

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

КУПЧИН Юрій Юрійович

**Система передачі сенсорних даних від фізичного двійника до цифрового
/ System for transferring sensor data from a physical twin to a digital one**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-42
Ю. Ю. Купчин

Науковий керівник:
к.т.н., професор В.В. Кочан

Дипломний проект допущено до
захисту

« ___ » _____ 2023 р.

Завідувач кафедри
_____ М.П. Комар

Тернопіль – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
« ____ » _____ 2022 р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ
КУПЧИН Юрій Юрійович
(прізвище, ім'я, по батькові)

1. Тема проекту: Система передачі сенсорних даних від фізичного двійника до цифрового / System for transferring sensor data from a physical twin to a digital one

керівник проекту к.т.н., проф. В.В. Кочан

затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- опис цифрового двійника;
- опис передумов виникнення;
- аналіз існуючих рішень цифрових двійників;
- формування вимог до системи передачі сенсорних даних від фізичного двійника до цифрового;
- опис загальної структури проектованої системи
- опис алгоритму моделювання цифрового двійника
- опис комунікацій цифрового двійника та комунікаційні потреби;
- вибір та аналіз апаратного та програмного забезпечення системи;
- розробка апаратного та програмне забезпечення системи;
- розробка програмного забезпечення апаратної частини та інтерфейсу цифрового двійника;

– проведення тестування системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- алгоритм моделювання цифрового двійника;
- Комунікація між системою та іншими об'єктами.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., професор В.В. Кочан		

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Передумови виникнення та концепція цифрових двійників	30.12.2022	
2	Алгоритмічне та інформаційне забезпечення системи	24.03.2023	
3	Програмно-технологічне забезпечення системи	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент _____ Ю. Ю. Купчин
(підпис)

Керівник проекту _____ В.В. Кочан
(підпис)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 69 с., 20 рис., 3 додатки, 52 джерела.

Метою дипломного проекту є розробка системи передачі сенсорних даних від фізичного двійника до цифрового. Основне призначення даної системи – скорочення розриву між концепцією та реалізацією цифрового двійника, забезпечуючи передачу даних з датчиків. Система дозволить додавати датчики до фізичного двійника та надавати інтерфейс для їх конфігурації віддалено через Інтернет.

В роботі використовувалися методи системного аналізу та синтезу, порівняння, логічного узагальнення результатів, методи інтелектуального аналізу даних.

Розроблено загальну структуру системи, алгоритм моделювання цифрового двійника, визначено комунікаційні потреби цифрового двійника, вибрано апаратне забезпечення системи, вибрано та проаналізовано програмне забезпечення для системи.

Розроблено програмне забезпечення апаратної частини системи та «серверної» частини системи.

Ключові слова: ЦИФРОВИЙ ДВІЙНИК, МІКРОКОМП'ЮТЕР, АЛГОРИТМ МОДЕЛЮВАННЯ, MQTT, ІОТ, RASPBERRY PI, PYCOM WIPY, REST API.

ABSTRACT

The bachelor's thesis report: 69 pages, 20 figures, 3 appendices, 52 sources.

The aim of the thesis project is to develop a system for transferring sensor data from a physical twin to a digital twin. The main purpose of this system is to bridge the gap between the concept and realization of a digital twin by providing data transfer from sensors. The system will allow adding sensors to the physical twin and provide an interface for their configuration remotely via the Internet.

The methods of system analysis and synthesis, comparison, logical generalization of results, and data mining techniques were used in the study.

The general structure of the system, the algorithm for modeling the digital twin, the communication needs of the digital twin, the system hardware, and the software for the system were developed, selected and analyzed.

The software for the hardware part of the system and the "server" part of the system has been developed.

Keywords: DIGITAL TWIN, MICROCOMPUTER, MODELING ALGORITHM, MQTT, IOT, RASPBERRY PI, PYCOM WIPY, REST API.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Система передачі сенсорних даних від фізичного двійника до цифрового.

1.2 Область застосування – обробка даних та аналіз даних від фізичних об'єктів.

2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломного проекту є розробка системи передачі сенсорних даних від фізичного двійника до цифрового.

4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

- збір даних;
- попередня обробка даних для побудови цифрового двійника;
- забезпечення зв'язку між фізичним двійником і цифровим двійником, а також між цифровим двійником і зовнішнім світом;
- Можливість змінювати налаштування сенсорних вузлів, включаючи

налаштування вимірювань та передачі даних, віддалено через Інтернет.

5.2 Вимоги до апаратних засобів:

- Raspberry PI, Pycom WiPy;
- ADXL345;
- LIS3DSH
- Seeed Grove I2C ADC

5.3 Вимоги до програмних засобів:

- MicroPython;
- Django REST framework
- Adafruit IO

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ Ю. Ю. Купчин _____
(підпис) (прізвище та ініціали)

Керівник дипломного проекту _____ В.В. Кочан _____
(підпис) (прізвище та ініціали)

ЗМІСТ

Вступ.....	9
1 Передумови виникнення та концепція цифрових двійників	10
1.1 Цифровий двійник	10
1.2 Аналіз існуючих рішень цифрових двійників.....	14
1.3 Формування вимог до системи передачі сенсорних даних від фізичного двійника до цифрового	18
2 Алгоритмічне та інформаційне забезпечення системи	22
2.1 Загальна структура проектованої системи	22
2.2 Алгоритм моделювання цифрового двійника.....	25
2.3 Комунікація цифрового двійника та комунікаційні потреби	28
3 Програмно-технологічне забезпечення системи	36
3.1 Апаратне та програмне забезпечення системи	36
3.2 Розробка програмного забезпечення апаратної частини та інтерфейсу цифрового двійника	40
3.3 Тестування системи	44
Додаток А Micropython код згенерований системою.....	65
Додаток Б Алгоритм моделювання цифрового двійника	68
Додаток В Комунікація між системою та іншими об'єктами.....	69

					<i>ДП.КН.9500038.069.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Купчин Ю.Ю..</i>			<i>Система передачі сенсорних даних від фізичного двійника до цифрового</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Кочан В.В.</i>				8	69	
<i>Реценз.</i>						<i>ЗУНУ.ФКІТ.КН-42</i>		
<i>Н. Контр.</i>		<i>Кочан В.В.</i>						
<i>Затверд.</i>		<i>Комар М.П.</i>						

ВСТУП

Актуальність теми дослідження. Цифровий двійник - це цифровий аналог фізичного об'єкта, який точно відображає поточний стан відповідного фізичного двійника. Він пропонує численні переваги протягом усього життєвого циклу продукту. Тому концепція цифрового двійника останнім часом привертає багато уваги. Крім того, її розробляють найбільші світові компанії, такі як Siemens, Bosh, Boeing.

Незважаючи на те, що концепція цифрових двійників постійно розвивається, бракує їй реальних реалізацій. Це пов'язано з тим, що цифровий двійник - це складна система систем з численними функціональними можливостями.

Метою дипломного проекту є розробка системи передачі сенсорних даних від фізичного двійника до цифрового. Основне призначення даної системи – скорочення розриву між концепцією та реалізацією цифрового двійника, забезпечуючи передачу даних з датчиків. Система дозволить додавати датчики до фізичного двійника та надавати інтерфейс для їх конфігурації віддалено через Інтернет.

Об'єктом дослідження системи цифрових двійників для реальних фізичних об'єктів

Предметом дослідження процес передачі даних від фізичного об'єкту до його цифрового двійника..

Методи та технології розробки. Технологією розробки було обрано REST API, MicroPython, сервер даних з підтримкою Django та Python Web framework. Для реалізації MQTT-комунікації та візуалізації - хмарний сервіс Adafruit IO.

Практичне значення одержаних результатів. В результаті проведеної роботи буде реалізовано: архітектуру системи передачі сенсорних даних від фізичного двійника до цифрового, програмне забезпечення для мікроконтролерів та сервера.

					ДП.КН.9500038.069.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ПЕРЕДУМОВИ ВИНЕКНЕННЯ ТА КОНЦЕПЦІЯ ЦИФРОВИХ ДВІЙНИКІВ

1.1 Цифровий двійник

Цифровий двійник (ЦД) можна визначити, як (фізичні та/або віртуальні) машини або комп'ютерні моделі, які імітують, емулюють, віддзеркалюють життя фізичної сутності, якою може бути об'єкт, процес, людина або властивість, пов'язана з людиною. Кожен ДТ пов'язаний зі своїм фізичним двійником за допомогою унікального ключа [1], [2], який ідентифікує фізичного двійника, а отже, дозволяє встановити зв'язок між ЦД та його двійником. ЦД - це більше, ніж проста модель або симуляція [3], [4], [5], [6]. ЦД - це жива, інтелектуальна модель, що розвивається, яка є віртуальним аналогом фізичного об'єкта або процесу. Вона повторює життєвий цикл свого фізичного двійника для моніторингу, контролю та оптимізації його процесів і функцій. Вона постійно прогнозує майбутні стани (наприклад, дефекти, пошкодження, збої) і дозволяє моделювати і тестувати нові конфігурації з метою превентивного застосування операцій з технічного обслуговування.

Більш конкретно, процес відтворення забезпечується безперервною взаємодією, зв'язком і синхронізацією (оптимізацією в замкнутому циклі) між ЦД, його фізичним двійником і зовнішнім, навколишнім середовищем.

Описові дані постійно обмінюються та оновлюються завдяки (сьогодні доступним) можливостям завантаження даних у реальному часі та зберігання великих обсягів даних. Завдяки оновленню в реальному часі від свого фізичного двійника і від цифрових двійників у навколишньому середовищі, цифровий двійник завжди в курсі того, що відбувається у фізичному світі.

Застосовуючи алгоритми злиття даних з подальшою аналітикою великих даних та описовими алгоритмами ІІІ [7], він розвивається разом зі своїм фізичним двійником завдяки модульній та високопараметризованій архітектурі [8], що дозволяє швидко реконфігуруватися.

										Арк.
										10
Змн.	Арк.	№ докум.	Підпис	Дата						

ДП.КН.9500038.069.ПЗ

Потім ЦД постійно синхронізується зі своїм фізичним двійником і змінюється разом з ним, причому ці зміни відображають властивості фізичного об'єкта, що відображається, і керуються ними [9]. На додаток до цієї комплексної емуляції, ЦД, оснащений штучним інтелектом, здатний розкривати інформацію, включаючи опис системи, приховані закономірності та невідомі кореляції.

Здатність реєструвати, контролювати і відстежувати стан і зміни фізичної системи дозволяє застосовувати ШІ для прогнозування збоїв, тестування результатів можливих рішень і активації механізмів самовідновлення. Це призводить до так званого предиктивного підходу до технічного обслуговування, коли збої прогножуються, а виправлення та/або модифікації можуть бути змодельовані, щоб уникнути помилок або знайти найкращі рішення. Така модифікація системи в режимі реального часу призводить до значних покращень, наприклад, у виробничій сфері, а також у медичній інженерії, оскільки можливість прогнозування дозволяє планувати ремонт або технічне обслуговування, таким чином запобігаючи перебоєм і потенційно дорогим поломкам. DT дозволяє всім користувачам і зацікавленим сторонам мати доступ до фізичного стану двійників і контролювати його, незалежно від того, де вони фізично знаходяться. Це сприяє покращенню та прискоренню співпраці.

1.1.1 Передумови

Термін "цифровий двійник" є досить новим і вперше був представлений громадськості в 2010 [10, 11]. Однак дуже схожі ідеї щодо цифрового двійника вже були розроблені на початку 2000-х років Ці концепції мають деякі спільні риси з сучасним баченням цифрового двійника, але їх ще не можна назвати цифровими двійниками.

Автори в роботі [12] представляють "Холони", які є автономними агентами для холонічної виробничої системи (HMS). Холони можуть

										ДП.КН.9500038.069.ПЗ	Арк.
											11
Змн.	Арк.	№ докум.	Підпис	Дата							

співпрацювати для досягнення спільної мети, а також реагувати на збурення та оптимізувати процес. HMS складається з трьох основних частин:

- 1) холон продукту, який містить інформацію про сам продукт, наприклад, актуальну інформацію про життєвий цикл продукту і специфікацію матеріалів,
- 2) холон ресурсів, який використовується для розподілу ресурсів і складається з фізичної частини (ресурсу) та інформаційно-обробної частини,
- 3) холон замовлення, який є "активною сутністю, відповідальною за виконання роботи правильно і вчасно".

Інші дослідники. вводять поняття інтелектуального продукту і досліджують його вплив на життєвий цикл продукту. Вони визначають інтелектуальний продукт як такий, що має принаймні деякі з наступних характеристик:

- 1) Унікальна ідентичність
- 2) Здатність спілкуватися з навколишнім середовищем
- 3) Може зберігати або зберігати дані про себе
- 4) Може виражати свої особливості, виробничі вимоги тощо
- 5) Здатність брати участь у прийнятті рішень, що стосуються її власної долі, або приймати їх.

Програмні агенти дозволяють інтелектуальним продуктам реалізовувати вищезгадані функції 4 і 5. Програмний агент визначається, як "окремий програмний процес, який може міркувати незалежно і може реагувати на зміни, спричинені іншими агентами та його оточенням, а також здатний співпрацювати з іншими агентами". На рисунку 1.1 продукт який має мітку. Мітка використовується для зв'язку продукту через локальну або віддалену мережу з інформацією про нього самого, а також про його програмного агента (Product Avatar) [13].

Кожен продукт має цифровий аналог, який називається Аватар, у віртуальній реальності. Аватар здатний до автономного прийняття рішень і сам є індивідуальним об'єктом. Для управління інформацією, пов'язаною з

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

продуктом, в якому сам продукт управляє і виступає в ролі сполучної ланки між інформацією, що стосується його самого.

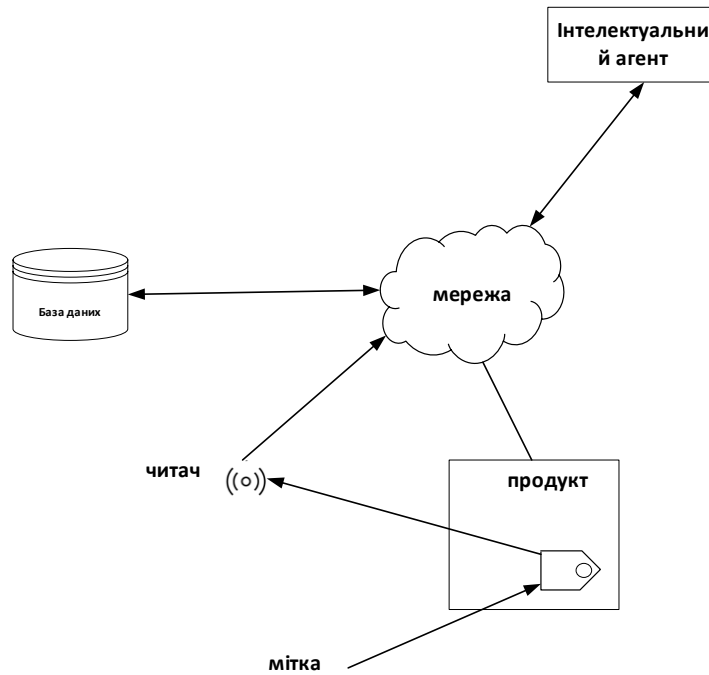


Рисунок 1.1 – Цифровий аватар

У традиційному підході (рисунок 1.2b) інформація зберігається окремими сторонами і тому не є легкодоступною. Доступ до даних, зібраних протягом життєвого циклу продукту, дозволяє оптимізувати, наприклад, експлуатацію, технічне обслуговування і ремонт.

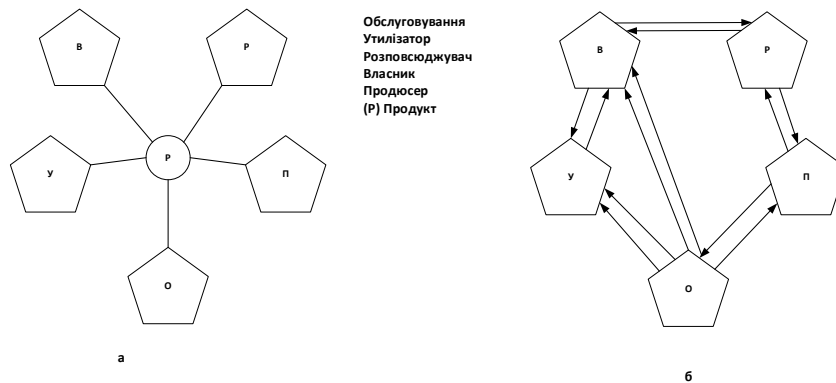


Рисунок 1.2 - Взаємовідносини між учасниками життєвого циклу продукту при продуктово-орієнтованому підході (а) та традиційному підході (б)

Також деякі автори пропонували архітектуру на основі агентів для управління інформацією про продукт протягом усього його життєвого циклу. Кожен продукт має відповідний "віртуальний аналог", що є автономним програмним компонентом, здатним взаємодіяти з іншими агентами (і, можливо, людьми), реагувати на зміни в навколишньому середовищі та діяти для досягнення певної мети. До агента продукту можна отримати доступ через Інтернет, і це робить інформацію про продукт доступною.

Поняття кіберфізичної системи (КФС) [14] тісно пов'язане з цифровим двійником, оскільки цифровий двійник можна розглядати як "кібернетичну частину кіберфізичної системи". У КФС дані з датчиків збираються і аналізуються для управління фізичним процесом. Кібер-фізичні системи можна розглядати як ще один шлях, що веде до розвитку цифрових двійників, оскільки КФС можуть використовувати цифрових двійників для обробки даних з датчиків і управління фізичною системою.

1.2 Аналіз існуючих рішень цифрових двійників

Перше і, мабуть, найвідоміше визначення цифрового двійника дано в [10 та 11] у дорожній карті НАСА. Там сказано: "Цифровий двійник - це інтегрована мультифізична, мультимасштабна, імовірнісна симуляція транспортного засобу або системи, яка використовує найкращі доступні фізичні моделі, оновлення датчиків, історію даних тощо, щоб відобразити життя свого відповідного двійника". Крім того, цифровий двійник інтегрує дані датчиків, історію технічного обслуговування.

Цифровий двійник містить практично всі доступні дані з усього життєвого циклу фізичного двійника - від виробничих аномалій до експлуатаційних даних. Крім того, що цифровий двійник може точно відображати поточний стан свого фізичного двійника, він також може використовуватися для моделювання для прогнозування майбутніх станів відповідного фізичного двійника.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

В літературі [15] наводять наступні застосування для цифрового двійника:

1) Попередній аналіз. Це дає змогу дослідити наслідки зміни параметрів місії та стратегій для пом'якшення наслідків несподіваних збоїв під час польоту. Цифровий двійник також може передбачити ймовірність позитивного результату.

2) Дзеркальне відображення стану фізичного двійника під час роботи, що дозволяє безперервно прогнозувати майбутні стани фізичного двійника шляхом запуску симуляцій.

3) Аналіз причин аномалій під час роботи.

4) Прогнозування наслідків зміни параметрів об'єкта під час роботи. Це може бути використано для прийняття найбільш обґрунтованого рішення, якщо є необхідність змінити параметри, наприклад, в результаті відмови однієї системи.

5) «Сертифікація» шляхом моделювання"

Автори [16] також розглядають цифрових двійників з точки зору авіації. Цифровий двійник описується як "ультрареалістичний в геометричних деталях, включаючи виробничі аномалії, і в матеріальних деталях, включаючи статистичний рівень мікроструктури".

Цифровий двійник здатний діяти як віртуальний датчик, інтерполюючи дані, отримані від реальних датчиків. Наразі існує окрема модель для кожного типу фізики, наприклад, модель обчислювальної гідродинаміки (CFD), модель структурної динаміки (SDM) і термодинамічна модель для прогнозування терміну служби конструкції об'єкта. За допомогою цифрового двійника ці моделі можуть бути інтегровані. Маючи цифрового двійника, що містить всю інформацію, пов'язану з конкретним ідентифікатором об'єкта, в тому числі оцінки надійності всіх основних структурних компонентів, можна оптимізувати технічне обслуговування.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

В іншій публікації [17] (рисунок 1.3) цифровий двійник представлений, як надреалістична комп'ютерна модель побудованого літака від етапу збірки коомпонентів до списання.



Рисунок 1.3 - Елементи цифрового двійника планера літака

Цифровий двійник складається з декількох інтегрованих субмоделей. Ці субмоделі використовують найкращі доступні фізичні моделі, обмінюються інформацією одна з одною та оновлюються протягом життєвого циклу літака, щоб підвищити точність моделей. Цифровий двійник літака дозволяє віртуально тестувати конструкцію, контролювати стан і прогнозувати потребу в технічному обслуговуванні.

Автори [18] підкреслюють, що цифровий двійник представляє збудовану версію транспортного засобу або системи і включає інформацію на рівні мікроструктури матеріалу. Ця інформація про фізичну структуру в масштабі від "менше мікрона до метрів" використовується для створення "надточних фізичних моделей для прогнозування майбутніх станів транспортного засобу".

Цифровий двійник може ідеально відображати стан свого фізичного двійника і використовувати симуляції з надвисокою точністю для прогнозування можливих майбутніх станів фізичного двійника. В роботі [19] стверджується, що за допомогою цифрового двійника можна відмовитися від емпіричних та евристичних правил проектування, які призводять до створення важких конструкцій, а також невизначеності, пов'язаної з фактичною надійністю конструкції. Крім того, автори в [20] стверджують, що цифровий

двійник зробить революцію в сертифікації, оскільки об'єкт можна буде випробувати віртуально.

Якщо дивитись на більш прості речі то в якості цифрового двійника може виступати не тільки великий об'єкт наприклад, літак, чи автомобіль, де реалізовано багато систем збору даних, математичних моделей а, й простих речей таких як потенціометр. Наприклад в роботі [21] (рисунок 1.4)

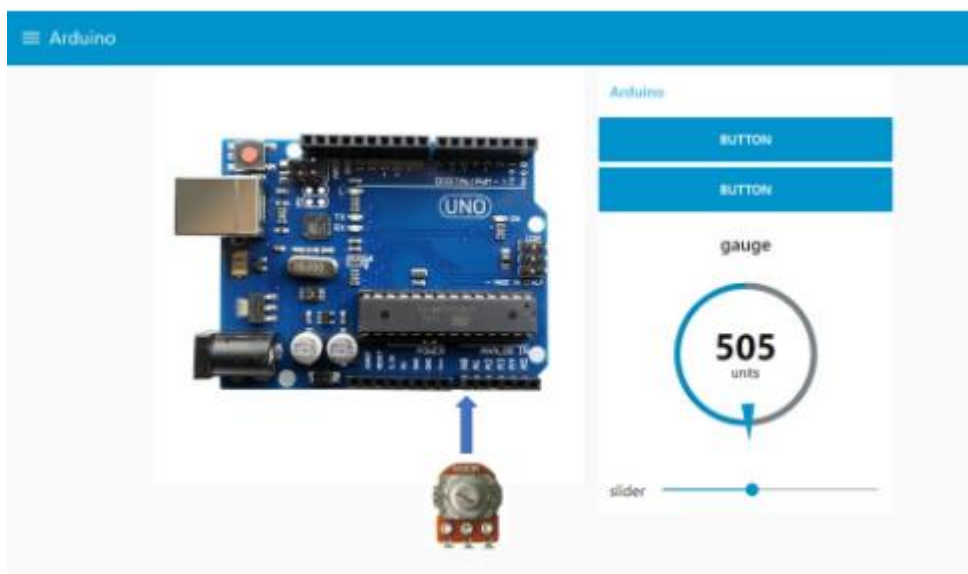


Рисунок 1.4 – Цифровий двійник на базі Arduino+NodeRed

За допомогою інструменту Node-RED [22] було розроблено візуалізацію цифрового двійника на основі потоків, який дає змогу розробляти різні архітектури IoT. Метод розробки мереж IoT заснований на обміні даними між вузлами за допомогою корисного навантаження повідомлень. Ці корисні навантаження налаштовуються для цифрових, аналогових і строкових типів даних. Вузли з'єднані за допомогою дротів, які передають дані в мережі додатків IoT.

Якщо потрібні математичні або умовні операції, функціональні вузли також можуть бути створені в Node-RED (рисунок 1.5).

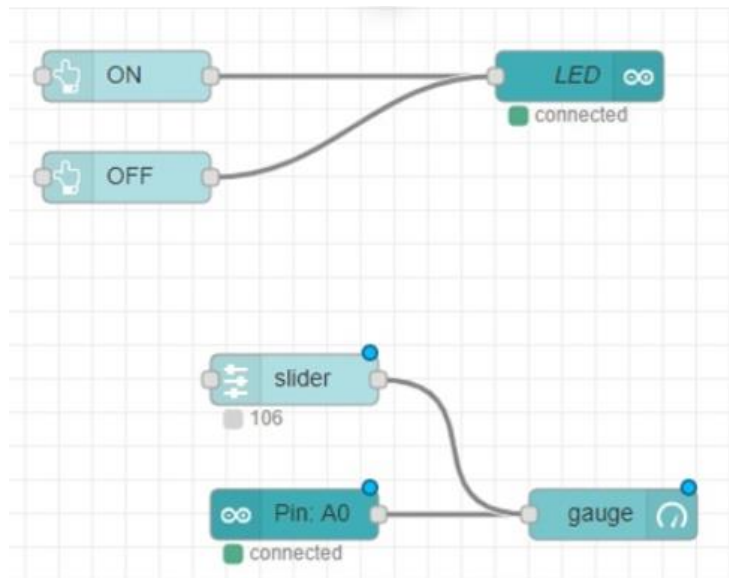


Рисунок 1.5 - Функціональні вузли Node-RED

Кожен вузол має детальну інформацію, яка пояснює його особливості та набір функцій. Крім того, можна додати вузли налагодження для перевірки правильності корисного навантаження повідомлень.

Після створення конкретного додатка IoT блок-схеми готові до розгортання. При створенні конкретного додатка IoT цифровий двійник може бути створений з використанням різних вузлів панелі управління. Ці вузли можуть відображати отримані дані в реальному часі від фізичного прототипу. Електронні датчики допомагають отримувати дані в режимі реального часу від вузла фізичного прототипу. Ці електронні датчики підключені до Arduino.

Raspberry Pi забезпечує дротову інфраструктуру для фізичного прототипу, що передає аналогові або цифрові дані назад цифровому двійнику. Arduino передає дані від підключеного датчика через послідовне з'єднання на основі USB до Raspberry Pi. Додаток Node RED - це один зі встановлених програмних пакетів на Raspberry Pi.

1.3 Формування вимог до системи передачі сенсорних даних від фізичного двійника до цифрового

Основною вимогою до системи - це дозволити керувати сенсорними вузлами віддалено через Інтернет і, таким чином, дозволити збір даних в контексті цифрового двійника.

Керування включає зміну налаштувань вимірювань, таких як частота дискретизації або чутливість, а також додавання/видалення датчиків.

Система повинна використовувати найбільш підходящі протоколи прикладного рівня та комунікаційні технології для передачі даних датчиків з фізичного двійника на цифровий двійник, щоб зробити систему придатною для різних випадків використання. З тієї ж причини вона має підтримувати кілька типів датчиків

Розроблювана система має фокусуватись лише на управлінні сенсорними вузлами. Це дозволяє зробити платформу легкою та простою.

Підтримувати необхідний інтерфейс, за допомогою якого підсистеми цифрового двійника можуть змінювати налаштування вимірювань, наприклад інтерфейс повинен бути REST API, оскільки REST API є простим у використанні та широко розповсюдженим способом реалізації інтерфейсів.

Щоб полегшити процес конфігурації датчиків, система повинна автоматично генерувати код для сенсорних вузлів. Це дозволить використовувати платформу без попередніх знань про мікроконтролери та їх програмування.

Розподіл вимоги також представлені в таблиці 1. Вони розділені на вимоги та побажання, щоб показати важливість вимоги для функціональності платформи.

Таблиця 1.1 - Перелік вимог до платформи конфігуратора датчиків

Вимоги	Пріоритет
Зв'язок	
Підтримка Wi-Fi	обов'язково
Підтримує принаймні одну іншу технологію зв'язку	по можливості

Продовження таблиці 1.1

Додавання нових технологій зв'язку Легко	обов'язково
Підтримка MQTT	обов'язково
Підтримка DDS	обов'язково
Підтримка інших протоколів прикладного рівня	по можливості
Додавання нових протоколів є простим	по можливості
Функціональні можливості	
Віддалене налаштування датчиків	обов'язково
Конфігурація через інтернет	обов'язково
Можливість вимірювання сплесків	по можливості
Підтримка датчиків шини I2C	обов'язково
Просте додавання нових датчиків	по можливості
Параметри, що налаштовуються	
Частота дискретизації	обов'язково
Чутливість	по можливості
Налаштування Wi-Fi (SSID, ключ, тип авторизації)	обов'язково
Швидкість передачі даних	по можливості
Довжина та частота пакетів	по можливості
Інтерфейс користувача	
Простий у використанні	по можливості
Візуалізація даних	по можливості
Можливість завантаження даних	по можливості
Сенсорні вузли	
Низьке споживання енергії	по можливості
Світлодіодні індикатори стану	по можливості
Підтримка декількох технологій зв'язку	по можливості
API	

Продовження таблиці 1.1

Датчики можна додавати та видаляти	обов'язково
Датчики можна змінювати	обов'язково
НТТР Базова автентифікація	обов'язково
Автентифікація на основі токенів	по можливості

Для досягнення мети потрібно вирішити наступні завдання:

1. дослідити передумови виникнення такого поняття як цифровий двійник;
2. провести аналіз існуючих рішень цифрових двійників;
3. сформувати вимоги до системи передачі сенсорних даних від фізичного двійника до цифрового;
4. реалізувати алгоритмічне та інформаційне забезпечення системи;
5. Розробити загальну структуру системи та описати її;
6. Розробити алгоритм моделювання цифрового двійника;
7. Визначити комунікаційні потреби цифрового двійника;
8. Вибрати апаратне забезпечення системи
9. Вибрати та проаналізувати програмне забезпечення для системи
10. Розробити програмне забезпечення апаратної частини системи
11. Розробити програмне забезпечення «серверної» частини системи
12. Провести тестування системи

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Загальна структура проектованої системи

Пропонується шестирівнева архітектура цифрового двійника, яка представлена на рисунку 2.1 і забезпечує зв'язок між фізичним двійником і цифровим двійником, а також між цифровим двійником і зовнішнім світом.

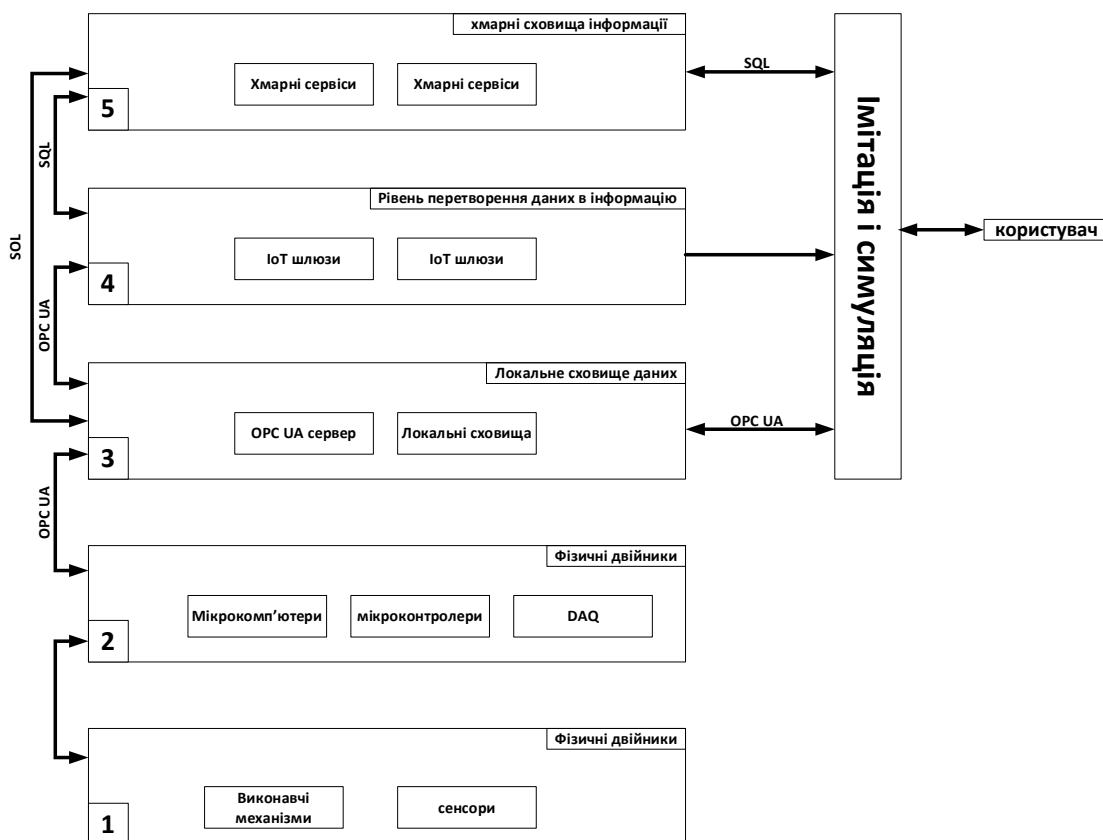


Рисунок 2.1 - архітектура цифрового двійника

Архітектура орієнтована на ситуації, коли у фізичному двійнику, а також у решті частини цифрового двійника використовуються продукти різних постачальників. Власні та розроблені на замовлення елементи зведені до мінімуму, щоб зменшити витрати на розробку та підтримку. Для зв'язку між рівнями використовуються відкриті або нейтральні до постачальників формати.

На рисунку 2 показано потоки даних/інформації між зазначеними рівнями. На рисунку показано, що дані/інформація передаються від фізичної системи або фізичного двійника (рівень 1) до хмари (рівень 5), де вони зберігаються в інформаційному сховищі, доступному в кіберпросторі.

Рівні 1 і 2 складаються з фізичного двійника. Рівень 3 складається з локального сховища даних, яке використовується для отримання значень датчиків від контролерів на рівні 2. Рівень 4 - це спеціально розроблений IoT-шлюз або перетворювач даних в інформацію. Рівні 5 і 6 складаються з хмарного сховища та інструментів емуляції та моделювання відповідно..

Рівні 1 і 2: фізичний двійник Вбудовані фізичні пристрої, об'єкти та обладнання є частиною "рівня інтелектуального з'єднання". Рівень 1 включає різні фізичні пристрої, такі як виконавчі механізми і датчики, які можуть надавати або споживати сигнали, якими обмінюються з локальним контролером або пристроєм збору даних, розташованим на рівні 2. Рівень 2 є джерелом даних для фізичного двійника.

Рівень 3 забезпечує незалежний від постачальника комунікаційний інтерфейс між фізичним двійником та іншими рівнями, як показано на рис. 2. Загалом, рівень 3 в основному взаємодіє з рівнем 4, але він також може реєструвати дані безпосередньо в хмарі на рівні 5 (якщо дані на рівні 3 підходять) і навіть взаємодіяти з рівнем 6 (наприклад, коли важливі низькі затримки зв'язку). Тут OPC UA [23] вважається цінним інструментом для обміну даними з різними контролерами та пристроями збору даних. OPC UA також забезпечує необхідну безпеку для захисту даних/інформації від витоку.

Деякі сервери OPC UA надають доступ до більш ніж 150 драйверів пристроїв, що робить його функціональним інструментом для зв'язку між різними пристроями, а також між реальним простором і кіберпростором. Це також функціонально, оскільки деякі компанії або виробничі галузі використовують обладнання, яке надається різними компаніями або постачальниками.

										Арк.
										23
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ					

Рівень 4: Шлюз IoT [24] "Рівень перетворення даних в інформацію", або шлюз IoT, додається як рівень 4. Цей рівень додає контекст до даних, отриманих з рівня 3, та/або обробляє дані у формі, більш корисній для вищих рівнів. Цей рівень з'єднує рівень 3 з рівнем 5, перетворюючи дані, отримані з рівня 2, в інформацію для відправки в хмарне сховище на рівні 5. Тут спеціально розроблений додаток може бути використаний для взаємодії з сервером OPC UA на рівні 3 за допомогою клієнтського з'єднання з сервером. Потім додаток може підписатися на теги на сервері OPC UA, які мають значення для вищих рівнів архітектури.

Рівень 5: хмарні сховища інформації [25]. Рівень 5 представляє собою хмарне сховище інформації, як показано на рис. 2. Рівень 5 складається з хмарних сервісів, які зберігають історичну інформацію, отриману з рівня 4. Ця інформація може бути доступна з рівня 6 і може бути корисною для прийняття рішень шляхом оцінки поточного стану фізичного двійника.

Передбачається кілька сховищ, оскільки різні зацікавлені сторони, ймовірно, матимуть різні інформаційні потреби і права доступу.

Наприклад, розробнику фізичного двійника може знадобитися доступ до критично важливих параметрів продуктивності, але він може захотіти контролювати доступ до цієї інформації, в той час як завод-виробник може вимагати зберігання інформації про забезпечення якості, але бажає зберігати цю інформацію конфіденційною.

Розміщення цих репозиторіїв у хмарі підвищує доступність і зв'язність цифрового двійника. Спеціалізовані знання, необхідні для управління таким сервером бази даних, враховуючи масштабованість, надійність і безпеку, зазвичай не є легкодоступними на виробничих підприємствах.

Рівень 6: емуляція та імітування. У той час як рівні з 1 по 5 забезпечують необхідну інфраструктуру, на рівні 6 додається інтелект цифрового двійника (див.рис. 2).

Як видно з рисунку 2, рівень 6 з'єднується з рівнями 3, 4 і 5, і, таким чином, може забезпечити функціональність користувачького інтерфейсу (або

										Арк.
										24
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ					

інформаційної панелі), який з'єднує користувача з інформацією в реальному часі та історичною інформацією про фізичного двійника. Тому рівень 6 повинен бути оснащений програмним забезпеченням для емуляції та моделювання, яке дозволяє користувачеві взаємодіяти з цим рівнем. Цей рівень, залежно від програмного забезпечення для емуляції та моделювання, може також надавати користувачеві графічне 3D-представлення фізичного двійника.

2.2 Алгоритм моделювання цифрового двійника

Створення цифрового двійника вимагає хорошого розуміння процесів. По-перше, необхідно створити карту процесу з усіма вхідними інформаційними потоками та операціями. Відомість матеріалів та робочі інструкції можуть допомогти підготувати карту процесу. Основним модулем ЦД є симуляція, яка моделюється на основі технологічної карти.

Технологічна карта містить вимоги до точок даних, необхідних для моделі. Ключовими вимогами до реалізації є: (1) визначення сценарію функціонування, який має бути виконаний, (2) збір та аналіз операційних даних для визначення ключових параметрів процесу роботи, (3) створення цифрового двійника з інтеграцією ключових параметрів руху та операційних обмежень, моделювання поведінки фізичних об'єктів, і (4) симуляція процесу функціонування та його оптимізація відповідно до набору оптимізаційних обмежень.

Що стосується алгоритму моделювання, то він складається з декількох етапів. По-перше, необхідно визначити період часу та надати початкові дані. Симуляція запускається і зупиняється після кожного періоду часу і чекає на "введення нових даних". Отже, після кожного запуску симуляція зупиняється і задає два питання 1) чи є нові дані? 2) чи хоче користувач переглянути найгірше виконані операції? На основі відповідей алгоритм моделювання розгалужується на 4 різні шляхи (рисунок 2.2).

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

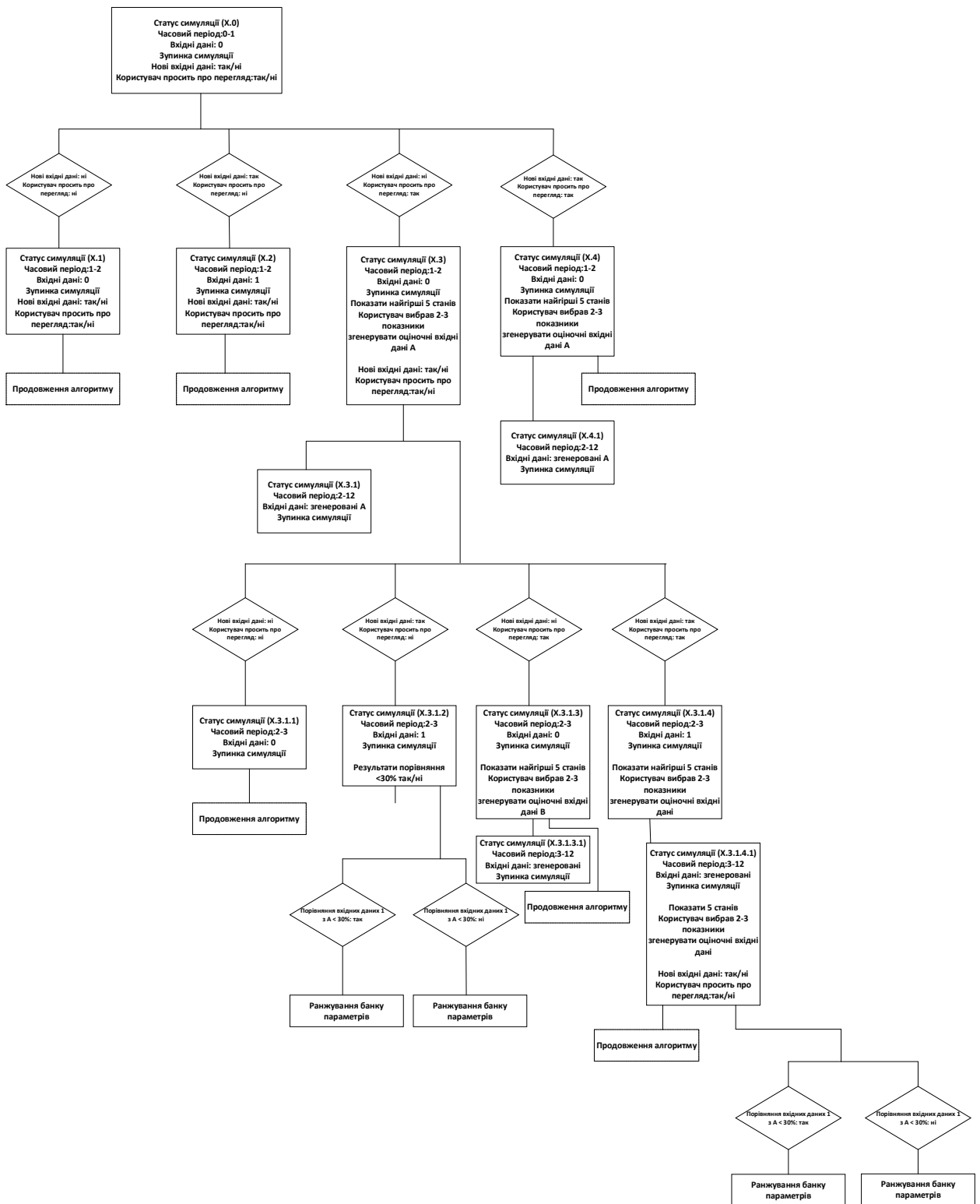


Рисунок 2.2 - Алгоритм моделювання цифрового двійника

Ці 4 варіанти описані нижче

а) "НЕ" введенню нових даних, і користувач "НЕ" хоче переглянути ефективність: моделювання переходить до наступного періоду часу з

Змн.	Арк.	№ докум.	Підпис	Дата

використанням попередніх даних. Огляд продуктивності також не буде виконано. Алгоритм продовжить роботу після цього.

б) Нові дані "існують", а користувач "НЕ" бажає переглянути ефективність: симуляція переходить до наступного часового періоду з використанням нових даних. Аналіз ефективності не буде виконано. Алгоритм продовжить роботу після цього.

с) "НІ" введення нових даних, а користувач "хоче" переглянути ефективність: симуляція переходить до наступного часового періоду з використанням попередніх даних. Для аналізу ефективності будуть показані 5 найгірших станів на основі обраних користувачем критеріїв ефективності. Потім з'явиться список рекомендацій на основі знань щодо кожного з найгірших станів. Користувач обирає одну або дві рекомендації зі списку і впроваджує їх на реальному об'єкті.

На цьому етапі симуляція буде розділена на дві частини. По-перше, припускається, що певний відсоток покращення відбудеться на основі рекомендованого вибору. Параметр критерію ефективності буде змінено/покращено, що створить нові вхідні дані, які називаються "Орієнтовні вхідні дані". Використовуючи оціночні дані, симуляція пройде весь шлях до кінця загального періоду часу, не чекаючи на вибір користувача. Для другої частини симуляція продовжуватиме зупинятися після кожного періоду часу і чекатиме на "нове введення даних". Після цього моделювання продовжиться згідно з алгоритмом моделювання 4-х розгалужень

д) Нові дані "існують" і користувач "хоче" переглянути ефективність: симуляція переходить до наступного часового періоду з використанням нових даних. Сегмент аналізу ефективності буде виконано відповідно до вищезгаданого сегменту.

Іншою важливою частиною алгоритму моделювання є ранжування та оновлення списку рекомендацій. Спочатку, на основі загальних знань про вибір, генеруються нові оціночні дані, які використовуються для завершення моделювання до кінця загального періоду часу. Після цього запускається інша

									Арк.
									27
Змн.	Арк.	№ докум.	Підпис	Дата					

ДП.КН.9500038.069.ПЗ

частина симуляції, яка зупиняється і шукає нові дані. Отримавши нові дані, симуляція порівнює їх з розрахунковими даними щодо покращених параметрів, тобто параметрів, на які вплинули рекомендації. Для оцінки того, чи дійсно обрана користувачем рекомендація вплинула на результат, буде призначено порогове значення. Якщо так, то список рекомендацій запам'ятає результат, скоригує свій список рекомендацій і відповідно змінить рейтинг.

Наступного разу, коли користувач перебуватиме на подібному етапі і шукатиме огляд результатів, список рекомендацій / банк варіантів буде ранжований і покаже історичну значущість.

2.3 Комунікація цифрового двійника та комунікаційні потреби

Оскільки цифровий двійник не є цифровим двійником за визначенням, якщо він не має інформації про поточний стан фізичного двійника. Дані вимірювань з датчиків використовуються для постійного оновлення цифрового двійника, щоб він міг мати інформацію про поточний стан.

У роботі основна увага приділяється даним вимірювань, зібраним датчиками, підключеними до фізичного двійника, тобто зв'язку між фізичним і цифровим двійником. Вузли датчиків, приєднані до фізичного двійника, повинні бути бездротовими, оскільки прокладання проводів стає складним завданням, якщо є велика кількість вузлів датчиків.

Сенсорний вузол - це пристрій IoT, який містить датчик(и) і мікроконтролер, здатний передавати дані бездротовим способом. Застосування пристроїв IoT можна розділити на наступні категорії [26]:

- додатки "точка-точка" в режимі реального часу, в яких стан пристрою повинен передаватися надійно з низькою затримкою

- додатки для моніторингу, які періодично збирають дані - обмін даними між об'єктами, такими як транспортні засоби

- 1) Дані з датчиків використовуються для контролю в режимі реального часу

									Арк.
									28
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ				

2) Дані з датчиків використовуються для моніторингу

У першому випадку найважливішою вимогою до комунікації є низька затримка. Крім того, вирішальними факторами є висока надійність і безпека. Наприклад, якщо зв'язок перерветься, об'єкт не зможе працювати, або якщо будуть отримані неправильні дані вимірювань, об'єкт може працювати неправильно або навіть небезпечно. Положення певного вузла об'єкта можна виміряти, наприклад, за допомогою лазерних датчиків, що вимірюють положення тіла, і датчиків обертання, які вимірюють, скільки обертів зробили колеса і двигун.

У другому випадку зв'язок повинен бути максимально ефективним, тобто використання смуги пропускання, а також енергоспоживання сенсорного вузла повинно бути якомога меншим.

Якщо споживання енергії достатньо низьке, сенсорні вузли можуть живитися від батарейок, що робить їх справді бездротовими. Оскільки обчислювальна потужність сенсорних вузлів обмежена, протоколи повинні бути легкими в обох випадках використання.

Крім того, важливими факторами є простота реалізації та наявність підтримки. У цьому випадку напруження вимірюється за допомогою тензодатчика, що можливо, якщо властивості матеріалу добре відомі.

Деформація вимірюється раз на секунду і зберігається як ціле число (4 байти). Таким чином, обсяг даних становить 240 байт на хвилину і 345,6 кБ на день. Дані вимірювань надсилаються до сховища даних раз на хвилину. Вимоги до зв'язку для обох варіантів використання наведені в таблиці 2.1.

У першому випадку дані з датчиків використовуються для керування фізичним двійником у режимі реального часу. Таким чином, потрібне безпечне, надійне з'єднання з низькою затримкою. Протокол, який здатний передавати дані в реальному часі - це DDS.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Таблиця 2.1- Вимоги до варіантів використання.

	Рішення1	Рішення2
Низька затримка	+	
Низька обчислювальна потужність	+	+
Низьке використання пропускну здатності		+
Низьке енергоспоживання		+
Безпека	+	+
Надійність	+	
Простота впровадження	+	+

Крім того, він пропонує вбудовану безпеку, надійність, низьку затримку і високу передбачуваність. Тому він є очевидним вибором для першого варіанту використання. До недоліків DDS можна віднести високе використання пропускну здатності і пам'яті, а також середнє використання процесора, що обмежує його використання в дуже обмеженому середовищі.

Другий варіант використання, в якому напруга вимірюється періодично, не має жорстких вимог до зв'язку з точки зору затримки, швидкості передачі даних або надійності. Тому всі протоколи певною мірою підходять для цього варіанту використання.

Однак низьке споживання енергії, придатність для обмежених пристроїв і загальна ефективність є бажаними властивостями для того, щоб сенсорні вузли працювали від батареї і, таким чином, були справді бездротовими. Найбільший акцент робиться на придатності для пристроїв з обмеженими можливостями, що включає низьке енергоспоживання і використання смуги пропускання.

Тому найбільш підходящими протоколами є CoAP, MQTT і AMQP.

MQTT [27] - це протокол публікації/підписки, CoAP [28]- це протокол запиту/відповіді, а AMQP [29] підтримує, на додаток до моделі публікації/підписки, зв'язок точка-точка.

У другому варіанті використання дані надсилаються до одного об'єкта: сховища даних цифрового двійника. Таким чином, модель "опублікувати-підписатися" не має переваг над моделлю "запит/відповідь". І MQTT, і AMQP використовують TCP як базовий протокол транспортного рівня, тоді як CoAP використовує UDP. Таким чином, CoAP повинен працювати краще в мережах з втратами, оскільки контроль перевантажень TCP не підходить для таких типів мереж.

Використання смуги пропускання для кожного з цих протоколів є досить схожим, так само як і використання процесора, але AMQP використовує трохи менше пам'яті.

CoAP споживає менше енергії, ніж MQTT, при розмірі пакету менше 1024 байт. При більших розмірах пакетів енергоспоживання CoAP збільшується через фрагментацію пакетів і, таким чином, MQTT має менше енергоспоживання при великих розмірах пакетів. але при низькій швидкості втрати пакетів і затримці MQTT споживає більше пропускну здатності, ніж CoAP, але збільшення обох показників зменшує споживання пропускну здатності, в той час як CoAP має стабільне споживання пропускну здатності. Тому, при високих втратах пакетів і затримках, MQTT працює краще, ніж CoAP з точки зору використання пропускну здатності.

До недоліків можна віднести те, що в AMQP не вистачає бібліотек з відкритим вихідним кодом для вбудованих пристроїв, а в CoAP відсутня підтримка інструментів і бібліотек. Наявність бібліотек робить реалізацію MQTT простішою, ніж CoAP і AMQP. Оскільки властивості протоколів в іншому досить схожі, це робить MQTT рекомендованим вибором для другого варіанту використання.

Однак, у мережах з втратами кращим вибором є CoAP, оскільки він використовує UDP замість TCP як базовий протокол транспортного рівня. Загалом, при виборі протоколу прикладного рівня важливими факторами є простота реалізації та інтеграції в існуючі системи, а також наявність підтримки. Крім того, при виборі протоколу прикладного рівня слід

									Арк.
									31
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ				

враховувати використовувану технологію зв'язку і навпаки, оскільки, наприклад, висока втрата пакетів в мережі по-різному впливає на різні протоколи прикладного рівня.

2.3.1 Широкопasmові мережі з низьким енергоспоживанням (LPWAN)[30].

Стільникові мережі мають дуже високі теоретичні швидкості і низькі затримки порівняно з LPWAN. Однак слід підкреслити, що теоретичні швидкості рідко досяжні.

Крім того, стільникові мережі споживають більше енергії, ніж LPWAN. Перевагою 4G [31] над іншими мережами є його доступність: він вже розгорнутий і активно використовується. І LPWAN, і стільникові мережі дозволяють пересування машини на великій території порівняно з мережами короткого радіусу дії. Однак обидві мережі підходять для великих територій, але якщо брати малі об'єкти то тоді можна зіткнутись з деякими проблемами. Рішення цього лежить в мережах короткого радіусу дії.

Мережі малого радіусу дії мають великий розкид у швидкості передачі даних. 802.11ah має найшвидшу швидкість передачі даних разом з Bluetooth.

Однак теоретична максимальна швидкість 802.11ah досяжна не на всіх континентах. Радіус дії мереж в основному залежить від використовуваного діапазону частот: у діапазоні 900 МГц радіус дії значно більший, ніж у діапазоні 2,4 ГГц. Єдиним промисловим протоколом який розглядався є WirelessHART [32]. Порівняно з іншими протоколами, він є більш надійним і має детерміновану затримку та вбудований захист. Тому він є найбільш підходящою комунікаційною технологією для випадків використання, в яких потрібна висока передбачуваність і безпека. Однак, висока затримка WirelessHART зменшує її придатність для керування машинами

Наведені вище комунікаційні потреби показують, що мережі короткого радіусу дії мають велику варіацію своїх властивостей. Тому кожна з мереж має свої найбільш підходящі застосування.

										Арк.
										32
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ					

Існує багато аспектів, які стосуються вибору комунікаційних технологій, таких як швидкість передачі даних, затримка, надійність, безпека та енергоспоживання. Крім того, деякі з властивостей, такі як енергоспоживання, залежать від варіанту використання, наприклад BLE має менше споживання енергії на передані байти, ніж ZigBee [32].

Деякі з комунікаційних технологій є IP-сумісними, що може полегшити розгортання системи, оскільки шлюзи не потрібні. Доступність мережі також впливає на вибір технології зв'язку. Наприклад, стільникові технології, такі як 4G і Wi-Fi, вже широко розгорнуті і використовуються. Крім того, важливим фактором є наявність пристроїв та аксесуарів. Наприклад, Bluetooth і Wi-Fi підтримуються практично будь-яким пристроєм/мікроконтролером порівняно з, наприклад, Z-Wave і ZigBee. Однак підтримку ZigBee і Z-Wave можна додати за допомогою зовнішніх модулів (Adafruit Industries,).

Жодна з комунікаційних технологій не є технічно кращою за іншу, і один і той самий варіант використання може бути реалізований за допомогою різних комунікаційних технологій. Це може бути причиною, чому існує так багато доступних комунікаційних технологій.

В обох випадках використання мережі малого радіусу дії є оптимальним варіантом, оскільки воно підходить до статичного об'єкту, оскільки діапазон переміщення становить кілька десятків метрів.

У першому випадку, коли дані з датчика використовуються для керування машиною, потрібна дуже низька затримка. Тому відповідними варіантами є 4G, 5G, Bluetooth або 802.11ah/802.11n. Однак всі ці технології також використовуються іншими пристроями, що може спричинити перевантаження. Перевантаження збільшує затримку і сповільнює швидкість передачі даних, що робить поведінку непередбачуваною. WirelessHART має інші відповідні властивості, такі як надійність і детермінована затримка, але його затримка занадто висока для керування. Вибір для першого випадку - 802.11n у частотному спектрі 5 ГГц, оскільки він має найнижчу затримку.

									Арк.
									33
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ				

Однак, з 802.11n сенсорний вузол не може працювати від батареї через високе енергоспоживання.

Для другого випадку використання не існує єдиної "найбільш підходящої технології", оскільки вимоги до швидкості передачі даних і затримки не є жорсткими. Лише Sigfox не відповідає вимогам, оскільки кількість висхідних повідомлень обмежена до 140 на день. Загалом, Sigfox підходить для випадків, коли статус об'єкту оновлюється досить рідко, а не для надсилання даних вимірювань через невелике корисне навантаження та обмежену кількість висхідних повідомлень.

Wi-Fi (802.11n) є кращим вибором для другого варіанту використання завдяки простоті реалізації. Він пропонує IP-сумісність, а розгортання мережі не вимагає особливих зусиль. Якщо низьке споживання енергії розглядається як найважливіший фактор, Bluetooth є рекомендованим вибором для мереж малого радіусу дії, а LoRaWAN - для мереж великого радіусу дії.

Загалом, якщо є кілька датчиків, які вимірюють з дуже високою частотою дискретизації, тільки 4G, 5G, Wi-Fi або Bluetooth здатні передавати дані на досить високих швидкостях. 4G і 5G дозволяють вільно пересуватися машині, в той час як Wi-Fi і Bluetooth дозволяють пересуватися тільки на дуже обмеженій ділянці (максимальна дальність Wi-Fi 802.11ah - 1 км). Загалом 802.11ah був би хорошим вибором, оскільки він має низьке енергоспоживання, великий радіус дії та високу швидкість передачі даних, але на ринку немає пристроїв, які б його підтримували.

2.3.2 Платформи IoT

Для того, щоб забезпечити збір даних з фізичного двійника на практиці, необхідна система управління сенсорними вузлами.

В огляді існуючих платформ IoT [34-36] були розглянуті наступні властивості, які є корисними для платформи, що дозволяє керувати сенсорними вузлами:

- Можливість змінювати налаштування вимірювань

									Арк.
									34
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ				

- Підтримка декількох протоколів прикладного рівня
- Підтримка декількох типів датчиків
- Легке додавання нових типів датчиків
- Безкоштовне та відкрите програмне забезпечення

Платформи IoT можна розділити на наступні три категорії

- 1) Управління пристроями
- 2) управління додатками, в якому платформа використовується для розробки додатків
- 3) Увімкнення додатків, в якому платформа дозволяє розробляти зовнішні додатки, наприклад, надаючи дані з датчиків.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Апаратне та програмне забезпечення системи

3.1.1 Апаратне забезпечення

Апаратна частина реалізованої системи складається з датчиків, мікроконтролерів та Raspberry PI (рисунок 3.1)[37].

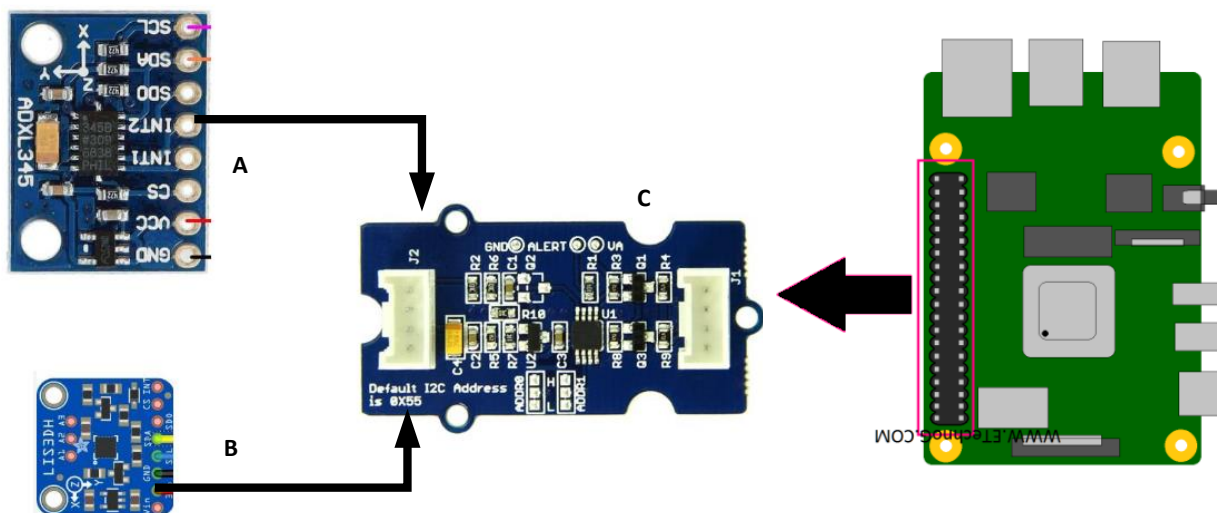


Рисунок 3.1 – Апаратне забезпечення системи

Вузол датчика складається з мікроконтролера та, наразі, двох датчиків. Датчики використовують шину I2C для зв'язку з мікроконтролерами. Система наразі підтримує наступні датчики:

- ADXL345, трьохосовий акселерометр [38], показаний на рисунку 3.1а.

Цей сенсор добре підходить для використання в мобільних пристроях. Він вимірює статичне прискорення сили тяжіння в системах вимірювання нахилу, а також динамічне прискорення, що виникає в результаті руху або удару. Висока роздільна здатність (4mg/LSB) дозволяє вимірювати зміни нахилу менше ніж на $1,0^\circ$.

Передбачено кілька спеціальних функцій зондування. Функція визначення активності та бездіяльності виявляє наявність або відсутність руху, порівнюючи прискорення по будь-якій осі з пороговими значеннями,

встановленими користувачем. Датчик дотиків виявляє одинарні та подвійні дотики в будь-якому напрямку. Датчик вільного падіння визначає, чи падає пристрій. Ці функції можуть бути призначені індивідуально на будь-який з двох виводів переривання. Вбудована система керування пам'яттю з 32-рівневим буфером FIFO (перший прийшов - перший пішов) може використовуватися для зберігання даних, щоб мінімізувати навантаження на процесор і знизити загальне енергоспоживання системи.

Режими низького енергоспоживання забезпечують інтелектуальне керування живленням на основі руху з визначенням порогових значень та активним вимірюванням прискорення при надзвичайно низькому рівні розсіювання енергії.

- LIS3DSH, тривісний акселерометр (рисунок 3.1 б) [39].

Даний сенсор є трьохосовий лінійний акселерометр з надвисокою роздільною здатністю і низьким рівнем шуму. LIS3DHH має повну шкалу у 2.5g і здатний передавати виміряні прискорення до програми через 4-провідний цифровий інтерфейс SPI.

Інтерфейс IC виготовлено за допомогою КМОП-процесу, який забезпечує високий рівень інтеграції для розробки спеціальної схеми.

LIS3DHH доступний у високоефективному (низьконапруженому) корпусі може працювати в діапазоні температур від -40 °C до +85 °C.

- Seeed Grove I2C ADC [40] (рисунок 3.1 с), 12-розрядний аналого-цифровий перетворювач на базі ADC121C021. Він допомагає підвищити точність значення, зібраного з аналогового датчика, забезпечуючи постійну опорну напругу. Оскільки його адреса є змінною, можна використовувати до 9 АЦП I2C одночасно. З іншого боку, цей модуль має функцію автоматичного переходу в режим сну, що значно знижує енергоспоживання.

Для організації підключення та передачі даних в різних комунікаційних мережах було використано мікроконтролер Русом WiPy (рисунок 3.2)

Raspberry Pi - це одноплатний комп'ютерів, створений Raspberry Pi Foundation, останні моделі, яких мають чотирьохядерний процесор з тактовою частотою понад 1,5 ГГц і 4 ГБ оперативної пам'яті.

У всьому світі люди використовують Raspberry Pi для програмування, створення апаратних проектів, домашньої автоматизації, впровадження кластерів Kubernetes і периферійних обчислень, і навіть для промислового застосування. Raspberry Pi працює під управлінням Linux і має набір виводів GPIO (загального призначення), що дозволяє керувати електронними компонентами для фізичних обчислень, досліджувати Інтернет речей (IoT) та цифрових двійників.

3.1.2 Програмне забезпечення

В даному проекті MicroPython використовується для програмування мікроконтролерів.

Платформа конфігуратора датчиків та сервер даних будуть використовувати Django, який є веб-фреймворком Python з відкритим вихідним кодом. Django було обрано тому, що він дозволяє швидко створювати та розвивати веб-сайти. Крім того, він має численні вбудовані функції, такі як автентифікація користувача та сторінка адміністрування. (Django Software Foundation)[44].

Також буде додано кілька плагінів, доступних для Django. Наприклад, Django REST framework був використаний для створення REST API (Django REST framework) [45] та Simple JWT [46] для автентифікації. У цьому контексті REST API описує інтерфейс, який інші системи можуть використовувати для зв'язку з платформою конфігуратора датчиків. REST API широко використовуються, в багатьох додатках реалізована підтримка REST API, і вони, як правило, прості у використанні.

База даних SQLite використовується, як для платформи конфігуратора датчиків, так і для сервера даних.

На фронтенді SCP були використані наступні бібліотеки/фреймворки:

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

- Bootstrap - фреймворк з відкритим вихідним кодом для створення адаптивних веб-інтерфейсів [47].

- jQuery, бібліотека JavaScript для полегшення "обходу та маніпуляцій з HTML-документами, обробки подій, анімації та Ajax" [48].

- Plotly.js, бібліотека JavaScript з відкритим кодом для створення графіків [49].

- Handsontable, JavaScript-компонента для створення електронних таблиць [50].

Для реалізації MQTT-комунікації було використано хмарний сервіс Adafruit IO. Він пропонує візуалізацію даних та можливість додавання тригерів для виконання певних дій на основі отриманих даних. Крім того, він має безкоштовну версію з обмеженими властивостями. Властивостей безкоштовної версії достатньо для другого варіанту використання.

3.2 Розробка програмного забезпечення апаратної частини та інтерфейсу цифрового двійника

3.2.1 Програмне забезпечення для ADXL345

Підтримка послідовних протоколів I2C та SPI робить сенсор простим у використанні з Raspberry Pi, оскільки не потрібно ніякого додаткового обладнання для інтерпретації даних, що надходять з акселерометра. На рисунку 3.4 представлено варіант підключення.

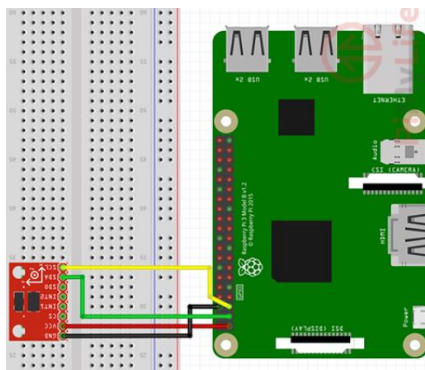


Рисунок 3.4 – Підключення сенсора ADXL345

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Після того, як Raspberry Pi завершить оновлення, потрібно буде запуснути інструмент конфігурації Raspberry, щоб увімкнути I2C на Raspberry Pi : `sudo raspi-config`. Після потрібно перейти до меню "5 параметри інтерфейсу", потім в меню параметрів інтерфейсу перейти до пункту "P5 I2C" та увімкнути інтерфейс та перезавантажити Raspberry. Після того як інтерфейс налаштований, необхідно завантажити додаткові бібліотеки підтримки сенсора за допомогою команди `pip3 install adafruit-circuitpython-ADXL34x`. Далі представлено основні функції роботи з даним сенсором:

- `import time` - використання бібліотеки "time" робиться для того, щоб зупиняти скрипт на короткий час.

- `import board` - бібліотека від Adafruit. Ця бібліотека призначена для того, щоб швидко дізнатися, які контакти доступні на пристрої.

- `import busio` - модуль "busio" від Adafruit. Цей модуль містить безліч різних бібліотек для роботи з різними послідовними протоколами - в проекті для роботи з послідовним протоколом I2C.

- `import adafruit_adxl34x` - імпорт бібліотеки "adafruit_adxl34x". Ця бібліотека містить весь необхідний код для зчитування інформації з акселерометра ADXL345.

- `accelerometer = adafruit_adxl34x.ADXL345(i2c)` – інсталяція бібліотеки ADXL345 у об'єкт "акселерометр". Далі цей об'єкт буде використовуватись для зчитування та отримання інформації з датчика. У конструкторі бібліотеки передається дескриптор I2C.

- `while True: print("%f %f %f"% accelerometer.acceleration) time.sleep(0.5)`
– запуск нескінченного циклу, використовуючи "while True:". У цьому нескінченному циклі виводиться значення прискорення за осями X, Y та Z, отримані бібліотекою з акселерометра. Після того, як виведено значення X, Y і Z, скрипт переводиться у режим сну на півсекунди. Скрипт потрібно зупинити, щоб він не переповнював командний рядок значеннями, що надаються акселерометром.

3.2.2 Програмне забезпечення для LIS3DSH

На рисунку 3.5 представлено діаграму підключення сенсора до мікрокомп'ютера.

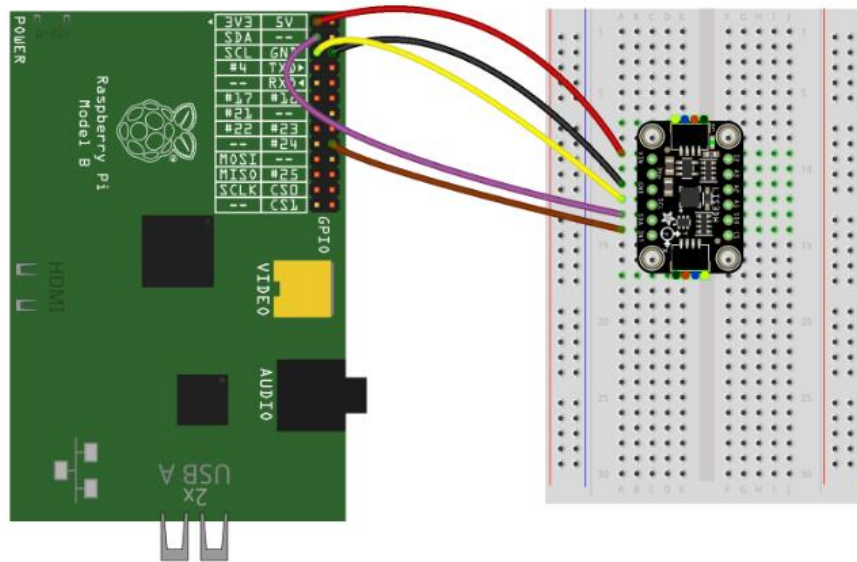


Рисунок 3.5 - LIS3DSH+Raspberry PI

Для зв'язку з акселерометром LIS3DH буде використовуватись бібліотека `circuitPython` від Adafruit. Щоб встановити цю бібліотеку, потрібно виконати наступну команду - `sudo pip3 install adafruit-circuitpython-lis3dh`.

Нижче приведено основні функції які будуть використовуватись:

- `import time, board, digitalio, busio, adafruit_lis3dh`: Імпортує необхідні бібліотеки.

- `i2c = busio.I2C(board.SCL, board.SDA)`: Ініціалізує інтерфейс I2C.

- `int1 = digitalio.DigitalInOut(board.D24)`: Встановлює переривання на Raspberry Pi GPIO 24

- `x, y, z = lis3dh.acceleration`: Запитує у LIS3DH значення акселерометра.

- `print("%0.3f %0.3f %0.3f %0.3f" % (x, y, z))`: Виводить значення з 3 десятковими знаками.

- `while True: if lis3dh.shake(shake_threshold=15): print("Shaken!")`. Щоб використовувати тремтіння, можна встановити додатковий параметр

										Арк.
										42
Змн.	Арк.	№ докум.	Підпис	Дата						

shake_threshold=15. За замовчуванням встановлено 30, але менші значення полегшують отримання реакції на струс.

3.2.3 Програмне забезпечення для Seeed Grove I2C ADC

Це не було основним завданням, але дуже часто потрібно додати аналогові сенсори для більш кращого розуміння навколишнього середовища навколо а в середині об'єкта дослідження.

Для початку потрібно клонувати останню версію бібліотеки сховища python.py: `git clone https://github.com/Seeed-Studio/grove.py`

Після всіх базових налаштувань Raspberry Pi можна запуснути демонстраційний код світлодіода для перевірки роботи збірки в цілому.

3.2.4 Програмне забезпечення для Django

Щоб запуснути Django на Raspberry Pi, необхідно встановити Apache. Щоб встановити apache2 на Raspberry Pi, потрібно виконати в терміналі наступну команду: `sudo apt install apache2 -y`.

На додаток до базової версії Apache, потрібно буде встановити деякі додаткові пакунки, щоб Apache міг спілкуватися з Python. Відповідний пакет можна встановити, виконавши наступну команду на Raspberry Pi : `sudo apt install libapache2-mod-wsgi-py3`.

Коли веб-сервер і Python встановлено, потрібно налаштувати сам Django. Тепер, коли всі наші папки налаштовано і віртуальне середовище для Python готове, ми можемо нарешті встановити Django. Щоб створити віртуальне середовище, потрібно використати його як джерело. Щоб використати віртуальне середовище, як джерело, потрібно виконати наступну команду: `source djenv/bin/activate`. Після запуску цієї команди буде виведено повідомлення "(djenv)" на початку командного рядка. Це означає, що використовується нове віртуальне середовище як джерело.

										Арк.
										43
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ					

Тепер, коли є нове віртуальне середовище можна встановлювати фреймворк Django. Оскільки Django доступний у вигляді пакунка Python, можна скористатись наступною командою - `python3 -m pip install django`.

Після встановлення Django можна створити новий проект виконавши команду `django-admin startproject pidjango`, де останнім бітом тексту буде назва проекту. Після чого можна відкрити браузер та ввести адресу `http://127.0.0.1:8000`, після чого появиться головне вікно проекту (рисунок 3.6).

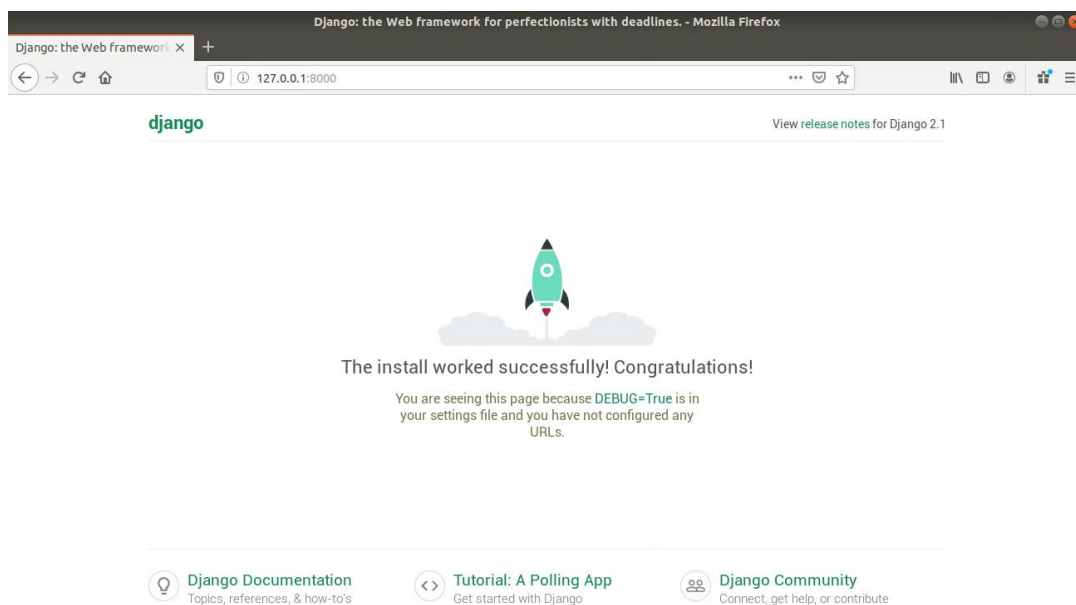


Рисунок 3.6 – Головне вікно Django

3.3 Тестування системи

На першому етапі є платформа конфігуратора датчиків. Мета платформи конфігуратора датчиків - забезпечити керування вузлами датчиків, підключених до фізичного двійника, та передачу їхніх даних. Керування включає додавання/видалення сенсорних вузлів, а також зміну їхніх налаштувань вимірювання та передачі даних.

Платформа дозволяє змінювати налаштування сенсорного вузла віддалено через Інтернет. Передача даних забезпечується за допомогою конфігурації сенсорних вузлів без кодування через веб-інтерфейс користувача

									Арк.
									44
Змн.	Арк.	№ докум.	Підпис	Дата					

(WUI), показаний на рисунку 3.7, і пропонує кілька варіантів передачі даних, таких як вибір між кількома протоколами прикладного рівня.

Id	Name	Model	Location	Date added	Last modified	Status	Actions
108	Test1	ADXL345		Jan. 14, 2023, 1:01 p.m.	Jan. 30, 2023, 9:31 a.m.	Measuring. Waiting-for-update	Info Edit Data Delete
109	test	ADXL345		Jan. 17, 2023, 11:11 a.m.	Feb. 25, 2023, 10 a.m.	Measuring	Info Edit Data Delete
110	ADC test	seeed Grove i2C ADC		Jan. 18, 2023, 10:11 a.m.	Jan. 28, 2023, 11:02 a.m.	Waiting-for-update	Info Edit Data Delete

Рисунок 3.7 - конфігурація сенсорних вузлів через веб-інтерфейс

Загальна картина зв'язку між платформою та іншими об'єктами представлена на рисунку 3.8. Користувач керує датчиками через веб-інтерфейс, тоді як інші системи, такі як зовнішні системи або системи, з яких складається цифровий двійник, використовують REST API платформи. SCP не зберігає і не має доступу до даних, які передають вузли датчиків. Натомість дані надсилаються до окремого сховища даних, яке може бути зовнішньою системою або частиною цифрового двійника. Суб'єкти, зацікавлені в даних вимірювань, використовують інтерфейс сховища даних для отримання даних.

Далі для демонстрації роботи платформи покроково представлено процес оновлення налаштувань сенсорного вузла.

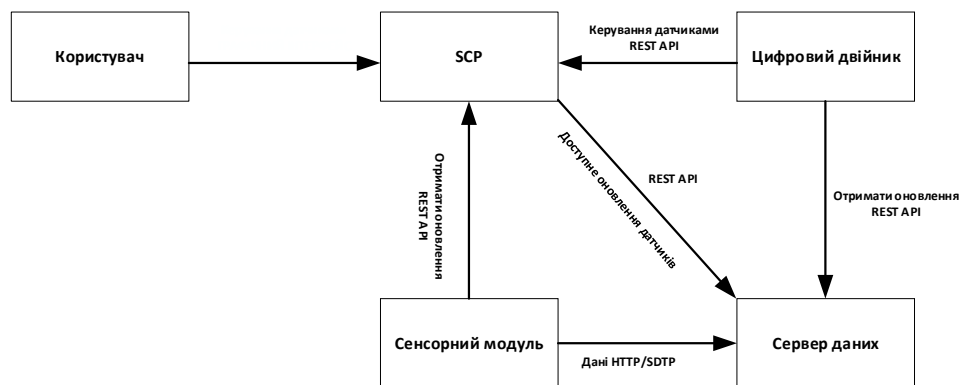


Рисунок 3.8 -Комунікація між платформою та іншими об'єктами

Спочатку змінюються налаштування сенсорного вузла через WUI або API і конфігуратор сенсорів генерує MicroPython-файл на основі цих налаштувань. Якщо використовується протокол SDTP (Sensor Data Transmission Protocol), то, інформація про доступне оновлення надсилається на сервер даних через REST API з використанням JWT-аутентифікації. Вузли датчиків можуть отримати інформацію про доступне оновлення двома способами:

- Датчик завжди запитує оновлення періодично (часовий проміжок вільно налаштовується) у конфігураторі датчиків за допомогою HTTP-запиту.
- Датчик отримує інформацію від сервера даних, після чого запитує оновлення з вказаної IP-адреси за допомогою HTTP-запиту.

Перший варіант не вимагає жодного зв'язку між платформою конфігуратора датчиків та сервером даних, що є одним з основних принципів побудови системи: платформа конфігуратора датчиків повинна працювати як незалежна система без жодних зв'язків з іншими системами.

Ключовими особливостями розробленої платформи є:

- Можливість змінювати налаштування сенсорних вузлів, включаючи налаштування вимірювань та передачі даних, віддалено через Інтернет
- Доступ до функціоналу через REST API
- Додавання нових сенсорних вузлів без кодування
- Архітектура платформи підтримує додавання нових типів датчиків, нових технологій зв'язку та протоколів прикладного рівня

										Арк.
										46
Змн.	Арк.	№ докум.	Підпис	Дата						

- Підтримка декількох протоколів прикладного рівня

Якщо на момент запиту оновлення є доступне оновлення, конфігуратор датчика відповідає HTTP-відповіддю, що містить згенерований файл MicroPython у складі корисного навантаження. Після того, як вузол датчика отримав новий файл MicroPython, файл записується до файлової системи і мікроконтролер перезавантажується. Файл boot.py, який запускається при кожному запуску, видаляє старий файл main.py і перейменовує новий файл на main.py. Після кожного завантаження сенсорний вузол запитує оновлення.

Коли запитується оновлення, платформі передається поточна версія програмного забезпечення. Маючи цю інформацію, платформа знає, чи отримав датчик останнє оновлення, і може змінити його статус з "Очікує на оновлення" або "Вимірювання, очікує на оновлення" на "Вимірювання".

В даному проекті вузол датчика може використовувати такі протоколи прикладного рівня для надсилання даних вимірювань, як: SDTP, HTTP та MQTT.

При використанні HTTP або MQTT дані кодуються у форматі JSON, як показано на рисунку 3.9, а при використанні SDTP дані надсилаються у бітовому форматі.

```
2 {
3   'Value1':41,
4   'Value2':-259,
5   'Value3':34,
6   'Timestamp':104491
7 },
8 {
9   'Value1':41,
10  'Value2':-261,
11  'Value3':35
12  'Timestamp':198691
13 },
```

Рисунок 3.9 Дані в форматі JSON, коли дані вимірювань надсилаються за допомогою HTTP.

Сховище даних відповідає за те, щоб зробити дані доступними для систем, з яких складається цифровий двійник, або для зовнішніх систем.

										Арк.
										47
Змн.	Арк.	№ докум.	Підпис	Дата						

ДП.КН.9500038.069.ПЗ

Реалізація сервера даних, що використовується в цьому проекті, пропонує REST API для доступу до даних. Однак розробка інтерфейсу для отримання даних покладається на розробника сервера даних.

Ключовою особливістю платформи конфігуратора сенсорів є можливість змінювати налаштування сенсорних вузлів віддалено через Інтернет. Загальні конфігураційні параметри наведені в таблиці 3.1.

Таблиця 3.1 - Конфігурація налаштувань сенсорного вузла.

Параметр	Опис/додаткова інформація	Приклад значення
IP-адреса сервера даних	Вузол датчика надсилає дані на цю IP-адресу	86.70.113.151:2500
Перевірка оновлення IP-адреси	Вузол датчика запитує оновлення з цієї IP-адреси	86.70.113.151:8000
Ліміт перевірки оновлення	Інтервал часу між послідовними перевірками оновлень	3600 (с)
Модель датчика	Датчики, які підтримуються на даний момент: ADXL345, LIS3DSH, Seeed Grove I2C ADC	ADXL345
Частота дискретизації	Максимальна частота дискретизації залежить від протоколу прикладного рівня	100 Hz

Продовження таблиці 3.1

Чутливість	Залежить від моделі датчика	+2g
Довжина серії	Тривалість пачки в секундах. Може бути встановлена на 0 для безперервного вимірювання	5.0 (с)
Частота серій	Інтервал часу між послідовними серіями	3.0 (с)
Швидкість передачі даних	Інтервал часу між передаванням даних на сервер даних. Якщо встановлено на 0, дані надсилаються миттєво тобто надсилаються негайно	10 (с)
Ліміт закриття з'єднання	Дозволяє налаштувати проміжок часу, через який з'єднання з сервером даних буде закрито, якщо не буде надіслано нових даних	3 (с)
Ліміт закриття мережі	Дозволяє налаштувати часовий проміжок, через який з'єднання з мережею мережею, якщо нові дані не надсилаються	30 (с)

Змн.	Арк.	№ докум.	Підпис	Дата

ДП.КН.9500038.069.ПЗ

Арк.

49

Таблиця 3.2 - Налаштування параметрів Wi-Fi.

Налаштування	Опис/додаткова інформація	Приклад значення
SSID	Ідентифікатор сервісного набору	TP_Link_Arch
Безпека	Параметри: Нічого, WEP, WPA, WPA2, WPA2_ENT	WPA2
Ім'я користувача	Необов'язково, залежить від налаштувань безпеки	test_user
Ключ	Необов'язково, залежить від налаштувань безпеки	пароль1

Протокол прикладного рівня також налаштовується. Наразі доступні три варіанти: SDTP, HTTP і MQTT. Для SDTP немає налаштувань, що налаштовуються, а для HTTP можна змінити шлях надсилання даних. MQTT має наступні параметри, які можна налаштувати, представлені в Таблиці 3.3.

Таблиця 3.3 - Параметри, що налаштовуються для MQTT.

Налаштування	Опис/додаткова інформація	Приклад значення
Користувач	Ім'я користувача для брокера	test_user

Продовження таблиці 3.3

Ключ	Токен, який використовується для аутентифікації	b401jdlfper3r59c2a6241dk258bf48
Тема	Тема, до якої публікуються дані	test_user/feeds/topic_name
Адреса сервера даних	Адреса брокера	io.adafruit.com
Порт	Використаний порт	1883

Сторінка адміністратора пропонує простий інтерфейс для керування користувачами (рисунок 3.10). Для автентифікації користувачів використовується вбудована система автентифікації Django.

Реалізовано три рівні автентифікації:

- Користувач, який може переглядати деталі вузлів датчиків
- Менеджер, який може додавати, змінювати та видаляти вузли датчиків
- Адміністратор, який може керувати користувачами та додавати нові типи сенсорів.



Рисунок 3.10 - Сторінка адміністратора

Окрім керування користувачами, сторінка адміністратора використовується для додавання нових типів датчиків. Далі цей процес представлено більш детально.

По-перше, в базі даних створюється новий екземпляр типу датчика. Цей екземпляр містить I2C-адресу, опис та додаткову функцію обробки даних у вигляді скрипту на мові python. Скрипт потрібен, якщо потрібні додаткові бітові операції, такі як зсув бітів, для перетворення даних вимірювань у бітовому форматі в числа.

Далі створюються об'єкти частоти дискретизації. Кожна частота дискретизації, що підтримується типом датчика, вимагає власного об'єкта частоти дискретизації, який потім прив'язується до типу датчика. Об'єкт частоти дискретизації містить пари значень запису, що складаються з адреси та значення, яке потрібно записати, які необхідні для конфігурації датчика, та пари значень зчитування, які складаються з адреси та кількості байт, що зчитуються з адреси, для зчитування даних. Ці пари значень можна знайти в інструкції до датчика.

Потім, створюються об'єкти чутливості, які працюють подібно до об'єктів частоти дискретизації. Метод на основі пар значень, що використовується для конфігурації датчика та зчитування даних, дозволяє повністю вільно додавати датчики без кодування, якщо не потрібні бітові операції.

Для зручного керування вузлами датчиків реалізовано веб-інтерфейс користувача. Інтерфейс користувача підтримує додавання, конфігурацію та видалення сенсорних вузлів. Крім того, він дозволяє переглядати сенсорні вузли, відстежувати їхні статуси та перевіряти поточні налаштування вузлів.

Для демонстраційних цілей було реалізовано декілька додаткових функцій інтерфейсу користувача. Ці функції можуть бути використані, якщо сенсорний вузол використовує SDTP як протокол прикладного рівня і надсилає дані на реалізований сервер даних. Ці додаткові можливості

									ДП.КН.9500038.069.ПЗ	Арк.
										52
Змн.	Арк.	№ докум.	Підпис	Дата						

включають візуалізацію даних (рисунок 3.11), перегляд даних у вигляді таблиці та завантаження даних у вигляді CSV-файлу.

REST API пропонує ті ж функції, що і веб-інтерфейс користувача та сторінка адміністратора, за винятком управління користувачами. Кореневе представлення API, яке генерується фреймворком Django REST, візуально показує доступні ресурси (рисунок 3.12).

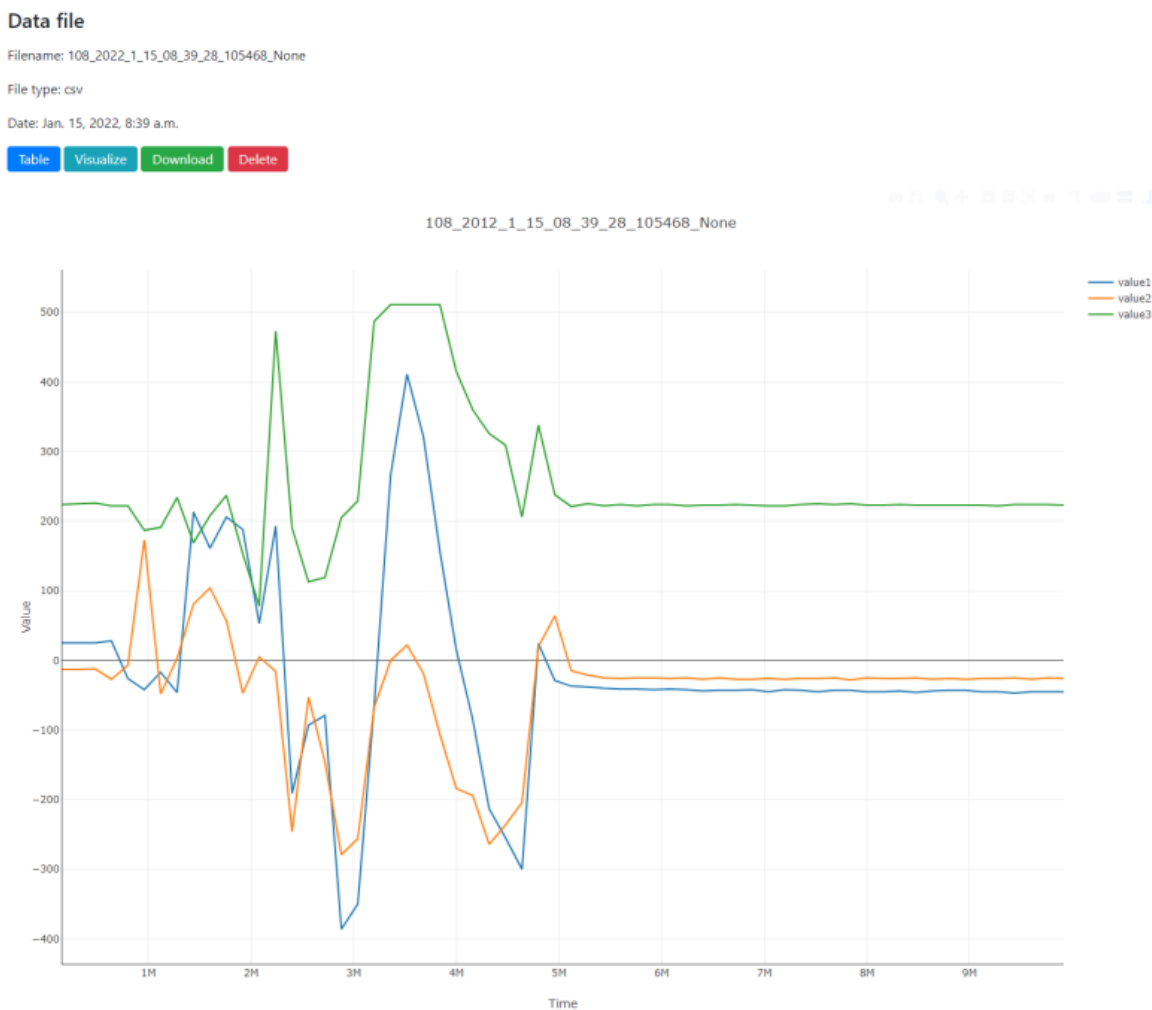


Рисунок 3.11 - Візуалізація даних трьохкоординатного акселерометра ADXL345

API дозволяє, наприклад, переглядати вузли датчиків (Рис. 32) та змінювати їх налаштування. Підтримувані методи автентифікації для API - базова автентифікація HTTP та автентифікація JSON Web Token (JWT).

```

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "sensors": "http://86.50.143.154:8000/api/v1.0/sensors/",
  "models": "http://86.50.143.154:8000/api/v1.0/models/",
  "wlans": "http://86.50.143.154:8000/api/v1.0/wlans/",
  "nb-iots": "http://86.50.143.154:8000/api/v1.0/nb-iots/",
  "http": "http://86.50.143.154:8000/api/v1.0/http/",
  "https": "http://86.50.143.154:8000/api/v1.0/https/",
  "lwdtp": "http://86.50.143.154:8000/api/v1.0/lwdtp/",
  "mqtt": "http://86.50.143.154:8000/api/v1.0/mqtt/",
  "samplerates": "http://86.50.143.154:8000/api/v1.0/samplerates/",
  "sensitivys": "http://86.50.143.154:8000/api/v1.0/sensitivys/",
  "valuepairs": "http://86.50.143.154:8000/api/v1.0/valuepairs/"
}

```

Рисунок 3.12 - Кореневе подання API, згенероване Django REST Framework

При базовій автентифікації HTTP для автентифікації користувача використовуються ім'я користувача та пароль. При використанні автентифікації JWT (JSON Web Token) спочатку сервер генерує рядок, який називається токен, і передає його клієнту. Після цього клієнт передає токен разом із запитом до API, який використовує токен для перевірки особи користувача.

```

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "sensor_id": 108,
    "url": "http://86.50.143.154:8000/api/v1.0/sensors/108/",
    "sensor_name": "Test1",
    "model": "http://86.50.143.154:8000/api/v1.0/models/ADXL345/",
    "status": 2,
    "description": "",
    "location": "",
    "sensor_key": "vTdD3bky9fXPHXVBqJjC",
    "sample_rate": "http://86.50.143.154:8000/api/v1.0/samplerates/9/",
    "sensitivity": "http://86.50.143.154:8000/api/v1.0/sensitivys/2/",
    "data_send_rate": 10.0,
    "burst_length": 2.1,
    "burst_rate": 2.2,
    "connection_close_limit": 3.0,
    "network_close_limit": 30.0,
    "update_check_limit": 60.0,
    "update_check_ip_address": "86.50.143.154:8000",
    "data_server_ip_address": "86.50.143.154:2500",
    "communication_object": "http://86.50.143.154:8000/api/v1.0/wlans/1/",
    "protocol_object": "http://86.50.143.154:8000/api/v1.0/lwdtp/2/"
  },
  {
    "sensor_id": 109,
    "url": "http://86.50.143.154:8000/api/v1.0/sensors/109/",
    "sensor_name": "test",
    "model": "http://86.50.143.154:8000/api/v1.0/models/seed%20Grove%20i2C%20ADC/"
  }
]

```

Рисунок 3.13 - Список датчиків, що надається API, візуалізованим Django REST Framework.

									Арк.
									54
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ				

Максимальна частота дискретизації була протестована з датчиками ADXL345 та I2C АЦП Seeed Grove. Швидкість передачі даних була встановлена на 10 секунд, а з'єднання закривалося після кожної передачі даних. Для обох типів датчиків і SDTP максимальна частота дискретизації становить 200 Гц. З HTTP максимальна частота дискретизації становить 12,5 Гц, що вказує на те, що обчислювальна потужність мікроконтролера обмежує максимальну частоту дискретизації. Максимальну частоту дискретизації з MQTT не вдалося виміряти, оскільки безкоштовна версія брокера обмежує частоту дискретизації.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

ВИСНОВКИ

Розроблена система є єдиною готовою до використання системою для набору пов'язаних систем, з яких складається цифровий двійник, і дозволяє передавати дані з фізичного двійника на цифровий. Вона дозволяє легко додавати нові сенсорні вузли до фізичного двійника, віддалено керувати ними та передавати дані з використанням найбільш підходящих стандартів.

Таким чином, система скорочує розрив між концепцією цифрового двійника та його реалізацією. Вона пропонує REST API, який інші підсистеми цифрового двійника можуть використовувати для віддаленого керування датчиками. Крім того, система має веб-інтерфейс, що дозволяє її використання людиною-оператором.

Оскільки система дозволяє легко додавати, конфігурувати та керувати сенсорними вузлами, її також можна використовувати поза контекстом цифрового двійника. Наприклад, вона дозволяє виконувати прості вимірювальні завдання в лабораторії і може бути використана в навчанні для демонстрації вимірювальної системи.

У студентських проєктах система дозволяє студентам зосередитися на обробці даних, а не на їх збиранні. У цьому розділі розглядаються майбутні напрямки розвитку платформи, а також те, чи відповідає розроблена платформа поставленим до неї вимогам. Крім того, на основі огляду літератури представлені рекомендації щодо найбільш підходящих протоколів прикладного рівня та комунікаційних технологій для передачі даних вимірювань від фізичного двійника до цифрового двійника.

Розроблена платформа конфігуратора датчиків скорочує розрив між концепцією цифрового двійника і його реалізацією, дозволяючи передавати дані датчиків. Передача забезпечується за рахунок простого додавання, конфігурації та управління вузлами датчиків.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Система відповідає всім вимогам, окрім підтримки DDS. DDS не було реалізовано через відсутність відповідної бібліотеки, а реалізація нової бібліотеки виходить за рамки цієї роботи.

Реалізація вимог:

- Підтримка декількох технологій зв'язку. Підтримка декількох технологій зв'язку була відкинута, оскільки Wi-Fi підходить для більшості випадків використання.

- Низьке енергоспоживання. Низьке енергоспоживання не вдалося перевірити, оскільки виникли проблеми з живленням мікроконтролерів.

Незважаючи на те, що платформа повністю функціональна, є кілька областей для вдосконалення. Наразі користувацьке управління платформою не дозволяє обмежити видимість сенсорних вузлів: кожен сенсорний вузол може бачити кожен користувач. По суті, це означає, що для зменшення видимості кожна група користувачів системи повинна мати власний екземпляр платформи, що працює на сервері.

Ще одним обмеженням є підтримка лише мікроконтролерів Русом. Однак мікроконтролери базуються на загальному ESP32, а згенерований MicroPython-файл має лише одну специфічну для Русом функціональність: миготіння світлодіодних індикаторів, що вказують на стан плати.

Однією з цілей подальшого розвитку є додавання підтримки декількох мікроконтролерів та можливість генерування коду іншими мовами, такими, як C та Arduino.

Крім того мікроконтролери використовують свою внутрішню пам'ять для зберігання даних, зчитаних з датчиків. Це обмежує часовий інтервал, який датчик може виміряти без пересилання даних на сервер даних, оскільки доступна пам'ять складає лише близько двох мегабайт. Втім, підтримка збереження даних на карту пам'яті може бути додана в майбутньому, оскільки плати розширення, що використовуються з мікроконтролерами, мають слот для карт пам'яті microSD.

					ДП.КН.9500038.069.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

Продуктивність процесора мікроконтролера обмежує частоту дискретизації. В даний час максимальна частота дискретизації становить близько 200 Гц для SDTP і 10 Гц для HTTP. Однак використовувані датчики здатні працювати з частотою дискретизації понад 1000 Гц. Як видно з частоти дискретизації HTTP, мікроконтролер не повинен виконувати ніякої обробки даних (в даному випадку, перетворення бітових даних в цілі та побудови JSON). Якщо потрібна більш висока швидкість передачі даних, наприклад, з HTTP, можна використовувати периферійні обчислення. За допомогою периферійних обчислень дані вимірювань можна надсилати як необроблені дані у бітовому форматі на шлюз, який може перетворити їх у формат JSON і переслати дані у вигляді HTTP-запиту на сервер даних.

Наразі система використовує базу даних SQLite, яка не підтримує одночасні операції запису.

Безпека, яка є важливим фактором у галузі, не була в центрі уваги при розробці системи. Дані надсилаються незашифрованими через протоколи HTTP, MQTT або SDTP, а датчики отримують свої оновлення без шифрування.

Оскільки система має бути придатною для різних сценаріїв використання, наступним кроком у розвитку платформи може бути додавання підтримки декількох протоколів прикладного рівня та комунікаційних технологій. Крім того, можуть бути додані інструменти бенчмаркінгу для оцінки протоколів.

Розроблена система вирізняється з-поміж інших IoT-платформ простотою використання, можливістю додавання нових датчиків без кодування та легкістю (зосередженість лише на управлінні сенсорними вузлами).

									Арк.
									58
Змн.	Арк.	№ докум.	Підпис	Дата	ДП.КН.9500038.069.ПЗ				

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Trauer, Jakob, Sebastian Schweigert-Recksiek, Carsten Engel, Karsten Spreitzer, and Markus Zimmermann. "What is a digital twin?—definitions and insights from an industrial case study in technical product development." In Proceedings of the Design Society: DESIGN Conference, vol. 1, pp. 757-766. Cambridge University Press, 2020.
2. Wagner, Anna, and Uwe Rüppel. "BPO: The building product ontology for assembled products." In Proceedings of the 7th Linked Data in Architecture and Construction workshop (LDAC 2019)', Lisbon, Portugal, p. 12. 2019.
3. Stavropoulos, Panagiotis, and Dimitris Mourtzis. "Digital twins in industry 4.0." In Design and operation of production networks for mass personalization in the era of cloud technology, pp. 277-316. Elsevier, 2022.
4. Kritzinger, Werner, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihm. "Digital Twin in manufacturing: A categorical literature review and classification." Ifac-PapersOnline 51, no. 11 (2018): 1016-1022.
5. Sudarsan, Rachuri, Steven J. Fenves, Ram D. Sriram, and Fujun Wang. "A product information modeling framework for product lifecycle management." Computer-aided design 37, no. 13 (2005): 1399-1411.
6. Sjarov, Martin, Tobias Lechler, Jonathan Fuchs, Matthias Brossog, Andreas Selmaier, Florian Faltus, Toni Donhauser, and Jörg Franke. "The digital twin concept in industry—a review and systematization." In 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1, pp. 1789-1796. IEEE, 2020.
7. Shi, Yuanming, Kai Yang, Tao Jiang, Jun Zhang, and Khaled B. Letaief. "Communication-efficient edge AI: Algorithms and systems." IEEE Communications Surveys & Tutorials 22, no. 4 (2020): 2167-2191.
8. Vathoopan, Milan, Maria Johny, Alois Zoitl, and Alois Knoll. "Modular fault ascription and corrective maintenance using a digital twin." IFAC-PapersOnLine 51, no. 11 (2018): 1041-1046.

9. Han, Yue, Dusit Niyato, Cyril Leung, Dong In Kim, Kun Zhu, Shaohan Feng, Xuemin Shen, and Chunyan Miao. "A dynamic hierarchical framework for iot-assisted digital twin synchronization in the metaverse." IEEE Internet of Things Journal 10, no. 1 (2022): 268-284.

10. Shafto, M., Conroy, M., Doyle, R. & Glaessgen, E. (2012) Modeling, Simulation, information Technology & Processing Roadmap. Technology Area 11. Available at: https://www.nasa.gov/sites/default/files/501321main_TA11-ID_rev4_NRCwTASR.pdf.

11. Schroeder, G. N., Steinmetz, C., Pereira, C. E. & Espindola, D. B. (2016) Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange. IFAC-PapersOnLine. Elsevier, 49(30), pp. 12–17. doi: 10.1016/J.IFACOL.2016.11.115. ISSN 2405-8963.

12. Brussel, H. Van, Bongaorts, L., Wyns, J., Valckenaers, P. & Van Ginderachter, T. (1999) A Conceptual Framework for Holonic Manufacturing: Identification of Manufacturing Holons. Journal of Manufacturing systems, 18(1), pp. 35–52.

13. Romero, David, Thorsten Wuest, Ramy Harik, and Klaus-Dieter Thoben. "Towards a cyber-physical PLM environment: The role of digital product models, intelligent products, digital twins, product avatars and digital shadows." IFAC-PapersOnLine 53, no. 2 (2020): 10911-10916.

14. Romero, David, Thorsten Wuest, Ramy Harik, and Klaus-Dieter Thoben. "Towards a cyber-physical PLM environment: The role of digital product models, intelligent products, digital twins, product avatars and digital shadows." IFAC-PapersOnLine 53, no. 2 (2020): 10911-10916.

15. Liu, Tong, Lun Tang, Weili Wang, Qianbin Chen, and Xiaoping Zeng. "Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network." IEEE Internet of Things Journal 9, no. 2 (2021): 1427-1444.

16. Tao, Fei, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, and Fangyuan Sui. "Digital twin-driven product design, manufacturing and service with

						ДП.КН.9500038.069.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			60

big data." The International Journal of Advanced Manufacturing Technology 94 (2018): 3563-3576.

17. Cai, Hongxia, Jiamin Zhu, and Wei Zhang. "Quality deviation control for aircraft using digital twin." Journal of Computing and Information Science in Engineering 21, no. 3 (2021).

18. Ye, Yumei, Qiang Yang, Fan Yang, Yanyan Huo, and Songhe Meng. "Digital twin for the structural health management of reusable spacecraft: a case study." Engineering Fracture Mechanics 234 (2020): 107076.

19. Hash, David, Raj Venkatapathy, Paul Wercinski, Don Ellerby, Mike Wright, and Alan Cassell. Presentation to Technology, Innovation & Engineering Committee NASA Advisory Council. No. ARC-E-DAA-TN60731. 2018.

20. Almalki, Abduladheem, David Downing, Bill Lozanovski, Rance Tino, Anton Du Plessis, Ma Qian, Milan Brandt, and Martin Leary. "A digital-twin methodology for the non-destructive certification of lattice structures." JOM 74, no. 4 (2022): 1784-1797.

21. The Steps in Creating a Digital Twin, URL:https://www.designnews.com/electronics-test/steps-creating-digital-twin?_x_tr_hist=true (дата звернення: 12.04.2023)

22. NodeRED nodes for modeling Digital Twins, URL:<https://flows.nodered.org/node/@digitaltwins/node-red-contrib-digital-twin> (дата звернення: 12.04.2023)

23. Чому OPC UA такий важливий, URL:<https://indusoft.com.ua/blog/2022/05/18/chomu-opc-ua-takij-vazhlivij/> (дата звернення: 12.04.2023)

24. Ooi, Boon-Yaik, Zan-Wai Kong, Wai-Kong Lee, Soung-Yue Liew, and Shervin Shirmohammadi. "A collaborative IoT-gateway architecture for reliable and cost effective measurements." IEEE Instrumentation & Measurement Magazine 22, no. 6 (2019): 11-17.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

25. Yang, Kan, and Xiaohua Jia. "Data storage auditing service in cloud computing: challenges, methods and opportunities." World Wide Web 15 (2012): 409-428.

26. Rafique, Wajid, Lianyong Qi, Ibrar Yaqoob, Muhammad Imran, Raihan Ur Rasool, and Wanchun Dou. "Complementing IoT services through software defined networking and edge computing: A comprehensive survey." IEEE Communications Surveys & Tutorials 22, no. 3 (2020): 1761-1804.

27. Dinculeană, Dan, and Xiaochun Cheng. "Vulnerabilities and limitations of MQTT protocol used between IoT devices." Applied Sciences 9, no. 5 (2019): 848.

28. Arvind, S., and V. Anantha Narayanan. "An overview of security in CoAP: attack and analysis." In 2019 5th international conference on advanced computing & communication systems (ICACCS), pp. 655-660. IEEE, 2019.

29. Prajapati, Ambar. "AMQP and beyond." In 2021 International Conference on Smart Applications, Communications and Networking (SmartNets), pp. 1-6. IEEE, 2021.

30. Farrell, Stephen. Low-power wide area network (LPWAN) overview. No. rfc8376. 2018.

31. Barb, Gordana, and Marius Ottesteanu. "4G/5G: A comparative study and overview on what to expect from 5G." In 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), pp. 37-40. IEEE, 2020.

32. Devan, P. Arun Mozhi, Fawnizu Azmadi Hussin, Rosdiazli Ibrahim, Kishore Bingi, and Farooq Ahmad Khanday. "A Survey on the application of WirelessHART for industrial process monitoring and control." Sensors 21, no. 15 (2021): 4951.

33. Rahman, Taibur, and Swarnendu Kumar Chakraborty. "Provisioning technical interoperability within zigbee and ble in iot environment." In 2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), pp. 1-4. IEEE, 2018.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

34. Ucuz, Derya. "Comparison of the IoT platform vendors, microsoft Azure, Amazon web services, and Google cloud, from users' perspectives." In 2020 8th international symposium on digital forensics and security (ISDFS), pp. 1-4. IEEE, 2020.

35. Naik, Seena, and E. Sudarshan. "Smart healthcare monitoring system using raspberry Pi on IoT platform." ARPJ Journal of Engineering and Applied Sciences 14, no. 4 (2019): 872-876.

36. RRNV, Mr Nissanka, Mr Jagath Wickramaratne, Mr Supunya Swarnakantha, and Ms Bhagyanie Chaturika. "Train Accident Prevention & Breakdown Detection using Machine Learning based Safty System."

37. Raspberry Pi Foundation, URL:<https://www.raspberrypi.org/about/> (дата звернення: 13.04.2023)

38. Raspberry Pi Accelerometer using the ADXL345, URL:<https://pimylifeup.com/raspberry-pi-accelerometer-adxl345/> (дата звернення: 13.04.2023)

39. HOW TO SETUP AN ACCELEROMETER ON THE RASPBERRY PI, URL:<https://www.circuitbasics.com/what-is-an-accelerometer/> (дата звернення : 13.04.2023)

40. Grove Base Hat for Raspberry Pi, URL:https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_Pi/ (дата звернення: 15.04.2023)

41. MicroPython, URL: <https://micropython.org/> (дата звернення: 15.04.2023)

42. Pycom, URL: <https://www.antratek.com/boards/pycom> (дата звернення: 15.04.2023)

43. pycom/Pyмаkr, URL:<https://github.com/pycom/Pyмаkr> (дата звернення: 15.04.2023)

44. About the Django Software Foundation, URL:<https://www.djangoproject.com/foundation/> (дата звернення: 15.04.2023)

45. Django REST framework, URL: <https://www.django-rest-framework.org/> (дата звернення: 15.04.2023)

46. Simple JWT, URL: https://django-rest-framework-simplejwt.readthedocs.io/en/latest/getting_started.html (дата звернення: 17.04.2023)

47. Bootstrap Raspberry Pi without a USB keyboard or mouse (2019), URL: <https://gist.github.com/atenni/1cabc375195c9abfb67fe8da93fae1c8> (дата звернення: 17.04.2023)

48. jQuery URL: <https://jquery.com/> (дата звернення: 17.04.2023)

49. Plotly JavaScript Open-Source Graphing Library, URL: <https://plotly.com/javascript/> (дата звернення: 17.04.2023)

50. JavaScript data grid with spreadsheet UX, URL: <https://handsontable.com/>

51. Adafruit IO, URL: <https://io.adafruit.com/> (дата звернення: 17.04.2023)

52. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. – Тернопіль: ЗУНУ, 2021. – 56 с.

					<i>ДП.КН.9500038.069.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64


```

109         data = s.recv(MAXLINE)
110         if data:
111             data_read += data
112         else:
113             break
114     s.close()
115     decoded_data = data_read.decode('ascii')
116     payload_begins = decoded_data.find("import") #Payload string always
117     #begins with an import statement
118     if payload_begins == -1:
119         if decoded_data.find("UP-TO-DATE") != -1:
120             print("Software up-to-date")
121             return
122         else:
123             print("There's an error with the server response")
124     else:
125         data = decoded_data[payload_begins:]
126         f = open("new_main.txt", "w")
127         f.write(data)
128         f.close()
129         print("Writing data succeed!")
130         #Confirm that data is received
131         s = socket.socket()
132         s.connect(addr)
133         content_length = len("sensor_id={}&sensor_key={}".format(SENSOR_ID,
134             SENSOR_KEY))
135         confirmation = """POST /confirm_update HTTP/1.1\r\nHost:
136             {}\r\nContent-Type:
137             application/x-www-form-urlencoded\r\nContent-Length:
138             {}\r\n\r\nsensor_id={}&sensor_key={}\r\n\r\n""".format(ip_address,
139                 content_length, SENSOR_ID, SENSOR_KEY)
140         p = s.send(bytes(confirmation, 'utf8'))
141         s.close()
142         print("update confirmed")
143         utime.sleep(1)
144         machine.reset()
145     except:
146         print("Software update failed.")
147     """Sends data to the given socket and returns total amount of bytes sent"""
148     def send_data(data, s, length):
149         bytes_sent = 0
150         for value_pair in data:
151             ts_packed = struct.pack("<L", value_pair[1])
152             sent = sender(s, value_pair[0] + ts_packed, length)
153             if sent == CONNECTION_BROKEN:
154                 return CONNECTION_BROKEN
155             else:
156                 bytes_sent += sent
157         return bytes_sent
158     """Connect to wifi"""
159     def connect_network(flash_light=True):
160         if flash_light:
161             pycom.heartbeat(False)
162             ssid, auth, identity = SETTINGS_DICT["NETWORK_SETTINGS"]
163             wlan = WLAN(mode=WLAN.STA)
164             wlan.connect(ssid=ssid, auth=auth, identity=identity)
165             while not wlan.isconnected():
166                 if flash_light:
167                     pycom.rgbled(0xCD7300) #Orange
168                     utime.sleep(0.5)
169                 if flash_light:
170                     pycom.rgbled(0x000000) # Black
171                     utime.sleep(0.5)
172         return wlan

```

Змн.	Арк.	№ докум.	Підпис	Дата

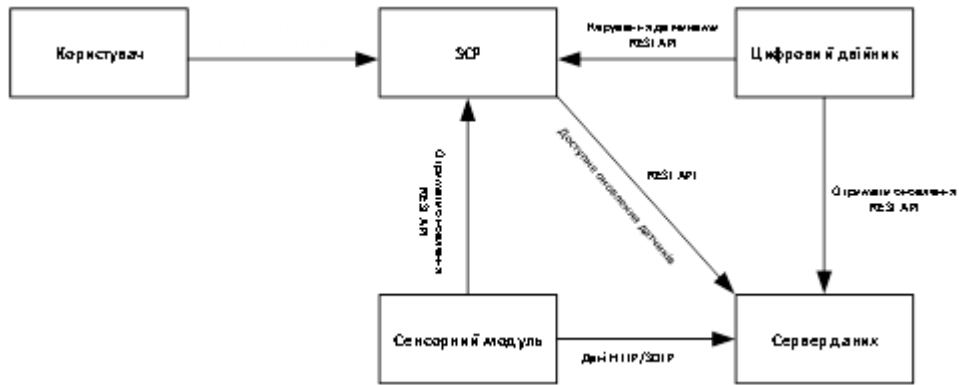
ДП.КН.9500038.069.ПЗ

Арк.

67

Додаток В

Комунікація між системою та іншими об'єктами



Змн.	Арк.	№ докум.	Підпис	Дата

ДП.КН.9500038.069.ПЗ

Арк.

69