовах, будучи захищеним від впливу пилу, вібрацій і оснащеним надійним механізмом аварійної зупинки для забезпечення безпеки [7-8].

Він функціонує автономно, але підтримує можливість моніторингу через спеціалізоване програмне забезпечення, що полегшує контроль і аналіз роботи системи.

Для забезпечення масштабованості модуль підтримує додавання нових типів труб, таких як каналізаційні чи промислові, а також збільшення пропускної

здатності системи. Це дозволяє адаптувати модуль до зростаючих потреб виробництва та забезпечувати його актуальність у довгостроковій перспективі.

1.4 Огляд існуючих рішень

Сучасний ринок автоматизованих систем для сортування та логістичної обробки труб пропонує низку рішень, які мають свої сильні сторони, але також і певні обмеження, що робить їх менш універсальними порівняно з розробленим модулем.

У цьому розділі розглядаються два ключові приклади таких систем: рішення від компанії JR Automation для Advanced Drainage Systems і запатентована система від Xuzhou DKEC Electrical Technology Co Ltd. Обидва рішення аналізуються з точки зору їх функціональності, технічних характеристик, рівня автоматизації та гнучкості, з подальшим порівнянням із розробленим модулем.

Компанія JR Automation розробила високоефективну автоматизовану систему для сортування, зв'язування та палетування поліетиленових труб, яка була впроваджена для компанії Advanced Drainage Systems [9].

Ця система здатна працювати в цілодобовому режимі, обробляючи труби стандартної довжини 10 футів, що забезпечує високу продуктивність і надійність у промислових умовах.

Процес починається з транспортування труб за допомогою конвеєрної системи до спеціалізованої сортувальної станції. На цій станції труби групуються за діаметром у стоси, які можуть складатися з 2, 3, 5 або 7 одиниць, залежно від технічних вимог.

Після формування стоси автоматично зв'язуються за допомогою спеціального обладнання, а потім розміщуються на палетах за допомогою промислового маніпулятора FANUC, що забезпечує точність і швидкість виконання операцій.

Для ідентифікації труб використовуються високошвидкісні сканери Zebra DS3608, які зчитують штрих-коди за одну секунду, що значно прискорює процес обробки.

Керування системою здійснюється через програмований логічний контролер (PLC) Rockwell CompactLogix, який підключений до мережі через протокол Ethernet/IP, забезпечуючи стабільне та централізоване управління.

Незважаючи на високу продуктивність, система JR Automation має певні обмеження, які впливають на її універсальність. Основна орієнтація на палетування труб обмежує гнучкість сортування за іншими параметрами, такими як типи труб чи пріоритети замовлень, що є ключовою особливістю розробленого модуля.

Крім того, відсутність інтеграції з системою управління виробництвом (MES) унеможливорює автоматичне коригування виробничих процесів, наприклад, у разі виявлення браку чи зміни пріоритетів замовлень. Інформація про обробку бракованих труб у системі JR Automation відсутня, що дозволяє припустити, що цей процес, ймовірно, виконується вручну, що знижує рівень автоматизації. На рисунку 1.1 зображено роботу маніпулятора FANUC, який є ключовим елементом системи.



Рисунок 1.1 – Робота маніпулятора FANUC у системі JR Automation

Інше рішення, описане в патенті CN104229223A від компанії Xuzhou DKES Electrical Technology Co Ltd, представляє собою систему для сортування та підрахунку пластикових труб [10], яка спрямована на автоматизацію логістичних процесів у виробництві.

Ця система включає кілька ключових компонентів: вхідний і перехідний контейнери для приймання труб, піднімальний механізм для їх транспортування та пристрій сортування, який керується промисловим комп'ютером.

Сортування труб здійснюється за кількістю одиниць у партії, що визначається за допомогою оптичних сенсорів, які забезпечують точний підрахунок.

Надлишкові труби відокремлюються за допомогою пневматичного механізму, що дозволяє підтримувати порядок у процесі обробки.

Дані про кожну партію труб записуються до бази даних, яка інтегрована з заводською цифровою шиною, що забезпечує централізоване управління логістичними операціями.

Проте система має низку обмежень, які впливають на її ефективність і гнучкість. Маршрути транспортування труб є фіксованими, що значно знижує можливість адаптації системи до різних типів труб чи змін у виробничих потребах.

Обробка бракованих труб частково автоматизована: оптичні сенсори можуть сигналізувати про недостатню кількість труб у партії, але автоматичне перенаправлення браку до окремої зони не передбачено, що вимагає ручного втручання.

Інтеграція через цифрову шину, ймовірно, не підтримує шифрування даних і має затримки передачі, які можуть перевищувати 100 мс, що знижує її ефективність у реальному часі.

Крім того, система не підтримує пріоритизацію замовлень, що обмежує її здатність адаптуватися до динамічних виробничих умов.

Масштабування системи ускладнене через фіксовану конструкцію бункерів і контейнерів, що вимагає фізичних модифікацій обладнання для обробки нових типів труб, таких як труби з іншими діаметрами чи матеріалами.

Для наочного порівняння характеристик розглянутих систем і розробленого модуля наведено таблицю 1.1, яка висвітлює ключові критерії оцінки [11].

Порівняно з розглянутими рішеннями, розроблений модуль демонструє значні переваги, які роблять його більш універсальним і ефективним для сучасного виробництва.

Повна автоматизація обробки браку – на відміну від систем JR Automation (де обробка браку, ймовірно, виконується вручну) та Xuzhou DKEC (де автоматизація обмежена оповіщенням), розроблений модуль автоматично перенаправляє браковані труби до зони «Defect» і формує запит до MES для виробництва їх заміни, що значно знижує потребу в ручній праці.

Інтеграція з MES через OPC UA – розроблений модуль використовує сучасний протокол OPC UA із затримкою до 100 мс і шифруванням AES-256, що забезпечує швидку, безпечну та надійну передачу даних. Це контрастує з відсутністю інтеграції в системі JR Automation і потенційно ненадійною цифровою шиною в системі Xuzhou DKEC.

Гнучкість і пріоритизація замовлень – розроблений модуль дозволяє динамічно сортувати труби за пріоритетами замовлень, використовуючи дані з бази, тоді як обидві альтернативні системи не підтримують такої функціональності. Це робить модуль більш адаптивним до змін у виробничих планах.

Масштабованість – архітектура розробленого модуля спроектована з урахуванням можливості легкого додавання нових типів труб, конвеєрних ліній чи сенсорів без значних змін у програмному чи апаратному забезпеченні. На противагу цьому, система JR Automation обмежена палетуванням, а Xuzhou DKEC вимагає фізичних модифікацій для масштабування.

Таким чином, розроблений модуль перевершує існуючі рішення за рахунок вищого рівня автоматизації, гнучкості маршрутизації, інтеграції з MES, динамічної пріоритизації замовлень і масштабованої архітектури. Ці характеристики забезпечують його здатність відповідати сучасним вимогам промислового виробництва та адаптуватися до майбутніх змін у технологічних процесах.

Таблиця 1.1 – Порівняння систем сортування

Критерій	JR Automation	Xuzhou DKEC Electrical Technology Co Ltd	Розроблений модуль
Сортування	За діаметром, стоси (2, 3, 5, 7 труб)	За кількістю	За замовленням
Інтеграція з MES	Відсутня	Через цифрову шину	OPC UA, затримка до 100 мс, шифрування AES- 256
Обробка браку	Ймовірно ручна	Частково автоматизована (оповіщення)	Автоматична
Пріоритизація замовлень	Не підтримується	Не підтримується	Через базу даних
Масштабність	Обмежена (палетування)	Обмежена (ліміт лічильної зони)	Гнучка

2 РОЗРОБКА АРХІТЕКТУРИ, АЛГОРИТМІВ РОБОТИ МОДУЛЯ

На основі детального аналізу предметної області, включаючи особливості виробництва поліетиленових труб і вимоги до автоматизації логістичних процесів, було прийнято рішення про розробку власного модуля автоматизованого керування сортуванням. Цей розділ присвячений обґрунтуванню характеристик модуля, його структури, алгоритмів роботи та програмного забезпечення, які забезпечують ефективне виконання поставлених завдань. У наступних підрозділах розглядаються завдання на розробку модуля, його апаратно-програмна структура, принципи роботи, а також алгоритми, що лежать в основі автоматизації сортувальних процесів.

2.1 Структура та принципи роботи модуля

Модуль автоматизованого керування сортуванням поліетиленових труб являє собою інтегровану систему, яка поєднує апаратні та програмні компоненти для реалізації повного циклу сортування, обробки бракованих труб і взаємодії з іншими виробничими системами.

Його структура розроблена з урахуванням вимог до високої продуктивності, надійності та гнучкості, що дозволяє адаптувати систему до змін у виробничих умовах і масштабувати її для обробки нових типів труб.

Апаратна частина модуля

Основним елементом апаратного забезпечення є програмований логічний контролер (PLC) Siemens S7-1200 (модель CPU 1214C DC/DC/DC, розміри 110 мм×100 мм×75 мм), який виконує центральну роль в обробці вхідних даних, керуванні виконавчими механізмами та передачі інформації [12].

PLC обробляє сигнали від сенсорів і сканерів, видає команди конвеєрам і розподільним вузлам, а також забезпечує зв'язок із зовнішніми системами через протокол OPC UA. Цей протокол використовує Ethernet-з'єднання із затримкою

передачі даних до 100 мс і шифруванням AES-256, що гарантує безпечну та швидку комунікацію в реальному часі.

Сканери штрих-кодів відповідають за ідентифікацію труб, зчитуючи штрих-коди формату Code 128, нанесені на кожну одиницю. Отримані дані передаються до бази для визначення параметрів труби, таких як тип, розмір і зона призначення.

Конвеєрна система, оснащена розподільними вузлами, забезпечує транспортування труб до відповідних зон, керованих PLC на основі інформації з бази даних.

Система комп'ютерного зору з точністю виявлення поверхневих дефектів 99% і ультразвуковий дефектоскоп для контролю товщини стінок і внутрішніх аномалій ідентифікують браковані труби, оновлюючи їхній статус у базі даних для подальшої обробки.

Для забезпечення стабільного живлення всіх компонентів використовується блок живлення SITOP PSU100S (24 В, 5–10 А, розміри ~50 мм×125 мм×120 мм), який гарантує безперебійну роботу навіть у складних промислових умовах.

Модуль зв'язку SCALANCE XB005 (5 портів, 10/100 Мбіт/с, розміри ~40 мм×100 мм×80 мм) забезпечує надійне мережеве з'єднання, а модуль SM 1223 (8 входів/виходів, розміри ~45 мм×100 мм×75 мм) обробляє сигнали від сенсорів і виконавчих пристроїв.

Усі компоненти з'єднані екранованими кабелями (Ethernet Cat5e, M12 для мережевих з'єднань і сигнальні кабелі 24 В), що мінімізує електромагнітні перешкоди.

Фізична структура модуля реалізована в компактній монтажній шафі Rittal AE (розміри 600 мм×400 мм×200 мм, клас захисту IP54), яка захищає обладнання від пилу, вологи та вібрацій, забезпечуючи довговічність і надійність у промислових умовах.

Шафа спроектована з урахуванням можливості масштабування, дозволяючи додавати нові компоненти, такі як додаткові сенсори чи модулі зв'язку, без значних модифікацій.

Програмне забезпечення модуля відповідає за обробку даних, керування логістичними процесами та інтеграцію з іншими системами. Воно взаємодіє з базою даних, яка зберігає детальну інформацію про кожну трубу (ідентифікатор, тип, статус, зона призначення) та замовлення (пріоритет, дедлайн, обсяг).

База даних підтримує механізми пріоритизації, що дозволяють модулю оптимально розподіляти труби відповідно до терміновості замовлень.

Інтеграція з MES через OPC UA забезпечує синхронізацію даних у реальному часі, включаючи повідомлення про брак, статуси замовлень і коригування виробничого плану.

Програмна частина також включає інтерфейс для моніторингу, який може бути реалізований на зовнішньому комп'ютері через спеціалізоване програмне забезпечення [13].

Програмне забезпечення розроблено з урахуванням гнучкості, що забезпечує можливість додавання нових функцій, таких як підтримка нових типів труб чи розширення алгоритмів сортування.

Структурно-функціональна схема модуля, зображена на рисунку 2.1, ілюструє операції і взаємозв'язки між компонентами.

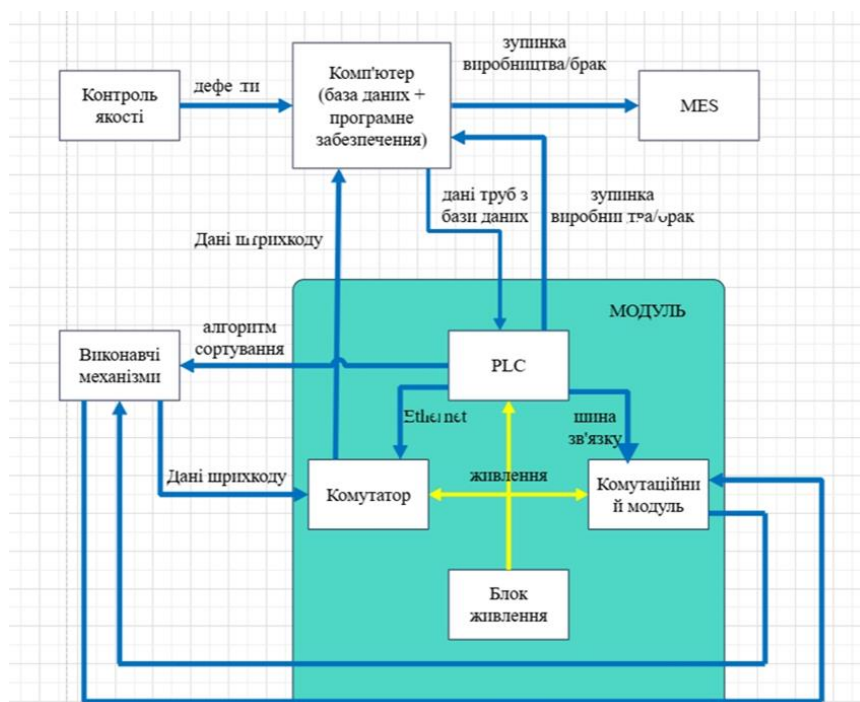


Рисунок 2.1 – Структурно-функціональна схема

Процес починається з контролю браку, дані з якого обробляються модулем та передаються до PLC.

Далі модуль обробляє вхідні дані, перевіряє інформацію про труби, визначає маршрути на основі пріоритетів замовлень і фіксує статуси в базі даних.

У разі виявлення браку чи переповнення складських зон модуль надсилає відповідні сигнали до MES або зупиняє виробничу лінію для коригування плану.

Схема чітко відображає логіку роботи системи, забезпечуючи прозорість і зручність для аналізу.

Таким чином, модуль являє собою комплексну систему, яка поєднує надійне апаратне забезпечення, гнучке програмне забезпечення та інтеграцію з виробничими системами для забезпечення високої ефективності сортувальних і логістичних процесів.

2.2 Алгоритми сортування

Алгоритми сортування труб, реалізовані в програмному забезпеченні модуля, базуються на чіткій і послідовній логіці, яка забезпечує автоматизацію класифікації, маршрутизації та обробки бракованих одиниць [14].

Ці алгоритми розроблені з урахуванням вимог до швидкості, точності та гнучкості, що дозволяє системі ефективно обробляти різноманітні типи труб і адаптуватися до змін у виробничих умовах.

Алгоритм починається зі зчитування штрих-коду виготовленої труби за допомогою сканера, який передає отримані дані до бази даних через PLC Siemens S7-1200. Ці дані включають ідентифікатор труби, тип, розміри та інформацію про замовлення.

Зчитування штрих-коду повторюється на ключових етапах: перед переміщенням труби по конвеєрній системі та перед її розміщенням у кінцевій зоні, що забезпечує точність маршрутизації та оновлення статусів.

На основі даних із бази алгоритм перевіряє характеристики труби, включаючи їх статус за результатами перевірки якості, виконаної системами комп'ютерного зору та ультразвукового дефекту.

Алгоритм визначає зону призначення, враховуючи пріоритет замовлення, визначений дедлайном, а також поточний стан складських зон. Наприклад, якщо замовлення має високу терміновість, труби для нього спрямовуються до зони відвантаження з вищим пріоритетом.

Після визначення маршруту модуль формує і передає команди до конвеєрної системи та розподільних вузлів через PLC, забезпечуючи транспортування труби до відповідної зони.

На кожному етапі статус труби (наприклад, «в обробці», «бракована», «готова до відвантаження») фіксується в базі даних, що забезпечує повну прозорість і можливість відстеження.

У разі виявлення браку алгоритм автоматично перенаправляє трубу до зони «Defect» і модуль надсилає запит до MES через OPC UA для додавання одиниці заміни до виробничого плану.

Алгоритм також включає місткість складських зон, відстежуючи доступний простір через базу даних. У разі переповнення основних та тимчасових зон модуль зупиняє виробництво, надсилаючи відповідний сигнал до MES, що дозволяє оперативно скоригувати графік. Крім того, алгоритм передбачає обробку аварійних ситуацій, таких як збої обладнання чи невідповідність даних, із запуском механізму аварійної зупинки для забезпечення безпеки.

Блок-схема алгоритму сортування, зображена на рисунку 2.2, детально ілюструє всі етапи виконання: ініціалізацію системи, зчитування штрих-коду, перевірку даних, визначення маршруту, транспортування труби, оновлення статусів і обробку виняткових ситуацій, таких як брак чи переповнення зон. Схема чітко відображає логічні зв'язки між етапами, полегшуючи розуміння та аналіз роботи алгоритму.

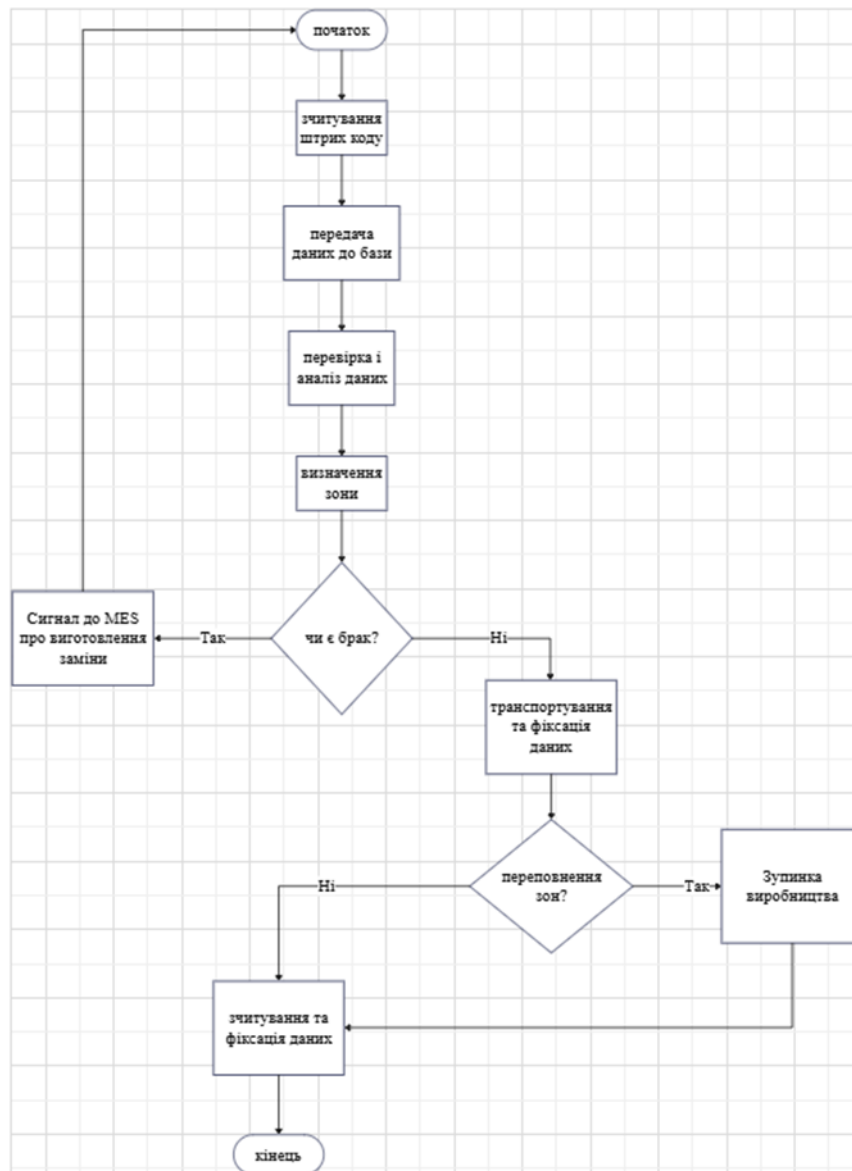


Рисунок 2.2 – Блок-схема алгоритму сортування

Таким чином, алгоритм сортування забезпечує високу ефективність і надійність модуля, дозволяючи автоматизувати складні логістичні процеси, оптимізувати маршрутизацію труб і оперативно реагувати на зміни у виробничому циклі. Їхня гнучка структура підтримує масштабування системи та адаптацію до нових вимог виробництва.

3 РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА СИМУЛЯЦІЯ РОБОТИ МОДУЛЯ

3.1 Проектування та розробка бази даних

Для забезпечення ефективного моделювання та тестування роботи модуля автоматизованого керування сортуванням поліетиленових труб було розроблено реляційну базу даних на основі SQLite.

Вибір SQLite як основи бази даних для симуляції зумовлений її легкістю, портативністю, високою продуктивністю та простотою інтеграції з програмним середовищем Python [15], яке використовується для реалізації логіки модуля.

Ці характеристики роблять SQLite ідеальним інструментом для створення симуляційного середовища, де можна швидко тестувати функціональність модуля, перевіряти алгоритми сортування та відпрацьовувати сценарії обробки даних. SQLite повністю задовольнила потреби симуляції, забезпечуючи надійне зберігання даних і ефективну обробку запитів у контрольованих умовах.

Однак варто зазначити, що для реальних промислових систем, де вимоги до масштабованості, паралельної обробки великого обсягу даних і високої доступності є критично важливими, SQLite може бути менш оптимальним вибором.

Незважаючи на це, для цілей симуляції SQLite виявилася достатньо ефективною, дозволяючи розробникам зосередитися на відпрацюванні логіки модуля без необхідності розгортання складної серверної інфраструктури.

У реальних умовах промислового виробництва вибір бази даних залежатиме від конкретних вимог, таких як обсяг даних, частота транзакцій і потреба в інтеграції з іншими корпоративними системами, наприклад, ERP або SCADA, що взаємодіють із MES через OPC UA.

База даних відіграє центральну роль у забезпеченні всіх ключових функцій модуля, включаючи зберігання та обробку інформації про труби, замовлення, складські зони, а також фіксацію подій і відстеження відвантажень.

Вона підтримує пріоритизацію замовлень на основі дедлайнів, автоматичне перенаправлення бракованих труб до зони «Defect», контроль місткості складських зон і логування всіх значущих подій для подальшого аналізу.

Архітектура бази даних спроектована з урахуванням вимог до гнучкості, надійності та масштабованості, що дозволяє адаптувати її до нових типів труб, додаткових зон або змін у виробничих процесах без значних модифікацій.

Для промислових умов рекомендуються такі альтернативи:

1) PostgreSQL – Ця база даних підтримує складні запити, розширене індексування, реплікацію та паралельну обробку, що ідеально підходить для інтеграції з MES через OPC UA з затримкою до 100 мс. Вона також забезпечує надійне шифрування та управління доступом, що важливо для промислових систем;

2) MySQL – відома своєю швидкістю та ефективністю для операцій читання/запису, MySQL підходить для систем із високою частотою транзакцій, таких як відстеження штрих-кодів на конвеєрі;

3) Microsoft SQL Server – ця база даних пропонує розширені можливості для інтеграції з корпоративними системами, аналітики та високої доступності, що може бути корисним для великих виробництв із складною інфраструктурою.

Ці бази даних забезпечують кращу підтримку масштабування, обробки великих обсягів даних і забезпечення безпеки, що є важливим для реальних промислових умов, де система має працювати безперебійно 24/7.

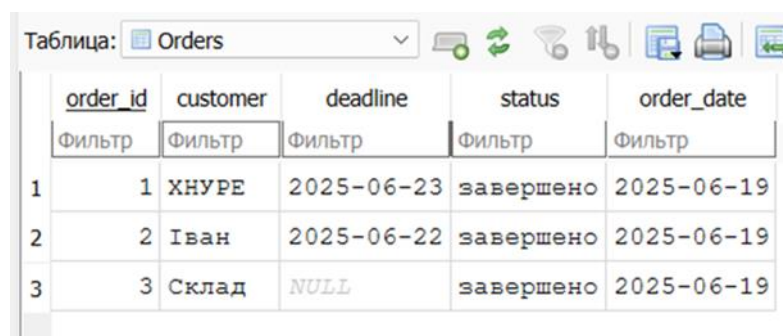
База даних для симуляції складається з шести реляційних таблиць, які охоплюють усі аспекти сортувально-логістичних процесів, включаючи управління замовленнями, ідентифікацію та маршрутизацію труб, контроль складських зон, фіксацію подій і відстеження відвантажень. Кожна таблиця має чітко визначену структуру, що забезпечує ефективне зберігання, обробку та швидкий доступ до

даних. Зв'язки між таблицями реалізовані через зовнішні ключі, що гарантують цілісність даних і підтримують логічну послідовність операцій.

Нижче наведено опис кожної таблиці, їх призначення та зв'язків, що відповідає вимогам, описаним у попередніх розділах:

1) Orders (рисунок 3.1) – таблиця призначена для зберігання інформації про замовлення, які є основою для планування та пріоритизації сортувальних процесів. Вона містить наступні поля:

- а) order_id (унікальний ідентифікатор замовлення, первинний ключ);
- б) customer (назва або ідентифікатор клієнта, наприклад, назва компанії);
- в) deadline (дата завершення замовлення, яка визначає пріоритетність);
- г) status (поточний статус замовлення, наприклад, «в обробці», «завершено», «очікує відвантаження»);
- г) order_date (дата створення замовлення для відстеження хронології);
- д) поле deadline відіграє ключову роль у визначенні пріоритету замовлень, що дозволяє модулю оптимізувати маршрутизацію труб.



order_id	customer	deadline	status	order_date
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1 ХНУРЕ	2025-06-23	завершено	2025-06-19
2	2 Іван	2025-06-22	завершено	2025-06-19
3	3 Склад	NULL	завершено	2025-06-19

Рисунок 3.1 – Таблиця Orders

2) OrderDetails (рисунок 3.2) – ця таблиця деталізує склад замовлень, зберігаючи інформацію про кількість і типи труб, необхідних для виконання кожного замовлення. Поля включають:

- а) order_id (зовнішній ключ, що пов'язує з таблицею Orders);
- б) purpose (тип або призначення труб, наприклад, «водопровідні», «каналізаційні»);

в) pipe_count (кількість труб певного типу для замовлення).

Таблиця забезпечує відстеження потреб замовлень і підтримує логіку пріоритизації.

	<i>order_id</i>	purpose	pipe_count
	Фільтр	Фільтр	Фільтр
1	1	водопровідні	40
2	2	водопровідні	10
3	2	газові	10
4	3	водопровідні	43
5	3	водопровідні	67

Рисунок 3.2 – Таблиця OrderDetails

3) Pipes (рисунок 3.3) – таблиця керує інформацією про кожну трубу, що проходить через сортувально-логістичний процес. Таблиця підтримує автоматичне оновлення статусу бракованості на основі даних від системи комп'ютерного зору та ультразвукового дефектоскопа. Поля:

- а) pipe_id (унікальний ідентифікатор труби, первинний ключ, відповідає штрих-коду Code 128);
- б) purpose (тип труби, узгоджений з OrderDetails);
- в) status (поточний статус, наприклад, «в обробці», «бракована», «ГОТОВА»);
- г) order_id (зовнішній ключ до Orders);
- г) zone_name (зона призначення, наприклад, «Defect» або складська зона);
- д) status_time (час останньої зміни статусу для відстеження хронології).

Таблиця: Pipes

	pipe_id	purpose	status	order_id	zone_name	status_time
	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1	водопровідні	готово	1	Send2	2025-06-19 16:08:24
2	2	водопровідні	готово	1	Send2	2025-06-19 16:08:24
3	3	водопровідні	готово	1	Send2	2025-06-19 16:08:24
4	4	водопровідні	готово	1	Send2	2025-06-19 16:08:24
5	5	водопровідні	готово	1	Send2	2025-06-19 16:08:24
6	6	водопровідні	готово	1	Send2	2025-06-19 16:08:24
7	7	водопровідні	готово	1	Send2	2025-06-19 16:08:24
8	8	водопровідні	готово	1	Send2	2025-06-19 16:08:24
9	9	водопровідні	готово	1	Send2	2025-06-19 16:08:24
10	10	водопровідні	готово	1	Send2	2025-06-19 16:08:24
11	11	водопровідні	готово	1	Send2	2025-06-19 16:08:24
12	12	водопровідні	готово	1	Send2	2025-06-19 16:08:24
13	13	водопровідні	готово	1	Send2	2025-06-19 16:08:24
14	14	водопровідні	готово	1	Send2	2025-06-19 16:08:24
15	15	водопровідні	готово	1	Send2	2025-06-19 16:08:24
16	16	водопровідні	готово	1	Send2	2025-06-19 16:08:24
17	17	водопровідні	готово	1	Send2	2025-06-19 16:08:24
18	18	водопровідні	готово	1	Send2	2025-06-19 16:08:24
19	19	водопровідні	готово	1	Send2	2025-06-19 16:08:24
20	20	водопровідні	готово	1	Send2	2025-06-19 16:08:24
21	21	водопровідні	готово	1	Send2	2025-06-19 16:08:24
22	22	водопровідні	готово	1	Send2	2025-06-19 16:08:24
23	23	водопровідні	готово	1	Send2	2025-06-19 16:08:24
24	24	водопровідні	готово	1	Send2	2025-06-19 16:08:24

1 - 24 of 174

Рисунок 3.3 – Таблиця Pipes

4) StorageZones (рисунок 3.4) – таблиця контролює складські зони, забезпечуючи управління їх місткістю. Таблиця дозволяє модулю зупиняти виробництво при переповненні зон. Поля:

- а) zone_id (унікальний ідентифікатор зони, первинний ключ);
- б) zone_name (назва зони, наприклад, «Склад А», «Defect»);
- в) purpose (тип труб, які зберігаються в зоні);

- г) `current_count` (поточна кількість труб у зоні);
- г) `available_count` (максимальна місткість зони).

Таблиця: StorageZones

<u>zone_id</u>	zone_name	purpose	current_count	available_count
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1 StorageVT	водопровідні	210	250
2	2 StorageGT	газові	100	250
3	3 TempVT	водопровідні	10	75
4	4 TempGT	газові	10	75
5	5 Defect	NULL	9	50
6	6 Send1	NULL	20	200
7	7 Send2	NULL	40	200

Рисунок 3.4 – Таблиця StorageZones

5) Events (рисунок 3.5) – таблиця фіксує всі значущі події в процесі роботи модуля для забезпечення прозорості та аналізу. Логування подій підтримує аналіз процесів. Поля:

- а) `event_id` (унікальний ідентифікатор події, первинний ключ);
- б) `event_time` (час події);
- в) `event_message` (опис події, наприклад, «Виявлено брак», «Завершено замовлення»).

Таблиця: Events

<u>event_id</u>	event_time	event_message
Фільтр	Фільтр	Фільтр
1	2025-06-19 16:08:23	Початок виробництва замовлення ID:2
2	2025-06-19 16:08:24	Початок виробництва замовлення ID:1
3	2025-06-19 16:08:24	Початок виробництва замовлення ID:3
4	2025-06-19 16:08:24	Виявлено брак труби ID:110. Додано заміну.
5	2025-06-19 16:08:24	Виявлено брак труби ID:160. Додано заміну.
6	2025-06-19 16:08:25	Замовлення ID:2 завершено.
7	2025-06-19 16:08:25	Замовлення ID:1 завершено.
8	2025-06-19 16:08:25	Замовлення ID:3 завершено.

Рисунок 3.5 – Таблиця Events

б) ShippedPipes (рисунок 3.6) – таблиця відстежує відвантаження труб, зберігаючи інформацію про завершені замовлення. Таблиця забезпечує відстеження кінцевого етапу логістичного процесу. Поля:

- а) pipe_id (зовнішній ключ до Pipes);
- б) purpose (тип труби);
- в) order_id (зовнішній ключ до Orders);
- г) zone_name (зона відвантаження);
- г) shipped_time (час відвантаження).

Таблиця: ShippedPipes

pipe_id	purpose	order_id	zone_name	shipped_time
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1 водопровідні	1	Send2	2025-06-19 16:16:46
2	2 водопровідні	1	Send2	2025-06-19 16:16:46
3	3 водопровідні	1	Send2	2025-06-19 16:16:46
4	4 водопровідні	1	Send2	2025-06-19 16:16:46
5	5 водопровідні	1	Send2	2025-06-19 16:16:46
6	6 водопровідні	1	Send2	2025-06-19 16:16:46
7	7 водопровідні	1	Send2	2025-06-19 16:16:46
8	8 водопровідні	1	Send2	2025-06-19 16:16:46
9	9 водопровідні	1	Send2	2025-06-19 16:16:46
10	10 водопровідні	1	Send2	2025-06-19 16:16:46
11	11 водопровідні	1	Send2	2025-06-19 16:16:46
12	12 водопровідні	1	Send2	2025-06-19 16:16:46
13	13 водопровідні	1	Send2	2025-06-19 16:16:46
14	14 водопровідні	1	Send2	2025-06-19 16:16:46
15	15 водопровідні	1	Send2	2025-06-19 16:16:46
16	16 водопровідні	1	Send2	2025-06-19 16:16:46
17	17 водопровідні	1	Send2	2025-06-19 16:16:46
18	18 водопровідні	1	Send2	2025-06-19 16:16:46
19	19 водопровідні	1	Send2	2025-06-19 16:16:46
20	20 водопровідні	1	Send2	2025-06-19 16:16:46
21	21 водопровідні	1	Send2	2025-06-19 16:16:46
22	22 водопровідні	1	Send2	2025-06-19 16:16:46
23	23 водопровідні	1	Send2	2025-06-19 16:16:46
24	24 водопровідні	1	Send2	2025-06-19 16:16:46

1 - 24 of 60

Рисунок 3.6 – Таблиця ShippedPipes

Зв'язки між таблицями через зовнішні ключі (наприклад, `order_id` у таблицях `OrderDetails`, `Pipes`, `ShippedPipes`) забезпечують реляційну цілісність, що запобігає невідповідностям у даних. Наприклад, кожна труба в таблиці `Pipes` пов'язана з конкретним замовленням через `order_id`, що дозволяє точно відстежувати її маршрут і статус. База підтримує автоматичне перенаправлення бракованих труб до зони «Defect» на основі даних від систем комп'ютерного зору та ультразвукового дефектоскопа, а також формує запити до MES через OPC UA для заміни браку.

Архітектура бази даних спроектована з урахуванням гнучкості, що дозволяє додавати нові типи труб шляхом оновлення поля `purpose` у таблицях `OrderDetails`, `Pipes` і `StorageZones`, а також нові складські зони через таблицю `StorageZones`, це відповідає вимогам масштабованості. Використання транзакцій SQLite (через виклик `conn.commit()`) гарантує надійність запису даних, що є критично важливим для безперервної роботи в промислових умовах. Компактність SQLite спрощує створення резервних копій і забезпечує швидке відновлення даних у разі збоїв, що підвищує надійність системи.

Дані зі сканерів штрих-кодів записуються до таблиці `Pipes`, а інформація від систем комп'ютерного зору та ультразвукового дефектоскопа оновлює поле `status`, що забезпечує точне відстеження браку. Контроль місткості зон через порівняння `current_count` і `available_count` у таблиці `StorageZones` дозволяє модулю зупиняти виробництво при переповненні.

Для цілей симуляції SQLite виявилася оптимальним вибором, забезпечуючи швидке розгортання, ефективну обробку даних і надійну підтримку всіх функцій модуля, включаючи пріоритизацію замовлень, обробку браку, контроль зон і логування подій.

Її продуктивність і простота дозволили успішно протестувати логістично-сортувальні процеси. Однак для реальних промислових систем, де потрібна обробка великих обсягів даних, висока паралельність і розподілена архітектура, доцільно використовувати більш потужні бази даних, такі як PostgreSQL або

MySQL, які краще відповідають вимогам масштабованості та надійності в умовах безперервного виробництва.

Таким чином, SQLite ідеально підходить для етапу розробки та тестування, тоді як для промислового впровадження слід розглянути альтернативні рішення, адаптовані до складніших умов експлуатації.

3.2 Створення вебресурсу для тестування

Для моделювання роботи модуля автоматизованого керування сортуванням поліетиленових труб було розроблено вебресурс, який слугує як інструмент для симуляції логістичних і сортувальних процесів у контрольованому середовищі. Цей вебресурс створено виключно для цілей симуляції, дозволяючи тестувати функціональність модуля, перевіряти взаємодію з базою даних і відпрацьовувати сценарії обробки замовлень без необхідності розгортання реального промислового обладнання.

Вебресурс побудовано на основі фреймворку Flask із використанням Python для серверної логіки та шаблонізатора Jinja2 для створення динамічних HTML-сторінок [16].

Такий підхід забезпечує простоту розробки, гнучкість у налаштуванні та ефективну інтеграцію з базою даних SQLite, яка використовується для симуляції.

Для реальних промислових систем, де потрібна висока продуктивність, обробка великої кількості одночасних користувачів і розширені функції безпеки, зазвичай застосовуються більш складні технології, такі як Django з PostgreSQL або Node.js із MongoDB.

Однак для цілей симуляції Flask і SQLite виявилися оптимальними, дозволяючи швидко реалізувати необхідний функціонал і протестувати ключові аспекти роботи модуля, включаючи обробку замовлень, пріоритизацію та розподіл труб.

Основними компонентами вебресурсу є файл `app.py`, який відповідає за серверну логіку, обробку HTTP-запитів і взаємодію з базою даних `pipes.db`, а також

шаблон `index.html`, який формує користувацький інтерфейс для взаємодії з системою.

Додатково вебресурс інтегрується з модулем `sorting_module.py`, який імітує логіку сортування труб і їх розподілу по складських зонах. Інтерфейс вебресурсу, зображений на рисунку 3.7, забезпечує зручне введення даних про замовлення, відображення вмісту корзини та управління процесом їх обробки.

Замовлення труб

Зовнішнє замовлення додано до корзини!

Зовнішнє замовлення

Ім'я/Компанія:

Тип труби:

Кількість:

Дедлайн (YYYY-MM-DD):

Внутрішнє замовлення

Тип труби:

Кількість:

Корзина

Замовник	Тип	Кількість	Дедлайн	Ціна
ХНУРЕ	водопровідні	30	2025-06-22	12000 грн

Загальна ціна: 12000 грн

Рисунок 3.7 – Інтерфейс вебресурсу

Використання Flask дозволяє обробляти запити користувачів у реальному часі, забезпечуючи динамічне оновлення інтерфейсу через шаблони Jinja2. Python, як мова серверної логіки, забезпечує простоту реалізації складних операцій, таких як взаємодія з базою даних, обробка форм і управління сесіями. SQLite, обрана для симуляції, забезпечує швидке зберігання та обробку даних, що дозволяє тестувати функціонал модуля без значних витрат ресурсів.

Серверна логіка, реалізована в `app.py`, відповідає за обробку HTTP-запитів, управління сесіями користувачів і взаємодію з базою даних. Для підключення до бази даних `pipes.db` використовується функція `get_db()`, яка створює з'єднання з SQLite і налаштовує повернення даних у форматі словників для зручної обробки:

```
def get_db():
    conn = sqlite3.connect('pipes.db')
    conn.row_factory = sqlite3.Row
    return conn
```

Сесії Flask використовуються для зберігання тимчасових даних, таких як корзина замовлень (cart) і інформація про замовника (customer). При ініціалізації сесії створюються порожні значення, якщо вони ще не визначені:

```
if 'cart' not in session:
    session['cart'] = []
if 'customer' not in session:
    session['customer'] = ''
```

Основна функція index() обробляє GET- і POST-запити, відповідаючи за відображення інтерфейсу та обробку дій користувача. Наприклад, для додавання зовнішнього замовлення реалізовано наступну логіку:

```
if 'add_external' in request.form:
    customer = request.form['customer']
    session['customer'] = customer
    purpose = request.form['purpose']
    count = int(request.form['count'])
    deadline = request.form['deadline']
    price = 400 if purpose == 'водопровідні' else 600
    item = {
        'customer': customer,
        'purpose': purpose,
        'count': count,
        'deadline': deadline,
        'price': count * price
    }
    cart.append(item)
    session['cart'] = cart
```

```
session.modified = True
```

Цей код обробляє введені користувачем дані (замовник, тип труб, кількість, дедлайн), обчислює ціну залежно від типу труб (400 грн за водопровідні, 600 грн за інші) і додає замовлення до корзини, зберігаючи його в сесії. Внутрішні замовлення додаються аналогічно, але без дедлайну та з замовником «Склад», що відповідає логіці складських операцій.

При підтвердженні замовлення дані записуються до таблиць бази даних Orders, OrderDetails і Pipes за допомогою SQL-запитів через параметризовані запити. Наприклад:

```
cursor.execute("""
    INSERT INTO Orders (customer, deadline, status, order_date)
    VALUES (?, ?, 'в обробці', ?)
""", (cust, dl, datetime.now().strftime('%Y-%m-%d')))
order_id = cursor.lastrowid
for item in items:
    cursor.execute("""
        INSERT INTO OrderDetails (order_id, purpose, pipe_count)
        VALUES (?, ?, ?)
""", (order_id, item['purpose'], item['count']))
    for _ in range(item['count']):
        cursor.execute("""
            INSERT INTO Pipes (purpose, status, order_id, status_time)
            VALUES (?, 'в обробці', ?, ?)
""", (item['purpose'], order_id, datetime.now().strftime('%Y-%m-%d
%H:%M:%S')))
```

Ця логіка забезпечує створення нового замовлення, додавання деталей про типи та кількість труб, а також реєстрацію кожної труби з початковим статусом «в обробці» та актуальним часом.

Шаблон index.html, створений із використанням Jinja2, формує інтерактивний інтерфейс із формами для створення зовнішніх і внутрішніх

замовлень, таблицею для відображення корзини та кнопками для управління замовленнями. Таблиця корзини має наступну структуру:

```

<table>
  <tr>
    <th>Замовник</th><th>Тип</th><th>Кількість</th><th>Дедлайн</th><th>Ціна</th>
  >
  </tr>
  {% for item in cart %}
  <tr>
    <td>{{ item.customer or 'Внутрішнє' }}</td>
    <td>{{ item.purpose }}</td>
    <td>{{ item.count }}</td>
    <td>{{ item.deadline or '-' }}</td>
    <td>{{ item.price }} грн</td>
  </tr>
  {% endfor %}
</table>

```

Таблиця відображає вміст корзини, показуючи замовника, тип труб, кількість, дедлайн (або «-» для внутрішніх замовлень) і загальну ціну. Форми дозволяють вводити дані для нових замовлень, а кнопки «Замовити» та «Очистити» керують збереженням або видаленням корзини. У разі помилок, наприклад, неправильного формату дедлайну, повідомлення виводяться через функцію flash, що забезпечує зручну взаємодію з користувачем.

Розроблений вебресурс на основі Flask і SQLite є ефективним інструментом для симуляції роботи модуля автоматизованого сортування. Він забезпечує зручний інтерфейс, надійну обробку даних і інтеграцію з модулем сортування, дозволяючи тестувати всі ключові функції системи в контрольованому середовищі. Для реальних промислових систем доцільно використовувати більш потужні

технології, але для цілей симуляції цей вебресурс повністю відповідає поставленим завданням, забезпечуючи швидке налаштування, гнучкість і точність моделювання.

3.3 Програмна реалізація з допомогою Python

Програмна реалізація модуля автоматизованого керування сортуванням поліетиленових труб розроблена на мові Python, яка була обрана завдяки своїй універсальності, простоті розробки та потужним бібліотекам [17], що ідеально підходять як для симуляції, так і для створення реальних промислових систем.

Ця реалізація створена передусім для моделювання роботи модуля в контрольованому середовищі, дозволяючи тестувати ключові функції, такі як обробка замовлень, маршрутизація труб, контроль якості та управління складськими зонами.

Водночас Python є одним із найкращих варіантів для реалізації подібних модулів навіть у реальних промислових умовах завдяки своїй гнучкості, широкій екосистемі бібліотек і здатності інтегруватися з апаратними компонентами через протоколи, такі як OPC UA.

Для симуляції Python забезпечує швидке розгортання, легкість налаштування та зручність тестування, що робить його оптимальним вибором для розробки та відпрацювання логіки модуля.

Реалізація включає вебдодаток для створення та управління замовленнями, модуль сортування для імітації логістичних операцій і базу даних для зберігання інформації. Python дозволяє ефективно поєднувати ці компоненти, забезпечуючи автоматизацію всіх етапів – від введення замовлення до відвантаження труб.

Для реальних промислових систем Python також є потужним інструментом, оскільки підтримує інтеграцію з програмованими логічними контролерами (PLC), обробку великих обсягів даних і створення масштабованих рішень.

У цьому розділі описано структуру програмної реалізації, її ключові функції та переваги використання Python для симуляції та потенційного промислового застосування.

Програмна реалізація складається з двох основних компонентів: файлу `app.py`, який відповідає за вебінтерфейс і обробку замовлень, та файлу `sorting_module.py`, який реалізує логіку сортування, маршрутизації труб і управління складськими зонами. Використані бібліотеки Python включають:

- `sqlite3` для взаємодії з базою даних `pipes.db`, що забезпечує зберігання інформації про замовлення, труби, зони та події;
- `Flask` для створення вебдодатку, який дозволяє користувачам вводити замовлення та відстежувати їхній статус;
- `tabulate` для форматування виведення подій у консолі, що полегшує аналіз результатів симуляції.

Ці бібліотеки забезпечують швидку розробку, надійну обробку даних і зручне тестування. Python, завдяки своїй простоті синтаксису та широкій підтримці бібліотек, дозволяє створювати як прототипи для симуляції, так і повноцінні промислові додатки, інтегровані з апаратним забезпеченням, наприклад, через OPC UA для зв'язку з PLC Siemens S7-1200. Результати роботи програми, включаючи виведення подій і статусів, зображено на рисунку 3.8.

```

def sorting_module():
    conn.commit()

```

Усі таблиці є:

- Подія: Початок виробництва замовлення ID:3
- Подія: Початок виробництва замовлення ID:1
- Подія: Вивілено брак труби ID:50. Додано заміну.
- Подія: Вивілено брак труби ID:100. Додано заміну.
- Подія: Початок виробництва замовлення ID:2
- Подія: Замовлення ID:3 завершено.
- Подія: Замовлення ID:1 завершено.
- Подія: Замовлення ID:2 завершено.

Час	Подія
2025-06-19 16:20:59	Початок виробництва замовлення ID:3
2025-06-19 16:20:59	Початок виробництва замовлення ID:1
2025-06-19 16:20:59	Вивілено брак труби ID:50. Додано заміну.
2025-06-19 16:20:59	Вивілено брак труби ID:100. Додано заміну.
2025-06-19 16:20:59	Початок виробництва замовлення ID:2
2025-06-19 16:21:00	Замовлення ID:3 завершено.
2025-06-19 16:21:00	Замовлення ID:1 завершено.
2025-06-19 16:21:00	Замовлення ID:2 завершено.

PS C:\Users\denbi\pipes_project>

Рисунок 3.8 – Результат виконання в Python

Файл `app.py` реалізує вебдодаток на основі Flask, який забезпечує інтерактивний інтерфейс для створення зовнішніх і внутрішніх замовлень. Зовнішні замовлення містять дані про замовника, тип труб (водопровідні за 400 грн або газові за 600 грн), кількість і дедлайн, тоді як внутрішні замовлення обмежуються типом і кількістю труб із замовником «Склад». Для підключення до бази даних використовується функція `get_db()`, яка створює з'єднання з SQLite і налаштовує доступ до даних у форматі словників:

```
def get_db():
    conn = sqlite3.connect('pipes.db')
    conn.row_factory = sqlite3.Row
    return conn
```

Ця функція забезпечує зручну роботу з даними, дозволяючи обробляти їх як об'єкти. Дані про замовлення тимчасово зберігаються в сесії Flask через ключі `session['cart']` для корзини та `session['customer']` для замовника. При підтвердженні замовлення дані записуються до таблиць `Orders`, `OrderDetails` і `Pipes` за допомогою параметризованих SQL-запитів:

```
cursor.execute("""
    INSERT INTO Orders (customer, deadline, status, order_date)
    VALUES (?, ?, 'в обробці', ?)
""", (cust, dl, datetime.now().strftime('%Y-%m-%d'))
order_id = cursor.lastrowid
```

Цей код створює нове замовлення, отримує його унікальний ідентифікатор і додає деталі про типи та кількість труб, а також реєструє кожну трубу з початковим статусом «в обробці». Для обробки помилок, таких як неправильний формат дедлайну, використовуються блоки `try-except`, які виводять повідомлення через функцію `flash`, забезпечуючи зручну взаємодію з користувачем.

Файл `sorting_module.py` відповідає за імітацію сортувальних і логістичних операцій, включаючи маршрутизацію труб, обробку бракованих одиниць і контроль місткості складських зон.

Модуль починається з функції `check_tables()`, яка перевіряє наявність необхідних таблиць у базі даних (`Orders`, `OrderDetails`, `Pipes`, `StorageZones`, `Events`, `ShippedPipes`), забезпечуючи коректну ініціалізацію системи перед початком роботи.

Для фіксації подій, таких як створення замовлення чи виявлення браку, використовується функція `log_event()`, яка записує повідомлення до таблиці `Events` із зазначенням часу та виводить їх у консоль для моніторингу:

```
def log_event(message):
    cursor.execute("INSERT INTO Events (event_time, event_message) VALUES (?, ?)",
                  (datetime.now().strftime('%Y-%m-%d %H:%M:%S'), message))
    conn.commit()
    print(f"Подія: {message}")
```

Функція `get_orders()` отримує активні замовлення, сортуючи їх за дедлайном для забезпечення пріоритизації:

```
cursor.execute("""
SELECT DISTINCT o.order_id, o.deadline, o.status, o.order_date, o.customer
FROM Orders o
JOIN OrderDetails od ON o.order_id = od.order_id
WHERE o.status = 'в обробці'
ORDER BY CASE WHEN o.deadline IS NULL THEN 1 ELSE 0 END, o.deadline
ASC, o.order_date ASC
""")
```

Цей запит забезпечує обробку замовлень із найближчими дедлайнами в першу чергу, надаючи внутрішнім замовленням нижчий пріоритет.

Функція `sort_pipes()` імітує сортування труб, умовно позначаючи кожну 50-ту трубу як браковану для тестування логіки обробки дефектів:

```
if pipe_counter % 50 == 0:
    defect_count += 1
    zone = next((z for z in zones if z[0] == 'Defect'), None)
    if zone and zone[2] < zone[3]:
```

```
sorted_pipes.append((pipe_id, pipe_purpose, 'готово', None, 'Defect'))  
update_pipe_status(pipe_id, None, 'готово', 'Defect')  
update_zone_count('Defect', 1)
```

Браковані труби перенаправляються до зони «Defect», якщо вона не переповнена, а їхній статус оновлюється в таблиці Pipes. Внутрішні замовлення спрямовуються до зон StorageVT або StorageGT для водопровідних і газових труб відповідно, або до тимчасових зон TempVT і TempGT при переповненні. Зовнішні замовлення направляються до зон відвантаження Send1 або Send2. У разі переповнення зон, модуль зупиняє обробку, імітуючи припинення виробництва.

Функція ship_pipes() завершує процес, переносячи готові труби до таблиці ShippedPipes, оновлюючи місткість зон у таблиці StorageZones і змінюючи статус виконаних замовлень на «завершено» в таблиці Orders.

Транзакції SQLite через conn.commit() забезпечують цілісність даних, що дозволяє уникнути невідповідностей під час симуляції. Код спроектовано з урахуванням масштабованості: нові типи труб додаються шляхом оновлення таблиці StorageZones і внесення змін у правила маршрутизації.

Наприклад, для підтримки каналізаційних труб достатньо додати нові записи до бази даних і відповідні зони.

Для реальних систем Python може бути доповнений більш потужними базами даних, такими як PostgreSQL, для обробки великих обсягів транзакцій, або розгорнутий на серверних платформах із балансуванням навантаження. Це забезпечує адаптацію до промислових умов, де потрібна висока надійність і стійкість до відмов.

Хоча Python є потужним інструментом, для промислових систем із високими вимогами до реального часу та надійності можуть знадобитися спеціалізовані рішення, такі як програмування PLC мовами IEC 61131-3 або використання промислових платформ, наприклад, Ignition від Inductive Automation.

Для симуляції Python із Flask і SQLite забезпечує швидке розгортання та точне моделювання всіх процесів, що робить його ідеальним для тестування.

Програмна реалізація на Python із використанням SQLite, Flask і Tabulate є ефективним рішенням для симуляції роботи модуля автоматизованого сортування.

Вона забезпечує повну автоматизацію обробки замовлень, сортування труб і управління зонами, дозволяючи тестувати всі функції в контрольованому середовищі.

Python також є одним із найкращих варіантів для реалізації таких модулів у реальних умовах завдяки своїй гнучкості та підтримці інтеграції з апаратним забезпеченням. Для промислового використання можуть знадобитися додаткові оптимізації, але для симуляції ця реалізація повністю відповідає вимогам, забезпечуючи швидкість, зручність і точність моделювання.

3.4 Тестування

Тестування модуля автоматизованого керування сортуванням поліетиленових труб проводилося в симуляційному середовищі, створеному для перевірки коректності роботи програмних компонентів (app.py, sorting_module.py) та бази даних (pipes.db) [18].

Метою було оцінити функціональність модуля, зокрема створення та обробку замовлень, пріоритизацію за дедлайнами, маршрутизацію труб до відповідних зон, обробку бракованих одиниць і контроль місткості складських зон.

Тестування виконувалося з використанням Python, який завдяки своїй гнучкості, читабельності та потужним бібліотекам (Flask, SQLite, Tabulate) виявився ідеальним інструментом для симуляції, а також є одним із найкращих варіантів для реалізації подібних систем у реальних промислових умовах. Python забезпечує швидке налаштування тестового середовища, легку інтеграцію з базою даних і зручність аналізу результатів, що дозволяє детально перевірити всі аспекти роботи модуля.

Симуляційне середовище дозволило відтворити логістично-сортувальні процеси без використання реального обладнання, такого як конвеєри чи PLC Siemens S7-1200, але з моделюванням їхньої поведінки.

Тестування охопило всі ключові функції модуля, включаючи введення замовлень через вебінтерфейс, автоматичне сортування труб, обробку переповнення зон і фіксацію подій. Результати тестування підтвердили здатність модуля ефективно обробляти замовлення, забезпечувати пріоритизацію та правильно розподіляти труби, що відповідає вимогам, описаним у попередніх розділах.

Тестування проводилося з використанням початкових даних, заданих у таблиці StorageZones бази даних pipes.db. Складські зони були ініціалізовані наступним чином:

- StorageVT: 100 водопровідних труб, вільно 150 з 250;
- StorageGT: 100 газових труб, вільно 150 з 250;
- TempVT: 10 водопровідних труб, вільно 65 з 75;
- TempGT: 10 газових труб, вільно 65 з 75;
- Defect: 5 бракованих труб, вільно 45 з 50;
- Send1 і Send2: порожні, вільно 200 з 200 для кожної.

Для тестування було створено три замовлення через вебінтерфейс (index.html), які відображають різні сценарії використання:

- замовлення №1: Замовник «ХНУРЕ», дедлайн 2025-07-03, 50 водопровідних труб, 25 газових труб;
- замовлення №2: Замовник «Склад», без дедлайну, 200 водопровідних труб, 50 газових труб;
- замовлення №3: Замовник «Іван», дедлайн 2025-06-26, 10 водопровідних труб, 10 газових труб.

Дані замовлень, введені через Flask-інтерфейс, зображено на рисунку 3.9.

order_id	customer	deadline	status	order_date
1	ХНУРЕ	2025-07-03	в обробці	2025-06-19
2	Склад	NULL	в обробці	2025-06-19
3	Іван	2025-06-26	в обробці	2025-06-19

order_id	purpose	pipe_count
1	водопровідні	50
1	газові	25
2	водопровідні	200
2	газові	50
3	водопровідні	10
3	газові	10

Рисунок 3.9 – Дані замовлень

Замовлення створювалися через вебінтерфейс, реалізований у app.py на основі Flask, як показано на рисунку 3.10. Користувач вводив дані (замовник, тип труб, кількість, дедлайн для зовнішніх замовлень), які записувалися до таблиць Orders, OrderDetails і Pipes після підтвердження. Логіка обробки замовлень у app.py забезпечувала коректне збереження даних у сесії (session['cart']) і їх подальший запис до бази даних із захистом від помилок через блоки try-except і повідомленнями через flash.

Зовнішнє замовлення додано до корзини!

Зовнішнє замовлення

Ім'я/Компанія:

Тип труби:

Кількість:

Дедлайн (YYYY-MM-DD):

Внутрішнє замовлення

Тип труби:

Кількість:

Корзина

Замовник	Тип	Кількість	Дедлайн	Ціна
ХНУРЕ	водопровідні	50	2025-07-03	20000 грн
ХНУРЕ	газові	25	2025-07-03	15000 грн

Загальна ціна: 35000 грн

Рисунок 3.10 – Flask-інтерфейс

Модуль сортування, реалізований у sorting_module.py, обробляв замовлення в порядку пріоритетності, визначеної дедлайнами. Замовлення №3 (дедлайн 2025-06-26) було оброблено першим через найближчу дату завершення, за ним слідувало замовлення №1 (дедлайн 2025-07-03), а замовлення №2 (без дедлайну) оброблялося останнім.

Усього тестування охопило 345 труб (260 водопровідних і 85 газових), що дозволило перевірити всі сценарії маршрутизації, включаючи обробку браку та переповнення зон.

Труби для замовлень «Іван» і «ХНУРЕ» спрямовувалися до зон відвантаження Send1 (водопровідні) і Send2 (газові), тоді як труби для замовлення

«Склад» розподілялися до складських зон StorageVT і StorageGT, а за потреби – до тимчасових зон TempVT і TempGT.

Для замовлення №2, яке включало 200 водопровідних і 50 газових труб, було виявлено 5 бракованих одиниць (4 водопровідні, 1 газова, умовно кожна 50-та труба), які перенаправлено до зони Defect. Заміни для цих труб автоматично додано до зон StorageVT і StorageGT, що зафіксовано в таблиці Events.

Через перевищення місткості StorageVT ($100 + 200 > 250$) 50 водопровідних труб було перенаправлено до TempVT, збільшивши її заповнення до 60 (10 + 50) із 75 доступних. Газові труби для замовлення №2 розміщено в StorageGT, де загальна кількість склала 150 (100 + 50) із 250.

Деталі статусу труб в таблиці Pipes зображено на рисунку 3.11, а відображення розподілених труб по зонах рисунку 3.12.

Table (a) data:

pipe_id	purpose	status	order_id	zone_name	status_time
65	газові	готово	1	Send2	2025-06-19 17:07:46
66	газові	готово	1	Send2	2025-06-19 17:07:46
67	газові	готово	1	Send2	2025-06-19 17:07:46
68	газові	готово	1	Send2	2025-06-19 17:07:46
69	газові	готово	1	Send2	2025-06-19 17:07:46
70	газові	готово	1	Send2	2025-06-19 17:07:46
71	газові	готово	1	Send2	2025-06-19 17:07:46
72	газові	готово	1	Send2	2025-06-19 17:07:46
73	газові	готово	1	Send2	2025-06-19 17:07:46
74	газові	готово	1	Send2	2025-06-19 17:07:46
75	газові	готово	1	Send2	2025-06-19 17:07:46
76	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
77	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
78	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
79	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
80	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
81	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
82	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
83	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
84	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
85	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
86	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
87	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46
88	водопровідні	готово	2	StorageVT	2025-06-19 17:07:46

Table (b) data:

pipe_id	purpose	status	order_id	zone_name	status_time
318	газові	готово	2	StorageGT	2025-06-19 17:07:47
319	газові	готово	2	StorageGT	2025-06-19 17:07:47
320	газові	готово	2	StorageGT	2025-06-19 17:07:47
321	газові	готово	2	StorageGT	2025-06-19 17:07:47
322	газові	готово	2	StorageGT	2025-06-19 17:07:47
323	газові	готово	2	StorageGT	2025-06-19 17:07:47
324	газові	готово	2	StorageGT	2025-06-19 17:07:47
325	газові	готово	NULL	Defect	2025-06-19 17:07:47
326	водопровідні	готово	3	Send1	2025-06-19 17:07:46
327	водопровідні	готово	3	Send1	2025-06-19 17:07:46
328	водопровідні	готово	3	Send1	2025-06-19 17:07:46
329	водопровідні	готово	3	Send1	2025-06-19 17:07:46
330	водопровідні	готово	3	Send1	2025-06-19 17:07:46
331	водопровідні	готово	3	Send1	2025-06-19 17:07:46
332	водопровідні	готово	3	Send1	2025-06-19 17:07:46
333	водопровідні	готово	3	Send1	2025-06-19 17:07:46
334	водопровідні	готово	3	Send1	2025-06-19 17:07:46
335	водопровідні	готово	3	Send1	2025-06-19 17:07:46
336	газові	готово	3	Send1	2025-06-19 17:07:46
337	газові	готово	3	Send1	2025-06-19 17:07:46
338	газові	готово	3	Send1	2025-06-19 17:07:46
339	газові	готово	3	Send1	2025-06-19 17:07:46
340	газові	готово	3	Send1	2025-06-19 17:07:46
341	газові	готово	3	Send1	2025-06-19 17:07:46

а)

б)

Рисунок 3.11 – Дані труб у Pipes

Таблиця: StorageZones

zone_id	zone_name	purpose	current_count	available_count
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1 StorageVT	водопровідні	250	250
2	2 StorageGT	газові	150	250
3	3 TempVT	водопровідні	60	75
4	4 TempGT	газові	10	75
5	5 Defect	NULL	11	50
6	6 Send1	NULL	20	200
7	7 Send2	NULL	75	200

Рисунок 3.12 – Розподіл по зонам

Усі значущі події, такі як початок обробки замовлення, виявлення браку, переповнення StorageVT і перенаправлення труб до TempVT, фіксувалися в таблиці Events. Приклади подій включають: «Початок замовлення ID:3», «StorageVT заповнена», «Труби перенаправлено до TempVT».

Лог подій зображено на рисунку 3.13.

Таблиця: Events

event_id	event_time	event_message
Фільтр	Фільтр	Фільтр
1	2025-06-19 17:07:46	Початок виробництва замовлення ID:3
2	2025-06-19 17:07:46	Початок виробництва замовлення ID:1
3	2025-06-19 17:07:46	Виявлено брак труби ID:50. Додано заміну.
4	2025-06-19 17:07:46	Початок виробництва замовлення ID:2
5	2025-06-19 17:07:46	Виявлено брак труби ID:125. Додано заміну.
6	2025-06-19 17:07:47	Виявлено брак труби ID:175. Додано заміну.
7	2025-06-19 17:07:47	Виявлено брак труби ID:225. Додано заміну.
8	2025-06-19 17:07:47	Зона StorageVT заповнена. Труби перенаправлено в TempVT.
9	2025-06-19 17:07:47	Виявлено брак труби ID:275. Додано заміну.
10	2025-06-19 17:07:47	Виявлено брак труби ID:325. Додано заміну.
11	2025-06-19 17:07:47	Замовлення ID:3 завершено.
12	2025-06-19 17:07:47	Замовлення ID:1 завершено.
13	2025-06-19 17:07:47	Замовлення ID:2 завершено.

Рисунок 3.13 – Таблиця подій

Тестування підтвердило коректну роботу модуля в симуляційному середовищі. Модуль успішно обробив три замовлення, що включали 345 труб (260 водопровідних і 85 газових), забезпечивши правильну пріоритизацію за дедлайнами.

Труби для замовлень «Іван» (20 труб) і «ХНУРЕ» (75 труб) були розміщені в зонах відвантаження Send1 і Send2, тоді як труби для замовлення «Склад» розподілено між StorageVT (150 труб), TempVT (50 труб) і StorageGT (50 труб).

Система виявила 5 бракованих труб для замовлення №2, які перенаправлено до зони Defect, із автоматичним додаванням замін до відповідних складських зон.

Переповнення StorageVT було коректно оброблено шляхом перенаправлення 50 труб до TempVT, що супроводжувалося відповідним логуванням. Усі операції, включаючи створення замовлень, сортування, обробку браку та оновлення статусів, фіксувалися в таблиці Events, забезпечуючи повну прозорість процесу.

Тестування в симуляційному середовищі продемонструвало ефективність модуля автоматизованого сортування, підтвердивши його здатність обробляти замовлення з урахуванням пріоритетів, правильно розподіляти труби, виявляти та обробляти брак, а також керувати місткістю зон. Python виявився ідеальним інструментом для створення тестового середовища, забезпечуючи швидкість розробки, гнучкість і зручність аналізу. Його переваги, такі як читабельність коду та підтримка інтеграцій, роблять Python одним із найкращих варіантів як для симуляції, так і для реалізації модуля в реальних промислових системах. Результати тестування підтверджують готовність модуля до подальшої оптимізації та впровадження.

3.5 Тестування системи на стійкість

Тестування системи на стійкість модуля автоматизованого управління сортуванням поліетиленових труб оцінювало надійність контуру регулювання маршрутів труб у промислових умовах.

Модуль, реалізований на базі програмного забезпечення Python, використовує базу даних для зберігання інформації про труби та замовлення, забезпечуючи їх автоматичне сортування після виробництва за заданими пріоритетами. Контур управління маршрутами, який визначає зони призначення (StorageVT, StorageGT, TempVT, TempGT, Defect, Send1, Send2), досліджувався методами теорії автоматичного управління (ТАУ) для підтвердження стійкості та оцінки динамічних характеристик при зміні маршрутів.

Для аналізу ТАУ конвеєрну систему змодельовано як інерційний об'єкт першого порядку з передатною функцією (3.1) [19]:

$$W(s) = \frac{1}{0,5s+1} \quad (3.1)$$

де $T = 0,5$ с – часова постійна .

Відображає затримку перемикання маршруту. Регулятор обрано пропорційним із коефіцієнтом підсилення $Kp = 10$, що задається передатною функцією $W_r(s) = 10$. Система має замкнутий контур із одиничним негативним зворотним зв'язком, де вхідний сигнал це бажана зона призначення з бази даних, а вихідний це фактична зона, до якої спрямована труба. Передатна функція замкнутої системи (3.2):

$$W_{closed}(s) = \frac{W_r(s) \times W(s)}{1 + W_r(s) \times W(s)} = \frac{10}{0,5s+1+10} = \frac{10}{0,5s+11}, \quad (3.2)$$

Характеристичне рівняння замкнутої системи (3.3):

$$0,5s + 11 = 0, \quad (3.3)$$

де $s = -22$ – єдиний корінь.

Розташований у лівій півплощині комплексної площини, що свідчить про стійкість. Для підтвердження стійкості застосовано алгебраїчний критерій Гурвіца.

Для полінома першого порядку $a_1 s + a_0 = 0,5s + 11$ умови стійкості: $a_1 > 0$, $a_0 > 0$. Оскільки $a_1 = 0,5$, $a_0 = 11$, система стійка.

Комп'ютерне моделювання виконано в Python із бібліотекою *control* (версія 0.10.1). Модель замкнутої системи сформовано за допомогою функції *ctrl.feedback*, поєднуючи передатні функції об'єкта ($1/0,5s + 1$) та регулятора ($K_p = 10$). Стійкість перевірено через розташування полюсів за допомогою функції *ctrl.poles*.

Отриманий полюс $s = -22$ підтверджує стійкість, оскільки його реальна частина від'ємна. Для оцінки динамічних характеристик симулювався перехідний процес при ступінчастому впливі, що відповідає зміні маршруту, наприклад, із зони StorageVT на Send1. Час моделювання становив 2 с із розбиттям на 1000 точок. Усталена величина вихідного сигналу розрахована як (3.4):

$$y(\infty) = \frac{K_p}{1+K_p} = \frac{10}{11} \approx 0,909, \quad (3.4)$$

Результати моделювання, отримані за допомогою *ctrl.step_response*, показали, що вихідний сигнал плавно зростає до $y = 0,909$, без ознак коливань. Відносне перегулювання, визначене як (3.5):

$$\sigma = \frac{\max(y) - y(\infty)}{y(\infty)} \times 100\%, \quad (3.5)$$

де σ – відносне перегулювання в % , дорівнює 0%.

Вихід системи після подачі вхідного сигналу не перевищив установленної величини, оскільки максимальне значення $\max(y) \approx 0,909$ збігається з усталеною величиною. Час регулювання, визначений як момент, коли $|y - 0,909| < 0,02 \cdot 0,909$, склав $0,18$ с, що вказує на швидке реагування системи на зміну маршруту. Перехідна характеристика збережена у вигляді графіка (*step_response.png*), який демонструє плавне зростання вихідного сигналу без коливань до рівня $y = 0,909$ (рисунок 3.14).

Це дозволяє ефективно використовувати модуль для автоматизації логістичних процесів на виробництві поліетиленових труб, забезпечуючи високу точність і надійність сортування навіть при інтенсивних навантаженнях.

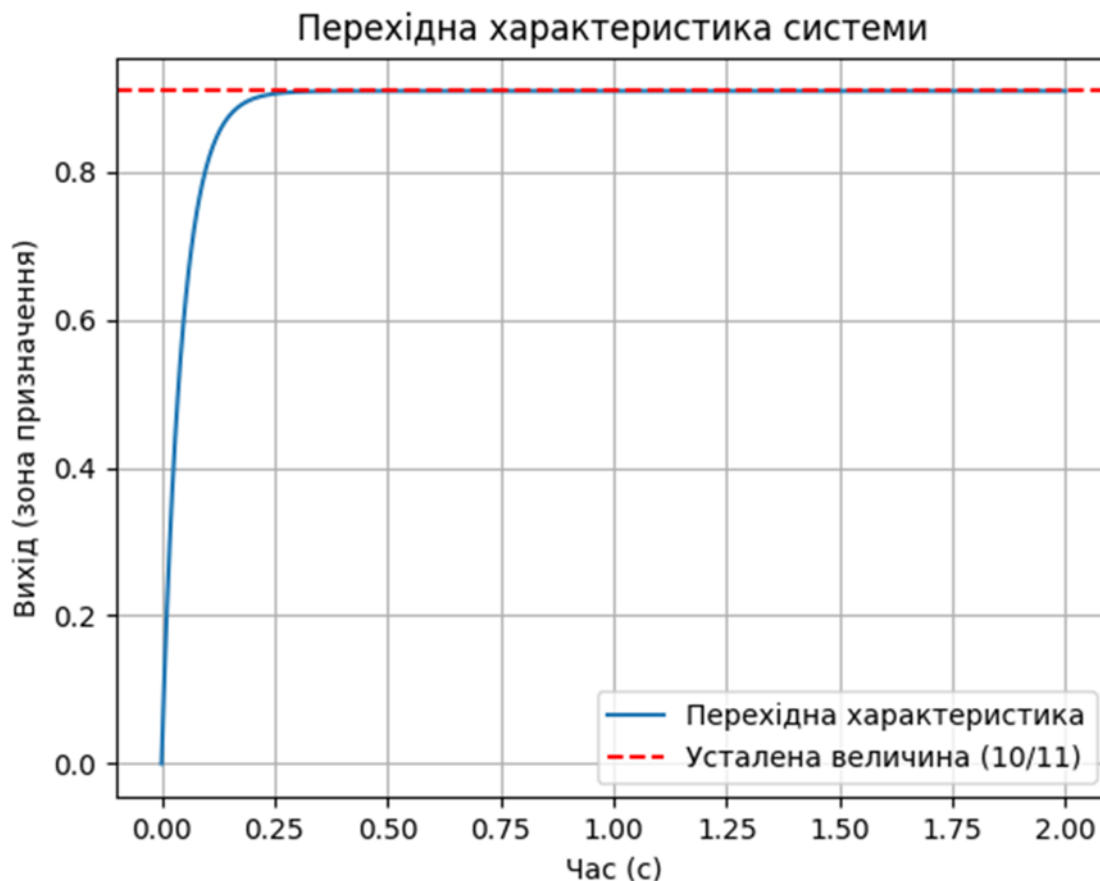


Рисунок 3.14 – Результати моделювання

3.6 Охорона праці

Охорона праці є невід'ємною частиною роботи підприємств, що виготовляють поліетиленові труби [8-9]. Безпека працівників забезпечує не лише захист їхнього здоров'я та життя, але й стабільну роботу обладнання, високу якість продукції та мінімізує аварійні простой. У сучасному виробництві, де застосовуються високотемпературні установки, електромеханічні системи та складні технологічні лінії, охорона праці має бути системною, науково обґрунтованою та безперервною.

Виробництво поліетиленових труб охоплює підготовку полімерів, екструзію, охолодження, калібрування, різання, маркування, сортування, пакування та зберігання. Кожен етап несе ризики: високі температури в екструдерах і термостатах, контакт із рухомими частинами, електричне напруження, шум від обладнання, вібрація, полімерний пил, випари від нагрівання сировини, використання стисненого повітря та ручне переміщення труб.

Для захисту працівників створюється служба охорони праці, яка розробляє інструкції та нормативні акти. Робочі місця регулярно перевіряються на відповідність умовам безпеки, а залучення жінок і неповнолітніх до небезпечних робіт забороняється. Усі працівники проходять медичний огляд перед працевлаштуванням і періодичні перевірки здоров'я. Для робіт із підвищеною небезпекою підприємство отримує дозвіл, а також розробляє план дій на випадок аварій.

Технологічні процеси організуються так, щоб мінімізувати ризики. Екструдери, тягнучі пристрої, ріжучі механізми та конвеєри оснащуються захисними кожухами, блокуваннями запуску при відкритих люках, аварійними кнопками зупинки, датчиками тиску й температури та сигналізацією. Робочі поверхні шнеків і циліндрів не повинні мати подряпин чи вм'ятин, а деталі, що контактують із розплавом, виготовляються з хромистих сталей або мають хромоване покриття.

Очищення обладнання проводиться латунними чи мідними скребками одразу після зупинки, коли температура не нижча за 10–20 °С від температури плавлення полімеру. Експлуатація обладнання з пошкодженими покриттями чи нещільними з'єднаннями забороняється.

У зонах нагрівання полімерів встановлюється припливно-витяжна вентиляція, яка видаляє випари та пил, із автоматичним увімкненням аварійного режиму за сигналом газоаналізаторів і можливістю ручного запуску.

Рівень шуму на робочих місцях не перевищує 80 децибел, а вібрація відповідає санітарним нормам.

Вміст шкідливих речовин у повітрі не виходить за межі гранично допустимих концентрацій.

Виробничі приміщення забезпечуються природним і штучним освітленням: 300–500 люкс для загальних робіт і 750–1000 люкс для точних операцій, уникаючи тіней. Викиди шкідливих речовин в атмосферу та водовідведення контролюються, а відходи пластмас переробляються в окремих приміщеннях із звукоізоляційними кожухами та пилозахисними фільтрами.

Безпека праці починається з підготовки персоналу. Працівники допускаються до роботи після медичного огляду, вступного та первинного інструктажів і стажування під наглядом досвідченого спеціаліста. Повторні інструктажі проводяться раз на пів року, а позапланові – при змінах у технології, переведенні на іншу ділянку чи введенні нового обладнання. Усі інструктажі записуються в журнали, а працівники знають інструкції для своїх посад і порядок дій в аварійних ситуаціях.

Згідно з документом «Правил охорони праці на підприємствах з виробництва пластмасових виробів», необхідно забезпечити працівників засобами індивідуального захисту.

На екструзійних, калібрувальних і сортувальних ділянках видається вогнестійкий спецодяг із бавовни чи спеціальних полімерних тканин, захисне взуття з антиковзкою підошвою та металевим носком, діелектричні рукавиці для роботи з електрообладнанням, порізостійкі рукавиці для операцій із ріжучими механізмами, захисні окуляри чи щитки для захисту від бризок розплаву, навушники чи вушні вкладиші для роботи в зонах із шумом вище 80 децибел, а також респіратори з фільтрами проти органічних випарів і пилу.

Обладнання забезпечується заземленням, а працівники, які обслуговують електроустановки, отримують діелектричні килимки.

Виробничі приміщення оснащуються порошковими чи вуглекислотними вогнегасниками, аптечками, пунктами медичної допомоги, схемами евакуації та інструкціями з пожежної безпеки.

Працівники знають розташування аварійних виходів і порядок евакуації. Зберігання сировини та готової продукції не перевищує добової потреби, а порошкоподібні матеріали транспортуються в закритій тарі, щоб уникнути розпилення.

Технічне обслуговування є ключовим для безпеки. Обладнання регулярно перевіряється на справність електропроводки, заземлення, пневмоліній, гальмівних і привідних систем. Особлива увага приділяється ріжучим механізмам, транспортерам і сортувальним приводам, які швидко зношуються. Обслуговування проводиться за графіком, затвердженим головним інженером, із записами в документації.

Згідно з документом «Правил охорони праці на підприємствах з виробництва пластмасових виробів», необхідно впроваджувати оцінку професійних ризиків. Для кожної ділянки визначаються джерела небезпеки, імовірність їх виникнення та наслідки. На основі цього аналізу розробляються заходи для запобігання аварій ще на етапі планування модернізації чи змін у виробництві.

Ефективна охорона праці створює безпечне та контрольоване середовище. Згідно з документом «Правил охорони праці на підприємствах з виробництва пластмасових виробів», необхідно інтегрувати заходи безпеки на всіх етапах – від проектування до експлуатації та модернізації. Це забезпечує відповідність законодавству, стабільність підприємства, якість продукції та захист персоналу.

ВИСНОВКИ

У рамках виконаної роботи розроблено модуль автоматизованого керування сортуванням поліетиленових труб, який оптимізує логістичні процеси на виробництві, мінімізує ручні операції та підвищує точність і ефективність. Робота включала аналіз предметної області, проектування апаратно-програмної архітектури, розробку програмного забезпечення на Python, створення бази даних SQLite, тестування в симуляційному середовищі, оцінку стійкості системи та аналіз вимог до охорони праці. Використання Python забезпечило швидке створення симуляційного середовища та гнучкість для потенційного промислового впровадження завдяки його універсальності та підтримці інтеграцій, таких як OPC UA.

Проведено аналіз предметної області, визначено особливості сортувально-логістичних процесів і вимоги до модуля. Порівняння з аналогами (JR Automation, Xuzhou DKEC) показало переваги розробленого модуля в автоматизації браку, інтеграції з MES через OPC UA та гнучкості маршрутизації.

Розроблено архітектуру модуля на базі PLC Siemens S7-1200, конвеєрів, сканерів штрих-кодів і систем контролю якості. Алгоритми сортування, реалізовані в Python, забезпечують пріоритизацію замовлень і обробку браку, а структурно-функціональна схема відображає логіку роботи.

Створено базу даних SQLite для управління замовленнями, трубами та подіями. SQLite виявилася оптимальною для симуляції, а Python — ідеальним інструментом для її інтеграції.

Розроблено вебресурс на Flask для симуляції введення замовлень і тестування логіки сортування. Модуль забезпечує маршрутизацію труб і контроль зон, використовуючи бібліотеки Flask, SQLite і Tabulate.

Проведено тестування, яке підтвердило коректну обробку 345 труб (260 водопровідних, 85 газових) із пріоритизацією за дедлайнами, автоматичним

перенаправленням 5 бракованих труб до зони Defect і обробкою переповнення StorageVT. Python забезпечив швидке налаштування та аналіз результатів.

Оцінено стійкість контуру управління методами ТАУ, підтвердивши стійкість (полюс $s = -22$), нульове перерегулювання та швидкий час регулювання (0,18 с) за допомогою моделювання в Python із бібліотекою control.

Визначено заходи охорони праці, включаючи захисне обладнання, вентиляцію, інструктажі та аварійні зупинки, що забезпечують безпеку на виробництві.

Робота виконана відповідно до вимог законодавства України та внутрішніх положень ХНУРЕ щодо організації освітнього процесу та забезпечення академічної доброчесності [20-23].

Робота відповідає цілям сталого розвитку, визначеним ООН, зокрема Цілі 9 «Індустріалізація, інновації та інфраструктура» та Цілі 12 «Відповідальне споживання і виробництво».

Результати роботи демонструють, що модуль відповідає сучасним вимогам до автоматизації, забезпечуючи точне сортування, обробку браку та контроль зон. Python виявився оптимальним для симуляції та перспективним для промислового застосування з інтеграцією через OPC UA. Модуль готовий до впровадження після оптимізації, наприклад, із використанням PostgreSQL для масштабування, і сприяє підвищенню продуктивності та безпеки на виробництві.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Автоматизація внутрішньо-складських виробничих логістичних процесів для впровадження концепції industry 4.0: енергоощадливість, продуктивність, мобільність, модульність, автономність. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:4OULZ7Gr8RgC
2. Удосконалення автоматизованих логістичних процесів на логістичних об'єктах. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:NaGl4SEjCO4C
3. Використання нейронних мереж для задач розпізнавання технологічних об'єктів в комп'ютерно-інтегрованому виробництві. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&cstart=20&pagesize=80&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:GnPB-g6toBAC
4. Розробка кінематичної схеми транспортувального шатлу для внутрішньоскладської виробничої логістики. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:fPk4N6BV_jEC
5. Промислові інтерфейси та протоколи передачі даних інтегрованих систем для автоматизованого управління в умовах Industry 4.0. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&cstart=20&pagesize=80&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:isC4tDSrTZIC
6. Топ 7 трендів у логістиці 2023 року | Haski.ua. Режим доступу: <https://haski.ua/blog/logistic/logistychni-trendy-2023-roku>
7. Комплекс навчально-методичного забезпечення навчальної дисципліни «Безпека праці в індустрії IT-технологій» підготовки освітнього рівня бакалавр усіх

спеціальностей та усіх напрямів університету [<http://catalogue.nure.ua/knmz>] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 122 с.

8. Про затвердження Правил охорони праці на підприємствах з виробництва пластмасових виробів. Режим доступу: <https://zakon.rada.gov.ua/laws/show/z1750-13#n15>

9. Jr automation & advanced drainage systems: гнучка система сортування труб та палетування. Режим доступу: https://www.automate.org/case-studies/jr-automation-and-advanced-drainage-systems-flexible-pipe-sorting-and-palletizing-system?utm_source

10. Xuzhou DKEC Electrical Technology Co Ltd: Система прийому, сортування та підрахунку матеріалу пластикових труб. Режим доступу: <https://patents.google.com/patent/CN104229223A/en>

11. Аналіз систем координації управління підсистемами підприємства. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:pyW8ca7W8N0C

12. Удосконалення та оптимізація автоматизованих логістичних процесів у логістичних приміщеннях. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&cstart=20&pagesize=80&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:HoB7MX3m0LUC

13. Архітектура програмного забезпечення для супроводження автоматизованих систем оповіщення на виробництві. Режим доступу: https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&cstart=20&pagesize=80&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:YFjsv_pBGBYC

14. Математичне моделювання мехатронних човників як об'єктів автоматизації багаторівневих систем внутрішньоскладської логістики. Режим доступу:

https://scholar.google.com.ua/citations?view_op=view_citation&hl=uk&user=aT-GufsAAAAJ&sortby=pubdate&citation_for_view=aT-GufsAAAAJ:yD5IFk8b50cC

15. Лабораторія Автоматизованих логістичних систем | Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки. Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки | Сайт кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки ХНУРЕ. URL: <https://tapr.nure.ua/dijalnist-kafedri/materialno-tehnichne-zabezpechennja/laboratorija-avtomatizovanih-logistichnih-sistem>

16. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології освітньої [Електронний ресурс] / упоряд.: І. Ш. Невлюдов, О. І. Филипченко, О. В. Токарева, С. П. Новоселов, О.В Сичова ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – електрон. вид. – Харків : ХНУРЕ, 2023. – 760 Кб.

17. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник / І. Ш. Невлюдов, О. Филипченко, О. Токарева, С. Новоселов, О. Сичова. Харків : Видавництво Іванченка І. С., 2023. – 151 с.

18. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.

19. Теорія автоматичного управління (збірник задач): навчальний посібник / І. Ш. Невлюдов, О. В. Токарева. Харків: ХНУРЕ, 2020. 240 с. 35.

20. Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки. Режим доступу: <https://tapr.nure.ua/>

21. Закон України “Про вищу освіту” від 01.07.2014 № 1556-VII. Режим доступу: <https://zakon.rada.gov.ua/laws/show/1556-18#Text>

22. Положення про організацію освітнього процесу у ХНУРЕ [електронний ресурс] : Наказ ХНУРЕ від 05.02.2025 р. No 47. Режим доступу: https://nure.ua/wp-content/uploads/2025/47_05.02.2025.pdf

23. Положення про протидію академічному плагіату в ХНУРЕ [Електронний ресурс]: Наказ ХНУРЕ від 31.12.2024 р. No 386. Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/386_31.12.2024.pdf