

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ
ІНДУСТРІЇ 4.0
SOFTWARE FOR IDENTIFICATION OF INDUSTRY 4.0 OBJECTS

Виконала: студент(ка) 4 курсу, групи КНТ-111сп
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)
забезпечення

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

ШИМБОРЕЦЬКА А.О

(ПРИЗВИЩЕ та ініціали)

Керівник ПАРХОМЕНКО А.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет комп'ютерних наук та технологій
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ШИМБОРЕЦЬКОЇ Антоніни Олександрівни
 (ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для ідентифікації об'єктів
Індустрії 4.0. Software for Identification of Industry 4.0 Objects

керівник проєкту (роботи) к.т.н., доцент, ПАРХОМЕНКО Анжеліка
Володимирівна

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Методи та засоби ідентифікації об'єктів. 3. Проєктування програмно-апаратного забезпечення. 4. Конструювання програмно-апаратного забезпечення. 5. Інструкція з використання програмного забезпечення.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ПАРХОМЕНКО А.В., доцент		
Нормоконтроль	БЄЛОВА А.В., асистент		

7. Дата видачі завдання «15» квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання
2	Аналіз предметної області.	1 тиждень	Розділи 1,2
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 3
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Розділ 5, Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Антоніна ШИМБОРЕЦЬКА
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Анжеліка ПАРХОМЕНКО
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 91 с., 12 табл., 35 рис., 2 дод., 58 джерел.

ІНДУСТРІЯ 4.0, ПРОМИСЛОВИЙ ІНТЕРНЕТ РЕЧЕЙ,
ІДЕНТИФІКАЦІЯ ОБ'ЄКТІВ, RFID.

Об'єкт дослідження – процес ідентифікації об'єктів.

Предмет дослідження – методи та засоби ідентифікації об'єктів Індустрії 4.0.

Мета роботи – дослідження та розробка програмно-апаратних рішень для взаємодії з RFID-мітками та ідентифікації об'єктів Індустрії 4.0.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, зчитувач радіочастотної ідентифікації FONKAN FM-505, мова програмування C#, Microsoft Visual Studio 2022, персональний комп'ютер з процесором AMD Ryzen 5 5600H під управлінням операційної системи Microsoft Windows 10.

Результати. Створено програмний Windows-застосунок для взаємодії з RFID-мітками на основі використання зчитувача радіочастотної ідентифікації FONKAN FM-505.

Висновки. Розроблено програмну систему ідентифікації об'єктів для спрощення їх обліку.

Галузь використання – магазини та промислові склади.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 91 pages, 12 tables, 35 figures, 2 appendixes, 58 sources.

INDUSTRY 4.0, INDUSTRIAL INTERNET OF THINGS, OBJECT IDENTIFICATION, RFID.

Object of study is process of objects' identification.

Subject of study are methods and tools of identification of Industry 4.0 objects.

The purpose of the work is research and development of hardware-software solutions for interaction with RFID tags and identification of Industry 4.0 objects.

Materials, methods, and tools: structured and object-oriented programming, FONKAN FM-505 RFID reader, C# programming language, Microsoft Visual Studio 2022, personal computer with AMD Ryzen 5 5600H processor running Microsoft Windows 10.

Results. The Windows software application is created which interacts with RFID tags based on the use of the FONKAN FM-505 RFID reader.

Conclusions. A software system has been developed for objects identification to simplify their tracking.

The field of use are stores and industrial warehouses.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Індустрія 4.0.....	10
1.2 Промисловий Інтернет речей	16
1.3 Постановка задач дипломної роботи	20
2 Методи та засоби ідентифікації об'єктів	21
2.1 Автоматична ідентифікація та збір даних	21
2.2 RFID.....	22
2.3 NFC	30
2.4 Штрих-код.....	33
2.5 QR-код.....	37
2.6 Оптичне розпізнавання символів	41
2.7 Біометричні дані.....	46
2.8 Висновки до розділу	49
3 Проектування програмно-апаратного забезпечення	50
3.1 Аналіз вимог до системи	50
3.2 Розробка архітектури системи	50
3.3 Вибір та обґрунтування апаратних засобів розробки.....	51
3.4 Вибір та обґрунтування вибору мови програмування	52
3.5 Вибір та обґрунтування вибору середовища розробки.....	53
3.6 Висновки до розділу	54
4 Конструювання програмно-апаратного забезпечення	55
4.1 Розробка алгоритму функціонування системи	55
4.2 Розробка БД	56
4.3 Опис особливостей програмної реалізації.....	57
4.4 Опис реалізації інтерфейсу	61

4.5 Висновки до розділу	62
5 Інструкція з використання програмного забезпечення	63
5.1 Підключення застосунку	63
5.2 Практичне застосування застосунку	64
Висновки	68
Перелік джерел посилань	69
Додаток А Текст програми.....	75
Додаток Б Слайди презентації	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ARAT	– Active Reader Active Tag;
ARPT	– Active Reader Passive Tag;
CMOS	– Complementary Metal–Oxide–Semiconductor;
CTLS	– Contactless;
ERP	– Enterprise Resource Planning;
HF	– High Frequency;
ICR	– Intelligent Character Recognition;
IIoT	– Industrial Internet of Things;
LF	– Low Frequency;
LTE	– Long Term Evolution;
NFC	– Near Field Communication;
OCR	– Optical Character Recognition;
OWL	– Web Ontology Language;
PRAT	– Passive Reader Active Tag;
RFID	– Radio Frequency IDentification;
RDF	– Resource Description Framework;
RO	– Read Only;
RW	– Read and Write;
SHF	– Super High Frequency;
UPC	– Universal Product Code;
UHF	– Ultra High Frequency;
VHF	– Very high frequency;
WORM	– Write Once Read Many;
QR	– Quick Response;
АІЗД	– автоматична ідентифікація та захоплення даних;
БД	– база даних.

ВСТУП

Сьогодні більшість країн світу розвивають промисловість, орієнтуючись на тенденції новітньої індустріальної епохи, яка передбачає перехід до повністю автоматизованого цифрового виробництва, керованого інтелектуальними системами в режимі онлайн. Це виробництво підтримує постійний зв'язок із зовнішнім середовищем, виходячи за межі окремого підприємства, з перспективою об'єднання в глобальну промислову мережу товарів і послуг. Актуальність активного розвитку діджиталізації та цифрової економіки у світових процесах підтверджується державними службовцями, представниками бізнесу, освіти і науки [1].

Індустрія 4.0, яка є центральною частиною четвертої промислової революції, включає безліч нанотехнологій. Її головна мета - створення єдиного простору для обміну даними, віртуальної візуалізації процесів, систем та об'єктів. Індустрія 4.0 також передбачає створення роботизованих систем у поєднанні з Інтернет-технологіями, що створюють "розумне" підприємство [2].

Задачею технології Інтернету речей є ідентифікація та з'єднання об'єктів Індустрії 4.0 через мережу Інтернет. Завдяки цьому стає можливим перетворення традиційних виробництв у «розумні» системи, що забезпечують моніторинг виробничих процесів, оптимізацію ланцюжків постачання та нові можливості аналізу даних для прийняття обґрунтованих рішень.

Для ідентифікації об'єктів Індустрії 4.0 використовуються різноманітні засоби автоматичної ідентифікації та захоплення даних, такі як QR та штрих-коди, RFID/NFC мітки, біометрична ідентифікація та ін. Але розробка нових рішень залишається актуальною задачею.

Метою дипломної роботи є дослідження та розробка програмно-апаратних рішень для взаємодії з RFID-мітками та ідентифікації об'єктів Індустрії 4.0.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Індустрія 4.0

У зв'язку зі зростанням кількості обладнання з підключенням до мережі Інтернет, з'явилася можливість збирати величезну кількість даних з обладнання, датчиків, машин і інших елементів виробничого процесу. Ці дані можна аналізувати та використовувати для покращення процесів виробництва, зменшення витрат та підвищення якості продукції [1].

Індустрія 4.0 означає обмін даними між різними учасниками виробничого процесу, такими як фахівці підприємства, системи ERP (Enterprise Resource Planning), роботи, виробництво та інші системи. Продукція, що виробляється підприємством в межах "Індустрії 4.0" буде сама «казати» обладнанню як, де, ким та чим вона може бути виготовлена. Автомати та виробничі лінії будуть автоматично змінювати свою конфігурацію, відповідаючи запитам, які надходять від продукції на конвеєрній стрічці. Компоненти, системи управління, ERP-системи та персонал будуть обмінюватися даними про технологічні процеси, спрощуючи виробничі операції і підвищуючи ефективність [3].

Термін «Індустрія 4.0» використовується для опису еволюції виробництва, яка відбувається протягом всього життєвого циклу продукції. Він охоплює виробничий ланцюг, ланцюг постачання та створення вартості. Індустрія 4.0 зацікавлює багато сторін, включаючи клієнтів, які очікують підвищення продуктивності, що значно впливає на весь ланцюг поставок [4].

Четверта промислова революція включає в себе ряд компонентів, які можна виявити, детально досліджуючи наше суспільство та сучасні цифрові тенденції. До таких компонентів надходять такі цифрові технології як:

- мобільні пристрої;
- технології визначення місцезнаходження;
- аутентифікація та виявлення шахрайства;
- 3D друк;

- розумні сенсори;
- велика аналітика та вдосконалені процеси;
- багаторівневе взаємодія з клієнтами та профілювання клієнтів;
- доступність ресурсів комп'ютерної системи за запитом;
- вдосконалений інтерфейс людина-машина;
- візуалізація даних та реалізоване "живе" навчання [5].

Розвиток нових цифрових промислових технологій, відомих як Індустрія 4.0, є процесом трансформації, що базується на дев'яти ключових технологічних досягненнях (рис.1.1) [6].

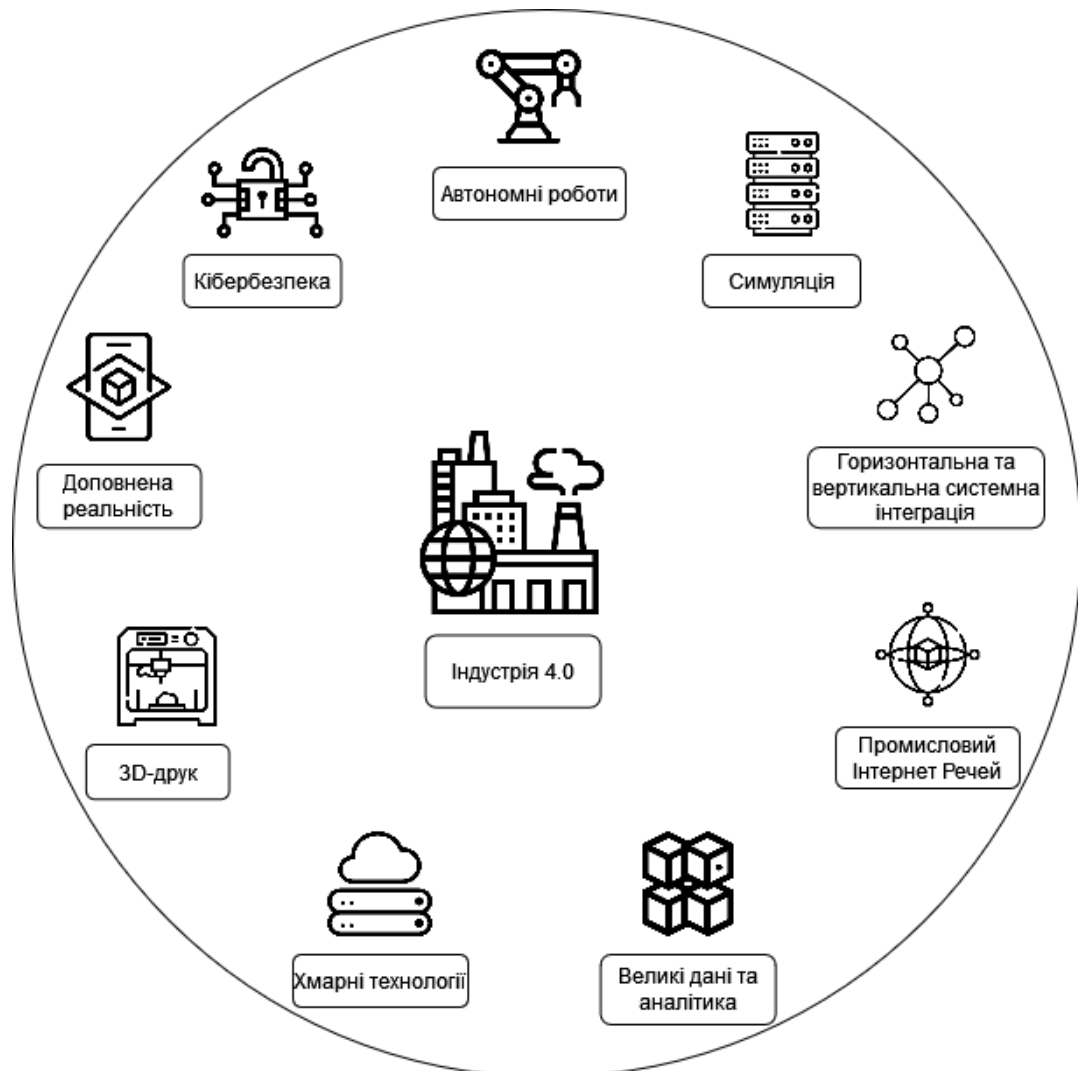


Рисунок 1.1 – Дев'ять технологій, що трансформують промислове виробництво [6]

У цій трансформації з'єднані машини, датчики, ІТ-системи та заготовки будуть інтегровані вздовж ланцюга створення вартості, виходячи за межі окремих підприємств. Ці підключені системи, також відомі як кіберфізичні системи, зможуть взаємодіяти одна з одною за допомогою стандартних Інтернет-протоколів та аналізувати дані для передбачення відмов, налаштування та адаптації до змін. Індустрія 4.0 дозволить збирати та аналізувати дані з різних машин, що дозволить здійснювати швидкі, гнучкі та ефективні процеси для виробництва товарів вищої якості за менші витрати. Це, у свою чергу, збільшить продуктивність виробництва, змінить економіку, сприятиме промислового зростанню та змінить профіль робочої сили, що в кінцевому рахунку змінить конкурентоспроможність компаній і регіонів [6].

Багато з технологічних досягнень, що лежать в основі Індустрії 4.0, вже використовуються у виробництві, але саме Індустрія 4.0 трансформує цей процес: раніше ізольовані й оптимізовані модулі об'єднуються в повністю інтегрований, автоматизований та оптимізований виробничий процес. Це призводить до підвищення ефективності та зміни традиційних виробничих відносин між постачальниками, виробниками та клієнтами, а також між людиною та машиною [6].

Індустрія 4.0 об'єднує широкий спектр нових технологій з метою створення значної вартості. Використовуючи кіберфізичні системи, які контролюють фізичні процеси, можна створити віртуальну модель реального світу. Особливості кіберфізичних систем включають можливість приймати децентралізовані рішення, що дозволяє досягнути високого рівня автономності [5].

Навіть якщо термін "Індустрія 4.0" існує вже багато років, в Україні він мало використовується серед науковців та фахівців. Незважаючи на складну економічну ситуацію в країні, важливо, щоб вітчизняні експерти приділяли увагу цій новій концепції розвитку. В іншому випадку, ігноруючи цей тренд зараз, Україна може назавжди втратити своє місце в світовій економіці [7].

1.1.1 Великі дані та аналітика

Аналітика, заснована на великих обсягах даних, недавно з'явилася у виробництві, де вона оптимізує якість, зменшує споживання енергії та поліпшує обслуговування обладнання. У рамках концепції "Індустрія 4.0" збір та комплексна оцінка даних з різних джерел - виробничого обладнання, систем управління підприємством та взаємодії з клієнтами - стануть стандартом для прийняття рішень в реальному часі [6].

Наприклад, виробник напівпровідників Infineon Technologies зменшив кількість дефектних виробів, порівнюючи дані процесора, отримані під час тестування наприкінці виробничого циклу, з даними, зібраними процесом стану пластини на початку процесу. Таким чином, Infineon може ідентифікувати закономірності, які допомагають виявляти несправні чіпи на ранніх етапах виробничого процесу та покращувати якість продукції [6].

1.1.2 Автономні роботи

Виробники у багатьох галузях давно користуються роботами для вирішення складних завдань, проте роботи продовжують розвиватися, набуваючи ще більшої значущості. Вони стають більш автономними, гнучкими та співпрацюють один з одним. У майбутньому вони будуть взаємодіяти з людьми, працюючи поруч з ними та навчаючись від них. Ці роботи будуть менш дорогими та матимуть більший спектр можливостей, ніж ті, що використовуються сьогодні у виробництві [6].

1.1.3 Симуляція

На стадії проєктування вже використовується тривимірне моделювання продуктів, матеріалів і виробничих процесів. Однак у майбутньому симуляції будуть використовуватися ще ширше, в процесі

роботи заводу. Ці симуляції використовуватимуть дані в реальному часі для відображення фізичного світу у віртуальній моделі, що може включати машини, продукти та людей. Це дозволить операторам тестувати та оптимізувати налаштування машини для наступного продукту в черзі у віртуальному світі перед фізичним впровадженням, скорочуючи час налаштування машини та підвищуючи якість продукції [6].

1.1.4 Горизонтальна та вертикальна системна інтеграція

Більшість існуючих ІТ-систем не мають повної інтеграції. Зв'язки між компаніями, постачальниками та клієнтами, а також між різними відділами, такими як, наприклад, інженерний, виробничий та сервісний, є рідкісними. Навіть в межах самої компанії функції від рівня підприємства до рівня цеху не інтегровані належним чином. Не відбувається повна інтеграція навіть на етапі розробки, від створення продуктів до управління обладнанням та автоматизації. З Індустрією 4.0 ця ситуація зміниться: компанії, відділи, функції та можливості стануть більш інтегрованими, оскільки мережі інтеграції даних між компаніями будуть розвиватися, створюючи повністю автоматизовані ланцюги виробництва [6].

1.1.5 Кібербезпека

Багато компаній залишаються залежними від систем управління та виробництва, які часто не пов'язані між собою або відкриті для взаємодії. З впровадженням Індустрії 4.0 та розширеним використанням стандартних протоколів зв'язку зростає необхідність у захисті критичних промислових систем і виробничих ліній від кіберзагроз. Тому безпечний і надійний зв'язок, а також ефективне управління ідентифікацією та доступом для машин і користувачів стають надзвичайно важливими [6].

1.1.6 Хмарні технології

Компанії уже використовують хмарні програмні засоби для деяких корпоративних та аналітичних застосунків, але з Індустрією 4.0 збільшиться потреба в обміні даними між виробничими підрозділами та між компаніями. В той же час, продуктивність хмарних технологій покращиться, що дозволить зменшити час реакції до кількох мілісекунд. Це призведе до того, що все більше машинних даних та функцій буде використано в хмарному середовищі, надаючи більше послуг, управління даними для виробничих систем. Навіть системи, що відслідковують та контролюють процеси, можуть переходити до хмарного середовища [6].

1.1.7 3D-друк

Компанії тільки почали впроваджувати адитивне виробництво, таке як 3D-друк, яке переважно використовується для прототипування та виробництва окремих компонентів. З Індустрією 4.0 ці методи адитивного виробництва будуть широко використовуватися для виробництва невеликих партій індивідуальних продуктів, які мають конструктивні переваги, наприклад, складні та легкі конструкції. Високоєфективні децентралізовані системи адитивного виробництва допоможуть скоротити транспортні відстані та запаси на руках [6].

1.1.8 Доповнена реальність

Системи на основі доповненої реальності надають різноманітні послуги, такі як вибір деталей на складі та надсилання інструкцій з ремонту через мобільні пристрої. Наразі ці системи перебувають у зародковому стані, але в майбутньому компанії значно ширше використовуватимуть доповнену реальність, щоб надавати працівникам інформацію в режимі реального часу

для поліпшення процесу прийняття рішень та роботи. Наприклад, працівники можуть отримати інструкції з ремонту щодо того, як замінити певну частину, коли вони розглядають фактичну систему, яка потребує ремонту. Ця інформація може відображатися безпосередньо в полі зору працівників за допомогою таких пристроїв, як окуляри доповненої реальності [6].

1.2 Промисловий Інтернет речей

Застосування Інтернету речей у виробництві створює ряд можливостей для його розвитку та вирішує ряд важливих задач: оптимізація виробничого процесу, зниження матеріальних та енергетичних витрат, підвищення якості, зростання рентабельності виробництва та конкурентоспроможності на світовому ринку [8].

Промисловий Інтернет речей (Industrial Internet of Things, IIoT) — це система взаємопов'язаних датчиків, приладів та інших пристроїв, об'єднаних у мережу разом із об'єктами промисловості. Такий зв'язок дозволяє створювати виробництва, які виявляються більш ощадливими, гнучкими і ефективними [8].

Промисловий Інтернет речей дозволяє виробництвам використовувати нові гнучкі бездротові технології для пришвидшення обміну інформацією у реальному часі без участі людини. Завдяки розширенню і доповненню використання на робочих зонах бездротових мереж можна суттєво розширити зони застосування моніторингу та управління. Чим більше отримується даних о виробництві, тим більш досконалий аналіз ефективності можна зробити [3].

Підвищення застосування IIoT-рішень у виробництві дозволяє створювати автоматизовані виробництва, забезпечити автономну взаємодію між машинами та підвищити ефективність виробництва в декілька разів завдяки використанню різноманітних технологій. Термін окупності переходу до такого виробництва не перевищую декількох місяців [9].

Промисловий Інтернет речей використовує нові технології для вирішення довгострокових проблем у виробництві, таких як управління активами в хмарі, яке раніше здійснювалось локально. Основна ідея полягає в зборі та аналізі всієї доступної інформації про процеси та активи з метою забезпечення оператора реальними даними для прийняття необхідних рішень [10].

ІоТ - це мережа інтелектуальних пристроїв, підключених до формування систем, які контролюють, збирають, обмінюються та аналізують дані. Загалом, кожна промислова система ІоТ складається з набору елементів (рис. 1.2) [11].



Рисунок 1.2 – Схема роботи Інтернету речей на промисловому виробництві [11]

1.2.1 Ідентифікація

Ідентифікація, перша вимога в елементах IoT, вважається доказом інформації про ідентичність для кожного об'єкта у світі IoT. Ідентичність зазвичай використовується для визначення конкретної особи, пристрою або сутності. Крім того, вона вважається ключовим фактором у створенні зв'язку або відносин між особами і критичною для успіху системи IoT. Це дозволяє нам ідентифікувати мільярди різноманітних об'єктів та управляти віддаленими об'єктами через Інтернет [11].

Ідентифікація також пов'язує об'єкти з інформацією, пов'язаною з конкретним об'єктом, яка може бути отримана з сервера. Це дозволяє об'єкту взаємодіяти з іншими об'єктами через Інтернет у різних обсягах. Повинен бути спосіб координувати ідентичності всіх об'єктів IoT у межах, щоб дозволити безпечну міжоб'єктну комунікацію. Управління ідентичністю потрібне для трьох основних сторін: користувача, ідентифікаторів об'єктів та відносин, в залежності від певних принципів, які застосовуються до ідентичності об'єктів. Крім того, воно повинно вирішувати унікальні виклики в моделі IoT [11].

1.2.2 Збір даних

Процес збирання інформації з об'єктів. Зібрана інформація надсилається до носіїв зберігання. Існує багато пристроїв для збирання інформації з об'єктів, таких як RFID-мітки, розумні сенсори, пристрої для зчитування даних з одягу та інші [12].

1.2.3 Комунікація

Підключення великої кількості різноманітних об'єктів та обмін інформацією для надання певної послуги є важливою проблемою і вимагає

використання технологій зв'язку. Зв'язок в Інтернеті речей є важливою складовою з'єднання об'єктів, проте зв'язок постійно обмежений характеристиками об'єктів, такими як тривалість роботи батареї або обмежений діапазон передачі даних [12].

Існують різноманітні технології зв'язку, що використовуються для Інтернету речей, такі як Wi-Fi, Bluetooth, RFID, NFC, IEEE 802.15.4, Z-wave, Zigbee та Long Term Evolution (LTE) [12].

1.2.4 Обчислення

Обчислення здійснюється на зібраній інформації з об'єктів за допомогою сенсорів. Воно використовується для видалення зайвої інформації, яка не потрібна. Багато апаратних та програмних платформ розроблено для виконання обробки в застосуваннях Інтернету речей. Для апаратних платформ використовуються Audrino, Raspberry Pi та Intel Galileo, тоді як для програмних платформ операційна система відіграє важливу роль у виконанні обробки. Існують багато типів операційних систем, які використовуються, такі як Tiny OS, Lite OS, Android, тощо [12].

1.2.5 Послуги

Існує чотири типи послуг, які надають додатки Інтернету речей. Перша з них - це послуга, пов'язана з ідентифікацією. Вона використовується для отримання ідентифікаторів об'єктів, щодо яких надіслали запит. Ще одна послуга - це послуга збирання інформації, мета якої полягає в зборі всієї інформації з об'єктів. Обробка також виконується службою збору інформації. Третя послуга - це спільна послуга, яка приймає рішення згідно зі збіраною інформацією та надсилає відповідні відповіді пристроям. Остання послуга - це послуга загального користування, яка використовується для негайного відповіді пристроям без жорстких вимог з точки зору часу і місця [12].

1.2.6 Семантика

Семантика вважається мозком Інтернету речей. Вона використовується для розумного витягування знань з пристроїв для прийняття відповідних рішень з метою надання потрібних послуг. Вона виявляє та використовує ресурси та інформацію про моделювання. Фреймворк опису ресурсів (Resource Description Framework, RDF) та мова веб-онтологій (Web Ontology Language, OWL) є прикладами технологій семантичного вебу [11].

1.3 Постановка задач дипломної роботи

У даному розділі був наведений аналіз предметної області, детально розглянуто поняття Індустрії 4.0, IIOT, його структура та основні компоненти.

Показано, що створення застосунку для ідентифікації об'єктів у приміщеннях та збору даних про їх надходження та відвантаження для їх подальшого обліку є актуальною задачею.

Мета роботи – дослідження та розробка програмно-апаратних рішень для взаємодії з RFID-мітками та ідентифікації об'єктів Індустрії 4.0.

Для виконання цієї мети у роботі бакалавра необхідно виконати такі задачі:

- провести дослідження доступних методів та засобів ідентифікації об'єктів;
- здійснити проєктування програмного забезпечення;
- створити програмне забезпечення для ідентифікації об'єктів;
- виконати тестування розробленого програмного забезпечення для перевірки його ефективності.

2 МЕТОДИ ТА ЗАСОБИ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ

2.1 Автоматична ідентифікація та збір даних

Автоматична ідентифікація та збір даних (АІЗД), є методом автоматичної ідентифікації та збору даних про об'єкти, їх безпосередньої передачі до комп'ютерних систем без участі людини. АІЗД ще можуть називати «автоматичною ідентифікацією», або «автоматичним захопленням даних» [13].

До АІЗД зазвичай відносять такі технології як: QR та штрих-коди, RFID/NFC мітки, біометрична ідентифікація (сканування відбитків пальців, розпізнавання голосу, зображення обличчя і т. д.), оптичне розпізнавання символів (OCR), штрих-коди, QR-коди, та ін. (рис. 2.1) [14].



Рисунок 2.1 – Технології що входять до АІЗД [14]

АІЗД — це процес або засіб отримання зовнішніх даних, зокрема, шляхом аналізу зображень, звуків чи відеозаписів. Для збору даних використовується перетворювач, який перетворює фактичне зображення або звук у цифровий файл. Потім файл зберігається, а згодом його можна аналізувати комп'ютером або порівнювати з іншими файлами в базі даних для перевірки ідентичності або надання дозволу на введення захищеної системи. Захоплення даних може здійснюватися різними способами; найкращий метод залежить від застосування [13].

Системи біометричної безпеки використовують фізіологічні, або психологічні характеристики людини для її ідентифікації. Вони завжди відрізняються, тому що у двох різних людей не буває ідентичних рис [15].

Існує багато різноманітних методів біометричної ідентифікації, зазвичай їх розділяють на дві великі групи: статистичні та динамічні. Статистичні методи ґрунтуються на унікальній природжений фізіологічній характеристиці людини, яка є її невід'ємною частиною. Динамічні ґрунтуються на поведінковій (динамічній) характеристиці людини, тобто підсвідомі рухи у процесі відтворення якої-небудь дії, наприклад почерк, голос, манера ходи [16].

Технологія радіочастотної ідентифікації була розроблена в 1980-х роках, що робить її відносно молодого. Вона є основою в сучасних безконтактних інформаційних системах, також широко використовується в автоматизованих системах збору, ідентифікації та аналізу даних. RFID-системи застосовуються в різноманітних випадках, коли потрібно швидко і точно відстежувати та ідентифікувати чисельні рухомі об'єкти. Ця АІЗД технологія може бути ефективна у тих виробничих середовищах, де інші системи не можуть бути використані [17].

2.2 RFID

Radio Frequency IDentification (RFID) — спосіб автоматичної ідентифікації об'єктів, у якому за допомогою радіосигналів зчитуються або записуються дані, що зберігаються в так званих транспондерах, або RFID-мітках [17].

RFID-мітки складаються з трьох частин: мікросіпа (інтегральна схема, яка зберігає і обробляє інформацію, а також модулює і демодулює радіочастотні сигнали), антени для прийому та передачі сигналу та підкладки (рис. 2.2). Інформація про мітки зберігається в енергонезалежній пам'яті.

Мітка RFID містить фіксовану або програмовану логіку для обробки даних передачі та датчиків відповідно [18].

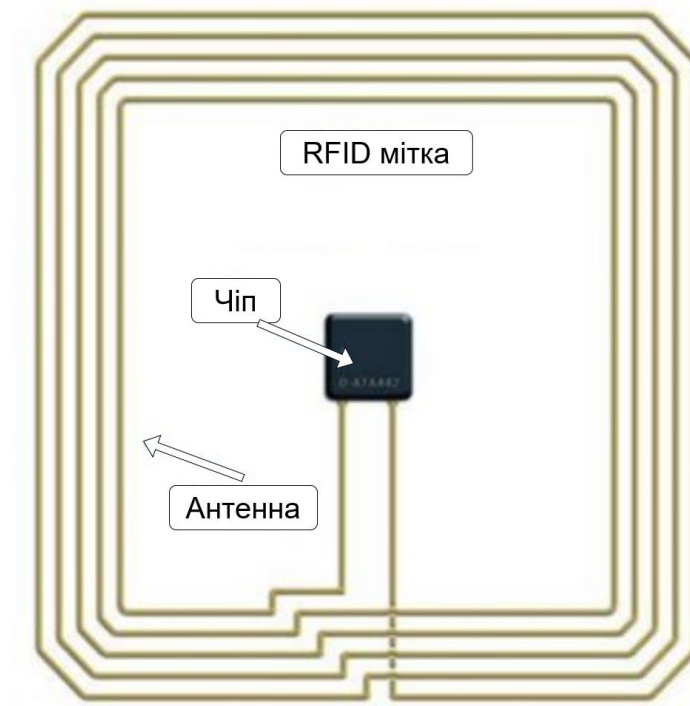


Рисунок 2.2 – Конструкція RFID міток [20]

RFID-мітки можуть бути пасивними, активними або пасивними з акумулятором [17].

Активні RFID-мітки мають власне джерело живлення і не потребують енергії від зчитувача, тому вони можуть бути зчитані на великій відстані. Такі мітки зазвичай мають великі розміри і можуть бути додатково оснащені електронікою. Проте, вони вважаються більш дорогими, і мають обмежений час роботи, якщо працюють через батарею. У більшості випадків активні мітки виявляються надійнішими, оскільки мають спеціальну сесію зв'язку з пристроєм зчитування, що зменшує ймовірність помилок [19].

Активні RFID-мітки, які мають своє джерело живлення, можуть генерувати сильніший вихідний сигнал, ніж пасивні мітки. Це дозволяє їх застосовувати в умовах, де середовище для радіочастотного сигналу може бути агресивним, таких як водне середовище (включаючи людей і тварин, які

переважно складаються з води), металеві поверхні (наприклад, корабельні контейнери, автомобілі), або для великих відстаней у повітрі. Більшість активних міток можуть передавати сигнал на відстань у сотні метрів із тривалістю життя батареї до 10 років. Деякі RFID-мітки також мають вбудовані сенсори, наприклад, для відслідковування температури продуктів, які можуть швидко псуватися. Інші типи сенсорів, спільно з активними мітками, можуть використовуватися для вимірювання вологості, реєстрації поштовхів/вібрації, вимірювання рівня світла, радіації, температури і концентрації газів у повітрі (наприклад, етилену) [19].

Пасивна мітка, що працює від акумулятора, має невеликий акумулятор і активується при наявності зчитувача RFID. Пасивна мітка дешевша і менша, оскільки в ній немає акумулятора, замість цього вона використовує електричний струм, що індукується в антені електромагнітним сигналом від зчитувача, що забезпечує достатню потужність для функціонування кремнієвого CMOS-чипа, розміщеного в мітці, і передачі сигналу у відповідь [19].

За типом використовуваної пам'яті RFID-мітки діляться на:

- RO (Read Only) дані записуються тільки один раз, відразу при виготовленні. Такі мітки придатні тільки для ідентифікації. Ніяку нову інформацію в них записати не можна;
- WORM (Write Once Read Many) окрім унікального ідентифікатора такі мітки містять блок одноразово записуваної пам'яті, яку надалі можна багато разів зчитувати;
- RW (Read and Write) такі мітки містять ідентифікатор і блок пам'яті для читання/запису інформації. Дані в них можуть бути перезаписані багаторазово [21].

Системи RFID можна класифікувати за типом мітки та зчитувача. Є 3 типи:

- система пасивного зчитування активних міток (Passive Reader Active Tag, або PRAT) має пасивний зчитувач, який приймає радіосигнали

лише від активних міток (працює від батареї, лише передає). Дальність прийому зчитувача системи PRAT можна регулювати від 0 до 600м, що забезпечує гнучкість у таких програмах, як захист активів і нагляд;

- система активного зчитування пасивних міток (Active Reader Passive Tag, або ARPT) має активний зчитувач, який передає сигнали запитувача, а також отримує відповіді аутентифікації від пасивних міток;

- система активного зчитування активних міток (Active Reader Active Tag, або ARAT) використовує активні мітки, активовані сигналом запитувача від активного зчитувача. Варіант цієї системи також може використовувати пасивну мітку з акумулятором [22].

Також зчитувані ділять на стаціонарні та мобільні [19].

Стаціонарні сенсори можуть бути зафіксовані нерухомо на стінах, дверях або рухомих пристроях у складах, таких як штабелятори та навантажувачі. Їх можна встановлювати у вигляді замків, вбудовувати у стільницю або закріплювати поруч з конвеєром, що проходить продукцію [20].

Порівняно з мобільними пристроями, стаціонарні зчитувачі зазвичай мають більшу зону читання та потужність, що дозволяє їм одночасно обробляти значний потік інформації. На виробництві стаціонарні зчитувачі інтегруються в інформаційну систему, що дозволяє поетапно фіксувати переміщення маркованих об'єктів в реальному часі або ідентифікувати положення маркованих предметів у просторі [20].

Мобільні зчитувачі мають більш обмежену дальність дії і працюють без постійного зв'язку з програмою контролю та обліку. Вони зазвичай обладнані внутрішньою пам'яттю, в яку можуть записуватися дані, прочитані з міток (пізніше цю інформацію можна синхронізувати з системою обліку), і також, так само як і стаціонарні зчитувачі, мають можливість записувати дані на мітку, наприклад, інформацію про проведений контроль [20].

RFID використовує радіочастотні (RF) сигнали для ідентифікації та відстеження фізичних об'єктів. Радіочастоти - це частоти в

електромагнітному спектрі, пов'язані з поширенням радіохвиль. Коли струм подається на антену, створюється електромагнітне поле, яке потім може поширюватися в просторі. Багато бездротових технологій ґрунтуються на поширенні електромагнітного поля. В табл. 2.1 перераховані радіочастоти, які використовуються в поточних стандартах індустрії RFID [21].

Таблиця 2.1 – Діапазони радіочастот RFID та їх застосування [21]

Діапазони	Переваги	Недоліки	Застосування
Низькі 125-134,2 кГц 140-148,5 кГц	Працює біля металу Наразі використовується скрізь Всесвітньо прийнято	Дальність зчитування <1,5 метрів Непрактично для використання на складах Немає в стандартах ЕРС	ID для тварин Бочки для пива Ключі для авто Бібліотечні книжки
Високі 13,56 МГц	Працює в більшості середовищ Наразі використовується скрізь Всесвітньо прийнято	Дальність зчитування <1,5 метрів Не працює біля металу	Відстеження рівня предметів Багаж авіакомпаній Відстеження вантажівок і причепів
Надвисокі 862-928 МГц 950 МГц 865-867 МГц	Дальність зчитування >1,5 метрів Зростаюче комерційне використання	Не працює у вологому середовищі Розстроювання сигналу при безпосередній фізичній близькості	Відстеження ящиків, піддонів і контейнерів Відстеження вантажівок і причепів
Мікрохвильове 2,45 ГГц 5,8 ГГц	Дальність зчитування >1,5 метрів	Немає ліцензії на комерційне використання в деяких країнах ЄС	Контроль доступу для транспортних засобів

Частоти RFID знаходяться у діапазонах низьких частот (LF, Low Frequency), високих частот (HF, High Frequency), дуже високих частот (VHF, Very high Frequency), ультрависоких частот (UHF, Ultra High Frequency) та надвисоких частот (SHF, Super High Frequency) радіочастотного спектру. Мітки низької частоти 125–134,2 кГц та 140–148,5 кГц, а також мітки високої частоти 13,56 МГц можуть використовуватися глобально без ліцензії. Мітки високої частоти 868 МГц-928 МГц не можуть бути використані глобально без ліцензії, оскільки не існує ще одного глобального стандарту для UHF RFID.

Слід зауважити, що у різних країнах різні діапазони частот мають різне призначення [21].

Крім частотних діапазонів та рівня потужності, мітки RFID виготовляються з різних матеріалів для відповідності різним цілям. Мітки RFID можуть бути виготовлені з майже будь-якого міцного матеріалу, що здатний утримувати чіп RFID. Це охоплює різні види пластмас, епоксидні смоли, метали, кераміку, тефлон та власні матеріали. Кожне застосування RFID унікальне, і матеріал повинен відповідати його цілі [23].

Існує кілька застосувань, які потребують, щоб мітки RFID були стійкими до впливу тепла. Чи потрібно вони проходити по збірній лінії, в підземний нафтогазовий трубопровід або пройти через пральню, мітка RFID може бути використана з різними стійкими до тепла і навіть вогнезахисними матеріалами [23].

Спеціалізовані мітки RFID можуть бути стійкі до тепла, діапазон їх стійкості варіюється від стандартних (до 230°C), екстремальних (до 300°C) та ультра-високих (до 400°C) температур. Їх можуть виготовляти з різних матеріалів, включаючи нейлон, тефлон, кераміку або FR4, а також власні матеріали. В основному, поки мітка RFID може бути закрита стійким матеріалом і поки читач міток RF може її зчитати, її можна використовувати для розміщення чіпу RFID [23].

Також окрім високих температур, мітки RFID іноді можуть контактувати з рідинами. Це може бути вода, мийні засоби або тілесні рідини, які іноді можуть пошкодити мітки RFID, якщо вони не захищені належним чином. З цієї причини існує різноманіття стійких до рідин типів міток RFID [23].

Додатковою складністю до матеріалу, який використовується для утримання чіпу RFID, є спосіб його прикріплення. Багато клею розкладаються при контакті з водою або хімічними речовинами. Окрім водостійких клеїв, мітки RFID можуть бути пришиті до білизни або розміщені в малих міцних коробках [23].

Нові досягнення в технології RFID зробили впровадження RFID більш доступним для ряду бізнес-застосувань. Це призвело до значного зростання використання RFID протягом останніх кількох років. Наприклад, магазин Prada у Нью-Йорку використовує RFID для відображення відповідних аксесуарів на екранах у магазині. Деякі з поточних бізнес-застосувань RFID включають:

- управління ланцюгом поставок;
- громадський транспорт;
- поштові та кур'єрські послуги;
- харчова промисловість;
- бібліотечні системи;
- охорона здоров'я, фармацевтична промисловість;
- керування доступом;
- боротьба з підробкою;
- стягнення мита;
- обробка багажу авіакомпанією;
- ідентифікація тварин;
- відстеження документів [21].

Деякі цікаві можливі майбутні застосування технології RFID включають:

- полиці холодильника, які повідомляють, коли молоко або інші продукти з RFID-мітками закінчуються строк придатності;
- авіаквитки з RFID-мітками, які допомагають направляти пасажирів в аеропорту;
- мікрохвильові печі, які читають інструкції з приготування з RFID-міток на упаковках їжі;
- кухні, які рекомендують рецепти готування на основі продуктів з RFID-мітками, які є в наявності;
- RFID-мітки на поштових марках;

– RFID-мітки на рахунках-фактурах, купонах та чеках [22].

RFID-мітки можна розділити за різними типами інкапсуляції тегів: наклейки, пластикові картки, скляні трубки, керамічні, браслети та інші, що продемонстровані на рисунку 2.3 [24].



Рисунок 2.3 – Приклади інкапсуляції RFID міток [25]

RFID-наклейки часто складаються з покривного матеріалу, вкладки, клейового шару та паперу для зняття. Завдяки наявності клейового шару, їх можна легко і безпосередньо прикріпити до цільового об'єкта. Більш того, RFID-наклейка є дуже дешевою [24].

Вони мають широке застосування. Наприклад, управління активами, відстеження упаковочних коробок на заводі, управління одягом тощо [24].

Пластикові RFID-теги використовують пластикову основу, таку як PVC, ABS, PET, для капсулювання мікросхеми та антени у різні форми тегів. Найпоширенішим є RFID-картки з PVC. Колір пластикового матеріалу може бути різноманітним. Крім того, пластикові матеріали, як правило, можуть витримувати високі температури [24].

RFID-мітки у скляних трубках капсулюють чіпи та антени всередині скляних контейнерів різної форми. Скляні RFID-мітки можуть бути вжиті для імплантації в тварин для їх ідентифікації та відстеження, а також для управління домашніми тваринами. Наприклад, це можуть бути цінні рибки, собаки, коти тощо. Їх також можна використовувати для маркування зброї, шоломів, пляшок, форм, прикрас або брелків [24].

Керамічна мітка інкапсулюється керамічним матеріалом. Вона гнучка, крихка і стійка до зносу. Крім того, це одна з анти-металевих міток, яка працює добре на металевих поверхнях. Керамічні RFID-мітки широко використовуються в управлінні паркуванням, управлінні електронними компонентами, тощо [24].

RFID-браслети - популярна форма упаковки міток. Їх виготовляють в багатьох кольорах і стилях, таких як силіконові браслети, паперові браслети, тканинні браслети і т. д. Ця форма дуже зручна і легка в носінні. Її часто використовують на фестивалях, концертах, у спортзалах, виставках, басейнах [24].

Також існують RFID-мітки запечатані епоксидним матеріалом, який є екологічним та безпечним для здоров'я. Вони мають невеликі розміри, кришталево прозорі, стійкі до зношення та високих температур. Вони можуть бути невеликою прикрасою, яка висить на ключі або рюкзаку. Епоксидні мітки часто використовуються для контролю доступу, членських карток, управління домашніми улюбленцями [24].

2.3 NFC

NFC - це скорочення від Near Field Communication (Комунікація близької дії). NFC – це технологія радіозв'язку короткого радіусу дії, яка дозволяє здійснювати комунікацію між пристроями, які знаходяться в безпосередній близькості один до одного. NFC Forum стандартизує цю технологію для використання відкритою платформою на мобільних

пристроях і системах, з якими вони взаємодіють. NFC-системи базуються на традиційній HF RFID, що працює на частоті 13,56 МГц. Типовий радіус дії для NFC може становити до 10 сантиметрів і залежить від протоколу, що використовується, та конструкції антени. На сьогодні стандарт NFC підтримує різні швидкості передачі даних до 424 кбіт/с. Принциповий механізм комунікації NFC між двома пристроями той самий, що й у традиційного RFID на частоті 13,56 МГц, де є випромінювач або зчитувач, та мітка/картка [26].

Антени випромінювача та мітки з'єднуються за допомогою електромагнітного поля, і цю систему можна розглядати як повітряну трансформаторну котушку (рис 2.4). Змінний струм проходить через первинну котушку (випромінювач), і цей струм створює поле в повітрі, викликаючи струм у вторинній котушці (мітці). Мітка може використовувати струм з поля для свого живлення. Дизайн, конфігурація та налаштування обох антен визначає ефективність зв'язку між пристроями. Загалом, індуктивний зв'язок через повітря дуже неефективний, і тому діапазон читання/запису досить обмежений. Зазвичай мітка - це пристрій, який містить антену, зазвичай називається вкладкою, а також невелику кількість пам'яті. Мітка - це пасивний пристрій, а енергію, необхідну для його роботи, забезпечує електромагнітне поле, створене випромінювачем [26].



Рисунок 2.4 – Приклад роботи NFC технології [26]

Також існує технологія активної комунікації між мітками, в якій використовується кодування бітів Міллера з 100-відсотковою модуляцією амплітуди. Тут сигнал змінюється між майже нульовою амплітудою та повною амплітудою, щоб сигналізувати низькі та високі значення. Кодування Міллера, або кодування із затримкою, працює за рахунок зміни часу між переходами сигналу в залежності від послідовності бітів порівняно з тактовим сигналом. Також передає дані зі швидкістю 106 кБіт/с [27].

Залежно від типу мітки, пам'ять може бути лише для читання, читання/запису або одноразового запису. Мітки NFC можуть використовуватися в майже будь-якій програмі, де вам потрібно, щоб мобільний пристрій отримував інформацію або мав певну взаємодію з користувачем. Дані можуть бути збережені на мітках, такі як URL-адреси, парування Wi-Fi або Bluetooth, SMS, Email, Vcard і конфігурації телефону. Ці набори даних відформатовані у структуру, визначену NFC Forum, і можуть бути використані операційною системою безпосередньо для асоціації даних з встановленими програмами, такими як браузер, текстові повідомлення, бездротові або інші програми. Користувач також може зберігати на цих мітках власні дані та використовувати їх за допомогою спеціально розробленої програми, встановленої ним. Ці дані не обов'язково повинні відповідати всім вимогам структури, визначеним NFC Forum, але виробник операційної системи має рекомендації в своїх API-визначеннях щодо того, як дані витягуються та обробляються. Для забезпечення сумісності серед різних постачальників міток NFC та виробників пристроїв NFC, NFC Forum наразі визначив п'ять типів міток. Більше типів міток визначаються, щоб дати розробникам пристроїв NFC конкурентоспроможний вибір [26].

Існуючі типи NFC міток представлені у таблиці 2.2.

Таблиця 2.2 – Типи NFC міток [28]

	Тип 1	Тип 2	Тип 3	Тип 4	Тип 5
Стандарт	ISO/IEC 14443A	ISO/IEC 14443A	ISO/IEC 18092 JIS X 6319-4 FELICA	ISO/IEC 14443A ISO/IEC 14443B	ISO/IEC 15693
Пам'ять	96 Б – 2 кБ	48 Б – 2 кБ	2 кБ	32 кБ	64 кБ
Швидкість	106 кбіт/с	106 кбіт/с	212 кбіт/с 424 кбіт/с	106 кбіт/с 212 кбіт/с 424 кбіт/с	26.48 кбіт/с
Анти-колізія	Ні	Так	Так	Так	Так

NFC забезпечує як односторонній, так і двосторонній зв'язок між передавачами, що підходить для багатьох програм. Пристрої NFC можуть діяти як електронні документи і ключ-карти. Вони використовуються в системах безконтактної оплати і дозволяють мобільну оплату, замінюючи або доповнюючи системи, такі як кредитні картки та електронні смарт-карти квитків. Іноді їх називають NFC/CTLS або CTLS NFC, при цьому безконтактність (contactless) скорочується як CTLS. NFC можна використовувати для обміну невеликими файлами, такими як контакти, а також для швидкого встановлення з'єднання для обміну більшими медіафайлами, такими як фотографії, відео та інші файли [29].

Пристрої з підтримкою NFC можуть діяти як електронні документи, які знаходяться в паспортах та посвідченнях особи, а також як ключ-карти для використання у квитках на проїзд, проїздних квитках, картках для входу, автомобільних ключах та пропусках. Короткий діапазон дії NFC та підтримка шифрування роблять його більш підходящим, ніж менш приватні системи RFID [30].

2.4 Штрих-код

Штрих-код є послідовністю чорних ліній і білих проміжків розташованих на чітко визначеній відстані один до одного, з чією допомогою відбувається кодування інформації в зручній формі для машинного зчитування. Кожний елемент кодується набором штрихів та

прогалин за чітко визначеними правилами (стандартами штрих-коду). Для зчитування штрих-коду використовується спеціальний сканер штрих-коду [31].

Штрих-коди поділяються на два типи: одновимірні (лінійні) та двовимірні. Одновимірний штрих-код кодує інформацію або дані між паралельними лініями та прогалинами між ними. Ці паралельні лінії мають різну ширину та різні прогалини між ними. Одновимірний штрих-код може зберігати лише алфавітно-цифрові дані; якщо потрібно зберегти більше інформації, то потрібно збільшити довжину штрих-коду [32].

Двовимірні штрих-коди можуть представляти більше інформації на одиницю площі і були розроблені для подолання обмеженої кількості інформації, яку можна упакувати в одновимірні штрих-коди. Двовимірні штрих-коди відображають інформацію по двом осям, утворюючи площу штрих-коду. Прикладами двовимірних штрих-кодів є 3-DI, Array Tag, PDF417, QR-код, Data Matrix, MaxiCode і Aztec Code. Крім того, двовимірні штрих-коди менше за розміром і мають меншу помилковість порівняно з їх одновимірними аналогами. Наприклад, у штрих-коді PDF417 найгірша помилковість становить 1 помилка на 10,5 мільйонів символів, порівняно з UPC-A, у якого вища помилковість – 1 помилка на кожні 394 тисячі символів. Однак двовимірні штрих-коди потребують складних сканерів, що робить їх витратнішими і менш популярними у роздрібній торгівлі [33].

Особливість лінійних, або їх називають смужковими штрих-кодами, полягає у тому що їх зчитують лише в одному напрямку (по горизонталі). Найбільш поширеними прикладами лінійних штрих-кодів є EAN (EAN-8 складається з 8 цифр, EAN-13 – використовуються 13 цифр), UPC (UPC-A, UPC-E), Code56, Code128 (UPC/EAN-128), Codabar «Interleaved 2 of 5» [32].

Символи в дискретному кодуванні складаються з n смуг і $n - 1$ пробілів. Між символами є додатковий пробіл, але він не передає інформацію і може мати будь-яку ширину, якщо він не є кінцем коду [34].

Символи в безперервному кодуванні складаються з n смужок і n пробілів, що примикають один до одного, причому один символ закінчується пробілом, а наступний починається смугою, чи навпаки. Для завершення коду потрібен спеціальний зразок кінця зі смугами на обох кінцях [33].

Стандарти перевірки штрих-кодів визначені Міжнародною організацією зі стандартизації (ISO) у ISO/IEC 15426-1 (лінійний) або ISO/IEC 15426-2 (двовимірні). Поточна міжнародна специфікація якості штрих-коду — ISO/IEC 15416 (лінійний) і ISO/IEC 15415 (двовимірні). Європейський стандарт EN 1635 був відкликаний і замінений на ISO/IEC 15416. Оригінальна специфікація якості штрих-коду США була ANSI X3.182. (UPC, що використовуються в США – ANSI/UCC5). Станом на 2011 рік робоча група ISO JTC1 SC31 розробляла стандарт якості прямого маркування деталей (DPM): ISO/IEC TR 29158 [35].

UPC, або Universal Product Code, є американським стандартом штрих-коду, який призначений для відстеження товарів у магазинах. Цей стандарт був розроблений у 1973 році Джорджем Джозефом Лорером, інженером з корпорації IBM. Код UPC складається з 12 цифр і є попередником європейського стандарту EAN-13, який містить 13 цифр. Для перетворення коду UPC на EAN-13, до дванадцяти цифр коду UPC додається нуль. Тому товари з США і Канади у європейському кодуванні починаються з нуля [36].

Код UPC - це практично симетричний лінійний штрих-код, який був розроблений у часи, коли цифрова техніка ще не була настільки розвиненою, як зараз. Симетричність цього коду і його висока перешкодозахищеність обумовлені тим, що він складається з двох груп цифр, по 6 чисел у кожній групі – лівої і правої. Кожна група цифр оточена захисними, або огорожувальними, штрих-шаблонами (Guard Patterns), які містять штрихи одиначної ширини (рис 2.5). Ці шаблони допомагають у синхронізації сканера штрих-коду. Наявність трьох таких полів обумовлена в першу чергу можливістю нанесення штрих-коду на закруглену поверхню [36].



Рисунок 2.5 – Штрих-код UPC [34]

У часи створення цього коду, знання ширини одиничного штриха на початку, середині і кінці було важливим для сканерів. Ліві і праві захисні шаблони складаються з трьох штрихів одиничної ширини - двох темних і одного світлого між ними. Середній захисний шаблон має п'ять штрихів - трьох світлих і двох темних. Решта шаблону складалася з цифр [37].

Сканер штрих-коду використовується для сканування штрих-кодів та отримання закодованої інформації, що міститься в них. Він складається з трьох основних компонентів: сенсора штрих-коду та декодера. Під час сканування штрих-коду сканер випромінює червоне світло на символ штрих-коду, підсвічуючи паралельні чорні та білі смуги та прогалини. Відбите світло потім виявляється сенсором усередині сканера, який генерує аналоговий сигнал на основі інтенсивності відбитого світла. Цей аналоговий сигнал приймається декодером, який інтерпретує його, перевіряє штрих-код на валідність за допомогою контрольної цифри та перетворює його на текст, зрозумілий для людини [38].

Розшифрований текст потім передається на комп'ютерну систему за допомогою спеціалізованого програмного забезпечення, яке забезпечує комунікацію між системою та сканером штрих-коду. Існують різні типи сканерів штрих-кодів, кожен з яких використовує різні методи сканування штрих-кодів, кожен зі своїми перевагами та недоліками. Ці сканери використовуються для конкретних цілей залежно від їх можливостей [39].

2.5 QR-код

QR-код - це тип матричного штрих-коду або двовимірного коду, який може зберігати інформацію та призначений для читання смартфонами (рис. 2.6). QR - це скорочення від "Quick Response" (швидка відповідь), що вказує на те, що вміст коду повинен бути декодований дуже швидко з високою швидкістю. Закодована інформація може бути текстом, URL-адресою або іншими даними. QR-код був розроблений для того, щоб його вміст можна було декодувати з високою швидкістю. Популярність QR-кодів стрімко зростає по всьому світу. У наш час мобільні телефони з вбудованою камерою широко використовуються для розпізнавання QR-кодів [40].



Рисунок 2.6 – Приклад QR-коду [40]

QR-код складається з чорних квадратів, розташованих у квадратній сітці на білому тлі, які можна зчитувати пристроєм обробки зображень, таким як камера, і обробляти за допомогою кода Ріда-Соломона, доки зображення не буде належним чином інтерпретовано. Потім необхідні дані витягуються з візерунків, які присутні як у горизонтальних, так і в вертикальних компонентах зображення (рис. 2.7) [41].

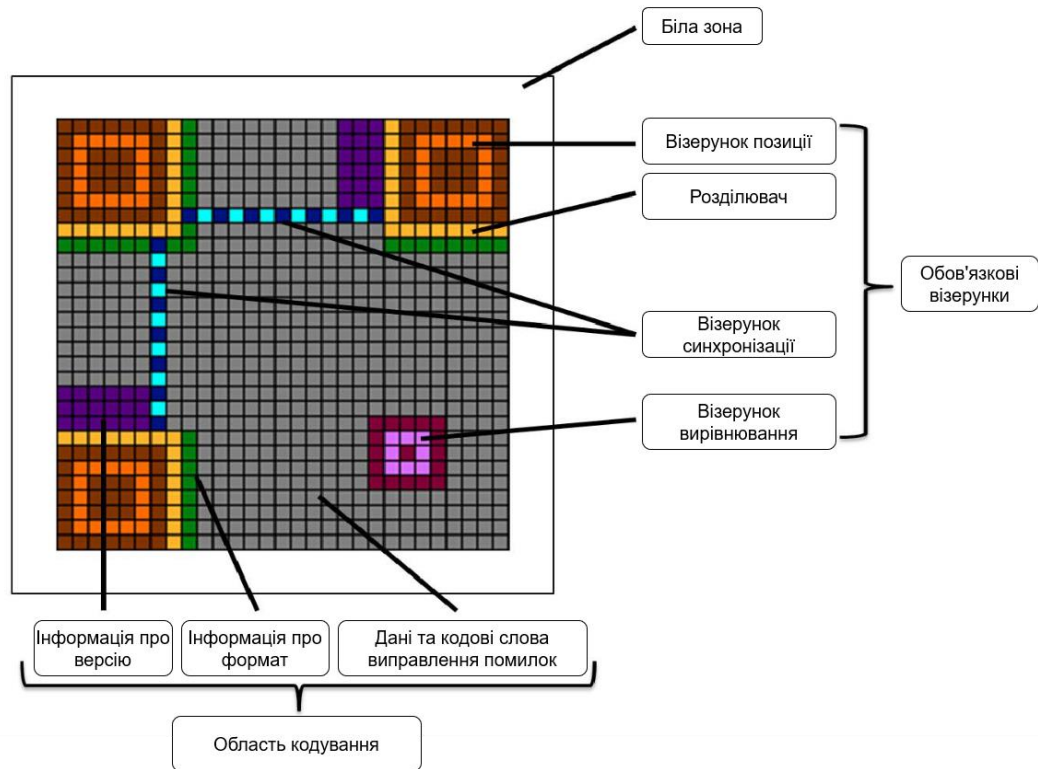


Рисунок 2.7 – Структура QR-коду з позначеними функціональними елементами [40]

Кожен символ QR-коду повинен складатися з квадратних модулів, розташованих у квадратному масиві і містити обов'язкові візерунки, та область кодування. І весь символ повинен бути оточений з усіх чотирьох сторін білою зоною [42].

Обов'язкові візерунки - це візерунки, які повинні бути розміщені в певних областях QR-коду, щоб забезпечити правильну ідентифікацію та орієнтацію коду для декодування. Існує 4 типи обов'язкових візерунків: розділювач, візерунок позиції, вирівнювання, та синхронізації [40].

Візерунок позиції – це спеціальний візерунок для визначення положення, що розташований в трьох кутах (верхній лівий, верхній правий і нижній лівий) кожного QR-коду. Вони складаються з зовнішнього темного квадрата розміром 7×7 , внутрішнього світлого квадрата розміром 5×5 модуля (модуль відноситься до чорно-білих квадратів, які складають QR-код.) і твердого темного квадрата у центрі розміром 3×3 модуля. Співвідношення ширини модулів у кожному шаблоні визначення положення – 1:1:3:1:1

(рис. 2.8). Візерунок позиції призначений бути візерунком, який мало ймовірно з'явиться в інших частинах QR-коду, щоб сканери могли знаходити це співвідношення світлих і темних модулів для виявлення візерунка позиції та правильної орієнтації QR-коду для декодування [40].

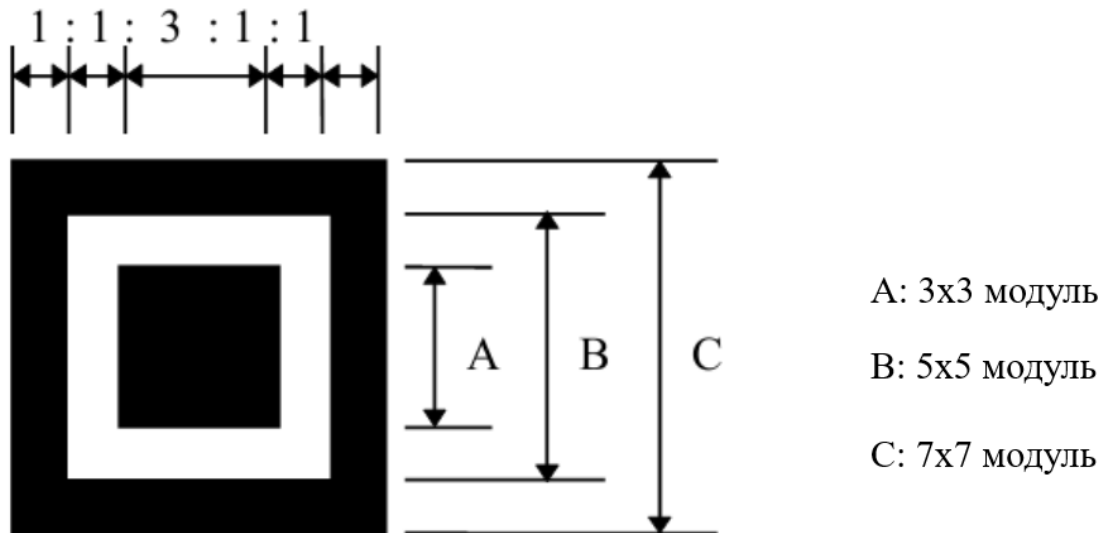


Рисунок 2.8 – Структура візерунка позиції [40]

Візерунок вирівнювання складається з 5×5 темного модуля, 3×3 світлого модуля та одного темного модуля в центрі. QR-коди, починаючи з версії 2 та більше, повинні мати візерунки вирівнювання, і кількість таких візерунків залежить від версії QR-коду [40].

Розділювачі - це області з одиничною шириною, які містять вільний простір між кожним візерунком позиції та областю кодування [40].

Існує два візерунки синхронізації: горизонтальний і вертикальний. Вони складаються з темних і світлих модулів, що чергуються. Горизонтальний візерунок синхронізації розташований у 6-му рядку QR-коду між розділювачами. Вертикальний шаблон синхронізації знаходиться у 6-му стовпці QR-коду між розділювачами. Ці візерунки корисні для моделювання координат, визначення щільності символів та розташування області інформації про версію [40].

Область кодування містить інформацію про формат, інформацію про версію, дані та кодові слова виправлення помилок. Для інформації про формат необхідно залишити одномодульний блок біля верхнього лівого, верхнього правого та нижнього лівого шаблонів пошуку, а також блок розміром 6×3 над нижнім лівим шаблоном пошуку і блок розміром 3×6 ліворуч від верхнього правого шаблону пошуку [40].

Біла зона – це область шириною 4 модуля, в якій немає даних і призначена для того, щоб забезпечити те, що навколишній текст або маркування не спричинять помилкового розпізнавання даних QR-коду [40].

Версії символів QR-коду варіюються від версії 1 до версії 40 (рис. 2.9). Кожна версія має різну конфігурацію модуля або кількість модулів. «Конфігурація модуля» відноситься до кількості модулів, що містяться в символі, починаючи з версії 1 (21×21 модуль) до версії 40 (177×177 модулів). Кожен вищий номер версії складається з 4 додаткових модулів на кожную сторону [43].

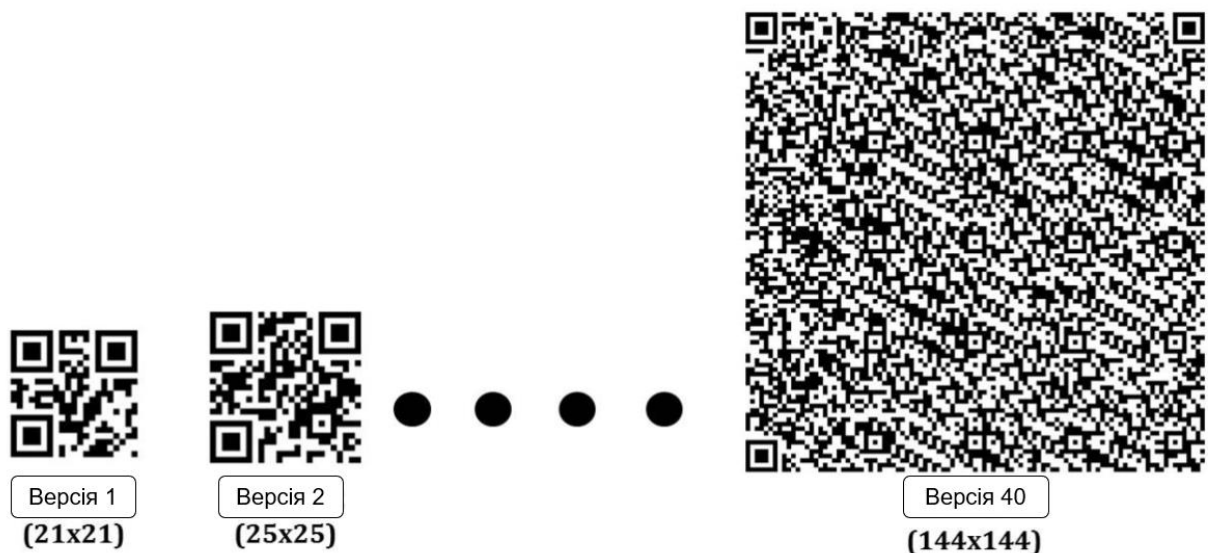


Рисунок 2.9 – Версії QR-коду [40]

QR-код здатний обробляти всі типи даних, наприклад, числові та алфавітні символи, кандзі, кана, хірагана, символи, двійкові та керуючі коди. В одному символі можна закодувати до 7089 символів (таблиця 2.3) [41].

Таблиця 2.3 – Максимальна ємність для зберігання символів [43]

Режим введення	Макс. символів	Можливі символи, кодування за замовчуванням
Лише числові		0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Буквено-цифровий	4296	0–9, A–Z (тільки великими), пробіл, \$, %, *, +, -, ., /,
Двійковий (8 біт)	2953	ISO 8859-1
Кандзі	1817	Shift JIS, X 0208

2.6 Оптичне розпізнавання символів

Перша використана техніка розпізнавання рук називається «оптичне розпізнавання символів» або просто OCR (Optical Character Recognition). Це дозволяє користувачам сканувати рукописні тексти в прості, легкодоступні та доступні для пошуку цифрові форми. Більшість пакетів сканування включають певний тип OCR [44].

OCR - це механічне або електронне перетворення зображень надрукованого, рукописний або друкований текст у машинно-кодований текст. Ранні версії потрібно було навчати із зображеннями кожного символу, і вони могли працювати з одним шрифтом за раз. Удосконалені системи, здатні забезпечувати високу точність розпізнавання більшості поширених нині шрифтів [45].

Сучасні системи оптичного розпізнавання символів можуть бути умовно розділені на дві категорії. Власне OCR-системи вирішують класичні завдання розпізнавання друкованих символів, нанесених на папір за допомогою принтера, плоттера або друкарської машинки (при цьому мається на увазі, що будь-яка система розпізнавання працює з електронним зображенням документа, зазвичай отримуються за допомогою сканера). Крім того, виділяють клас ICR-систем (Intelligent Character Recognition), в завдання яких входить обробка документів, заповнених друкованими літерами і цифрами від руки, або, інакше кажучи, розпізнавання рукописних символів [46].

Процес перетворення зображення в редагований документ ділиться на кілька етапів. Кожен крок - це набір пов'язаних алгоритмів, які виконують частину роботи OCR. Загальні етапи процесу розпізнавання тексту такі:

- завантаження зображення;
- виявлення найважливіших характеристик зображення;
- покращення якості зображення;
- «бінаризація» зображення;
- виявлення та видалення ліній;
- «зонування» зображення;
- виявлення текстових рядків і слів;
- комбіновано-розбитий аналіз символів;
- розпізнавання символів;
- підтримка словника;
- збереження результатів у вибраному вихідному форматі.

При завантаженні зображення як растрового зображення джерелом може бути файл або вказівник на блок пам'яті. Крім того, хороша система OCR повинна розуміти багато форматів зображень: BMP, TIFF (як односторінкові, так і багатосторінкові зображення), JPEG, PNG і так далі. Він також повинен підтримувати PDF-файли, оскільки багато документів зберігаються як зображення у форматі PDF, і єдиний спосіб отримати текст із таких файлів — розпізнати текст.

При виявленні характеристик зображення найважливішими з них є такі характеристики як роздільна здатність і інверсія. Багато алгоритмів оптичного розпізнавання символів очікують попередньо визначеного діапазону розмірів шрифту та кольорів переднього/фонового плану, тому зображення має бути перемасштабовано та інвертовано перед обробкою, якщо це необхідно.

Зображення може бути перекошеним або містити багато шуму, тому для покращення якості зображення застосовуються алгоритми вирівнювання та усунення шумів (рис. 2.10).

Optical character recognition, machine-encoded text, computer vision, telegraph code, computer program, spam-busting system, CAPTCHA, WebOCR, OCR algorithm, pattern matching, image correlation, speech synthesizers.

Character Recognition (CR) has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and also creates an increasing demand on many emerging application domains, which require more advanced methodologies.

Optical character recognition (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text.

Early versions needed to be trained with images of each character, and could work on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Early optical character recognition can be traced back to technologies involving telegraphy and creating reading devices for the blind. In 1914, Emanuel Goldberg developed a machine that read characters and converted them into standard telegraph code.

In 1974, Ray Kurzweil started the company Kurzweil Computer Products, Inc. and continued the

Optical character recognition, machine-encoded text, computer vision, telegraph code, computer program, spam-busting system, CAPTCHA, WebOCR, OCR algorithm, pattern matching, image correlation, speech synthesizers.

Character Recognition (CR) has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and also creates an increasing demand on many emerging application domains, which require more advanced methodologies.

Optical character recognition (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text.

Early versions needed to be trained with images of each character, and could work on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Early optical character recognition can be traced back to technologies involving telegraphy and creating reading devices for the blind. In 1914, Emanuel Goldberg developed a machine that read characters and converted them into standard telegraph code.

In 1974, Ray Kurzweil started the company Kurzweil Computer Products, Inc. and continued the

Оригінальна картинка Картинка після вирівнювання

Рисунок 2.10 – Демонстрація вирівнювання зображення [47]

Багато алгоритмів оптичного розпізнавання символів можуть працювати лише з двотональними зображеннями, тому кольорові зображення або зображення у відтінках сірого потрібно перетворювати на двотонові. Процес називається «бінаризація». Цей крок дуже важливий, оскільки неправильна бінаризація викличе масу проблем (рис. 2.11).



Рисунок 2.11 – Демонстрація бінаризації зображення [47]

Крок виявлення та видалення ліній потрібен для покращення аналізу макета сторінки, досягнення кращої якості розпізнавання підкресленого тексту, виявлення таблиць тощо (рис. 2.12).

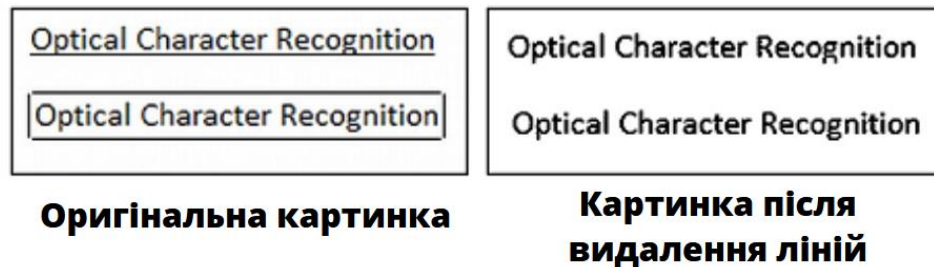


Рисунок 2.12 – Демонстрація видалення ліній на зображенні [47]

При аналізі макета сторінки (також називається «зонування») система OCR повинна визначати положення та типи всіх важливих областей на зображенні.

Іноді виявлення текстових рядків і слів є непростим завданням через різний розмір шрифту та малі проміжки між словами.

Часто деякі символи розбиті на кілька частин, або деякі символи торкаються один одного. Необхідно виявляти такі випадки і за допомогою комбіновано-розбитого аналізу символів знаходити правильну позицію кожного символу (рис. 2.13).



Рисунок 2.13 – Комбіновані та розбиті символи [47]

Основним алгоритмом OCR є розпізнавання символів. Зображення кожного символу має бути перетворено на відповідний код символу. Іноді цей алгоритм створює кілька кодів символів для невизначених зображень.

Наприклад, розпізнавання зображення символу «I» може виробляти коди «I», «|», «l»; остаточний код символу буде вибрано пізніше.

Підтримка словника може покращити якість розпізнавання. Деякі символи, такі як «l» і «I», «C» і «G», можуть виглядати дуже схожими, і словник може допомогти прийняти рішення.

Останнім кроком є збереження результатів у вибраному вихідному форматі, наприклад PDF, DOC, RTF або TXT з можливістю пошуку. Важливо зберегти оригінальний макет сторінки: колонки, шрифти, кольори, малюнки, фон і так далі [47].

Порівняння програмного забезпечення для оптичного розпізнавання символів наведено у табл. 2.4.

Таблиця 2.4 – Програми розпізнавання символів [48]

Назва	Ліцензія	Операційні системи	Замітки
1	2	3	4
ABBYY FineReader	Комерційна Власницька	Windows; Linux, Mac OS	Для роботи з різними мовами потрібна підтримка відповідної мови.
Brainware	Комерційна	Windows	Добування даних із документів і їхня обробка; наприклад, рахунки, повідомлення, накладні й платіжні доручення
COCR2	Безкоштовна	Windows 9X, ME, 2000, XP	Програма для розпізнавання спрощених і традиційних китайських ієрогліфів. Основне обмеження програми: для кожного ієрогліфа користувач повинен сам вибирати варіант його розпізнавання за допомогою миші або клавіатури. Але кількість розпізнаваних ієрогліфів досить велика — більше 10 000.
CuneiForm	BSD	Windows (з GUI), Linux, Mac OS, FreeBSD (CLI)	Промислова, багатомовна система, уміє зберігати форматування тексту й розпізнає заплутані таблиці довільної структури
ExperVision TypeReader & RTK	Комерційна	Windows, Mac OS X, Unix, Linux, OS/2	Отримувала високі оцінки на початку 1990-х.

Продовження таблиці 2.4

1	2	3	4
FreeOCR	Apache	Інтерфейс: Браузер; Сервер: POSIX, Unix	Платформа для браузерних систем розпізнавання символів. Використовує Tesseract. Велика кількість підтримуваних мов.
HOOCR	GPL	Linux	Розпізнавання текстів на івриті
Kirtas Technologies Arabic OCR	Комерційна	Windows	Може розпізнавати арабські й англійські символи на одній сторінці.

2.7 Біометричні дані

На сьогоднішній день існують різноманітні методи і способи біометричної ідентифікації, проте вони в основному базуються на фізіологічних характеристиках та особливостях поведінки людини. До таких напрямків належать розпізнавання за геометрією руки та пальців, венозною структурою, райдужною оболонкою, внутрішньою структурою дна ока, рисами обличчя та відбитками пальців. Також відомі спроби використання зовнішньої форми вуха, структури долоней та запаху людського тіла для цих цілей, хоча результати їх практичного застосування залишаються невідомими [49].

У сучасних біометричних системах швидкість операцій розпізнавання або ідентифікації навіть за наявності тисяч користувачів (наприклад, у великих підприємствах, аеропортах, атомних станціях) вимірюється в секундах, що дозволяє задовольняти вимоги виробничого режиму. Спочатку системи біометричної ідентифікації використовувалися організаціями з високим рівнем безпеки, такими як банки, арсенали та паспортні системи [50].

Сучасні методи біометричної аутентифікації можна класифікувати на дві основні групи: статичні і динамічні методи аутентифікації [51].

Статичні методи біометричної аутентифікації базуються на застосуванні фізіологічних характеристик людини, які є статичними та унікальними з моменту народження:

– за портретом обличчя. Макіяж, вікові зміни та вживання алкоголю можуть ускладнити процес ідентифікації людини за обличчям. Навіть для експерта, ідентифікація людини за фотографією, якій вже 10 років, може бути дуже складним завданням. Біометричні технології дозволяють автоматично проводити контроль осіб за обличчям, порівнюючи параметри обличчя особи, знятої камерою, з даними в базі. Точність такого аналізу складає 86-93%, що значно перевищує звичайні людські можливості. Однак, для здійснення цієї функції потрібне високоякісне обладнання. Навіть для візуальної ідентифікації особи, відстань між центрами зіниць має бути еквівалентною 200 пікселям [50];

– за відбитком пальця. Цей метод біометричної аутентифікації ґрунтується на унікальності папілярних узорів на пальцях кожної людини. Сканер зчитує відбиток пальця, який потім перетворюється в цифровий код і порівнюється з еталоном, збереженим у базі даних. Цей метод є найбільш поширеним серед інших методів біометричної аутентифікації [52];

– за формою долоні. Цей метод базується на геометрії грона руки, яке є унікальним для кожної людини. Для отримання ідентифікуючої інформації застосовується спеціальний пристрій, що складається з камери та декількох підсвічуючих діодів. Включаючись по черзі, діоди створюють різні проекції долоні, за якими будується тривимірний образ грона руки. Після цього створюється цифровий код (згортка), який порівнюється з введеним раніше еталоном, що дозволяє розпізнати людину [52];

– за розташуванням вен на лицьовій стороні долоні. Інфрачервона камера зчитує теплове випромінювання від кожної вени, що має свій унікальний тепловий підпис. Отриманий зображення вен обробляється за допомогою комп'ютерних алгоритмів, що дає змогу створити цифрову згортку венозної системи руки. Ця технологія біометричної ідентифікації на основі венозного малюнку вважається надійною, проте недоліком цієї технології є відсутність єдиної бази даних венозних малюнків, крім того, використання інфрачервоних камер може бути витратним і складним у

деяких умовах, наприклад, при низьких температурах або в присутності джерел інтерференції [51].

Динамічні методи базуються на характерних поведінкових особливостях людини, таких як підсвідомі рухи під час виконання певної дії:

- за голосом. Щоб ідентифікувати абонента за голосом, необхідно мати мовний шаблон, з яким буде порівнюватися голосовий ключ, що вводиться в систему. Порівняння ключа і шаблону може здійснюватися як в цілому, так і за окремими характеристиками мовного сигналу (тобто цифровий мовний сигнал, що пройшов обробку і адаптований до поставленого завдання): амплітудою і потужністю (гучністю), часовими, частотними (тембр), енергетичними та фазовими характеристиками;

- за динамікою натискання клавіш. У традиційних системах захисту інформації доступ здійснюється за допомогою паролів. Якщо пароль зробити довшим, з'являється можливість спостерігати характерний для користувача клавіатурний почерк. Наприклад, використовується така парольна фраза: «Пароль — це спосіб захисту інформації». При введенні такої парольної фрази біометрична система фіксує час натиснення кожної клавіші та інтервал часу між натисканням чергової клавіші і відпуском попередньої;

- за динамікою рукописного підпису. Досліджується час затримки (проміжок часу, протягом якого користувач утримує конкретну клавішу) та час «польоту» (проміжок часу, який потрібен користувачеві для переходу з однієї клавіші на іншу);

- за динамікою роботи з маніпулятором «миша». Досліджується образ характерної траєкторії маніпулятора «миша» при роботі користувача в системі [52].

2.8 Висновки до розділу

У даному розділі були розглянуті відомі методи та засоби ідентифікації об'єктів, їх особливості та структуру. А саме такі методи як RFID, NFC, QR-код, OCR, штрих-код та біометричні дані.

Серед них найбільш підходящим для виконання мети дипломної роботи було обрано ідентифікацію за допомогою RFID. Кожен з методів має свої можливі застосування, але кожна з них не підходить по різних причинах. Такі методи як QR-код, OCR та штрих-коди не були обрані через необхідність сканувати такі коди напряду, без перешкод між сканером та мітками такого типу. Ідентифікація за біометричними даними, в свою чергу, дозволяє сканувати лише людей, що також не підходить для майбутнього застосування.

Найбільше відповідають вимогам застосування два методи ідентифікації: RFID та NFC. Ці технології схожі між собою по функціям та можливостям, але перевага була надана саме RFID через його простотішу структуру. Хоч NFC може передавати більше різноманітних даних, але оскільки метою застосування є лише ідентифікація об'єктів, то ці дані були би зайвими і повні можливості технології не використовувалися. Все це зробило вибір RFID-міток найбільш оптимальним варіантом.

3 ПРОЄКТУВАННЯ ПРОГРАМНО-АППАРАТНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз вимог до системи

За проведеним аналізом предметної області було встановлено, яким вимогам має відповідати програмно-апаратне рішення.

Функціональні вимоги:

- розпізнавати RFID-мітки;
- можливість додавати дані о RFID-мітках у БД;
- можливість видаляти дані о RFID-мітках у БД;
- виконувати пошук міток у БД;

Нефункціональні вимоги:

- швидкодія;
- стабільність;
- зручність використання.

3.2 Розробка архітектури системи

Архітектура системи була розроблена на основі аналізу вимог і представлена на рисунку 3.1.

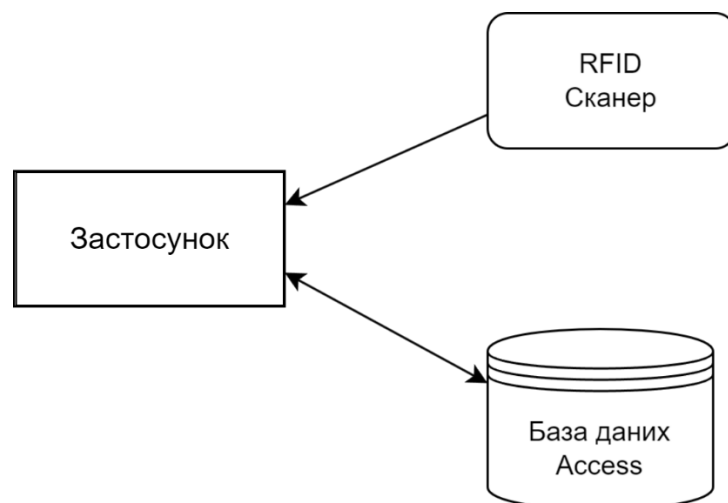


Рисунок 3.1 – Узагальнена архітектура системи

Архітектура системи складається з трьох основних частин:

- RFID-сканера;
- застосунку;
- бази даних.

RFID-сканер (антена) має бути під'єднаний до персонального комп'ютера на якому встановлено застосунок, використовуються щоб надати сканувати дані що надходять з RFID-міток для надсилання цих даних застосунку.

База даних – сховище із даними про RFID-мітки та інформацією про їх місцезнаходження.

Застосунок – програмне забезпечення за допомогою якого проводиться взаємодія з RFID-мітками і БД.

3.3 Вибір та обґрунтування апаратних засобів розробки

Для розробки проєкту апаратною базою було обрано зчитувач радіочастотної ідентифікації FONKAN FM-505.

FONKAN FM-505 – це UHF RFID зчитувач, що підтримує мультизчитування RFID-міток і має дальність зчитування до трьох метрів. Він має можливість легкого підключення до будь-яких платформ, серед яких є персональні комп'ютери, телефони з операційною системою Android, Arduino, Raspberry Pi та ін.

Для обґрунтування вибору у якості апаратної бази FONKAN FM-505 роздивимось його характеристики, та характеристики близьких аналогів.

Характеристики наведено у таблиці 3.1.

Дивлячись на характеристики можливо побачити що FONKAN FM-505 не є кращим із можливих зчитувачів, але є доступним і задовольняє усім критеріям проєкту завдяки швидкодії, достатньої потужності та низької вартості.

Таблиця 3.1 – Характеристики RFID-зчитувачів

Характеристика	Назва модуля		
	FONKAN FM-505	CiAODA SRF-A820	FONKAN FA-305A
Робоча частота	865-868 MHz 902-928 MHz	920-928 MHz 902.75-927.25 MHz 865-868 MHz	860-960 MHz
Посилення антени	3 dBi	7.56 - 10.59 dBi	1 - 5.5dbi
Відстань зчитування	до 3 м	до 9 м	до 7 м
Швидкість читання	50 міток/с	750 міток/с	невідомо
Робоча напруга	3.3 - 5 V	12 V	невідомо
Ціна \$	15-18	невідомо	20-22

3.4 Вибір та обґрунтування вибору мови програмування

Програмний продукт написаний на мові C#.

C# – це популярна мова програмування, яка широко використовується для розробки програм, зокрема в екосистемі Microsoft [53].

C# був розроблений під впливом мов C, C++ і Java, але творці взяли найкраще з них і впровадили нові ідеї, такі як типи значень, властивості та події. Наприклад, C# не дозволяє використовувати сирі вказівники безпосередньо до пам'яті та не підтримує множинне успадкування класів. C# має збирач сміття, який автоматично керує пам'яттю. У більшості випадків це означає, що вам не доведеться турбуватися про управління пам'яттю, що значно спрощує роботу з C# [54]-[55].

Також важливим є факт того що C# є мовою програмування з відкритим вихідним кодом, що дозволяє спільноті безперешкодно його вивчати, розробляти та оновлювати [56].

Обґрунтування вибору середовища розробки наведено у таблиці у таблиці 3.2.

Таблиця 3.2 – Обґрунтування вибору мови програмування

Характеристика	Мова програмування		
	C#	C++	Java
Наявність стандартних бібліотек для роботи з портами	+	-	-
Підтримка властивостей	+	-	-
Множинне наслідування класів	-	+	-
Швидкодія	+-	+	+-
Збирач сміття	+	-	-
Відкритий код	+	-	+

Таким чином, C# відповідає найбільшій кількості критеріїв для реалізації проєкту. А саме в першу чергу за її підтримку обраної у попередньому розділі мови програмування, можливість легко завантажити програму на платформу, доступність та простоту у використанні.

3.5 Вибір та обґрунтування вибору середовища розробки

Середовищем розробки було обрано Microsoft Visual Studio 2022, що є популярним інтегрованим середовищем розробки для створення багатофункціональних кросплатформених додатків для Windows, Mac, Linux, iOS та Android. Вона є результатом довгої історії розвитку мов програмування та інтерфейсів і увібрала у собі досягнення багатьох середовищ розробки програмного забезпечення [57].

Microsoft Visual Studio має наступні переваги:

- інсталятор на основі робочих навантажень дозволяє встановлювати лише те, що потрібно для розробки проєкту;
- потужні інструменти та можливості програмування;
- розробка для кількох платформ;

- інтеграція контролю версій;
- інтеграція штучного інтелекту, що дозволяє писати код більш ефективно завдяки його підтримці [58].

Оскільки у C# було спеціально створено Microsoft для платформи .NET, то і обрання Microsoft Visual Studio є логічним вибором для проєкту.

3.6 Висновки до розділу

В якості апаратної бази було обрано зчитувач радіочастотної ідентифікації FONKAN FM-505, за його невелику ціну та непогані характеристики.

Для реалізації проєкту ідентифікації об'єктів було обрано мову програмування C#, що є універсальною, статично типізованою, яка використовується для створення застосунків на Windows.

Для створення програмного забезпечення та його відладки обрано середовище розробки Microsoft Visual Studio.

4 КОНСТРУЮВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розробка алгоритму функціонування системи

На рисунку 4.1 наведено узагальнену схему алгоритму функціонування системи.

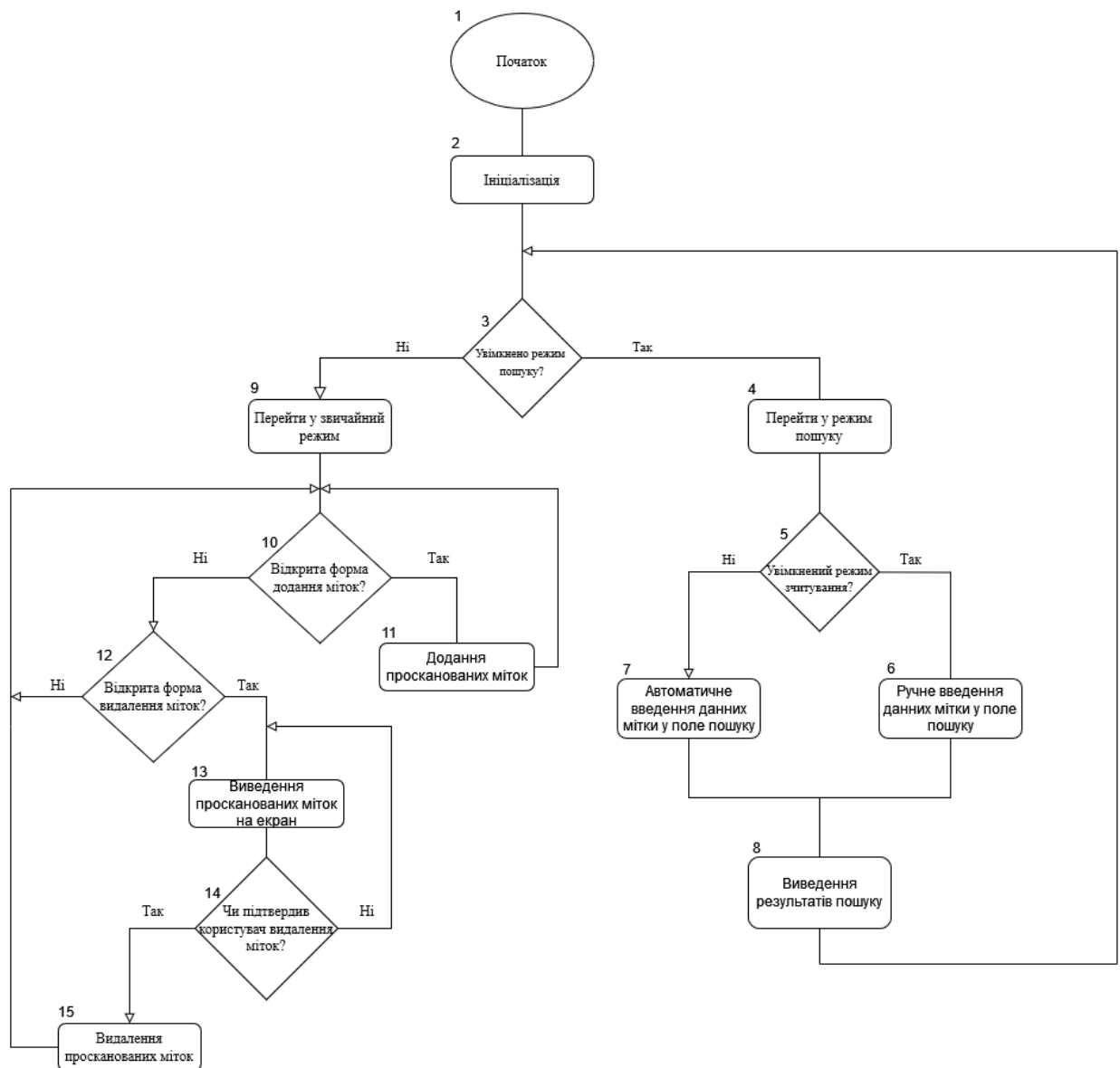


Рисунок 4.1 – Узагальнений алгоритм функціонування системи

Блок 1. Початок.

Блок 2. Встановлення значення змінних підключення бібліотек.

Блок 3. Вибір режиму функціонування. Перехід до блоку 4 в разі вибору режиму пошуку міток, інакше програма залишається у звичайному режимі і перехід до блоку 9.

Блок 4. Вимкнення можливості додавання та видалення міток, увімкнення можливості шукати мітки.

Блок 5. У разі вибору сканування за допомогою антени перехід до блоку 7, інакше до блоку 5.

Блок 6. Введення назви, або складу мітки котру треба знайти.

Блок 7. Сканування зчитувачем мітки і введення її номеру у поле пошуку.

Блок 8. Пошук даних у БД і виведення результатів.

Блок 9. Вимкнення можливості шукати мітки та увімкнення можливості додавання та видалення міток у БД.

Блок 10. Перехід до блоку 13 в разі вибору режиму видалення міток, інакше перехід до блоку 10.

Блок 11. Відкриття форми сканування міток та додання даних о відсканованих мітках до БД.

Блок 12. Перехід до блоку 11 в разі вибору режиму сканування міток, інакше перехід до блоку 12.

Блок 13. Відкриття форми видалення міток і виведення інформації о всіх просканованих міток в час відкриття форми них на екран.

Блок 14. У разі підтвердження видалення міток перехід до блоку 15, інакше до блоку 13.

Блок 15. Видалення даних обраних міток в БД.

4.2 Розробка БД

Для зберігання даних об RFID-мітках був обраний спосіб завантаження даних в базу даних локального сховища. У якості локального сховища була використана платформа система управління базами даних

Microsoft Office Access. Загалом БД містить дві таблиці: Items, Storages. В таблиці Items зберігаються дані про проскановані RFID-мітки. В таблиці Storages зберігаються дані про існуючі склади.

Структура таблиць представлена на рисунку 4.2.

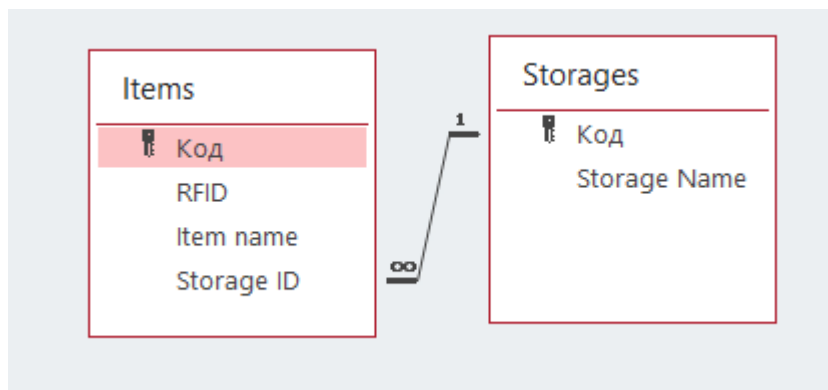


Рисунок 4.2 – Структура бази даних

У таблиці 4.1 наведено опис атрибутів таблиць.

Таблиця 4.1 – Опис атрибутів таблиць бази даних

Таблиця	Атрибут	Тип	Опис
Items	Код	Лічильник	Унікальний ідентифікатор
	RFID	Короткий текст	Код RFID-мітки
	Item name	Короткий текст	Назва об'єкту
	Storage ID	Числовий	Зовнішній ключ на таблицю Items
Storages	Код	Лічильник	Унікальний ідентифікатор
	Storage Name	Короткий текст	Назва складу

4.3 Опис особливостей програмної реалізації

Текст програми наведено в Додатку А.

Взаємодія з портом до якого підключена антена відбувається за допомогою бібліотеки System.IO.Ports, що містить класи для роботи з послідовними портами (COM-портами).

У застосунку використовувались наступні можливості System.IO.Ports:

- конфігурація параметрів порту, що дозволяє налаштовувати такі параметри, як швидкість передачі (BaudRate), кількість бітів даних, біти парності, стопові біти і керування потоком;
- відкриття і закриття порту за допомогою методів для відкриття Open() і закриття Close() послідовного порту;
- підтримка подій для обробки асинхронних операцій, а саме DataReceived, що дозволяють реагувати на подію, передачі даних з порту.

Опис використаних бібліотек які використовувалися для створення застосунку наведено у таблиці 4.2.

Таблиця 4.2 – Опис використаних бібліотек

Бібліотека	Опис бібліотеки
System.IO.Ports	Надає класи та функції для роботи з послідовними портами (COM-портами)
System.Windows.Forms	Надає засоби для розробки форм, діалогових вікон та інших елементів інтерфейсу
System.Data	Забезпечує можливість роботи з базами даних і джерелами даних
System	Містить фундаментальні класи та базові типи, використовувані у всіх інших просторах імен

Застосунок складається із трьох форм на відображення і ядра програми: Form1.cs, Form2.cs, Form3.cs, Program.cs. Form1.cs є головним вікном програми, де можна здійснити пошук міток у БД, а бо перейти до інших форм програми. Form2.cs є вікном де за допомогою сканера можна видалити мітки з БД. Form3.cs є вікном де можна сканувати та додавати мітки у БД. Далі буде детально описано класи і функції кожної з форм.

Форма Program.cs окрім автоматично створених середовищем класів містить користувацький клас MySerialPort, що відповідає за взаємодію RFID сканером через послідовний порт.

Клас MySerialPort містить наступні методи:

- ChangeText(). Цей метод фільтрує текст який приходить зі сканеру через послідовний порт. Прибирає перший символ, так як він технічний, не пропускає текст, розмір якого відрізняється по довжині коду міток, а саме від 32 символів. Робить виклик події і відсилає підготовлені дані з порту;
- Open(). Цей метод відкриває порт, якщо порт вже відкрито, то спочатку закриває його, після чого відкриває знову;
- SerialPort_DataReceived(). Це метод, який спрацьовує коли приходять дані у послідовний порт. Видаляє зайві пробіли і переведення рядка. Визиває метод ChangeText().

У таблиці 4.3 наведено основні функції форми Form1.cs.

Таблиця 4.3 – Функції форми Form1.cs

Назва функції	Опис
1	2
public Form1()	Спрацьовує при ініціалізації форми. Додає обробник подій з класу MySerialPort
Program_TextUpdated()	Функція, що викликається при події зчитування RFID мітки. Якщо увімкнен режим пошуку і режим зчитування RFID для пошуку, тоді ця функція записує RFID у поле пошуку по RFID
private void Form1_Load()	Спрацьовує при завантаженні форми. Була створена автоматично, заповнює даними з бази даних елементи database1DataSet.
private void button1_Click()	Спрацьовує при натисненні кнопки «Видалити записи». Передає у Form2 екземпляр поточної форми, тобто Form1 і викликає Form2
private void button2_Click()	Спрацьовує при натисненні кнопки «Додати записи». Передає у Form3 екземпляр поточної форми, тобто Form1 і викликає Form3
public Database1DataSet.DataTable e1DataTable GetItems()	Функція повертає DataTable від database1Dataset з Form1 для використання в інших формах
public void data_grid_update()	Функція додає нову строку у database1Dataset, потрібна для виклику з інших форм

Продовження таблиці 4.3

1	2
private void checkBox1_Checked Changed()	Спрацьовує при зміні стану checkBox1, якщо включено «Режим пошуку», то кнопки «Додати записи» і «Видалити записи» стають неактивними, а усе поле пошуку стає активним. І навпаки, якщо прибрати «Режим пошуку», тоді поле пошуку стає неактивним, а кнопки «Додати записи» і «Видалити записи» стають активними
private void checkBox2_Checked Changed()	Спрацьовує при зміні стану checkBox2 «Зчитати RFID», якщо включено, відкриває послідовний порт, при виключенні закриває послідовний порт
private void button3_Click()	Спрацьовує при натисненні кнопки «Пошук». Робить SQL запит до database1DataSet по усім полям пошуку і виводить результат чи результати у dataGridView2

У таблиці 4.4 наведено основні функції форми Form2.cs.

Таблиця 4.4 – Функції форми Form2.cs

Назва функції	Опис
public Form2()	Спрацьовує при ініціалізації форми. Додає обробник подій з класу MySerialPort. Передає екземпляр Form1 у локальну змінну типу Form1
private void Program_TextUp dated()	Спрацьовує при події зчитування RFID. Додає у dataGridView1 записи об'єктів код яких при зчитуванні співпадає з вже існуючими об'єктами у базі даних. При цьому додає об'єкт з одним кодом тільки один раз
private void button1_Click()	Спрацьовує при натисненні кнопки «Видалити відскановане». Видаляє записи з бази даних, які співпадають з просканованими і занесеними у dataGridView1
private void button2_Click()	Спрацьовує при натисненні кнопки «Видалити зі списку». Видаляє з dataGridView1 строку, якщо та була помилково зчитана
private void Form2_Load()	Спрацьовує при завантаженні форми. Відкриває послідовний порт
Form2_FormClos ing()	Спрацьовує при закритті форми. Закриває послідовний порт

У таблиці 4.5 наведено основні функції форми Form3.cs.

Таблиця 4.5 – Функції форми Form3.cs

Назва функції	Опис
public Form3()	Спрацьовує при ініціалізації форми. Додає обробник подій з класу MySerialPort. Передає екземпляр Form1 у локальну змінну типу Form1
private void Program_TextUpdated()	Спрацьовує при події зчитування RFID. Додає у dataGridView1 нові записи об'єктів при зчитуванні. При цьому додає об'єкт з одним кодом тільки один раз. Після додавання треба ще заповнити назву і вибрати склад
private bool IsRecordExists()	Функція, яка використовується у Program_TextUpdated. Перевіряє чи є у dgv запис record у полі RFID
private void Form3_Load()	Спрацьовує при завантаженні форми. Відкриває послідовний порт
private void button1_Click()	Спрацьовує при натисненні кнопки «Додати записи». Додає усі записи з dataGridView1 у базу даних
private void Form3_FormClosing()	Спрацьовує при закритті форми. Закриває послідовний порт
dataGridView1_CellClick()	Спрацьовує при натисненні одну з комірок dataGridView1. Передає стовбці «Ім'я» та «Назва складу» у textBox1 і comboBox1. Передає з строки у якій знаходиться комірка на яку натиснули
textBox1_TextChanged()	Спрацьовує при будь якому змінінні тексту у textBox1. Передає текст з textBox1 у колонку «Ім'я» виділеної строки
comboBox1_SelectedIndexChanged()	Спрацьовує при змінінні значення comboBox1. Передає текст з comboBox1 у колонку «Назва складу» виділеної строки

4.4 Опис реалізації інтерфейсу

Програма використовує стандартний стиль оформлення інтерфейсу. Було створено головну форму програми, від якої можна перейти до інших вікон програми. Цей підхід відповідає загальному принципу побудови інтерфейсу Windows та є дуже поширеним у більшості програм, що робить його зручним для користувачів.

При проектуванні застосунку було використано багато компонентів Visual Studio 2022, які інкапсулюють можливості стандартних компонентів системи Windows (табл. 4.6).

Таблиця 4.6 – Компоненти, використані при побудові інтерфейсу

Компонент	Опис
ComboBox	Компонент, який забезпечує відображення даних у випадяючому полі зі списком
Label	Компонент, який відображає текст на формі
TextBox	Компонент, який забезпечує функції введення та редагування тексту
Button	Компонент, який забезпечує функції кнопки
DataGridView	Компонент, який надає потужний і гнучкий спосіб відображення даних в табличному форматі.
CheckBox	Компонент, який вказує, чи включена яка-небудь умова або вимкнена.
GroupBox	Дозволяє об'єднувати компоненти в одну логічну групу, що також дозволяє їх згрупувати візуально для користувача

4.5 Висновки до розділу

Було розглянуто та описано алгоритм функціонування програми, зазначені та описанні бібліотеки які використовувались, а також основні функції та зміни які є важливими для проекту та його інтерфейсу. Також була описана структура БД, її таблиць та атрибутів.

5 ІНСТРУКЦІЯ З ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Підключення застосунку

Для функціонування розробленого програмного застосунку, а також в разі його допрацювання, необхідно мати таку мінімальну апаратну платформу: персональний комп'ютер з мікропроцесором Intel Pentium 1,8 ГГц та вище, оперативна пам'ять 512 Мб і вище, вільного простору на жорсткому диску 200Мб, монітор, клавіатура та маніпулятор типу «миша». Для зчитування даних з RFID-міток необхідно мати зчитувач радіочастотної ідентифікації FONKAN FM-505, або аналогічний. На всі пристрої, що використовуються в системі потрібні драйвери цих пристроїв. На персональному комп'ютері повинно бути встановлено наступне програмне забезпечення: операційна система Windows 7, Microsoft Access пакету Microsoft Office 2007.

Завантажувальний файл програмного продукту має назву WindowsFormsApp1.exe, для редагування даної програми необхідно відкрити файл WindowsFormsApp1.sln в середовищі Microsoft Visual Studio 2022 та внести необхідні зміни.

База даних, далі БД, програми має назву Database1.accdb, для редагування даної бази необхідно відкрити файл Database1.accdb за допомогою Microsoft Access 2007 пакету Microsoft Office та внести необхідні зміни. Для редагування назв і кількості складів у БД треба змінити їх напряму у файлі Database1.accdb в таблиці Storages.

Для інсталяції програмного продукту необхідно скопіювати на комп'ютер файл RFIDStorageLogger.exe та базу даних Database1.accdb.

За замовченням застосунком використовує COM3 (третій порт) для взаємодії з антенною. Якщо антенна підключена до іншого порту, то треба самостійно замінити порти у коді Form1.cs, Form2.cs, Form3.cs. У випадку

відсутності підключення до порту після будь-якої взаємодії окрім пошуку застосунок буде екстренно завершувати свою роботу.

5.2 Практичне застосування застосунку

Підсумковий застосунок містить у собі 3 форми для відображення і взаємодії з БД, що можна викликати за допомогою натискання на кнопки в формах:

- головна форма програми (Form1.cs);
- форма видалення даних з БД (Form2.cs);
- форма сканування міток для їх подальшого додання на склад (Form3.cs).

Головна форма застосунку, що з'являється при запуску клієнта, показана на рисунку 5.1.

Код	RFID	Item name	Storage Name
1	3000AAC301000000000028AB18E1961B	Худі 023	Склад 1
2	3000AAC3010000000000289F94E1D18B	Парасолька 03	Склад 1
3	3000AAC3010000000000289986E1063A	Роутер Mikrotik	Склад 2
4	3000AAC301000000000028A811E175D3	Модем TP-Link	Склад 2
5	3000AAC301000000000028A532E16437	Шарф 002	Склад 1
6	3000AAC301000000000028A029E1504E	Окуляри 234	Склад 1
7	3000AAC3010000000000289881E1A89D	Сукня 380	Склад 1
8	3000AAC3010000000000289D9AE19CE4	Монітор HP	Склад 2

Рисунок 5.1 – Головна форма програми

Головна форма застосунку має два режими роботи: звичайний та режим пошуку. Активація режиму пошуку залежить від стану відповідного прапорця на головній формі програми.

Під час звичайного режиму роботи можна на головному вікні можна оглянути діючу базу даних, та на формі активні кнопки додавання, та видалення RFID-міток зі складу, але неактивне поле для пошуку об'єктів у БД. Це було зроблене для уникнення можливого конфлікту запитів.

Під час роботи режиму пошуку (рис. 5.2) активується поле для пошуку та деактивуються кнопки додавання та видалення записів у БД.

Склад-контроль

	Код	RFID	Item name	Storage Name
▶	1	3000AAC30100000000028AB18E1961B	Худі 023	Склад 1
	2	3000AAC301000000000289F94E1D18B	Парасолька 03	Склад 1
	3	3000AAC301000000000289986E1063A	Роутер Mikrotik	Склад 2
	4	3000AAC30100000000028A811E175D3	Модем TP-Link	Склад 2
	5	3000AAC30100000000028A532E16437	Шарф 002	Склад 1
	6	3000AAC30100000000028A029E1504E	Окуляри 234	Склад 1
	7	3000AAC301000000000289881E1A89D	Сукня 380	Склад 1
	8	3000AAC301000000000289D9AE19CE4	Монітор HP	Склад 2

Додати запис Видалити записи

☒ Режим пошуку

Пошук

	Код	RFID	Item name	Sto
▶	3	3000AAC301000...	Роутер Mikrotik	2
	4	3000AAC301000...	Модем TP-Link	2
	8	3000AAC301000...	Монітор HP	2
	10	3000AAC301000...	Клавіатура Logit...	2
	11	3000AAC301000...	Світ CISCO	2
	12	3000AAC301000...	Планшет Samsung	2
*				

Пошук

RFID:

☐ Зчитати RFID

Ім'я:

Склад:

Пошук

Рисунок 5.2 – Головна форма програми (режим пошуку)

У режимі пошуку можна виконувати пошук об'єктів за декількома способами. Доступні пошук за назвою об'єкта до якого прикріплена RFID-мітка, за складом, за номером RFID-мітки, що можна ввести самостійно, або за допомогою сканера відсканувати RFID-мітку, що введе її номер у

поле пошуку автоматично. Останнє може використовуватись у випадку якщо мітка відокремитись від товару і треба буде дізнатися від кого саме мітка відокремилась. Для виходу із цього режиму необхідно зняти прапорець в полі «Режим пошуку».

Форма видалення записів з БД зображена на рисунку 5.3.

Відскануйте RFID мітку				
	Код	RFID	Item name	Storage ID
	-1	3000AAC301000...	Новий товар 1	
▶	-2	3000AAC301000...	Новий товар 2	
*				

Below the table is a large grey rectangular area.

At the bottom, there are two buttons: "Видалити відскановане" (Delete scanned) and "Прибрати зі списку" (Remove from list).

Рисунок 5.3 – Форма видалення записів

Під час роботи форми видалення записів сканер постійно сканує навколишній простір на наявність RFID-міток і додає відскановані мітки у поле де можна їх побачити. За допомогою кнопки «Прибрати зі списку» можна прибрати виділені користувачем поля, якщо RFID-мітки були помилково проскановані і користувач не бажає їх видалити з БД. За допомогою кнопки «Видалити відскановане» видаляються усі записи що відображаються у вікні.

Коли запускається форма додання нових міток застосунок приймає усі дані сканера, котрий неперервно сканує доступний простір. Усі відскановані мітки будуть відображені у полі з таблицею. Там їм можна

додати назву та обрати склад куди буде додана мітка. Після завершення редагування полів користувач повинен натиснути кнопку «Додати» для підтвердження введених даних і додання їх до таблиці.

Форма додавання RFID-міток зображена на рисунку 5.4.

RFID	Item_name	Storage name
3000AAC301000000000028AB18E1961B	Новий товар 1	Склад 2
3000AAC3010000000000289F94E1D18B	Новий товар 2	Склад 1
*		

Ім'я:

Склад:

- Склад 1
- Склад 2

Рисунок 5.4 – Форма додання записів

ВИСНОВКИ

В результаті виконання дипломної роботи бакалавра було розроблено програмну систему для ідентифікації об'єктів, яка дозволяє взаємодіяти з RFID-мітками для ідентифікації об'єктів. Було проведено аналіз предметної області і розглянуто відомі методи та засоби ідентифікації об'єктів, їх особливості та структуру. На основі цього аналізу було обрано підходящий для задач застосунку метод ідентифікації об'єктів, а саме RFID.

Розроблено архітектуру системи яка задовольняє вимогам які були виведенні під час аналізу. Обґрунтовано вибір апаратної частини якою став зчитувач радіочастотної ідентифікації FONKAN FM-505, а також мови програмування та середовища розробки якими стали мова програмування C# та середовище Microsoft Visual Studio. Було створено програмне забезпечення, яке виконує наступні функції: розпізнавання RFID-міток, зберігання, пошук та видалення інформації про них з БД.

Практична цінність розробки полягає в простому та ефективному способі ідентифікації та збереження даних про об'єкти Індустрії 4.0, що дозволяє значно полегшити та прискорити їх облік.

У створеного рішення багато можливостей розширення функціоналу від додавання більшої кількості сканерів та комп'ютерів у локальній мережі для обліку у багатьох різних місцях одночасно, до додання історії надходження об'єктів і створення звітів за допомогою програми.

Результати роботи було представлено на щорічних науково-практичних конференціях викладачів, науковців, молодих учених, аспірантів та студентів «Тиждень науки»: «Дослідження методів та засобів розробки систем ідентифікації об'єктів індустрії 4.0» (2024); «Методи та засоби ідентифікації об'єктів на виробництві» (2023); «Методи та засоби авторизації об'єктів в робочих зонах технологічного процесу» (2022).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Скіцько, В. І. Індустрія 4.0 як промислове виробництво майбутнього. Інвестиції: практика та досвід / В.І. Скіцько. – 2016. – № 5. – С. 33–40.
2. Завербний, А. С. Проблеми та перспективи розвитку індустрії 4.0 в Україні за умов євроінтегрування. / А. С. Завербний, К. Р. Сало // SMEU. – 2022. – Випуск 4. № 2. – С. 374–382.
3. Охрімчук, Є. І. Моніторинг ефективності існуючих рішень промислового Інтернету речей / Є. І Охрімчук // Вісник Сумського державного університету. Серія Економіка. – 2022. – № 3. – С. 97–105.
4. Додд, Ф. Що таке Індустрія 4.0? Все, що вам потрібно знати [Електрон. ресурс] / Ф. Додд. – Режим доступу: <https://fiberroad.com/uk/resources/new-trends/what-is-industry-4-0/>.
5. Erboz, G. How to define industry 4.0: main pillars of industry 4.0. Managerial trends in the development of enterprises in globalization era / G. Erboz. – 2017. – P. 761–767.
6. The future of productivity and growth in manufacturing industries / [M. Rüßmann, M. Lorenz, P. Gerbert et al.] // Boston consulting group. – 2015. – Vol. 9. – P. 54–89.
7. Как нам начать четвертую промышленную революцию [Электрон. ресурс]. – Режим доступа: <https://web.archive.org/web/20151001091705/http://www.dsnews.ua/future/kak-nam-nachat-chetvertuyu-promyshlennuyu-revolutsiyu-05092015101200>.
8. Жадько, В. О. Промисловий Інтернет речей / В. О. Жадько, О. А. Кулик // Матеріали Всеукр. наук.-практ. конф. – Ніжин, 2018. – С. 204–208.
9. Tsymbal, O. IoT in Manufacturing: 6 Industrial IoT Trends in 2024 [Electronic resource] / O. Tsymbal. – Access mode: <https://mobidev.biz/blog/industrial-iot-internet-of-things-trends>.
10. Gandhi, Y. What is the Industrial Internet of Things (IIoT)?

Analyticssteps. [Electronic resource] / Y. Gandhi. – Access mode: <https://www.analyticssteps.com/blogs/what-industrial-internet-things-iiot>.

11. Bkheet, S. A. A Review of Identity Methods of Internet of Things (IOT) / S. A. Bkheet, J. I Agbinya. // Advances in Internet of Things. – 2021. – Vol. 11. – P. 153-174.

12. IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey / [M. Burhan, R. A. Rehman, B. Khan et al.] // Sensors. – 2018. – Vol. 18. – 2796 p.

13. Автоматична ідентифікація та збір даних [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/Автоматична_ідентифікація_та_збір_даних.

14. Wood, L. Automatic Identification and Data Capture (Barcodes, Magnetic Stripe Cards, Smart Cards, OCR Systems, RFID Products & Biometric Systems) Market-Global Forecast to 2023 / L. Wood. – Dublin: BusinessWire, 2018. – 133 p.

15. Царьов, Р. Ю. Біометричні технології: навч. посіб. [для вищих навчальних закладів] / Р. Ю. Царьов, Т. М. Лемеха. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – 140 с.

16. Біометрична автентифікація [Електрон. ресурс]. – Режим доступу: https://thereaderwiki.com/uk/Біометрична_автентифікація.

17. Радіочастотна ідентифікація [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Радіочастотна_ідентифікація.

18. Shun, L. C. RFID tag design. [Final Year Project]. – Hong Kong, China, Chinese University of Hong Kong.

19. Застосування систем контролю доступу та розробка макету приладу на основі RFID технології / [А. Ю. Слободчук, В. О. Позняков, К. Л. Ноздрачова та ін.] // Вісник Національного технічного університету «ХПІ». Серія: Електроенергетика та перетворювальна техніка. – Харків : НТУ «ХПІ», 2019. – № 1. – С. 35–47.

20. Maynard, J. Examining the Security and Privacy Concerns of RFID: Part 1 [Electronic resource] / J. Maynard. – Access mode: <https://medium.com/@jack.maynardpartners/examining-the-security-and-privacy-concerns-of-rfid-part-1-c5596c16e08d>.
21. Want, R. An Introduction to RFID Technology. – IEEE pervasive Computing, 2006. – Vol. 5, № 1. – P. 25–33.
22. Phillips, G. How Does RFID Technology Work? [Electronic resource] / G. Phillips. – Access mode: <https://www.makeuseof.com/tag/technology-explained-how-do-rfid-tags-work/>.
23. Sulyok, A. RFID Technology Different Types [Electronic resource] / A. Sulyok. – Access mode: <https://lowrysolutions.com/blog/what-are-the-different-types-of-rfid-technology/>.
24. What are the Different Types of RFID Tags? [Electronic resource]. – Access mode: <https://www.asiarfid.com/different-types-of-rfid-tags>.
25. Shekhar, J. Automated classroom monitoring with iot and virtuino app [Electronic resource] / J. Shekhar. – Access mode: https://www.researchgate.net/publication/332319065_AUTOMATED_CLASSROOM_MONITORING_WITH_IOT_AND_VIRTUINO_APP.
26. Proehl, G. RF Memories and Transceivers, STMicroelectronics Europe B.V. An Introduction to Near Field Communications [Electronic resource] / G. Proehl. – Access mode: <https://eu.mouser.com/applications/rfid-nfc-introduction/>.
27. Foresman, C. Near Field Communications: a technology primer [Electronic resource]. / C. Foresman. – Access mode: <https://arstechnica.com/gadgets/2011/02/near-field-communications-a-technology-primer/>.
28. NFC Standards and NFC Forum RFID4U [Electronic resource]. – Access mode: <https://www.reclamosyquejas.cps.com.pe/?k=nfc-standards-and-nfc-forum-rfid4u-hh-eBajeVSE>.

29. Pelly, N. How to NFC [Electronic resource] / N. Pelly, J. Hamilton. – Access mode: <https://www.youtube.com/watch?v=49L7z3rxz4Q>.
30. NFC as Technology Enabler. NFC Forum [Electronic resource]. – Access mode: https://web.archive.org/web/20131222014541/http://members.nfc-forum.org/aboutnfc/tech_enabler/.
31. Штриховой код [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Штриховой_код.
32. Mirshahi, S. Implementation of structural health monitoring based on rfid and wsn / S. Mirshahi, A. Akbari, S. Uysal // Electrical and Computer Engineering (CCECE). – 2015.
33. Wil, J. Two Dimensional Dot Codes for Product Identification / J. Will, Van Gis // Identification. IEEE Transaction on Information Theory – 1986. – Vol. IT-33, № 5. – P. 620-63.
34. Barcode [Electronic resource]. – Access mode: <https://en.wikipedia.org/wiki/Barcode>.
35. Technical committees. Automatic identification and data capture techniques. – JTC 1/SC 31. – ISO. Retrieved 28 November 2011.
36. Скубак, О. М. Технологія кодування сучасних штрих-кодів / О. М. Скубак, О. А. Мокринцев // Телекомунікаційні та інформаційні технології. – 2016. – № 1. – С. 74–79.
37. Joseph, E. Bar code waveform recognition using peak locations / E. Joseph, T. Pavlidis // IEEE Trans. Pattern Analysis and Machine Intelligence. – June 1994. – Vol. 16, № 6. – P. 630–640.
38. Su, W. A research on integrated application of rfid-based lean manufacturing / W. Su, L. Ma, K. Hu and L. Zhang // Control and Decision Conference. – Chinese: CCDC'09, 2009 . – P. 5781–5784.
39. Pradhan, N. Barcode recognition techniques: review & application. / N. Pradhan, T. Kumar, Dr Rajesh, Ms Pooja Nagpal // International Journal of Innovative Research in Computer Science & Technology (IJIRCS). – 2021.

40. An introduction to QR code technology / International conference on information technology (ICIT). – Tiwari: IEEE, 2016. – P. 39–44.
41. QR Code Essentials [Electronic resource]. – Access mode: <https://www.denso-adc.com/download/qr-code-white-paper/>.
42. Information technology. Automatic identification and data capture techniques. Bar Code symbology : ISO/IEC 18004:2015. – [Effective from 2015-02]. – Geneve: ISO, 2015.
43. Information capacity and versions of QR Code [Electronic resource]. – Access mode: <https://www.qrcode.com/en/about/version.html>.
44. 17 Things Explanation of basic handwriting recognition principles and history [Electronic resource]. – Access mode: <https://web.archive.org/web/20090201055503/http://technology.17things.com/what-is-handwriting-recognition.html>.
45. Дівидюк, С. С. Оптичне розпізнавання символів / С. С. Дівидюк. – Вінниця: ВНТУ, 2016.
46. Системи розпізнавання тексту або OCR-системи. Системи оптичного розпізнавання документів [Електронний ресурс]. – Режим доступу: <http://surl.li/uezny>.
47. Optical Character Recognition (OCR) – How it works [Electronic resource]. – Access mode: <https://web.archive.org/web/20200921035750/https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/>.
48. Порівняння програмного забезпечення для оптичного розпізнавання символів [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Порівняння_програмного_забезпечення_для_оптичного_розпізнавання_символів.
49. Мороз, А.О. Біометричні технології ідентифікації людини. Огляд системи / А.О. Мороз // Мат. машини і системи. – 2011. – №1. – С. 39–45.
50. Різник, О. «Біокон» – система біометричної ідентифікації користувача комп'ютерної мережі / О. Різник, Д. Дзюба, А. Чернодуб //

Системи підтримки прийняття рішень. Теорія і практика: зб. доп. наук.-прак. конф. з міжнар. участю. «СППР 2009». – Київ, 2009. – С. 189–193.

51. The History of Fingerprints [Electronic resource]. – Access mode: <http://onin.com/fp/fphistory.html>

52. Бідюк, П. Сучасні методи біометричної ідентифікації / П. Бідюк, В. Бондарчук // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні : науково-технічний збірник. – 2009. – Вип. 1(18). – С. 137–146.

53. TIOBE Index for May 2024 [Electronic resource]. – Access mode: <https://www.tiobe.com/tiobe-index/>

54. Angella, A. 15 reasons why you should learn C# in 2024 [Electronic resource] / A. Angella. – Access mode: <https://www.linkedin.com/pulse/15-reasons-why-you-should-learn-c-2023-andrea-angella>

55. Fundamentals of garbage collection [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/dotnet/standard/garbage-collection/fundamentals>

56. C# The modern, innovative, open-source programming language for building all your apps [Electronic resource]. – Access mode: <https://dotnet.microsoft.com/en-us/languages/csharp>

57. Visual Studio 2022 [Electronic resource]. – Access mode: <https://visualstudio.microsoft.com/vs/>

58. What is Visual Studio? [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>

ДОДАТОК А
Текст програми

A.1 Фрагмент Program.cs

```

namespace WindowsFormsApp1
{
    internal static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }

    public class MySerialPort : SerialPort
    {
        public delegate void TextUpdatedEventHandler(string newText);
        public static event TextUpdatedEventHandler TextUpdated;
        static void ChangeText(string newText)
        {
            if (string.IsNullOrEmpty(newText) || newText.Length <= 1)
            {
                return;
            }
            newText = newText.Substring(1);
            if (newText.Length == 32)
            {
                TextUpdated?.Invoke(newText);
            }
        }
        static string _receivedData = string.Empty;
        public string outData = string.Empty;

        public MySerialPort(string port): base()
        {
            base.BaudRate = 31250;
            base.DataBits = 8;
            base.StopBits = StopBits.Two;
            base.Parity = Parity.None;
            base.ReadTimeout = 100;
            base.DataReceived += SerialPort_DataReceived;
        }

        public void Open(string portName)
        {
            if (base.IsOpen)
            {
                base.Close();
            }
            base.PortName = portName;
            base.Open();
        }

        void SerialPort_DataReceived(object sender,
        SerialDataReceivedEventArgs e)
    }
}

```

```

    {
        SerialPort sp = (SerialPort)sender;
        string indata = sp.ReadExisting();
        _receivedData += indata;
        if (_receivedData.EndsWith("\n"))
        {
            _receivedData = _receivedData.Trim();
            outData = _receivedData;
            ChangeText(outData);
            _receivedData = string.Empty;
        }
    }
}

```

A.2 Фрагмент Form1.cs

```

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            MySerialPort.TextUpdated += Program_TextUpdated;
        }
        private string last_rfid;
        private void Program_TextUpdated(string newText)
        {
            if (checkBox1.Checked)
            {
                last_rfid = newText;
                if (label1.InvokeRequired)
                {
                    label1.Invoke(new MethodInvoker(delegate { textBox2.Text =
newText; }));
                }
                else
                {
                    textBox2.Text = newText;
                }
            }
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            this.itemsTableAdapter.Fill(this.databases1DataSet1.Items);
            this.storagesTableAdapter.Fill(this.databases1DataSet1.Stora
ges);
            this.dataTable1TableAdapter.Fill(this.databases1DataSet1.Dat
aTable1);
            this.storagesTableAdapter.Fill(this.databases1DataSet1.Stora
ges);
        }
    }
}

```

```

this.dataTable1TableAdapter.Fill(this.databaselDataSet1.DataTabl
e1);
    }
    public MySerialPort port = new MySerialPort("COM3");
    private void button1_Click(object sender, EventArgs e)
    {
        Form2 form2 = new Form2(this);
        form2.Show();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        Form3 form3 = new Form3(this);
        form3.Show();
    }
    public DatabaselDataSet.DataTable1DataTable GetItems()
    {
        return databaselDataSet1.DataTable1;
    }
    public void data_grid_update(string name, string
rfid, string storage, int storage_id)
    {
        DatabaselDataSet.DataTable1Row      newRow      =
databaselDataSet1.DataTable1.NewDataTable1Row();
        newRow.RFID = rfid;
        newRow.Item_name = name;
        newRow.Storage_Name = storage;
        newRow.Storage_ID = storage_id;
        databaselDataSet1.DataTable1.Rows.Add(newRow);
        foreach (DatabaselDataSet.DataTable1Row row in
databaselDataSet1.DataTable1.Rows)
        {
            if (row.RowState == DataRowState.Added)
            {
                DatabaselDataSet.ItemsRow
existingItemRow = databaselDataSet1.Items.FindByКод(row.Код);
                DatabaselDataSet.ItemsRow      itemsRow      =
databaselDataSet1.Items.NewItemsRow();
                itemsRow.RFID = row.RFID;
                itemsRow.Item_name = row.Item_name;
                itemsRow.Storage_ID = row.Storage_ID;
                databaselDataSet1.Items.Rows.Add(itemsRow);
            }
        }
        itemsTableAdapter.Update(databaselDataSet1.Items);
        databaselDataSet1.DataTable1.AcceptChanges();
        dataGridView1.Refresh();
    }
    public void data_grid_delete(string name, string
rfid, string storage)
    {
        itemsTableAdapter.Fill(databaselDataSet1.Items);
        DatabaselDataSet.DataTable1Row      rowToDelete      =
databaselDataSet1.DataTable1.FirstOrDefault(row =>

```

```

        row.RFID == rfid &&
        row.Item_name == name &&
        row.Storage_Name == storage);
    if (rowToDelete != null)
    {
        DatabaselDataSet.ItemsRow itemsRowToDelete
= databaselDataSet1.Items.FirstOrDefault(row =>
    row.RFID == rfid &&
    row.Item_name == name &&
    row.Storage_ID == rowToDelete.Storage_ID);
        if (itemsRowToDelete != null)
        {
            databaselDataSet1.DataTable1.Rows.Remove(rowToDelete);
            itemsRowToDelete.Delete();
            itemsTableAdapter.Update(databaselDataSet1.Items);
            Debug.WriteLine("удалено");
        }
        else
        {
            Debug.WriteLine("Не удалось найти строку в таблице Items");
        }
    }
    else
    {
        Debug.WriteLine("Не удалось найти строку в DataTable1");
    }
}
public void check_uncheck()
{
    checkBox1.CheckState = CheckState.Unchecked;
}
private void checkBox1_CheckedChanged(object sender, EventArgs
e)
{
    if(checkBox1.Checked)
    {
        button1.Enabled = false;
        button2.Enabled = false;
        foreach(Control elem in groupBox1.Controls)
        {
            elem.Enabled = true;
        }
    }
    else
    {
        dataGridView2.Columns.Clear();
        button1.Enabled = true;
        button2.Enabled = true;
        foreach (Control elem in groupBox1.Controls)
        {
            elem.Enabled = false;
        }
    }
}

```

```

    }
    private void checkBox2_CheckedChanged(object
sender, EventArgs e)
    {
        if (checkBox2.Checked)
        {
            port.Open("COM3");
        }
        else
        {
            port.Close();
        }
    }
    private void button3_Click(object sender, EventArgs e)
    {
        DataView dv_search =
databaselDataSet1.DataTable1.DefaultView;
        dv_search.RowFilter = $"RFID LIKE
'#{textBox2.Text}%' AND [Item name] LIKE ' #{textBox1.Text}%' AND
[Storage name] LIKE ' #{comboBox1.Text}%'";
        if (dv_search.Count > 0)
        {
            if (dataGridView2.InvokeRequired)
            {
                dataGridView2.Invoke(new MethodInvoker(delegate {
dataGridView2.DataSource = dv_search.ToTable(); }));
            }
            else
            {
                dataGridView2.DataSource = dv_search.ToTable();
            }
        }
        else
        {
            dataGridView2.DataSource = null;
        }
    }
}
}

```

A.3 Фрагмент Form2.cs

```

namespace WindowsFormsApp1
{
    public partial class Form2 : Form
    {
        private Form1 form1;
        public Form2(Form1 form1)
        {
            InitializeComponent();
            MySerialPort.TextUpdated += Program_TextUpdated;
            this.form1 = form1;
        }
    }
}

```

```

    }

    private string last_rfid
    private void Program_TextUpdated(string newText)
    {
        last_rfid = newText;
        string searchText = last_rfid;
        DataTable dataTable =
(DataTable)dataGridView1.DataSource;
        DataView dv = form1.GetItems().DefaultView;
        dv.RowFilter = $"RFID LIKE '{searchText}%'";
        if (dataGridView1.DataSource == null)
        {
            if (dv.Count > 0)
            {
                if (dataGridView1.InvokeRequired)
                {
                    dataGridView1.Invoke(new MethodInvoker(delegate
{ dataGridView1.DataSource = dv.ToTable(); }));
                }
                else
                {
                    dataGridView1.DataSource = dv.ToTable();
                }
            }
            else
            {
                dataGridView1.DataSource = null;
            }
        }
        else
        {
            if (dv.Count > 0)
            {
                DataTable newData = dv.ToTable();
                DataTable currentData = (DataTable)dataGridView1.DataSource;
                DataView dv_repeat = new DataView(currentData);
                dv_repeat.RowFilter = $"RFID LIKE '{searchText}%'";
                if (dv_repeat.Count <= 0)
                {
                    if (dataGridView1.InvokeRequired)
                    {
                        dataGridView1.Invoke(new MethodInvoker(delegate {
dataGridView1.DataSource = null; }));
                    }
                    else
                    {
                        dataGridView1.DataSource = null;
                    }
                    currentData.Merge(newData);
                    if (dataGridView1.InvokeRequired)
                    {

```

```

        dataGridView1.Invoke(new MethodInvoker(delegate
{ dataGridView1.DataSource = currentData; }));
        }
        else
        {
            dataGridView1.DataSource = currentData;
        }
    }
}

private void button1_Click(object sender, EventArgs e)
{
    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        if (row.Cells[0].Value != null)
        {
            form1.data_grid_delete(row.Cells[2].Value?.ToString(),
row.Cells[1].Value?.ToString(), row.Cells[4].Value?.ToString());
        }
    }
    this.Close();
}

private void Form2_Load(object sender, EventArgs e)
{
    form1.port.Open("COM3");
}

private void button2_Click(object sender, EventArgs e)
{
    if (dataGridView1.SelectedRows.Count > 0)
    {
        foreach (DataGridViewRow row in dataGridView1.SelectedRows)
        {
            if (row.Cells[1].Value != null)
            {
                dataGridView1.Rows.Remove(row);
            }
        }
    }
}

private void Form2_FormClosing(object sender,
FormClosingEventArgs e)
{
    form1.port.Close();
}
}
}

```

A.2 Фрагмент Form1.cs

```

namespace WindowsFormsApp1
{
    public partial class Form3 : Form

```

```

{
    private string last_rfid;
    private Form1 form1;
    public Form3(Form1 form1)
    {
        InitializeComponent();
        MySerialPort.TextUpdated += Program_TextUpdated;
        this.form1 = form1;
    }
    private void Program_TextUpdated(string newText)
    {
        last_rfid = newText;
        string searchText = last_rfid;
        DataView dv = form1.GetItem().DefaultView;
        dv.RowFilter = $"RFID LIKE '{searchText}%'";
        if (dataGridView1.RowCount <= 1)
        {
            Debug.WriteLine(dataGridView1.RowCount);
            if (dv.Count > 0)
            {
                if (dataGridView1.InvokeRequired)
                {
                    dataGridview1.Invoke(new MethodInvoker(delegate {
                        dataGridview1.Rows.Add(last_rfid, "", ""); }));
                }
                else
                {
                    if (dataGridView1.Columns.Count > 0)
                    {
                        dataGridview1.Rows.Add(last_rfid, "", "");
                    }
                    else
                    {
                        dataGridview1.Columns.Clear();
                        dataGridview1.Columns.Add("RFID", "RFID");
                        dataGridview1.Columns.Add("Item_name", "Item name");
                        dataGridview1.Columns.Add("Storage_name", "Storage name");
                        dataGridview1.Rows.Add(last_rfid, "", "");
                    }
                }
            }
        }
        else
        {
            if (dv.Count > 0)
            {
                if (!IsRecordExists(dataGridView1, searchText))
                {
                    if (dataGridView1.InvokeRequired)
                    {
                        dataGridview1.Invoke(new MethodInvoker(delegate {
                            dataGridview1.Rows.Add(last_rfid, "", ""); }));
                    }
                }
            }
        }
    }
}

```

```

        }
        else
        {
            if (dataGridView1.Columns.Count > 0)
            {
dataGridView1.Rows.Add(last_rfid, "", "");
            }
            else
            {
                dataGridView1.Columns.Clear();
                dataGridView1.Columns.Add("RFID", "RFID");
                dataGridView1.Columns.Add("Item_name", "Item name");
                dataGridView1.Columns.Add("Storage_name", "Storage name");
                dataGridView1.Rows.Add(last_rfid, "", "");
            }
        }
    }
}

private bool IsRecordExists(DataGridView dgv, string record)
{
    foreach (DataGridViewRow row in dgv.Rows)
    {
        if (row.Cells["RFID"].Value != null)
        {
            if (row.Cells["RFID"].Value.ToString().Equals(record))
            {
                return true;
            }
        }
    }
    return false;
}

private void Form3_Load(object sender, EventArgs e)
{
    this.storagesTableAdapter.Fill(this.databases1DataSet1.Storages);
    this.storagesTableAdapter.Fill(this.databases1DataSet1.Storages);
    this.storagesTableAdapter.Fill(this.databases1DataSet1.Storages);
    form1.port.Open("COM3");
    dataGridView1.DataSource = null;
}

private void button1_Click(object sender, EventArgs e)
{
    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        if (row.Cells[0].Value != null)
        {
            form1.data_grid_update(row.Cells[1].Value?.ToString(),
            row.Cells[0].Value?.ToString(), row.Cells[2].Value?.ToString(),
            (int)row.Cells[3]?.Value);
        }
    }
}

```

```

        this.Close();
    }
    private void Form3_FormClosing(object sender,
FormClosingEventArgs e)
    {
        form1.port.Close();
    }
    private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
    {
        if (e.RowIndex >= 0)
        {
            string name =
dataGridView1.Rows[e.RowIndex].Cells[1].Value?.ToString();
            string storage =
dataGridView1.Rows[e.RowIndex].Cells[2].Value?.ToString();
            textBox1.Text = name;
            comboBox1.Text = storage;
        }
    }
    private void textBox1_TextChanged(object sender,
EventArgs e)
    {
        if (dataGridView1.SelectedCells.Count > 0)
        {
            dataGridView1.SelectedCells[1].Value = textBox1.Text;
        }
    }
    private void comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
    {
        if (dataGridView1.SelectedCells.Count > 0)
        {
            dataGridView1.SelectedCells[2].Value = comboBox1.Text;
            dataGridView1.SelectedCells[3].Value = comboBox1.SelectedValue;
        }
    }
}
}

```

ДОДАТОК Б
Слайди презентації

1

2024

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ ІНДУСТРІЇ 4.0

НУ «Запорізька політехніка»
Дипломна кваліфікаційна робота бакалавра

Студентка КНТ-111сп: Антоніна ШИМБОРЕЦЬКА
Керівник: к.т.н., доцент Анжеліка ПАРХОМЕНКО

Рисунок Б.1 – Слайд 1

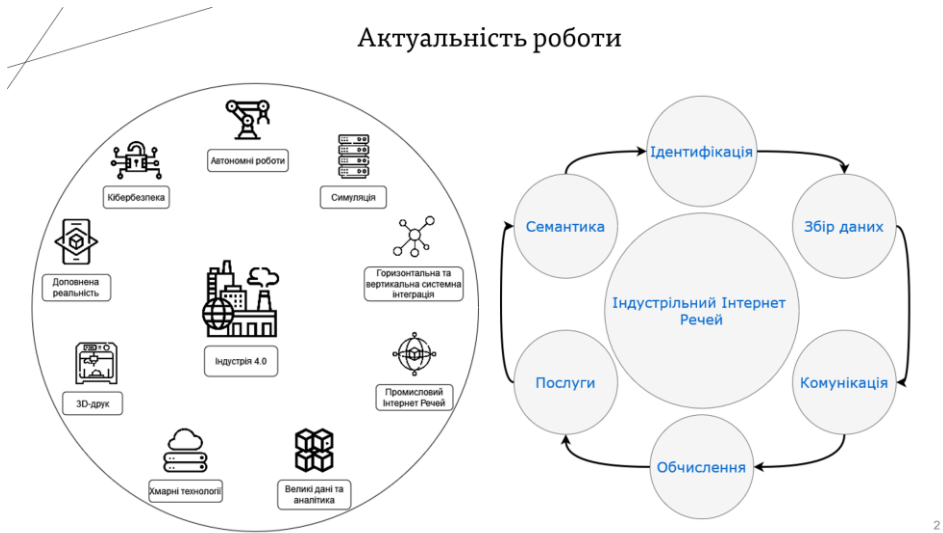


Рисунок Б.2 – Слайд 2

Мета та задачі дипломної роботи

Мета роботи – дослідження та розробка програмно-апаратних рішень для взаємодії з RFID-мітками для ідентифікації об'єктів Індустрії 4.0.

Задачі роботи:

- провести дослідження існуючих методів та засобів ідентифікації об'єктів;
- здійснити проектування програмного забезпечення;
- створити програмне забезпечення для ідентифікації об'єктів;
- виконати тестування розробленого програмного забезпечення для перевірки його ефективності.

3

Рисунок Б.3 – Слайд 3

Огляд існуючих методів автоматичної ідентифікації та збору даних



4

Рисунок Б.4 – Слайд 4

Дослідження RFID технологій



5

Рисунок Б.5 – Слайд 5

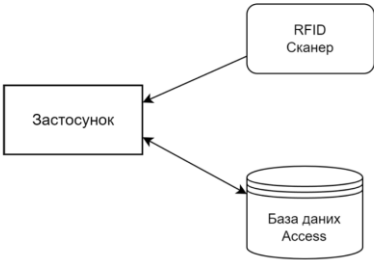
Схема функціонування застосунку



6

Рисунок Б.6 – Слайд 6

Архітектура системи



7

Рисунок Б.7 – Слайд 7

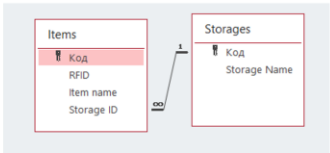
Використані бібліотеки

Бібліотека	Опис бібліотеки
System.IO.Ports	Надає класи та функції для роботи з послідовними портами (COM-портами)
System.Windows.Forms	Надає засоби для розробки форм, діалогових вікон та інших елементів інтерфейсу
System.Data	Забезпечує можливість роботи з базами даних і джерелами даних
System	Містить фундаментальні класи та базові типи, використовувані у всіх інших просторах імен

8

Рисунок Б.8 – Слайд 8

Структура БД



Таблиця	Атрибут	Тип	Опис
Items	Код	Лічильник	Унікальний ідентифікатор
	RFID	Короткий текст	Код RFID-мітки
	Item name	Короткий текст	Назва об'єкту
	Storage ID	Числовий	Зовнішній ключ на таблицю Items
Storages	Код	Лічильник	Унікальний ідентифікатор
	Storage Name	Короткий текст	Назва складу

9

Рисунок Б.9 – Слайд 9

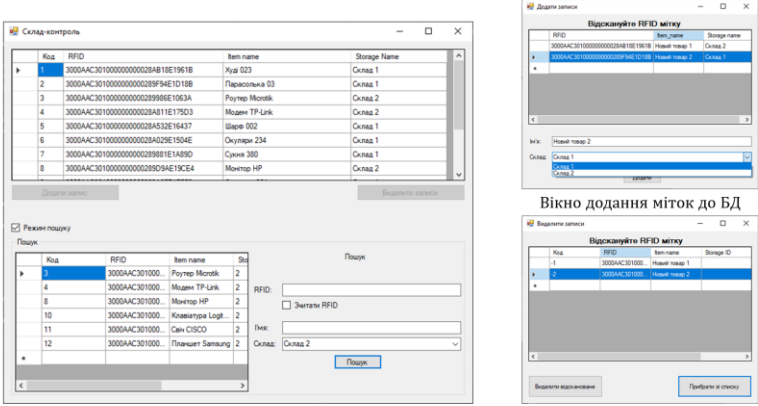
Обґрунтування вибору мови програмування

Характеристика	Мова програмування		
	C#	C++	Java
Наявність стандартних бібліотек для роботи з портами	+	-	-
Підтримка властивостей	+	-	-
Множинне наслідування класів	-	+	-
Швидкодія	+-	+	+-
Збирач сміття	+	-	-
Відкритий код	+	-	+

10

Рисунок Б.10 – Слайд 10

Інтерфейс програми



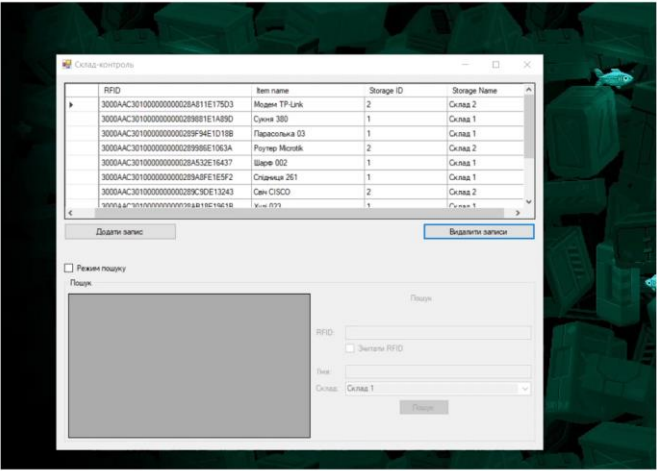
Головне вікно програми

Вікно видалення міток з БД

11

Рисунок Б.11 – Слайд 11

Демонстрація роботи програми



12

Рисунок Б.12 – Слайд 12



Висновки

В результаті виконання дипломної роботи бакалавра було розроблено програмну систему для ідентифікації об'єктів, яка дозволяє взаємодіяти з RFID-мітками для ідентифікації об'єктів.

Було проведено аналіз предметної області і розглянуто відомі методи та засоби ідентифікації об'єктів, їх особливості та структуру. На основі цього аналізу було обрано підходящий для задач застосування метод ідентифікації об'єктів.

Було створено програмне забезпечення, яке виконує наступні функції: розпізнавання RFID-міток, зберігання, пошук та видалення інформації про них з БД.

Практична цінність розробки полягає в простому та ефективному способі ідентифікації та збереження даних про об'єкти Індустрії 4.0, що дозволяє значно полегшити та прискорити їх облік.

Результати роботи доповідалися на щорічних науково-практичних конференціях викладачів, науковців, молодих учених, аспірантів та студентів «Тиждень науки»:

- ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ РОЗРОБКИ СИСТЕМ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ ІНДУСТРІЇ 4.0 (2024);
- МЕТОДИ ТА ЗАСОБИ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ НА ВИРОБНИЦТВІ (2023);
- МЕТОДИ ТА ЗАСОБИ АВТОРИЗАЦІЇ ОБ'ЄКТІВ В РОБОЧИХ ЗОНАХ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ (2022).

13

Рисунок Б.13 – Слайд 13