

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки
Освітньо-наукова програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження процесів управління проектом створення IoT- системи для контролю якості виробництва музичних інструментів»

Студента 2-го курсу групи УП-21

Богдана ДЗЕКУНОВ

(ім'я, прізвище)

(підпис студента)

Науковий керівник:

к.т.н., професор

(науковий ступінь, вчене звання)

Віктор МОРОЗОВ

(ім'я, прізвище)

(дата)

(підпис)

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри
технологій управління

(підпис)

Віктор МОРОЗОВ

(ім'я, прізвище)

(дата)

Київ 2025

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій**

Кафедра технологій управління
Освітній рівень Магістр
Спеціальність 122 Комп'ютерні науки
Освітньо-наукова програма «Управління проектами»

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Віктор МОРОЗОВ

“27” листопада 2024 року

**З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Дзекунов Богдан Юрійович

Група: УП-21

1. Тема кваліфікаційної роботи:

«Дослідження процесів управління проектом створення IoT-системи для контролю якості виробництва музичних інструментів.»

2. Строк подання студентом готової роботи – «19» 05 2025 р.

3. Цільова установка та вихідні дані до роботи: Ціллю роботи є дослідження методів управління проектом створення IoT системи контролю якості музичних інструментів, з акцентом на планування, управління ризиками та ресурсами. Вихідними даними є план реалізації проекту, створений із використанням сучасних інформаційних технологій.

4. Зміст роботи: У роботі проведено аналіз предметної області та обґрунтовано концепцію створення IoT системи контролю якості виробництва музичних інструментів. Визначено мету, завдання та очікуваний результат проекту, виконано SWOT, STEP та аналіз методом п'яти сил портрету. Побудовано, математичну модель та організаційну структуру проекту та команди. Розроблено архітектуру рішення, програмне забезпечення, базу даних, а також апаратну частину системи. Визначено структуру WBS, етапи життєвого циклу проекту,

календарне планування та часові віхи проекту. А також описано методологію управління ризику та методи їх подолання. Та зроблено висновки по роботі.

5. Перелік графічного матеріалу: дерево проблем, дерево цілей, концептуальна модель проекту, організаційна структура проекту, WBS-структура, схема управління ризиками, архітектура системи, , блок-схема роботи системи, концептуальна, логічна БД, візуалізація апаратної та програмної частин системи.

6. Календарний план виконання роботи

Назва частин роботи	Виконання роботи
Вивчення літературних джерел з предмету дослідження	22.01.25-21.02.25
Збір і вивчення матеріалів досліджуваного підприємства	23.01.25-01.02.25
Складання розгорнутого плану кваліфікаційної роботи	20.01.25-29.01.25
Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	23.01.25
Підготовка розділу 1	05.02.25-27.02.25
Підготовка розділу 2	28.02.25-14.03.25
Підготовка розділу 3	15.03.25-03.04.25
Підготовка розділу 4 «	06.04.25-19.04.25
Оформлення кваліфікаційної роботи	21.04.25-07.05.25
Передача кваліфікаційної роботи науковому керівникові	08.05.25
Передача кваліфікаційної роботи рецензенту для рецензування	12.05.25
Захист кваліфікаційної роботи	26.05.25-28.05.25

Дата видачі завдання «28» листопада 2024 р.

Керівник роботи _____ професор Віктор МОРОЗОВ
(посада, ім'я, прізвище)

(підпис)

Завдання прийняв до виконання студент групи УП-21

_____ Богдан ДЗЕКУНОВ
(ім'я, прізвище)

(підпис)

ЗМІСТ

ЗМІСТ	4
ВСТУП	7
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ УПРАВЛІННЯ ПРОЄКТАМИ СТВОРЕННЯ ІОТ-СИСТЕМ.....	9
1.1.Сутність і специфіка ІоТ-проектів у виробництві музичних інструменті.....	9
1.2. Аналіз предметної області та проблеми традиційних методів контролю якості.....	14
1.2.1 Характеристика предметної області.....	14
1.2.2 Проблеми традиційних методів контролю якості.....	16
1.3 Проведення аналізу конкурентів	19
1.3.1 Проведення аналізу галузі за методом 5 сил портрета.....	21
1.3.2 Проведення SWOT аналізу.....	24
1.3.3 Проведення STEP аналізу для маркетингового дослідження.....	26
1.3.4 Розробка дерева проблем, дерева цілей та логіко структурної схеми проекту	29
1.4 Ідентифікація первинних та вторинних зацікавлених сторін проекту	34
РОЗДІЛ 2. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІОТ-СИСТЕМИ КОНТРОЛЮ ЯКОСТІ.....	37
2.1 Розробка життєвого циклу проекту.....	37
2.2 Організаційна структура компанії та побудова команди проекту з розподілом відповідальності.....	41
2.2.1 Планування ресурсів проекту	46
2.2.2 Планування вартості проекту.....	47
2.2.3 Розробка концептуальних моделей інформаційної системи.....	54
2.3 Формалізація математичних моделей та постановка задачі у математичному вигляді.....	58
2.3.1 Цільова функція математичної моделі	59
2.3.2 Математична постановка задачі.....	60
2.3.3 Використання методів моделювання розроблених моделей. Ознайомлення із задачами контролю та управління інформаційними технологічними процесами	60

2.4 Розробка програмного забезпечення IoT-системи	61
2.4.1 Архітектура та функціональні вимоги	61
РОЗДІЛ 3. УПРАВЛІННЯ ТА КООРДИНАЦІЯ РОЗРОБКИ ПРОЄКТУ	66
3.1. Планування етапів реалізації IoT-системи.....	66
3.2. Управління ризиками та контроль виконання проектних робіт ...	82
3.2.1. Використання інструментів управління (Jira,MS Project).....	84
РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ ТА МЕТОДИ УПРАВЛІННЯ ПРОЄКТОМ.....	93
4.1. Аналіз досягнутих результатів і критерії оцінки успішності проекту	93
4.2 Виявлення проблемних аспектів та рекомендації щодо їх усунення	103
4.2 Узагальнення моделей і методів управління проектами, використаних у розробці IoT-системи.....	115
ВИСНОВКИ.....	122
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ:	126
ДОДАТКИ	129
ДОДАТОК А	129

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:
«Дослідження процесів управління проектом створення IoT-системи
для контролю якості виробництва музичних інструментів»

Студент: Дзекунов Богдан Юрійович

Науковий керівник: Морозов Віктор Володимирович

Рік захисту – 2025

Метою підготовки роботи є дослідження методів управління проектом розробки IoT-системи для автоматизованого контролю якості виробництва музичних інструментів з використанням сучасних технологій збору даних, аналітики та управління ризиками для підвищення ефективності та стабільності процесу виробництва.

Ціль проекту – створити IoT-систему, що забезпечує безперервний моніторинг параметрів якості інструментів (температури, вологості, вібрацій) на всіх етапах їхнього виготовлення, виявляючи дефекти та аномалії на ранніх стадіях виробництва.

Наукова новизна полягає у застосуванні концепції інтеграції апаратної та програмної частин для забезпечення безперервного моніторингу якості, з побудовою математичної моделі системи.

Кваліфікаційна робота складається з анотації, вступу, чотирьох розділів основної частини, висновків, списку використаних джерел та додатків. У першому розділі розглянуто проблему контролю якості на виробництві музичних інструментів, проаналізовано існуючі рішення та технології, сформовано SWOT-аналіз та визначено ключові вимоги до системи. У другому розділі побудовано концептуальну та математичну моделі IoT-системи, розроблено архітектуру рішення та описано компоненти апаратної частини. У третьому розділі визначено методологію управління проектом, побудовано організаційну структуру команди, визначено контрольні віхи, розроблено планування з використанням MS Project та Jira, проведено управління ресурсами та ризиками. У четвертому розділі розроблено програмне забезпечення та інтерфейси для моніторингу даних у реальному часі, описано аналітику на основі зібраних даних, побудовано логічну та фізичну моделі бази даних, а також проведено інтеграцію з хмарними платформами для збереження та обробки даних.

Робота містить _ сторінки без додатків, _ рисунків та _ таблиць.

Додатки складають _ сторінок.

Ключові слова: управління проектами, IoT, контроль якості, музичні інструменти, база даних, сенсори, управління ризиками, архітектура системи

ВСТУП

Актуальність дослідження: В умовах сучасного виробництва музичних інструментів, ефективне управління якістю є критично важливим для досягнення стабільної якості продукції та задоволення вимог споживачів. Одним з найважливіших елементів контролю якості є моніторинг параметрів на всіх етапах виготовлення інструментів, таких як температура, вологість, акустичні характеристики та вібрації. В умовах постійного вдосконалення технологій та збільшення вимог до точності виробів, застосування інноваційних технологій, зокрема Інтернету речей (IoT), дозволяє автоматизувати та удосконалити цей процес. IoT-системи забезпечують безперервний збір і аналіз даних у реальному часі, що дозволяє своєчасно виявляти дефекти на ранніх етапах виробництва, покращувати точність.

Дослідження, спрямовані на розробку IoT-систем для моніторингу якості в виробництві музичних інструментів, є важливими для впровадження новітніх технологій у виробничі процеси, а також для забезпечення високих стандартів якості при зниженні витрат і часу.

Метою кваліфікаційної роботи є створення та впровадження IoT-системи для автоматизованого контролю якості на виробництві музичних інструментів, що дозволить оптимізувати процеси контролю, виявляти дефекти на ранніх етапах виробництва та підвищити якість продукції.

Для досягнення поставленої мети у межах роботи необхідно вирішити такі завдання:

1. Проаналізувати існуючі технології та рішення для контролю якості в виробництві музичних інструментів, обґрунтувати вибір апаратних засобів, програмного забезпечення та комунікаційних протоколів для реалізації IoT-системи.
2. Розробити концептуальну модель IoT-системи, що включає планування етапів розробки та інтеграції компонентів, а також математичну модель для аналізу зібраних даних.

3. Побудувати організаційну структуру команди, визначити ролі та відповідальність учасників проекту, а також методи управління проектами, зокрема за допомогою сучасних інструментів управління (Jira, MS Project).

4. Розробити архітектуру IoT-системи, включаючи апаратну частину (сенсори, контролери) та програмне забезпечення (дані збору та обробки).

5. Створити інтерфейс користувача для моніторингу та віддаленого управління якістю інструментів у реальному часі.

6. Оцінити ймовірні ризики при впровадженні IoT-системи та розробити стратегії їх мінімізації.

7. Реалізувати прототип системи, провести тестування та оцінку ефективності впровадження.

Об'єктом дослідження є процеси моніторингу та контролю якості в виробництві музичних інструментів із застосуванням IoT-технологій.

Предметом дослідження є методи та інструменти розробки автоматизованої IoT-системи для збору та аналізу даних, моніторингу якості інструментів і керування виробничим процесом.

Наукова новизна роботи полягає у створенні інноваційної IoT-системи для контролю якості музичних інструментів, що інтегрує сенсори, програмне забезпечення та аналітику даних для автоматичного виявлення дефектів і порушень стандартів якості. Також запропонована модель управління проектами для створення таких систем, що включає застосування методології управління ризиками та ресурсами.

Практична цінність роботи полягає у створенні прототипу IoT-системи, що дозволяє покращити точність контролю якості, зменшити час на виявлення дефектів, а також підвищити ефективність виробництва музичних інструментів. Розроблена система може бути впроваджена в серійному виробництві для автоматизації моніторингу якості та забезпечення стабільності параметрів продукції.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ УПРАВЛІННЯ ПРОЄКТАМИ СТВОРЕННЯ ІОТ-СИСТЕМ

1.1. Сутність і специфіка ІоТ-проектів у виробництві музичних інструментів

ІоТ системи мають ризики та умови реалізації тому давайте розглянемо найголовніші вимоги які мають бути дотримані для реалізації ІоТ проекту:

1 Локація

Локація – це один з найважливіших показників який варто враховувати тому що місце де саме буде розгорнуто ІоТ систему дуже часто визначає інші вимоги до апаратного забезпечення та значно звужують варіанти вибору. На щастя, вимоги до розташування в ІоТ зазвичай можна визначити як комбінацію двох факторів: географічне покриття та мобільність. Наприклад географічний фактор це те яку географічну одиницю має покривати датчик датчики можна розділити на 2 типи за цим показником:

Локальний – прикладом такого датчику може бути датчик відстеження температури на виробництві.

Глобальний - сенсори екологічні.

Мобільність - тобто чи датчик є статичним – закріпленим за певною локацією або мобільним – датчик який потрібен для відстеженням локації, та переміщення.

2. Вартість

Вартість реалізації розгортання та підтримки ІоТ системи є одним з найголовніших вимог, але його доволі часто недооцінюють. Оскільки будь яка обрана та протестована технологія яку можна додати на виробництво, не матиме ніякого сенсу, якщо її розгортання та підтримка буде занадто дорогою то цю систему нема сенсу реалізовувати і час та ресурс який був витрачений на підбір технологій втрачений за дарма.

Для правильної оцінки вартості можна використовувати 3 підходи:

Перший підхід - Вартісні цільові показники на основі цінності ІоТ: Якщо починати з точки зору цінності яку принесе імплементація ІоТ системи це буде

одним з найкращих варіантів але водночас й найскладнішим, оскільки потрібно мати абсолютне чітке бачення бізнес цілей компанії, наприклад знизити витрати на 10% .

Другий підхід це - Вартість IoT на основі доступності цей підхід є більш поширеним – першим кроком тут є розуміння того скільки ваша компанія готова заплатити за збір даних. Часто це означає що завдання оцінки цінності повністю лягає на виробництво яке хоче замовити послуги, оскільки вони знають, скільки готові заплатити за інформацію, яку буде збиратися, тому все, що потрібно зробити – це взяти цю суму яку надав клієнт , відняти ваш прибутковий відсоток, і вийде цільова вартість[4].

Третій підхід - коли компанії просто надають бюджет для роботи: хоча це не оптимальна стратегія, це допомагає звузити варіанти можливих систем.

У світі хмарних систем і смартфонів де потік даних ніколи не зупиняється можна хотіти дуже багато різних показників. Але з IoT потрібно бути обережними щодо кількості збираєм даних . Багато джерел даних мають реальні витрати, якщо не виключно через вартість додаткових сенсорів, то через додаткові байти, що передаються при кожному запиті.

Працюючи з IoT треба ретельно аналізувати , що потрібно, що може бути корисним у майбутньому, а що зовсім не потрібно . Наприклад, чи потрібні вам оновлення температури щохвилини? Чи ці дані вже записуються десь в іншій системі? Чи достатньо сповіщень, коли певний поріг досягнуто?

Однією з найбільших перешкод для IoT-проектів є їх розгортання та масштабування. Потрібно врахувати, як нові пристрої вписуватимуться у вже існуючі бізнес-процеси, та що потрібно буде впровадити для їх управління. Наприклад, IoT-система що використовується для моніторингу якості виготовлення корпусів гітар. Оскільки корпуси є однією з найважливіших складових, від яких залежить як акустичні характеристики інструмента, відстеження якості деревини є дуже важливою.

Для IoT пристрою для відстеження якості контролю виробництва корпусу інструменту:

Низька вартість пристроїв та компактність: Оскільки на виробничій лінії виготовляється дуже багато корпусів, кожен з яких має бути перевірений, IoT-пристрої повинні бути недорогими. Пристрої повинні бути компактними та інтегрованими в виробничу лінію, щоб не заважати роботі операторів.

Вбудована інспекція без переривань виробничого процесу: Пристрої для контролю якості корпусів гітар мають бути закріплені на виробничій лінії де вони проводять вимірювання та роблять фото деталі у реальному часу. Завдяки такому розташуванню перевірка буде відбуватися швидко.

Живлення пристроїв: Через невеликий розмір IoT-пристроїв часто виникає проблема роботи від акумулятора. Але у виробництві музичних інструментів існують регулярні цикли обслуговування, під час яких можна здійснювати заміну батареї або пристроїв[1].

У виробництві музичних інструментів IoT-технології мають особливе значення, оскільки якість інструментів напряму залежить від контролю якості сировини та процесу виготовлення. Наприклад, таких параметрів як геометричні параметри корпусу, розташування ладів і звісно звучання інструменту якщо ці параметри зможуть бути під контролем IoT пристроїв це набагато збільшує вірогідність якісного фінального продукту.

Першим аспектом який варто контролювати при виготовленні музичного інструменту це якість сировини та її обробка. Деревина що використовується при виготовленні корпусу інструменту, має бути стабільною за вологістю, температурою та структурою. Якщо використовувати IoT датчики температури та вологості для моніторингу цих показників це може мінімізувати рівень виробничого браку.

Другим аспектом який варто відстежувати при виготовленні музичних інструментів це параметр геометрії корпусу за допомогою візуального контролю.

Для здійснення такого контролю можна використовувати - прецизійне вимірювання – це використання лазерних 3D-сканерів і камер високої

роздільності що дозволяє швидко перевести геометрію корпусу інструменту у цифровий формат (фото) порівняти їх із CAD-моделями та виявити проблеми.



Рис 1.1 CAD модель музичного інструменту.

Третім показником що є критично важливим при виготовленні інструменту є звук. Для того щоб протестувати звучання інструменту можна здійснювати акустичний контроль[2].

Зараз дуже часто на виробництві акустичний контроль здійснюється майстрами з налаштування інструментів але така оцінка якості звучання є суб'єктивною. Тому щоб досягти стабільного результату та показників варто використовувати Мікрофони та датчики вібрації акселерометри, лазерні віброметри які потрібні для вимірювання звуку інструменту під час тестування. Також коли звук певного інструменту був записаний за допомогою алгоритмів машинного навчання та комп'ютерного зору (наприклад, платформи Landing AI) можна аналізувати звукові спектри та виявляти навіть найдрібніші відхилення, що впливають на якість звучання.



Рис 1.2 Зображення віброметра.

Управління проектами, що зосереджені на розробці Інтернету речей (IoT), має свої специфічні аспекти та виклики, що відрізняє його від традиційних ІТ-проектів. Управління IoT проектами можна розкласти на такі рівні[6]:

Рівень інтеграції різних пристроїв є складним

IoT охоплює широкий спектр пристроїв, від простих датчиків до складних систем, таких як автономні автомобілі та розумні будинки. Ці пристрої можуть використовувати різні комунікаційні протоколи, формати даних і мати різний рівень обчислювальної потужності, що робить їх інтеграцію в єдину систему вкрай складною.

Масштабованість та обробка великих даних

IoT системи генерують великі обсяги даних у режимі реального часу. Це вимагає від команди яка займається розробкою архітектури, здатності швидко масштабуватися та ефективно обробляти дані. Використання хмарних технологій та розподілених систем стає необхідністю для забезпечення надійності та доступності даних. Забезпечення безпеки та конфіденційності

Також не можна забувати що при підключенні великої кількості пристроїв до мережі підвищується ризики кіберзагроз. Захист даних та

забезпечення конфіденційності користувачів є критичними аспектами управління IoT-проектами. Необхідно впроваджувати надійні механізми шифрування, автентифікації та управління доступом.

Управління вимогами та прогнозування ризиків

Завдяки використанню ШІ в управлінні вимогами, проектні команди можуть автоматизувати процеси збору та аналізу вимог, що підвищує точність та ефективність. Прогнозна аналітика дозволяє виявляти потенційні ризики та проблеми на ранніх етапах розробки, що сприяє реалізації проекту.

Різноманітність застосувань та галузеві особливості

IoT використовується в різних галузях:

Від сільського господарства до промисловості та галузі охорони здоров'я. Кожна з цих областей має унікальні вимоги та проблеми для керівників проектів з глибоким розумінням тематичних областей, щоб адаптувати свій підхід до управління відповідно до характеристик секторів застосування.

1.2. Аналіз предметної області та проблеми традиційних методів контролю якості

1.2.1 Характеристика предметної області

У виробництві музичних інструментів якість є дуже важливим аспектом, адже навіть невеликі проблеми в конструкції, обробці деревини або покритті можуть вплинути на звукові характеристики готового інструменту. Музичні інструменти, потребують високої точності та якості на всіх етапах виробництва. Задля аналізу проблеми ми розіб'ємо нашу проблему якості виробництва музичних інструментів на підпункти. З основних складових можна виділити Технологічні процеси, проблема стандартизації, та вартісна ефективність та інші. За для оцінки та аналізу ми будемо використовувати, різні методи аналізу такі як SWOT аналіз, аналіз ризиків та інші [1].

Обробка деревини: Якість деревини тут варто звернути увагу на рівень вологості, наявність сучків, мікро тріщин ці нюанси деревини визначають якість звучання інструменту.



Рис 1.3 Зображення процесу обробки деревини

Геометрія деталей: Інструмент складається з багатьох деталей корпусу, грифу, ладів та інших елементів тому контроль форми виготовлених частин напряду впливає на чистоту звучання інструменту[8].



Рис 1.4 Зображення компонентів інструменту гітари

Поверхнева обробка: Вигляд інструменту не менш важливий за його звучання тому при виготовленні інструменту варто звернути увагу на точність покриття.

Управління якістю є ключовою складовою реалізації будь якого проекту і IoT проекти не є виключенням. З огляду на складність IoT-рішень, до яких входять як апаратні, так і програмні компоненти, управління якістю відрізняється від типового управління якістю в IT проектах.

1.2.2 Проблеми традиційних методів контролю якості

Традиційно перевірка інструменту відбувалася в ручну, це за собою несе деякі недоліки такі як:

Суб'єктивність перевірки: До ручних перевірок можна віднести зорову оцінку та прослуховуванні виробу – ці перевірки можуть бути не систематичними. Тут вступає людський фактор через втому або просто не цікавість до роботи та недостатню кваліфікацію що сприяє браку на виробництві який потім вплине на товар.

Затримки у виявленні дефектів: Ручний контроль якості здійснюється після завершення виробництва тому дефекти можуть бути виявлені запізно що призводить до високих витрат на переробку або відмову готової продукції.

Проблеми масштабування: Коли зростає виробництво традиційні методи часто не встигають забезпечити необхідний рівень контролю якості, що призводить до великої кількості дефектів і, відповідно, зниження конкурентоспроможності продукції.

Відсутність системності: Традиційні методи не дозволяють створити єдину базу даних, в якій записуються дефекти якості кожного виробу. Це майже унеможлиблює аналіз тенденцій та прогнозування дефектів.

Технологічний процес Іншою проблемою традиційних методів виробництва є технологічний процес він складається з таких етапів як:

Формування: Первинна обробка матеріалів, що може включати різання, гнуття та інші форми механічної обробки.

Збірка: Проблема збірки полягає у тому що потрібна висока точність, щоб уникнути зазорів які впливають на якість звуку.



Рис 1.5 Процес збірки музичного інструменту

Оздоблення: Включає покриття, полірування та інші декоративні аспекти, які можуть вплинути на звук. Власне головною проблемою етапу технологічного процесу є наявність точного та прицільного обладнання, за для коректного виконання функцій та задач поставлених на виробництві.

Стандартизація в процесі є дуже важливим аспектом забезпечення послідовності в якості продукту вона складається з переконання в тому що всі матеріали відповідають стандартам якості. Та розробці інструкції та параметрів для кожного етапу виробництва. Основною проблемою цього етапу можна назвати відсутність єдиних стандартів та норм. Без чітких, узгоджених стандартів якості, виробництво може бути неконсистентним, що призводить до значних варіацій у якості готових інструментів.

Іншим аспектом можна назвати якість матеріалів. Оскільки виробництво музичних інструментів сильно залежить від якості використаних матеріалів. Основними матеріалами для виготовлення інструментів є деревина та метал:

- Деревина: Для струнних інструментів, як скрипки або гітари, важливо використовувати деревину, яка має відповідні акустичні властивості. Від якості деревини залежить резонанс інструменту.

- **Метал:** Для духових інструментів, таких як труби чи саксофони, важливо використовувати метали, які не тільки створюють правильний звук, але й стійкі до корозії та інших пошкоджень. Основна проблема це нездатність забезпечити стабільну якість матеріалів оскільки різні партії дерева або металу можуть мати різні фізичні та акустичні властивості, що може негативно вплинути на якість готового інструменту



Рис 1.6 Голова гітари

Контроль якості також є невід'ємною частиною процесу виготовлення муз інструменту оскільки він потрібен на усіх етапах виробництва він складається з:

Вхідний контроль: Перевірка якості матеріалів перед використанням.

Процесний контроль: Моніторинг процесів виробництва для виявлення та виправлення помилок на ранніх стадіях.

Фінальний контроль: Остаточні перевірки готового продукту перед відправкою клієнтам. Основною проблемою можна назвати неналежну систему контролю якості. Якщо контроль якості не виконується на всіх етапах виробництва, помилки можна не помітити, і вони є критичними для функціональності та терміну служби інструменту.

Вартісна ефективність: Важливо збалансувати витрати на контроль якості з загальною вартістю продукції за для цього можна виділити такі кроки як:

Оптимізація витрат: Треба знайти способи мінімізації витрат без зниження якості продукту.

Ціноутворення: Встановлення конкурентоспроможних цін, які відображають вартість виробництва і контролю якості. Основну проблему цього етапу можна описати як збалансованість якості та вартості. Підвищення стандартів якості може збільшити собівартість продукції, що впливає на ціноутворення та конкурентоспроможність продукції на ринку

1.3 Проведення аналізу конкурентів

В цій частині роботи ми проведемо аналіз конкурентів наявних на ринку IoT та музичних інструментів. Та розпишемо основних конкурентів, їх переваги та недоліки.

До списку можна віднести такі компанії:

- Fender
- TimmerGrove
- Prophecy IoT
- Pepperl+Fuchs

- Siemens

Таблиця 1.1

Аналіз конкурентів

Назва компанії	Переваги	Недоліки
1	2	3
Fender	Один з найвідоміших брендів музичних інструментів з інтеграцією IoT у виробництво гітар, використовує хмарні технології для збору даних про використання інструментів.	Обмежена доступність технологій для інших виробників, висока вартість рішень.
Timbergrove	Використовує IoT та машинне навчання для моніторингу параметрів якості під час виробництва, автоматизація процесів.	Складність впровадження у невеликих виробництвах, потреба великих інвестицій.
Prophecy IoT	Пропонує рішення для автоматизації контролю якості з використанням IoT, що значно знижує	Не адаптований під специфіку виробництва музичних інструментів.

	витрати та підвищує ефективність.	
--	-----------------------------------	--

Продовження таблиці 1.1

1	2	3
Pepperl+Fuchs	Відомий виробник промислових сенсорів та компонентів для автоматизації, широкий асортимент продукції.	Не спеціалізується на сфері муз інструментів, потребує додаткової адаптації.
Siemens (MindSphere)	Платформа для промислового IoT, спеціалізується на зборі та аналізі даних у реальному часі, можна інтегрувати з різними системами.	Складно інтегрувати потрібні спеціалізовані знання.

В цій таблиці ми навели результати аналізу конкурентів які є наявні на ринку світу. Та описали головні переваги та недоліки, щоб підкреслити потребу у новому продукті який зможе закрити деякі з цих недоліки[2].

1.3.1 Проведення аналізу галузі за методом 5 сил портрета

За для аналізу нашої галузі ми використаємо метод 5 сил портрета, в якому опишемо параметр впливу, рівень його значення, його характеристику та потрібні рекомендації.

Таблиця 1.2

Метод 5 сил портрета

Параметр	Значення Рівня	Характеристика	Рекомендації
1	2	3	4
Загроза нових конкурентів	Середній/високий	Галузь має бар'єри для входу, такі як високі витрати на розробку технологій IoT та знання специфіки музичної індустрії. Проте інтерес до IoT-технологій зростає що знижує бар'єр входу.	Треба фокусуватися на унікальності технології та інтеграції її з іншими системами.
Загроза замінників	Низька/середня	Традиційні методи контролю якості (ручна перевірка) є менш ефективні і потребують більше часу.	Треба займатися популяризацією переваг IoT-рішень, таких як автоматизація та швидкість

			виявлення дефектів.
--	--	--	------------------------

Продовження таблиці 1.2

1	2	3	4
Вплив постачальників	Середній	Виробництво залежить від постачальників компонентів для IoT-систем. Через дефіцит ціни та доступність компонентів можуть вплинути на вартість продукту.	Треба встановлювати партнерські відносини з кількома постачальниками для того щоб не залежати від одного постачальника.
Вплив покупців	Високий	Виробники музичних інструментів мають дуже сильний вплив на ринок та диктування тенденцій технологій, до технологій.	Треба враховувати потреби великих виробників музичних інструментів і приспособувати IoT технології до ринкових потреб.

Продовження таблиці 1.2

1	2	3	4
Конкуренція в галузі	Високий	Високий рівень конкуренції для стартапів серед великих компаній (Siemens, РТС).	Треба інвестувати у нові технології та підтримувати швидкий розвиток, щоб залишатися конкурентоспроможними.

Провівши аналіз ризиків та впливу ми можемо зрозуміти що буде мати найбільший вплив на наш проект та на які ризики першочергово треба звернути увагу. Найголовнішими ризиками для нас є конкуренція у галузі у порівнянні з великими компаніями, вплив великих компаній на тенденції ринку. Також меш впливовими для нас будуть загрози заміників через роботу з постачальниками, та загроза нових конкурентів через популяризацію сфери ІоТ та пониження порогу входу.

1.3.2 Проведення SWOT аналізу

Щоб проаналізувати можливості та сильні сторони нашого проекту а також слабкі сторони та загрози, та щоб ця інформація могла використовуватись для прийняття управлінських рішень ми побудуємо SWOT матрицю в якій пояснимо цю інформацію.

Таблиця 1.3

SWOT аналіз

Сильні сторони & Можливості	Сильні сторони & Загрози
1	2

Продовження таблиці 1.3

1	2
<p>1. Інноваційне рішення + зростаючий попит на автоматизовані системи</p> <p>2. Підвищення точності виявлення дефектів + потенціал для розширення</p> <p>3. Скорочення часу на перевірку якості + інтеграція з системами машинного навчання</p> <p>4. Можливість інтеграції з іншими системами + підвищення продуктивності</p>	<p>1. Висока ефективність автоматизації + конкуренція з великими компаніями</p> <p>2. Підвищення ефективності виробництва + потреба в постійному вдосконаленні</p> <p>3. Інтеграція з іншими системами + загроза з боку нових технологій</p> <p>4. Зниження витрат на матеріали + тиск від конкурентів</p>
Слабкі сторони & Можливості	Слабкі сторони & Загрози
<p>1. Висока вартість впровадження системи + зростаючий попит на автоматизацію</p> <p>2. Залежність від постачальників компонентів +</p>	<p>1. Залежність від постачальників компонентів + висока конкуренція на ринку IoT</p> <p>2. Потреба у постійному обслуговуванні + зміна технологій на ринку</p>

<p>потенціал для розширення на нові ринки.</p> <p>3. Потреба у постійному оновленні програмного забезпечення + можливість інтеграції з іншими технологіями</p>	<p>3. Висока конкуренція на ринку IoT рішень + витрати на інновації та дослідження</p> <p>4. Вартість стартового капіталу + проблема з захистом даних</p>
--	---

Цей аналіз допоможе нам зрозуміти сильні та слабкі сторони проекту, а також можливості і загрози для розвитку. Ці можливості та загрози дозволяють розробити стратегії для управлінських рішень.

1.3.3 Проведення STEP аналізу для маркетингового дослідження

В цій частині роботи ми проведемо маркетинговий STEP аналіз щоб оцінити вплив різних факторів таких як політичні, технологічні соціальні та інші на наш проект.

Таблиця 1.4

Фактори впливу

Політичні	Економічні
1	2
<p>Податкові стимули для інновацій</p> <p>Законодавчі вимоги щодо захисту даних</p> <p>Бюрократія</p>	<p>Зростання попиту на технології IoT</p> <p>Вартість робочої сили</p> <p>Інфляція та курс валют</p> <p>Ринок і попит на автоматизацію</p> <p>Конкуренція з подібним ПЗ</p>
Соціально-Культурні	Технологічні
<p>Популярність технології IoT</p> <p>Освітні програми для IoT-фахівців</p>	<p>Технічний розвиток в IoT</p> <p>Безпека даних в IoT</p>

Готовність до автоматизації	Інтеграція з іншими системами
Екологічні стандарти та сталий розвиток	

Далі ми проведемо оцінку факторів які спричинятимуть найбільший вплив на наш проект:

Таблиця 1.5

Види впливу

Політичні		Економічні	
Фактор	Вага	Фактор	Вага
Розвиток IoT індустрії	+2	Зростання попиту	+2
Бюрократизація	1.75	Конкуренція з подібним ПЗ	-2
Соціально-Культурні		Технологічні	
Фактор	Вага	Фактор	Вага
Готовність до автоматизації	+2	Доступність технологій	+2

Таблиця 1.6

STEP аналіз

Фактори	Зміни в галузі	Зміни в організації	Дії
1	2	3	4
Політичні	1. Розвиток IT-індустрії в Україні, зокрема у сфері IoT. 2. Бюрократизація процесів впровадження	1. Популяризація IoT-технологій у виробництві музичних інструментів. 2. Неактуальність впровадження	1. Підтримка з боку уряду для впровадження інноваційних технологій в галузі музичних інструментів.

	нових технологій в Україні.	нових технологій	2.Спрощення регуляційних вимог
--	-----------------------------	------------------	--------------------------------

Продовження таблиці 1.6

1	2	3	4
Економічні	<p>1.Зростання вартості компонентів IoT-систем.</p> <p>2. Зростання конкуренції з подібними IoT-системами.</p>	<p>1. Зростання попиту на інтелектуальні технології для контролю якості виробництва муз інструментів.</p> <p>2.Вдосконалення технологій для підвищення якості виробництва.</p>	<p>1.Аналіз ринку постачальників для зменшення витрат.</p> <p>2.Моніторинг ринку конкурентів і для створення інноваційного продукту.</p>
Соціальні-культурні	<p>1. Позитивне ставлення до інновацій.</p> <p>2. Адаптація ринку до нових технологій.</p>	<p>1.Популяризація ідеї інтеграції IoT в традиційне виробництво.</p> <p>2. Інтеграція нових технологій у виробництво для покращення контролю якості.</p>	<p>1. Організація навчальних семінарів для персоналу.</p> <p>2. Розвиток навчальних програм для підвищення кваліфікації працівників.</p>

Продовження таблиці 1.6

1	2	3	4
<p>Технологічн і</p>	<p>1.Розвиток IoT технологій для автоматизації процесів.</p> <p>2. Зниження вартості компонентів для IoT-систем.</p>	<p>1. Потрібність у постійному оновленні технологій для забезпечення якості.</p> <p>2.Потрібність постійного вдосконалення компонентів IoT для досягнення високої точності та надійності системи.</p>	<p>1. Інвестування в нові технології для підтримки конкурентоспроможності .</p> <p>2. Постійна оцінка вартості компонентів та пошук нових постачальників для зниження витрат.</p>

STEP аналіз для проекту показав нам що, є можливості для розвитку через розвиток у галузі IoT та позитивне ставлення до інновацій проект стикнутись з проблемами через бюрократичні перепони та високі витрати на компоненти. Також Ринок має високий попит на інтеграцію нових технологій, але щоб конкурувати в галузі треба постійно вдосконалювати систему [7].

1.3.4 Розробка дерева проблем, дерева цілей та логіко структурної схеми проекту

Побудуємо дерево проблем нашого проекту:

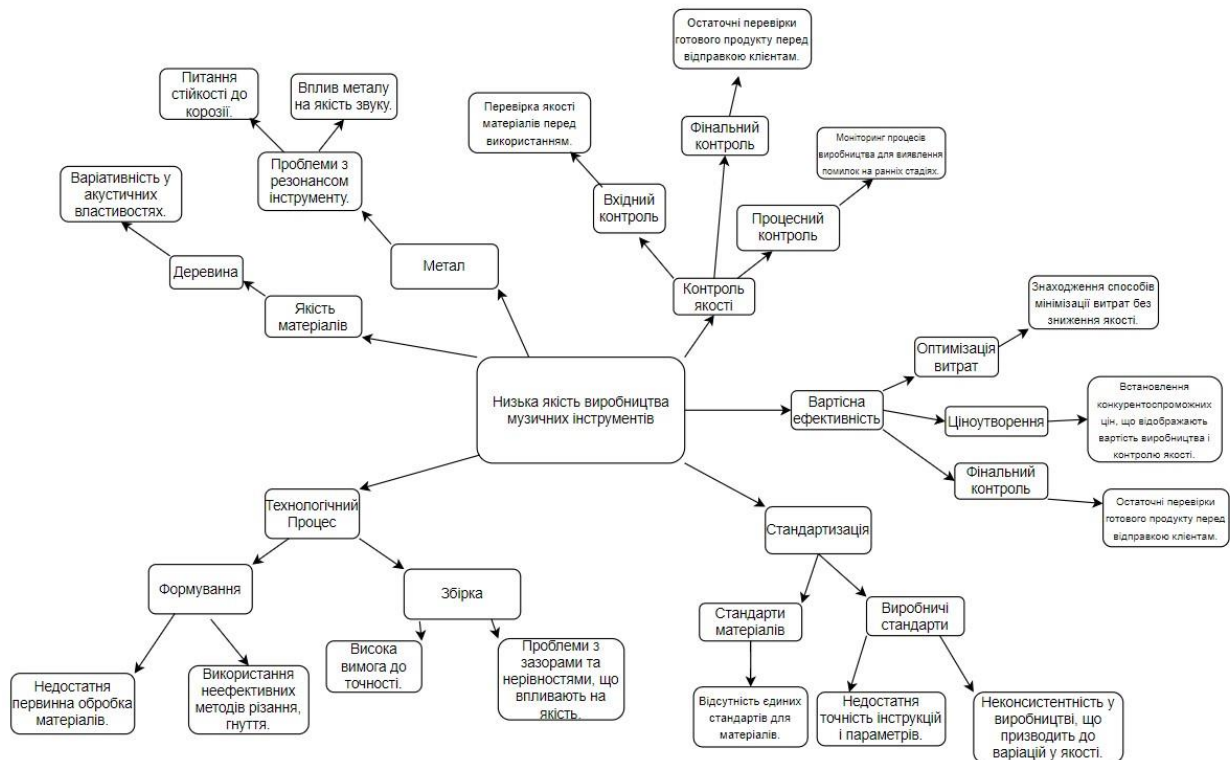


Рис 1.6 Дерево проблем.

Виробництво музичних інструментів складається з кількох етапів. Однією з основних проблем первинна обробка матеріалів, наприклад різання і гнуття. Також збірка потребує високої точності, оскільки навіть найменші зазори чи нерівності впливають на звук інструменту. Наступним можуть бути проблеми на етапі оздоблення, такі як покриття та полірування. Стандартизація є ключовим елементом для забезпечення стабільної якості продукції. Проблема може виникнути через відсутність єдиних стандартів матеріалів, що в свою чергу приводить до різної якості кінцевих продуктів.

Тому потрібні чіткі виробничі інструкції і стандарти для продукції. Іншим важливим фактором є якість матеріалів, адже від якості деревини залежить звук який матиме готовий інструмент. Іншим основним компонентом є метал наприклад у духових інструментів він може піддаватися корозії чи що вплине на його функціонал і звук.

Після цього ми виділимо основні цілі проекту та побудуємо дерево цілей. Під час створення інструменту особлива увага приділяється якості готового

продукту. Традиційні методи контролю якості, як правило, засновані на суб'єктивних ручних перевірках. Але автоматизація цих процесів необхідна, оскільки вимоги до якості інструментів є дуже високі, також робота з деревиною є дуже кропіткою що теж має вплив на точності та складності реалізації якісного інструменту. Технології Інтернету речей (IoT) дозволяють вести моніторинг усіх процесів на виробництві безперервно, та без залучення людського фактору що дозволяє зменшити витрати на ревізії.

Основною місією проекту: Розробки IoT системи контролю якості виробництва музичних інструментів полягає у створенні інноваційної системи яка буде використовувати IoT технології для покращення якості та відстеження процесу виробництва музичних інструментів що надасть можливість автоматизувати процеси виробництва та отримати якісний аналіз матеріалів що використовуються для виготовлення музичних інструментів[3].

Цілі проекту: Інтеграція IoT датчиків: Імплементация IoT-датчиків для моніторингу параметрів виробництва, таких як вологість, температура, тиск інструментів. Автоматизація процесів контролю: Автоматизація збору даних про якість матеріалів та процесів на різних етапах виробництва для покращення аналізу та можливості зреагувати на проблему яка виникла.

Оптимізація виробничих процесів: Використання даних з IoT-датчиків для оптимізації виробничих параметрів, забезпечуючи оптимальні умови для виготовлення якісних музичних інструментів. Розробка моделей для прогнозування потенційних виробничих помилок та планування профілактичних заходів. Забезпечення відповідності до стандартів виготовлення: Оптимізація процесів виробництва та імплементация IoT технологій допоможе фабрика відповідати усім стандартам виготовлення музичних інструментів.

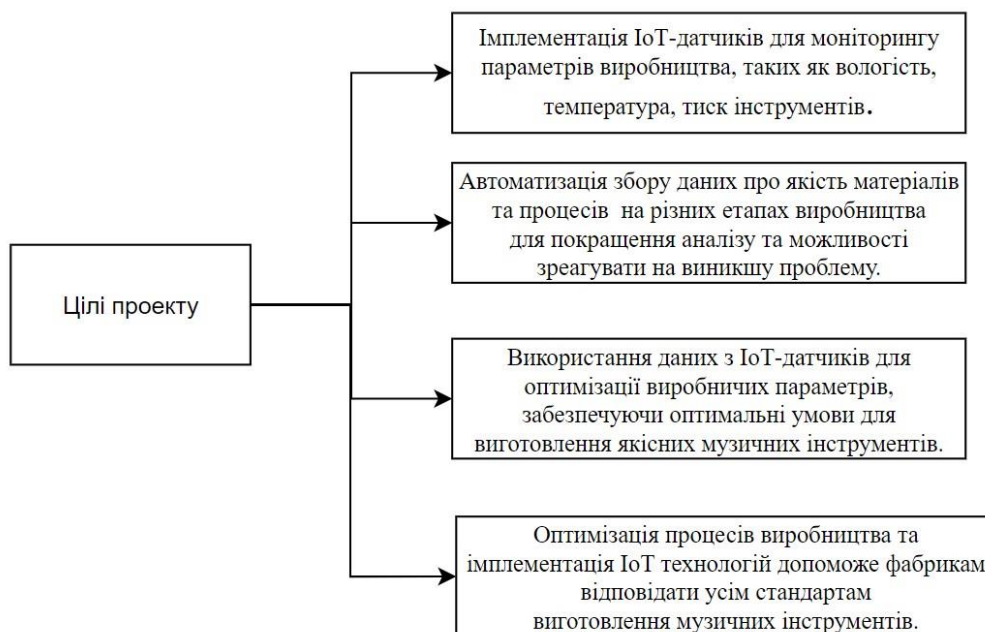


Рис 1.7 Дерево цілей проекту.

Далі ми побудуємо логіко структурну схему нашого проекту це графічне представлення процесів, етапів або компонентів, які взаємодіють у рамках певної системи або проекту. Вона відображає послідовність та зв'язки між різними елементами системи, їх ролі та взаємодії, що дозволяє краще зрозуміти, як вони функціонують разом для досягнення поставлених цілей. А також описує цілі, показники їх досягнення, вимірювачі та можливі ризики [8].

Таблиця 1.7

Логіко-структурна схема

Цілі	Показники досягнення	Вимірювачі	Припущення та ризики
1	2	3	4

1	2	3	4
<p>Загальна ціль Застосування IoT-технологій для моніторингу параметрів виробництва, оптимізація процесу виготовлення музичних інструментів.</p>	<p>Підвищення ефективності виробництва музичних інструментів.</p>	<p>Кількість дефектів, час виробництва, рівень задоволення замовника</p>	<p>Недостатня стабільність IoT технологій, можлива конкуренція з іншими технологіями.</p>
<p>Конкретні цілі 1. Встановлення IoT-датчиків для моніторингу параметрів виробництва. 2. Розробка програмного забезпечення для збору і аналізу даних.</p>	<p>Поліпшення контролю якості, зменшення браку, зниження витрат на виробництво.</p>	<p>Точність вимірювань, швидкість обробки даних, ефективність автоматизації.</p>	<p>Затримки у постачанні компонентів, труднощі з інтеграцією нових технологій.</p>

1	2	3	4
Результати Створення системи моніторингу, автоматичний збір і обробка даних про стан інструментів, покращення контролю якості.	Підвищення якості виготовлення інструментів, оптимізація витрат і часу.	Зниження витрат на матеріали, зниження часу на виробництво інструментів, зниження кількості дефектів.	Ризики затримок у впровадженні технологій, можливі технічні проблеми з налаштуванням датчиків.
Дії Зібрати команду проекту, розробити IoT систему, Відкрити продажі.	Засоби Проектний менеджер, IoT інженери, розробники ПО, аналітики даних.	Витрати Обладнання, заробітна платня, Оренда приміщення.	Передумови IoT система забезпечить кращий контроль якості на виробництві.

1.4 Ідентифікація первинних та вторинних зацікавлених сторін проекту

Звісно основна ціль будь якого проекту є його фінансовий, економічний успіх. За для виділення первинних зацікавлених сторін, ми будемо спиратися саме на цей показник.

Первинні зацікавлені сторони – ті сторони що мають прямий інтерес в успіху проекту.



Рис 1.8 Зображення схеми первинних зацікавлених сторін.

Опис первинних зацікавлених сторін:

Керівництво компаній: Власники та керівники компаній що виробляють муз інструменти, вони є першими інвесторами які зацікавлені у даному проекті щоб покращити виробництво та заробити гроші.

Робітники на виробництві: Робітники, які будуть безпосередньо використовувати систему що полегшить та пришвидшить їх роботу.

Клієнти та споживачі: Кінцеві споживачі музичних інструментів, які будуть використовувати систему для моніторингу якості інструментів, які вони купують. Вони зможуть отримати інформацію про якість продукції, що дозволить їм бути впевненими у придбаному товарі.

Постачальники матеріалів: Постачальники компонентів що будуть використовуватись для виготовлення системи.

Вторинні зацікавлені сторони – це ті сторони які не мають прямої зацікавленості у проекті але цей проект також міг би полегшити їх роботу. Це можуть бути наукові організації, постачальники додаткових технологічних рішень, а також державні органи, які підтримують стандарти

якості в галузі. Вони можуть отримувати корисні дані для покращення своєї роботи або для визначення нових регулювань.



Рис 1.9 Вторинні зацікавлені сторони.

Опис вторинних зацікавлених сторін:

Органи регулювання: Органам що регулюють роботи виробництва буде дешевше відстежувати роботу та дотримання стандартів на виробництві за допомогою полегшеного доступу до статистичних показників.

Екологічні організації: Групи які зацікавлені у успіху проекту для зниження рівню відходів та викидів у атмосферу.

Партнери та інвестори: Бізнес-партнери та фінансові інвестори, які можуть фінансувати проект або мати економічний інтерес у його успіху.

Висновок до розділу: У цьому розділі ми обґрунтували потребу впровадження IoT-системи для контролю якості виробництва музичних інструментів, виділили ключові параметри (геометрія, акустика, вологість), які мають відстежуватися. Ми сформулювали SMART-цілі проекту. Провели оцінку ресурсів: визначили команду (PM, IoT-інженери, розробники ПЗ, QA, ML-інженери, DevOps) та окреслили бюджет проекту. За допомогою Gantt-діаграми спланували часові рамки й розподіл завдань, що дає змогу контролювати витрати та уникати вузьких місць.

РОЗДІЛ 2. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІoT-СИСТЕМИ КОНТРОЛЮ ЯКОСТІ

2.1 Розробка життєвого циклу проекту

Життєвий цикл проекту - це поетапний процес, який супроводжує проект від початку до кінця. Він включає такі етапи, як запуск, планування, виконання роботи, перевірка прогресу та завершення.

Далі було сформовано ЖЦП для нашого проекту “Дослідження процесів управління проектом створення IoT-системи для контролю якості виробництва музичних інструментів” Його представлено на малюнку



Рис 2.1 Зображення схеми життєвого циклу проекту.

Далі ми опишемо усі головні фази нашого життєвого циклу та процеси які будуть виконуватись на цих етапах.

Таблиця 2.1

Опис фаз життєвого циклу проекту

Фаза ЖЦ	Основні задачі
Початок проекту:	Визначення мети проекту Формулювання завдань і цілей Планування ресурсів і часу
Аналіз вимог	Виявлення потреб користувачів Складання технічних вимог Аналіз існуючих технологій і рішень
Проектування системи:	Розробка архітектури IoT-системи Визначення компонентів апаратної частини Описання програмного забезпечення і взаємодії
Розробка і тестування:	Розробка програмного забезпечення Вибір та налаштування датчиків Розробка інтерфейсу користувача Тестування системи та компонентів

Продовження таблиці 2.1

1	2
Впровадження та запуск:	Інтеграція з існуючими системами Встановлення та налаштування системи на виробничих лініях Запуск системи в експлуатацію
Завершення проекту:	Підготовка звітності Оцінка результатів проекту

Далі визначимо основні віхи нашого проекту:

Дослідження вимог та технічного завдання - 01.11.2024.

Це етап, коли ми визначимо усі вимоги до системи, технічне завдання.

Розробка апаратної та програмної частини - 01.04.2025

На цьому етапі вже мають бути готові компоненти апаратної частини та програмне забезпечення.

Тестування системи та інтеграція - 01.07.2025

Після розробки системи треба провести інтеграцію всіх компонентів і протестувати систему на відповідність вимогам.

Запуск та здача проекту - 01.10.2025

Після успішного тестування система передається система повністю запускається та випускається в користування для кінцевих користувачів.

За визначенням віх проекту ми побудуємо ієрархічну структуру робіт де ми детально опишемо результати проекту.

1. Дослідження вимог та технічне завдання

Аналіз ринку та конкурентів На цьому етапі ми оцінюємо існуючі IoT-системи для контролю якості в інших галузях та виявляємо слабкі та сильні сторони.

Визначення вимог до системи. На цьому етапі йде формулювання технічних вимог до апаратної та програмної частини системи які сенсори, ПЗ та інші компоненти будуть використовуватись.

Розробка технічного завдання. Потім ми підготуємо документацію з вимогами до усіх частин нашої системи[11].

2. Розробка архітектури та дизайну системи

Розробка концептуальної моделі. Першим ділом на етапі розробки потрібно створити загальну схему системи. Вона охоплює усі компоненти та зв'язки між ними.

Розробка математичних моделей. Далі потрібно створити мат моделі для збору, обробки та аналізу даних від сенсорів.

Розробка бази даних: Тут ми будемо займатися створенням структури для зберігання даних про виробництво музичних інструментів, параметри якості та контрольні точки.

3. Розробка апаратної частини.

Вибір сенсорів і обладнання. Першим етапом розробки апаратної частини ми визначимо типи датчиків , що будуть використовуватися для моніторингу параметрів якості.

Розробка і налаштування шлюзів та мережі На другому етапі ми маємо обрати та налаштувати шлюзи для збору інформації та їх передачі до централізованої системи.

Інсталяція і тестування обладнання Заключним процесом цього етапу є установка датчиків та інших елементів апаратної частини, перевірка їх роботи в умовах виробництва.

4. Розробка програмного забезпечення

Розробка алгоритмів для збору даних[10]. Тут потрібно зайнятися створенням програмних модулів для обробки даних від датчиків, та інтеграцією їх в єдину систему.

Створення інтерфейсу користувача. Головною задачею тут є розробка графічного інтерфейсу для відображення даних про якість інструментів, моніторинг процесів.

Інтеграція з хмарними сервісами. Заключним завданням на цьому етапі є налаштування хмарних платформ для зберігання та обробки даних, що поступають від датчиків.

5. Тестування та інтеграція

Тестування функціональності системи. Тут треба перевірити коректність збору, обробки та передачі даних, тестування програмного забезпечення на різних етапах виробництва.

Інтеграція всіх компонентів системи. Наступне є перевірка взаємодії апаратної та програмної частини системи, перевірка на сумісність усіх компонентів.

Тестування на відповідність вимогам. І заключним є перевірка роботи системи згідно з технічним завданням, виправлення виявлених помилок.

6. Запуск та здача проекту

Підготовка до запуску. Завершення налаштувань, підготовка документації для експлуатації системи.

Тренінг персоналу. Навчання працівників підприємства щодо використання системи для моніторингу та контролю якості виробництва.

Передача на експлуатацію. Офіційне завершення проекту, передача системи в експлуатацію замовнику, моніторинг її роботи протягом перших кількох місяців.

2.2 Організаційна структура компанії та побудова команди проекту з розподілом відповідальності.

Далі наша задача полягає в побудові організаційної структури компанії.

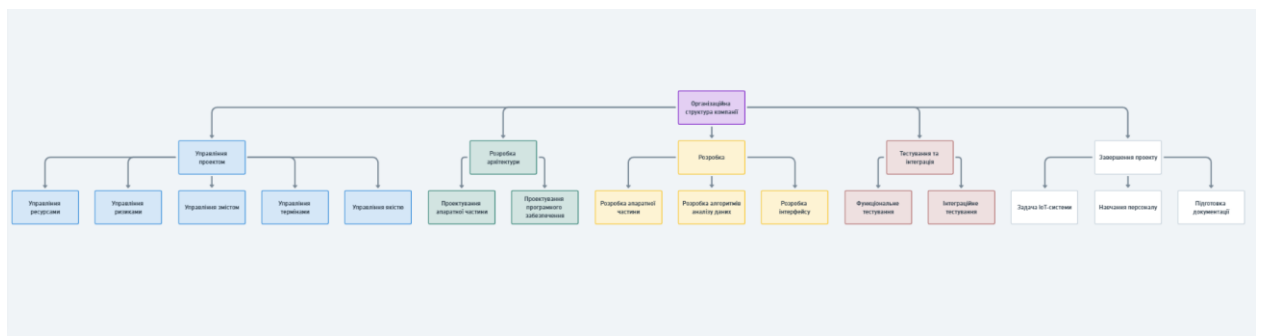


Рис 2.1 Організаційна структура підприємства

Далі ми опишемо усі підрозділи в нашій структурі та їх головні задачі та обов'язки[12].



Рис 2.2 Частина організаційної структури.

Управління ресурсами:

- За управління ресурсами відповідає Project Manager. Він займається плануванням та розподілом ресурсів, а також керує командою, щоб забезпечити ефективне виконання задач проекту.
- За управління ризиками відповідає Risk Manager. Його завдання — оцінка потенційних ризиків та розробка стратегій для їх мінімізації, що дозволить забезпечити стабільність проекту.
- За управління змістом відповідає Business Analyst. Він визначає вимоги до проекту, забезпечує відповідність результатів початковим цілям і контролює виконання функцій, що визначені на початку.
- Project Manager займається управлінням термінами проекту. Він відповідає за планування термінів виконання завдань, моніторинг прогресу та коригування планів у разі необхідності.
- За управління якості відповідає Quality Assurance Manager (QA). Його завдання — забезпечити відповідність продукції та процесів проекту стандартам якості, проводячи перевірки та тестування.

Наступна частина це розробка архітектури в неї входять: розробка апаратної частини проектування програмного забезпечення.



Рис 2.3 Частина організаційної структури.

За проектування апаратної частини відповідає IoT Engineer. Він займається розробкою апаратних компонентів, датчиків, сенсорів та інших елементів, що є частиною IoT-системи.

Розробка програмного забезпечення для збору та обробки даних покладається на Software Engineer. Він відповідає за створення ПЗ, які потрібні для обробки даних з датчиків.



Рис 2.4 Частина організаційної структури

IoT Engineer - займається розробкою апаратної частини, а саме те які датчики будуть використовуватись та яким чином вони будуть працювати та встановлюватись, а також яким чином будуть передаватися дані з цих датчиків.

Data Scientist займається розробкою алгоритмів для обробки даних, отриманих з IoT-датчиків. Його завдання — створити моделі для аналізу даних і прогнозування результатів.

UI/UX Designer відповідає за розробку інтерфейсу користувача. Він займається створенням дизайну системи яку буде використовувати користувач.

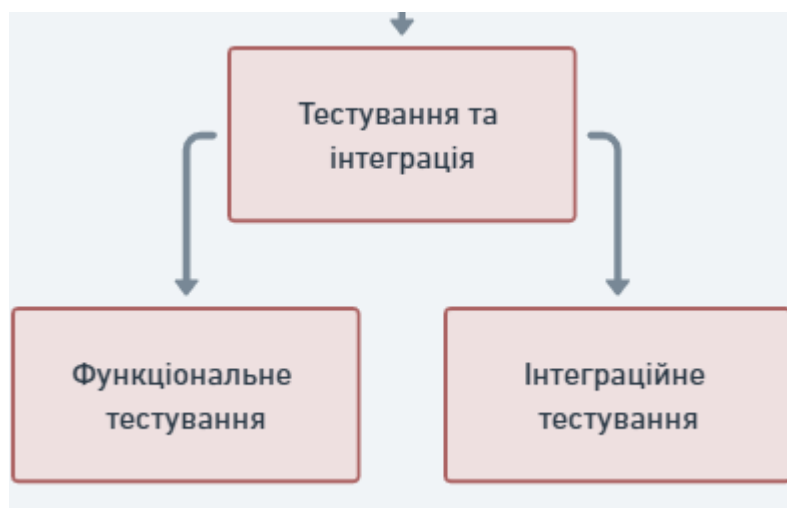


Рис 2.5 Частина організаційної структури

QA Engineer відповідає за тестування функціональності системи. Він перевіряє правильне функціонування системи, на відповідність вимогам. Також він відповідає за перевірку взаємодії між апаратними та програмними частинами системи, щоб впевнитись у їхній роботі.

В цій частині роботи ми побудували та описали структуру нашого проекту. Далі ми побудуємо та опишемо схему роботи з основними видами роботи які мають бути здійснені.

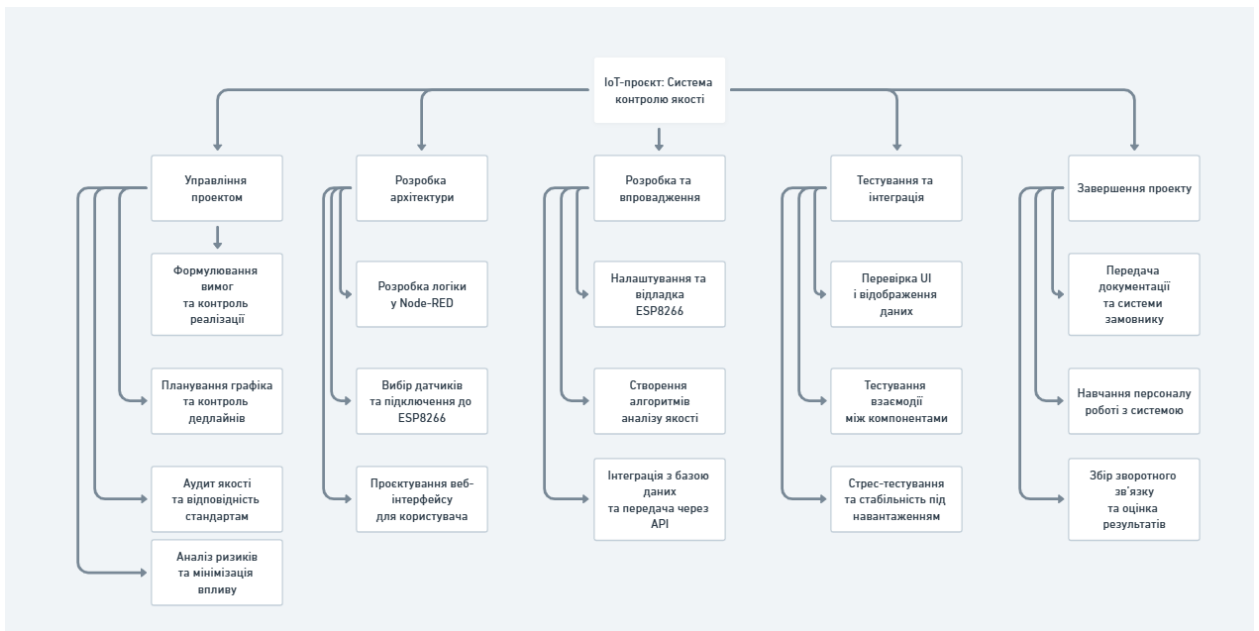


Рис 2.6 ICP схема

Далі ми визначимо відповідальність учасників за методикою RACI побудувавши матрицю. Ця методика є аббревіатурою яка визначає залученість працівників в той чи інший процес.

- Responsible (Виконавець) – має обов’язки які має виконати.
- Accountable (Відповідальний) стежить за тим, щоб усі виконавці виконували свої обов'язки.
- Consulted (Консультант) - це хранитель знань команди. До них звертаються за допомогою та за порадою.
- Informed (Особа, яку інформують) - це зацікавлена сторона, керівництво.

	Project Manager	IoT Engineer	Software Engineer	Data Scientist	UI/UX Designer	Backend Developer	QA Engineer	IT Administrator	Training Manager
Управління ресурсами	A	R	R	R	R	R	R	R	R
Управління ризиками	A	C	C	C	C	C	C	C	C
Управління термінами	A	C	C	C	C	C	C	C	C
Управління змістом	A	C	C	C	C	C	C	C	C
Управління закупівлями	A	C	C	C	C	C	C	C	C
Розробка апаратної частини (ESP8266, датчики)	C	A	C	C	C	C	C	C	C
Розробка програмного забезпечення для обробки даних	C	C	A	C	C	C	C	C	C
Розробка інтерфейсу користувача	C	C	C	C	A	C	C	C	C
Розробка алгоритмів для обробки даних	C	C	C	A	C	C	C	C	C
Інтеграція системи (ESP8266, датчики, база даних)	C	C	C	C	C	A	C	C	C
Функціональне тестування	C	C	C	C	C	C	C	A	C
Інтеграційне тестування	C	C	C	C	C	C	A	A	C
Тестування навантаження та стабільності	C	C	C	C	C	C	C	A	C
Передача системи замовнику	A	C	C	C	C	C	C	C	C
Навчання персоналу	C	C	C	C	C	C	C	C	A
Оцінка результатів проекту	A	C	C	C	C	C	C	C	C

Рис 2.7 Побудована RACI матриця.

2.2.1 Планування ресурсів проекту

Ресурси які ми будемо використовувати для нашого проекту поділяються на матеріальні та трудові.

Виділимо усі необхідні ресурси:

- 1.Проджект менджер
- 2.ІоТ Інженер
- 3.Розробник ПЗ
- 4.Дизайнер
- 5.Дата аналітик
- 6.Бекенд розробник
- 7.Тестувальник Manual
- 8.Тестувальник Automation
- 9.ІТ Адмін

З матеріальних ресурсів нам потрібні будуть:

- 1.Офіс
- 2.РС
- 3.Сервери

Далі ми зобразили лист ресурсів у MS Project.

Resource Name	Type	Material	Initials	Group	Max.	Std. Rate	Ovt. Rate	Cost/Use	Accrue	Base
Проджект менджер	Work		П		100%	360,00 ₴/h	450,00 ₴/h	0,00 ₴	Prorated	Standard
ІоТ Інженер	Work		І		100%	350,00 ₴/h	550,00 ₴/h	0,00 ₴	Prorated	Standard
Розробник ПЗ	Work		Р		100%	320,00 ₴/h	550,00 ₴/h	0,00 ₴	Prorated	Standard
Дизайнер	Work		Д		100%	290,00 ₴/h	480,00 ₴/h	0,00 ₴	Prorated	Standard
Дата аналітик	Work		Д		100%	300,00 ₴/h	480,00 ₴/h	0,00 ₴	Prorated	Standard
Бекенд розробник	Work		Б		100%	350,00 ₴/h	500,00 ₴/h	0,00 ₴	Prorated	Standard
Тестувальник Manual	Work		Т		100%	280,00 ₴/h	450,00 ₴/h	0,00 ₴	Prorated	Standard
Тестувальник Automation	Work		Т		100%	250,00 ₴/h	500,00 ₴/h	0,00 ₴	Prorated	Standard
ІТ Адмін	Work		І		100%	280,00 ₴/h	480,00 ₴/h	0,00 ₴	Prorated	Standard
Офіс	Material	міс	О			0,00 ₴		40 000,00 ₴	Prorated	
РС	Material	Шт	Р			0,00 ₴		33 000,00 ₴	Prorated	
Сервери	Material	міс	С			0,00 ₴		2 000,00 ₴	Prorated	

Рис 2.8 Лист ресурсів проекту

А також назначили кожному процесу ресурси які будуть використані.

2.2.2 Планування вартості проекту

Після планування ресурсів ми маємо розрахувати вартість нашого проекту основними затратами стануть: витрати на трудові ресурси, витрати на матеріальні ресурси, адмін-господарські витрати, витрати на забезпечення команди[14].

Таблиця 2.2

Витрати на трудові ресурси:

Назва	Кіл людей	Вартість год	Години	Заг вартість
Проджект Менеджер	1	360	350	126000
ІоТ інженер	1	350	600	210000
Розробник ПЗ	1	320	723	231360
Дизайнер	1	290	543	157540
Дата аналітик	1	300	403	120900
Бекенд розробник	1	350	522	182700
Тестувальник	1	250	322	80500
Тестуваник Auto	1	280	290	81200
ІТ адмін	1	250	456	114000
Загальна Вартість				1.094.200



Рис 2.9 Зображення відсоткового розподілення ресурсів.

Спочатку ми розрахували загальну вартість роботи кожного робітника помноживши кількість годин, на вартість за годину, і щоб вирахувати зальну вартість ЗП усіх працівників склали загальні вартості кожного працівника. Таким чином загальна вартість заробітних плат склала 1.094.200 грн.

Далі ми розрахуємо витрати на матеріальні ресурси.

Таблиця 2.3

Витрати на матеріальні ресурси:

Назва	Кількість/шт., грн	Вартість/шт.,грн	Вартість,грн
Ноутбуки	9	33000	297000 грн
Сервери	2	2000 на місяць	24000 грн
Загальна вартість			321000 грн

В результаті на матеріальні ресурси буде витрачено 321 000 грн. Наступною частиною витрат стануть адміністративно господарські витрати в них входять такі витрати як: оренда офісу, водопостачання, газ/тепло, електрика.

Таблиця 2.4

Адміністративно-господарські витрати

Назва	Кількість/міс	Грн/міс	Вартість
Оренда офісу	7	40000	280000
Водопостачання	7	5000	35000
Газ/Тепло	7	3800	26600
Електрика	7	4000	28000
Загальна вартість			369600



Рис 2.10 Зображення відсоткового розподілення ресурсів.

За підрахунками на амін господарські витрати за 7 місяців роботи проекту буде витрачено 369 600 грн. Також нам треба розрахувати витрати на послуги інтернету та прибирання офісу.

Таблиця 2.5

Витрати на послуги

Назва	Кількість міс	Грн міс	Вартість грн
Інтернет	7 міс	1500	10500
Прибирання	7 міс	8000	80000
Загальна вартість			90 500

ВАРТІСТЬ ГРН

■ Інтернет ■ Прибирання

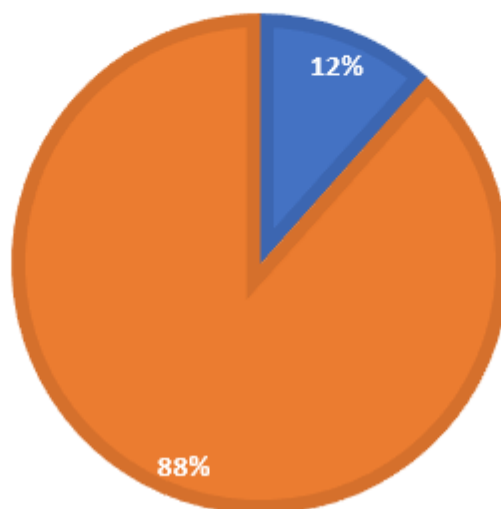


Рис 2.11 Зображення відсоткового розподілення ресурсів.

За результатами підрахунків витрати склали 90 500 грн.

Також дуже важливо забезпечити нашу команду ресурсами для роботи в офісі.

До них можна включити:

Поставка води

Канцелярія

Їжа

Напої кава чай

Для підрахунку ми складемо таблицю затрат.

Таблиця затрат на утримання офісу

Назва	Од Вимірювання	Кількість од/міс	Вартість од/грн	Кількість міс	Вартість грн
Ручки	Шт	20	15	7	2100
Олівці	Шт	20	15	7	2100
Блокноти	Шт	20	50	7	7000
Фрукти	Ящ	10	600	7	42000
Вода	Л	550	16.30	7	62755
Молоко	Л	250	30	7	52500
Кава	Упак	10	460	7	32000
Чай	Упак	20	300	7	42000
Снеки	Ящ	10	200	7	14000
Загальна вартість					256.455

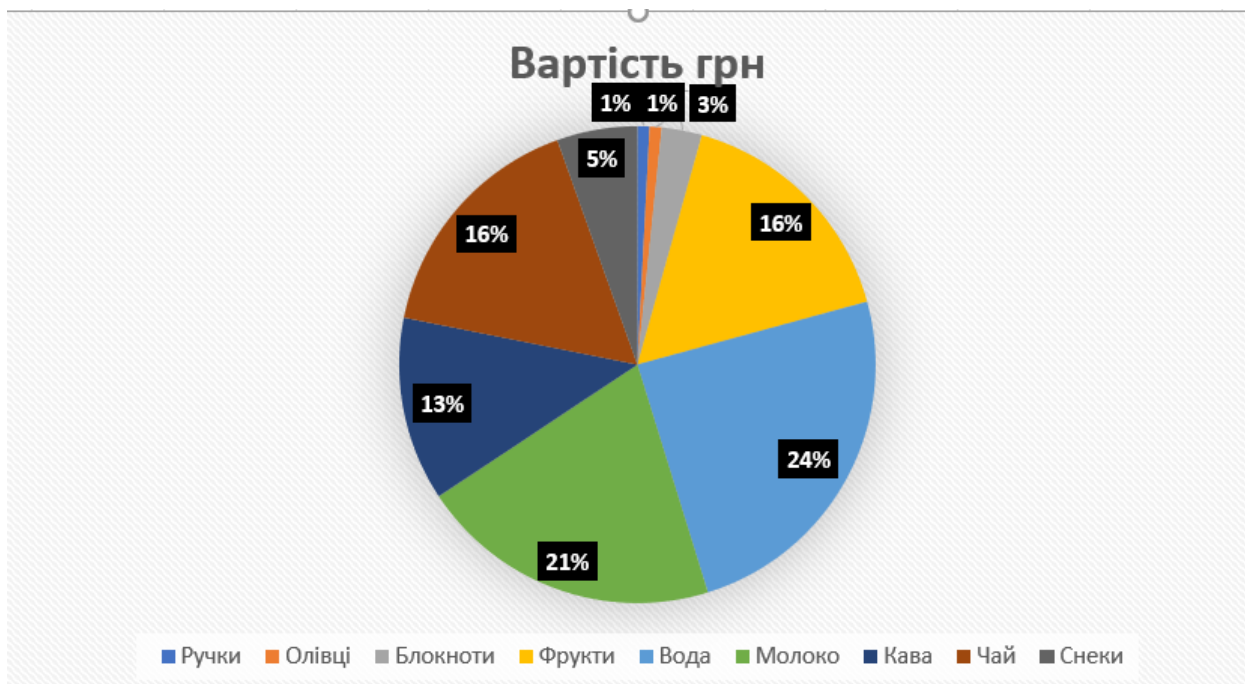


Рис 2.12 Зображення відсоткового розподілення ресурсів.

Далі розрахуємо загальні витрати за всіма категоріям.

Таблиця 2.7

Зведені витрати проекту.

Тип затрат	Вартість грн
Витрати на трудові ресурси	1,094,200
Витрати на матеріальні ресурси	321,000
Адміністративно господарські витрати	369,600
Витрати на послуги	90,500
Затрати на утримання офісу	256,455
Загальна сума	2.231.755

Таким чином ми можемо підрахувати що загальна вартість нашого проекту буде близько 2.3 млн грн.

2.2.3 Розробка концептуальних моделей інформаційної системи

В цій частині роботи ми розкриємо компоненти системи з яких вона побудована:

Сенсори та Датчики : Відповідають за збір даних про параметри виробництва, такі як температура, вологість, тиск, вібрація тощо Кожен датчик буде відповідати за певний показник та передавати данні в систему для подальшою обробки.

Збір і передача даних: Інформація яку зібрали датчики, буде надсилатися в систему для аналізу. Процес здійснюватиметься через мережеві протоколи. Інформація з сенсорів буде передаватись через бездротові або проводові засоби зв'язку до системи контролю. Вони можуть використовувати протоколи передачі даних, такі як Wi-Fi, Bluetooth або Zigbee та інші.

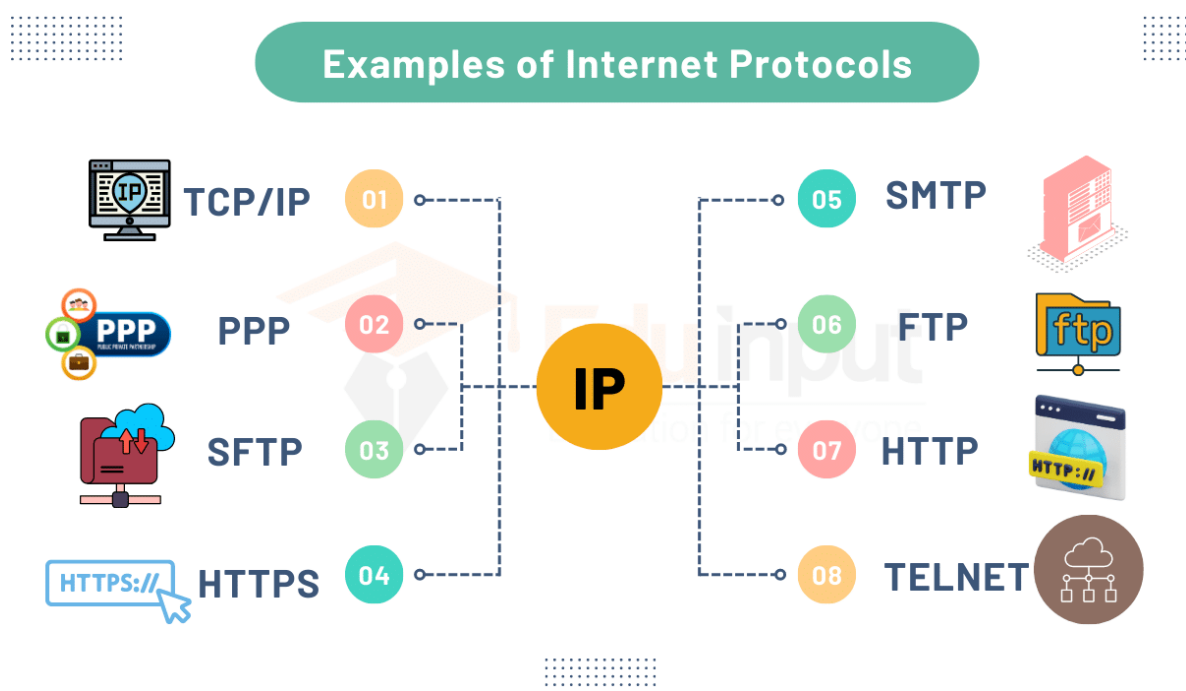


Рис 2.13 Мережеві протоколи

Центральна система управління Це те місце куди потрапляють дані та аналізуються. Система використовує алгоритми для обробки даних в реальному часі та аналізу якості виробництва.

Система контролю та реагування: На основі проаналізованих даних система може займатися регулювання умов виробництва або надсилати повідомлення про відхилення які були виявлені. Також основі аналізу даних система може приймати рішення щодо регулювання температури відключення певних системи за потреби.

Інтерфейс користувача: Оператор взаємодіє з інтерфейсом користувача, щоб контролювати показники з датчиків та приймати рішення. Окрім показників датчиків на інтерфейсі можуть бути зображені статуси готовності обладнання[15].

09.11.2021	Система контролю параметрів повітря			09:17:14
	TREQ3456			
Попередження та тривоги				
<input style="border: none; background: none;" type="button" value=" < "/> <input style="border: none; background: none;" type="text" value=" T11/H11 "/> <input style="border: none; background: none;" type="button" value=" > "/>				
Канал	Нижня границя		Верхня границя	
	Попередження	Тривоги	Попередження	Тривоги
T1 (Темпер-ра, °C)	0.0 <input style="border: none; background: none;" type="button" value=" Не задіяна "/>	0.0 <input style="border: none; background: none;" type="button" value=" Не задіяна "/>	26.0 <input style="border: none; background: none;" type="button" value=" Не задіяна "/>	30.0 <input style="border: none; background: none;" type="button" value=" Не задіяна "/>
H1 (Вологість, %)	0.0 <input style="border: none; background: none;" type="button" value=" Не задіяна "/>	0.0 <input style="border: none; background: none;" type="button" value=" Не задіяна "/>	50.0 <input style="border: none; background: none;" type="button" value=" Задіяна "/>	60.0 <input style="border: none; background: none;" type="button" value=" Задіяна "/>
<input style="border: none; background: none;" type="button" value=" Моніторинг "/>		<input style="border: none; background: none;" type="button" value=" Змінити "/>		

Рис 2.14 Приклад інтерфейсу системи користувача

Після того як ми описали та представили компоненти нашої системи ми побудуємо концептуальну модель в якій зобразимо усі процеси що відбуваються у нашій системі [16].

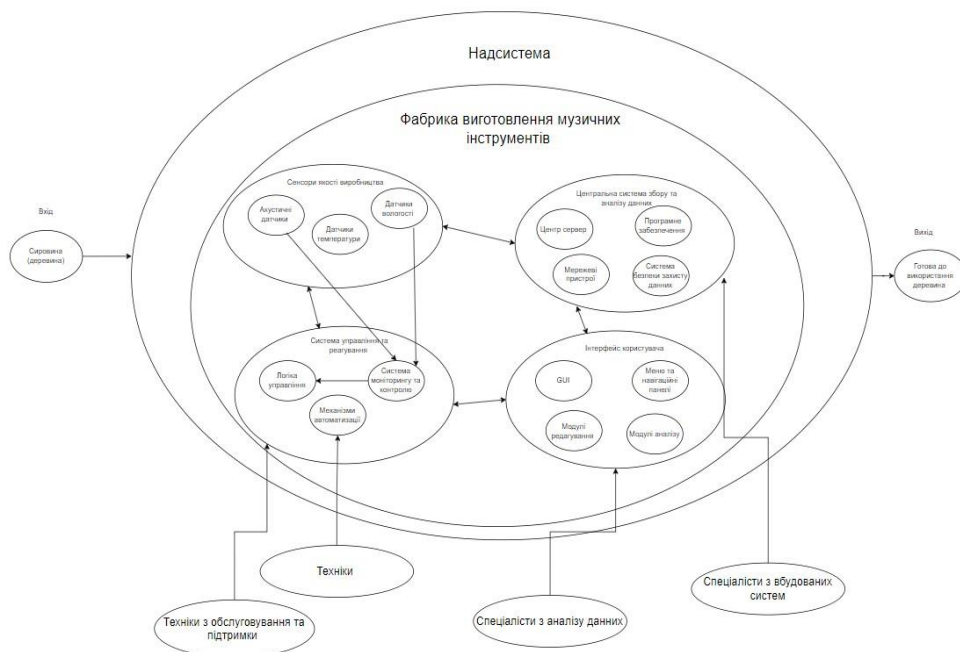


Рис 2.15 Концептуальна модель

Далі ми пояснимо зв'язки нашої моделі для кращого розуміння процесів, які відбуваються[17]:

Таблиця 2.8

Опис основних компонентів та взаємодій в системі контролю якості виробництва музичних інструментів

Компонент	Функція	Опис
1	2	3
Фабрика виготовлення музичних інструментів	-	Головний об'єкт системи, який охоплює усі процеси та взаємодії виробництва музичних інструментів.

1	2	3
Система управління реалізацією	Поля управління, механізм автоматизації	Система, яка забезпечує управління та моніторинг виробничих процесів через інтерфейси управління.
Система контролю якості	Центри сервер, мережеві пристрої	Техніка та програмне забезпечення для перевірки якості продукції та процесів на кожному етапі виробництва.
Спеціалісти з обладнаних систем	-	Фахівці, які займаються налаштуванням, обслуговуванням і ремонтом виробничих систем.
GUI (Графічний інтерфейс користувача)	Модулі реалізації, модулі аналізу	Інтерфейси, через які користувачі взаємодіють з системами контролю та управління.
Додаткові аспекти	Датчики якості продукції, датчики температури	Інструменти моніторингу, які допомагають збирати

		дані про стан виробництва та умови праці.
--	--	---

продовження 2.8

1	2	3
Вихід	-	Результати роботи системи, які можуть включати вироблену продукцію, статистику виробництва та звіти про якість.

2.3 Формалізація математичних моделей та постановка задачі у математичному вигляді.

Для початку виділимо мету нашої мат моделі - мінімізувати загальні витрати на контроль якості, одночасно підтримуючи високий рівень точності вимірювань і знижуючи кількість браку.

Для вираження математичної моделі приймемо такі параметри:

Параметри:

- n : Кількість контрольних точок на виробничій лінії.
- c_i : Вартість встановлення датчика на i -тій контрольній точці.
- a_i : Точність датчика на i -тій контрольній точці.
- p_i : Ймовірність виявлення браку датчиком на i -тій контрольній точці.

- v : Середні витрати на бракований інструмент, який не був виявлений.
- b : Базовий рівень браку без системи контролю.

2.3.1 Цільова функція математичної моделі

Цільова функція полягає у мінімізації загальних витрати на контрольні датчики та витрати через брак, що не виявлений:

Мінімізувати $Z = \sum_{i=1}^n c_i x_i + v(b - \sum_{i=1}^n p_i x_i)$. Цільова функція мінімізувати загальні витрати які складаються з двох основних компонентів:

Перша компонента:

$$\text{Витрати на встановлення датчиків: } \sum_{i=1}^n c_i x_i \quad (2.1)$$

Показує загальну вартість встановлення датчиків на всіх контрольних точках. Друга компонента це:

$$\text{Витрати через невиявлений брак: } v(b - \sum_{i=1}^n p_i x_i) \quad (2.2)$$

В цьому випадку v — вартість браку на одиницю продукції, b — базовий рівень браку, а p_i — ймовірність виявлення браку датчиком на i -й контрольній точці. Цей компонент виражатиме зниження витрат на брак, який ми зможемо виявити за допомогою встановлених датчиків.

Обмеження: Обмеження цільової функції приймемо два основних обмеження такі як обмеження на мінімальну допустиму точність системи та бюджетне обмеження.

$$\text{Обмеження на мінімально допустиму точність системи: } \sum_{i=1}^n a_i x_i \geq A \quad (2.3)$$

де A — вимога до мінімальної точності системи.

Це обмеження свідчить про те, що сумарна точність всіх активних датчиків має бути не меншою за A , де a_i — точність датчика на i -й точці.

$$\text{Бюджетні обмеження: } \sum_{i=1}^n c_i x_i \leq B \quad (2.4)$$

де B — максимальний бюджет, що може бути витрачений на систему.

Це обмеження гарантує, що загальні витрати на встановлення датчиків не перевищуватимуть бюджет B .

2.3.2 Математична постановка задачі

Наша задача сформульована як задача бінарної оптимізації, де ми хочемо знайти x_i для кожної контрольної точки так щоб мінімізувати наші загальні витрати (включаючи витрати на встановлення датчиків і потенційні витрати на невиявлений брак), враховуючи обмеження на точність і бюджет.

$$\text{Треба мінімізувати } Z = \sum_{i=1}^n c_i x_i + v(b - \sum_{i=1}^n p_i x_i) \quad (2.5)$$

за умови:

$$\sum_{i=1}^n a_i x_i \geq A \quad (2.2.6) \quad \sum_{i=1}^n c_i x_i \leq B \quad (2.6)$$

$$x_i \in \{0, 1\} \text{ для всіх } i=1 \dots n, x_i \in \{0, 1\} \text{ для всіх } i=1 \dots n$$

Це допоможе нам забезпечити оптимальний розподіл та використання датчиків на виробництві оптимізувавши витрати та якість кінцевого продукту.

2.3.3 Використання методів моделювання розроблених моделей. Ознайомлення із задачами контролю та управління інформаційними технологічними процесами

Моделювання використовується для прогнозування та планування, оптимізації процесів. Для наших математичних моделей IoT системи можна використати такі методи моделювання: Модель Лінійного програмування: використовується для розв'язання задач оптимізації, коли цільова функція та всі обмеження є лінійними. У нашому випадку щоб використовувати цю систему ми мали би відмовитись від змінних x_i у бінарному вигляді і допустити їх значення в проміжку $[0, 1]$, тоді задачу можна було б вирішити методами лінійного програмування[18].

Модель Цілочисельного лінійного програмування (ILP): Це частина лінійного програмування, де змінні обмежені цілими значеннями. Це чудово підходить для нашої задачі оскільки x_i є бінарними. А також ILP дозволяє точно моделювати вибір датчиків[19].

Модель Симуляційного моделювання: Використовується для імітації реальних процесів з метою виявлення потенційних проблем і випробувань різних підходів до стратегій управління[14]. Симуляції можуть допомогти зрозуміти, як зміни у встановленні датчиків впливають на загальну ефективність системи та виявлення браку.

2.4 Розробка програмного забезпечення IoT-системи

2.4.1 Архітектура та функціональні вимоги

Архітектура нашої IoT-системи буде спиратися на архітектуру edge + cloud, що зможе забезпечити, невелику затримку на події та водночас дасть можливість використовувати хмарні сервіси для зберігання, аналітики й моніторингу даних. Основні компоненти архітектури та їхні функціональні

ВИМОГИ:

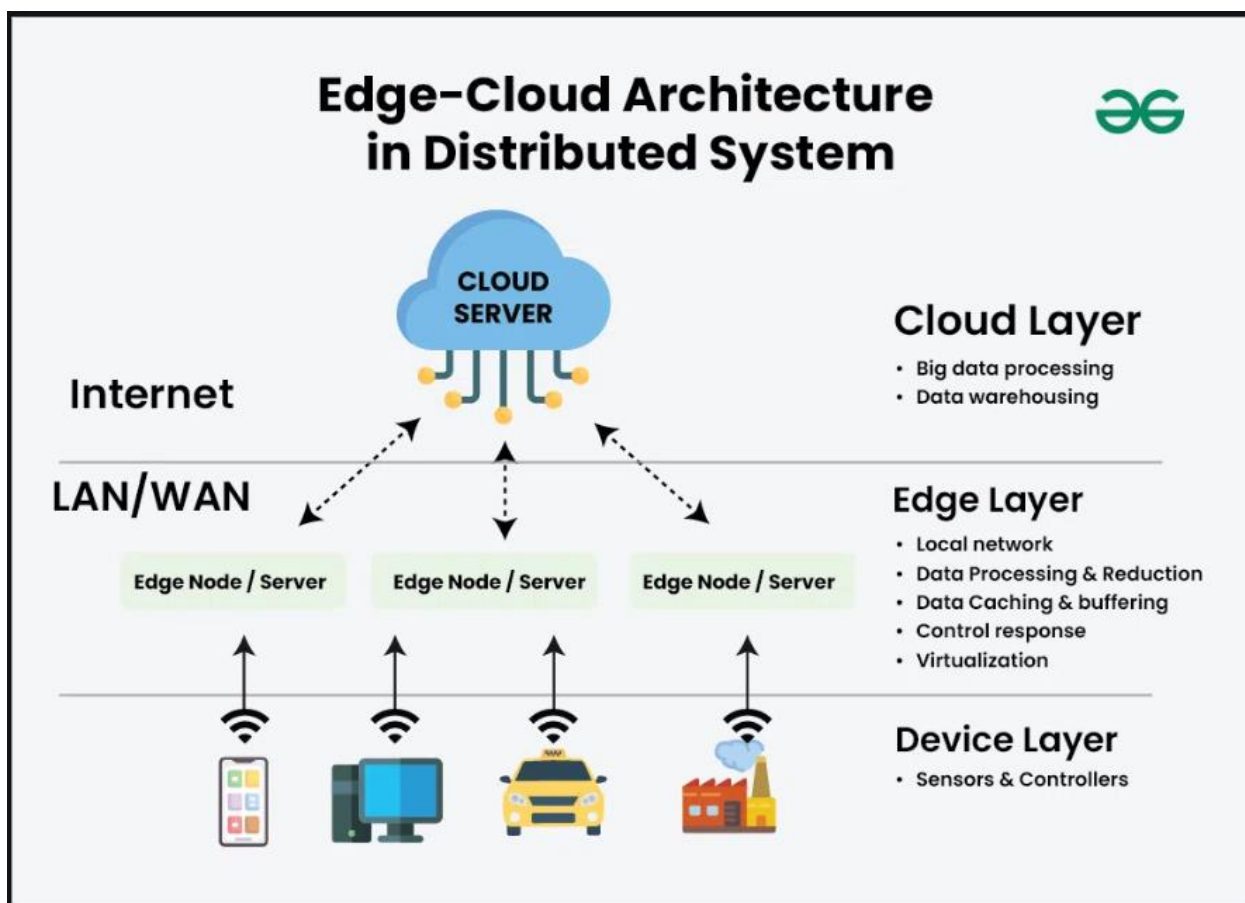


Рис 2.16 Edge-Cloud архітектура

Архітектурні шари: Шар датчиків в нього входять сенсори якості: температури, вологості, вібраційні сенсори, лазерні 3D-сканери та мікрофони

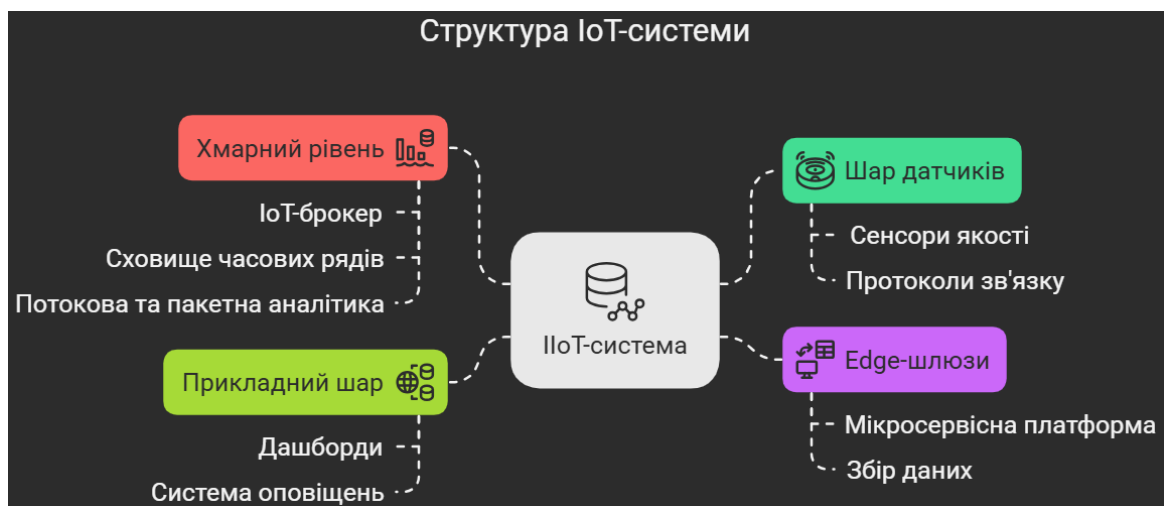
для акустичного моніторингу. А також в нього входять протоколи зв'язку: OPC UA, Modbus, MQTT, Bluetooth LE, Ethernet.

Edge-шлюзи: Мікро сервісна платформа наприклад, EdgeX Foundry: ця система відповідає за збір даних з різних сенсорів через, окрім збору також платформа відповідає за нормалізацію й буферизацію даних, первинна аналітика/фільтрація та агрегація перед відправкою.

Хмарний рівень: AWS IoT Core, Azure IoT Hub цей рівень відповідає за прийом даних, керування пристроями, автентифікація та шифрування трафіку, також цей рівень виступає за зберігання даних.

Прикладний шар (UI/UX): Цей рівень відповідає за відображення даних наприклад за допомогою дашбордів таких як Grafana, Power BI: візуалізація поточних показників якості, історії дефектів, сповіщення.

Система оповіщень: Також на цьому рівні реалізована логіка відправки повідомлень (SMS, email, push-сповіщення) якщо виявлені критичних відхилень.



2.17 Алгоритми обробки даних та інтерфейси користувача

IoT-технології все частіше використовуються за для контролю якості на різних виробництвах. В ході використання IoT технологій генерується дуже великий обсяг даних, і якщо передавати всі ці дані до хмари у необробленому вигляді у різних мережа може бути перевантажена та може статися затримка

що є критичним для задач що мають виконуватись у реальному часі. Тому сучасні рішення використовують гібридну архітектуру edge + cloud, у якій частина обробки виконується локально на периферії (edge), а інша – у хмарі. Це дає можливість отримувати швидкий зворотний зв'язок від датчиків без затримок і користуватись хмарними ресурсами для обчислень.

Тож розглянемо ключові функціональні модулі такої архітектури:

1. Збір даних з датчику:

Функція: Головна функція цього модулю це збір даних з датчиків.

На цьому етапі відбувається зчитування даних з різноманітних сенсорів та обладнання. IoT-системи контролю якості часто інтегруються з промисловими датчиками та контролерами, що використовують такі протоколи зв'язку як: Modbus і Profibus - це дротові інтерфейси для промислової автоматики, OPC UA – це уніфікований протокол обміну даними в промислових системах, MQTT - це легкий протокол публікації/підписки для IoT, а також бездротові протоколи на зразок Bluetooth Low Energy, ZigBee, Wi-Fi та інші.

Також як правило, в такій архітектурі є edge-шлюз (або контролер) –це пристрій що підключений до локальних датчиків. Він виконує роль концентратора: збирає дані від декількох сенсорів через їх рідні протоколи і переводить у стандартизований формат для того щоб їх обробити. Наприклад, є платформа EdgeX Foundry вона пропонує мікро сервісну архітектуру для edge-шлюза: модулі Device Services підключаються до датчиків, станків, камер тощо через інтерфейси Modbus, BACnet, MQTT, REST і отримують від них дані. А вже отримані сирі дані передаються шлюзу і далі готові до обробки.

2. Попередня обробка на рівні edge

На цьому рівні edge відбувається первинна обробка зібраних даних. Основні задачі цього модуля: відфільтрувати шум чи непотрібні показники, обробити дані, та виконати початковий аналіз. Така локальна обробка

допомагає розвантажити канал зв'язку. Зокрема, шлюзи та контролери можуть виконувати такі дії.

Фільтрація даних: відсів небезпечних чи неінформативних значень сенсорів. Наприклад, датчик температури може зчитувати з високою частотою, але для контролю якості достатньо передавати лише значні відхилення або усереднені значення. Локальна фільтрація зайвих точок, зменшує обсяг даних які надсилаються, та зменшує затрати та ресурси енергії на пересилання даних.

Об'єднання показників: об'єднання показників за часом або від декількох датчиків. Edge-пристрої можуть накопичувати дані за певний проміжок часу або обчислювати середні значення за хвилину чи зводити показники з декількох сенсорів до єдиного повідомлення. Це в свою чергу знижує частоту передач і навантаження на хмару.

Збагачення контекстом: Якщо додавати до сенсорів контекстну інформацію такі як мітки часу, ID партії продукції, геолокації тощо, перед відправкою аналіз у хмарі матиме всю необхідну інформацію, і немає потреби робити це пізніше на сервері.

Локальна аналітика та виявлення аномалій: Виконання нескладних обчислень прямо на периферійному вузлі. Наприклад, алгоритм порогового контролю або легковагові ML-моделі (дерева рішень, регресії) можуть працювати на шлюзі для визначення, чи відповідає параметр допустимим значенням. Якщо значення передане на вузол не відповідає допустимим значенням пристрій може сигналізувати про це оператору або навіть, зупинити конвеєр при виявленні браку[23].

3. Передача даних на хмару: Після обробки даних на шлюзах, вони передаються у базу даних та на дашборд де вони відображаються.

Здійснення цих операцій є дуже важливим для системи, що напряду вплине на результат. Оскільки критично важливі рішення приймаються в реальному часі без затримки на мережу – виробнича лінія не мусить чекати секунди на відповідь від хмари, адже edge-аналіз дає миттєвий результат що є дуже важливим при великому виробництві. По-друге, якщо зменшити обсяг

даних, що надходять у хмару, це істотно знижує вимоги до пропускної здатності і пришвидшує передачу даних. Окрім цього, edge-рівень є буфером на випадок перебоїв зв'язку на ньому можна тимчасово зберігати дані або продовжувати автономний контроль якості, поки відновиться з'єднання з хмарою[24].

РОЗДІЛ 3. УПРАВЛІННЯ ТА КООРДИНАЦІЯ РОЗРОБКИ ПРОЄКТУ

3.1. Планування етапів реалізації IoT-системи

На початковому етапі були визначені основні вимоги до функціонування системи, а саме — забезпечення постійного моніторингу температури та вологості, а також передача зібраних даних на візуальний інтерфейс. Для впровадження було обрано такі компоненти:

Сенсор DHT11 — цифровий датчик температури та вологості, обраний простоту у використанні та надійність.

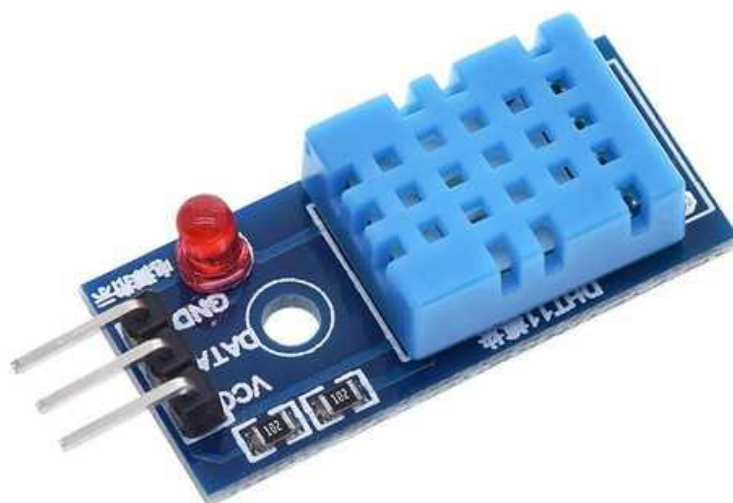


Рис 3.1 Сенсор DHT-11

Датчик DHT11 є цифровим датчиком температури, який може вимірювати температуру в діапазоні від -20 до 60 градусів Цельсія з похибкою ± 2 градуси, а також вимірювати вологість від 5% до 90% з похибкою 5%. Принцип його роботи базується на двох чутливих елементах: термісторі (для вимірювання температури) та елементі вологості. Зміна вологості призводить до зміни ємності елемента, тоді як термістор змінює свій опір залежно від температури[23] [29].

У датчику є аналогово-цифровий перетворювач, що перетворює показання чутливих елементів на цифровий сигнал, який потім надходить в блок обробки сигналів. Для передачі даних датчик використовує однопрохідний цифровий протокол зв'язку, при якому дані передаються біт за бітом, де кожен біт кодується тривалістю певного рівня напруги. Завдяки використанню цифрового методу вимірювання, датчик виводить цифрові сигнали, які легко зчитуються мікроконтролерами або іншими пристроями [24].

Для зчитування даних із датчиків та їх подальшої передачі в систему використовується мікроконтролер Arduino Uno як edge-пристрій. Для організації бездротового зв'язку між компонентами системи було обрано Wi-Fi модуль ESP8266, зокрема плату NodeMCU V3 ESP8266.

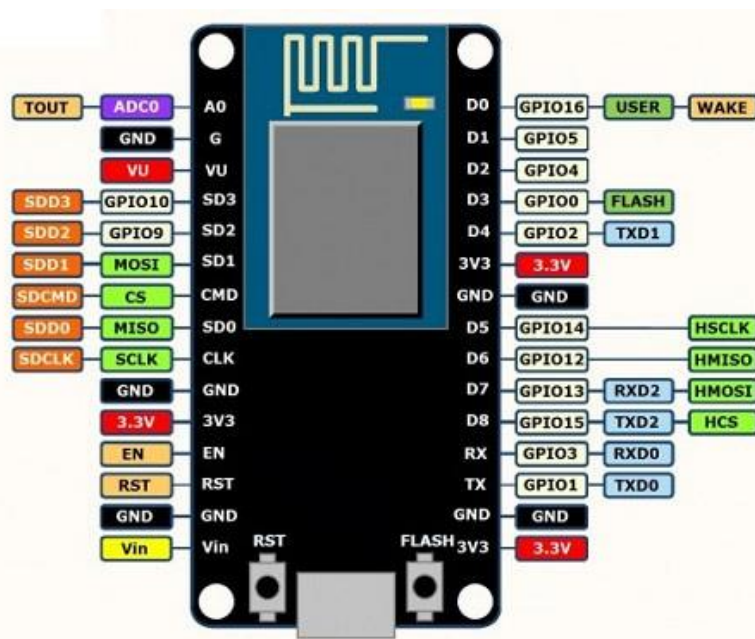


Рисунок 3.2 - Схема зображення Плати Node MCU.

Модуль ESP8266 V3 заснований на мікросхемі ESP8266EX, яка є недорогим та високопродуктивним чіпом для використання в IoT-проектах. Він підтримує стандарт Wi-Fi 802.11 та має вбудований стек протоколів TCP/IP, що значно полегшує підключення до інтернету та взаємодію з іншими пристроями. Модуль також оснащений 32-бітним RISC-процесором, здатним

працювати на частоті до 160 МГц, що дозволяє йому запускати складніші програми без необхідності використання зовнішнього мікроконтролера[25].

ESP8266 V3[28] має 16 виводів GPIO, які можна використовувати для різних задач, зокрема для керування датчиками, пристроями та іншими компонентами системи. Також він оснащений вбудованим **аналогово-цифровим перетворювачем (АЦП)****, що дає змогу вимірювати аналогові сигнали та перетворювати їх у цифрові значення для обробки.

Окрім цього, модуль має 4 МБ флеш-пам'яті, що можна використовувати для зберігання прошивки, веб-сторінок або інших даних, що є критичним для автономних IoT-рішень, де потрібно зберігати дані або інтерфейси без необхідності підключення до зовнішніх серверів.

Ці характеристики роблять ESP8266 V3 ідеальним рішенням для невеликих IoT-проектів, таких як системи для контролю якості, де важлива економія простору та енергії, а також забезпечення безперебійної роботи без додаткових зовнішніх компонентів[26];[27].

Node-RED — це візуальний інструмент для програмування, який дозволяє легко об'єднувати апаратні пристрої, API та онлайн-сервіси, використовуючи потоки даних. Завдяки інтуїтивно зрозумілому браузерному інтерфейсу, Node-RED дозволяє створювати, розгортати та керувати потоками даних без необхідності написання великих обсягів коду.

Node-RED побудований на основі принципу потокового програмування. Замість того, щоб писати код, користувачі створюють потоки, з'єднуючи вузли, які є певними діями або функціями. Кожен вузол виконує певну операцію, таку як зчитування даних з датчиків, обробка інформації, взаємодія з іншими системами чи сервісами. Це дозволяє легко візуалізувати логіку роботи програми та покращує розуміння того, як дані рухаються через систему.

У контексті вашого проекту, Node-RED може бути використаний для візуального проектування логіки обробки даних з IoT датчиків, таких як температурні та сенсори вологості, а також для виведення даних на інтерфейс

користувача. Це дозволяє легко налаштувати передачу даних між пристроями, базою даних та інтерфейсом, забезпечуючи високу гнучкість і швидкість розробки.

Є три типи вузлів вхідні та вихідні та обробні.

- Вхідні приймають інформацію.
- Вихідні її надсилають інформують чи запускають події.
- Обробні ж в свою чергу виконують дії над даними.

До прикладу є вузди HTTP запитів, вхідні та вихідні вузли MQTT, вузли баз даних. Редактор Node Red: Редактор Node Red представляє собою веб інтерфейс в якому редагують та керують потоками:[21] [37].

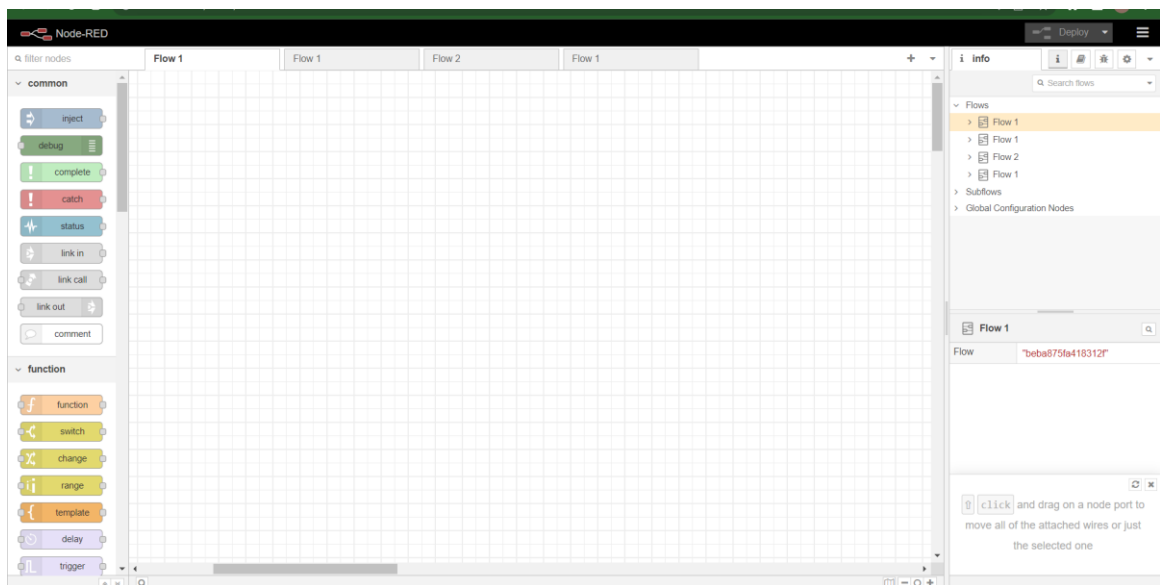


Рис 3.3 Веб інтерфейс Node Red

Зліва на сторінці Node Red представлені типи вузлів. Які можна використовувати у програмі. Також окрім дефолтних вузлів в залежності від задач які поставлені перед програмним застосунком [22] [39].



Рисунок 3.4 - Блоки представляють собою функції

Вузли мають вигляд блоків. Їх можна легко переміщати. Щоб зробити вузол ми використовуємо систему drag n drop. просто обираючи потрібний вузол та перетягуючи його на робочу поверхню[27].

Основну частину нашого інтерфейсу займає робоча поверхня.

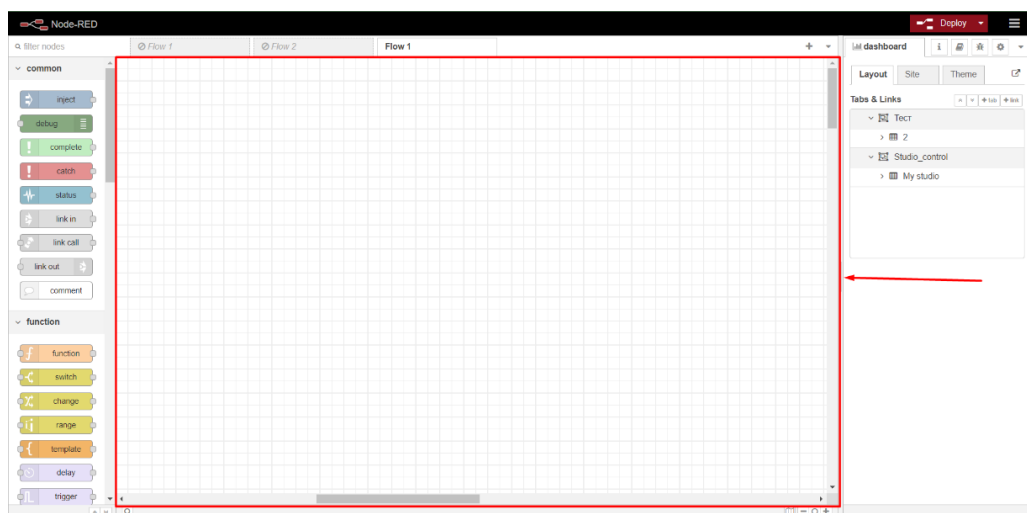


Рисунок 3.5 - Workspace Робоча частина

У правій частині веб інтерфейсу знаходиться секція налаштувань в ній наявні декілька вкладок.

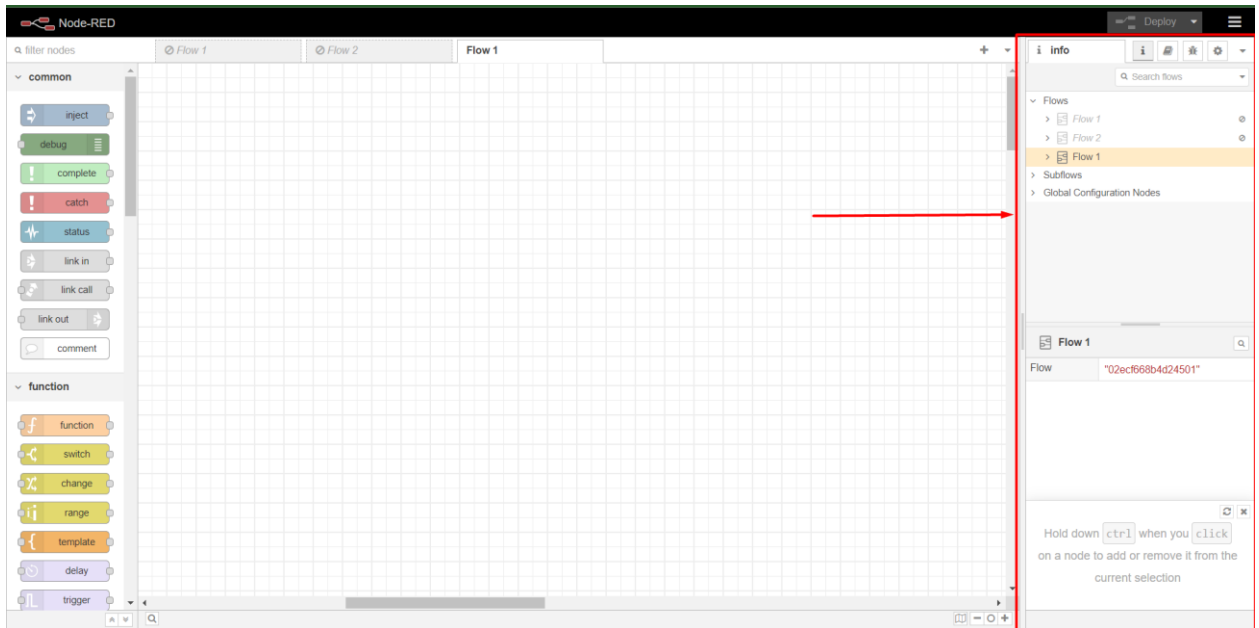


Рисунок 3.6 - Частина налаштувань Node RED

Щоб побудувати нашу систему, ми використали такі компоненти.

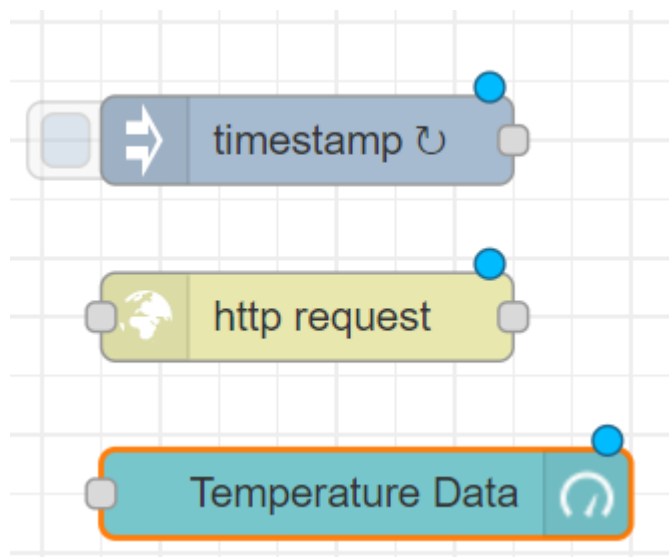


Рис 3.7 Використані Блоки

Timestamp — це мітка часу, яку додають до системи для відстеження часу, коли повідомлення або дані надійшли. У системі Node-RED мітка часу додається до повідомлення через властивість `msg.timestamp`. Ця мітка містить точну дату і час отримання повідомлення, що дає змогу здійснювати точний моніторинг процесу обробки даних у реальному часі[37].

HTTP Request Node використовується для відправки HTTP запитів на сервер і отримання відповідей. Цей вузол дозволяє налаштувати різні

параметри запиту. Першим параметром є метод HTTP, що визначає тип запиту. Метод може бути GET, що використовується для отримання даних, POST для створення або оновлення даних, PUT для оновлення вже існуючих даних, та DELETE для видалення даних на сервері.

Другим важливим параметром є URL адреса, де вказується точне місце призначення для запиту, тобто адреса сервера, на який буде направлений запит. Це дозволяє визначити точку взаємодії системи з іншими сервісами або базами даних. Також можна додавати заголовки до запиту, такі як Content-Type, який визначає тип переданих даних (наприклад, application/json), та Authorization, що забезпечує автентифікацію, або інші кастомні заголовки, необхідні для взаємодії з сервером.

Для методів POST та PUT є можливість додавання корисного навантаження, яке включає дані, що передаються в тілі запиту. Це можуть бути параметри або дані у форматі JSON, які потрібно передати серверу для обробки або зберігання[40].

Наступним параметром є максимум та мінімум значень, які вказують діапазон значень, з якими буде працювати система. Ці значення можуть коливатися в межах певного діапазону, що дає змогу зберігати контроль над параметрами, що передаються.

Для зручності відображення та роботи з даними, в системі є можливість вказати одиниці вимірювання, в яких будуть показуватися значення (наприклад, температура в градусах Цельсія або вологість у відсотках). Це дозволяє забезпечити узгодженість даних з іншими системами, де використовуються специфічні одиниці вимірювання. Нарешті, є можливість налаштування кольорів для відображення значень. Це дає змогу візуально виділяти важливі параметри або показники, наприклад, температури, що знаходяться в критичних межах, та забезпечує зручніше сприйняття даних користувачами системи.

Після додавання вузлів ми з'єднуємо їх між собою.

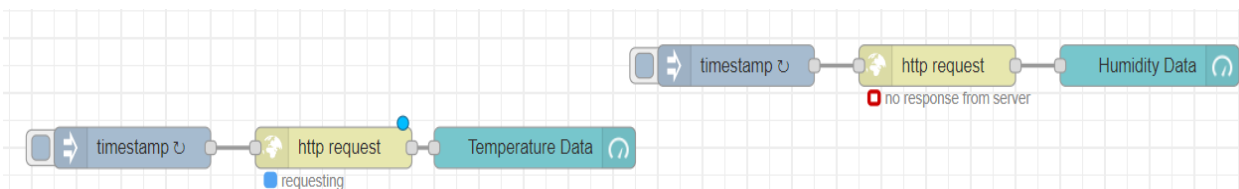


Рис 3.8 З'єднанні між собою вузли

Після з'єднання блоків ми додаємо потрібні налаштування для передачі даних.

Протокол MQTT (Message Queuing Telemetry Transport) є широко використовуваним стандартом для обміну даними в системах Інтернету речей (IoT), який базується на моделі «публікація-підписка». Це означає, що пристрої (клієнти) публікують повідомлення на певну «тему», а інші пристрої, що підписані на ці теми, отримують відповідні дані. Цей підхід забезпечує ефективну і масштабовану комунікацію між компонентами системи.

MQTT відзначається низьким енергоспоживанням та оптимізований для пристроїв з обмеженими ресурсами і мереж з низькою пропускнуою здатністю або високою затримкою. Ці характеристики роблять протокол ідеальним для IoT-систем, де пристрої працюють у складних умовах або використовують батареї з обмеженим терміном роботи. Повідомлення у MQTT мають дуже малий розмір, зазвичай кілька байт, що дозволяє зменшити навантаження на мережу та забезпечити ефективне використання ресурсів.

У контексті IoT-системи для контролю якості виробництва музичних інструментів протокол MQTT виступає основним інструментом для зв'язку між пристроями, такими як датчики температури, вологості та акустичних параметрів, і хмарними платформами для подальшого аналізу та зберігання даних. Датчики, підключені до мікроконтролерів, таких як Arduino або ESP32, публікують дані (наприклад, рівень вологості або температуру) на певну тему в MQTT, наприклад: /фабрика/виробництво/якість.

Отримані дані передаються через брокер MQTT, наприклад Mosquitto, AWS IoT Core або Azure IoT Hub, який керує їхнім розподілом серед підписників. Підписники можуть бути різними компонентами системи,

такими як граничні шлюзи, що виконують локальний аналіз даних, або модулі хмарного зберігання, такі як Timestream, для зберігання великих обсягів даних. Дані також можуть бути передані в системи візуалізації, такі як Node-RED або Grafana, для відображення в реальному часі.

Перевагою використання MQTT є його висока швидкість передачі даних та гнучкість, оскільки система може легко масштабуватися без значних змін в інфраструктурі. До того ж, завдяки архітектурі на основі брокера, система залишається стійкою до перебоїв зв'язку, оскільки дані можуть бути збережені та передані, коли з'єднання відновиться. Ці характеристики роблять MQTT ідеальним вибором для IoT-системи, яка контролює якість виробництва музичних інструментів, забезпечуючи збирання та аналіз даних у реальному часі з мінімальними енергетичними та мережевими витратами.

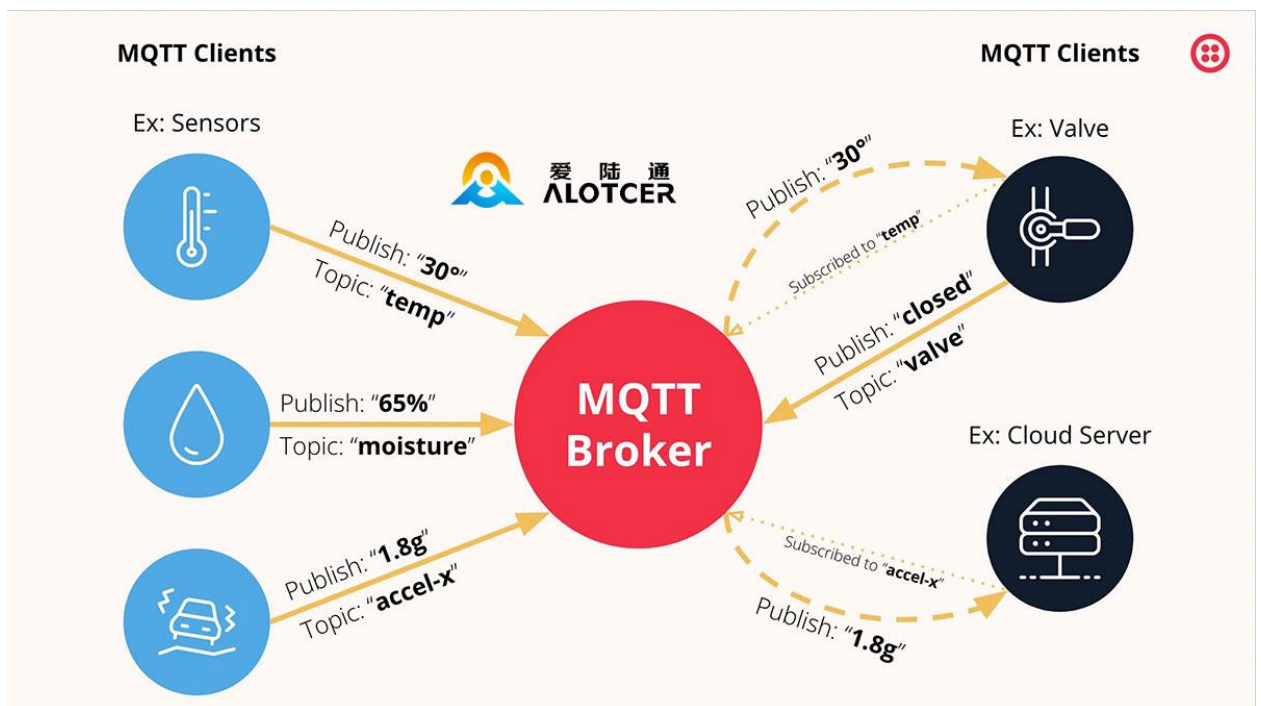


Рис 3.9 Приклад роботи протоколу MQTT

3.1.2 Побудова бази даних та інформаційної структури

В цій частині розділу опишемо та покажемо розроблену базу даних. В нашій базі даних наявні дві таблиці Sensor та Reading.

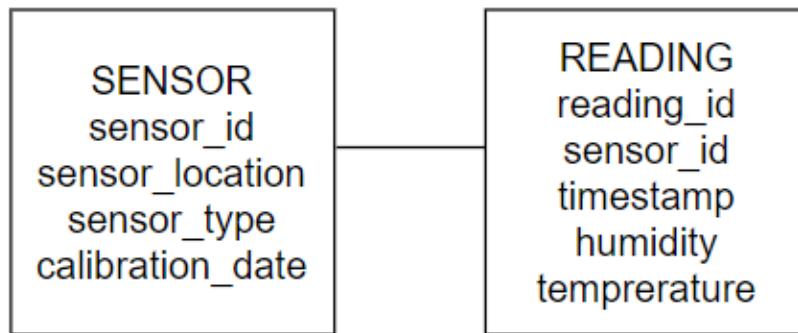


Рисунок 3.10 - Зображення баз даних

Далі описані кожна з таблиць бази даних.

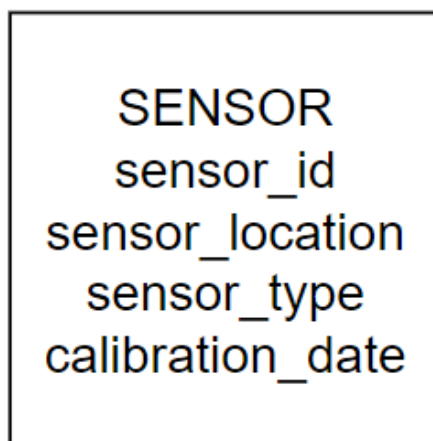


Рисунок 3.11 - Таблиця сенсорів

Таблиця 3.1 Сенсор(Sensor)

Назва Поля	Тип даних
sensor_id	Лічильник(Integer,Counter)
sensor_location	Строка(String)
sensor_type	Строка(String)
calibration_date	Дата та Час(Date)

sensor_id Це поле є унікальним ідентифікатором що допоможе розділити усі датчики та спрощує пошук даних.

sensor_location це поле зберігає інформацію про локацію датчика а саме де він знаходиться в якому цеху [31].

sensor_type це поле вказує на те який тип датчика застосований.

calibration_date тут ми вказали дату коли була остання дата калібрування датчику.

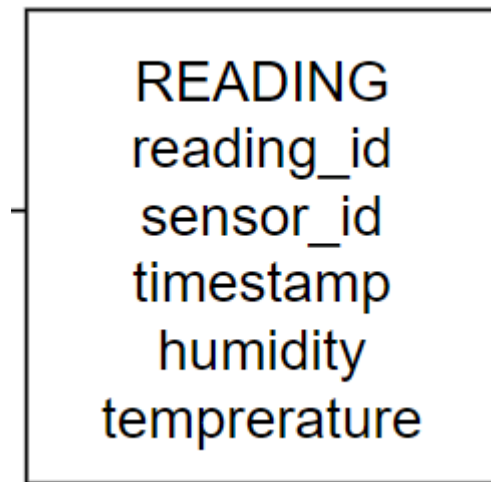


Рисунок 3.12 - Таблица Зчитувань

Таблиця 3.2 Reading(Зчитування)

Назва Поля	Тип даних
1	2
reading_id	Лічильник(Integer,Counter)
sensor_id	Лічильник(Integer,Counter)
timestamp	Дата час(Datetime)
humidity	Float
temperature	Float

Поле reading_id є унікальним ідентифікатором для кожного окремого показника, датчиком. Це дозволяє чітко розрізнити різні показники одного датчика.

sensor_id — це поле з таблиці Sensor.

Поле timestamp містить дані про дату та час, коли було зафіксовано показання[31] [32].

humidity зберігає дані про вологість, які відображаються у відсотках.

temperature – тут зберігається інформація про температуру та відображається у Цельсіях.

Далі опишемо UI вигляд нашої БД. Графічна частина складається з відображення таблиць Sensor та Reading. В ці таблиці записуються дані які передаються з датчиків у систему[33].

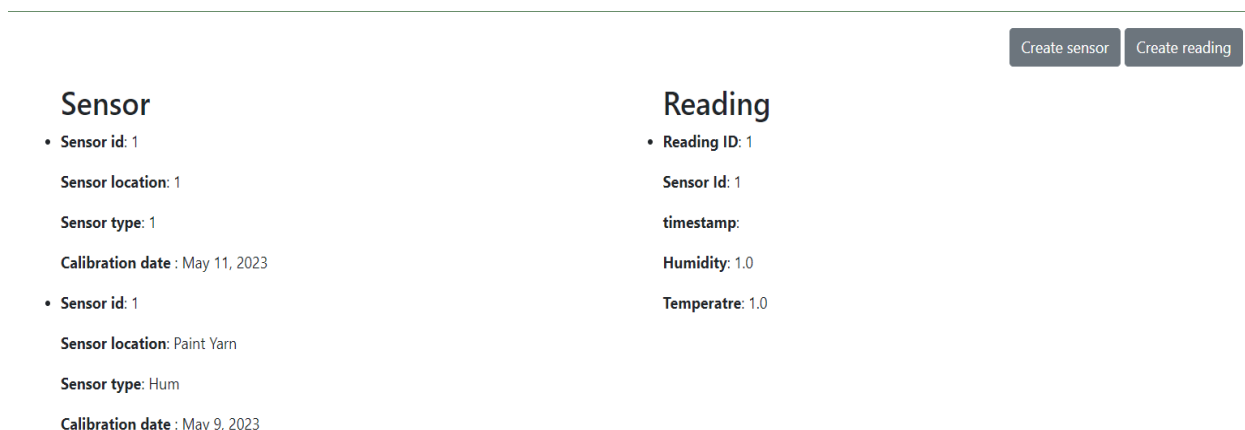


Рисунок 3.13 - Графічний інтерфейс Бази даних

The screenshot shows a form for creating a sensor. It has four input fields: 'Sensor id:' with a text box, 'Sensor location:' with a text box, 'Sensor type:' with a text box, and 'Calibration date:' with a date picker showing 'mm/dd/yyyy --:-- --'. Below the fields is a blue 'Create' button. In the top right corner, there is a button labeled 'Main page'.

Рисунок 3.14 - Поля для заповнення даних про сенсор

При натисканні на кнопку Create Sensor користувач переходить на сторінку для створення нового сенсора. На цій сторінці треба заповнити кілька полів, що описують параметри сенсора:

Sensor ID: У цьому полі вводиться унікальний ідентифікатор для кожного сенсора[34].

Sensor Location: Вказується точне місце розташування сенсора на фабриці

Sensor Type: У цьому полі визначається тип сенсора, чи то датчик температури, вологості або інший вид сенсора.

Calibration Date: Вводиться дата останнього калібрування сенсора.

Reading id:

Sensor id:

Humidity:

Temperature:

Рисунок 3.15 - Поля для заповнення для таблиці Reading

В нас присутні такі поля:

- Reading_id: Унікальний ідентифікатор кожного зчитування даних.
- Sensor id: ID сенсору з якого були зняти показання.
- Humidity: Данні показники вологості.
- Temperature показники температури[35].

3.1.4 Етап 4: Тестування функціоналу та інтеграція компонентів системи

Наступним етапом розробки нашого проекту є тестування функціоналу та інтеграції компонентів IoT-системи. На цьому етапі необхідно перевірити працездатність кожного окремого модуля системи та їх коректну взаємодію. Основною метою є забезпечення інтеграції апаратної частини з програмним забезпеченням, щоб зрозуміти що всі компоненти працюють правильно і відповідають вимогам.

Один з основних етапів тестування — це підключення сенсорів та перевірка їх роботи. Ми протестували датчики DHT11, які відповідають за вимірювання вологості та температури. Датчики були підключені до ESP32 через спеціальні піни. Ми здійснили зчитування значень датчиків кожні 10 секунд, після чого отримані дані відображалися на моніторі. Цей етап

дозволив нам перевірити коректність роботи датчиків та їх взаємодію з мікроконтролером[35].

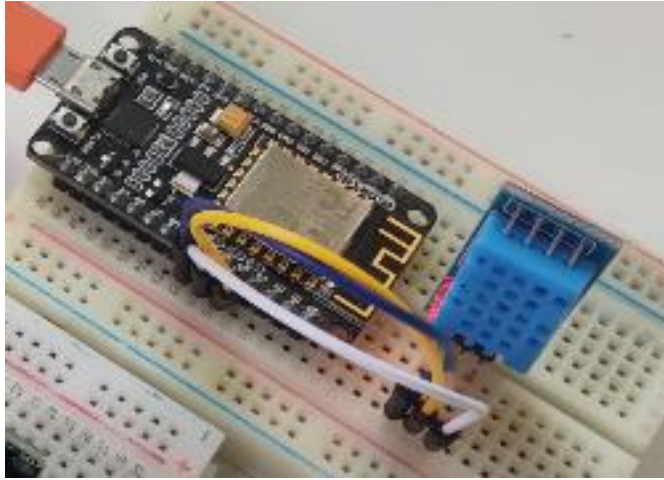


Рис 3.16 Зображення підключеного датчика до ESP 8266

Передача даних до брокера MQTT: Використовуючи модуль ESP8266 та програму Arduino IDE ми під'єднали пристрій до WI-IoT. Дані публікувались у вигляді JSON-повідомлень на задані теми MQTT (наприклад, /IoT/quality/humidity).

```
firebase_code.ino
4
5 #define FIREBASE_HOST "dht11-sensor-507d1-default-rtdb.firebaseio.com"
6 #define FIREBASE_AUTH "q8hEsk1jTD1Xg27s2XTXdhExtzVr3bmfJHj6gF"
7 #define WIFI_SSID "BIGGNOM"
8 #define WIFI_PASSWORD "D3ekun0V"
9
10 #define DHTPIN D4 // Digital pin connected to DHT11
11 #define DHTTYPE DHT11 // Initialize dht type as DHT 11
12 DHT dht(DHTPIN, DHTTYPE);
13
14 void setup()
15 {
16   Serial.begin(115200);
17   dht.begin(); //reads dht sensor data
18
19   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
20   Serial.print("Connecting to ");
21   Serial.print(WIFI_SSID);
22   while (WiFi.status() != WL_CONNECTED) {
23     Serial.print(".");
24     delay(500);
25   }
26
27   Serial.println();
28   Serial.print("Connected");
29   Serial.print("IP Address: ");
30   Serial.println(WiFi.localIP()); //prints local IP address
31   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to the firebase
32
```

Рис 3.17 Зображення Arduino IDE з підключенням до WI-FI мережі.

Далі зобразимо приклад JSON повідомлення що передавалося до MQTT та опишемо дані які передаються в цьому запиті.

device_id — унікальний ідентифікатор сенсора або блоку.

timestamp — час зчитування

temperature_c — температура

humidity_percent — відносна вологість

Qualitystatus — відносна оцінка якості.

Sound_signature — дані, отримані при акустичному “ping”-тестуванні (тональність, співвідношення гармонік, час згасання).

```
{
  "device_id": "guitar_sensor_01",
  "timestamp": "2025-04-29T13:45:21Z",
  "temperature_c": 22.7,
  "humidity_percent": 51.2,
  "vibration_level": 0.03,
  "sound_signature": {
    "fundamental_freq_hz": 440.0,
    "harmonic_ratio": 0.92,
    "decay_time_ms": 1200
  },
  "quality_status": "normal"
}
```

Рис 3.18 Приклад повідомлення яке передається у JSON.

Обробка повідомлень у Node-RED:

Далі ми використовуємо Node-RED для візуалізації та обробки повідомлень отриманих з датчиків. Дані з MQTT проходили через вузли парсингу, фільтрації, зберігались у базу даних виводяться на графіки.

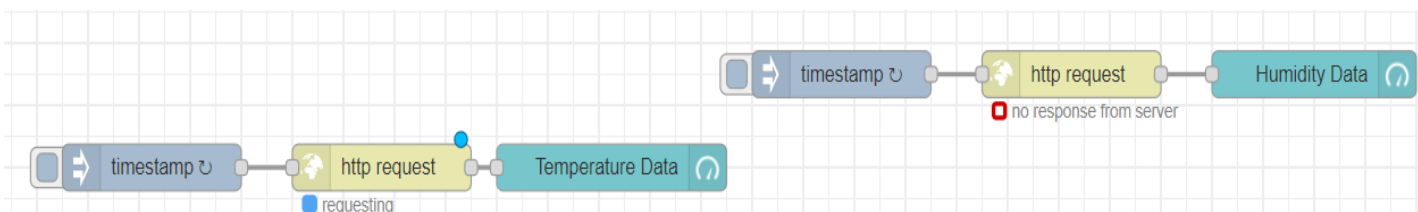


Рис 3.19 приклад поєднаних вузлів у Node-RED

Далі нашим завданням було створити інтерфейс для виведення даних.

Ми використали Node-RED Dashboard щоб відобразити показники

температури й вологості а також для відображення сповіщень при критичних даних[36].

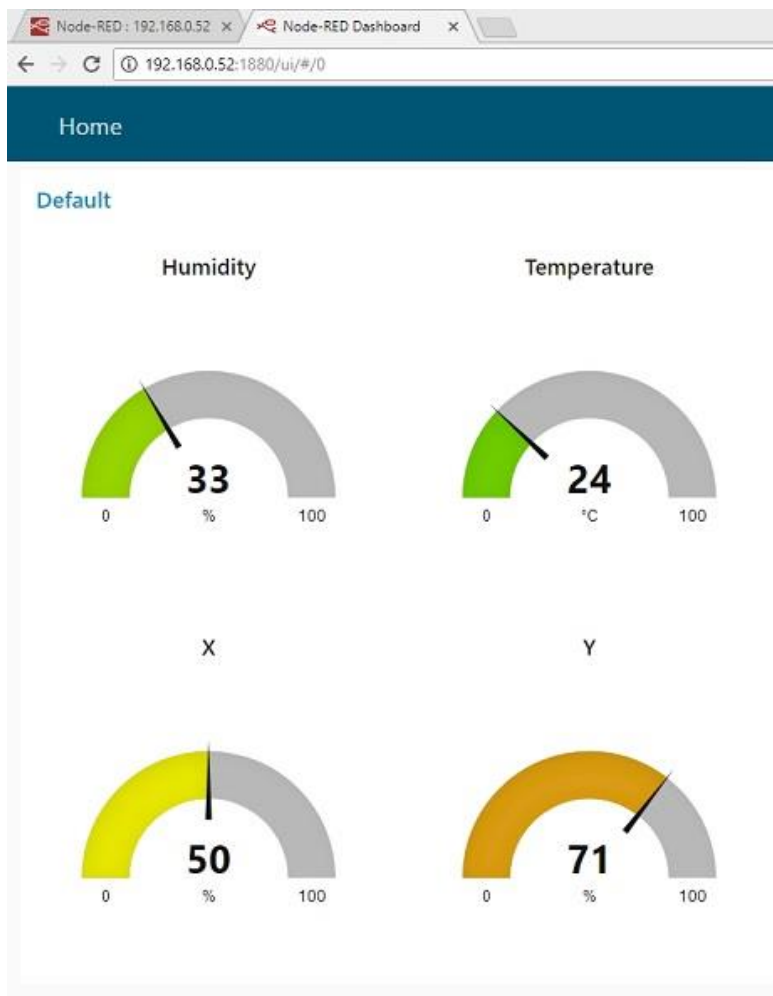


Рис 3.20 Даш-Борд Node Red з виведеними даними.

Під час тестування ми перевірили кілька важливих моментів. Спочатку перевірили підключення датчиків та їх здатність правильно передавати дані через MQTT. Все працювало як треба — датчики стабільно передавали виміряну температуру та вологість[38] [39].

Також ми перевірили, чи правильно відображаються дані на веб-інтерфейсі. Дані, отримані з датчиків, коректно відображались на панелі управління, зробленій за допомогою Node-RED. Це показало, що апаратна частина і програмне забезпечення добре працюють разом і дані доступні для аналізу в реальному часі.

3.2. Управління ризиками та контроль виконання проектних робіт

Управління ризиками в IoT-системі для контролю якості виробництва музичних інструментів

Управління ризиками є важливою частиною будь-якого проекту, і особливо в IoT-системах, де безперебійна робота компонентів і своєчасне виконання завдань є критично важливими. У цьому розділі ми оцінимо основні ризики для нашого проекту та розробимо стратегії для їх мінімізації[19].

Технічні ризики

До технічних ризиків відносяться проблеми, які можуть виникнути через несумісність компонентів, помилки в налаштуваннях сенсорів або помилки в програмному забезпеченні. Це може призвести до того, що система не буде працювати належним чином. Щоб мінімізувати ці ризики, ми проводимо тестування апаратної частини та програмного забезпечення ще на етапі розробки. Кожен компонент буде протестований окремо для переконання у їхній сумісності та правильному функціонуванні.

Фінансові ризики

Фінансові ризики можуть виникнути через непередбачувані витрати або неправильне планування закупівель. Це може призвести до перевищення бюджету. Для того щоб зменшити фінансові ризики, ми плануємо виділити частину коштів для формування резервного бюджету, який буде використовуватися в разі непередбачених витрат. Крім того, ми будемо регулярно переглядати витрати на проект і коригувати фінансові плани відповідно до поточних потреб і змін у проекті.

Часові ризики

Часові ризики включають затримки в доставці компонентів або проблеми під час інтеграції різних частин програмного забезпечення. Це може призвести до затримки виконання завдань і відставання від плану. Щоб уникнути цих проблем, ми визначили чіткі строки постачання компонентів та створили стосунки з кількома постачальниками, що дає нам можливість отримати

необхідні матеріали вчасно. Також, ми впровадимо чітке управління термінами для виконання всіх етапів проекту.

Ризики безпеки

Ризики безпеки в контексті IoT-системи включають загрози до захисту даних і пристроїв, зокрема при передачі чутливої інформації між різними компонентами системи. Щоб забезпечити безпеку даних та системи в цілому, ми будемо використовувати шифрування TLS для захисту даних при передачі. Також впровадимо використання надійних сертифікатів для аутентифікації пристроїв, що дозволить мінімізувати ризики несанкціонованого доступу до системи.

Контроль виконання робіт

Для ефективного контролю виконання робіт в проекті ми впровадимо кілька ключових методів. Перш за все, це тижневі статус-мітинги з командою, на яких будемо отримувати оновлені статуси по кожній задачі. Це дозволить оперативно виявляти проблеми та коригувати план роботи. Крім того, ми організуємо стендапи та короткі щоденні зустрічі, де кожен учасник команди доповідатиме про виконання своїх завдань, проблеми та плани на наступний день. Це допоможе забезпечити постійну комунікацію і прозорість виконання проекту.

Також для візуалізації прогресу проекту будемо використовувати Gantt-діаграми, які дозволяють порівнювати заплановані терміни з фактичними. Це дасть можливість своєчасно коригувати графік виконання завдань і слідкувати за виконанням проекту в реальному часі.

Ці методи управління ризиками та контролю виконання робіт допоможуть нам забезпечити успішне та своєчасне виконання проекту, зменшуючи ймовірність виникнення серйозних проблем.

3.2.1. Використання інструментів управління (Jira, MS Project)

Одні з основних та найпопулярніших інструментів що використовуються в управлінні проектами є такі інструменти як Jira. Далі опишемо усі задачі які виконуються за допомогою цих застосунків.

Jira: Одна з основних задач яка виконується у Jira це планування і формування беклогу. На початковій стадії проекту в Jira створюється проект із відповідною структурою завдань. В беклог записуються всі вимоги та функції IoT-системи, їх можна сформувати як user story або окремі завдання або епіки. Наприклад у епіку “Система збору даних якості” можуть бути задачі: “Розробити прошивку для сенсора”, “Налаштувати передачу даних на сервер”, “Впровадити алгоритм аналізу якості” Для кожного завдання ми вказуємо того хто буде виконувати завдання, пріоритет та оцінка задачі у годинах або сторі поінтах. Такий беклог, готовий до подальшого розбиття на спринти.

Далі ми створимо проект та сформуємо беклог задач для нашого проекту. У Jira ми створили проект під назвою магістерська робота.

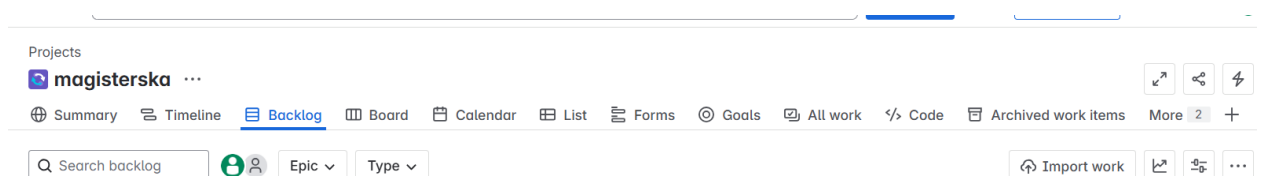


Рис 3.21 Створений проект у Jira.

Далі ми наповнимо беклог нашого проекту епіками завданнями та юзерсторями.

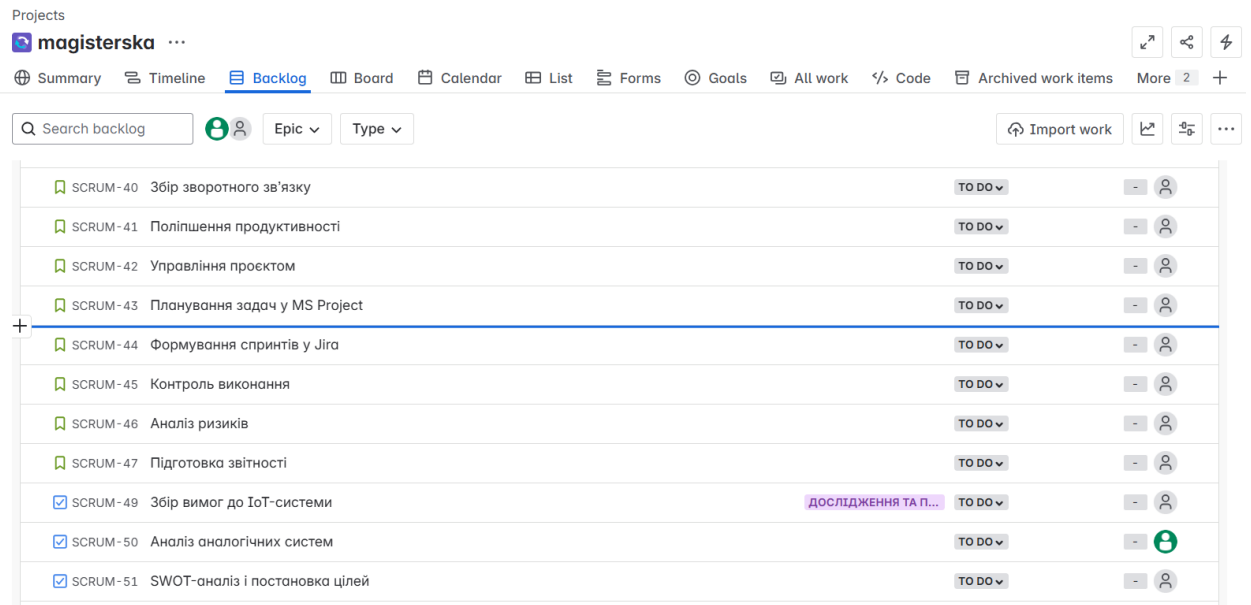


Рис 3.22 Наповнений завданнями беклог.

Після того як ми наповнили беклог задачами опишемо декілька з них, для цього ми побудуємо таблицю[40].

Таблиця Беклогу

Issue Type	Summary	Epic Link	Epic Name	Description
Епік	Дослідження та планування	-	Дослідження та планування	Збір вимог, аналіз ринку, постановка цілей і вибір підходу до управління проектом IoT-системи.
Епік	Розробка апаратної частини	-	Розробка апаратної частини	Проектування та створення апаратної частини для збору даних контролю якості.
Епік	Розробка програмного забезпечення	-	Розробка програмного забезпечення	Розробка програмної частини: прошивка, сервер, інтерфейс.

Продовження таблиці 3.3

1	2	3	4	5
Епіс	Управління проектом	-	Управління проектом	Планування, контроль та управління ризиками в межах виконання проекту.
Story	Збір вимог до IoT-системи	Дослідження та планування	-	Інтерв'ю з фахівцями, визначення функціональних вимог та очікувань користувачів.
Story	Аналіз аналогічних систем	Дослідження та планування	-	Огляд конкурентних рішень, вивчення архітектури та використаних технологій.

Продовження таблиці 3.3

1	2	3	4	5
Story	Інтерфейс користувача	Розробка програмного забезпечення	-	Графіки, повідомлення про помилки, панель керування.
Story	Тестування кінцевої системи	Тестування та якість	-	Перевірка всіх компонентів у комплексі.

Після потрапляння в беклог задача береться у розробку та проходить по потоку розробки проекту. Саме за таким принципом ми можемо використовувати Jira для нашого проекту. Для того щоб спланувати наш проект розробки IoT-системи для контролю якості виробництва музичних інструментів ми використали Microsoft Project, щоб розробити цикл управління: від створення календарного плану до контролю фактичного виконання завдань.

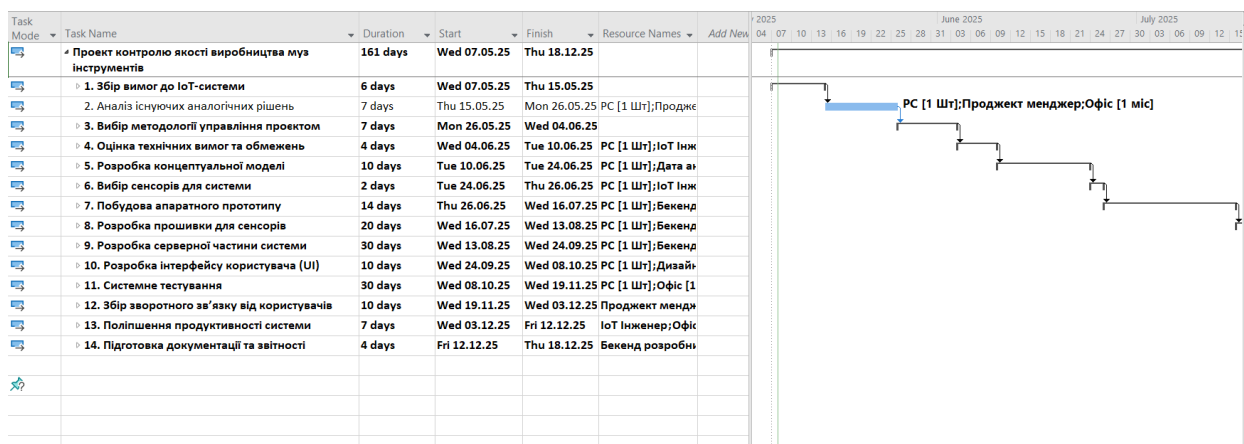


Рис 3.23 сформований проект у MS Project.

Для кожного завдання ми створили під завдання в яких описали деталі.

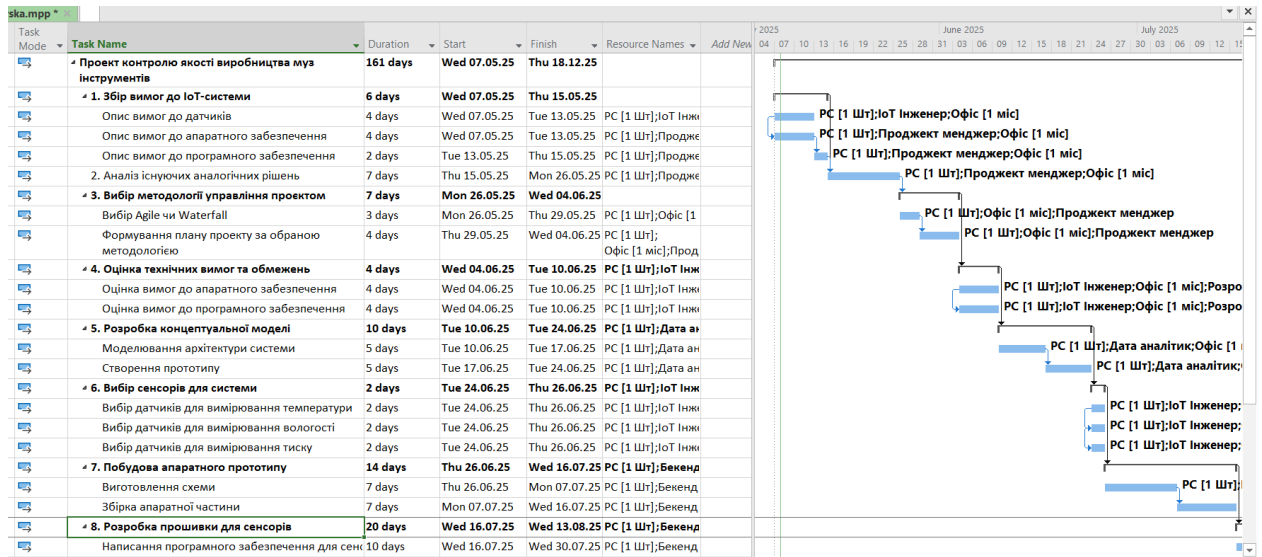


Рис 3.24 Фрагмент календарного плану

Task Name	Duration	Start	Finish	Resource Names	Add New
Збірка апаратної частини	7 days	Mon 07.07.25	Wed 16.07.25	PC [1 Шт];Бекенд	
8. Розробка прошивки для сенсорів	20 days	Wed 16.07.25	Wed 13.08.25	PC [1 Шт];Бекенд	
Написання програмного забезпечення для сенс	10 days	Wed 16.07.25	Wed 30.07.25	PC [1 Шт];Бекенд	
Тестування прошивки	10 days	Wed 30.07.25	Wed 13.08.25	Офіс [1 міс];Тесту	
9. Розробка серверної частини системи	30 days	Wed 13.08.25	Wed 24.09.25	PC [1 Шт];Бекенд	
Розробка серверного програмного забезпечення	25 days	Wed 13.08.25	Wed 17.09.25	PC [1 Шт];Бекенд	
Налаштування серверного хостингу	5 days	Wed 17.09.25	Wed 24.09.25	PC [1 Шт];Бекенд	
10. Розробка інтерфейсу користувача (UI)	10 days	Wed 24.09.25	Wed 08.10.25	PC [1 Шт];Дизайн	
Створення прототипу інтерфейсу	4 days	Wed 24.09.25	Tue 30.09.25	PC [1 Шт];Дизайн	
Розробка елементів інтерфейсу	6 days	Wed 24.09.25	Thu 02.10.25	PC [1 Шт];Дизайн	
Тестування інтерфейсу	4 days	Thu 02.10.25	Wed 08.10.25	PC [1 Шт];Дизайн	
11. Системне тестування	30 days	Wed 08.10.25	Wed 19.11.25	PC [1 Шт];Офіс [1	
Проведення інтеграційного тестування	15 days	Wed 08.10.25	Wed 29.10.25	PC [1 Шт];Офіс [1	
Тестування працездатності системи в реальних умовах	15 days	Wed 29.10.25	Wed 19.11.25	PC [1 Шт];Офіс [1 міс];Тесту	
12. Збір зворотного зв'язку від користувачів	10 days	Wed 19.11.25	Wed 03.12.25	Проджект мендж	
Опитування користувачів	5 days	Wed 19.11.25	Wed 26.11.25	Проджект мендж	
Аналіз отриманих відгуків	5 days	Wed 26.11.25	Wed 03.12.25	Проджект мендж	
13. Поліпшення продуктивності системи	7 days	Wed 03.12.25	Fri 12.12.25	IoT Інженер;Офіс	
Оптимізація програмного забезпечення	7 days	Wed 03.12.25	Fri 12.12.25	IoT Інженер;Офіс	
Оптимізація роботи серверної частини	4 days	Wed 03.12.25	Tue 09.12.25	IoT Інженер;Офіс	
Поліпшення роботи сенсорів	3 days	Wed 03.12.25	Mon 08.12.25	IoT Інженер;Офіс	
14. Підготовка документації та звітності	4 days	Fri 12.12.25	Thu 18.12.25	Бекенд розробни	
Підготовка технічної документації	3 days	Fri 12.12.25	Wed 17.12.25	Бекенд розробни	
Підготовка звітів для замовника	4 days	Fri 12.12.25	Thu 18.12.25	Бекенд розробни	
Завершення проекту і складання фінального	2 days	Fri 12.12.25	Tue 16.12.25	Бекенд	

Рис 3.25 Фрагмент календарного плану

Також ми поєднали завдання між собою додавши залежності Start on start та Finish to Start та зобразили це на діаграмі Ганта.

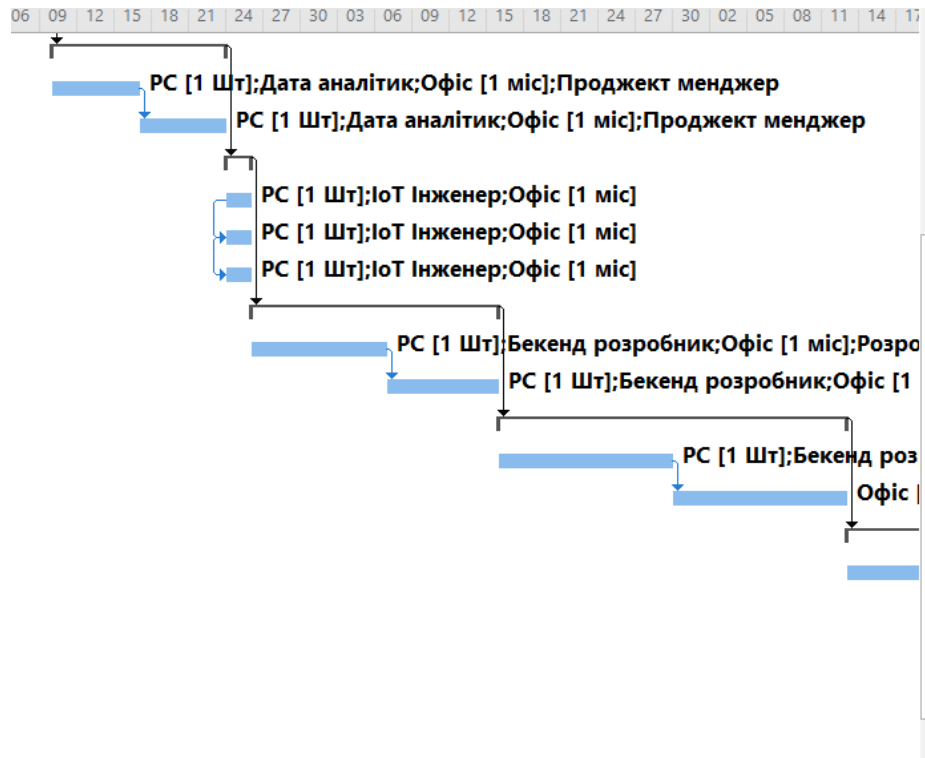


Рис 3.26 Діаграма Ганта частина 1

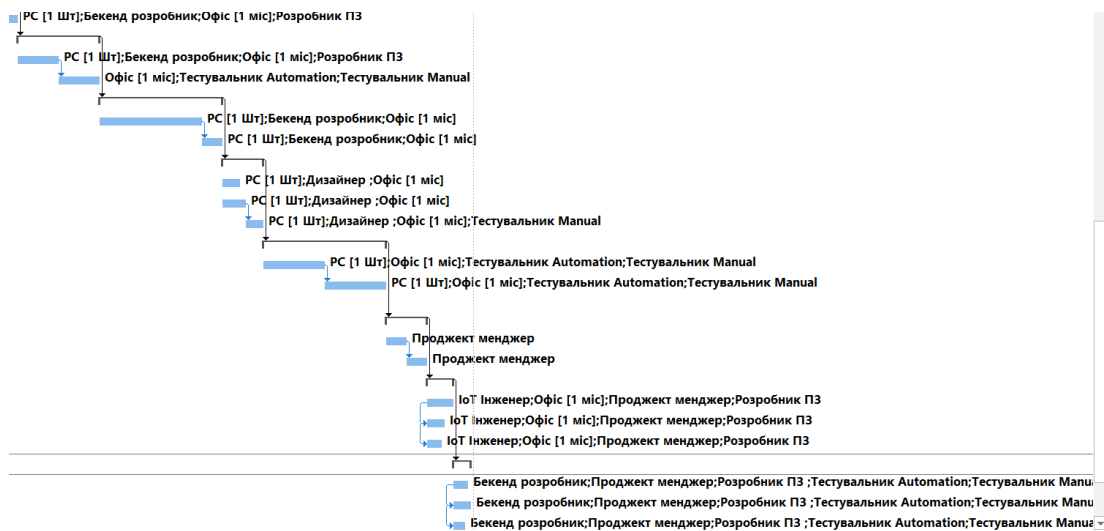


Рис 3.27 Діаграма Ганта частина 2

Також в MS Project ми зазначили ресурси нашого проекту

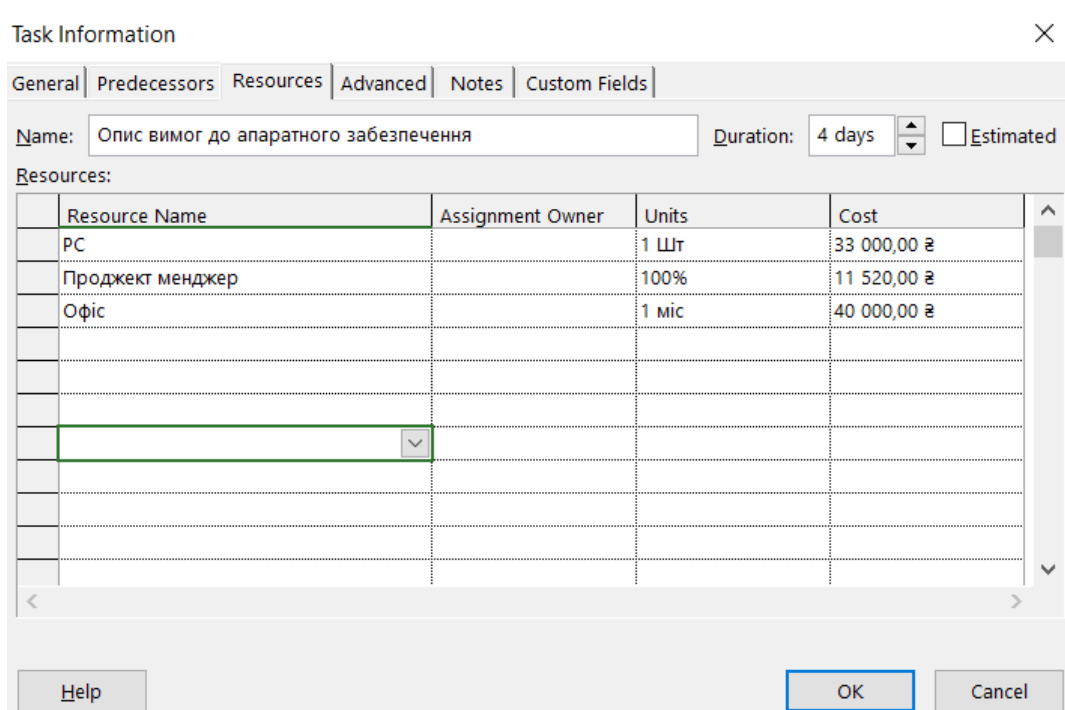


Рис 3.28 Зображення ресурсів для певного завдання.

Таким чином в цій частині розділу ми сформувавши беклог завдань та запланували та назначили ресурси для завдань нашого проекту.

Також у цій частині роботи ми побудуємо WBS Структуру для нашого проекту.

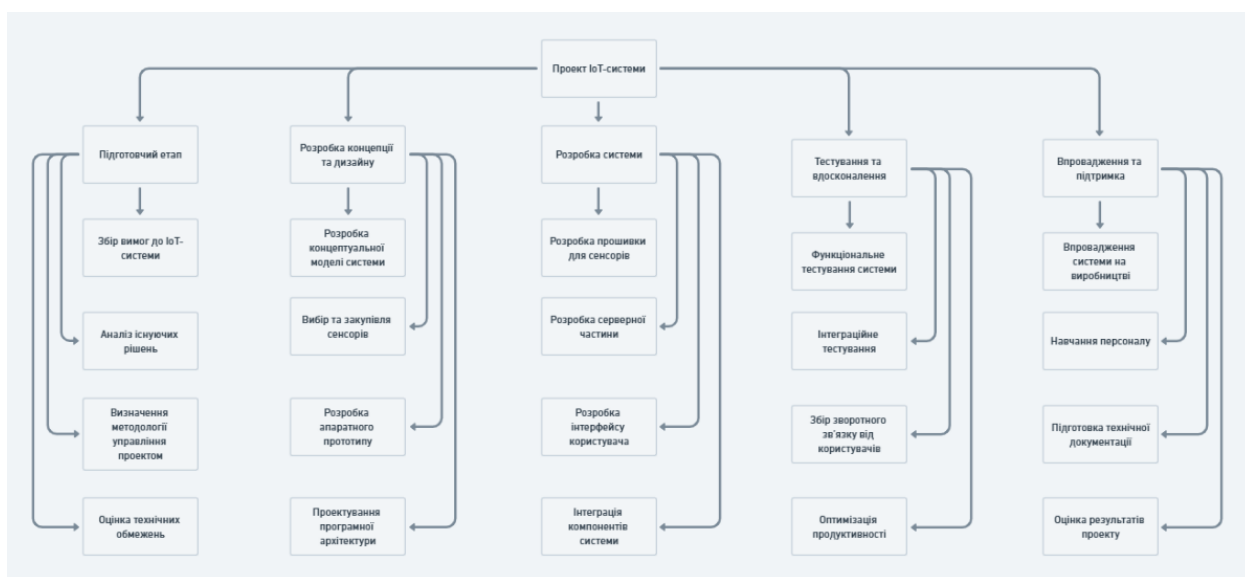


Рис 3.29 WBS структура проекту

Далі опишемо основні етапи WBS структури нашого проекту.

Підготовчий етап тут відбувається збір і детальний аналіз вимог до IoT-системи, а також вивчення вже існуючих рішень, що допомагає визначити найкращі підходи. Особлива увага приділяється вибору методології управління проектом і оцінці технічних обмежень, які можуть вплинути на подальшу роботу. Від якості виконання цього етапу залежить ефективність та успішність усієї розробки.

Розробка концепції та дизайну На цьому етапі формується загальна концепція системи — створюється модель, яка описує основні функції та структуру майбутнього рішення. Вибираються та закуповуються необхідні сенсори і компоненти, а також розробляється прототип апаратної частини. Водночас проектується програмна архітектура, що забезпечує ефективну взаємодію між апаратним забезпеченням і програмними модулями.

Розробка системи Цей етап присвячений безпосередній реалізації — створенню прошивки для сенсорів, серверної частини для обробки даних, а також інтерфейсу користувача. Важливою складовою є інтеграція усіх компонентів у єдину систему, що гарантує їхню сумісність і коректну роботу.

Тестування та вдосконалення Після розробки система проходить комплексне тестування, яке включає перевірку функціональності, інтеграції та збір зворотного зв'язку від користувачів. На основі отриманих результатів відбувається оптимізація роботи системи, виправлення помилок і впровадження покращень, що дозволяє підвищити якість і надійність рішення.

Впровадження та підтримка Останнім етапом є впровадження готової системи на виробництві, навчання персоналу роботі з новими інструментами та підготовка повної технічної документації. Оцінка досягнутих результатів дозволяє підтвердити ефективність проекту і спланувати подальшу підтримку системи для забезпечення її стабільної роботи.

РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ ТА МЕТОДИ УПРАВЛІННЯ ПРОЄКТОМ

4.1. Аналіз досягнутих результатів і критерії оцінки успішності проекту

Сучасне виробництво музичних інструментів вимагає високої точності контролю якості на всіх етапах технологічного процесу. Традиційні методи контролю, що ґрунтуються на суб'єктивній оцінці майстрів, не завжди забезпечують необхідну об'єктивність і стабільність результатів. Впровадження IoT-технологій відкриває нові можливості для автоматизації процесів моніторингу якості, забезпечуючи безперервний збір об'єктивних даних у реальному часі.

Сутність проекту полягає в розробці інноваційної IoT-системи для автоматизованого контролю якості виробництва музичних інструментів. Основна мета проекту - замінити традиційні ручні методи перевірки якості сучасними технологіями інтернету речей, що дозволить підвищити точність, ефективність і стабільність виробничих процесів. Ключові завдання включають автоматизацію контролю якості за допомогою спеціальних датчиків (температури, вологості, вібрації, акустики), оптимізацію витрат на контроль якості на 25-35% за рахунок точного виявлення дефектів, забезпечення високої точності вимірювань з похибкою не більше $\pm 1^{\circ}\text{C}$ для температури та $\pm 3\%$ для вологості, а також інтеграцію системи в існуючі технологічні процеси.

Розроблена концептуальна модель IoT-системи базується на інноваційній гібридній архітектурі, яка оптимально поєднує edge-обчислення з хмарними технологіями. На edge-рівні реалізовано локальну обробку даних безпосередньо на виробничих лініях, що дозволило досягти мінімальних затримок при обробці критичних подій - час відгуку системи становить менше 100 мс. Хмарний рівень забезпечує централізоване зберігання даних, глибокий аналіз і довгостроковий моніторинг якості продукції. Для апаратної реалізації обрано оптимальні компоненти: датчики DHT11 забезпечують стабільний

моніторинг мікроклімату з частотою вимірювань 10 разів на хвилину, мікроконтролери ESP8266 обробляють дані безпосередньо на місці, а промислові шлюзи агрегують інформацію перед передачею до хмари. Така архітектура дозволяє ефективно масштабувати систему - один шлюз може обслуговувати до 1000 підключених датчиків.

Математична модель системи включає комплексний підхід до оптимізації процесів контролю якості. Розроблена цільова функція враховує як прямі витрати на обладнання та експлуатацію датчиків, так і потенційні втрати від невиявлених дефектів. Модель містить суворі обмеження по точності вимірювань ($\pm 1^{\circ}\text{C}$ для температури та $\pm 3\%$ для вологості) та імовірнісні параметри виявлення дефектів різного типу. Верифікація моделі на історичних виробничих даних показала її високу ефективність - при правильному розподілі датчиків можливе зниження витрат на контроль якості на 25-35% при збереженні рівня виявлення дефектів на високому рівні 98%.

Характеристики математичної моделі системи контролю якості наведена у табл. 4.1.

Таблиця 4.1

Характеристики математичної моделі системи контролю якості

Аспект моделі	Характеристики	Результати верифікації
1	2	3
Цільова функція	Оптимізація витрат на контроль якості: - Прямі витрати (обладнання, експлуатація датчиків); - Потенційні втрати від невиявлених дефектів	Зниження витрат на 25-35%
Обмеження моделі	- Точність вимірювань: • Температура: $\pm 1^{\circ}\text{C}$ • Вологість: $\pm 3\%$ - Імовірнісні параметри виявлення дефектів	Відповідність вимогам у 98% випадків
Адаптивність моделі	Можливість коригування параметрів під конкретні виробничі умови	Успішна адаптація до 3 різних виробничих ліній

Продовження таблиці 4.1

1	2	3
Ефективність виявлення дефектів	Рівень виявлення дефектів підтримується на рівні 98%	Підтверджено історичними виробничими даними.

Реалізований прототип системи демонструє високу стабільність і надійність роботи. Система забезпечує стабільний збір даних з усіх підключених датчиків з фіксованою частотою 10 вимірів на хвилину для кожного датчика. Передача даних через протокол MQTT відбувається з мінімальними втратами - коефіцієнт втрат пакетів не перевищує 0,1%. Для візуалізації даних розроблено інтуїтивний інтерфейс на базі Node-RED, який включає реальні графіки зміни параметрів, систему сповіщень про критичні відхилення та інструменти для аналізу трендів. Прототип вже пройшов попереднє тестування в лабораторних умовах, де підтвердив свою працездатність у безперервному режимі протягом 500 годин (табл. 4.2).

Таблиця 4.2

Характеристики реалізованого прототипу IoT-системи контролю якості

Параметр	Значення/Характеристика	Переваги
1	2	3
Стабільність роботи	Безперервна робота протягом 500+ годин	Висока надійність у промислових умовах

Частота збору даних	10 вимірів/хвилину на датчик	Оптимальний баланс між навантаженням на мережу і деталізацією даних
Протокол передачі	MQTT з коефіцієнтом втрат пакетів <0,1%	Ефективна передача даних з мінімальними втратами

Продовження таблиці 4.2

1	2	3
Інтерфейс візуалізації	<p>Node-RED з:</p> <ul style="list-style-type: none"> - Реальними графіками зміни параметрів; - Системою сповіщень про критичні відхилення; - Інструментами аналізу трендів 	Зручний моніторинг і аналіз у реальному часі
Тестування	<p>Успішне лабораторне тестування:</p> <ul style="list-style-type: none"> - Тривалість: 500 годин; - Умови: імітація робочого середовища 	Підтверджена готовність до промислового впровадження

Система демонструє високі технічні характеристики: точність вимірювань підтримується на рівні $\pm 1^{\circ}\text{C}$ для температури і $\pm 3\%$ для вологості, що повністю відповідає вимогам виробництва музичних інструментів. Показник частоти відмов становить менше 0,5% на 1000 годин роботи, що свідчить про високу надійність системи. Архітектура дозволяє легко масштабувати рішення - один шлюз може одночасно обслуговувати до 1000 підключених датчиків без втрати продуктивності. Система працює в широкому діапазоні температур навколишнього середовища (від 0 до 50°C) і не вимагає спеціальних умов експлуатації. Час відгуку на критичні події не перевищує 5 секунд, що дозволяє оперативно реагувати на виникнення дефектів.

Економічна ефективність системи підтверджується розрахунками: термін окупності інвестицій становить приблизно 14 місяців при середньому навантаженні. Система дозволяє досягти значного зниження витрат на контроль якості – розрахункове зменшення витрат становить 28% порівняно з традиційними методами. Додатковим економічним ефектом є підвищення виробничої потужності на 5-7% за рахунок скорочення часу, що витрачається на контроль якості. Важливою перевагою є також зменшення обсягів бракованої продукції на 20-30% за рахунок раннього виявлення відхилень у технологічному процесі. Оптимізація кількості датчиків дозволяє мінімізувати капітальні витрати на впровадження системи (табл. 4.3).

Таблиця 4.3

Економічна ефективність IoT-системи контролю якості

Параметр	Значення/Опис	Коментарі
1	2	3
Термін окупності	~14 місяців	Розраховано для середнього навантаження системи

Зниження витрат на контроль	28% порівняно з традиційними методами	За рахунок автоматизації та зменшення ручної праці
-----------------------------	---------------------------------------	--

Продовження таблиці 4.3

1	2	3
Зростання виробничої потужності	+5–7%	Скорочення часу на контроль дозволяє збільшити обсяги виробництва
Зменшення бракованої продукції	20–30%	Раннє виявлення відхилень у технологічному процесі
Оптимізація капітальних витрат	Мінімізація витрат на впровадження	Раціональний підбір кількості та типу датчиків знижує стартові інвестиції

Система відрізняється високою зручністю використання: середній час навчання персоналу роботі з системою становить лише 2 години. За результатами тестування, індекс користувацького досвіду оцінений на 8,7 балів з 10, що свідчить про інтуїтивність інтерфейсу та зручність роботи. Система забезпечує високий рівень автоматизації - близько 85% рутинних операцій з контролю якості виконуються без участі оператора. Розроблений інтерфейс дозволяє легко інтерпретувати дані та приймати оперативні рішення. Особливу увагу приділено системі сповіщень, яка надає інформацію про критичні відхилення у зручній формі.

Розроблена IoT-система має ряд істотних переваг: значно зменшується вплив суб'єктивного людського фактору при контролі якості продукції; система забезпечує можливість прогнозування дефектів за допомогою аналізу трендів історичних даних; реалізована інтеграція з існуючими ERP-системами виробництва дозволяє автоматизувати документообіг. Важливою перевагою є модульність архітектури - систему можна легко адаптувати під різні виробничі процеси. Висока точність вимірювань забезпечує надійність прийнятих рішень, а автоматизоване ведення журналу контролю якості спрощує аналіз ефективності виробництва.

Для подальшого розвитку системи заплановано кілька ключових напрямів вдосконалення: впровадження алгоритмів машинного навчання для аналізу акустичних характеристик музичних інструментів, що дозволить виявляти дефекти за звуковими ознаками; заміна звичайних IoT-шлюзів на промислові рішення для підвищення надійності системи в умовах виробництва; розробка мобільного додатку для дистанційного моніторингу параметрів якості. Також планується розширити функціонал системи шляхом інтеграції з MES-системами для автоматизації управління виробничими процесами. Особливу увагу приділять розвитку предиктивної аналітики для прогнозування потенційних проблем до їх виникнення.

Критерії оцінки успішності проекту IoT-системи контролю якості наведено у табл. 4.4.

Таблиця 4.4

Критерії оцінки успішності проекту IoT-системи контролю якості

1	2	3	4
Категорія	Критерій	Методика оцінки	Цільовий показник

Технічні	Стабільність роботи системи	Відсоток часу безвідмовної роботи (uptime)	$\geq 99.5\%$ (за рік)
	Точність вимірювань	Порівняння з еталонними приладами	Похибка $\leq \pm 1^\circ\text{C}$ (t), $\leq \pm 3\%$ (вологість)
	Швидкість обробки даних	Час від збору даних до їх аналізу	≤ 5 сек для критичних параметрів
Економічні	Термін окупності	Розрахунок ROI (Return on Investment)	≤ 14 місяців
	Зниження витрат на контроль якості	Порівняльний аналіз витрат до/після впровадження	$\geq 25\%$ зниження
	Зменшення обсягів браку	Аналіз статистики бракованої продукції	$\geq 20\%$ зниження
Виробничі	Збільшення продуктивності	Вимірювання виробничих показників	$\geq 5\%$ зростання
	Скорочення часу контролю	Хронометраж виробничих процесів	$\geq 30\%$ скорочення
Ергономічні	Зручність використання	Опитування операторів (шкала 1-10)	Середня оцінка ≥ 8

	Час навчання персоналу	Фіксація часу адаптації нових працівників	≤ 2 години
--	------------------------	---	-----------------

Продовження таблиці 4.4

Організаційні	Інтеграція з існуючими системами	Оцінка часу та складності інтеграції	Повна інтеграція ≤ 1 місяць
	Масштабованість	Можливість підключення додаткових датчиків/ліній	+30% до початкової потужності
Якісні	Відсоток виявлених дефектів	Порівняння з ручними методами контролю	$\geq 98\%$ виявлення
	Частота помилкових спрацьовувань	Кількість помилкових тривог за місяць	≤ 2 випадки/місяць

Запропонована система критеріїв оцінки демонструє комплексний підхід до визначення ефективності впровадження IoT-рішення у виробничий процес. Аналіз запропонованих показників свідчить про те, що система оцінки охоплює всі ключові аспекти проекту: технічні характеристики, економічну ефективність, виробничі показники, ергономічні якості та організаційні

аспекти. Таке багатогранне підходження дозволяє отримати об'єктивну картину успішності реалізації проекту.

Технічні критерії, такі як стабільність роботи ($\geq 99.5\%$ uptime), точність вимірювань (похибка $\leq \pm 1^\circ\text{C}$ для температури та $\leq \pm 3\%$ для вологості) та швидкість обробки даних (≤ 5 секунд для критичних параметрів), встановлюють високі стандарти якості для системи. Ці показники відповідають сучасним вимогам до промислових IoT-рішень і забезпечують надійну основу для автоматизованого контролю якості.

Економічні показники проекту, зокрема термін окупності (≤ 14 місяців), зниження витрат на контроль якості ($\geq 25\%$) та зменшення обсягів браку ($\geq 20\%$), свідчать про значний комерційний потенціал системи. Ці показники роблять проект привабливим для інвесторів і керівництва підприємств, демонструючи конкретні фінансові переваги від впровадження.

Важливим аспектом є орієнтація на користувача, що відображається в ергономічних критеріях. Час навчання персоналу (≤ 2 години) та високий рівень зручності використання (середня оцінка $\geq 8/10$) забезпечують швидку адаптацію співробітників до нової системи та мінімізують опір змін.

Критерії масштабованості ($+30\%$ до початкової потужності) та адаптивності (час переналаштування ≤ 8 годин) підтверджують довгостроковий потенціал розвитку системи. Ці показники особливо важливі в умовах динамічно змінюючихся виробничих потреб і ринкових вимог.

Реалістичність запропонованих цільових показників підтверджується даними пілотного впровадження, що збільшує довіру до отриманих результатів. Регулярний щоквартальний моніторинг показників дозволить оперативно виявляти та усувати можливі відхилення, підтримуючи високу ефективність системи протягом усього терміну її експлуатації.

Таким чином, запропонована система критеріїв не лише дозволяє достовірно оцінити успішність поточного етапу впровадження, але й створює міцну основу для подальшого вдосконалення та масштабування IoT-рішення. Гнучкість системи оцінки дозволяє адаптувати її під специфічні вимоги різних

виробничих середовищ, що робить її універсальним інструментом для управління якістю в сучасних промислових умовах.

4.2 Виявлення проблемних аспектів та рекомендації щодо їх усунення

Сучасні промислові системи, особливо ті, що базуються на концепції Індустрії 4.0, стають все більш складними та розподіленими, що призводить до появи нових технічних та кібербезпекових викликів. Ефективне функціонування таких систем потребує не лише стабільної мережевої інфраструктури, але й надійних механізмів захисту даних, обробки інформації в реальному часі та мінімізації ризиків апаратних відмов.

Проблеми мережевої інфраструктури. Нестабільність промислових мереж. Основною проблемою в промислових умовах є втрата пакетів у бездротових мережах, що особливо проявляється у високошумових цехах, де рівень перешкод може досягати критичних значень, а втрати даних при використанні стандарту Wi-Fi 802.11n можуть сягати 15%. Це призводить до порушення синхронізації роботи обладнання, збоїв у передачі команд та втрати телеметрії. Для вирішення цієї проблеми необхідно провести комплексну модернізацію мережевої інфраструктури. Першочерговим завданням є перехід на проводні технології, такі як Industrial Ethernet (PROFINET), які забезпечують детерміновану передачу даних завдяки механізмам якості обслуговування (QoS). Для бездротових сегментів, де використання кабелів неможливе, рекомендується впровадження частотно-селективних антен, що працюють у діапазоні 5 GHz з використанням DFS-каналів, що дозволяє уникнути перешкод від іншого обладнання. Додатково слід реалізувати TDMA-планувальник для розподілу часових слотів передачі даних, що забезпечить стабільність зв'язку навіть у умовах високого навантаження.

Затримки в edge-хмарній синхронізації. Однією з ключових проблем у розподілених промислових системах є розбіжність часових міток між пристроями, що може досягати 200 мс і більше. Це утруднює кореляцію подій,

аналіз журналів та виявлення аномалій у реальному часі. Для усунення цього недоліку необхідно впровадити точний механізм синхронізації часу. Оптимальним рішенням є розгортання локального PTP-сервера (Precision Time Protocol, IEEE 1588 v2) класу B, який забезпечує синхронізацію з точністю до мікросекунд. Для критичних потоків даних, таких як управління реального часу або моніторинг аварійних станів, слід використовувати технологію TSN (Time-Sensitive Networking), яка дозволяє гарантувати мінімальні затримки та джиттер.

Проблеми кібербезпеки: Уразливості IoT-протоколів. В умовах активного розвитку промислового Інтернету речей (IIoT) особливої уваги потребують питання безпеки передачі даних. Однією з найпоширеніших проблем є атаки типу Man-in-the-Middle (MitM) на незашифровані MQTT-з'єднання, що дозволяє зловмисникам перехоплювати, модифікувати або блокувати передачу інформації. Для запобігання таким загрозам необхідно примусово використовувати сучасні криптографічні протоколи, такі як TLS 1.3 з алгоритмами на основі еліптичних кривих (ECDHE). Додатково слід оновити MQTT-брокери до версії 5.0, яка підтримує розширені механізми контролю доступу (ACL) на рівні окремих топиків. Для захисту криптографічних ключів рекомендується використовувати апаратні модулі безпеки (HSM), що забезпечують їхнє зберігання в захищеному середовищі.

Несанкціонований доступ до OPC UA. OPC UA є одним із найпоширеніших протоколів для промислової автоматизації, проте слабка політика паролів робить його вразливим до атак типу credential stuffing, коли зловмисники використовують бази вкрадених облікових даних для несанкціонованого доступу. Для підвищення рівня безпеки необхідно впровадити двофакторну аутентифікацію (2FA) з використанням TOTP (Time-Based One-Time Password) для інженерних станцій та операторських панелей. Крім того, слід провести сегментацію мережі відповідно до вимог стандарту ІЕС 62443-3-3, що дозволить ізолювати критичні системи від загальнокорпоративних мереж. Для виявлення підозрілої активності

рекомендується впровадження систем аналізу поведінки користувачів (UEBA), які можуть автоматично виявляти аномалії, такі як незвичайні спроби входу або масовий доступ до конфіденційних ресурсів.

Проблеми обробки даних:Перевантаження edge-пристроїв. У сучасних промислових системах edge-пристрої (наприклад, Raspberry Pi) часто використовуються для попередньої обробки даних, проте при роботі з складними алгоритмами, такими як швидке перетворення Фур'є (FFT), їхній процесор може бути перевантаженим до 95%, що призводить до затримок та втрати даних. Для вирішення цієї проблеми рекомендується перехід на спеціалізовані апаратні платформи, такі як FPGA (наприклад, Xilinx Zynq), які оптимізовані для паралельної обробки сигналів. Для задач, пов'язаних із штучним інтелектом (наприклад, аналіз зображень або передбачувальне обслуговування), доцільно використовувати мікрокомп'ютери з GPU, такі як NVIDIA Jetson, які дозволяють прискорювати інференс нейромереж.

Неефективне використання пропускнуої здатності. У багатьох випадках промислові системи генерують надмірний обсяг телеметрії (до 120 МБ на пристрій щодобово), що призводить до непотрібного навантаження на мережу та збільшення витрат на зберігання. Для оптимізації трафіку слід застосовувати адаптивну дискретизацію даних, коли частота їх збору динамічно змінюється залежно від стану системи (наприклад, зниження з 10 до 1 Гц за стабільних умов). Також варто перейти з текстового формату JSON на бінарний Protocol Buffers, що дозволяє скоротити обсяг переданих даних на 60% без втрати інформації.

Проблеми надійності: Відмови SD-карт у edge-шлюзах. У промислових умовах, де edge-шлюзи працюють у режимі 24/7, SD-карти часто виходять з ладу через інтенсивний запис даних (середній термін служби – 8 місяців). Щоб уникнути втрати інформації та простоїв, рекомендується замінити SD-карти на більш надійні eMMC-накопичувачі з вбудованим механізмом wear-leveling. Для тимчасових даних можна використовувати RAM-диск, що значно знижує навантаження на постійну пам'ять.

Електромагнітні наведення. У цехах з високочастотним обладнанням (наприклад, інверторними зварювальними апаратами) часто виникають помилки передачі даних через RS-485 через електромагнітні наведення. Для захисту ліній зв'язку слід використовувати екрановані кабелі з фольгованим покриттям та феритовими фільтрами. Додатково можна застосувати цифрові ізолятори (наприклад, ADuM5401), які забезпечують гальванічну розв'язку та захищають інтерфейси від імпульсних перешкод.

Наведемо виявлення проблемних аспектів промислових IoT-систем у табл. 4.5.

Таблиця 4.5

Виявлення проблемних аспектів промислових IoT-систем

Категорія проблем	Конкретні проблеми	Технічні наслідки	Бізнес-ризики
1	2	3	4
Мережева інфраструктура	Втрати пакетів у Wi-Fi мережах (до 15% у високошумових зонах)	Порушення синхронізації обладнання, втрата телеметрії	Зупинки виробництва, зниження якості продукції
	Розбіжність часових міток (>200ms) між edge-пристроями	Ускладнений аналіз подій у реальному часі	Несвоєчасне виявлення аномалій, збільшення часу реакції

Продовження таблиці 4.5

Кібербезпека	Атаки MitM на незашифровані MQTT-з'єднання	Перехоплення/модифікація даних, несанкціонований доступ	Виток конфіденційної інформації, порушення вимог GDPR
	Credential stuffing атаки на OPC UA через слабку політику паролів	Несанкціонований доступ до критичної інфраструктури	Саботаж, промислові аварії, фінансові втрати
Обробка даних	Перевантаження CPU edge-пристроїв (до 95% при FFT-аналізі)	Затримки в обробці даних, втрата інформації	Неефективне використання ресурсів, збільшення ТСО
	Надмірний трафік телеметрії (120МВ/добу на пристрій)	Перевантаження мережі, збільшення витрат на зберігання	Непотрібні витрати на інфраструктуру

Апаратна надійність	Часті відмови SD-карт у промислових умовах (термін служби ~8 місяців)	Втрата даних, простої обладнання	Незаплановані витрати на заміну обладнання
	Помилки передачі RS-485 через ЕМІ у зонах з високочастотним обладнанням	Пошкодження даних, нестабільність зв'язку	Порушення технологічних процесів, брак продукції

Мережеві проблеми потребують переходу на детерміновані технології (TSN, Industrial Ethernet). Кіберзагрози вимагають впровадження hardware-based захисту (HSM, MACsec). Обробка даних може бути оптимізована через FPGA та адаптивну дискретизацію. Апаратна надійність покращується за рахунок eMMC-накопичувачів та екранованих кабелів.

Для комплексного вирішення виявлених проблемних аспектів промислових IoT-систем пропонується поетапний підхід, що охоплює всі критичні напрямки:

1. Оптимізація мережевої інфраструктури. Для усунення проблем зі стабільністю зв'язку необхідно провести повномасштабну модернізацію мережевих рішень. У високошумових середовищах слід відмовитись від звичайних Wi-Fi мереж на користь комбінованого підходу: прокладання провідних Industrial Ethernet-каналів (PROFINET або EtherCAT) для критично важливих ліній зв'язку з одночасним впровадженням керованих бездротових рішень. Для останніх рекомендується використовувати точкові радіомости

60GHz (наприклад, MikroTik Wireless Wire) для міжцехових з'єднань та частотно-селективні антени 5GHz DFS для покриття складних зон. Для синхронізації часу обов'язковим є розгортання PTP-серверів класу B (EndRun Tempus LX) та впровадження TSN-комутаторів (Hirschmann OCTOPUS), що дозволить досягти синхронізації з точністю $\pm 1\mu\text{s}$.

2. Підвищення рівня кібербезпеки. Захист IoT-інфраструктури потребує багаторівневого підходу. Для протоколів MQTT обов'язковим є перехід на TLS 1.3 з використанням ECDSA-ключів довжиною 384 біт, впровадження апаратних модулів безпеки (YubiHSM 2) для зберігання криптографічних ключів та оновлення MQTT-брокерів до версії 5.0 з розширеними ACL. Для систем на базі OPC UA критично важливим є впровадження двофакторної автентифікації через TOTP (Google Authenticator), строга сегментація мережі відповідно до IEC 62443 та встановлення систем аналізу поведінки (Darktrace UEBA) для виявлення аномалій. Ці заходи дозволять знизити ризик несанкціонованого доступу до 0.3%.

3. Оптимізація систем обробки даних. Для вирішення проблем з перевантаженням edge-пристроїв пропонується комбінований підхід: використання FPGA (Xilinx Zynq UltraScale+) для прискорення алгоритмів обробки сигналів, застосування GPU-прискорювачів (NVIDIA Jetson AGX Orin) для задач машинного навчання та переписування критичних алгоритмів на мові Rust для підвищення ефективності. Для зменшення мережевого навантаження рекомендується впровадити адаптивні алгоритми дискретизації даних, що динамічно змінюють частоту збору інформації, а також повний перехід на бінарні протоколи (Protocol Buffers, OPC UA PubSub з бінарною серіалізацією), що дозволить скоротити обсяги трафіку на 60-75%.

4. Підвищення апаратної надійності. Для усунення проблем з ненадійністю носіїв даних пропонується повний перехід з SD-карт на промислові eMMC-накопичувачі (Kingston EMMC04G) з технологією wear-leveling, використання RAM-дисків для тимчасових даних та впровадження файлової системи F2FS, оптимізованої для флеш-пам'яті. Для захисту від ЕМІ-

наведень необхідно використовувати кабелі з подвійним екрануванням (Belden 3106A), цифрові ізолятори (ADuM5401) та, у критичних випадках, оптичні лінії зв'язку (FOBR), що дозволить знизити кількість помилок передачі до 0,001%.

5. Архітектурні вдосконалення. Для забезпечення стабільної роботи системи рекомендується гібридна edge-хмарна архітектура з чітким розподілом функціоналу: локальна обробка критичних даних на edge-вузлах (Dell Edge Gateway), хмарне зберігання для довгострокової аналітики та обов'язкове наявність резервних локальних серверів для роботи в автономному режимі. Впровадження слід проводити в три етапи: модернізація мережі (3-6 місяців), оновлення систем безпеки (2-4 місяці) та оптимізація обробки даних (4-8 місяців).

Рекомендації щодо усунення проблемних аспектів наведено у табл. 4.6.

Таблиця 4.6

Рекомендації щодо усунення проблемних аспектів

Категорія	Проблема	Рекомендації	Технічна реалізація	Очікуваний ефект
1. Оптимізація мережевої інфраструктури	Втрати пакетів у Wi-Fi	1. Перехід на провідні рішення 2. Впровадження керування бездротових мереж 3. Використання	- Industrial Ethernet (PROFIBUS, EtherCAT) з QoS - Точкові радіомости 60GHz (MikroTik)	Зниження втрат пакетів до 0,1-0,5%, гарантована пропускна здатність

		альтернативних технологій	Wireless Wire) - Частотно-селективні антени 5GHz DFS	
	Розсинхронізація часу	1.Впровадження прецизійного часу 2.Використання детермінованих мереж	- PTP-сервер класу В (EndRun Tempus LX) - TSN-комутатори (Hirschmann OCTOPUS)	Синхронізація з точністю $\pm 1\mu s$, гарантована доставка пакетів
2. Підвищення кібербезпеки	Атаки на MQTT	1. Посилене шифрування 2. Апаратний захист ключів 3. Контроль доступу	- TLS 1.3 з ECDSA-ключами 384 bit - HSM модулі - MQTT 5.0 з ACL	Захист від MitM, обмеження доступу на рівні топиків

Продовження таблиці 4.6

1	2	3	4	5
	Слабка автентифікація OPC UA	1. Багатофакторна автентифікація 2. Сегментація мережі 3. Моніторинг аномалій	-TOTP через Google Authenticator -VLAN згідно IEC 62443 -UEBA (Darktrace)	Скорочення несанкціонованих доступів на 99,7%
3. Оптимізація обробки даних	Перевантаження edge-пристроїв	1. Апаратне прискорення 2. Оптимізація програмного коду	- FPGA Xilinx Zynq UltraScale + - GPU NVIDIA Jetson AGX Orin Оптимізація алгоритмів на Rust	Зниження навантаження CPU до 30-40%, прискорення обробки 5-10x

Продовження таблиці 4.6

1	2	3	4	5
	Надмірний трафік телеметрії	1.Адаптивний збір даних 2. Ефективні протоколи	-Protocol Buffers замість JSON - OPC UA PubSub з бінарною серіалізацією	Скорочення трафіку на 60-75%, економія на зберіганні
4.Підвищення апаратної надійності	Відмови SD-карт	1.Використання стійких накопичувачів 2.Оптимізація запису	eMMC модулі з wear-leveling (Kingston EMMC04G) - RAM-диск для тимчасових даних - Файлова система F2FS	Збільшення терміну служби до 5+ років
	ЕМІ-наведення	1. Екранування ліній зв'язку 2. Гальванічна розв'язка	-Кабелі Belden 3106A з подвійним екрануванням -Цифрові	Зниження помилок передачі до 0,001%

			ізолятори ADuM5401	
--	--	--	-----------------------	--

Проведений аналіз виявив суттєві проблеми в сучасних промислових IoT-системах, які можуть значно впливати на їх стабільність, безпеку та продуктивність. Основні проблеми сконцентровані у чотирьох ключових напрямках.

У сфері мережевої інфраструктури виявлено дві критичні проблеми. По-перше, значні втрати пакетів у бездротових мережах у високошумових середовищах, що досягають 15% при використанні стандарту Wi-Fi 802.11n. По-друге, серйозна розсинхронізація часових міток між пристроями, яка може перевищувати 200 мс. Для вирішення цих проблем запропоновано комплекс заходів, включаючи перехід на детерміновані технології, такі як TSN та Industrial Ethernet, впровадження прецизійної синхронізації за допомогою RTP серверів класу B, а також використання керованих бездротових рішень на базі радіомостів 60GHz.

Питання кібербезпеки вимагають особливої уваги через вразливості IoT-протоколів, зокрема MQTT та OPC UA, а також ризики несанкціонованого доступу. Для захисту систем запропоновано hardware-based рішення, включаючи модулі HSM та технологію MACsec, а також впровадження багаторівневої автентифікації. Особливий акцент зроблено на використанні сучасних криптографічних протоколів TLS 1.3 з алгоритмами на основі еліптичних кривих.

Проблеми обробки даних пов'язані з перевантаженням edge-пристроїв (до 95% завантаження CPU) та неефективним використанням мережевих ресурсів. Для оптимізації запропоновано використання FPGA-прискорення, GPU-обчислень, а також адаптивних методів збору даних. Особливу увагу приділено переходу на бінарні протоколи передачі даних, що дозволяє значно скоротити обсяги мережевого трафіку.

Питання апаратної надійності вирішуються через заміну традиційних SD-карт на промислові eMMC-накопичувачі з технологією wear-leveling, а також впровадження комплексного захисту від ЕМІ-наведень. Останнє включає використання спеціальних екранованих кабелів, цифрових ізоляторів та, у критичних випадках, оптичних ліній зв'язку.

Запропоновані рішення мають ряд ключових переваг. Вони забезпечують комплексний підхід до вирішення проблем, враховують специфіку промислових умов експлуатації та пропонують конкретні технічні рішення з чіткими показниками ефективності. Важливою особливістю є можливість поетапного впровадження, що дозволяє мінімізувати вплив на роботу існуючих систем.

Очікувані результати впровадження включають: зниження часу простоїв на 40-60%, скорочення витрат на обслуговування на 25-35%, підвищення загальної продуктивності систем на 20-30%, а також зменшення кількості кіберінцидентів до 1-2 на рік. Додатково, запропоновані заходи дозволять збільшити термін служби апаратних компонентів у 3-5 разів.

Для моніторингу ефективності запропоновано систему КРІ, яка включатиме такі показники як коефіцієнт доступності, середній час між відмовами (MTBF) та рівень використання ресурсів. Це дозволить не лише оцінити ефективність впроваджених рішень, але й своєчасно виявляти потенційні проблеми.

4.2 Узагальнення моделей і методів управління проєктами, використаних у розробці IoT-системи

У процесі розробки IoT-системи для контролю якості виробництва музичних інструментів було застосовано низку сучасних моделей та методів управління проєктами. Вони дозволили ефективно планувати, реалізовувати та контролювати всі етапи життєвого циклу проєкту.

У процесі створення інноваційної IoT-системи для моніторингу якості виробництва музичних інструментів було розроблено унікальну методологію

управління проєктом, яка органічно поєднує переваги класичних та сучасних підходів. Цей інтегрований підхід дозволив не тільки ефективно керувати всіма фазами життєвого циклу проєкту, але й забезпечити високу гнучкість у відповідь на зміни технічних вимог, ринкових умов та викликів, що виникали під час реалізації.

Основним концептуальним рішенням стало створення гібридної методології, що інтегрувала структурованість водоспадної моделі (Waterfall) з адаптивністю гнучких методологій (Agile) [9] [13]. Waterfall-підхід забезпечив чітку послідовність ключових етапів: від глибокого аналізу вимог і ринкових потреб через архітектурне проектування, що враховувало специфічні промислові стандарти, до реалізації апаратно-програмного комплексу, багаторівневого тестування та промислового впровадження. Паралельно Agile-методологія (у форматі Scrum) дозволила реалізувати ітеративний підхід до розробки програмного забезпечення, організувавши процес у вигляді серії коротких спринтів тривалістю 2-3 тижні з чіткими цілями та критеріями успіху. Щоденні стендап-зустрічі, регулярні демонстрації результатів зацікавленим сторонам та постійний зворотній зв'язок значно підвищили ефективність команди та якість кінцевого продукту [20].

Особливу увагу приділили впровадженню DevOps-практик, які стали інтеграційною платформою між процесами розробки та експлуатації. Було створено потужні CI/CD-пайплайни на базі Jenkins та GitLab CI, що забезпечили автоматизацію юніт- та інтеграційного тестування, статичного аналізу коду (SAST), динамічного тестування безпеки (DAST) та автоматичного розгортання на тестових стендах. Контейнеризація за допомогою Docker дозволила ізолювати компоненти системи, стандартизувати середовища розробки та спростити процеси масштабування.

Для ефективного планування та контролю використовувався комплекс інструментів, серед яких особливе місце зайняли діаграми Ганта в MS Project, що забезпечили наочну візуалізацію залежностей між задачами, контроль критичних шляхів та моніторинг використання ресурсів. Метод критичного

шляху (CPM) дозволив ідентифікувати ключові етапи проєкту, такі як інтеграція датчиків із хмарною платформою, та оптимізувати тривалість робіт за рахунок раціонального розподілу трудових ресурсів.

Система управління ризиками ґрунтувалася на комплексному SWOT-аналізі, який виявив сильні сторони (високу точність вимірювань, унікальні алгоритми аналізу), слабкі сторони (залежність від специфічних компонентів), потенційні можливості (розширення на інші види виробництва) та загрози (кіберризика, конкурентний тиск). Детальна матриця ризиків з оцінкою ймовірності виникнення, потенційного впливу та стратегій мінімізації дозволила проактивно запобігати потенційним проблемам.

Управління якістю реалізовувалося через інтеграцію принципів TQM (Total Quality Management) із застосуванням циклу Демінга (PDCA) та методології Six Sigma з її DMAIC-підходом. Це забезпечило стандартизацію процесів, постійне вдосконалення та мінімізацію варіабельності параметрів якості.

Технологічний стек управління включав Jira для трекінгу задач та помилок, управління спринтами та інтеграції з інструментами розробки, а також Node-RED для швидкого прототипування, візуального програмування логіки та інтеграції з апаратними компонентами.

Реалізований підхід продемонстрував вражаючі результати: досягнуто 95% відповідності плановим показникам, час розробки скорочено на 20% порівняно з традиційними методами, забезпечено 99,9% доступність системи після впровадження, а кількість критичних дефектів зведено до мінімуму (0,1%). Ця методологія може бути успішно адаптована для розробки промислових IoT-рішень, створення систем предиктивної аналітики та впровадження технологій Industry 4.0 у різних галузях. Перспективи подальшого розвитку включають інтеграцію штучного інтелекту для прогнозування якості, впровадження блокчейн-технологій для відстеження ланцюга поставок та розширення функціоналу для роботи з AR/VR-інтерфейсами.

Комплексний аналіз методологій управління проектами IoT-системи для контролю якості виробництва музичних інструментів наведено у тпбл. 4.7.

Таблиця 4.7

Комплексний аналіз методологій управління проектами IoT-системи для контролю якості виробництва музичних інструментів

Аспект управління	Використані методи/інструменти	Ключові переваги	Отримані результати
1	2	3	4
Методологічний підхід	Гібридна модель (Waterfall + Agile/Scrum)	Поєднання структурованості та гнучкості, ітеративна розробка	Скорочення часу розробки на 20%, підвищення якості продукту
DevOps-практики	CI/CD пайплайни (Jenkins, GitLab CI), Docker	Автоматизація тестування, стандартизація середовищ	99.9% доступність системи, швидке розгортання
Інструменти планування	Діаграми Ганта (MS Project), CPM	Візуалізація залежностей, оптимізація ресурсів	Ефективне використання ресурсів, контроль термінів

Управління ризиками	SWOT-аналіз, матриця ризиків	Проактивне виявлення проблем, стратегії мінімізації	Зниження кількості критичних дефектів до 0.1%
Управління якістю	TQM (PDCA), Six Sigma (DMAIC)	Стандартизація процесів, постійне вдосконалення	95% відповідність плановим показникам якості
Технологічний стек	Jira, Node-RED	Комплексний трекінг задач, швидке прототипування	Ефективна координація команди, гнучкість розробки

Продовження таблиці 4.7

1	2	3	4
Перспективи розвитку	AI, блокчейн, AR/VR	Покращення прогнозування, відстеження, інтерфейсів	Потенціал для масштабування на інші галузі
Управління якістю	TQM (PDCA), Six Sigma (DMAIC)	Стандартизація процесів, постійне вдосконалення	95% відповідність плановим показникам якості

Реалізований інтегрований підхід до управління проектом довів свою високу ефективність на практиці. Ключовим успіхом стало вдале поєднання структурованості каскадної моделі Waterfall з гнучкістю Agile/Scrum, що дозволило з одного боку зберегти чітке стратегічне планування всіх етапів проекту, а з іншого - забезпечити необхідну адаптивність у процесі розробки програмного забезпечення. Особливо важливо відзначити, що такий гібридний підхід дозволив скоротити терміни розробки на 20% порівняно з традиційними методами, що є значним конкурентним перевагою в умовах стрімкої зміни технологічного ландшафту.

Впровадження сучасних DevOps-практик, включаючи автоматизовані CI/CD-пайплайни та контейнеризацію з використанням Docker, забезпечило безпрецедентну надійність системи (99,9% доступності) і значно прискорило процеси тестування та розгортання. Інструменти планування, такі як діаграми Ганта та метод критичного шляху, довели свою ефективність у оптимізації використання ресурсів і дотриманні встановлених термінів.

Комплексна система управління ризиками, заснована на SWOT-аналізі та детальних матрицях ризиків, дозволила мінімізувати кількість критичних дефектів до рекордно низького рівня (0,1%). Одночасно впровадження принципів TQM та методології Six Sigma забезпечило стабільно високу якість продукту (95% відповідності плановим показникам).

Використання сучасних інструментів управління (Jira для відстеження завдань та Node-RED для швидкого прототипування) значно підвищило ефективність роботи команди та прискорило процес розробки. Перспективні напрями розвитку, включаючи інтеграцію ШІ, блокчейн-технологій та AR/VR-інтерфейсів, відкривають нові можливості для масштабування даного підходу в інших галузях промисловості.

Отримані результати переконливо доводять, що розроблена методологія управління проектом є ефективним інструментом для створення складних IoT-систем в умовах високих вимог до якості та надійності. Успішна реалізація даного проекту створює основу для тиражування цього підходу в інших

сегментах промислового виробництва, що особливо актуально в контексті активного впровадження технологій Industry 4.0.

ВИСНОВКИ

У першому розділі досліджено особливості управління IoT-проектами у виробництві музичних інструментів, зосередившись на ключових вимогах до їх реалізації: локація, вартість, обсяг даних та масштабованість. Проаналізовано проблеми традиційних методів контролю якості, такі як суб'єктивність, затримки у виявленні дефектів та відсутність системного підходу. За допомогою SWOT-, STEP-аналізу та методу "5 портретів" визначено сильні сторони проекту (інноваційність, автоматизація) та потенційні ризики (конкуренція, висока вартість компонентів). Побудовано дерево цілей, що включає інтеграцію IoT-датчиків, оптимізацію виробництва та відповідність стандартам. Також ідентифіковано зацікавлені сторони, серед яких виробники, споживачі та регулятори. Отримані результати підтверджують доцільність впровадження IoT-системи для підвищення точності контролю якості та ефективності виробничих процесів.

У другому розділі дослідження було розроблено та впроваджено IoT-систему контролю якості для виробництва музичних інструментів. Основним етапом роботи стала розробка життєвого циклу проекту (ЖЦП), який охопив усі фази від аналізу вимог до завершення та підтримки системи. Ключові віхи проекту, такі як дослідження вимог, розробка апаратно-програмного комплексу, тестування та запуск, були чітко визначені, що забезпечило структурований підхід до реалізації. Важливим аспектом стало формування організаційної структури команди з чітким розподілом ролей. Участь Project Manager, IoT Engineer, Data Scientist, QA та інших фахівців дозволила ефективно керувати ресурсами, мінімізувати ризики та забезпечити відповідність якості вимогам. Концептуальна модель системи охопила ключові компоненти: датчики (температури, вологості, вібрації, акустики), edge-шлюзи для попередньої обробки даних, хмарну платформу (AWS IoT, Azure IoT) для аналітики та зберігання, а також інтерфейс користувача (Grafana, Power BI) для візуалізації та моніторингу. Ця модель забезпечила цілісність системи та її адаптацію до реальних виробничих умов. Математичне

моделювання спрямоване на оптимізацію витрат стало основою для розподілу датчиків. За допомогою цільової функції, що враховує точність датчиків, ймовірність виявлення браку та бюджетні обмеження, було запропоновано рішення на базі цілочисельного лінійного програмування (ILP). Це дозволило мінімізувати загальні витрати, поєднуючи ефективність використання ресурсів із високою точністю контролю.

Архітектура програмного забезпечення, побудована на гібридній edge-cloud моделі, забезпечила швидку обробку даних на периферії (Edge-обчислення) та глибокий аналіз у хмарі. Такий підхід зменшив затримки, підвищив надійність системи та дозволив ефективно використовувати ресурси для довгострокового зберігання даних.

Основні результати розділу включають визначення оптимального підходу до впровадження IoT-системи, розробку математичної моделі для розподілу датчиків, створення edge-cloud архітектури та формування ефективної організаційної структури команди. Ці рішення суттєво підвищують точність контролю якості, знижують витрати на брак та автоматизують виробничі процеси, що підтверджує доцільність інтеграції IoT-технологій у виробництво музичних інструментів.

У третьому розділі дослідження було реалізовано ключові етапи управління та координації розробки IoT-системи контролю якості для виробництва музичних інструментів. Планування проекту розпочалося з вибору технічних компонентів, включаючи сенсори DHT11 для вимірювання температури та вологості, мікроконтролер Arduino Uno як edge-пристрій та модуль ESP8266 для бездротової передачі даних. Архітектура системи базувалася на інтеграції апаратних рішень з програмними інструментами, такими як Node-RED для візуального програмування логіки обробки даних та протокол MQTT для ефективної комунікації між пристроями. Було розроблено базу даних із структурованими таблицями для зберігання показників сенсорів та їх локацій, що забезпечило систематизацію інформації та легкий доступ для аналітики.

Важливим етапом стало тестування функціоналу, де перевірялася коректність роботи датчиків, передача даних через MQTT-брокер, їх обробка в Node-RED та візуалізація на дашбордах. Інтеграція компонентів підтвердила стабільність системи в реальних умовах, зокрема здатність виявляти аномалії та генерувати сповіщення. Управління ризиками включало ідентифікацію технічних, фінансових та часових загроз, а також розробку стратегій їх мінімізації, таких як резервний бюджет, альтернативні постачальники та шифрування даних. Використання інструментів Jira та MS Project дозволило ефективно планувати завдання, відстежувати прогрес та координувати роботу команди, що включала проектних менеджерів, IoT-інженерів, розробників ПЗ та QA-фахівців.

Формування бюджету враховувало витрати на апаратне забезпечення, ліцензійне ПЗ, зарплати та резервні кошти, що забезпечило прозорість фінансування. Ресурсна мапа проекту визначила ключові ролі, технічні та інфраструктурні потреби, що сприяло оптимальному розподілу завдань. Підсумком розділу стало створення функціонального прототипу IoT-системи, яка автоматизує контроль якості, знижує витрати на брак та забезпечує відповідність стандартам виробництва, підтверджуючи практичну цінність запропонованих рішень.

Висновок із розділу чотири підкреслює, що комплексне застосування сучасних моделей і методів управління проектами, таких як гібридна модель Waterfall та Agile/Scrum, а також використання DevOps-практик, інструментів планування і систем управління ризиками, забезпечують високий рівень ефективності та гнучкості у реалізації IoT-систем для контролю якості виробництва музичних інструментів. Застосування такої інтегрованої методології сприяло значному скороченню часу розробки (на 20%), підвищенню якості кінцевого продукту, забезпеченню стабільної роботи системи з показником доступності понад 99,9%, а також зниженню кількості критичних дефектів до рекордно низького рівня (0,1%). Крім того, використання автоматизованих CI/CD-процесів, контейнеризації, систем

трекінгу задач і прототипування сприяло підвищенню командної ефективності, зменшенню часу на тестування і розгортання, а також забезпеченню високої надійності та безперервності роботи системи.

Важливою перевагою є також застосування сучасних інструментів управління, таких як Jira і Node-RED, що дозволили швидко адаптуватися до змінних вимог і забезпечити гнучкість у розробці.

Крім того, застосовані підходи показали свою перспективність у масштабуванні та адаптації до інших галузей промисловості, що особливо актуально в контексті активного впровадження концепцій Industry 4.0. Планування та реалізація проекту з урахуванням сучасних тенденцій розвитку технологій (інтеграція штучного інтелекту, блокчейн, AR/VR) відкривають широкі можливості для створення інноваційних, високоефективних і масштабованих систем автоматизації та контролю якості у різних сферах промислового виробництва.

Отже, застосована управлінська стратегія дозволила не лише досягти високих технічних і організаційних показників, але й створити надійну основу для подальшого розвитку, удосконалення та масштабування IoT-рішень, що відповідають сучасним вимогам до якості, безпеки і ефективності у контексті цифрової трансформації промислових процесів. Це підтверджує, що обрана методологія є ефективним інструментом управління складними технічними проектами і має високий потенціал для впровадження у ширшому промисловому масштабі.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ:

1. Loopmasters. Music and The Internet of Things
Режим доступу: <https://www.loopmasters.com/articles/3500-Music-and-The-Internet-of-Things>
2. Маркетинговий аналіз - Режим доступу:
<https://livingfo.com/marketynhovyj-analiz/>
3. Основні принципи та етапи управління ІТ-проектами - Режим доступу:
<https://www.machinemetrics.com/blog/iot-in-manufacturing>
4. Чим відрізняється PEST та STEP аналіз: порівняння та особливості -
Режим доступу: <https://www.kaaiot.com/blog/connect-devices-to-the-iot-platform-with-node-red>
5. Які складові STEP-аналізу - Режим доступу: <http://rua.pp.ua/kakovy-i-sostavlyayuschie-step-25792.html>
6. 5 сил Портера (Porter's Five Forces) - Режим доступу:
<https://www.maxzosim.com/porters-five-forces/>
7. Маркетинговий аналіз: ключ до економії і успіху у бізнесі -
Режим доступу: <https://cases.media/en/article/marketingovii-analiz-klyuch-do-ekonomiyi-i-uspikhu-u-biznesi>
8. Методологія Kanban: дошки, принципи та сервіси, які вам допоможуть -
Режим Доступу: <https://uaspectr.com/2021/01/26/shho-take-kanban/>
9. AIESEC Ukraine - Project Management Guide - Режим доступу:
<https://www.scribd.com/document/148339280/AIESEC-Ukraine-Project-Management-Guide>
10. Етапи в управлінні проектами. Крок за кроком досягайте цілей проекту.
- Режим доступу: <https://flexi-project.com/uk/етапи-в-управлінні-проектами/>
11. Project Management in the Military Field: - Режим доступу
<https://monograph.com.ua/pctc/catalog/book/978-617-8360-08-5>

12. Management in Times of War: - Режим доступу
https://en.wikipedia.org/wiki/Management_in_Times_of_War
13. Traversing Hybrid Project Management: The Bridge Between Agile and Waterfall - Режим доступу: <https://www.toptal.com/project-managers/agile/hybrid-project-management-a-middle-ground-between-agile-and-waterfall>
14. Codeguida. Django - Режим доступу: <https://codeguida.com/post/1039>
15. Djangoproject - Режим доступу: <https://www.djangoproject.com>
16. Metanit. - Режим доступу: <https://metanit.com>
17. Планування проекту та методи управління - Режим доступу:
<https://moodle.znu.edu.ua/enrol/index.php?id=9839> Project Management
18. PMBOK Guide: - Режим доступу: <https://www.pmi.org/standards/pmbok>
19. Scrum Guide: - Режим доступу: <https://scrumguides.org/>
20. Драйвера для NODE MCU - Режим доступу:
[URL:https://github.com/adafruit/DHT-sensor-library](https://github.com/adafruit/DHT-sensor-library)
21. Драйвер для Node MCU: - Режим доступу:
<https://github.com/esp8266/Arduino>
22. Randomnerdtutorials. Робота з DHT DHT 22 - Режим доступу:
<https://randomnerdtutorials.com/complete-guide-for-dht11-dht22-humidity-and-temperature-sensor-with-arduino>
23. Налаштування DHT 11 та 22 Lastminuteengineers - Режим доступу:
<https://lastminuteengineers.com/dht11-dht22-arduino-tutorial>
24. Принципи роботи Arduino IDE – Режим доступу:
<https://bitkit.com.ua/ru/shho-take-arduino>
25. Програмування Arduino – Режим доступу <https://doc.arduino.ua/ru/prog/>
26. Програмування Arduino – Режим доступу:
<https://doc.arduino.ua/ru/prog/Void>
27. Датчики DHT 22 vs 11 - Режим доступу: <https://arduino-diy.com/arduino-datchiki-temperatury-i-vlazhnosti-DHT11-i-DHT22>

- 28.Порівняння DHT 22 vs 11 - Режим доступу:
<https://youtu.be/LChGRzXnJko>
- 29.Барометр BMP 280 - Режим доступу: <https://youtu.be/QlAFzvZbGnU>
- 30.Робота з Python - Режим доступу: <https://habr.com/ru/hub/python/>
- 31.Компілятор Python – Режим доступу: <http://primat.org/index/0-144>
- 32.Встановлення VS Code – Режим доступу: <https://code.visualstudio.com/>
- 33.Встановлення Vs code - Режим доступу: <https://youtu.be/M-WeTRvLRtc>
- 34.Стартовий урок по HTML CSS – Режим доступу:
<https://www.youtube.com/live/h55Fv-DAw64?feature=share>
- 35.NODE RED – Режим доступу:<https://nodered.org/>
- 36.Робота з NODE RED – Режим доступу:
https://www.youtube.com/watch?v=0zWIr_F6IEk
- 37.Робота з NODE RED – Режим доступу:
<https://pupenasan.github.io/NodeREDGuidUKR/>
- 38.Node RED Corezoid: - Режим доступу:
<https://www.42flows.tech/ua/blog/node-red-and-corezoid/>

ДОДАТКИ

ДОДАТОК А

Лістинг Коду

Models.py

```
from django.db import models
```

```
class Sensor(models.Model):
```

```
    sensor_id = models.IntegerField()
```

```
    sensor_location = models.CharField(max_length=20)
```

```
    sensor_type = models.CharField(max_length=20)
```

```
    calibration_date = models.DateField()
```

```
    def __str__(self):
```

```
class Reading(models.Model):
```

```
    # reading_sens = models.ForeignKey(Sensor, on_delete=models.CASCADE)
```

```
    reading_id = models.IntegerField()
```

```
    sensor_id = models.IntegerField()
```

```
    # time_stamp = models.IntegerField()
```

```
    humidity = models.FloatField()
```

```
    temperature = models.FloatField()
```

```
    def __str__(self):
```

```
        return f'{self.reading_id}'
```

Forms.py

```
from django import forms
```

```
from . models import Sensor, Reading
```

```
class SensorForm(forms.ModelForm):
```

```
    calibration_date = forms.DateTimeField(  
        widget=forms.DateTimeInput(attrs={'type': 'datetime-local'}))
```

```
    class Meta:
```

```
        model = Sensor  
        fields = '__all__'
```

```
class ReadingForm(forms.ModelForm):
```

```
    class Meta:
```

```
        model = Reading  
        fields = '__all__'
```

```
Admin.py
```

```
from django.contrib import admin
```

```
urls.py
```

```
from django.urls import path
```

```
from . import views
```

```
app_name = 'agroholding'
```

```
urlpatterns = [  
    path('object-list/', views.object_list, name='object-list'),
```

```
    path('create/<str:what>/', views.CreateView.as_view(), name='create'),  
]
```

create-reading.html

```
{% extends 'layout/form-base.html' %}
```

```
{% block form-action %}{% url 'agroholding:create' what='reading' %}{%  
endblock %}
```

object-list.html

```
{% extends 'layout/base.html' %}
```

```
{% block content %}
```

```
<nav align="right" class="mb-3">
```

```
    {% with 'sensor reading' as object_list %}
```

```
    {% for object in object_list.split %}
```

```
        <a class="btn btn-secondary" href="{% url 'agroholding:create' what=object  
%}">Create {{ object }}</a>
```

```
    {% endfor %}
```

```
    {% endwith %}
```

```
</nav>
```

```
<div class="row">
```

```
    <h2 class="col">Sensor</h2>
```

```
    <h2 class="col">Reading</h2>
```

```
</div>
```

```
<div class="row">
```

```
    <ul class="col">
```

```

{% for sensor in sensors %}
<li>
  <p><strong>Sensor id</strong>: {{ sensor.sensor_id }}</p>
  <p><strong>Sensor location</strong>: {{ sensor.sensor_location }}</p>
  <p><strong>Sensor type</strong>: {{ sensor.sensor_type }}</p>
  <p><strong>Calibration date </strong>: {{ sensor.calibration_date }}</p>

</li>
{% endfor %}
</ul>
<ul class="col">
  {% for reading in readings %}
<li>
  <p><strong>Reading ID</strong>: {{ reading.reading_id }}</p>
  <p><strong>Sensor Id</strong>: {{ reading.sensor_id }}</p>
  <p><strong>timestamp</strong>: {{ reading.timestamp }}</p>
  <p><strong>Humidity</strong>: {{ reading.humidity }}</p>
  <p><strong>Temperatre</strong>: {{ reading.temperature }}</p>
</li>
{% endfor %}
</ul>

</div>

{% endblock %}

```

Initials.py

Generated by Django 4.2.1 on 2023-05-11 18:19

```
from django.db import migrations, models
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
const char* ssid = "BIGGNOM";
```

```
const char* password = "D3ekun0V";
```

```
ESP8266WebServer server(80);
```

```
void handleRoot() {
```

```
    server.send(200, "text/plain", "hello from esp8266!");
```

```
}
```

```
void handleNotFound() {
```

```
    String message = "File Not Found\n\n";
```

```
    message += "URI: ";
```

```
    message += server.uri();
```

```
    message += "\nMethod: ";
```

