

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Розроблення програмного модуля автоматизованої системи
моніторингу стану роботів на базі сенсорної технології
(тема)

Виконала:
студентка 2 курсу, групи КТРСм-21-1
Стеценко К.В.
(прізвище, ініціали)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми Освітньо-професійна

Освітня програма Комп'ютеризовані та
робототехнічні системи
(повна назва освітньої програми)

Керівник проф. Филипенко О.І.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТАМ _____
(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри Невлюдов І.Ш

«_____» _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Стеценко Катерині Вадимівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення програмного модуля автоматизованої системи моніторингу стану роботів на базі сенсорної технології

затверджена наказом по університету від 07.11. 2022 р. № 1462 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14.12. 2022 р.

3. Вихідні дані до роботи Спосіб визначення стану промислового обладнання: за допомогою сенсорної технології, датчиків, що розміщуються на засобах автоматизації; організація зв'язку з користувачем: з використанням технології чат-ботів та API Telegram; період зчитування даних за допомогою сенсорів: не рідше 10 с.

4. Перелік питань, що потрібно опрацювати в роботі

Вступ. Аналіз предметної області та технічного завдання. Розробка автоматизованої системи. Розроблення методу моніторингу стану роботів на базі сенсорної технології. Розробка структури бази даних. Експериментальні дослідження. Висновки. Перелік джерел посилання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 15 с., формату А4

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на виконання кваліфікаційної роботи	15.09.2022	Виконано
2	Визначення актуальності роботи, мети, предмета, об'єкта, та розробка завдань для досягнення мети	20.09 – 30.09.22	Виконано
3	Аналіз літературних джерел	01.10 – 15.10.22	Виконано
4	Вибір середовища розробки	16.10 – 5.11.22	Виконано
5	Оформлення пояснювальної записки та документації	06.11 – 20.11.22	Виконано
6	Оформлення додатків	21.11 – 01.12.2	Виконано
7	Оформлення графічного та презентаційних матеріалів комп'ютерного захисту	01.12 – 06.12.22	Виконано
8	Представлення роботи на рецензування	10.12 – 12.12.22	Виконано

Дата видачі завдання 15.09.2022

Студентка

(підпис)

Керівник роботи

(підпис)

Стеценко К.В.

(прізвище, ініціали)

проф. Филипенко О.І.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 103 с., 8 табл., 33 рис., 3 дод., 30 джерел.

АВТОМАТИЗОВАНА СИСТЕМА, МОНІТОРИНГ СТАНУ, ДАТЧИКИ,
СЕНСОРНА ТЕХНОЛОГІЯ, РОБОТ, РОЗРОБКА .

Мета роботи – розробка функціонального, математичного та алгоритмічно-програмного забезпечення автоматизованої системи моніторингу стану роботів на базі сенсорної технології.

Об’єкт дослідження – моніторинг стану роботів.

Предмет дослідження – програмний модуль автоматизованої системи моніторингу стану роботів.

Методи дослідження – логічний та системний аналіз, теорія алгоритмів, методи і засоби збору та обробки даних.

Для досягнення поставленої мети було вирішено наступні завдання: проведено аналіз технічного завдання та предметної області; розглянуто класифікацію автоматизованих систем; моніторинг стану технологічного обладнання, переваги використання сенсорних технологій. При розробці програмного забезпечення було зроблене математичне забезпечення: до кожного датчика у програмі проводився розрахунок низки параметрів сигналу. Зроблено схеми та алгоритми роботи метода, реалізована база даних та зроблено експериментальне дослідження.

ABSTRACT

Explanatory note: 103 p., 8 tab., 33 fig., 3 adj., 30 sources.

AUTOMATED SYSTEM, STAND MONITORING, SENSORS, SENSOR TECHNOLOGY, ROBOT, DEVELOPMENT.

The purpose of the work is the development of functional, mathematical and algorithmic software of an automated system for monitoring the condition of robots based on sensor technology. The object of research is the monitoring of the condition of robots.

The subject of research is the software module of the automated system monitoring the status of robots.

Research methods – logical and system analysis, theory of algorithms, methods and means of data collection and processing.

In order to achieve the set goal, the following tasks were solved: the analysis of the technical task and subject area was carried out; the classification of automated systems is considered; monitoring the condition of technological equipment, advantages of using sensor technologies. During the development of the software, mathematical support was made: a number of signal parameters were calculated for each sensor in the program. Schemes and algorithms of the method were made, a database was implemented and an experimental study was carried out.

ПЕРЕЛІК СКОРОЧЕНЬ

- АІПС – автоматизована інформаційно-пошукова система;
- АС – автоматизована система;
- АСІТО – автоматизована система інформаційно-термінологічного обслуговування;
- АСК – автоматизовані системи керування;
- АСК ТП – автоматизовані системи керування технологічними процесами;
- АСКВ – автоматизовані системи керування виробництвом;
- АСКП – автоматизовані системи керування підприємствами;
- АСНД – автоматизовані системи наукових досліджень;
- АСТПВ – автоматизована система технологічної підготовки виробництва;
- БКС – блок контролю стану;
- ПЕОМ – персональна електроннообчислювальна машина
- ПК – персональний комп'ютер;
- ПНВЕС – перетворювач неелектричних величин в електричні сигнали;
- САП – системи автоматизованого проектування;
- САПР ТП – системи автоматизованого проектування технологічних процесів;
- САПР – системи автоматизованого проектування і розрахунку;
- ЧЕ – чутливий елемент;
- ІТ – information technology.

ЗМІСТ

Перелік скорочень.....	6
Вступ.....	10
1 Аналіз предметної області та технічного завдання.....	11
1.1 Автоматизована система моніторингу.....	11
1.1.1 Класифікація автоматизованих систем.....	11
1.2 Моніторинг стану технологічного обладнання.....	13
1.2.1 Визначення терміну «Моніторинг стану».....	13
1.2.2 Актуальність стану моніторингу технологічного обладнання.....	14
1.2.3 Недоліки моніторингу стану технологічного обладнання.....	15
1.3 Відстеження параметрів робота в режимі реального часу за допомогою датчиків.....	15
1.4 Переваги використання сенсорних технологій.....	18
1.5 Висновки до першого розділу.....	19
2. Розробка автоматизованої системи.....	20
2.1 Функціональне забезпечення АС.....	20
2.1.1 Визначення вимог до датчиків системи.....	24
2.1.2 Опис датчиків та технічні характеристики.....	24
2.2. Інформаційне забезпечення АС.....	31
2.2.1 Вибір мови програмування.....	32
2.3 Висновки до другого розділу.....	35
3 Розроблення методу моніторингу стану роботів на базі сенсорної технології.....	36
3.1 Опис технології автоматизованого збирання даних з механізмів промислових роботів.....	36
3.2 Розроблення методу моніторингу стану роботів на базі сенсорної технології.....	39

3.3 Висновки до третього розділу.....	43
4 Розробка структури бази даних.....	44
4.1 Аналіз даних та розробка моделі структури бази даних.....	44
4.2 Висновки до четвертого розділу.....	49
5 Експериментальні дослідження.....	50
5.1 Постановка задачі для проведення експерименту.....	50
5.2 Розробка алгоритму роботи програми.....	53
5.3 Розробка програмного засобу для виконання експериментальних досліджень.....	55
5.4 Опис результатів проведення експерименту.....	61
5.5 Висновки до п'ятого розділу.....	66
6 Охорона праці.....	67
6.1 Значення охорони праці у роботі фахівців з комп'ютерних технологій.....	67
6.2 Забезпечення безпеки життєдіяльності при роботі з ПК.....	68
6.3 Висновки до шостого розділу.....	70
Висновки.....	71
Перелік джерел посилання.....	72
Додаток А Лістинг коду програмного модуля на C#.....	76
Додаток Б Демонстраційний матеріал.....	95
Додаток В Відомість кваліфікаційної роботи.....	103

ВСТУП

Системи моніторингу стану створюють передумови для раннього виявлення несправностей у машинах та устаткуванні, а також цілеспрямованого планування робіт з технічного обслуговування для скорочення часу простоїв до мінімуму.

В міру того, як ускладнюється продукт та виробничі процеси, збільшується ймовірність людських помилок при обробці матеріалів та зборі даних. Автоматизація виробничих процесів запобігає виникненню подібних ситуацій. Інтеграція інноваційного обладнання та автоматизація підприємства забезпечують засоби точного та своєчасного збору даних, управління процесами та оптимізації роботи всього виробництва.

Більшість сучасних підприємств, корпорацій та установ мають власні веб-портали, на яких розміщується конкретна інформація. Однак є проблеми неоднорідності даних і дуже низька якість, низький час відгуку сервісу та висока вартість тому робота є актуальною.

Мета роботи – розробка функціонального, математичного та алгоритмічно-програмного забезпечення автоматизованої системи моніторингу стану роботів на базі сенсорної технології.

Об'єкт дослідження – моніторинг стану роботів.

Предмет дослідження – програмний модуль автоматизованої системи моніторингу стану роботів.

Методи дослідження – логічний та системний аналіз, теорія алгоритмів, методи і засоби збору та обробки даних.

При розробці програмного модулю в математичному забезпеченні: до кожного датчика у програмі проводиться розрахунок низки параметрів сигналу.

Зроблено схеми та алгоритми роботи метода, реалізована база даних та зроблено експериментальне дослідження.

Розроблений метод може використовуватися на різного роду виробництвах, де виникає необхідність автоматизованого моніторингу в режимі реального часу без значних витрат на реалізацію програмного забезпечення, а також може бути використаний під час проведення практичних і лабораторних практикумів в технологічних і навчальних лабораторіях в межах спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології та бути корисним для фахівців, що пов'язані з промисловою автоматизацією.

Робота виконана згідно [1-4]. Під час навчання на другому (магістерському) рівні матеріали, що були використані для написання атестаційної роботи було апробовано та висвітлено в статтях [5-7].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Автоматизована система моніторингу

Автоматизована система моніторингу виробничого процесу є ефективним засобом, що дозволяє оперативно, точно та адекватно оцінювати та аналізувати поточну ситуацію, приймати обґрунтовані та своєчасні управлінські рішення [8].

Основні функції автоматизованої системи моніторингу:

- контролює стан основних, допоміжних та забезпечують технологічних агрегатів, установок, виробничих ліній та транспортних систем та механізмів;
- візуалізація стану технологічного процесу в підрозділах підприємства, розрахунок та подання ключових показників виробництва;
- оперативне формування виробничої звітності: рапорт виробництва по агрегатах, цехах, рапорт відвантаження, стеження за виконанням замовлень, моніторинг технології та якості, моніторинг складів;
- врахування часу роботи агрегатів. Контроль та реєстрація часу та причин простоїв, розрахунок погодинних та сумарних простоїв. Формування рапорту простоїв з цехів, причин, служб.

1.1.1 Класифікація автоматизованих систем

Автоматизована виробнича система здійснює збирання інформації з об'єкта керування, передає, перетворює й обробляє її, формує керуючі команди та виконує їх на керованому об'єкті, тобто ті функції, які піддаються автоматизації.

Людина визначає цілі та критерії керування й коригує їх, коли змінюються умови, зокрема, виконує функції нагляду за роботою

автоматизованих пристроїв, а в разі потреби змінює програму їхньої роботи (завдання) і приймає загальні рішення щодо керування в змінених або складних ситуаціях [9].

Автоматизована система у інформаційних технологіях – система, що реалізує інформаційну технологію виконання встановлених функцій за допомогою персоналу і комплексу засобів автоматизації.

Залежно від виду діяльності розрізняють такі різновиди АС:

- АСК (автоматизовані системи керування), які у свою чергу в залежності від виду об'єкту керування поділяються на: АСК (автоматизовані системи керування) технологічними процесами (АСК ТП); АСК підприємствами (АСКП), виробництвом (АСКВ) тощо;

- системи автоматизованого проектування: САПР (системи автоматизованого проектування і розрахунку); САПР ТП (системи автоматизованого проектування технологічних процесів) тощо;

- АСНД (автоматизовані системи наукових досліджень);

- АС оброблення та передавання інформації: АПС (автоматизована інформаційно-пошукова система); АСІТО (автоматизована система інформаційно-термінологічного обслуговування) тощо;

- АСТПВ (АС технологічної підготовки виробництва);

- автоматизовані системи контролю та випробувань;

- АС, що поєднують функції, перелічених вище систем.

АС реалізують інформаційну технологію у вигляді певної послідовності інформаційно пов'язаних функцій, завдань або процедур, що виконуються в автоматизованому (інтерактивному) або автоматичному режимах.

1.2 Моніторинг стану технологічного обладнання

1.2.1 Визначення терміну «Моніторинг стану»

Для підвищення надійності обладнання, зниження аварійних ситуацій, що у свою чергу підвищує прибуток підприємства, потрібний моніторинг стану технологічного обладнання в тому числі і роботів.

Різні типи підприємств потребують різних рішень щодо моніторингу стану технологічного обладнання, але незалежно від типу підприємства отримання оперативної інформації про стан є істотно важливим для планування та проведення ремонтних робіт.

Моніторинг стану – це ряд технічно організаційних заходів, що дозволяють на основі певного набору вимірних даних (наприклад, вібрації, температури і т. д.) визначати несправність, що розвивається [10].

Без систем моніторингу стану обладнання неможливо перейти від планово-попереджувальних ремонтів до обслуговування за станом. Моніторинг стану дозволяє планувати технічне обслуговування та вживати превентивних дій для запобігання подальшій відмові та подальшому незапланованому простою.

Переваги моніторингу стану технологічного обладнання:

- скоротити час та гроші на технічне обслуговування та ремонт;
- продовжити термін безвідмовної роботи обладнання;
- підвищити рівень безпеки персоналу;
- збільшити прибутки;
- підвищити надійність підприємства загалом.

Моніторинг стану це перевірка стану обладнання у нормальних умовах експлуатації. До нього входять збирання та обробка даних, а також зіставлення даних з тенденціями, вихідними даними та репрезентативними даними, отриманими з аналогічного обладнання. Тобто перевіряється та діагностується

стан обладнання під час його роботи [11].

Основне завдання моніторингу стану – визначити відхилення від нормальної роботи.

Успішне використання різних методів моніторингу стану дозволяє прогнозувати, які компоненти можуть вийти з ладу найближчим часом.

1.2.2 Актуальність стану моніторингу технологічного обладнання

В системі моніторингу стану можна встановити оповіщення для конкретного компонента, коли він досягає певного рівня показань, що, у свою чергу, дозволяє обслуговуючого персоналу знати, що вони можуть розглянути можливість заміни цього компонента.

Моніторинг стану також може працювати в режимі онлайн, тому датчики та програмне забезпечення з підключенням до Інтернету та бездротовим зв'язком забезпечують цілодобові вимірювання в режимі реального часу.

Ці виміри передаються інтегрованому програмному забезпеченню для аналізу та зберігання наборів історичних даних або виявлення своєчасної поломки обладнання робота.

Інструменти моніторингу стану, які видають дані в режимі реального часу, дозволяють швидше визначити основну причину проблеми, а бездротові датчики на активах автоматично підключають співробітників до даних реального часу через віддалений доступ за допомогою смартфонів або планшетів [12].

Моніторинг стану дає змогу підприємству перейти від реактивного підходу до програми профілактичного обслуговування.

Моніторинг стану в режимі 24/7 надає чітку картину стану робота без додаткових трудовитрат.

1.2.3 Недоліки моніторингу стану технологічного обладнання

Системи моніторингу стану покладаються на візуальні дані, зібрані з кількох датчиків, інтегрованих у програмну систему.

Це означає додаткові витрати на придбання та встановлення цих датчиків, а також на придбання інструментів, необхідних для моніторингу стану (аналіз вібрації, інфрачервоні термографи тощо).

Є також додаткові витрати на навчання співробітників точного та ефективного використання технології моніторингу стану.

Крім того, датчики моніторингу стану можуть не працювати належним чином за особливо важких умов експлуатації.

Такі умови можуть пошкодити датчики, що змусить замінювати їх частіше, ніж очікувалося [13].

1.3 Відстеження параметрів робота в режимі реального часу за допомогою давачів

Датчиком або сенсором називається інформаційний пристрій, що перетворює контрольований фізичний параметр в сигнал, зручний для подальшої обробки в каналі вимірювальної системи.

Досить часто зустрічаються і інші назви датчиків: вимірювальний перетворювач, давач, детектор, вимірювач, чутливий елемент, зонд, сенсор, рецептор.

Датчик (рис.1.1) включає в себе чутливий елемент і перетворювач неелектричних величин в електричні сигнали.

Часто між чутливим елементом і перетворювачем включається передавально-розмножувальний механізм, необхідний для підвищення чутливості датчика.

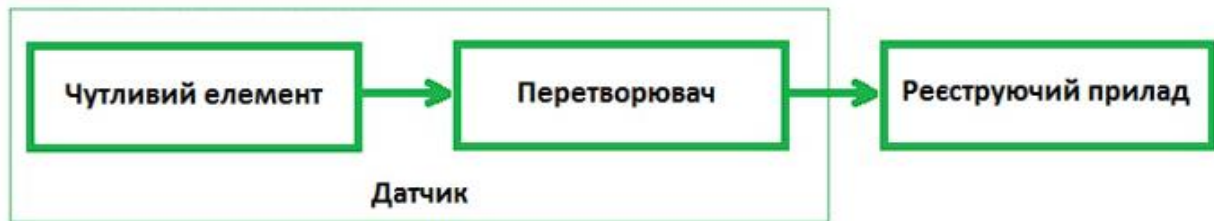


Рисунок 1.1 – Загальна схема датчика [14]

Області використання сенсорів дуже різноманітні: промислова техніка, вимірювання і регулювання, робототехніка, автомобілебудування, медична техніка [15].

Датчик обов'язково містить у своїй конструкції чутливий елемент і часто – перетворювальну частину.

Давачі є елементами технічних систем, призначених для вимірювання, сигналізації, регулювання, керування приладами та процесами. Давачі перетворюють величину, яка контролюється (тиск, температура, витрата, концентрація, частота, швидкість, переміщення, електрична напруга, електричний струм тощо) у сигнал (електричний, оптичний, пневматичний), зручний для вимірювання, передавання, перетворення, зберігання та реєстрації інформації про стан об'єкта вимірювання [16].

Останнім часом стосовно давачів застосовуються терміни: «багатофункційний давач» чи «інтелектуальний давач», що відбиває напрямки розвитку сучасних давачів [17]. Під цими термінами, крім функцій первинного вимірювального перетворення, мають на увазі додаткові можливості вимірювання декількох фізичних величин та використання вбудованих аналого-цифрових перетворювачів з мікроконтролерами, що суттєво розширює функціональний діапазон давачів, а саме:

- попередня обробка сигналів (лінеаризація, фільтрування, корекція похибок);
- само-діагностування;

- дистанційне конфігурування (діапазону вимірювань, одиниць вимірювань, узгодження частотних характеристик);
- окремі елементи керування;
- передавання інформації з використанням промислових мереж Profibus, Ineterbus, Profinet та інших.

Розглянемо класифікацію сенсорів.

За принципом перетворення енергії розрізняють активні та пасивні давачі, що відрізняються способами формування сигналу та схемами підключення:

- активні (генераторні) – давачі, у яких здійснюється перетворення видів енергії від входу до виходу;
- пасивні (параметричні) – давачі, у котрих вхідна енергія змінює параметри визначених елементів первинних вимірювальних перетворювачів.

За вхідними фізичними величинами, що підлягають перетворенню давачі бувають:

- електричні та магнітні;
- теплових величин;
- механічних величин;
- оптичних параметрів;
- форми та розмірів;
- акустичних величин;
- концентрації та складу;
- іонізаційного випромінення.

За фізико-хімічними ефектами, що лежать в основі роботи вимірювальних перетворювачів, розрізняють давачі:

- резистивні (потенціометричні, терморезистивні, тензорезистивні);
- ємнісні (електростатичні);
- індуктивні та електромагнітні;
- електричного заряду, напруги або струму;

- зміни геометричних розмірів, маси або положення;
- оптичних ефектів;
- біохімічні.

За видом вихідного сигналу давачі бувають:

- дискретні;
- аналогові;
- цифрові;
- імпульсні.

За фізичною природою вихідного сигналу давачі бувають:

- з електричним вихідним сигналом (найпоширеніші);
- пневматичним вихідним сигналом;
- оптичним вихідним сигналом (перспективні).

1.4 Переваги використання сенсорних технологій

Розгортання бездротових сенсорних мереж доцільно, передусім, там, де використання традиційних дротових технологій створює труднощі підключення сенсорних вузлів [18].

Перелік проблемних ситуацій включає:

- необхідність розміщення датчиків у важкодоступних для людини місцях;
- наявність віддалених або географічно розподілених датчиків, відстані до яких від центрального процесора або логера/маршрутизатора є значними (десятки та сотні метрів чи кілометри);
- розміщення проводів порушує нормальну роботу машин та споруд;
- присутність в електроніці, що носить людину, особливо імплантованих датчиків;
- середовище, в якому повинен працювати датчик, виключає або ускладнює електричне підключення за допомогою дротів (вода, екстремальні

температури тощо);

- мережеві мультисенсорні вузли безальтернативно вимагають взаємодії між собою та надмірної кількості сигнальних проводів (на додаток до проводів живлення);

- провід значної протяжності створюють перешкоди для проходження сигналу плюс перешкоди по ланцюгах живлення;

- застосування дротів обмежено форм-фактором датчика;

- системна ціна/вартість проводів обмежена.

1.5 Висновки до першого розділу

В результаті виконання першого розділу магістерської кваліфікаційної роботи були виконанні наступні завдання:

- проведено аналіз предметної області та технічного завдання;

- основні функції автоматизованої системи для моніторингу;

- розглянуто класифікацію автоматизованих систем;

- актуальність моніторингу стану та його недоліки;

- особливості давачів та переваги використання сенсорних технологій.

2 РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ

Основним завданням використання автоматизованої системи є підвищення ефективності роботи. Використання автоматизованої системи дає можливість швидко збирати та обробляти інформацію, що у свою чергу дає можливість підвищити точність роботи системи та мінімізувати втручання людини у прийняття рішень. Використання автоматизованої системи значно підвищує рівень контролю та дисципліни, знижуючи при цьому витрати.

2.1 Функціональне забезпечення АС

Під функцією автоматизованої системи розумітимемо сукупність її дій, вкладених у досягнення певної приватної мети управління [19].

Автоматизована система виробничого призначення здійснює збирання інформації з об'єкта керування, передає, перетворює й обробляє її, формує керуючі команди та виконує їх на керованому об'єкті, тобто ті функції, які піддаються автоматизації. Людина визначає цілі та критерії керування й коригує їх, коли змінюються умови, зокрема, виконує функції нагляду за роботою автоматизованих пристроїв, а в разі потреби змінює програму їхньої роботи і приймає загальні рішення щодо керування в змінених або складних ситуаціях. Автоматизовані системні операції це набір програмного та апаратного забезпечення, який дозволяє комп'ютерним системам, мережевим пристроям або машинам функціонувати без ручного втручання. тобто без оператора-людини, фізично розташованого на місці, де встановлено систему.

Автоматизована система за допомогою датчиків здійснює збір інформації з об'єкта керування, передає отриману інформацію по мережі на сервер, сервер приймає показання та обробляє їх [20]. Залежно від обробки отриманих параметрів – надсилає повідомлення в телеграм-бот людині-оператору, що

параметри виміру вийшли за задані межі. Система буде надсилати повідомлення до оператора о відхиленнях.

Отже, для роботи системи необхідні три об'єкти взаємодії: частини робота, на яких закріплені датчики, сервер для взаємодії між роботом та ботом, та бот у телеграмі.

Виходячи з аналізу літератури та розробки схеми організації зв'язку з промисловим обладнанням для моніторингу його стану розробимо схему автоматизованої системи моніторингу стану робіт на базі сенсорної технології.

Функціонально автоматизована система включають автоматизацію всіх підсистем, функцій і комплексів задач. Як і будь-яка складна система, автоматизована система складається з елементів – підсистем, які можна розглядати як підсистеми нижчого рівня, що також складаються з елементів.

Функціональна підсистема – це частина АС, виділена за спільністю функціональних ознак управління.

Інформаційне забезпечення є сукупністю єдиної системи класифікації та уніфікованих систем документації і масивів інформації, що використовуються в автоматизованій системі.

Математичне забезпечення – сукупність математичних методів, моделей та алгоритмів оброблення інформації в автоматизованій системі.

Програмне забезпечення – сукупність програм, які реалізують мету та задачі автоматизованої системи і забезпечують функціонування комплексу технічних засобів системи.

Технічне забезпечення – комплекс технічних засобів, які забезпечують роботу автоматизованої системи.

Розроблена схема автоматизованої системи показана на рисунку 2.1.

Основними функціями системи є:

- отримання даних від сенсорів (f_1);
- аналіз даних отриманих даних (f_2);

– відображення даних засобами мобільних носимих пристроїв (f_3).

Інформаційна модель даних представлена в автоматизованій системі СУБД SQLite (i_1);. Посередником між програмної підсистемою та інформаційної є програмна модель даних у вигляді відповідних класів.

Моделі обробки даних описана в підрозділі 3.1 та полягає в:

- первинній обробці показань з сенсорів (m_1);
- фільтрації даних (m_2).

Програмний комплекс поєднує такі підсистеми:

- сервер збору даних (p_1);
- чат-бот для взаємодії з користувачем (p_2);
- API Telegram (p_3).

Технічний комплекс включає такі модулі:

- набір сенсорів, що розташовані на механізмах та пристроях (t_1);
- модуль збирання даних, що передає отриману інформацію до серверу баз даних (t_2);
- сервер віддаленого моніторингу, що побудовано із застосуванням програмного забезпечення (t_3).

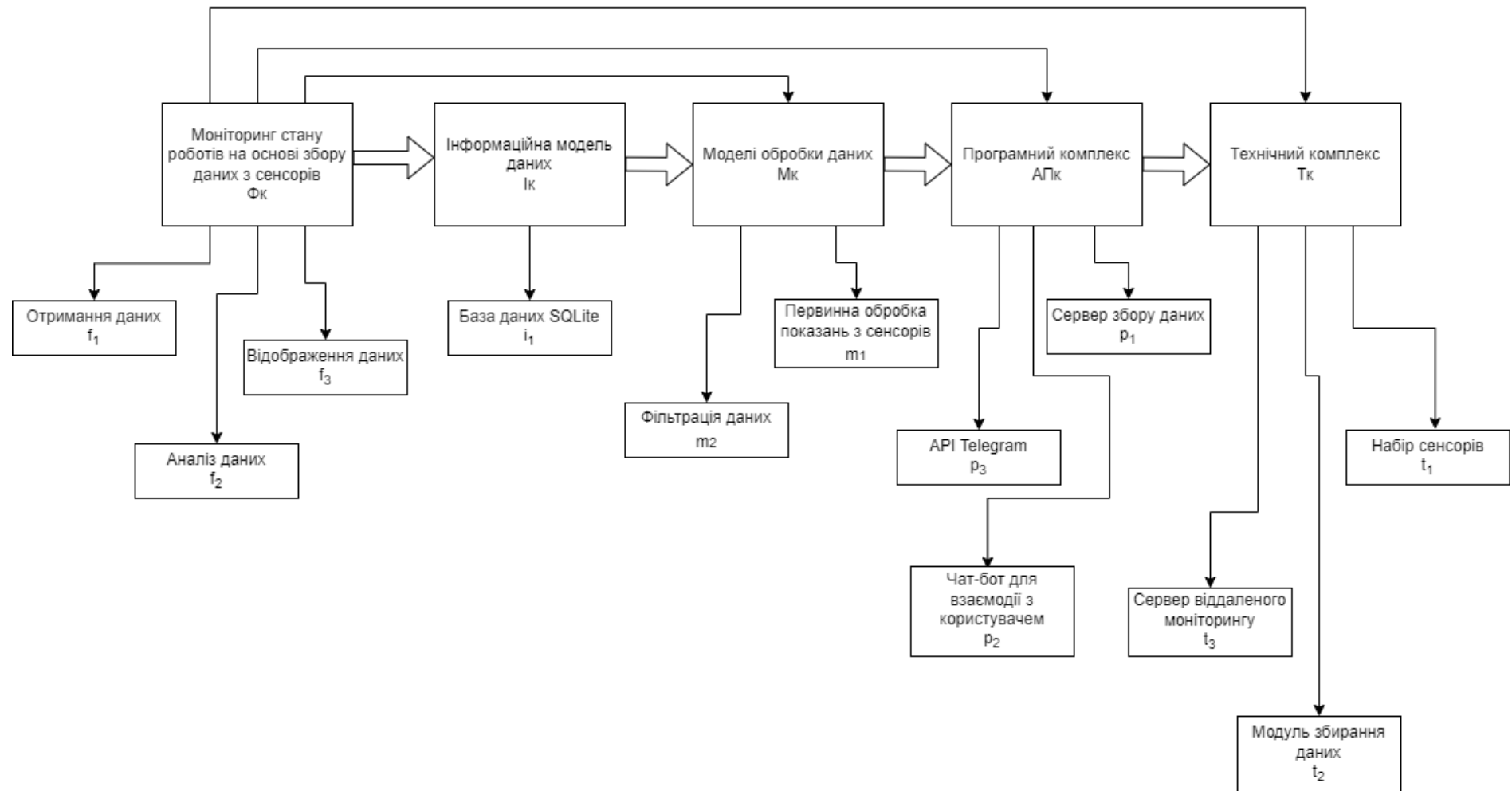


Рисунок 2.1 – Схема автоматизованої системи

2.1.1 Визначення вимог до датчиків системи

Експлуатаційно-технічні вимоги, що висуваються до датчиків засобів технічного діагностування, можна подати у вигляді трьох груп ознак, зумовлених: умовами експлуатації; видом та характером змін вхідної (контрольованої) величини; конструктивні особливості. Кожна з цих груп ознак має особливості [21].

У процесі діагностування на датчик з боку об'єкта впливає комплекс експлуатаційних факторів, з яких основними є механічні та температурні фактори. Відповідно до цього під час розробки датчика щодо нього пред'являються вимоги механічної міцності; стійкості до механічних та температурних впливів.

Проаналізувавши предметну область було обрано 3 робота. Перший робот представлено у вигляді промислового маніпулятора, другий робот представлено у вигляді робота укладальника. Третій робот представлено у вигляді робота-сортувальника. На кожному з яких закріплена певна кількість давачів.

Таким, чином у першого робота є датчик температури, датчик струму, датчик куту оберту.

У другого робота датчик струму та датчик зусилля.

У третього робота датчик напруги та датчик температури.

2.1.2 Опис датчиків та технічні характеристики

ДТСxx5. Термоопори з комутаційною головкою.

Термоопори з комутаційною головкою призначені для температурних вимірювань твердих, рідких та газоподібних середовищ, які неагресивні до захисної арматури та матеріалу чутливого елемента (ЧЕ) датчика. Технічні характеристики ДТСxx5 наведено у табл. 2.1.

Таблиця 2.1 – Технічні характеристики ДТСхх5

Характеристики	ДТСХХ5		
Номінальна статична характеристика (НСХ)	50М; 100М	50П; 100П	РТ100; РТ500; РТ100
Діапазон температур, що вимірюються: – клас допуску А – клас допуску В та С	від – 50 °С до + 180 °С	від – 100 °С до + 450 °С від – 196 °С до + 500 °С	від – 60 °С до + 300 °С від – 60 °С до + 500 °С
Умовний тиск	від 0,1 МПа до 6,3 МПа (залежить від конструктивного виконання)		
Показник теплової інерції	не більше від 10 с до 30 с		
Опір ізоляції	не менше 100 МОм		
Кількість чутливих елементів	1 або 2		
Схема внутрішніх з'єднувань провідників	2 – двоохдротова 3 – трьохдротова 4 – чотирьохдротова		
Виконання сенсора відносно корпусу	ізолюваний		
Виконання комутаційної головки	пластмасова, металева		
Тип нарізного штуцера	метрична різь, трубна різь		
Матеріал захисної арматури	сталь 12Х18Н10Т		
Ступінь захисту	IP54 IP65		

Термоопори з комутаційною головкою дозволяють вимірювати температуру до 500 °С (ДТС з платиновим ЧЕ) та до 180 °С (ДТС з мідним ЧЕ). Підмикання до вимірювальної лінії виконується мідним кабелем [22].

Зовнішній вигляд ДТСхх5 представлено нижче на рис. 2.2.



Рисунок 2.2 – Зовнішній вигляд ДСТхх5 в залежності від модифікації

Цифровий датчик струму і напруги CJMCU-219 на мікросхемі INA219 з шиною I2C.

Цифровий датчик струму і напруги CJMCU-219 призначений для вимірювання таких параметрів постійного струму як напруга, струм і споживана потужність. Модуль виконаний на мікросхемі INA219 - вимірювач струму та напруги з нульовим дрейфом і володіє малими розмірами і вагою при дуже великих можливостях та високою точністю вимірювань.

Мікросхема вимірює параметри протікання струму в будь-якому напрямку з автоматичним перемиканням полярності вимірювання. Застосувати модуль можна в системах, які контролюють процес заряду/розряду акумуляторних батарей, джерелах живлення з контролем напруги і споживаного навантаженням струму.

Зовнішній вигляд датчика наведено на рис. 2.3.

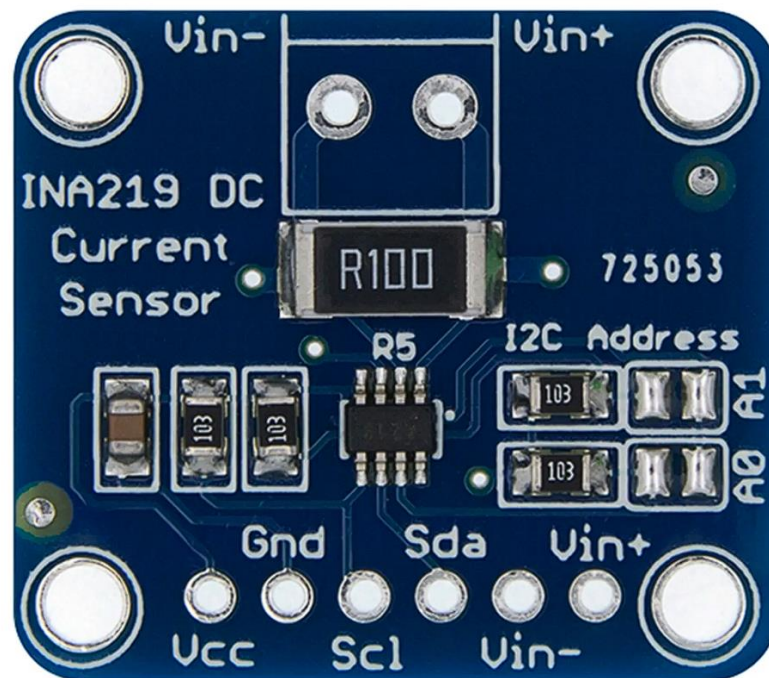


Рисунок 2.3 – Зовнішній вигляд CJMCU-219

Можливості зміни I2C адреси вимірювача дозволяють підключити на одну шину до 4-х таких пристроїв. Для збільшення точності вимірювань передбачений регістр калібрування [23]. Технічні характеристики CJMCU-219 наведено у таблиці 2.2.

Таблиця 2.2 – Технічні характеристики датчика CJMCU-219

Тип модуля	вимірювання електричного струму
Робоча температура	від - 40 °C до 85 °C
Дрейф в робочому температурному діапазоні:	100 мкВ
Максимальний вимірюваний струм	3,2 А
Точність вимірювання струму	0,8 мА

Продовження таблиці 2.2

Роздільна здатність вимірювача	12-біт
Швидкість інтерфейсу:	3,4 МГц
Максимальна вимірювана напруга	± 26 В
Калібрування	калібрувальний регістр
Внутрішні дані	вимірюваний струм та потужність
Фільтрація	x128 відліків
Напруга живлення	від 3 В до 5 В

Датчик кута повороту HRS100SSAB180.

Датчик кута повороту HRS100SSAB180 дозволяє вимірювати кут повороту в діапазоні від 0 ° до 180 °.

Завдання визначення кута повороту виникає коли необхідно визначати, наприклад, ступінь закриття або відкриття кришок, засувки або клапанів, що визначають режим обробки.

Для наближеного визначення кута повороту (точність: $\pm 2^\circ$) можна використовувати датчики на ефект Холла HRS100SSAB180, які працюють за принципом потенціометра [24]. Технічні характеристики HRS100SSAB180 наведено у таблиці 2.3.

Таблиця 2.3 – Технічні характеристики датчика HRS100SSAB180

Виробник	Honeywell
Категорія продукту	Промислові датчики руху та позиціонування
Продукт	Rotary Position Sensors
Тип виходу	Analog - Voltage

Продовження таблиці 2.3

Номінальна напруга	5 VDC
Лінійність	2 %
Мінімальна робоча температура	- 40 °C
Максимальна робоча температура	+ 85 °C
Діапазон опору	—
Допустиме відхилення	2°
Кут обертання – електричний, механічний:	від 0° до ~ 180°; від 0° до ~ 180°
Тип	Hall Effect Rotary Position Sensor
Технологія	Hall Effect
Довжина	27,69 мм
Ширина	27,69 мм
Підкатегорія	Sensors

Зовнішній вигляд HRS100SSAB180 представлено на рис 2.4.



Рисунок 2.4 – Зовнішній вигляд HRS100SSAB180

Датчик зусилля CZN-CP42.

Датчики зусилля CZN-CP42 – оснащені датчиками зусилля роботи здатні виконувати збирання, контурну обробку та вимірювання в широкому спектрі застосувань [25].

Технічні характеристики CZN-CP42 наведено у табл. 2.4.

Таблиця 2.4 – Технічні характеристики CZN-CP42

Поріг спрацьовування	від 0,2 Н до 100 Н
Робоча температура	від - 40 °С до 75 °С
Зовнішні розміри	27 мм x 63,75 мм x 0,34 мм
Особливості кріплення	самоклеючий
Робочий струм макс	1 мА

Завдяки конструкції з невеликою кількістю деталей і надійності датчики демонструють максимальний час безвідмовної роботи. Щоб прискорити роботу, вони повністю інтегруються в роботів. Зовнішній вигляд CZN-CP42 представлено на рис 2.5.



Рисунок 2.5 – Зовнішній вигляд CZN-CP42

2.2 Інформаційне забезпечення АС

Розробка та структурна побудова інформаційного забезпечення значною мірою залежить від задач, що розв'язуються, складу функцій управління, засобів перетворення інформації, форм подання даних тощо. У процесі розробки інформаційного забезпечення слід визначити: склад інформації, необхідної для розв'язання комплексу задач.

В ході виконання магістерської кваліфікаційної роботи було вирішено зробити 3 програмні частини.

- емулятор робота для моніторингу параметрів виміру;
- сервер для обробки показників. Сервер виконує функцію прийняття даних з давачів, обробляє їх та приймає рішення для подальшої передачі їх у застосунок «Telegram»;
- чат-бот для прийняття сповіщень.

Програмні емулятори успішно застосовуються в різних сферах: як засіб консолідації серверів, під час крос-платформної розробки, при заміщенні застарілої апаратури. Факт роботи відладчика у загальному оточенні легко виявляється, після чого програма, що вивчається, може або цілеспрямовано змінити свою поведінку, або спробувати порушити процес налагодження [26].

В умовах промислового виробництва датчики перебувають у значному видаленні як один від одного так і від місця де здійснюється централізована обробка вимірювальної інформації. Логічно передавати інформацію від датчиків по комп'ютерній локальній мережі, використовуючи добре розроблені мережеві технології.

Інтернет-технології дозволяють задовольнити потреби практично всіх користувачів, маючи певний функціонал, що полегшує вирішення певних завдань.

Більшість сучасних компаній, компаній і установ мають власні веб-портали, на яких розміщена та чи інша інформація. Однак існують великі

проблеми неоднорідності та дуже низької якості даних, низької швидкості відповіді сервісу, високої вартості.

У більшості випадків дані підготовлені погано, додатки та служби мають надлишковий інтерфейс, і важко зосередитися на головному. Багато часу витрачається на перемикання між службами або додатками для виконання певних завдань, що не є найкращим варіантом. Крім того, через високу вартість розробки та підтримки веб-сервісів і додатків страждає функціональність, у більшості випадків відсутнє машинне навчання, алгоритми, які використовуються в розробці, не є оптимальними, що, у свою чергу, впливає на продуктивність і якість додаток.

Виходячи з вищезазначених проблем, чат-боти все частіше впроваджуються у використання. Чат-бот – це програма, розроблена на основі технологій машинного навчання та нейронних мереж. Чат-бот створений і навчений для певного кола завдань.

Telegram має власне зашифроване хмарне сховище, розкидане в різних юрисдикціях, і воно набагато краще захищене, ніж Google та Apple. Що дає можливість використовувати ботів в корпоративних цілях, забезпечуючи конфіденційність даних.

Підсумовуючи, можна сказати, що використання платформи Telegram для створення чат-ботів є найбільш оптимальним, менше витрат на розробку, більше часу на створення інтелектуального сервісу, аналіз даних і навчання. Платформа Telegram є найбільш перспективною для розвитку та створення чат-ботів.

2.2.1 Вибір мови програмування

C# – це мова програмування, призначена для розробки найрізноманітніших додатків, призначених для виконання в середовищі .NET Framework.

Мова C# проста, типобезпечна і об'єктно-орієнтована. Завдяки безлічі нововведень C# забезпечує можливість швидкої розробки додатків, але при цьому зберігає виразність та елегантність, властиву мовам C [27].

Visual C# – це реалізація мови C# корпорацією Майкрософт.

Підтримка Visual C# у Visual Studio забезпечується за допомогою повнофункціонального редактора коду, компілятора, шаблонів проектів, конструкторів, майстрів коду, потужного та зручного відладчика та багатьох інших засобів.

Бібліотека класів .NET Framework надає доступ до багатьох служб операційної системи та інших корисних, правильних класів, що суттєво прискорює цикл розробки.

Програми C# виконуються в .NET, віртуальній системі виконання, що викликає загальномовне середовище виконання (CLR) та набір бібліотек класів.

Середовище CLR – це реалізація загальномовної інфраструктури мови (CLI), що є міжнародним стандартом від корпорації Майкрософт.

CLI є основою для створення середовищ виконання та розробки, у яких мови та бібліотеки прозоро працюють одна з одною.

Вихідний код, написаний мовою C# компілюється в проміжну мову (IL), що відповідає специфікаціям CLI.

Код на мові IL та ресурси, у тому числі растрові зображення та рядки, зберігаються у збірці, зазвичай з розширенням .dll. Складання містить маніфест з інформацією про типи, версії, мову та регіональні параметри для цієї збірки.

Під час виконання програми C# збірка завантажується у середу CLR. Середовище CLR виконує JIT-компіляцію з коду мовою IL в інструкції машинної мови.

Середовище CLR також виконує інші операції, наприклад, автоматичне складання сміття, обробку винятків та управління ресурсами.

Код, який виконується середовищем CLR, іноді називають "керованим кодом". "Некерований код" компілюється на машинну мову, призначену для

конкретної платформи.

Забезпечення взаємодії між мовами є ключовою особливістю .NET.

Код IL, створений компілятором C# відповідає специфікації загальних типів (CTS). Код IL, створений з коду C#, може взаємодіяти з кодом, створеним з версій .NET для мов F#, Visual Basic, C++.

Існує більше 20 інших мов, сумісних із CTS. Одна збірка може містити кілька модулів, написаних різними мовами .NET, і всі типи можуть посилатися один на одного, якби вони були написані однією мовою.

Крім служб часу виконання .NET також включає розширені бібліотеки. Ці бібліотеки підтримують безліч різних робочих навантажень.

Вони впорядковані за просторами імен, які надають різні корисні можливості: від операцій файлового введення та виведення до керування рядками та синтаксичного аналізу XML, від платформ веб-додатків до елементів керування Windows Forms.

Зазвичай програма C# активно використовують бібліотеку класів .NET для вирішення типових завдань.

Програми C# можна зберігати у кількох вихідних файлах. При компіляції програми C# всі вихідні файли обробляються разом, у своїй можуть вільно посилатися друг на друга.

По суті, це аналогічно тому, ніби всі вихідні файли були об'єднані в один великий файл перед обробкою.

У C# ніколи не використовуються випереджаючі оголошення, оскільки порядок оголошення, за рідкісним винятком, не відіграє жодної ролі.

У C# немає вимог оголошувати тільки один відкритий тип в одному вихідному файлі, а також ім'я вихідного файлу не повинно збігатися з типом, що оголошується в цьому файлі.

2.3 Висновки до другого розділу

В результаті виконання другого розділу магістерської кваліфікаційної роботи було виконано наступні завдання:

- розглянуто основне завдання для використання автоматизованої системи;
- функціональне забезпечення автоматизованої системи;
- визначено вимоги до датчиків системи;
- обрані давачі для автоматизованої системи, описані їх технічні характеристики;
- описано інформаційне забезпечення до автоматизованої системи;
- обрана мова програмування.

З РОЗРОБЛЕННЯ МЕТОДУ МОНІТОРИНГУ СТАНУ РОБОТІВ НА БАЗІ СЕНСОРНОЇ ТЕХНОЛОГІЇ

3.1 Опис технології автоматизованого збирання даних з механізмів промислових роботів

Для аналізу сигналу з сенсорів роботів насамперед необхідно провести збирання цих даних. У програмі, що розроблюється необхідно передбачити можливість опитування кількох датчиків, що дозволить проводити порівняльний аналіз їхньої поведінки.

Основний алгоритм роботи програми потребує, на початку роботи встановити кількість опитуваних датчиків [28]. Основний цикл програми опитує всі датчики і записує отримані значення відповідні змінні `sensor[i]`.

Після проведення опитування проводиться обробка даних. Отримані дані виводяться на екран і додаються масив `data[i, j]`. При цьому елементи масиву постійно зміщуються, тобто в пам'яті можуть зберігатися дані за останні кілька секунд роботи програми.

Використання масиву дозволить виводити на екран або в файл не тільки поточні значення з датчиків, але й аналізувати динаміку зміни параметрів, що вимірюються. Відповідно, дані з масивів виводяться на екран і за потреби зберігаються у файл. Алгоритм роботи програми в режимі опитування сенсорів роботів наведено рисунку 3.1.

До кожного датчика у програмі проводився розрахунок низки параметрів сигналу. Розраховуються максимальне та мінімальне значення сигналу (за допомогою виразів (3.1) та (3.2)), інтегральний показник (за допомогою виразу (3.3), середнє значення за допомогою виразу (3.4), диференціальний показник за допомогою виразу (3.5) та згладжені дані за допомогою виразу (3.6).

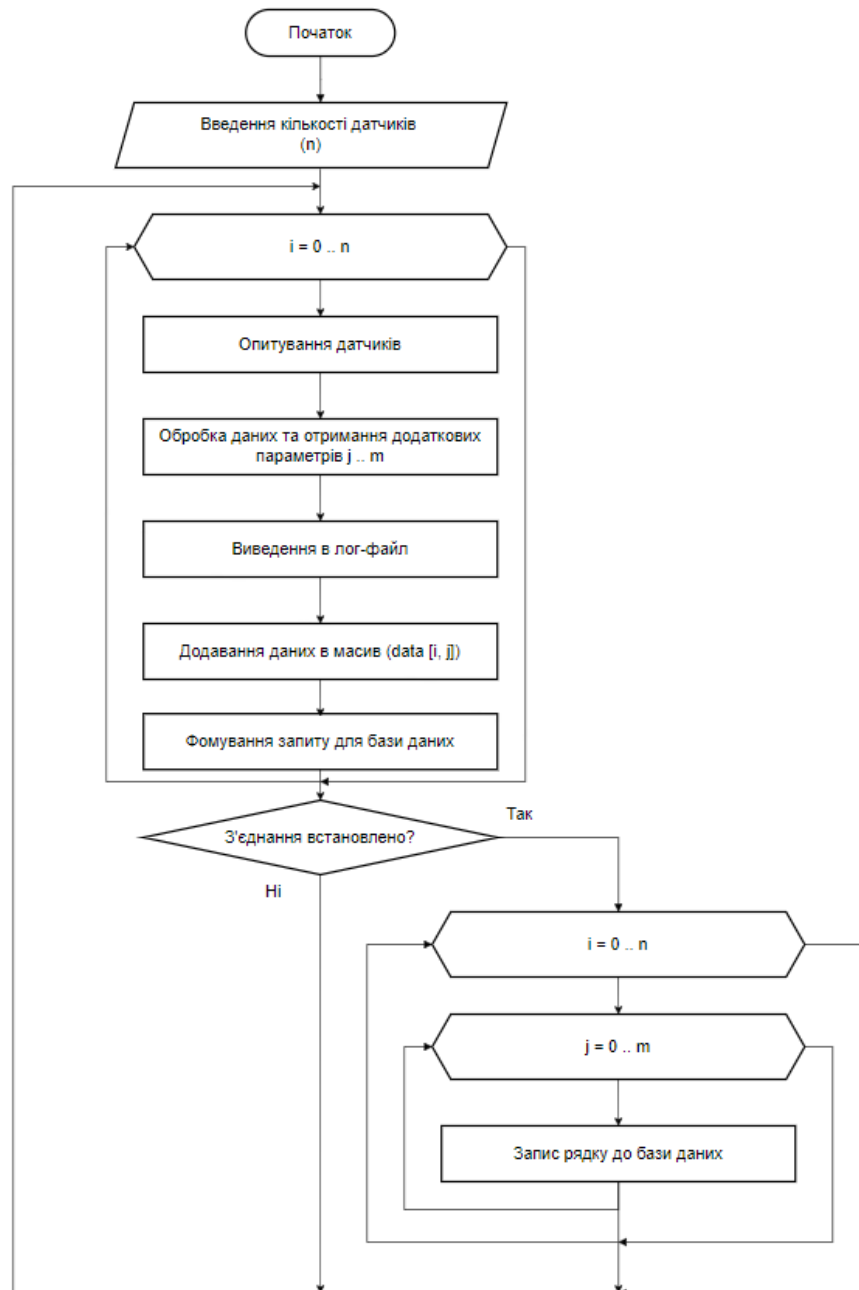


Рисунок 3.1 – Алгоритм опитування сенсорів роботів

Розрахунок максимуму ($\max$) і мінімуму ($\min$) дозволяє відстежувативати діапазон змін вхідного сигналу:

$$\max_i = \begin{cases} data_i, & data_i > \max_i \\ \max_{i-1}, & data_i \leq \max_i \end{cases}, \quad (3.1)$$

$$\min_i = \begin{cases} data_i, data_i < \min_i \\ \min_{i-1}, data_i \geq \min_i \end{cases} . \quad (3.2)$$

Інтегральний показник (sum) розраховується як сума всіх значень вхідного сигналу. Даний параметр цікавий для сигналів, що показують відхилення від нормального (нульового) значення та дозволяє відстежити накопичення помилки:

$$sum_i = \sum_{i=0}^n data_i . \quad (3.3)$$

Середнє значення (average) розраховується як відношення суми до кількості вимірів (n):

$$average_i = \frac{sum_i}{n} . \quad (3.4)$$

Зміна сигналу (диференціальний показник delta) визначається як різниця поточного та попереднього сигналів з датчика. Даний параметр характеризує швидкість зміни сигналу і разом із sum може використовуватися при доборі коефіцієнтів ПІД регулятора:

$$delta_i = data_i - data_{i-1} . \quad (3.5)$$

Згладжування проводиться за рахунок усереднення останніх вимірювань. Коефіцієнт k задає ступінь згладженості сигналу та дозволяє фільтрувати випадкові сплески та високочастотні шуми з датчика:

$$\text{smooth}_i = \frac{\sum_{j=i-k}^i \text{data}_j}{k}. \quad (3.6)$$

Отримані дані дозволять оцінити застосування певного датчика в автоматизованій системі і визначити необхідні алгоритми фільтрації даних з датчика.

3.2 Розроблення структури автоматизованої системи

Автоматизована система моніторингу стану роботів в реальному часі передбачає використання різноманітних сенсорів для отримання оперативної інформації про поточні значення технологічних параметрів.

На рисунку 3.2 показано приклад отримання поточної інформації про стан роботи промислового роботу.

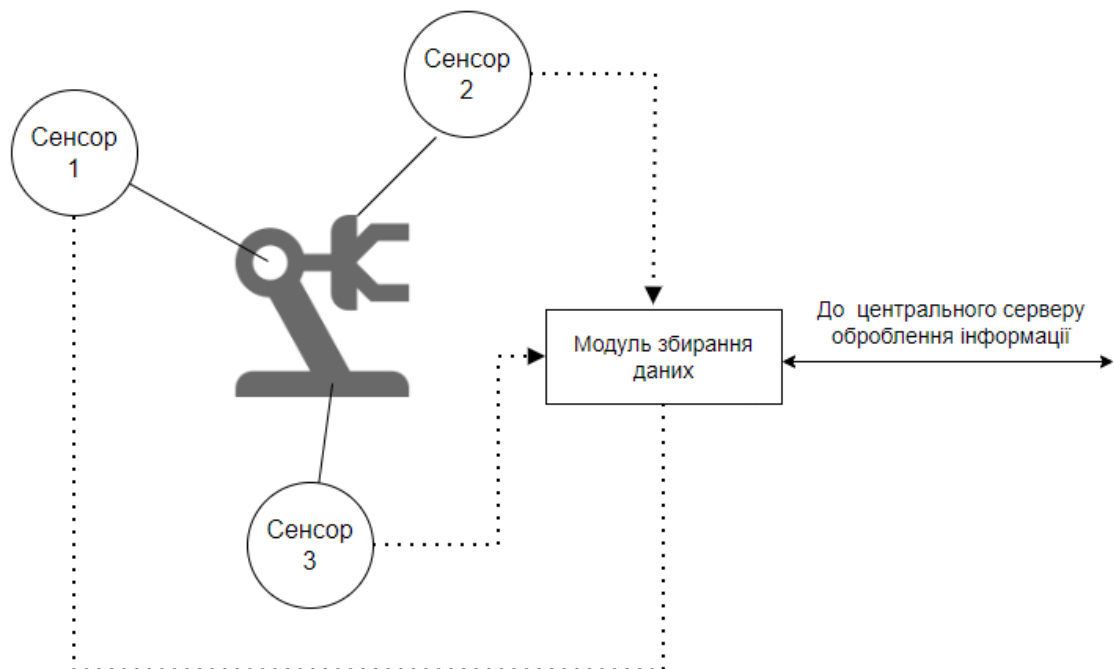


Рисунок 3.2 – Приклад отримання поточної інформації про стан роботи промислового роботу

На даному рисунку можна бачити, що на роботі можуть встановлюватись різні сенсори. Так, наприклад, сенсор 1 може контролювати стан суглобів промислового маніпулятора. Сенсор 2 встановлюється на захваті для моніторингу зусилля, що прикладається при перенесенні вантажів. Сенсор 3 контролює роботу високонавантажених механізмів обертання маніпулятора та використовується для прогнозування можливих поломок у зв'язку з, наприклад, перегріванням опорних підшипників або валів.

Використання різних за типом сенсорів та віддаленість їх від централізованого сховища даних потребує розроблення уніфікованих засобів збирання та передавання інформації до серверу бази даних.

Пропонується застосування проміжних модулів збирання даних із сенсорів. Тоді, вся інформація з сенсорів надходить на цей модуль збирання, що розташований поблизу даного роботу та розроблений спеціально для взаємодії із певним типом сенсорів. Його основне призначення, отримання даних із сенсорів, попередня обробка та фільтрація, накопичення у внутрішньому буфері до моменту надходження запиту від центрального модуля збирання даних.

Поєднання з центральним модулем збирання даних відбувається за допомогою захищеної промислової мережі використанням стандартних протоколів взаємодії між обладнанням.

Таких проміжних модулів може використовуватись декілька, по кількості робіт, що застосовуються на виробничій ділянці.

На рисунку 3.3 наведена загальна схема організації процесу взаємодії між роботами в виробничому середовищі та системою віддаленого моніторингу стану обладнання з використанням мобільних засобів отримання та візуалізації інформації.

З наведеного рисунку можна бачити, що використовується центральний модуль збирання даних на який сходяться всі потоки інформації в мережі поєднання сенсорів промислових робіт.

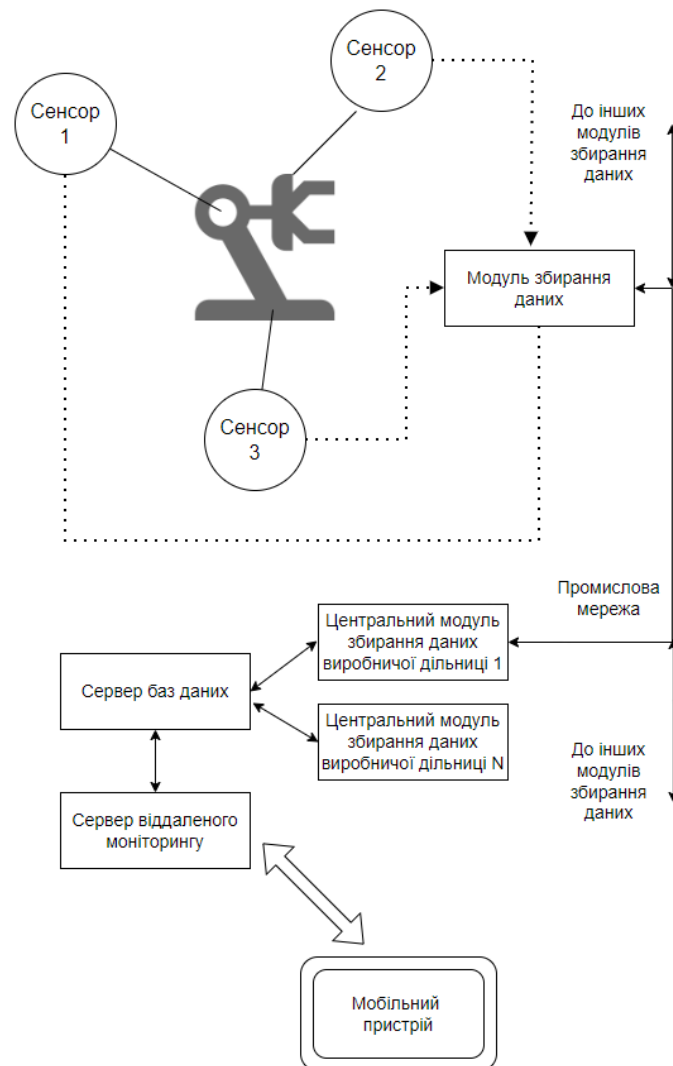


Рисунок 3.3 – Загальна схема організації процесу взаємодії між роботами в виробничому середовищі та системою віддаленого моніторингу

Центральний модуль збирання даних передає отриману інформацію до серверу баз даних. Сервер може розміщуватись віддалено, або локально безпосередньо на виробничій ділянці, якщо підприємство не велике і складається з одного або декількох приміщень.

Передбачається, що на кожній виробничій ділянці буде використовуватись свій окремий центральний модуль збирання даних. Всі окремі модуля збирання даних підприємства поєднуються з сервером баз даних.

Для віддаленого моніторингу стану роботів за показаннями їх сенсорів

пропонується розробити спеціальну програму-сервер. Програма-сервер призначена для організації зв'язку з мобільними пристроями в реальному часі. Для цього пропонується використовувати сучасну технологію чат-ботів для месенджера Telegram. Така технологія організації взаємодії з промисловим обладнанням дозволить впровадити сучасні та захищені протоколи передавання даних, що можуть реалізовані на сучасних програмних та апаратних засобах, та не потребують створення або закупівлі окремих кошовних серверів. Також вирішується задача обслуговування програмних ресурсів для організації обміну даними, тому що вона передбачена API Telegram.

На рисунку 3.4 наведена схема взаємодії модуля збирання та передавання інформації з сенсорів за допомогою програми чат-боту.

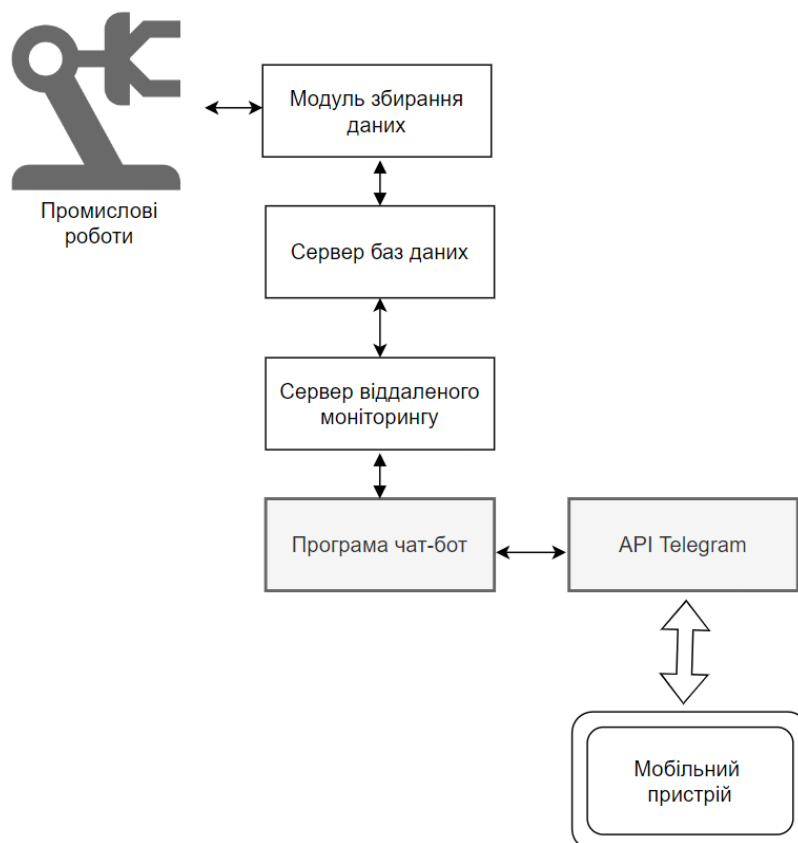


Рисунок 3.4 – Схема організації зв'язку з промисловим обладнанням для моніторингу його стану

3.3 Висновки до третього розділу

В результаті виконання третього розділу магістерської кваліфікаційної роботи запропонована методика отримання поточної інформації з сенсорів промислових роботів для виконання задачі моніторингу їх станів в реальному часі.

Запропонована схема передавання даних від сенсорів роботів на мобільні пристрої з використанням сучасної технології чат-ботів для месенджера Telegram.

Розроблена загальна схема організації процесу взаємодії між роботами в виробничому середовищі та системою віддаленого моніторингу. Описані всі компоненти, що входять до даної схеми.

4 РОЗРОБКА СТРУКТУРИ БАЗИ ДАНИХ

4.1 Аналіз даних та розробка моделі структури бази даних

В результаті проведеного аналізу літератури та предметної області визначено набір даних, що становить інформацію про поточний стан промислових роботів. Такою інформацією є:

- дані про промислове обладнання (роботів);
- довідник сенсорів, що використовуються в промисловому обладнанні;
- журнал отримання поточної інформації з сенсорів;
- інформація про прив'язку (встановлення) датчиків на промислові роботи.

Дані про промислове обладнання – це характеристики промислових роботів, що застосовуються на певній виробничій ділянці. Характеристикою промислового обладнання, що підлягає збереженню є:

- назва промислового роботу;
- тип роботу;
- напруга живлення;
- кількість ступенів рухомості;
- вага.

Для зберігання даної інформації використовується наступна структура таблиці бази даних (табл. 4.1).

Таблиця 4.1 – Структура таблиці «Промисловий робот»

Назва поля	Тип поля даних	Розмір поля даних	Ключ
ID_Robot	Integer	4	Primary

Продовження таблиці 4.1

Name_Robot	Text		
RobotType	Text		
Power	Decimal		
MoveStupen	Integer	4	
Weight	Decimal		

Довідник сенсорів, що використовуються в промисловому обладнанні описується такою інформацією:

- назва сенсору;
- напруга живлення;
- тип вимірюваної величини;
- максимальне значення величини на виході;
- порогове значення величини.

Порогове значення величини застосовується для визначення, чи перевищений вказаний поріг для видачі попередження оператору на мобільному пристрої.

Інформація про максимальне значення величини на виході використовується для розрахунку вихідного значення з аналого-цифрового перетворювача в залежності від його розрядності.

Тип вимірюваної величини використовується при формуванні повідомлення оператору через мобільний пристрій.

Для зберігання вказаних даних розроблена наступна структура таблиці бази даних, що показана в таблиці 4.2.

Інформація про прив'язку (встановлення) датчиків до промислового роботу використовується для визначення місця знаходження конкретного сенсора при роботі з базою даних.

Таблиця 4.2 – Структура таблиці «Довідник сенсорів»

Назва поля	Тип поля даних	Розмір поля даних	Ключ
ID_Sensor	Integer	4	Primary
Name_Sensor	Text		
PowerSensor	Decimal		
ValueType	Text		
MaxValue	Decimal		
PorogValue	Decimal		

В будь-який момент часу програміст та оператор може дізнатись про те, звідки надійшла інформація про поточний стан обладнання.

Для прив'язки до обладнання необхідно записати до бази даних ідентифікатор промислового роботи (береться із таблиці з даними про промислові роботи) та ідентифікатор потрібного сенсора (береться із таблиці з даними про сенсори).

Структура таблиці «Прив'язка датчиків» показана в таблиці 4.3.

Таблиця 4.3 – Структура таблиці «Прив'язка датчиків»

Назва поля	Тип поля даних	Розмір поля даних	Ключ
ID_BoundSensor	Integer	4	Primary
ID_Robot	Integer	4	
ID_Sensor	Integer	4	

Журнал отримання поточної інформації з сенсорів представлений наступним набором даних:

- ідентифікатор промислового роботи (береться із таблиці з даними про промислові роботи);
- ідентифікатор сенсора (береться із таблиці з даними про сенсори);
- час отримання даних;
- дата отримання даних;
- отримані значення вимірюваної величини;
- поточні дані про максимальний поріг значень;
- ознака про перевищення порогу значення.

Час і дата отримання даних необхідна для проведення подальших досліджень з накопленою інформацією. Поточні дані про максимальний поріг значень потрібно фіксувати на момент проведення вимірювання, тому, що в процесі роботи програми вони можуть змінюватись.

Отримані значення вимірюваної величини використовуються для формування пакету повідомлення операторові системи на його запит про поточний стан роботи промислових роботів.

Структура таблиці «Журнал інформації» показана в таблиці 4.4.

Таблиця 4.4 – Структура таблиці «Журнал інформації»

Назва поля	Тип поля даних	Розмір поля даних	Ключ
ID_Journal	Integer	4	Primary
ID_Robot	Integer	4	
ID_Sensor	Integer	4	
DateMeasure	Long	8	
TimeMeasure	Long	8	
MeasureValue	Decimal		
PorogValue	Decimal		
Result	Text		

В результаті проектування бази даних отримана наступна її структура, що показана на рисунку 4.5.

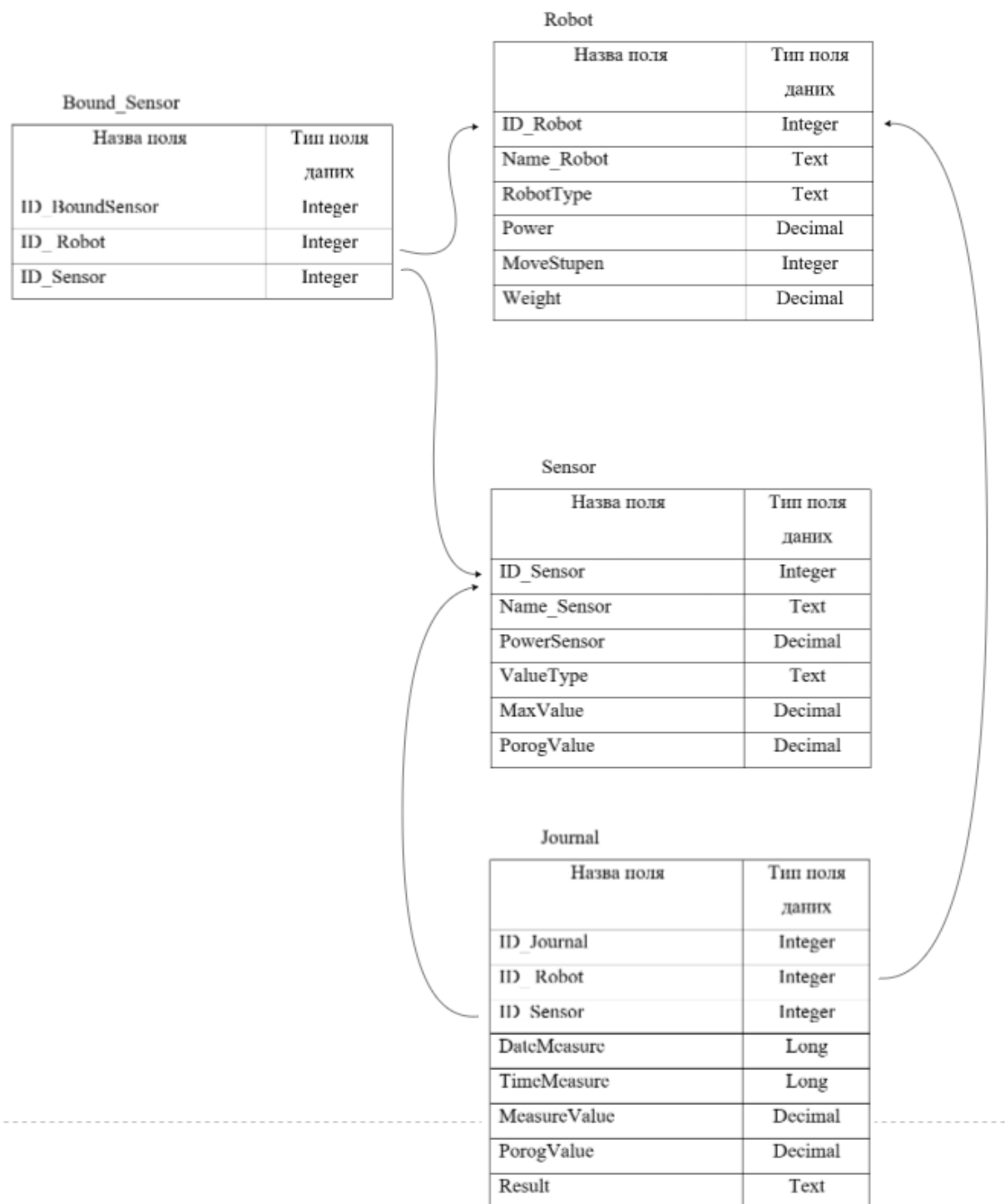


Рисунок 4.5 – Розроблена структура бази даних

В розробленій структурі бази даних використовуються зв'язки типу один-до-багатьох. В якості сервера бази даних обрано SQLite, як один з варіантів вбудованої СУБД, що добре підходить для впровадження в автоматизовані системи управління технологічними процесами.

4.2 Висновки до четвертого розділу

В результаті виконання четвертого розділу магістерської кваліфікаційної роботи розроблена структура бази даних для центрального серверу та програми чат-ботів для месенджера Telegram. Для цього було зроблено аналіз даних та розробка моделі структури бази даних, визначено набір даних, що становить інформацію про поточний стан промислових роботів.

5 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

5.1 Постановка задачі для проведення експерименту

Для проведення експериментальних досліджень розробленого методу автоматизованого моніторингу стану роботів на базі сенсорної технології необхідно розробити програму – сервер з можливістю запуску програмного чат-боту для месенджера Telegram.

Розроблений сервер в реальному часі повинен генерувати потік даних від сенсорів, що знімають показання з робототехнічних пристроїв та відправляють їх на мобільний пристрій користувача за його запитом.

Для управління сервером необхідно розробити структуру чат-боту зі інтерфейсом користувача, який буде в подальшому представлений в месенджері Telegram в разі звернення до цього чат-бота.

Ескіз головного екрану програми показано на рисунку 5.1.

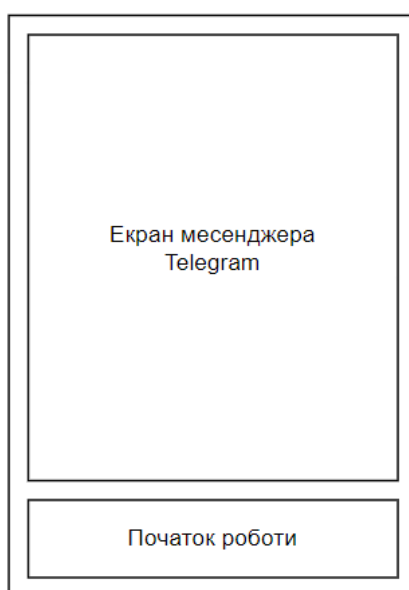


Рисунок 5.1 – Ескіз головного екрану програми

На головному екрані потрібно створити кнопку для початку роботи з телеграм-ботом. При натисканні на цю кнопку користувачу повинне прийти привітання, як індикатор того, що сервер почав з ним взаємодію. Також можна вивести додаткове інформаційне повідомлення про можливості, що представляє програма.

На основному екрані повинні з'явитись дві нові кнопки для обирання подальших дій. В даному випадку, це може бути пропозиція подивитись список наявних робіт (рис. 5.2).



Рисунок 5.2 – Екран переходу до списку робіт

Екран зі списком робіт показую повну інформацію про назву цих пристроїв та пропонує обрати один з них для детального перегляду інформації, що була отримана з датчиків, які з ним пов'язані.

На рисунку 5.3 показано ескіз екрану мобільного пристрою для відображення інформації про список робіт.

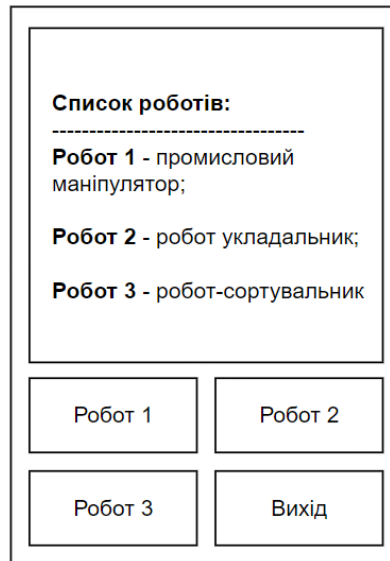


Рисунок 5.3 – Ескіз екрану мобільного пристрою для відображення інформації про список роботів

Після обирання потрібного робота користувач переходить на новий екран де бачить список датчиків, що закріплені за даним роботом та їх повні назви.

На рисунку 5.4 показано приклад екрану зі списком датчиків, що зареєстровані на сервері для робота 1.

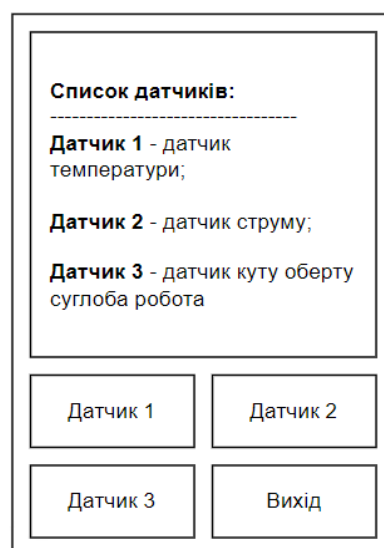


Рисунок 5.4 – Екран переходу до списку датчиків

Для перегляду інформації, що надійшла від певного датчика необхідно обрати його назву, натискаючи на відповідну кнопку знизу екрану.

На рисунку 5.5 показано приклад реалізації вікна для відображення даних з обраного датчика робота 1.

The image shows a graphical user interface window. At the top, there is a label **Назва датчика** (Sensor Name) followed by the text **Дані з датчика:** (Data from sensor:). Below this, a list of data points is displayed:

- назва параметру, що вимірюється;
- максимально припустиме значення параметру;
- поточне значення параметру;
- дата виміру;
- час виміру;
- оцінка результату.

 At the bottom of the window, there are four buttons arranged in a 2x2 grid: **Датчик 1**, **Датчик 2**, **Датчик 3**, and **Вихід** (Exit).

Рисунок 5.5 – Приклад реалізації вікна для відображення даних з обраного датчика

Для повернення на крок назад використовується кнопка «Вихід».

5.2 Розробка алгоритму роботи програми

Алгоритм роботи програми показано на рисунку 5.6.

Робота програми починається з підключення клієнту до серверу. Після цього йому надається можливість перейти до списку роботів, або вийти з програми.

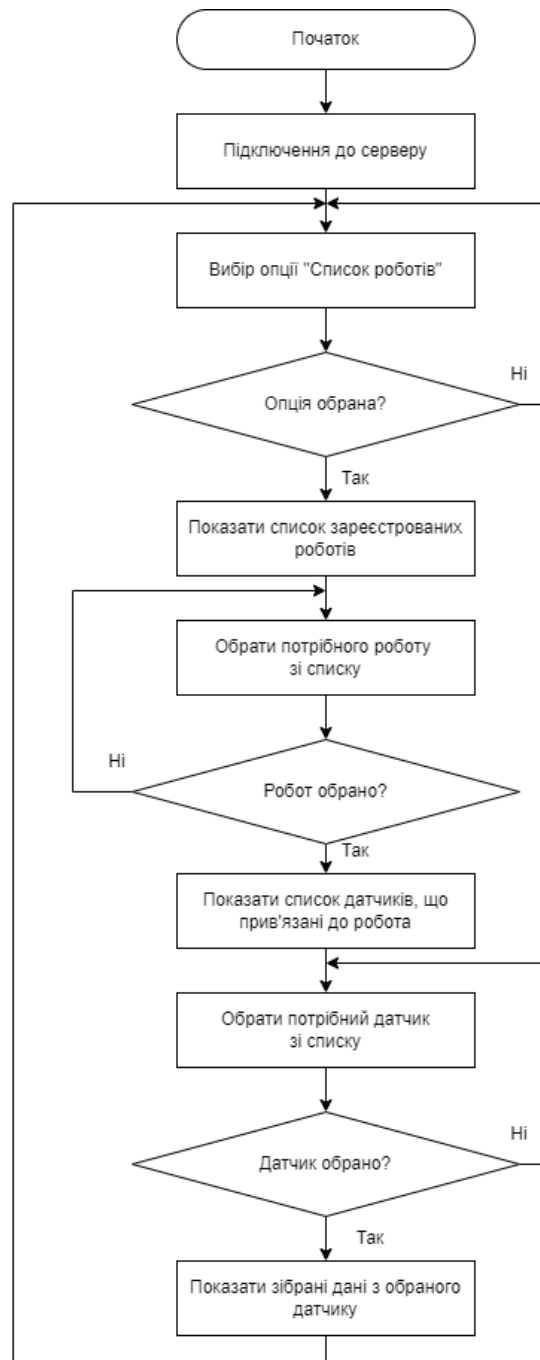


Рисунок 5.6 – Алгоритм роботи програми

Після підтвердження надання йому списку роботів користувачеві надається список зареєстрованих в базі даних роботів з можливістю обирання потрібного для перегляду його стану.

При обиранні потрібного робота користувачеві формується список датчиків, що до нього прив'язані. Користувачеві потрібно обрати необхідний

датчик для перегляду зібраних з нього даних.

При обиранні потрібного датчику на екран виводиться інформація про його характеристики та поточний стан, на основі зібраних під час роботи програми даних. Після перегляду необхідної інформації можна повернутись на початок програми для обирання іншої опції з меню користувача.

5.3 Розробка програмного засобу для виконання експериментальних досліджень

Програмний засіб створено мовою програмування C# з використанням бібліотеки Telegram.Bot. Bot API – це інтерфейс на основі HTTP, створений для розробників, які зацікавлені в створенні ботів для Telegram.

Запуск сервера відбувається при старті програми (рис. 5.7).

Ссылка 1

```
private void Form1_Load(object sender, EventArgs e)
{
    Client.GetUpdatesAsync(1000000, timeout: 3000000);
    Client.StartReceiving(mUpdate, mError);
    isConnect = true;
}
```

Рисунок 5.7 – Запуска сервера

Сервер одночасно є клієнтом спеціально створеного телеграм-бота, який пересилає повідомлення від користувача в нашу програму.

Для роботи нашого сервера необхідно задати ключ (токен) для доступу до чат-боту (рис. 5.8).

```
private readonly TelegramBotClient Client =
    new TelegramBotClient("5888416827:AAFw9c6qX_CAfGLC8axJYzG63-gXcmt30Sg");
```

Рисунок 5.8 – Токен для доступу до чат-бота

Обробка всіх повідомлень, що надходять до сервера відбувається в функції `mUpdate`. На рисунку 5.9 показан код цієї функції.

Ссылка: 2

```

async private Task mUpdate(ITelegramBotClient botClient, Update update, CancellationToken token)
{
    var message = update.Message;

    if (message.Text != null)
    {
        if (message.Text == "/start")...
        if (message.Text == "Список роботів")...
        if (message.Text.Contains("Робот"))...
        if (message.Text.Contains("Датчик"))...
        if (message.Text == "На головну")...
    }
}

```

Рисунок 5.9 – Код функції обробки повідомлень від користувача

Як можна бачити в даній функції є п'ять обробників повідомлень, що надходять. Першим повідомленням є команда «/start». Ця команда набирається на початку роботи з нашим сервером.

На рисунку 5.10 показано вихідний код обробника команди «/start».

```

if (message.Text == "/start")
{
    ReplyKeyboardMarkup keyboard_1_level = new ReplyKeyboardMarkup(new[] {
        new KeyboardButton[] { "Список роботів", "На головну" }
    })
    {
        ResizeKeyboard = true
    };

    String SensorReport = "";
    SensorReport += "Ласкаво просимо до системи автоматизованого " +
        "моніторингу стану роботів на базі сенсорної технології!";

    await botClient.SendTextMessageAsync(message.Chat.Id, SensorReport, replyMarkup: keyboard_1_level);
    return;
}

```

Рисунок 5.10 – Вихідний код обробника команди «/start».

З наведеного коду можна бачити, що у відповідь на отриману команду формується повідомлення з двома кнопками «Список роботів» та «На головну» у відповідності до розроблених раніш ескізів. Також формується привітання з описом дій для користувача.

Якщо користувач натисне на кнопку «Список роботів», то в його відповідь буде оброблена наступним фрагментом коду (рис. 5.11).

```

if (message.Text == "Список роботів")
{
    String sql = String.Format("Select * from robot ");
    List<System.Object> rowsRobot = (List<System.Object>)DataBase.LoadList("sensorRobot.robotMember", sql);

    int key = (rowsRobot.Count + 1);
    int row = key / 2;
    int rest = key % 2;
    if (rest > 0) row = row + 1;

    KeyboardButton[][] kbc = new KeyboardButton[row][];
    KeyboardButton[] keys = new KeyboardButton[key];

    int num = 0;
    foreach (System.Object item in rowsRobot)
    {
        keys[num] = new KeyboardButton((item as robotMember).Name_Robot);
        num++;
    }
    keys[num] = new KeyboardButton("На головну");

    int key_num = 0;
    for (int i = 0; i < row; i++)
    {
        if ((key_num + 1) < key)
        {
            kbc[i] = new KeyboardButton[] { keys[key_num], keys[key_num + 1] };
        }
        else kbc[i] = new KeyboardButton[] { keys[key_num] };
        key_num = key_num + 2;
    }

    ReplyKeyboardMarkup keyboard_robot = new ReplyKeyboardMarkup(kbc)
    {
        ResizeKeyboard = true
    };

    await botClient.SendTextMessageAsync(message.Chat.Id, "Оберить назву робота для перегляду списку датчиків ...", replyMarkup: keyboard_robot);

    return;
}

```

Рисунок 5.11 – Обробник команди «Список роботів»

В даному коді відбувається підключення до бази даних та формування до неї запиту для отримання повного списку роботів, що в неї зареєстровано.

З отриманого списку формується набір кнопок. Кожна кнопка відповідає одному роботу з бази даних. Також до списку додається ще одна кнопка «На головну» для виходу з даного режиму роботи програми.

Після формування списку кнопок, користувачеві виводиться

повідомлення «Оберіть назву робота для перегляду списку датчиків ...».

Опрацювання обраного робота відбувається наступним фрагментом коду (рис. 5.12).

```

if (message.Text.Contains("Пошук"))
{
    String sql_ID_Robot = String.Format("Select ID_Robot from robot where Name_Robot = '{0}'", message.Text);
    String ID_Robot = DataBase.Find_Value(sql_ID_Robot, "ID_Robot");
    ID_Robot_Last = ID_Robot;

    String sql = String.Format("Select * from BoundSensor where ID_Robot = '{0}'", ID_Robot);
    List<System.Object> rowsSensor = (List<System.Object>)DataBase.LoadList("sensorRobot.BoundSensorMember", sql);

    int key = (rowsSensor.Count + 1);
    int row = key / 2;
    int rest = key % 2;
    if (rest > 0) row = row + 1;

    KeyboardButton[][] kbc = new KeyboardButton[row][];
    KeyboardButton[] keys = new KeyboardButton[key];

    int num = 0;
    foreach (System.Object item in rowsSensor)
    {
        String NameSensor = DataBase.Find_Value(String.Format("Select Name_Sensor from Sensor where ID_Sensor = '{0}'",
            (item as BoundSensorMember).ID_Sensor), "Name_Sensor");
        keys[num] = new KeyboardButton(String.Format("{0} {1}", NameSensor, (item as BoundSensorMember).ID_Sensor));
        num++;
    }
    keys[num] = new KeyboardButton("На головну");

    int key_num = 0;
    for (int i = 0; i < row; i++)
    {
        if ((key_num + 1) < key)
        {
            kbc[i] = new KeyboardButton[] { keys[key_num], keys[key_num + 1] };
        }
        else kbc[i] = new KeyboardButton[] { keys[key_num] };
        key_num = key_num + 2;
    }

    ReplyKeyboardMarkup keyboard_robot = new ReplyKeyboardMarkup(kbc)
    {
        ResizeKeyboard = true
    };

    await botClient.SendTextMessageAsync(message.Chat.Id, "Оберіть датчик для перегляду даних з нього ...", replyMarkup: keyboard_robot);
}

```

Рисунок 5.12 – Опрацювання обраного робота

Дана функція знов звертається до бази даних, щоб отримати список датчиків, які пов'язані з обраним роботом.

На екрані формується набір кнопок з назвою датчика та його ідентифікаційним номером. Надалі користувач може натиснути відповідну кнопку, та отримати детальну інформацію про датчик. Також користувачеві буде надана інформація про останні виміри за допомогою обраного датчика.

На рисунку 5.13 показано фрагмент коду для формування детальної інформації по обраному датчику.

З наведеного фрагменту можна бачити, що формується SQL-запит до

бази даних для отримання відомості про обраний датчик. З даних, що надійшли від БД формується форматована відповідь в якій присутні такі поля:

- напруга живлення;
- параметр, що вимірюється;
- максимальне значення параметру;
- порогове значення параметру.

```
string ID_Sensor = "0";
string[] ar_sn = message.Text.Split(' ');
if (ar_sn.Length >= 2)
{
    ID_Sensor = ar_sn[ar_sn.Length - 1];
}

String SensorReport = "";
String sql = String.Format("Select * from Sensor where ID_Sensor = '{0}'", ID_Sensor);
List<System.Object> rowsSensorJournal = (List<System.Object>)DataBase.LoadList("sensorRobot.SensorMember", sql);
foreach (System.Object item in rowsSensorJournal)
{
    SensorReport += String.Format("<b><u>{0}</u></b>\n", (item as SensorMember).Name_Sensor.ToUpper());
    SensorReport += String.Format("\n");
    SensorReport += String.Format("<b>Характеристики датчика:</b>\n");
    SensorReport += String.Format("Напруга живлення: {0}\n", (item as SensorMember).PowerSensor);
    SensorReport += String.Format("Параметр, що вимірюється: {0}\n", (item as SensorMember).ValueType);
    SensorReport += String.Format("Максимальне значення параметру: {0}\n", (item as SensorMember).MaxValue);
    SensorReport += String.Format("Порогове значення параметру: {0}\n", (item as SensorMember).PorogValue);
    SensorReport += String.Format("\n");
    SensorReport += String.Format("<b>Дані з датчика за останні хвилини:</b>\n");

    String sql_journal = String.Format("Select * from Journal where ID_Sensor = '{0}' order by DateMeasure DESC", ID_Sensor);
    List<System.Object> rowsJournal = (List<System.Object>)DataBase.LoadList("sensorRobot.JournalMember", sql_journal);
    int max_row = 5;
    foreach (System.Object journal in rowsJournal)
    {
        Int64 unixTimeStamp = Int64.Parse((journal as JournalMember).DateMeasure);
        DateTime dateTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
        dateTime = dateTime.AddSeconds(unixTimeStamp).ToLocalTime();

        SensorReport += String.Format("Дата та час виміру: {0}\n", dateTime);
        SensorReport += String.Format("Виміряне значення: {0}\n", (journal as JournalMember).MeasureValue);
        SensorReport += String.Format("Результат перевірки виміру: {0}\n", (journal as JournalMember).Result);
        SensorReport += String.Format("\n");
        max_row--;
        if (max_row == 0) break;
    }
}
```

Рисунок 5.13 – Фрагмент коду для формування детальної інформації по обраному датчику

При формуванні відповіді використовуються можливості API Telegram з форматування тексту. Такими можливостями є виділення жирним шрифтом, або курсивом, підкреслення слів, або закреслення. Нажаль API Telegram не надає можливості виділення фрагменту тексту кольором, то вся інформація

надається чорним.

На рисунку 5.14 наведено фрагмент коду в якому виконується функція пошуку в базі даних інформації про останні п'ять рядків даних, що отримані від датчика.

Знайдена в БД інформація додається до характеристик датчика і виводиться на екран для аналізу користувачем.

```
String sql_robot = String.Format("Select * from BoundSensor where ID_Robot = '{0}'", ID_Robot_Last);
List<System.Object> rowsSensor = (List<System.Object>)DataBase.LoadList("sensorRobot.BoundSensorMember", sql_robot);

int key = (rowsSensor.Count + 1);
int row = key / 2;
int rest = key % 2;
if (rest > 0) row = row + 1;

KeyboardButton[][] kbc = new KeyboardButton[row][];
KeyboardButton[] keys = new KeyboardButton[key];

int num = 0;
foreach (System.Object item in rowsSensor)
{
    String NameSensor = DataBase.Find_Value(String.Format("Select Name_Sensor from Sensor where ID_Sensor = '{0}'",
        (item as BoundSensorMember).ID_Sensor), "Name_Sensor");

    keys[num] = new KeyboardButton(String.Format("{0} {1}", NameSensor, (item as BoundSensorMember).ID_Sensor));
    num++;
}
keys[num] = new KeyboardButton("На головну");

int key_num = 0;
for (int i = 0; i < row; i++)
{
    if ((key_num + 1) < key)
    {
        kbc[i] = new KeyboardButton[] { keys[key_num], keys[key_num + 1] };
    }
    else kbc[i] = new KeyboardButton[] { keys[key_num] };
    key_num = key_num + 2;
}

ReplyKeyboardMarkup keyboard_robot = new ReplyKeyboardMarkup(kbc)
{
    ResizeKeyboard = true
};

await botClient.SendTextMessageAsync(message.Chat.Id, SensorReport, parseMode: ParseMode.Html, replyMarkup: keyboard_robot);
return;
```

Рисунок 5.14 – Фрагмент коду для формування інформації по зібраним датчиком даним

Останній фрагмент коду виконується всякий раз, коли користувач натискає на кнопку «На головну» рис 5.15.

При виконанні даної функції на екрані формується привітання, та виводяться кнопки для початку роботи з програмою.

```

if (message.Text == "На головну")
{
    ReplyKeyboardMarkup keyboard_1_level = new ReplyKeyboardMarkup(new[] {
        new KeyboardButton[] { "Список роботів", "На головну" }
    })
    {
        ResizeKeyboard = true
    };

    String SensorReport = "";
    SensorReport += "Ласкаво просимо до системи автоматизованого моніторингу " +
        "стану роботів на базі сенсорної технології!";

    await botClient.SendTextMessageAsync(message.Chat.Id, SensorReport, replyMarkup: keyboard_1_level);
    return;
}

```

Рисунок 5.15 – Фрагмент коду для кнопки «На головну»

5.4 Опис результатів проведення експерименту

Розглянемо результати проведення експерименту з розробленою автоматизованою системою моніторингу стану роботів на базі сенсорної технології.

Для виконання досліджень розроблено програмний емулятор роботів. В програмі реалізовані три моделі:

- промисловий маніпулятор;
- робот укладальник;
- робот-сортувальник.

На рисунку 5.16 показано розроблено вікно емулятору роботів.

Для кожного робота виводиться його зображення та знизу наводяться параметри всіх датчиків, що до нього підключені.

Для кожного датчика виводяться останні отримані показання. На екран виводиться час вимірювання, отримане значення та результат опрацювання цих даних.

Дані оновлюються в реальному часі, та можна стежити, як вони змінюються в часі. Отримана з сенсорів інформація зберігається в базі та за вимогою сервера для обробки показчиків передаються до чат-боту, для відправки користувачеві. Чат бот та сервер обробки показчиків реалізовані на

одній програмній платформі.

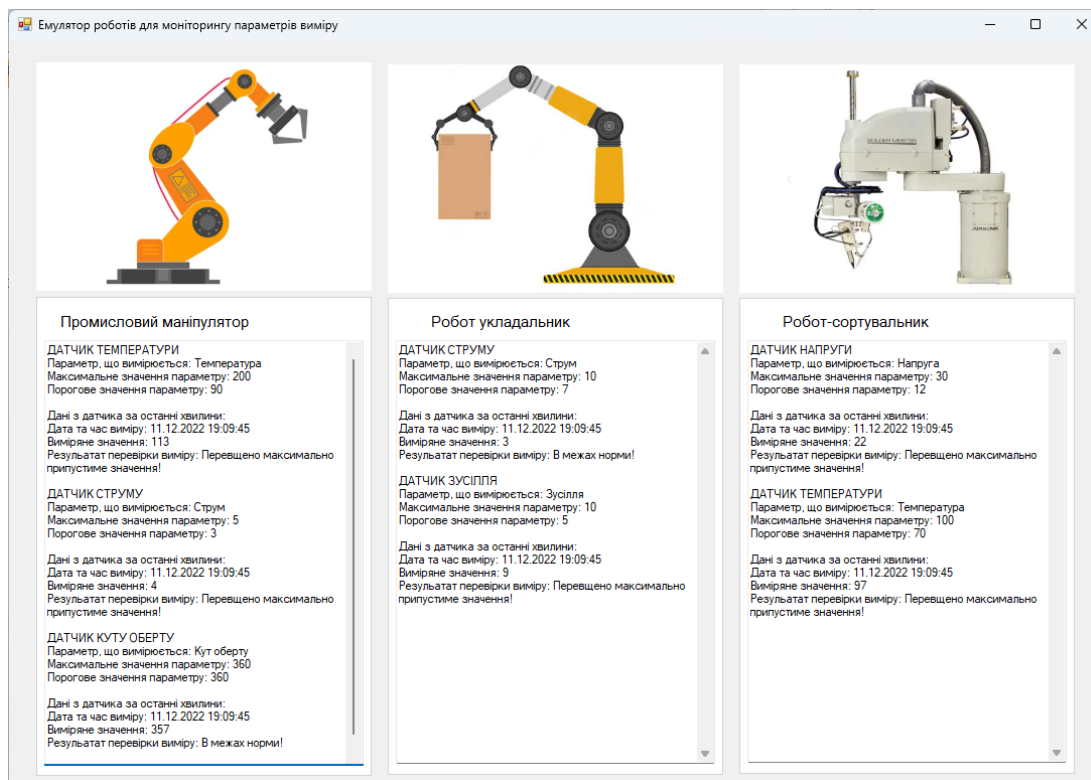


Рисунок 5.16 – Розроблено вікно емулятору роботів

Для початку роботи необхідно набрати в полі повідомлення команду «/start». В результаті обробки цієї команди сервер сформує відповідь та надсилає привітання. Результат обробки команди «/start» показано на рисунку 5.17.

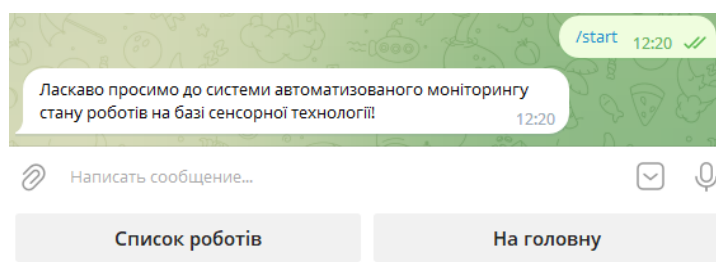
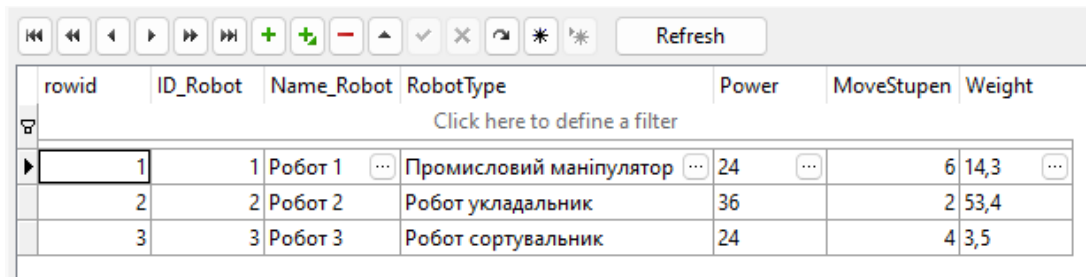


Рисунок 5.17 – Результат обробки команди «/start»

У користувача є можливість обрати команду «Список роботів» для перегляду всіх роботів, що зареєстровані в базі даних.

Наприклад, в нашій базі даних зареєстровано три роботи (рис. 5.18).



rowid	ID_Robot	Name_Robot	RobotType	Power	MoveStupen	Weight
1	1	Робот 1	Промисловий маніпулятор	24	6	14,3
2	2	Робот 2	Робот укладальник	36	2	53,4
3	3	Робот 3	Робот сортувальник	24	4	3,5

Рисунок 5.18 – Витяг з бази даних з інформацією про роботів

На рисунку 5.19 показано приклад відповіді серверу на команду «Список роботів».

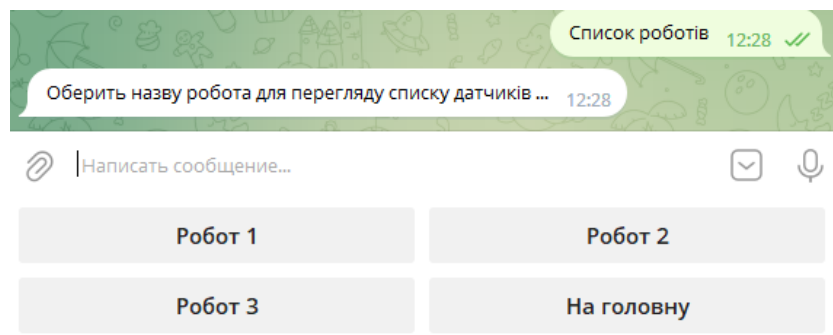


Рисунок 5.19 – Приклад відповіді серверу на команду «Список роботів»

З наведеного прикладу можна бачити, що список роботів відповідає зареєстрованому в базі даних.

Для прикладу, оберемо для дослідження першого робота. Для цього необхідно натиснути на кнопку «Робот 1».

В даному випадку з бази даних обирається список тих датчиків, що прив'язані до обраного роботу.

На рисунку 5.20 показано приклад відповіді від серверу.

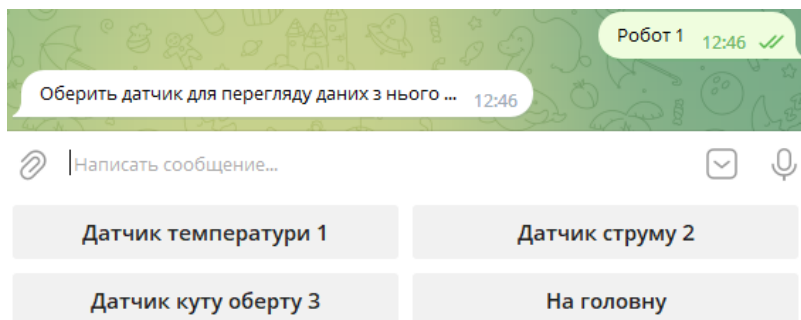


Рисунок 5.20 – Приклад списку датчиків, що прив’язані до обраного роботу

Для перегляду характеристик датчику та отриманих їм даних в процесі роботи необхідно натиснути на відповідну кнопку. На рисунку 5.21 показано приклад перегляду даних з датчику струму під номером 2.

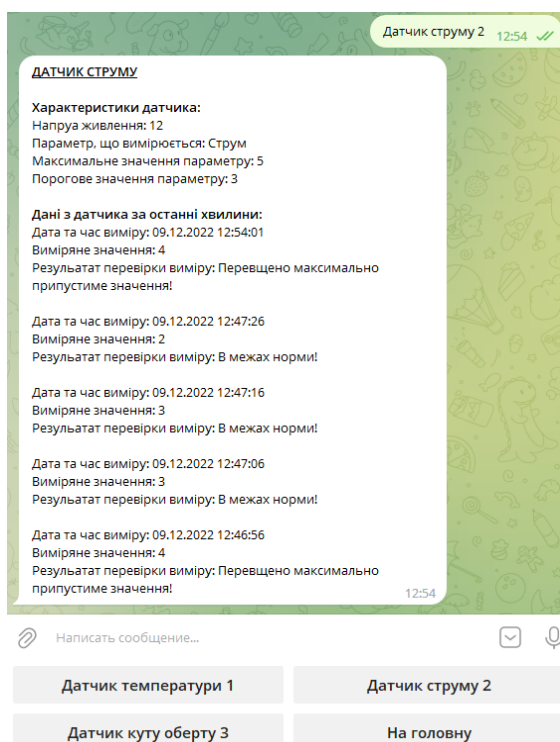


Рисунок 5.21 – Приклад перегляду даних з датчику струму під номером

Можна бачити, що на екран виводяться характеристика датчика, та останні п'ять показань з нього. Поряд з кожним виміром в базі даних зберігається також і результат опрацювання отриманих даних. Результат може бути позитивним «Показання в межах норми!» та негативним «Перевищено максимально припустиме значення!».

На рисунку 5.22 показано витяг з бази даних з отриманою інформацією від всіх зареєстрованих датчиків.

⏪

⏴

⏵

⏩

⏶

⏷

+

+

-

⬆

⬇

✖

↺

✳

✳

Refresh

rowid	ID_Journal	ID_Robot	ID_Sensor	DateMeasure	TimeMeasure	MeasureValue	PorogValue	Result		
Click here to define a filter										
1006	1006	1	1	1670533383		75	90	В межах норми!		
1007	1007	1	2	1670533383		1	3	В межах норми!		
1008	1008	1	3	1670533383		348	360	В межах норми!		
1009	1009	2	4	1670533383		4	7	В межах норми!		
1010	1010	2	5	1670533383		8	5	Перевщено максимально припустиме значення!		
1011	1011	3	6	1670533383		28	12	Перевщено максимально припустиме значення!		
1012	1012	3	7	1670533383		7	70	В межах норми!		
1013	1013	1	1	1670533383		119	90	Перевщено максимально припустиме значення!		
1014	1014	1	2	1670533384		1	3	В межах норми!		
1015	1015	1	3	1670533384		81	360	В межах норми!		
1016	1016	2	4	1670533384		1	7	В межах норми!		
1017	1017	2	5	1670533384		5	5	В межах норми!		
1018	1018	3	6	1670533384		8	12	В межах норми!		
1019	1019	3	7	1670533384		38	70	В межах норми!		
1020	1020	1	1	1670533384		39	90	В межах норми!		
1021	1021	1	2	1670533384		1	3	В межах норми!		
1022	1022	1	3	1670533384		310	360	В межах норми!		
1023	1023	2	4	1670533384		2	7	В межах норми!		
1024	1024	2	5	1670533384	...	5	...	5	...	В межах норми!

Рисунок 5.22 – Витяг з бази даних з отриманою інформацією від всіх зареєстрованих датчиків

Для повернення в головне меню необхідно обрати кнопку «На головну». В даному випадку всі дії можна повторити спочатку для обирання іншого роботу та його датчиків.

Таким чином, виконані експериментальні дослідження показують працездатність запропонованого методу.

5.5 Висновки до п'ятого розділу

В результаті виконання п'ятого розділу магістерської кваліфікаційної роботи було проведено експериментальні дослідження роботи програми. Для проведення експерименту необхідно було розробити програму. Зроблено ескіз екрану мобільного пристрою для відображення інформації. Розроблено алгоритм роботи програми та продемонстровано на рис. 5.6. Протестовано роботу програми, перевірено її працездатність та зроблено скріншоти, які показують, що виконані експериментальні дослідження підтверджують роботу створеного програмного методу.

6 ОХОРОНА ПРАЦІ

6. 1 Значення охорони праці у роботі фахівців з комп'ютерних технологій

Комп'ютерні технології вже давно є частиною нашого повсякденного життя, а норми роботи з ними добре вкоренилися в ІТ-компаніях, проте в нас на території України цей процес все ще триває, хоч і підходить до кінцевої завершальної стадії.

Звичайно, з першого погляду, робота за комп'ютером здається безпечною, але саме це зневажливе ставлення до неї може призвести до певних проблем для нашого здоров'я. Професія, яка вимагає сидячого, одноманітного положення, а отже програміста та інших фахівців ІТ-технологій пов'язана з колосальним розумовим напруженням. Розробники – це дуже зосереджені люди, навіть відволікаючись час від часу від роботи, вони продовжують думати про роботу. Нерідко відпочинком вони вважають заміну основної діяльності, наприклад, читання профільної літератури, верстку сайтів, вивчення нових мов програмування. Однак наш мозок не здатний до нескінченності сприймати виключно корисну інформацію, яку розробник прагне направляти в русло особистісного та професійного зростання.

Адже мозок людини не машина: він не може нескінченно зберігати і переробляти дані практично не втрачаючи продуктивності. Людина програє персональним електронним обчислювальним машинам за багатьма показниками тому для підтримки її працездатності потрібно зовсім інші умови.

В наш час багато ІТ-компаній обладнують свої офіси кімнатами відпочинку та зонами відпочинку, в яких забезпечують психофізіологічну розгрузку працівників. Адже вже давно нікого не здивуєш окремим робочим столом з ноутбуком. Тому, бажаючи підвищити продуктивність працівників, компанії змагаються, перетворюючи одноманітні та нудні офіси в креативні

простори, де нові ідеї народжуються без неймовірних зусиль.

Можна цілий день просидіти біля монітора, а у вечорі відчувати страшно втому, як від важкої фізичної праці. Однак це почуття помилкове і боротися з ним допоможе спорт. Відмінно бадьорить активний відпочинок, він розганяє кров, додає сил. Чимало програмістів відчують потребу в активному відпочинку на підсвідомому рівні, вибираючи спорт як хобі для проведення вільного часу.

При цьому не варто забувати, що умови праці програмістів також характеризуються можливістю впливу на них наступних небезпечних і шкідливих виробничих факторів: шуму; тепловиділень, причому шкоди організму можуть завдати не тільки високі, але і низькі температури; іонізуючих і неіонізуючих випромінювань: рентгенівське, інфрачервоне, електромагнітне випромінювання надвисоких частот діапазону; статичної електрики; недостатнє штучне освітлення [29].

За таких умов зростає роль та значення охорони праці, як системи правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці. Адже в кінцевому рахунку плоди науково-технічного прогресу можуть бути ефективними лише в тій мірі, в якій вони забезпечують людині безпеку, комфортність і зручність трудової діяльності.

6.2 Забезпечення безпеки життєдіяльності при роботі з ПК

Заходи щодо усунення небезпеки ураження електричним струмом зводяться до правильного розміщення устаткування та електричних кабелів. Інші заходи щодо забезпечення електробезпеки, збігаються з загальними заходами пожежно- та електробезпеки.

В якості профілактичних заходів для забезпечення пожежної безпеки слід

використовувати скриту електромережу, надійні розетки з пожежобезпечних матеріалів, силові мережі живлення устаткування виконувати кабелями, розрахованими на підключення в 3-5 разів більшого навантаження, включати й виключати живлення обладнання за допомогою штатних вимикачів. Треба регулярно робити очистку внутрішніх частин комп'ютерів, іншого устаткування від пилу, розташовувати комп'ютери на окремих неспалюваних столах. Для запобігання іскріння необхідно рідше встромляти і виймати штепсельні вилки з розеток [30].

Екран дисплея повинен бути розташованим перпендикулярно до напрямку погляду. Якщо він розташований під кутом, то стає причиною сутулості. Відстань від дисплея до очей повинна трохи перевищувати звичну відстань між книгою та очима. Перед екраном монітора, особливо старих типів, повинен бути спеціальний захисний екран. При його відсутності треба сидіти на відстані витягнутої руки від монітора. Ще одним моментом, який стосується зору, є необхідність створення неоднорідного поля зору. Для цього можна розвісити на поверхнях (стінах) плакати та картини, виконані у спокійних тонах. Наприклад, пейзажі.

Важливою є форма спинки крісла, яка повинна повторювати форму спини. Висота крісла повинна бути такою, щоб користувач не почував тиску на куприк або стегна. Крісло бажано обладнати бильцями. Його потрібно встановити так, щоб не треба було тягтися до клавіатури. Періодично користувачу необхідно рухатися, вчасно змінювати положення тіла і робити перерви у роботі.

При напруженій роботі за комп'ютером щогодини необхідно робити перерву на 15 хв через кожну годину і треба займатися іншою справою. Декілька разів на годину бажано виконувати серію легких вправ для розслаблення.

Режим праці та відпочинку при роботі з персональною електронно-обчислювальною машиною (ПЕОМ) залежить від категорії трудової діяльності.

Всі роботи з ПЕОМ ділять на три категорії. Перша - епізодичне зчитування і робота з інформацією не більше 2-х год за 8-год робочу зміну. Друга – зчитування інформації або творча робота не більше 4-х год за восьми годинну зміну. Третя – зчитування інформації або творча робота тривалістю більше 4-х год за зміну.

Якщо у приміщенні експлуатується більше одного комп'ютера, то треба врахувати, що на користувача одного комп'ютера можуть впливати випромінювання від інших, в першу чергу бокових, а також і задньої стінки сусіднього дисплея. Тому необхідний захист спеціальними фільтрами і щоб користувач розміщався від бічних і задніх стінок інших дисплеїв на відстані не ближче одного метра.

Отже, щоб запобігти негативним впливам необхідно знати й небезпечні сторони самого комп'ютера і правила безпечної роботи, знати засоби запобігання небезпек.

6.3 Висновки до шостого розділу

В результаті виконання шостого розділу магістерської кваліфікаційної роботи була зроблена охорона праці. Метою охорони праці є забезпечення здорових та безпечних умов праці. Завданням охорони праці є зведення до мінімуму вірогідності травмувань та виникнення професійних захворювань. Розглянуто значення охорони праці у роботі фахівців з комп'ютерних технологій та забезпечення безпеки життєдіяльності при роботі з ПК. Зроблено висновок до розділу.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено програмний модуль автоматизованої системи моніторингу стану роботів на базі сенсорної технології.

Під час роботи були виконанні наступні завдання: проведено аналіз предметної області та технічного завдання. Розглянуто класифікацію автоматизованих систем, актуальність моніторингу стану та його недоліки. Особливості давачів та переваги використання сенсорних технологій.

При плануванні розробки автоматизованої системи було зроблено функціональне забезпечення, а саме визначення вимог до датчиків системи. Інформаційне забезпечення, а саме аналіз програмного забезпечення та необхідні складові для розробки модуля.

При розробці метода було зроблено алгоритм роботи програми, математичне забезпечення системи, розроблена загальна схема організації процесу взаємодії між роботами в виробничому середовищі та системою віддаленого моніторингу.

Розроблено модель структури бази даних з сервером SQLite.

Виконані експериментальні дослідження, що можна побачити у п'ятому розділі, які показують працездатність роботи програмного модуля.

Таким чином, розроблений метод може використовуватися на різного роду виробництвах, де виникає необхідність автоматизованого моніторингу в режимі реального часу без значних витрат на реалізацію програмного забезпечення, а також може бути використаний під час проведення практичних і лабораторних практикумів в технологічних і навчальних лабораторіях в межах спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології та бути корисним для фахівців, що пов'язані з промисловою автоматизацією.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2021. 55 с.

2. Дипломне проектування для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»: Навч. посібник / І.Ш. Невлюдов та ін. К. : пр. Космонавта Комарова, 1. 2016. 245 с.

3. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. К.: ДП “УкрНДНЦ”. 2016. 30 с.

4. Положення про протидію академічному плагиату в ХНУРЕ [Електронний ресурс]. – Режим доступу: [www/ URL: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist. pdf](http://www.nure.ua/wpcontent/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf) – (дата звернення : 07.09.2022).

5. Стеценко К.В. Аналіз ітерферометрів для контролю форм поверхонь «Автоматизація та приладобудування» («Automation and Development of Electronic Devices») [Текст]: збірник студентських наукових статей / Харківський національний університет радіоелектроніки. Харків : ХНУРЕ, 2020. Вип. 1. С. 100 – 105.

6. Стеценко К.В. Системи слідкуючого привода промислового робота «Автоматизація та приладобудування» («Automation and Development of Electronic Devices») [Текст]: збірник студентських наукових статей /

Харківський національний університет радіоелектроніки. Харків : ХНУРЕ, 2020. Вип. 2. С. 136 – 140.

7. Стеценко К.В. Функціонування гнучких виробничих систем «Автоматизація та приладобудування» («Automation and Development of Electronic Devices») [Текст]: збірник студентських наукових статей / Харківський національний університет радіоелектроніки. Харків : ХНУРЕ 2020. Вип. 2. С. 185 – 188.

8. Іванов А. О. Теорія автоматичного керування: Підручник. Дніпропетровськ: Національний гірничий університет. 2015. 250 с.

9. Папушин Ю. Л., Білецький В. С. Основи автоматизації гірничого виробництва. Донецьк : Східний видавничий дім, 2016. 168 с.

10. Основи наукових досліджень: Навч. посібник / І.Ш. Невлюдов та ін. Кривий Ріг:Криворізький коледж НАУ. 2019. 396 с.

11. Виробничі процеси та обладнання об'єктів автоматизації. Збірник задач: Навчальний посібник / І.Ш. Невлюдов та ін. Кривий Ріг: КК НАУ. 2018. 332 с.

12. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов та ін. Кривий Ріг : Криворізький коледж НАУ. 2019. 366 с.

13. Невлюдов І.Ш. Техніко-економічне обґрунтування інженерних рішень: Підручник / Кривий Ріг : Криворізький коледж НАУ.2019. 448 с.

14. Гіль А. Промислові інтерфейси та протоколи передачі даних інтегрованих систем для автоматизованого управління в умовах Industry 4.0 / Гіль А., Чала О., Филипенко О. // Виробництво & Мехатронні Системи 2021: матеріали V-ої Міжнародної конференції, Харків, 21-22 жовтня 2021 р.: Харків, 2021. С.127 – 130.

15. Цимбал О.М., Бронніков А.І., Нгуєн Т.М, Бекметова А.О. Інтелектуальні виробничі системи та перспективи їх реалізації. Технологія приладобудування. 2016. № 1. С. 29-34.

16. Невлюдов І. Ш. Трансфер технологій у сучасній науці, освіті та виробництві в умовах четвертої промислової революції «ІНДУСТРІЯ 4.0» / Невлюдов І. Ш., Чала О. О., Олександров Ю. М. // Сучасний рух науки: тези доп. VIII міжнародної науково-практичної інтернетконференції, 3-4 жовтня 2019 р. Дніпро, 2019. Т.2. С. 604-608.

17. Колесник Л., Меркулов Р. Проблема прийняття рішень в умовах нечіткої інформації // Інформаційні системи та технології ІСТ-2020: матеріали 9- тої Міжнародної науково-технічної конференції (Харків, 17-20 листопада 2020 р.). Харків : ХНУРЕ, 2020. С. 95-98.

18. Бронніков А.І., Цимбал О.М., Фомовський Ф.М., Інтелектуальні засоби в системі керування виробничим агентом. Технологія приладобудування. Харків, 2014. № 2. С. 26 - 30.

19. Сучасні інформаційно-комунікаційні технології: Навчальний посібник / Г. Г. Швачич та ін. Дніпро: НМетАУ, 2017. 230 с.

20. Дроменко В.Б. Дослідження датчиків руху для застосування у автоматизованій системі управління зовнішнім освітленням вулиць/ В.Б. Дроменко, Т.О. Лавренюк, О.О. Кушнір // Вісник Інженерної академії України. 2019. № 4. С. 189-192.

21. Visual monitoring of the break surface of the installation connection of electronic equipment / Nevliudov I., Starodubcev N., Demska N., Omarov Sh. // Information system and innovative technologies in projekt management [Text]: Collective monograph edited byl. Linde, I. Chumachenko, V. Timofeyev. Kharkiv NURE, 2019. P. 258-270.

22. Термоопори з комутаційною головкою ДТСxx5 [Електронний ресурс]. – Режим доступу: <https://owen.ua/ua/datchyky/dtsnл5-termoopory-z-komutacijnoju-golovkoju> (дата звернення : 10.10.2022).

23. Цифровий датчик струму і напруги CJMCU-219 на мікросхемі

INA219 з шиною I2C [Електронний ресурс]. – Режим доступу: <https://arduino.ua/prod1661-cifrovoi-datchik-toka-i-napryajeniya-na-ina219-s-shinoi-i2c> (дата звернення : 12.10.2022).

24. Датчик кута повороту HRS100SSAB180 [Електронний ресурс]. – Режим доступу: <http://www.kosmodrom.com.ua/el.php?name=HRS100SSAB180> (дата звернення : 28.10.2022).

25. Датчик зусилля CZN-CP42 [Електронний ресурс]. – Режим доступу: <http://www.kosmodrom.com.ua/el.php?name=CZN-CP42> (дата звернення : 02.11.2022).

26. Муляр Ю. І., Репінський С. В. Автоматизація виробництва в машинобудуванні. Частина II. Навчальний посібник. Вінниця : ВНТУ, 2020. 123 с.

27. Підтримка життєвого циклу у виробничій інженерії [Текст] монографія / І.Ш. Невлюдов та ін. Кривий Ріг, Криворізький коледж НАУ, 2019. 252 с.

28. Невлюдов І.Ш., Токарева О.В. Автоматичне управління технологічними об'єктами [Текст] / навч. посібник. Київ: НАУ, 2018. 200 с

29. Яскілка В. Я., Олійник М.З. Конспект лекцій з курсу «Охорона праці в галузі» / Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. 56 с.

30. Бадищук В.І., Чихіра І.В. Охорона праці в галузі. Конспект лекцій [Текст] / Тернопільський національний технічний університет ім. Івана Пулюя. Тернопіль. 2016. 103 с