МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
д-р техн. наук, проф.

_____ І. М. Журавська

« __ » _____ 2023 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Створення IoT-пристрою на основі лічильника Гейгера

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.ПЗ.00 – 405з.22020504

Студент _____В. Р. Могила

підпис

«__» _____ 202__ р.

Керівник ст. викладач _____ Б. Г. Салтовський

підпис

«__» _____ 202__ р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

«_____» _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної бакалаврської роботи

Видано студенту групи 405з факультету комп'ютерних наук

Могила Владислав Романович_____
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи
__ Створення IoT-пристрою на основі лічильника Гейгера

Затверджена наказом по ЧНУ ім. Петра Могили від ___.____.202__ № _____.

2. Строк представлення кваліфікаційної роботи «____» _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні
___Очікуваним результатом роботи є апаратне та програмне забезпечення IoT-пристрою на основі лічильника Гейгера для розбудування IoT-мережі моніторингу радіаційного фону. Вхідними даними роботи є специфікація вимог, що описує характеристики зазначеного апаратного та програмного забезпечення._____
_____
_____

4. Перелік питань, що підлягають розробці
1) огляд сучасних підходів та систем розробки IoT-пристроїв;____
2) аналіз переваг та недоліків існуючих систем;_____
3) розробка апаратної частини IoT-пристрою на основі лічильника Гейгера;_____
4) розробка програмної частини передачі даних з IoT-пристрою на сервер та на мобільний пристрій для попередження про перевищення радіаційного фону в даній місцевості_____

5. Перелік графічних матеріалів

_слайди презентації_____

6. Завдання до спеціальної частини

_____

_____

_____

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Спеціальна частина з охорони праці	Алексєєва Анна Олександрівна, канд. техн. наук, доцент кафедри екології Медичного інституту ЧНУ імені Петра Могили	

Керівник роботи

_старший викладач Салтовський Борис Григорович_____

(посада, прізвище, ім'я, по батькові)

_____

(підпис)

Завдання прийнято до виконання

_Могила Владислав Романович_____

(прізвище, ім'я, по батькові студента)

_____

(підпис)

Дата видачі завдання «____» _____ 20_____ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної бакалаврської роботи

Тема:_ Створення IoT-пристрою на основі лічильника Гейгера

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КБР	27.10.2022	27.10.2022	виконано
2.	Огляд літератури за темою роботи	25 .11.2022	09 .12.2022	виконано
3.	Складання календарного плану КБР	10. 12.2022	10. 12.2022	виконано
4.	Аналіз предметної області	01.03.2023	03.03.2023	виконано
5.	Розробка проєктних рішень	04.03.2023	09.03.2023	виконано
6.	Моделювання	10.03.2023	14.03.2023	виконано
7.	Конструювання АПЗ	15.03.2023	12.03.2023	виконано
8.	Перевірка працездатності, тестування та апробація розробленого АПЗ,	23.03.2023	23.03.2023	виконано
9.	Аналіз результатів тестування	02.06.2023	12.06.2023	виконано
10.	Попередній захист	13.06.2023	13.06.2023	виконано
11.	Розробка керівництва користувача	14.06.2023	18.06.2023	виконано
12.	Відгук керівника КБР	19.06.2023	19.06.2023	виконано
13.	Оформлення КБР та презентації	19.06.2023	20.06.2023	виконано
14.	Рецензування	20.06.2023	20.06.2023	виконано
15.	Завершення оформлення КБР та презентації	20.06.2023	21.06.2023	виконано
16.	Захист кваліфікаційної роботи	27.06.2023	27.06.2023	

Розробив здобувач ВО Могила Владислав Романович_____ _____

(*прізвище, ім'я, по батькові)* *(підпис)*

«__» _____ 20__ р.

Керівник роботи зав. каф. КІ Салтовський Борис Григорович__ _____

(підпис)

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Створення IoT-пристрою на основі лічильника Гейгера»

Студент гр. 405з Могила Владислав Романович

Керівник: старший викладач Салтовський Борис Григорович

Актуальність теми кваліфікаційної роботи обумовлена важливістю вимірювання рівня радіації для оцінки потенційних ризиків та впровадження відповідних заходів безпеки на таких об'єктах, як атомні електростанції, медичні заклади та промислові об'єкти, де працівники та населення можуть зазнати впливу радіації.

Об'єкт дослідження: процес дистанційного моніторингу та обробки даних радіаційного фону.

Предмет дослідження: розгалужена у просторі мережа станцій радіаційного контролю на базі електронних сенсорних модулів.

Мета: побудування архітектури мережі станцій моніторингу, які будуть надавати інформацію про рівень радіаційного фону в режимі реального часу за допомогою створених IoT-пристроїв на основі лічильника Гейгера.

Для досягнення поставленої мети було поставлено такі завдання:

1) проаналізувати аналоги пристроїв вимірювання радіаційного фону;

2) створити прототип IoT-пристрою на основі типового модуля лічильника Гейгера, сумісного з Arduino, що придатний здійснювати автоматизований радіаційний моніторинг навколишнього середовища;

3) розробити мобільний застосунок, що зможе інформувати про підвищення рівня радіації в певних районах та передавати дані на сервер.

Кваліфікаційна робота містить: перелік скорочень, вступ, три розділи, висновок, перелік джерел посилання та два додатка.

В спеціальній частині з охорони праці розглянуто забезпечення вимог охорони праці на робочому місці.

Кваліфікаційна робота містить 61 сторінку (без додатків), 22 рис., 12 табл., перелік джерел посилання з 18 джерел, 2 додатки.

Ключові слова: *радіоактивний фон, лічильник Гейгера, мережеве сховище, віддалений доступ, протокол MQTT*

ASTRACT

of the Bachelor's Thesis

"Creating an IoT device based on a Geiger counter"

Student: Mohyla Vladyslav Romanovych

Supervisor: Senior Lecturer Saltovskyi Borys Hryhorovych

The relevance of the topic of the qualification work is due to the importance of measuring the level of radiation for assessing potential risks and implementing appropriate safety measures at facilities such as nuclear power plants, medical facilities and industrial facilities where workers and the population may be exposed to radiation.

The object of research: the process of remote monitoring and data processing of the radiation background.

The object of the research: a space-wide network of radiation control stations based on electronic sensor modules.

The main aim of the research: building the architecture of a network of monitoring stations that will provide information on the radiation background level in real time using the created IoT devices based on the Geiger counter.

To achieve the goal, the following tasks were set:

1) to analyze analogues of radiation background measurement devices;

2) to create a prototype of an IoT device based on a typical Arduino-compatible Geiger counter module suitable for automated environmental radiation monitoring;

3) to develop a mobile application that will be able to inform about the increase in the level of radiation in certain areas and transfer data to the server analyze modern household appliance control systems.

The qualification work contains a list of abbreviations, an introduction, three chapters, a conclusion, a list of references and two appendices.

The special part on labor protection considers the provision of labor protection requirements at the workplace.

The qualification work contains 61 pages (without appendices), 22 fig., 12 tabl., 18 references, 2 appendices.

Keywords: *radioactive background, Geiger counter, network storage, remote access, MQTT protocol*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

АПЗ – апаратно-програмне забезпечення

БД – база даних

ОС – операційна система

ПЗ – програмне забезпечення

ШІ – штучний інтелект

API – Application Programming Interface

ANN – Artificial Neural Networks

AVS – Alexa Voice Service

CASE – Computer-Aided Software Engineering

CNNs – Convolutional Neural Networks

CPU – Central Processing Unit

GCSM – Geiger counter sensor module

GPU – Graphics Processing Unit

IFTTT – If This Then That

IoT – Internet of Things

HTTPS – Hyper Text Transfer Protocol Secure

MQTT – Message Queue Telemetry Transport

RIMNET – Radioactive Incident Monitoring Network

ВСТУП

Зараз існує багато загроз радіоактивного характеру пов’язаного з російською агресією на території України. Це можливість ракетних атак на атомні станції, безвідповідальні дії на захопленій Запорізькій атомній станції, погрози застосування тактичної ядерної зброї [1]. Тому необхідність віддаленого збору та аналізу інформації про радіаційний фон є дуже важливою.

Зазначений збір інформації можливо здійснити завдяки використанню технології Інтернету речей (англ. Internet of Things, IoT) – сполучення кіберінфраструктури з реальним світом. Кіберінфраструктура включає в себе компоненти інформаційних технологій, такі як зберігання даних, хмарні сервіси, операційні системи, програмне забезпечення, мережеві технології, сервіси резервного копіювання, моніторинг та механізми безпеки, а також процеси автентифікації, авторизації та аудиту. Фізична інфраструктура складається з пристроїв та датчиків радіаційного фону різного типу, а також з систем управління, які забезпечують їх ефективну роботу.

Об'єкт дослідження: процес дистанційного моніторингу та обробки даних радіаційного фону.

Предмет дослідження: мережа дистанційних станцій радіаційного контролю на базі електронних сенсорних модулів.

Мета: створення IoT-пристрою на основі лічильника Гейгера як основи для побудування мережі станцій моніторингу, які будуть надавати інформацію про рівень радіаційного фону в режимі реального часу.

Для досягнення поставленої мети в роботі ставляться такі основні **завдання:**

– проаналізувати аналоги пристроїв вимірювання радіаційного фону;

– створити прототип IoT-пристрою на основі типового модуля лічильника Гейгера, сумісного з Arduino, що придатний здійснювати автоматизований радіаційний моніторинг навколишнього середовища;

– розробити мобільний застосунок, що зможе інформувати про підвищення рівня радіації в певних районах та передавати дані на сервер.

Методом дослідження є іонізаційний метод радіометрії та радіаційного контролю з використанням газорозрядних приладів (лічильника Гейгера).

Практична значимість результатів роботи полягає в

Робота пройшла **апробацію** на Міжнародному конкурсі студентських наукових робіт “Black Sea Science 2023” зі спеціальності «Інформаційні технології, автоматизація і робототехніка» (Одеський національний технологічний університет, березень 2023 р.), де зайняла 2-ге місце (додаток В).

За результатами кваліфікаційної роботи написано статтю, яку прийнято до **опублікування** у журналі категорії Б «Автоматизація технологічних і бізнес-процесів» [1].

1 ПРЕДМЕТНА ОБЛАСТЬ ТЕХНОЛОГІЇ IOT

Основною метою пристроїв, які з'єднані і об'єднані в одну систему, є забезпечення якості та безпеки повсякденного життя. Саме такими пристроями є система Інтернету речей (IoT), яка складається з взаємопов'язаних обчислювальних пристроїв, механічних і цифрових машин, об'єктів, тварин і людей. Кожен з них отримує унікальний ідентифікатор і має змогу обмінюватися даними в мережі [2].

1.1 Загальний огляд технології IoT

Поняття "Інтернет речей" (Internet of Things, IoT) було вперше згадане в 1990-х роках. Актуальний термін був запропонований Кевіном Аштоном у 1999 році [3]. За даними Gartner, індустріальний Інтернет складається з фізичних пристроїв, які можуть контролювати оточуюче середовище, передавати моніторингові дані іншим пристроям та здійснювати дії на основі отриманих даних.

Іншими словами, поняття "Інтернет речей" відноситься до з'єднаних пристроїв, які використовують різні методи і канали зв'язку. Ці пристрої можуть передавати та обмінюватися даними між собою. Засоби зв'язку та методи передачі можуть бути бездротовими або провідними, залежно від призначення та типу пристрою IoT. За даними різних наукових дослідницьких інститутів, таких як Gartner (2017), до 2020 року в світі буде встановлено до 25 мільярдів пристроїв IoT[4]. Проте, ця кількість може бути значно вищою.

Міжнародний союз електрозв'язку (ITU) описує IoT як глобальну інфраструктуру інформаційного суспільства, яка дозволяє взаємодію фізичних та віртуальних активів і речей за допомогою засобів зв'язку та технологій взаємодії. У своїй праці "Безпека IoT для початківців", Лоуренс Міллер дає таке визначення Інтернету речей: Інтернет речей включає в себе різні пристрої і об'єкти, які з'єднані між собою за допомогою різних протоколів зв'язку [5]. В

якості таких пристроїв можуть виступати комп'ютери, ноутбуки, персональні комп'ютери, планшети і смартфони. Зв'язок в Інтернеті речей зазвичай здійснюється через канали, такі як Bluetooth і Long Range Wide Area а також мобільний зв'язок 3G і 4G. Ці канали зазвичай використовуються для передачі невеликої кількості даних зі зниженою швидкістю передачі. В майбутньому пристрої Інтернету речей зможуть спілкуватися на великі відстані за допомогою передових технологій, таких як мобільні мережі п'ятого покоління (5G).

Як було вказано раніше, до 2020 року у світі можуть працювати до 25 мільярдів пристроїв Інтернету речей (IoT). Однак через впровадження новітніх розробок ця кількість підключених пристроїв може ще збільшитись. Раніше спостерігалась тенденція до зростання на 31% щороку. В даний час споживчий сегмент становить 63% від загальної кількості пристроїв IoT.

У майбутньому розвитку IoT також треба враховувати тенденцію до зниження ринкової вартості пристроїв IoT.

Варто відзначити, що, незважаючи на те, що багато пристроїв IoT придбують приватні споживачі, більше інвестицій в Інтернет речей здійснюється з боку бізнесу. Наприклад, у 2017 році 57% витрат на ринок IoT походили від бізнес-сектору [6]. Таблиця 1.1 показує вартість ринку IoT з 2019 по 2022 рік.

Таблиця 1.1 – Ринкова вартість приладів IoT

дол. США

Категорія	**2019 рік**	**2020 рік**	**2021 рік**	**2022 рік**
Приватні споживачі	532 515	725 696	985 348	1 494 466
Міжгалузеві підприємства	212 069	280 059	372 989	567 659
Спеціалізовані підприємства	634 921	683 817	736 543	863 662
Загальна сума	1 379 505	1 689 572	2 094 881	2 925 787

Економічний вплив Інтернету речей (IoT) оцінюється в трильйони, а кількість пристроїв IoT підраховується у мільярдах.

Зловмисники, що мають негативні наміри або прагнуть викрасти інформацію, постійно розробляють нові методи отримання несанкціонованого доступу, вторгнень до систем та збирання даних. Наслідки цих дій можуть бути нищівними, а витрати на відновлення — величезними. Протягом останніх двох років економічні збитки від кіберзлочинності перевищили трильйон доларів.

1.2 Системи радіаційного контролю

Зазвичай вимірювання радіації дозиметром виконується одноразово, на запит, та не набуває регулярності. Завдяки бездротовому зв’язку показники з таких приладів можна переглядати за декілька метрів [3]. В той же час радіаційний фон може змінюватись у поточній діяльності, наприклад, після виконання ремонтних робіт (насип покриття з гравію, оздоблення газовими лампами, гранітними панелями/стільницями тощо), обстеження на спіральній комп’ютерній томографії, при вдиханні радіоактивного аерозолю й т. і. Після поїздок через зони зі значним радіаційним забрудненням (нп., Чорнобильську зону відчуження) повітряні фільтри двигунів і салонів у автомобілях можуть затримувати радіоактивний пил та переносити його на великі відстані.

Ефективним у такому разі треба визнати розгортання детекторних мереж для вимірювання характеристик випромінювання в режимі реального часу. Нп., такою є європейська мережа MPX ATLAS, яка працює останні 20 років [4]. Але зазначене обладнання призначене для суто наукових досліджень та не відстежує радіаційний фон для потреб захисту населення.

В Україні питання щодо необхідності створення автоматизованої системи радіаційного контролю як основної складової радіаційної безпеки населення обговорюється вже більше 10 років [5]. Але проблема не вийшла за межі вирішення організаційних питань, на теперішній час така система не

функціонує. В даний час на території України діють лише відомчі автоматизовані системи контролю радіаційної обстановки, розташовані навколо АЕС. В той же час в деяких країнах світу вже функціонують аналогічні системи.

Так, у Фінляндії система радіаційного контролю складається з 290 станцій, що рівномірно розташовані по всій території країни (рис. 1.1). Результати вимірів записуються до Національного банку даних. Ця інформація доступна органам влади у режимі реального часу. Автоматизована система також отримує інформацію з інших скандинавських країн [6].

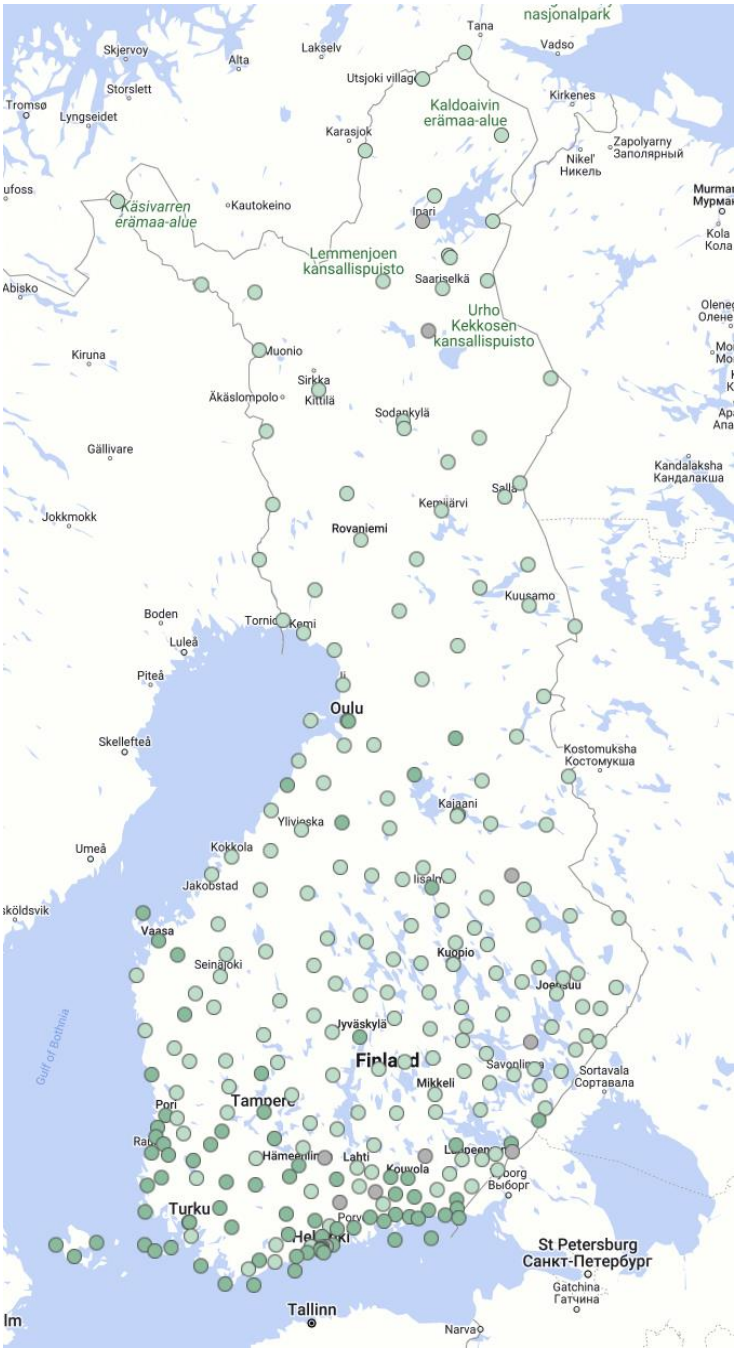


Рисунок 1.1 – Мережа радіаційного моніторингу STUK, Фінляндія

У Великій Британії Національна мережа радіаційного моніторингу та аварійного регулювання (RIMNET) введена в експлуатацію у 1988 р. для відстеження впливу на країну закордонних ядерних інцидентів [7]. RIMNET складається з 94 постів у всій країні (рис. 1.2), отримані дані зберігаються у Національній ядерній базі даних (UK National Nuclear Database).

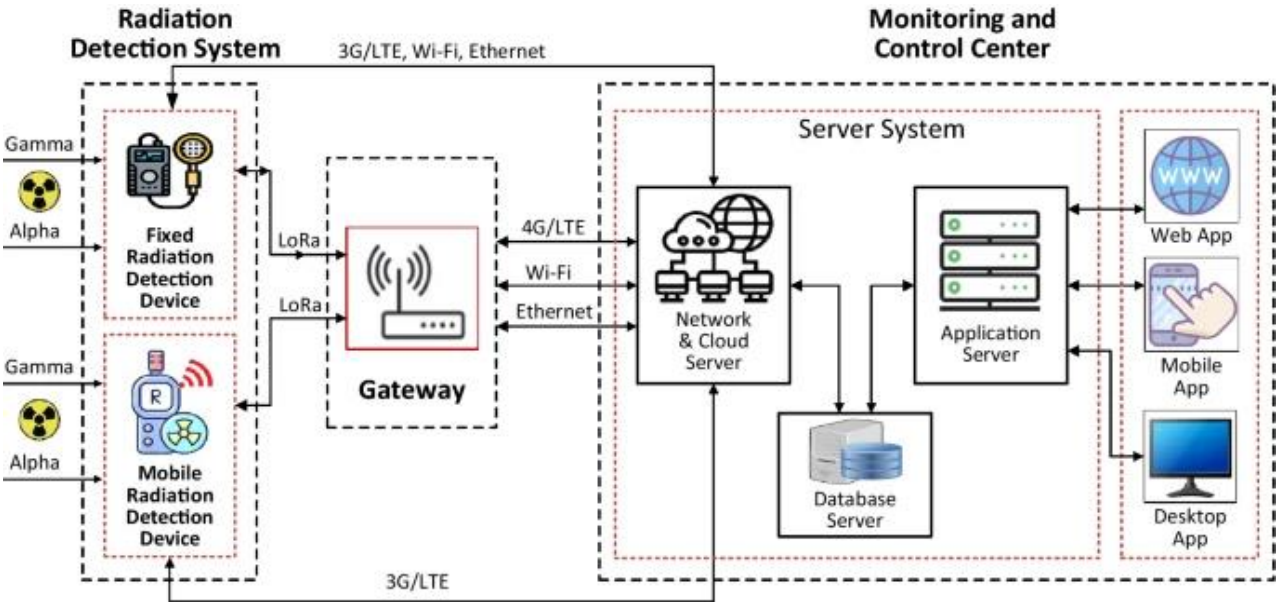


Рисунок 1.2 – Архітектура мережі радіаційного моніторингу RIMNET

Аналогічні комп'ютеризовані системи для автоматичного визначення радіаційної ситуації створені також у Болгарії, Бельгії, Японії та інших країнах.

Тому актуальними є дослідження в галузі створення малогабаритних бюджетних пристроїв, які можуть здійснювати моніторинг радіаційного фону певної території та у режимі реального часу надавати інформацію місцевим органам влади, відповідальним за безпеку населення безпосередньо населених пунктів.

Висновки за розділом 1

Інтернет речей, як і будь-яка технологія, що стрімко розвивається, може використовуватись для вирішення проблем безпеки, у т. ч. своєчасного вжиття заходів убезпечення людей під час підвищення радіаційного фону. У багатьох країнах світу запроваджені урядові системи моніторингу, за даними яких вживаються необхідні заходи щодо убезпечення або евакуації населення.

Але отриманими в якійсь місцевості даними не можна характеризувати радіаційний фон в іншій місцевості. Тому актуальним є створення малогабаритних бюджетних IoT-пристроїв, які можуть здійснювати моніторинг радіаційного фону невеликої, наприклад, приватної території та у режимі

реального часу передавати такі дані до центрального сервера для формування узагальненої картини.

Але чим більше «розумних» пристроїв підключається до мережі, тим вище ризики, пов'язані з несанкціонованим доступом в IoT-систему і використанням її можливостей зловмисниками. Тому треба спрямовувати зусилля на пошук рішень, які дадуть змогу мінімізувати загрози, що гальмують повноцінне впровадження IoT, та розвивати створення IoT-пристроїв, у т. ч. для моніторингу радіаційного фону середовища.

2 ПОБУДОВА МОДЕЛІ ІНФОРМАЦІЙНИХ РЕСУРСІВ ІОТ НА ОСНОВІ ЛІЧИЛЬНИКА ГЕЙГЕРА ТА АНАЛІЗ РЕЗУЛЬТАТІВ

У даному дослідженні, яке послужило основою для моделювання, увага була зосереджена виключно на вимогах безпеки внутрішніх елементів пристроїв Інтернету речей (IoT) та захищеного взаємодії між ними. Розробка безпечного програмного коду вийшла за межі цієї роботи. Зрозуміло, що багато з цих пристроїв часто взаємодіють з традиційними бекенд-системами, які функціонують у приватних центрах обробки даних або хмарних сервісах.

Припускалося, що безпека цих систем забезпечена на належному рівні. Однак, важливо пам'ятати, що якщо традиційні IT-системи використовують IoT-пристрої або обробляють дані від них, небезпечна взаємодія між IoT та традиційними IT-системами може повністю підірвати всю безпеку, яка була вбудована в систему IoT.

2.1 Модель найбільш поширеної архітектури IoT

У цьому розділі буде розглянута найпоширеніша архітектура Інтернету речей (IoT), яка об'єднує дві, на перший погляд, несумісні складові: з одного боку - велика кількість периферійних пристроїв з обмеженою обчислювальною потужністю, низьким енергоспоживанням та швидкою реакцією на події, а з іншого боку - хмарні сервери з високою обчислювальною потужністю для обробки великого обсягу даних, їх зберігання та класифікації, часто з використанням машинного інтелекту і аналітики. Ці два світи мають абсолютно відмінні принципи побудови та внутрішньої архітектури.

З погляду архітектури, основні пристрої або пристрої Інтернету речей (IoT) зазвичай мають такі складові:

1) Датчик або інший смарт-об'єкт, який здійснює збір даних.

2) Транспортний шар, який може бути повітряним (бездротовим) або провідним, використовується для передачі даних.

3) Комутатори або маршрутизатори, які забезпечують маршрутизацію та пересилання даних між різними пристроями та мережами.

4) Сервер збору даних, який займається зберіганням, обробкою та аналізом отриманих від пристроїв даних.

У середовищі Інтернету речей (IoT), основними модулями, як правило, є наступні компоненти: датчики для збору вимірюваних даних, транспортний рівень для передачі даних від датчиків, обчислювальні пристрої і програмні додатки для аналізу та обробки даних, а також для зберігання даних. На рис. 2.1 зображена архітектура IoT-середовища, яка використовує лічильник Гейгера як основу для побудови мережі станцій моніторингу.

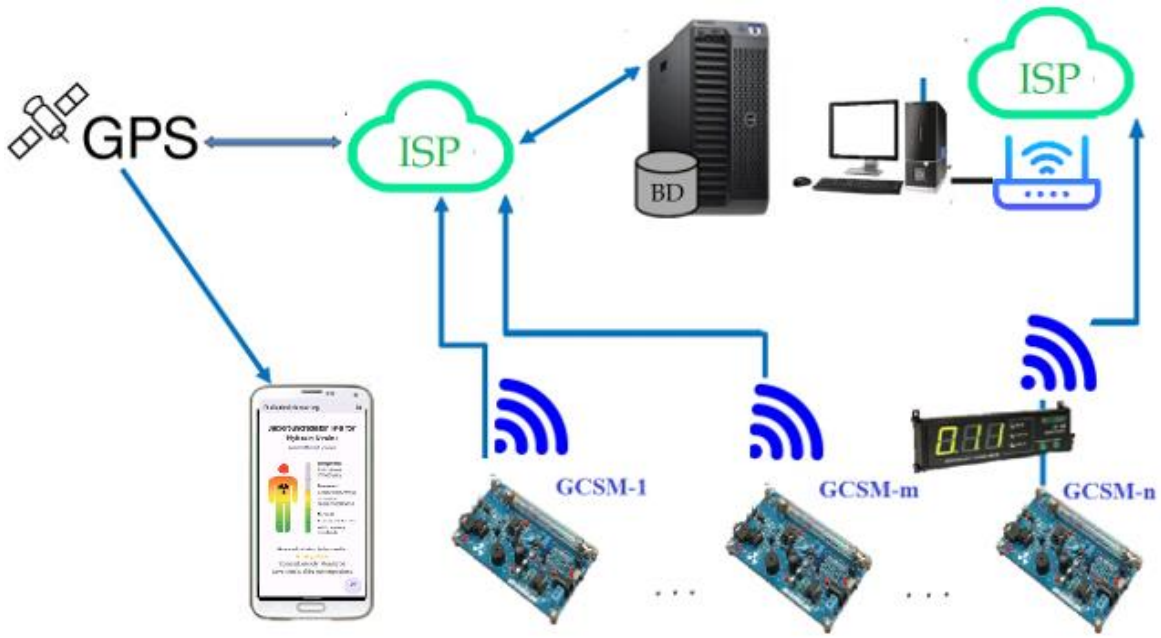


Рисунок 2.1 – Архітектура та функціональний огляд системи IoT

Від багаточисельних Geiger counter sensor module (GCSM) інформація у режимі реального часу надходить через Wi-Fi модулі та Internet Service Providers (ISP) до бази даних (BD) Центру прийняття рішень. Вказана BD встановлена на мережевому сховищі (див. рис. 2.7), яке також доступне віддалено. Оброблені дані доступні через мережу операторів мобільного зв’язку користувачам, які встановили відповідний застосунок моніторингу радіаційного

фону місцевості, обраної в зазначеному застосунку або за геолокацією смартфону (якщо вона включена).

Як видно з вищезазначеного, важливими компонентами є не лише датчики або розумні об'єкти та сервери для збору даних, але й передача даних та мережа, які відіграють важливу роль у концепції архітектури IoT. Мережа разом з використовуваним протоколом є ключовими складовими обладнання IoT. Докладніше основні компоненти архітектури IoT описані нижче. Якщо розглядати сам пристрій IoT і виключити зовнішні компоненти, такі як канали передачі, мережеве обладнання (наприклад, маршрутизатори) і сервери аналізу даних, внутрішня архітектура спрощеного IoT-пристрою може бути подана, як показано на рис. 2.2.

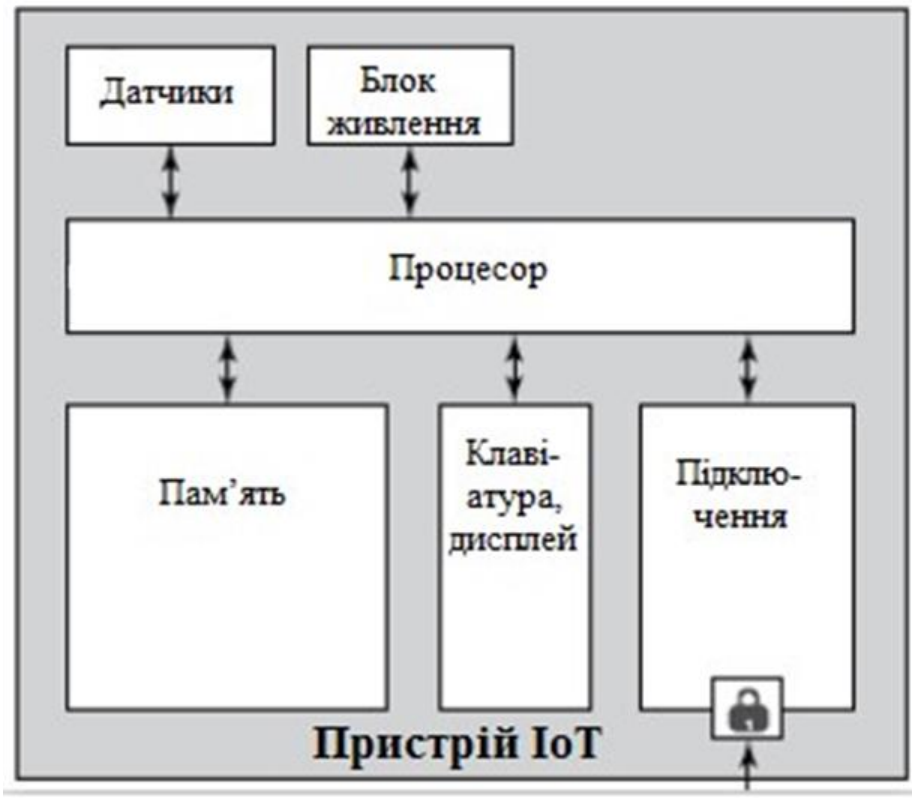


Рисунок 2.2 – Спрощена внутрішня архітектура IoT-пристрою

Архітектура пристроїв IoT має певну схожість зі звичайними комп'ютерами. Як видно, пристрій IoT включає головний процесор, який відповідає за контроль та управління ним. Пам'ять використовується для

зберігання даних з датчиків та програмного коду для виконання функцій. Пристрої введення/виведення, такі як клавіатура та монітор, призначені для користувацького інтерфейсу. Підключення до мережі здійснюється за допомогою провідних або бездротових з'єднань.

У разі бездротової мережі існує кілька способів реалізації, таких як Wi-Fi, Ethernet, Bluetooth або мобільний зв'язок через GSM/4G. Джерело живлення забезпечує необхідну потужність системи. Датчики виконують функцію збору даних, наприклад, температури навколишнього середовища. Усі ці модулі та компоненти є основною частиною стандартної комп'ютерної архітектури.

Основна відмінність між пристроями IoT та звичайними комп'ютерами полягає в наявності датчиків, які використовуються для збору інформації про навколишнє середовище. Крім цього, однією з ключових відмінностей між IoT та звичайними комп'ютерами є їх розмір. Зазвичай, пристрої IoT значно менші, а також мають нижчу споживчу потужність, ніж звичайні комп'ютери. Багато пристроїв IoT можуть працювати на невеликих акумуляторах або змінних батареях.

Датчики. Головна функція датчиків полягає у зборі вимірюваних даних про навколишнє середовище. Наприклад, датчик на основі лічильника Гейгера є електронним компонентом, який перетворює рівень радіаційного фону на електронний формат та передає ці дані на комп'ютер.

Ціна датчиків постійно знижується протягом останніх років, як показано на рис. 2.3, і очікується, що цей тренд зниження цін продовжиться у майбутньому. Це робить датчики ще більш доступними для використання в масштабному обсязі.

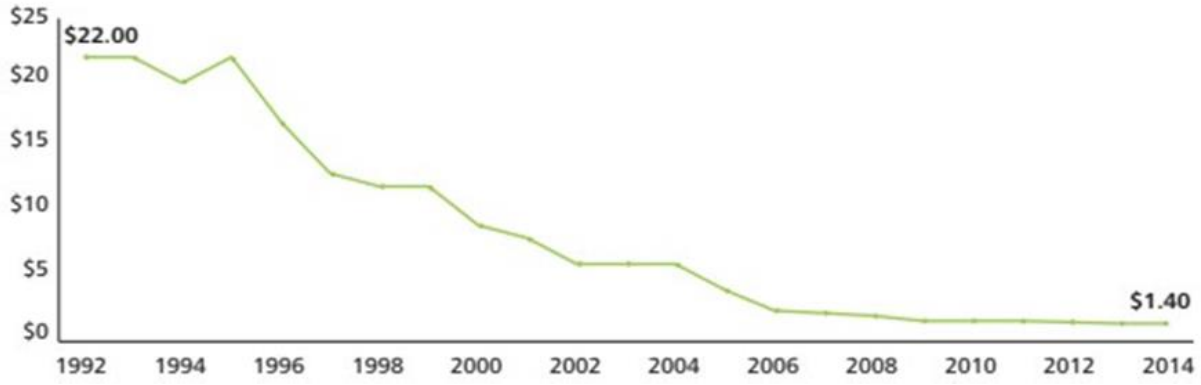


Рисунок 2.3 – Зменшення вартості датчиків за попередні 25 років

Сервери та сховища. З архітектурної точки зору, сервери в середовищі IoT виконують кілька ролей. Вони забезпечують процеси, починаючи з запуску програмного коду до керування необхідними програмними застосунками IoT. Пристрою IoT можуть знадобитись різні програми для виконання його функцій. Сервери застосунків можуть бути апаратними або програмними. Одним з ключових завдань серверів є керування та обробка даних від датчиків. У цьому контексті часто згадується термін "великі дані". Застосунки та аналітика великих даних будуть більш детально описані в наступному пункті.

Ефективне управління оновленнями різних компонентів в середовищі IoT є важливим аспектом. Середовище IoT складається з різних апаратних модулів, кожен з яких має власну прошивку. Протягом життєвого циклу пристрою IoT ці модулі повинні оновлюватися, тому в середовищі IoT може бути наявний спеціальний сервер оновлень. Це також стосується програмних модулів. Сервери відіграють важливу роль у зберіганні та керуванні необробленими даними, які генеруються різними датчиками. Обсяг даних, що надходить від датчиків, може бути значним, тому просте зберігання даних лише на сервері не є практичним. З цієї причини розумно мати окреме сховище даних, наприклад, локальний диск, який підключений по оптоволоконному каналу.

Один з можливих варіантів включає використання хмарної системи зберігання даних від відомого постачальника хмарних послуг. Однак, серед ключових функцій сервера є також автентифікація користувача, що гарантує

доступ до застосунків IoT та керування даними IoT лише для правомірних користувачів. Існує кілька варіантів управління процесом автентифікації користувача. Один з варіантів - це управління користувачами та можливими ключами безпеки на самому пристрої IoT або використання спеціального серверу керування або додатку для цих цілей. Можливості віддаленого доступу та автентифікації користувачів також повинні бути оброблені в межах сервера керування користувачами та програмами.

Великі дані та аналітичні програми. Одним з основних компонентів в архітектурі IoT є застосунки для обробки великих обсягів даних та аналітики. На сьогоднішній день існує ряд постачальників, які пропонують програми для управління великим обсягом даних та маніпулювання ними, зробивши їх більш зрозумілими для людей. Об'єм даних, що генеруються пристроями IoT, є величезним. Проте, величезна кількість даних в основному не має сенсу, поки вона не буде проаналізована і представлена таким чином, що людина може зрозуміти її значення. В найгіршому випадку, дані з датчиків можуть бути просто безладним текстом без будь-якого змісту.

Завдання програм для великих даних полягає в тому, щоб допомогти користувачам розуміти, які дані містяться інформації та, що ще важливіше, допомогти визначити, які дані є значущими для задоволення потреб організації. Можна сказати, що дані майже не мають значення без аналізу. Однією з ключових функцій у великих інструментах для обробки даних є також зберігання даних та забезпечення їх безпеки. На рис. 2.4 показано процес потоку даних від сенсора до програми аналітики, включаючи проміжні етапи.

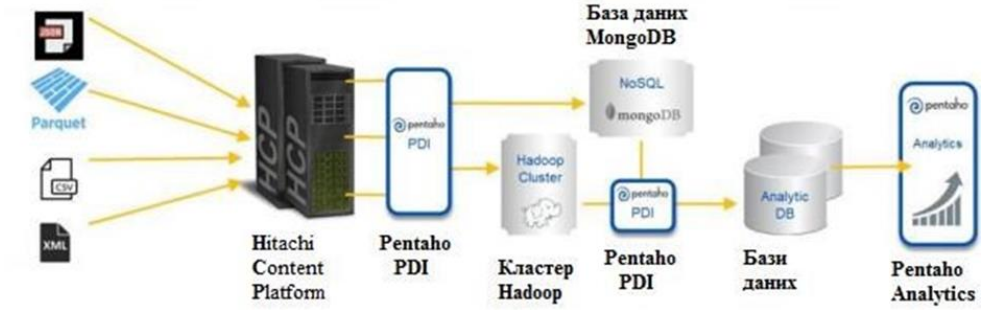


Рисунок 2.4 – Процес переходу даних

Вищезазначений приклад ілюструє типовий потік даних від датчиків та прикладних модулів, не включаючи самі датчики та мережеві модулі. Цей приклад базується на рішенні компанії Hitachi Vantara, яке використовується для обробки великих обсягів даних та аналітики згідно з даними на 2020 рік.

Потік даних рухається зліва направо. Спочатку незапроваджені дані з датчиків або неструктуровані/напівструктуровані дані збираються з різних джерел і потрапляють до централізованого сховища, в даному випадку Hitachi Content Platform (HCP).

Після збору дані піддаються перетворенню, щоб їх можна було використовувати з інструментами Pentaho PDI (Pentaho Data Integration). Pentaho PDI передає оновлені дані до Hadoop-кластера та бази даних MongoDB для зберігання. Інструменти аналітики Pentaho отримують ці дані, проводять зміни та форматують їх таким чином, щоб їх можна було легко сприймати, наприклад у вигляді графіків або діаграм.

Важливо зазначити, що цей приклад базується на рішенні, розробленому компанією Hitachi Vantara на 2020 рік, і деталі про потік даних можуть змінюватися залежно від конкретних потреб і використаної технології.

Технології та протоколи. очікується, що він революціонізуватиме взаємодію індивідів та корпорацій з цифровим і фізичним світом. В майбутньому IoT стане неот'ємною частиною повсякденного життя для кожної людини, розширюючи комунікаційні та мережеві можливості фізичних об'єктів та розумних пристроїв.

Однак лише з останнім прогресом і прийняттям технологій, таких як радіочастотна ідентифікація (RFID) і бездротові сенсорні мережі (WSN), стало можливим і доступним реалізувати концепцію Інтернету речей. Зазвичай, IoT дозволяє фізичним об'єктам взаємодіяти між собою та обмінюватися інформацією через Інтернет з метою виконання або генерації корисних дій. Таким чином, IoT можна розглядати як розширення інформаційних технологій

у всіх сферах життя, перетворюючи даний момент ізольовані мережі на нову глобальну взаємозалежну гетерогенну мережу розумних об'єктів або речей.

З урахуванням всього вищезазначеного, середовище Інтернету речей складатиметься з розмаїття технологій і пристроїв. Кожен з них має власну специфіку, розроблений різними постачальниками з різними можливостями, складностями та швидкістю передачі даних. Незважаючи на ці відмінності, пристрої Інтернету речей зазвичай називаються "розумними пристроями", хоча їх потужності можуть варіюватися. Простіше кажучи, середовище IoT представляє собою обмежену систему, що складається з будь-яких пристроїв, які на даний момент підключені до Інтернету, а також вбудованих пристроїв, встановлених у побутових об'єктах. Більш того, розумний пристрій є пристроєм або вузлом, який має наступні характеристики:

– фізичну наявність;
– засоби зв’язку повинні бути однозначно ідентифіковані;
– мають деякі базовими обчислювальні можливості;
– можуть взаємодіяти з навколишнім середовищем.

З огляду на необхідність Інтернету у функціонуванні Інтернету речей, можна визначити стек протоколів TCP/IP, аналогічний стеку, використовуваному для Інтернету, для середовища IoT. У табл. 2.1 представлений стек протоколів, призначених для Інтернету речей.

Таблиця 2.1 – IoT-стек протоколів

<table>
<tr><td rowspan="2">Прикладний рівень</td><td colspan="5">Програми IoT</td></tr>
<tr><td>HTTP</td><td>MQTT</td><td>XMPP</td><td>Rest/SOAP</td><td>CoAP</td></tr>
<tr><td rowspan="2">Транспортний рівень</td><td colspan="3">TLS</td><td colspan="2">DTLS</td></tr>
<tr><td colspan="3">TCP</td><td colspan="2">TCP/UDP</td></tr>
<tr><td rowspan="3">Мережевий рівень</td><td colspan="3">Roll – RPL</td><td colspan="2" rowspan="2">IPSec</td></tr>
<tr><td colspan="3">6LoWPAN</td></tr>
<tr><td colspan="5">IPv6</td></tr>
<tr><td>Канальний рівень</td><td rowspan="2">ZigBee IEEE 802.15.4</td><td rowspan="2">Bluetooth IEEE 802.15.1</td><td rowspan="2">RFID/NFC</td><td rowspan="2">Wi-Fi IEEE 802.11 a/b/g</td><td rowspan="2">GSM/LTE</td></tr>
<tr><td>Фізичний рівень</td></tr>
</table>

Для передачі інформації в даній роботі використовується протокол MQTT (Message Queue Telemetry Transport) [2]. Цей простий мережевий протокол працює над TCP/IP і дозволяє пристроям передавати інформацію з датчиків на базі доступних елементів, таких як мікроконтролери.

2.1.1 Модель архітектури IoT

Вище була наведена поширена архітектура рівнів Інтернету речей (IoT). Нижче представлена покращена модель архітектури системи IoT (рис. 2.5) з описом особливостей кожного рівня.

Перш за все, потрібно зазначити, що застосована модель складатиметься з шести шарів, а саме:

1) рівень кодування;
2) рівень обробки;
3) мережевий рівень;
4) рівень проміжного ПЗ;
5) рівень застосунків;
6) рівень бізнес-логіки.

Рисунок 2.5 – Рівні архітектури IoT

Рівень кодування є першим шаром в архітектурі IoT і виконує процес ідентифікації для кожного інтелектуального пристрою. Кожному пристрою присвоюється унікальний ідентифікатор, який відрізняє його від інших пристроїв.

Рівень обробки включає бездротову мережу датчиків (WSN), різні типи датчиків та інше обладнання. Основними функціями цього шару є збір даних з фізичних пристроїв та перетворення їх у цифровий сигнал. Після цього рівень обробки передає дані на мережевий рівень.

Мережевий рівень використовує мобільні мережі, Інтернет та інші засоби зв'язку. Цей шар отримує дані з рівня обробки і передає їх до проміжного програмного забезпечення через різні середовища передачі, такі як Wi-Fi, Bluetooth, WiMAX, ZigBee, GSM, 3G і 4G. Він використовує комунікаційні протоколи, такі як IPv4, IPv6, MQTT і DDS. Мережевий рівень відповідає за обробку, управління та обслуговування даних.

Рівень проміжного програмного забезпечення отримує велику кількість інформації з мережевого рівня та обробляє дані за допомогою інтелектуальних систем обробки, таких як хмарні обчислення. Цей рівень забезпечує прямий доступ до бази даних та зберігання всієї інформації в хмарі.

Рівень проміжного програмного забезпечення базується на сервісно-орієнтованій архітектурі (Service Oriented Architecture, SOA). SOA є шаблоном програмного забезпечення, який використовує модульний підхід до розробки програмного забезпечення. Він ґрунтується на використанні розподілених, слабко пов'язаних компонентів замінного призначення, які мають стандартизовані інтерфейси для взаємодії за допомогою стандартизованих протоколів.

Основною функцією програмного забезпечення на цьому етапі є забезпечення постачання всіх функцій системи до кінцевих користувачів. Процес створення сервісу надає можливості кожному смарт-об'єкту і керує ними. Процес абстракції об'єктів забезпечує обмін інформацією між різними

об'єктами за допомогою "спільної мови". Для забезпечення захисту обмінюваних даних використовується процес управління довірою, конфіденційністю та безпекою.

Рівень застосунків (прикладний рівень) використовує оброблені дані для подальшої роботи різних програм. Програмні застосунки IoT враховують потреби користувачів у промисловості, освіті, медичному секторі та комунікаціях. На прикладному рівні використовуються різноманітні протоколи, такі як протокол обмеженого застосування (CoAP), спрощений мережевий протокол, що працює на TCP/IP (MQTT), відкритий стандарт протоколу прикладного рівня для обробки повідомлень (AMQP), а також протокол XMPP - відкритий мережевий протокол для швидкого обміну повідомленнями та інформацією про присутність між користувачами мережі Інтернет.

Бізнес-рівень є останнім шаром архітектури IoT і відповідає за управління застосунками та послугами системи IoT. Бізнес-шар використовується для створення різних моделей, що задовольняють різні потреби.

2.2 Класифікація рівнів архітектури та відповідних засобів безпеки IoT

Архітектура безпеки IoT включає три основні шари, які можна класифікувати за рівнями обробки, мережі та прикладного рівня. Кожен шар має свої компоненти, стандарти зв'язку та протоколи. Шари безпеки IoT надають різні протоколи безпеки, послуги та механізми безпеки для підвищення загального рівня захисту системи IoT.

На рис. 2.6 зображена архітектура шару безпеки IoT. В наступних розділах будуть відображені компоненти, функції, загальні атаки, проблеми та заходи безпеки для кожного шару безпеки. На рис. 2.6 також представлені атаки і контрзаходи на кожному шарі безпеки IoT.

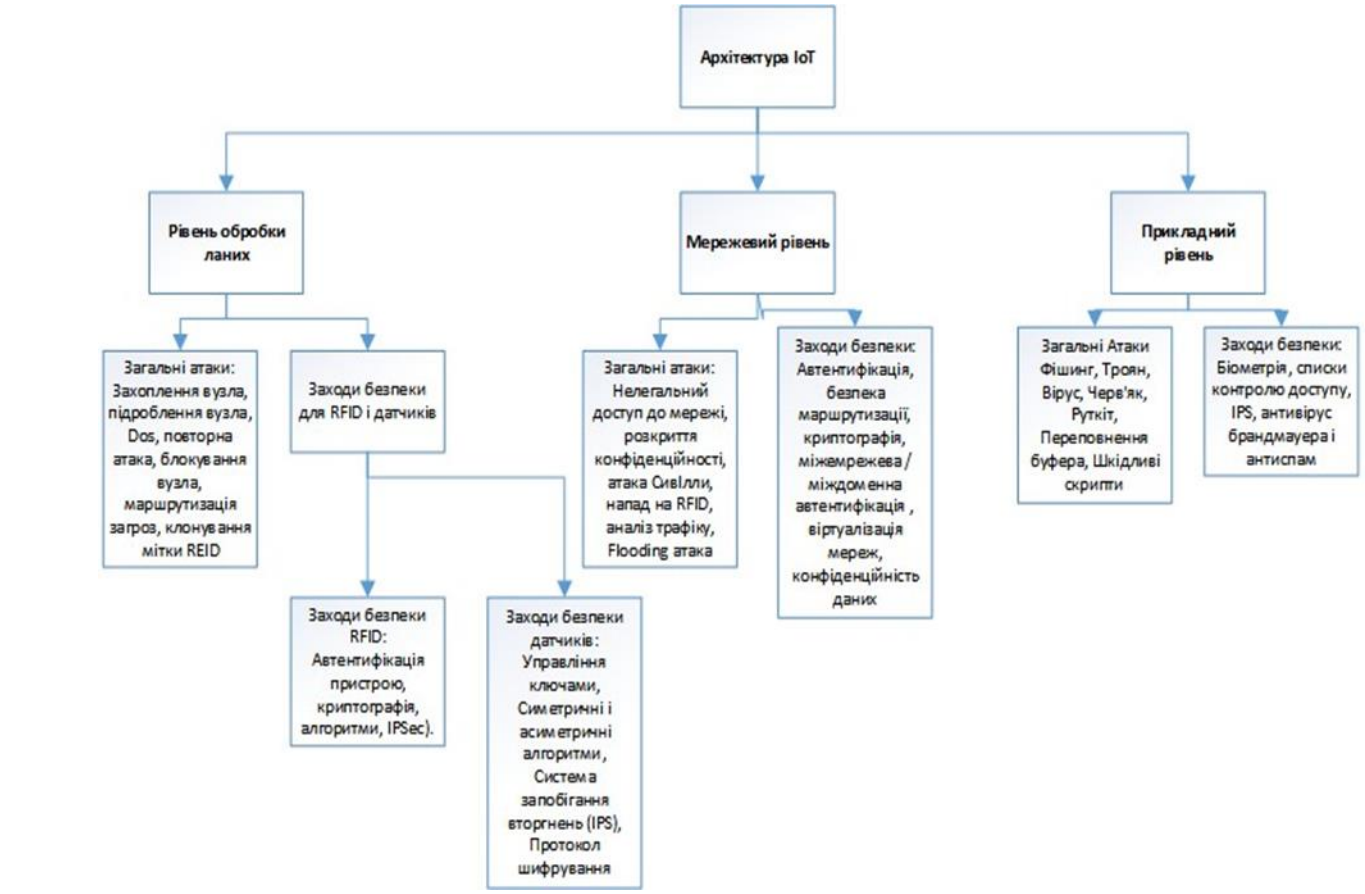


Рисунок 2.6 – Атаки та заходи безпеки на кожному шарі безпеки IoT

На рисунку представлені компоненти та функції кожного шару безпеки в системі IoT. Цей рисунок також відображає атаки, проблеми та вимоги безпеки, які важливі для кожного шару. Таблиця 2.2 надає ілюстрацію багатьох методів безпеки, які використовуються для вирішення проблем безпеки в системі IoT.

Що стосується вимог безпеки, вони гарантують високий рівень безпеки для мережі IoT і покращують продуктивність мережі. Таблиця 2.2 містить інформацію про атаки та найпоширеніші категорії проблем для кожного шару, а також опис необхідних заходів безпеки.

Таблиця 2.2 – Специфіка основних рівнів безпеки та методи їх захисту

Рівень	**Компоненти**	**Функції**	**Атаки/проблеми/ слабкі місця**	**Методи/Механізми безпеки**
Рівень обробки даних	Усі типи датчиків, RFID, GPS, Bluetooth	Використо-вується для зв'язку різних	Захоплення вузлів, підробка вузлів, відмова в обслуговуванні	Алгоритми хешування, криптографічні алгоритми, контроль доступу, IPSec,

Рівень	Компоненти	Функції	Атаки/проблеми/ слабкі місця	Методи/Механізми безпеки
		смарт-пристроїв в IoT, збору інформації та передачі даних на мережевий рівень	(DDoS), Replay Attack, блокування вузлів, загрози маршрутизації, клонування міток RFID тощо	Керування ключами (PKI), система запобігання вторгнень, протокол шифрування, оцінка ризиків
Мережевий рівень	Стільни-ковий зв'язок та Інтернет	Використо-вується для передачі інформації	Атаки: викрадення сеансів, Sybil, RFID Spoofing, аналіз трафіку та Flooding-атаки. Проблеми: сумісність, безпека кластерів, незаконна мережа доступу та розкриття конфіденційності	End to End authentication (автентифікація від одного кінця до іншого) безпечна маршрутизація, алгоритми криптографії, автентифікація між мережами/доменами, технологія мережевої віртуалізації, конфіденційність даних і цілісність для виправлення та контролю помилок
Прикладний рівень	Розумний зв'язок (правильно підібраний тип зв'язку)	Використо-вується для надання багатьох послуг та аналізу інформації	Дозвіл на доступ до даних, захист і відновлення даних, можливість виправлення вразливостей програмного забезпечення	Біометрія, списки контролю доступу (ACL), IPS, Антивіруси, Антиспами і Firewall

Крім того, табл. 2.2 відображає переваги і недоліки заходів безпеки, що застосовуються при розробці механізмів безпеки. Також в цій таблиці пояснюється, які атаки можна уникнути застосовуючи різні методи.

Крім вищезазначеного, існують інші методи захисту на рівні обробки, які описані нижче.

Одним з таких методів є безпечне завантаження. Воно використовує криптографічні хеш-алгоритми для перевірки цілісності пристроїв та програмного забезпечення Інтернету за допомогою цифрового підпису. Однак цей механізм безпеки не є ефективним для системи IoT, оскільки вимагає значної потужності та часу.

Ще одним рішенням для захисту особистої інформації користувачів в мережі IoT є анонімність. Анонімність є ефективним методом, проте його недоліком є необхідність використання більшої обчислювальної потужності.

Оцінка ризиків є важливим методом захисту мережі IoT, спрямованим на запобігання різним загрозам і атакам. Цей метод є найважливішим для системи IoT, оскільки він дозволяє виявляти помилки в системі безпеки. Він здатний виявляти різні загрози та атаки в пристроях IoT, використовуючи методи, наприклад, системи запобігання вторгнень (IPS). Оцінка ризиків передбачає використання різних механізмів безпеки, які відповідають особливостям середовища IoT. Ці механізми повинні забезпечувати ефективне використання енергії та часу для досягнення високої продуктивності мережі IoT. Тому методи безпеки повинні постійно модифікуватися, вдосконалюватися та адаптуватися до потреб інтелектуальних об'єктів системи IoT.

Заходи безпеки на мережевому рівні спрямовані на задоволення двох основних вимог безпеки - конфіденційності і цілісності даних. Вони включають в себе різні механізми безпеки, які допомагають виконувати ці вимоги і запобігати атакам. Таблиця 2.3 містить різні методи безпеки, які застосовуються на мережевому рівні.

Таблиця 2.3 – Методи безпеки мережевого рівня

Метод безпеки	**Корисність використання**	**Тип атаки, якої можна уникнути таким способом**	**Переваги**	**Недоліки методу**
Наскрізне шифрування та управління ключами	Всі вузли в мережі IoT повинні проходити перевірку автентичності з використанням механізму перевірки достовірності, інфраструктури відкритих ключів та	Нелегальний доступ до вузлів, DoS і атаки в Sinkhole	Забезпечує наскрізну автентифі-кацію і шифрування	Це важкий метод безпеки

Метод безпеки	Корисність використання	Тип атаки, якої можна уникнути таким способом	Переваги	Недоліки методу
	наскрізного шифрування			
Безпека маршрутизації. (Security Aware and Routing)	Наступним кроком заходів безпеки мережевого рівня є безпечна маршрутизація, яка відбувається після процесу автентифікації. Механізми безпеки маршрутизації важливі для захисту даних і збереження конфіденційності даних	Більшість загроз і атак	Забезпечує багатопро-меневе поширення для маршрутизації даних і розширює можливості системи по виявленню будь-яких помилок в системі	Цей метод потребує значної кількості часу опрацювання
Система криптографії	Використовується для перевірки передачі даних через інші вузли і виявлення будь-якої помилки в мережі	Запобігає підроблення даних на прийнятому вузлі	Вона може виявити мережеву помилку і перевірити дані. Криптографія з симетричним ключем потребує мало енергії і часу	Асиметрична криптографія потребує потужність і час
Автентифікація крос-мережі та домену	Для захисту протоколів використовується автентифікація між двома мережами. Для захисту DNS використовується автентифікація між доменами	Загрози маршрути-зації	Використо-вується для захисту мережевих протоколів	Потребує більше енергії
Технологія віртуалізації мережі	1. Це процес об'єднання апаратних і програмних ресурсів і мережевих функцій в одну або віртуальну мережу; 2. Існує два типи віртуальної мережі: зовнішня і внутрішня віртуалізація;	Більшість мережевих атак	Використо-вується для зменшення складності управління мережею	Цей метод потребує значної кількості часу опрацювання

Метод безпеки	Корисність використання	Тип атаки, якої можна уникнути таким способом	Переваги	Недоліки методу
	3. Зовнішня віртуалізація об'єднує безліч мережевих частин у віртуальний пристрій – LAN для підвищення точності мережі і ефективності даних; 4. Внутрішня віртуалізація забезпечує мережеву функціональність програмного забезпечення на одному мережевому сервері			
Метод перевірки цілісності та конфіденційності даних	Використовується для виявлення і контролю будь-якої помилки, яка відбувається в мережі. Цілісність даних використовує алгоритми шифрування для перевірки вихідних даних, які відправляються на сторону одержувача	Незаконний доступ і підробка (спуфінг)	Використовується для перевірки вихідних даних	Цей метод потребує значної кількості часу опрацювання
Виявлення флуду (Flooding)	Ідея цього методу полягає в тому, що відправник надсилає hello-запит одержувачу, яке використовується для перевірки якості сигналу. Якщо цей сигнал схожий на поодинокі в діапазоні радіо, приймач приймає повідомлення	Флуд атака	Використовується для перевірки справжності сигналу	Потребує затрат часу

2.3 Проєктування нової моделі архітектури IoT на основі лічильника Гейгера

Мета створення цієї моделі полягає в представленні системи управління безпекою мережі IoT, яка використовує лічильник Гейгера для зменшення часу обробки даних і використовує низькі потужності, щоб забезпечити відповідні механізми безпеки на кожному рівні безпеки IoT. Ця запропонована модель допомагає дослідникам і розробникам вибирати оптимальні протоколи і механізми безпеки для кожного рівня, що дозволяє захищати дані і інтелектуальні об'єкти. Модель спрямована на максимальне запобігання атак, загроз і проблем або зменшення їх впливу. Основні цілі цієї моделі можуть бути уточнені наступним чином:

- Проведення детального дослідження для вибору відповідних механізмів безпеки на кожному рівні безпеки IoT і розкриття переваг і недоліків різних методів безпеки.

- Забезпечення вимог безпеки, таких як контроль доступу, управління маршрутизацією, автентифікація, конфіденційність і цілісність на кожному рівні безпеки.

- Гарантування безпеки для різних застосунків.

- Забезпечення надійних функцій для кожного смарт-об'єкта в мережі IoT, включаючи системи виявлення та запобігання вторгнень (IDS/IPS) і відновлення безпеки.Представлена далі модель виявляє і запобігає більшості загроз і атак; захищає особисту інформацію користувачів; виявляє будь-яку помилку при передачі даних.

У запропонованій моделі використовується платформа ThingsBoard, яка надає широкий спектр механізмів безпеки. Це дозволяє управляти стратегією вибору алгоритмів безпеки для досягнення високого рівня вимог безпеки, а також зменшення споживання енергії та часу.

Запропонована модель складається з трьох частин (рис. 2.7). Перша частина включає рівні безпеки IoT, які відображають рівні обробки, мереж та застосунків.

Друга частина моделі містить протоколи безпеки та механізми на рівні безпеки IoT.

Третя частина включає сервери баз даних, які використовуються для кожного рівня безпеки IoT з метою зберігання всієї інформації про механізми безпеки. Ці сервери баз даних є корисними для адміністраторів та користувачів для збереження лог-файлів методів безпеки та інформації про користувачів.

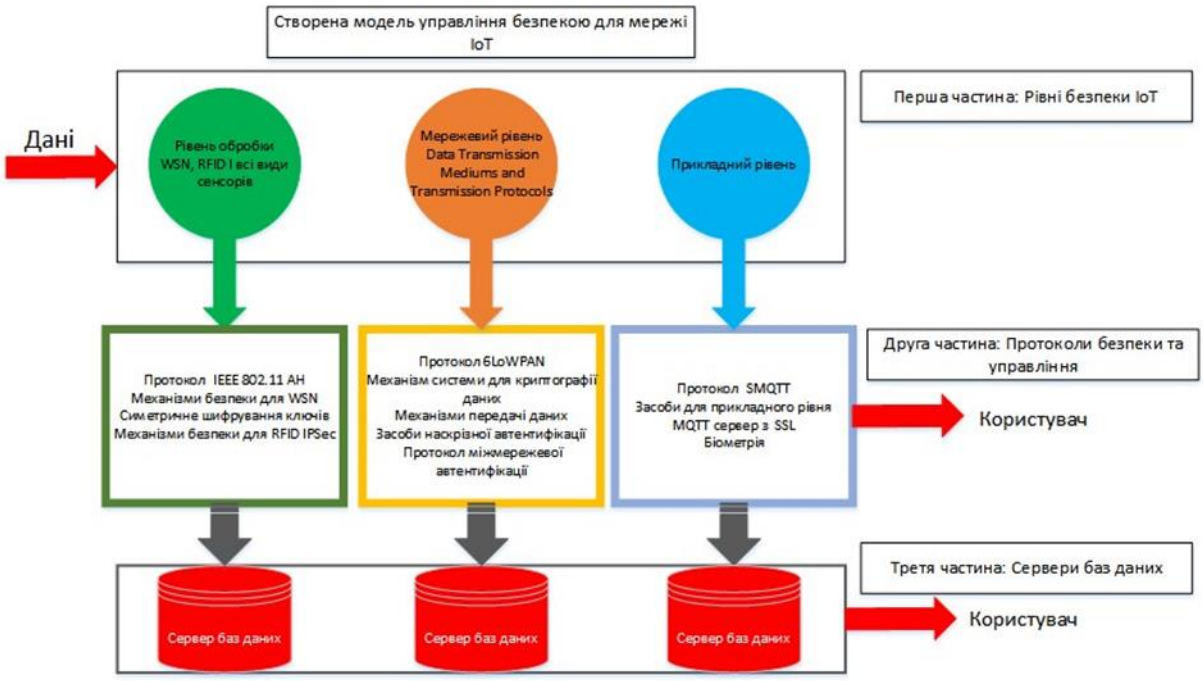


Рисунок 2.7 – Створена модель системи управління безпекою для мережі IoT

Перша частина запропонованої моделі вже була обговорена в попередніх розділах, тому далі зосередимося на другій і третій частинах.

Попередній розділ вже охопив першу частину запропонованої моделі, тому зараз зосередимось на другій і третій частині.

Друга частина моделі складається з трьох основних розділів, де детально описуються використовувані протоколи та механізми безпеки на рівнях безпеки IoT відповідно.

Третя частина моделі включає сервери баз даних, які зберігають повну інформацію і параметри механізмів безпеки для кожного рівня безпеки. Вони також зберігають профілі користувачів, інформацію про помилки механізмів безпеки, лог-файли системи IoT та списки контролю доступу. Третій етап допомагає адміністраторам і користувачам ефективно керувати всією інформацією про мережу IoT та користувачів.

2.4 Протоколи і механізми безпеки рівнів архітектури за моделлю OSI

Механізми безпеки на рівні обробки даних включають в себе протоколи і методи, що забезпечують безпеку та конфіденційність. Один з найбільш прийнятних протоколів для рівня обробки є IEEE 802.11ah, який є відповідним для бездротового зв'язку. Цей протокол є легким, споживає мало енергії і часу, що допомагає зменшити накладні витрати. Він також забезпечує ефективний двонаправлений обмін пакетами, що дозволяє датчику економити енергію шляхом використання зв'язку по висхідній і низхідній лініях між датчиками. Датчик передає дані і переходить в сплячий режим, коли завершує своє завдання. Додатково, короткий MAC-адрес використовується для збільшення часу очікування та енергозбереження. IEEE 802.11ah використовує алгоритми шифрування для забезпечення конфіденційності та приватності.

Механізми безпеки на рівні сприйняття можуть бути вибрані згідно з попереднім оглядом, який був наведений у попередніх таблицях. У цих таблицях представлені переваги та недоліки різних механізмів безпеки для бездротових сенсорних мереж (WSN) та систем ідентифікації з використанням радіочастотного ідентифікатора (RFID). Для WSN рекомендованими механізмами безпеки є управління ключами (PKI) та алгоритми безпечного

ключа, які використовують алгоритми симетричного шифрування з низьким енергоспоживанням.

IPSec - це механізм, який використовується для забезпечення безпеки RFID. Він надає алгоритми автентифікації і шифрування.

Для автентифікації використовується токен доступу, який надає односторонню хеш-функцію. Це дозволяє користувачам ввести імена користувачів і паролі для отримання токену доступу до певного ресурсу без використання самого імені користувача і пароля. Після отримання токену доступу, користувач може використовувати його для доступу до ресурсу на визначений період часу. Токен доступу може бути переданий як частина URL-адреси запиту або як ім'я користувача. Використання токену доступу забезпечує авторизацію, контроль доступу, доступність і конфіденційність.

Для шифрування використовується симетричне шифрування, яке забезпечує простий алгоритм шифрування з низьким споживанням енергії і часу. Оцінка анонімності і ризику використовується для захисту приватної інформації користувачів і виявлення мережевих помилок у всіх типах датчиків.

Найбільш прийнятним протоколом мережевого рівня є 6LowPAN, який використовується для інкапсуляції IPv6. IPv6 забезпечує довгий заголовок у невеликих пакетах. 6LowPAN має низьку пропускну здатність, мале енергоспоживання, низьку вартість, мобільність, масштабованість мережі та довгий час очікування. Відповідні механізми безпеки на мережевому рівні можуть бути класифіковані в залежності від безпеки передачі даних, середовища і протоколу передачі.

Система криптографії використовується для забезпечення безпеки передачі даних за допомогою симетричного ключа шифрування. Ця система використовує незначну кількість енергії і займає мало часу для виконання.

У передачі даних використовується наскрізний алгоритм з автентифікацією на основі сертифікатів X.509. Для цього використовується двостороннє з'єднання Socket Secure Layer (SSL), яке генерує сертифікат на

стороні клієнта та встановлює з'єднання з сервером. Для забезпечення цього механізму використовується PKI, що дозволяє уникнути необхідності розповсюдження відкритих ключів або перевірки відбитків пальців при створенні або оновленні ключів. Цей метод є масштабованим, оскільки вимагає довіри до одного або обмеженої кількості сертифікатів автентифікації (CA). Він також забезпечує перевірку особистості за допомогою секретних приватних ключів.

Механізм міжмережевої автентифікації використовується для захисту протоколу передачі в мережах Інтернету речей (IoT). Цей механізм спрощує управління мережею та зменшує її складність.

Для демонстрації доцільності та переваг протоколу MQTT порівняно з HTTP, були вивчені характеристики обох протоколів, а також зібрані показники часу відгуку та розміру пакетів при передачі однакового навантаження через MQTT і HTTP.

Для об'єктивного порівняння між протоколами необхідно врахувати всі етапи процесу автентифікації (рукостискання). У випадку MQTT це означає, що час відповіді для відправки повідомлень про підключення і відключення вимірюється послідовно з часом відповіді для фактичних повідомлень даних.

Були виміряні час відгуку для відправки 1, 100 і 1000 повідомлень через MQTT з одним циклом з'єднання, а також зареєстровані розміри пакетів, що були надіслані по дроту. Крім того, виміряно час відгуку для відправки одного повідомлення з 1, 10 і 100 полями властивостей через MQTT з одним циклом підключення, а також зафіксовано розмір відправленого пакета. Далі були виміряні середні часи відгуку для відправки корисного навантаження через HTTP з 1, 10 і 100 полями властивостей, а також зафіксовано розмір пакета по кабелю.

Нижче приведені дані про передачу пакетів через HTTP і MQTT з використанням лише одного імітованого користувача (табл. 2.4–2.6). Передане повідомлення представляє собою простий об'єкт, що складається з однієї пари ключ-значення.

Таблиця 2.4 – Вплив зміни кількості повідомлень на час

Кількість полів властивостей у повідомленні, шт.	**Середній час відповіді для циклу з'єднання, мс**	**Середній час відповіді на поле властивості, мс**
1	113	113
10	4724	47
100	40366	43

Таблиця 2.5 – Вплив зміни розміру корисного навантаження на час

Кількість полів властивостей у повідомленні, шт.	**Середній час відповіді для циклу з'єднання, мс**	**Середній час відповіді на поле властивості, мс**
1	207	207
10	212	21
100	191	3

Таблиця 2.6 – Час відповіді HTTP

Кількість полів властивостей у повідомленні, шт	**Середній час відповіді, мс**	**Середній час відповіді на поле властивості, мс**
1	289	289
10	280	28
100	247	3

Отримані дані щодо розміру пакетів. Для отримання більш точного уявлення про реально надсилані пакети через мережу, використовувалась програма Wireshark для захоплення всіх пакетів, що передавалися через TCP-порт. Розміри кожного пакета були також записані.

Журнал логів відображає процес рукостискання, який встановлює TLS-тунель для забезпечення MQTT-зв'язку (рис. 2.8). Основна частина цього

процесу полягає у взаємному обміні та перевірці сертифікатів і відкритих ключів.

Protocol	Length	Info
TCP	98	60434 → secure-mqtt(8883) [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1338 WS=32 TSval=939896599 TSecr=0 SACK_PERM=1
TCP	94	secure-mqtt(8883) → 60434 [SYN, ACK] Seq=0 Ack=1 Win=59312 Len=0 MSS=1360 SACK_PERM=1 TSval=3280903056 TSecr=939896599 WS=256
TCP	86	60434 → secure-mqtt(8883) [ACK] Seq=1 Ack=1 Win=131264 Len=0 TSval=939896627 TSecr=3280903056
TLSv1.2	603	Client Hello
TCP	86	secure-mqtt(8883) → 60434 [ACK] Seq=1 Ack=518 Win=60416 Len=0 TSval=3280903085 TSecr=939896628
TLSv1.2	1294	Server Hello
TLSv1.2	1294	Certificate [TCP segment of a reassembled PDU]
TCP	86	60434 → secure-mqtt(8883) [ACK] Seq=518 Ack=2417 Win=129856 Len=0 TSval=939896657 TSecr=3280903087
TLSv1.2	387	Server Key Exchange, Server Hello Done
TCP	86	60434 → secure-mqtt(8883) [ACK] Seq=518 Ack=2718 Win=130720 Len=0 TSval=939896658 TSecr=3280903087
TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
TLSv1.2	365	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
TCP	86	60434 → secure-mqtt(8883) [ACK] Seq=644 Ack=2997 Win=130784 Len=0 TSval=939896695 TSecr=3280903127
TLSv1.2	676	Application Data
TLSv1.2	119	Application Data

Рисунок 2.8 – MQTT через протокол підключення TLS

Принцип роботи полягає в наступному: журнал логів показує, що під час циклу передачі одного повідомлення в протоколі MQTT, повідомлення відправляється від клієнта до сервера, потім MQTT відправляє підтвердження ACK повідомлення назад до клієнта, а також клієнт надсилає ACK TCP для отриманого MQTT ACK. Процедура ініціалізації для налаштування захищеного TLS-тунелю для протоколу HTTP аналогічна MQTT і тепер встановлений захищений тунель використовується повторно для всіх наступних запитів (рис. 2.9).

Protocol	Length	Info
TCP	1294	https(443) → 54264 [ACK] Seq=1209 Ack=518 Win=28160 Len=1208 TSval=3779774246 TSecr=1012044973
TCP	86	54264 → https(443) [ACK] Seq=518 Ack=2417 Win=128832 Len=0 TSval=1012044983 TSecr=3779774246
TLSv1.2	98	Ignored Unknown Record
TCP	86	54264 → https(443) [ACK] Seq=518 Ack=2429 Win=130880 Len=0 TSval=1012044984 TSecr=3779774246
TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
TLSv1.2	365	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
TCP	86	54264 → https(443) [ACK] Seq=644 Ack=2708 Win=130784 Len=0 TSval=1012045000 TSecr=3779774264
TLSv1.2	972	Application Data
TCP	86	https(443) → 54264 [ACK] Seq=2708 Ack=1530 Win=29952 Len=0 TSval=3779774308 TSecr=1012045001

Рисунок 2.9 – HTTP через протокол підключення TLS

Протокол HTTP не встановлює постійне з'єднання, тому токен JWT, що використовується для автентифікації, передається у заголовку кожного окремого запиту. У наведеній табл. 2.7 нижче показано розмір пакета, що надсилається під час кожного етапу передачі для протоколів MQTT і HTTP.

Таблиця 2.7 – Сума розміру переданих пакетів

	MQTT	**HTTP**
Встановлення з'єднання	5572	2261
На опублікування повідомлення	388	3285
На роз'єднання	376	0
Сума	6336	5546

У табл. 2.8 показано, як зміна розміру корисного навантаження на розмір пакету у мережі.

Таблиця 2.8 – Вплив зміни розміру корисного навантаження на розмір пакету

Корисне навантаження (Payload)	MQTT 1	MQTT 10	MQTT 100	HTTP 1	HTTP 10	HTTP 100
Розмір пакета, що передається по кабелю, байт	388	686	3788	1982	2332	6683
Співвідношення розмірів пакету в порівнянні з одним полем	1	1,77	9,76	1	1,18	3,37

Таблиця 2.9 – Результуюча таблиця (кількість витрачених байт)

	MQTT	HTTP
Встановлення з'єднання	5572	2261
На опублікування повідомлення	388	3285
На роз'єднання	376	0
Сума	6336	5546

Аналізуючи результати, які порівнюють час відгуку за один цикл з'єднання для MQTT, ми можемо побачити, що ініціалізація з'єднання значно збільшує час відгуку для відправки окремих повідомлень, приблизно до рівня часу відгуку для відправки одного повідомлення за допомогою HTTP, що у даному випадку становить близько 120 мс на повідомлення. Вплив розміру даних (навантаження), які надсилаються, має ще більше значення для MQTT, де відправляється приблизно 6300 байтів для одного повідомлення, що є більшим,

ніж для HTTP, де розмір пакета становить 5600 байтів. Під час аналізу журналу трафіку пакетів видно, що понад 90% переданих даних використовуються для встановлення і розірвання з'єднання.

Перевага MQTT над HTTP стає очевидною, коли використовується одне з'єднання для передачі багатьох повідомлень. В такому випадку середній час відповіді на повідомлення становить близько 40 мс, а обсяг даних на повідомлення приблизно 400 байтів. Зазначимо, що ці покращення просто неможливі у випадку HTTP. Вибір між MQTT і HTTP залежить від того, наскільки ми можемо повторно використовувати одне з'єднання. Якщо з'єднання часто встановлюється і розривається для надсилання окремих повідомлень, то ефективність MQTT порівняно з HTTP не є значущою.

Найбільше покращення ефективності можна досягти, збільшуючи щільність інформації в кожному повідомленні MQTT.

Таким чином складемо загальну табл. 2.10 з основними характеристиками по двом протоколам.

Таблиця 2.10 – Основні характеристики протоколу MQTT та HTTP

	MQTT	**HTTP**
Повна назва	Message Queue Telemetry Transport -спрощений мережевий протокол	Hyper Text Transfer Protocol – протокол передачі гіпер-текстових документів
Архітектура	Публікація-підписка	Клієнт-серверна
Протокол, на якому працює	TCP	TCP и UDP
Розмір повідомлень	Маленький	Великий
Формат повідомлень	Двійковий з заголовком 2 байта	ASCII формат
Розподіл даних	Від 1 до 0/1 / N	Лише «один до одного»
Безпека даних	Так, використовує SSL / TLS для безпеки передачі даних	Сам по собі HHTP не надає безпечну передачу, для цього використовується HTTPS
Складність	Простий протокол	Більш складний через

	MQTT	HTTP
		використання ASCII парсеру
Шифрування	Шифрує корисне навантаження	Дані не шифруються перед передачею
Коли використовувати	Якщо потрібно холодильнику зв’язатись з термометром для адаптації насоса двигуна, то цей варіант значно кращий	Якщо мета зібрати якомога більше інформації, то достатньо HTTP

У даному випадку використовується протокол MQTT, який застосовує легкі алгоритми шифрування для забезпечення низького споживання енергії та швидкості. Використання SSL на сервері MQTT є основою для забезпечення безпеки в мережі речей IoT та захисту конфіденційної інформації. SSL використовує шифрування для захисту конфіденційної інформації, автентифікації, забезпечення критичної безпеки та цілісності даних для інтерфейсу програм та особистої інформації користувачів.

Реалізація запропонованої моделі та її варіацій підходить для різних платформ мереж речей IoT. Ця модель може служити основою для подальших досліджень та визначення впливу на алгоритми безпеки та споживання енергії в мережах IoT.

Висновки за розділом 2

У контексті IoT-пристроїв, безпека передбачає забезпечення цілісності коду, аутентифікацію користувачів (пристроїв), встановлення прав доступу та захист від віртуальних та фізичних атак. Проте, багато працюючих IoT-пристроїв фактично не мають належних заходів безпеки: вони мають доступні зовні інтерфейси управління, використовують стандартні паролі, не дотримуються необхідних стандартів та не шифрують комунікаційні канали.

Для досягнення низького споживання енергії та мінімізації часу доставки даних про радіаційний фон у вузькому просторовому масштабі з IoT-пристроїв до централізованого сервера, рекомендовано використовувати протокол MQTT, який використовує легкі алгоритми шифрування. Використання сервера MQTT

з підтримкою SSL є основою безпеки в мережі IoT, де конфіденційна інформація захищена.

Отже, необхідно визначити архітектуру IoT, яка належним чином описує необхідні стандарти, протоколи та засоби безпеки на кожному рівні.

3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОЇ ЧАСТИНИ ІОТ-МЕРЕЖІ

3.1 Апаратна частина

Сучасні датчики радіоактивності бувають двох основних типів:

1) сцинтиляторний детектор;

2) лічильник Гейгера-Мюллера.

У сцинтиляторному детекторі використовується сцинтиляційний кристал йодиду цезію, активований талієм. Даний кристал має властивість радіолюмінесценції – заряджені частинки та фотони високої енергії (рентгенівського та гамма-діапазону) збуджують у ньому світіння, причому світло випромінюється у вигляді короткої, близько мікросекунди, спалаху світла – сцинтиляції. Цей спалах занадто слабкий, щоб його можна було побачити оком або виявити звичайним способом. Зазвичай для уловлювання таких слабких імпульсів світла застосовували (і зараз застосовують) фотоелектронні помножувачі. У них кожен фотоелектрон, вибитий з фотокатода, розмножується на системі динодів, даючи посилення в мільйони разів, і імпульс струму на його аноді складає вже не нано-, а міліампери, і зареєструвати такий імпульс вже не важко. Але ФЕУ – це солідних габаритів скляний балон та кіловольти живлення, що вимагають також високої стабільності. Загалом він погано уявляється в приладі кишенькових розмірів.

Але зараз з’явились інші пристрої для детектування надслабких оптичних сигналів — SiPM (Silicon photomultipliers), які являють собою матрицю з великої кількості лавинних фотодіодів, що працюють у передпробійному режимі, в якому єдиний фотон здатний спровокувати розвиток лавинного пробою. Кожна з комірок має свою схему гасіння, за рахунок якої лавинний пробій негайно припиняється і комірка стає знову готовою до реєстрації нового фотона. Всі комірки (зі своїми схемами гасіння) з'єднані на кристалі паралельно, і імпульси струму, що протікають через них, складаються, так що

середній струм виявляється пропорційний освітленості кристала. Приклад такого детектора наведений на рис. 3.1:

Рисунок 3.1 – Сцинтиляторний детектор

До переваг таких детекторів можна віднести відсутність високовольтних перетворювачів, механічну стійкість та компактність пристроїв. До недоліків відноситься висока вартість компонентів та складність розробки аналогової частини пристрою.

Лічильник Гейгера-Мюллера, також відомий як Гейгера, працює за відмінним принципом. Складаючись з металевої або металізованої зсередини скляної трубки та тонкої металевої нитки, яка натягнута вздовж осі циліндра, циліндричний лічильник Гейгера-Мюллера (рис. 3.2) працює наступним чином. Нитка виконує роль анода, тоді як трубка виступає катодом. Трубка наповнена розрідженим газом, зазвичай аргоном або неоном. Між катодом та анодом створюється напруга, яка може досягати від сотень до тисяч вольт, залежно від геометричних розмірів, матеріалу електродів та газового середовища всередині лічильника. Широко поширені лічильники Гейгера, які потребують напруги 400 вольт.

Робота лічильника базується на явищі ударної іонізації. При проникненні гамма-квантів, які випускає радіоактивний ізотоп, на стінки лічильника, вони вибивають електрони з матеріалу стінок. Ці електрони рухаються у газовому середовищі лічильника, зіштовхуючись з атомами газу, і в результаті цих зіткнень вибиваються додаткові електрони та утворюються позитивні іони. Завдяки електричному полю, створеному між катодом та анодом, електрони прискорюються до енергій, за яких відбувається ударна іонізація. Це призводить до лавинного розмноження іонів та носіїв заряду. При досить велику напруженість поля енергії цих іонів стає достатньою, щоб породжувати вторинні лавини, здатні підтримувати самостійний розряд, у результаті струм через лічильник різко зростає. Цим лічильник Гейгера відрізняється від пропорційного лічильника, де напруженість поля недостатня виникнення вторинних лавин, і розряд припиняється після прольоту первинної лавини. При цьому на опорі утворюється імпульс напруги, який подається в пристрій, що реєструє.

Для того, щоб лічильник міг зареєструвати наступну частинку, яка потрапляє до нього, лавинний розряд потрібно припинити. Цей процес відбувається автоматично. Коли імпульс струму з'являється на опорі, відбувається значне падіння напруги. Це призводить до різкого зменшення напруги між анодом і катодом до такої міри, що розряд припиняється, і лічильник знову стає готовим до наступної роботи.

Для прискорення гасіння можуть використовуватися спеціальні схеми, що примусово знижують напругу на лічильнику, що дозволяє зменшити анодний опір і збільшити рівень сигналу. Однак частіше в газову суміш у лічильнику додають трохи галогену (брому або йоду) або органічної сполуки з відносно великою молекулярною масою (зазвичай будь-якого спирту) – ці молекули взаємодіють із позитивними іонами, даючи в результаті іони з більшою масою та меншою рухливістю. Крім того, вони інтенсивно поглинають

ультрафіолетове випромінювання розряду – ці два фактори призводять до швидкого та мимовільного гасіння розряду навіть з невеликим анодним опором. Такі лічильники називаються самогасящими. У разі застосування він гасить добавки спирту при кожному імпульсі, деяка його кількість руйнується, тому гасить добавка витрачається і лічильник має певний (хоч і досить великий) ресурс за кількістю зареєстрованих частинок. При його вичерпанні лічильник починає «горіти» – починає мимоволі зростати швидкість рахунку навіть без опромінення, а потім у лічильнику виникає безперервний розряд (рис. 3.2). У галогенних лічильниках молекули галогену, що розпалися, знову з'єднуються, тому їх ресурс значно більший (10^{10} імпульсів і вище).

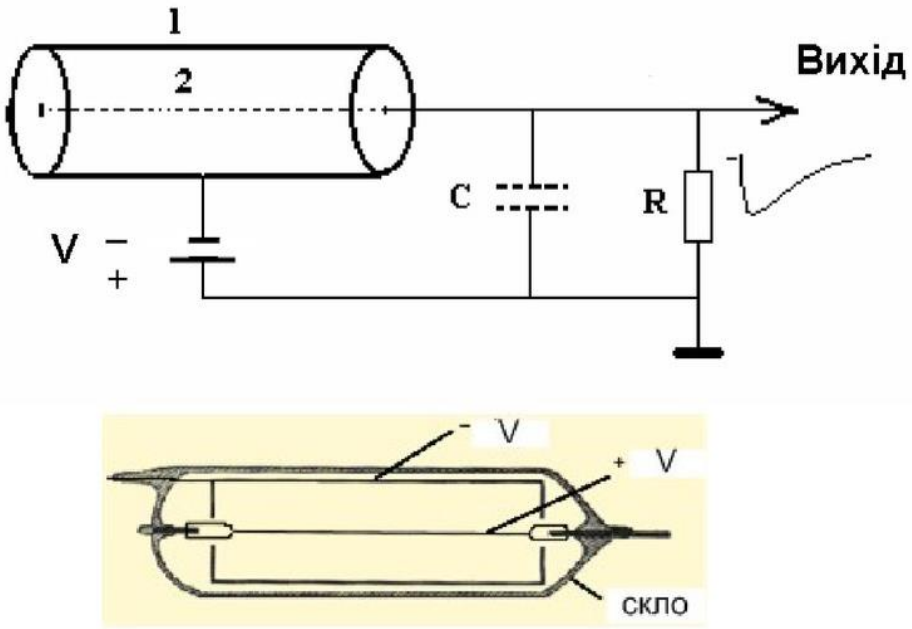


Рисунок 3.2 – Принцип дії газорозрядного лічильника

Для отримання опорної напруги 400 вольт використовується помножувач напруги (рис. 3.3), який робить низьку напругу змінного току від генератора **u** високою на обкладинках газорозрядної колби.

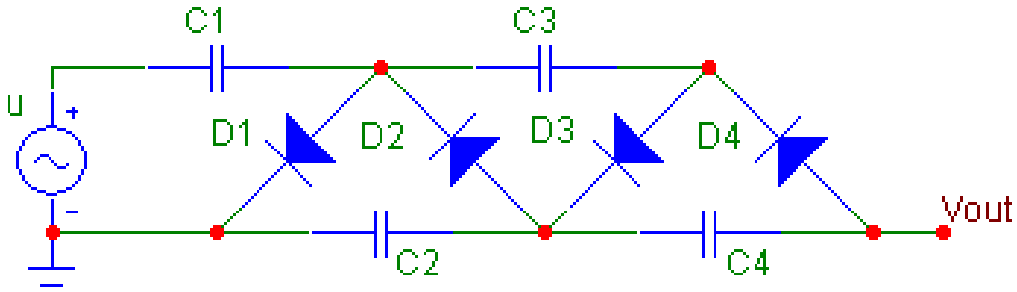


Рисунок 3.3 – Помножувач напруги

При виборі газорозрядної колби треба враховувати наступні чинники:

Типи випромінювання

Трубку потрібно обирати в залежності від того, яке випромінювання вам потрібно вимірювати.

Більшість поширених трубок чутливі до гамма- і бета- випромінювання. Деякі трубки також здатні вимірювати альфа-канал. Альфа-канал у трубки, як правило, реалізується за допомогою наявності слюдяного віконця у торці корпусу.

Щоб перетворити *a,b,g*-трубку на *b,g*-, достатньо щільно закрити слюдяне вікно аркушем паперу, або пластиковою кришкою корпусу.

Щоб перетворити *b,g*- чи *a,b,g*- трубку лише на *g*-, потрібно екранувати трубку від бета-частинок. З цим може впоратися алюмінієвий кожух товщиною в кілька міліметрів. Такий кожух з алюмінію одночасно екранує трубку не лише від бета-, але й від альфа-частинок.

Власний шум та нечутливість

Трубки Гейгера-Мюллера мають дві важливі характеристики, які варто брати до уваги під час їх порівняння.

Власний шум – це хибно-позитивні імпульси, які генерує трубка за відсутності зовнішнього радіоактивного випромінювання. Під час проектування чи калібровки трубки, виробник розміщує тестовий зразок у екранованому від радіації лабораторному середовищі і проводить вимірювання кількості хибно-позитивних імпульсів на одиницю часу. Зазвичай власний шум трубки зазначається у даташит у імпульсах на секунду.

Нечутливість – це час, протягом якого трубка відновлюється після попереднього лавиноподібного збурення та не здатна детектувати наступну таку подію. Цей час прийнято називати dead time трубки і вимірювати у мікросекундах. На практиці, як наслідок, у цей проміжок часу трубка не здатна генерувати вихідний імпульс.

Також варто звернути увагу на те, що час dead time прямо залежить від розмірів трубки. Чим довша трубка, тим більший цей час. Звісно, довжина це не причина, а лише наслідок загальної конструкції більшості трубок та їх принципу дії.

Рівень робочої напруги

Порівнюючи наявні варіанти трубок, потрібно пам’ятати, що трубки різного типу можуть мати індивідуальні рівні напруги живлення. Ці дані зазвичай вказуються у даташит на трубку.

На практиці також важливо враховувати, що вирішальне значення має модуль лічильника Гейгера (і його налаштування!), на який обрану вами трубку потрібно буде встановлювати.

Перевищення напруги живлення трубки гарантовано виведе її з ладу. За недостатньої напруги трубка просто не буде працювати.

Розміри та спосіб кріплення трубки

Деякі трубки, наприклад, J305 і SBM20 мають схожі розміри та зручний спосіб кріплення, який не вимагає пайки. З точки зору виготовлення мікроелектроніки для IoT-пристроїв, вони мають середні розміри в порівнянні з іншими трубками Гейгера. Їх можна назвати взаємозамінними, адже вони мають схожий рівень робочої напруги, однакові клеми, та майже однакові розміри, що дозволяє за необхідності замінювати трубки між собою, якщо є можливість встановлювати відповідні коефіцієнти перерахунку для CPM.

Часто готові пристрої мають кріплення, що підтримують декілька розмірів трубок. Також буває корисним те, що трубку Гейгера можливо швидко зняти з модуля, або замінити. Трубка LND712 має приблизно вдвічі меншу

довжину, що робить її ідеальною, як для розмірів сенсора *a,b,g*-випромінювання. Але у неї вивідні контакти зроблені таким чином, що її доведеться лише паяти. Тому LND712, у парі із значно вищою ціною, перестає бути такою ж «зручною», як SBM20 чи J0305. Іноді, для регулювання налаштувань модуля лічильника Гейгера, потрібно мати можливість зняти трубку, – у випадку з LND712 це буде неможливо зробити без пайки.

Країна походження та рік виготовлення

Країна походження є не менш важливою ніж решта характеристик трубки. Навіть із суто практичної точки зору (витрати коштів та часу на логістику, підтримка місцевого бізнесу, сплата податків, тощо), перебуваючи у США, краще придбати трубку, яка виробляється у США.

Запаси радянських трубок SBM20 у приватних продавців відчутно виснажуються. Строки зберігання та експлуатації радянських зразків вже давно минули. Однією з альтернатив є трубка китайського виробництва J305. Трубки J305, які продаються на Alibaba та Aliexpress, мають 2020–2022 рр. виробництва і цілком задовольняють вимогам за технічними характеристиками та якістю.

Трубки LND712 також мають чудові характеристики, якість та функції. Єдиним недоліком є те, що їх потрібно закуповувати в США, виконувати тривалу логістику в Європу. Враховуючи вищу відносну вартість даних трубок та відсутність організованої офіційної роздрібної мережі дистрибуції LND712 ці трубки важко використовувати у невеличких проектах.

Результати порівнянь різних трубок Гейгера можна побачити в Табл. 3.1.

Таблиця 3.1 — Порівняльні характеристики трубок Гейгера

Властивість	SBM20	J305	LND712
Тип радіації	beta, gamma	beta, gamma	alfa, beta, gamma
Розміри (макс.), мм	d11 x 109	d11 x 107	d15.1 x 49.2
Країна походження	срср або росія	Китай	США
Джерело калібровки виробником	Cs-137	Co-60	Co-60

Властивість	SBM20	J305	LND712
Чутливість	60 – 70 імп / мкР при 4 мкР/с Cs-137або це240-280 імп/с при 4 мкР/с Cs-137	44 імп/с при 1 мР/год Co-60	18 імп/с при 1 мР/год
Час нечутливості (Dead Time)	190 мкс	немає даних	90 мкс
Рівень при фоновому випромінюванні	60 імп./хв	25 імп./хв	немає даних
Власний шум трубки	1 імп./сабо60 імп./хв	0,2 імп/с або 12 імп./хв	0.17 імп./с або 10 імп./хв
Рекомендована робоча напруга живлення	400 В	glass tube 380 Bmetal tube 400 В	500 В

Виходячи з даних таблиці, найкращий результат по характеристикам у трубки LND712. Але оскільки зараз немає підтримки типорозмірів даної трубки, то вибір лишається між SBM20 і J305:

а) термін зберігання / придатності – J305 краща;

б) власний шум – J305 краща;

в) чутливість – J305 краща;

г) країна походження – J305 краща;

д) фонове випромінювання – J305 краща;

е) джерело калібровки – J305 краща (більшість трубок калібрують по Co-60);

ж) час нечутливості – немає різниці;

з) металевий корпус – SBM20 краща;

и) розміри і кріплення – немає різниці;

к) напруга живлення – немає різниці;

л) роздрібна мережа дистрибуції – J305 краща;

м) ціна і якість – J305 краща.

Керуючись даними з документації, статистикою з інтернет та власним досвідом, безумовно, варто обрати J305. SBM20 не варто обирати хоча б за країною походження.

Система збору інформації складається з наступних частин:

Модуль з лічильником Гейгера наведений на рис. 3.4 [8]:

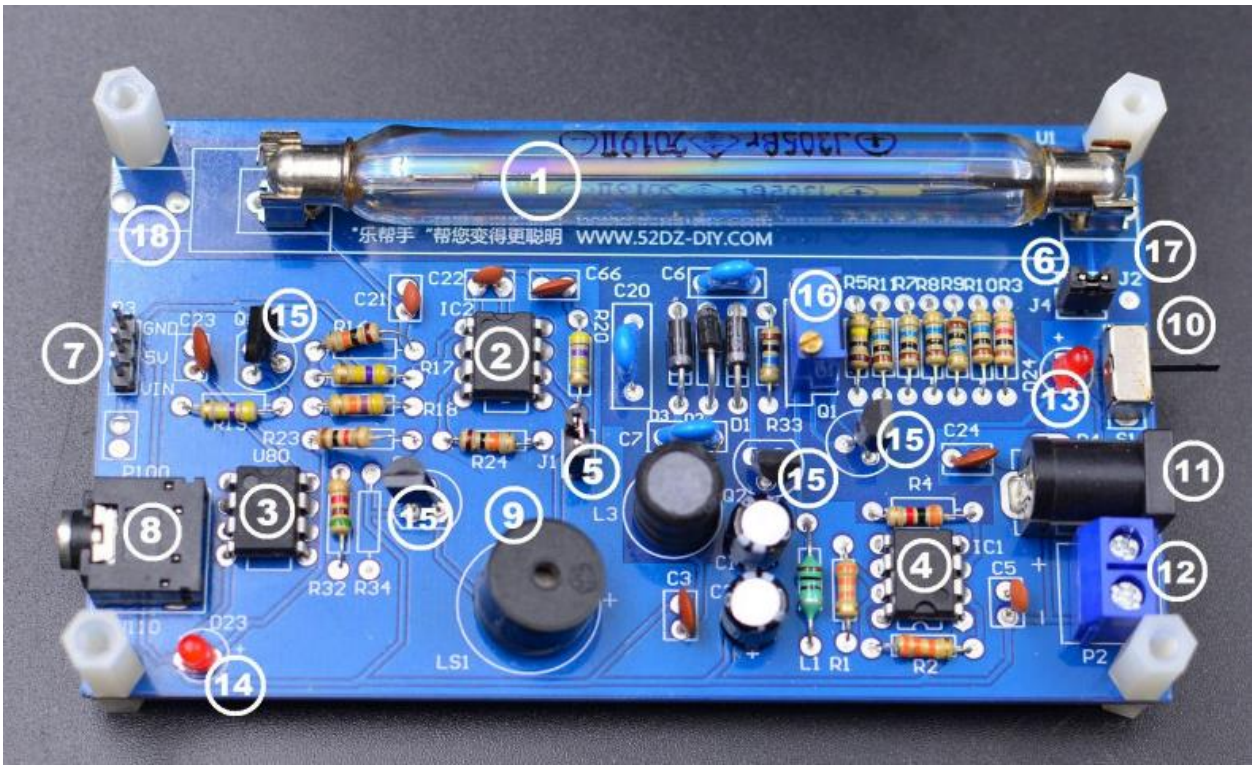


Рисунок 3.4 – Модуль з лічильником Гейгера

Основні компоненти:

1) трубка J305 (аналог M4011) [3], фіксує гама та бета-випромінювання;

2,4) мікросхеми 555, на яких власне і працює даний модуль;

3) LM358P, двоканальний операційний підсилювач;

5) перемичка J1 – відключає буззер, якщо звукова індикація не потрібна;

6) перемичка J4 – використовується для калібрування;

7) контакти 5 В, INT, GND, перший та останній з яких для підключення живлення 4,5–5,5 В. Контакт INT – сигнальний, використовується для підключення до MCU, до роз'єму, налаштованого на отримання зовнішніх переривань, і, відповідно, з подальшою обробкою отриманих імпульсів;

8) роз'єм 3,5 мм AudioJack, потрібен для підключення до аудіороз'єму у смартфонах. Можна на смартфоні завантажити програму та виводити підрахунки на екран;

9) буззер, що видає клацання при реєстрації імпульсу;

10) зсувний вимикач;

11) DC-роз'єм живлення 5 В (5,5 мм × 2,5 мм);

12) контактна колодка для підключення живлення 5 В до модуля;

13) світлодіод, горить постійно, якщо на модуль подається живлення;

14) світлодіод, коротко блимає при реєстрації імпульсу;

15) NPN-транзистори S8050;

16) калібрувальний потенціометр;

17) калібрувальний контакт J2;

18) додаткові контакти під затискач (якщо планується встановити трубку СБМ-20).

Через роз’єм 3,5 мм AudioJack для відлагодження роботи програми можна приєднати смартфон (рис. 3.5):

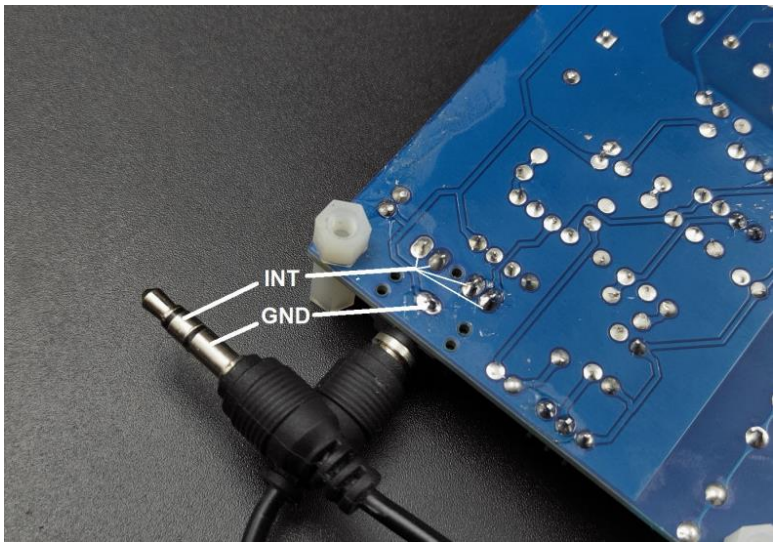


Рисунок 3.5 — Розпіновка роз’єму 3,5 мм AudioJack

Контроль роботи пристрою можна робити за допомогою програми GeigerCounter (рис. 3.6):

Рисунок 3.6 — Інтерфейс програми GeigerCouner

MCU модуль ESP32-CAM [9] з WiFi та Bluetooth, камерою OV2640 та материнською платою, яка необхідна для програмування модуля (рис. 3.7).

Рисунок 3.7 – Модуль ESP32-CAM

Для даної задачі підходить будь-який модуль на основі ESP32 або ESP8266 з вбудованою підсистемою Wi-Fi. Для прототипу був використаний цей модуль, бо він був в наявності.

Блок живлення з напругою 5 вольт. Застосовується як для живлення плати з лічильником Гейгера так і для модуля ESP32.

Перетворювач рівнів логічних сигналів. Плата лічильника Гейгера працює з рівнем логічної одиниці в 5 вольт, модуль ESP32 – з логічним рівнем 3,3 вольта. Для узгодження рівнів використовується проста схема, наведена на рис. 3.8.

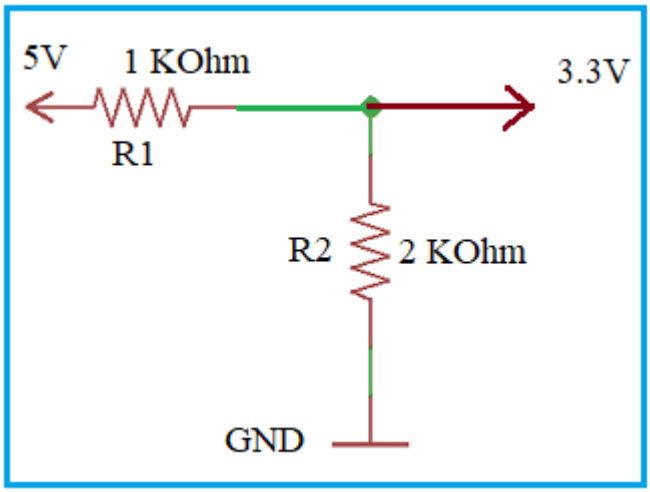


Рисунок 3.8 – Схема перетворення логічних рівнів

Зразок радіоактивної речовини для тестування системи. Це так званий «Енергетичний кулон» китайського виробництва, який, згідно з описом, «покращує енергетичний стан організму» (рис. 3.9). Має слабку радіоактивність.

Рисунок 3.9 – Радіоактивна речовина

При дослідженнях з радіоактивними матеріалами, треба пам’ятати, що ви не можете побачити випромінювання. Тому важливо знати і розуміти їх поведінку. Сполуки, які дають навіть слабке випромінювання, треба зберігати в холодильнику (морозильній камері), за оргсклом, бетонною стіною й т. п. [10].

3.2 Програмна частина

Для організації збору та обробки даних використовується протокол MQTT (Message Queue Telemetry Transport). Це спрощений мережевий протокол, що працює над TCP/IP [11].

В цьому протоколі для обміну даних задіяний принцип «видавець-підписник» (рис. 3.10).

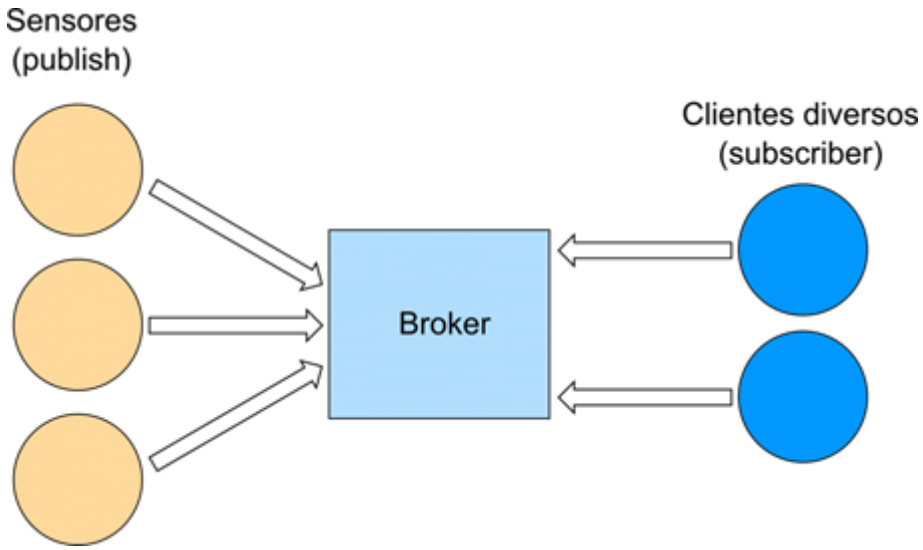


Рисунок 3.10 – Схема роботи протоколу MQTT

Протокол передбачає наявність таких основних компонентів:

а) «видавець» (publish) – це, як правило, пристрої для збирання даних;

б) «підписник» (subscriber) – пристрої-споживачі даних;

в) «брокер» (broker) – пристрій, який відповідає за взаємодію та роботу протоколу.

Результатом цієї роботи є створення «видавця» (publish), який буде взаємодіяти з іншими компонентами через протокол MQTT. Обробка та публікація даних буде відбуватися стандартними засобами протоколу MQTT.

Перевагою плат ESP32 є можливість програмування у середовищі Arduino, з використанням стандартних бібліотек, які написані для цієї платформи.

Програма складається з двох основних частин, наведених у додатку Б.

3.2.1 Обробка даних з лічильника Гейгера

Модуль лічильника видає імпульси, які використовуються для переривання з метою підрахунку їх кількості за хвилину.

Підготовка даних здійснюється за допомогою коду, наведеного на рис. 3.11.

```
#define LOG_PERIOD 20000    //період вивода CPM у мілісекундах
#define MAX_PERIOD 60000    //маскимальний період моніторинга

unsigned long counts;           //змінна для запису кількості імпульсів
unsigned long cpm;              //змінна для CPM (кількість розпадів у хв.)
unsigned int multiplier;        //множник для підрахунку CPM
unsigned long previousMillis;    //змінна для запису часу

float mkzvHours = 0.0; // мкЗв/ч

void tube_impulse(){            //обробник зовнішнього переривання,
  counts++;                    //де йде підрахунок імпульсів за LOG_PERIOD
}

void setup(){                     //Попередні налаштування
  counts = 0;                    //обнуління лічильника імпульсів
  cpm = 0;                       //обнуління CPM
  multiplier = MAX_PERIOD / LOG_PERIOD; //розрахунок множника CPM
  pinMode(2, INPUT);                  //вивід 2 зробити вхідним
  attachInterrupt(0, tube_impulse, FALLING);  //зовнішнє переривання на
}                                                //виводі 2 при зміні рівня
                                                 //з 1 на 0

В основному циклі:

  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis > LOG_PERIOD)
   {                                               //якщо різність
    previousMillis = currentMillis;                //змінних більше LOG_PERIOD,
                                                   //то виводимо CPM
    cpm = counts * multiplier;             //розраховуємо CPM
    mkzvHours = cpm / 151.0;               //перерахунок CPM в мкЗв/год
    counts = 0;                            //Скидаємо лічильник
   }
```

Рисунок 3.11 – Підготовка даних

У наведеному фрагменті програми число 151,0 – коефіцієнт кратності імпульсів – залежить від типу трубки Гейгера; це значення встановлюється для

трубки M4011, але для іншої трубки він буде відрізнятись. Наприклад, для СБМ-20 він становить 0,057.

3.2.2 Публікація даних через протокол MQTT

Спочатку треба задати параметри бібліотек та мережі, що використовується (рис. 3.12).

```
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "yourNetworkName";                //задаємо параметри Wi-Fi
const char* password =  "yourNetworkPassword"; //пароль
const char* mqttServer = "m11.cloudmqtt.com";    //MQTT сервер
const int mqttPort = 12948;
const char* mqttUser = "yourMQTTuser";               //логін
const char* mqttPassword = "yourMQTTpassword";       //та пароль до сервера

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {

 Serial.begin(115200); // вивід відладочних повідомлень у COM-порт
 WiFi.begin(ssid, password);

 while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.println("Connecting to WiFi..");
 }

 Serial.println("Connected to the WiFi network");

 client.setServer(mqttServer, mqttPort);

 while (!client.connected()) {
  Serial.println("Connecting to MQTT...");

  if (client.connect("ESP32Client", mqttUser, mqttPassword )) {

    Serial.println("connected");
```

```
  } else {

    Serial.print("failed with state ");
    Serial.print(client.state());
    delay(2000);

  }
}
```

Рисунок 3.12 – Публікація даних через протокол MQTT

У самій програмі виконується вивід повідомлень (рис. 3.13).

```
client.publish("esp/CMP", CPM); //вивід значення CPM в топік esp/CPM
```

Рисунок 3.13 – Вивід повідомлень за протоколом MQTT

В цій частині створюється з’єднання з MQTT-брокером та передаються показники лічильника Гейгера на заданий сервер.

3.3 Збереження результатів вимірювань IoT-станцій спостереження за радіаційним фоном

Для отримання результатів роботи був використаний брокер Majordomo, який встановлений на NAS Synology DS-220+, показаний на рис. 3.14 [12].

Рисунок 3.14 – NAS Synology DS-220+

3.4 Аналіз результатів

Сторонньою програмою-«підписником» був сформований CSV-файл, в якому збереглися дані вимірювань з кроком в 1 хвилину.

Результати вимірювань відображені на графіку, де по осі X відкладений час у хвилинах, а по осі Y — значення радіоактивності у мкЗв/год (рис. 3.15).

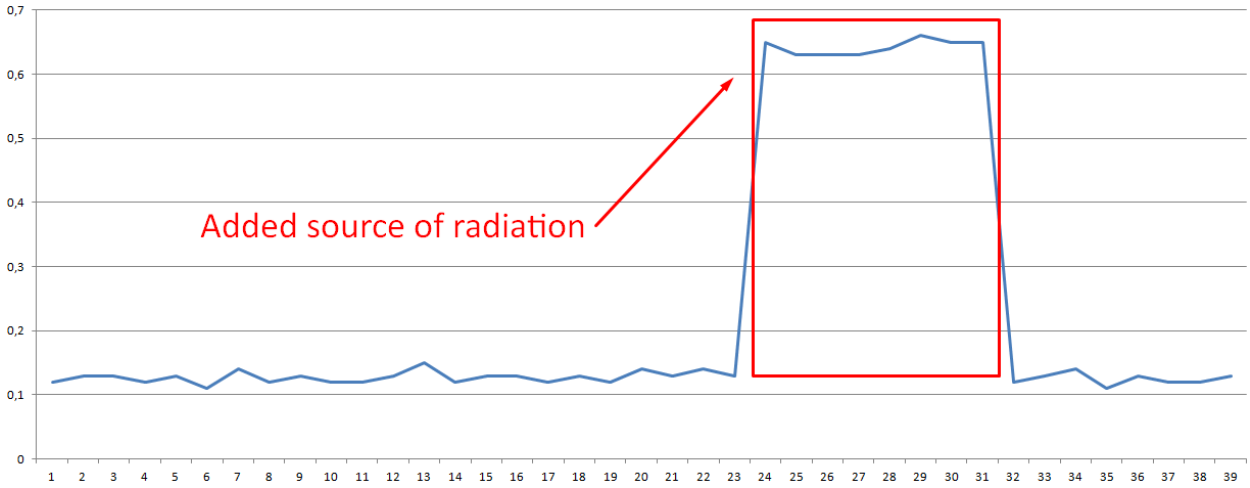


Рисунок 3.15 – Результати вимірювань

На графіку можна побачити, що при піднесенні радіоактивної речовини до трубки Гейгера, показники радіоактивності збільшуються до значень 0,61–0,65 мкЗв/год з фонових природних значень 0,11–0,14 мкЗв/год.

Отримане значення всплеску в 0,65 µSv/h на графіку рис. 3.15 перевищує санітарну норма для населення (яка становить 0,30 µSv/h) майже вдвічі, тому потребує вжиття спеціальних заходів з боку відповідних служб.

Це показує, що всі елементи ланцюжка «видавець» – «брокер» – «підписник» працюють вправно.

Висновки за розділом 3

Отже, IoT-пристрої на основі лічильників Гейгера можуть бути розташовані в різних місцях та об'єднані в спільну мережу з передачею даних на сервер за допомогою одного з протоколів IoT, наприклад, MQTT.

Однак складність такої системи становить найбільшу проблему, оскільки операції з IoT є комплексними, і не існує гнучкої інтеграції між пристроями. Система складається з різних пристроїв з різною архітектурою, реалізацією та обслуговуванням, тому будь-яка вразливість в програмному або апаратному забезпеченні одного пристрою може мати серйозні наслідки для багатьох інших пристроїв у мережі.

Мережа Інтернету речей стикається з проблемами автентифікації та контролю доступу, оскільки розумні об'єкти є пристроями різних типів, що базуються на різних платформах (апаратних засобах та мережах). Крім того, всі пристрої повинні взаємодіяти один з одним через різні мережі. Тому проблема безпеки є основною, оскільки всі пристрої та дані піддаються різним видам загроз і атак.

Існує багато різноманітних загроз та атак, які можуть спричинити серйозні проблеми в мережі. Наразі для IoT відсутні стандарти та правила, що пояснюють, як захищати пристрої та дані. Тому перспективою розвитку цієї роботи може бути побудова системи керування безпекою мережі IoT для реалізації відповідних механізмів безпеки на різних рівнях архітектури IoT.

ВИСНОВКИ

Застосування інформаційних технологій та засобів автоматизації вимірювання радіаційного фону у наведеному дослідженні дозволить своєчасно визначати рівень іонізуючого випромінювання, що становить загрозу здоров’ю людей, та приймати рішення щодо евакуації. Вбачається, що зазначена територія вкрита мережею розроблених станцій спостереження, з яких у реальному часі буде надходити актуальна інформація про радіаційний фон до центру прийняття рішень. У роботі також вивчені можливості легкого доступу до дозиметричної інформації через Інтернет з комп'ютера або смартфона.

Таким чином, в результаті роботи на основі проаналізованих існуючих аналогів та дібраної елементної бази виконана:

– розробка автоматизованої системи моніторингу радіаційного фону навколишнього середовища, здатної обробляти дані, отримані з необмеженої кількості стаціонарних або мобільних точок

– передача вимірювань датчиків з точки моніторингу на сервер через мережу Інтернет на основі економного протоколу MQTT, що забезпечує низьке енергоспоживання та масштабованість;

– розробка мобільного застосунку, в якому реалізовані попереджувальні та аварійні сповіщення при перевищенні порогових рівнів радіаційного фону.

Розроблена апаратно-програмна система збору і передачі значень радіоактивності дозволяє накопичувати та оброблювати великі обсяги даних на віддалених мережевих ресурсах. Практична значимість результатів роботи полягає у можливості побудування щільної мережі простих у розробці та обслуговуванні IoT-пристроїв для моніторингу рівня радіаційного фону, що дозволить оперативно виявляти небезпеку для здоров'я людей. Зібрані та передані до серверу дані нададуть можливість розробити інтерактивну карту, яка інформуватиме громадян та місцеву владу про будь-які проблеми, пов'язані з радіоактивністю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. S. Ovchar, V. Mohyla, B. Saltovskyi, S. Puzyrov, "Development of an IoT device based on a Geiger counter," Automation of technological and business processes, vol. 15, is. 2, 2023. ISSN-print: 2312-3125 *(прийнята до друку).*

2. A. K. Bollfrass and S. Herzog, "The War in Ukraine and Global Nuclear Order," Survival, vol. 64, no. 4, pp. 7–32, Aug. 2022, doi: 10.1080/00396338.2022.2103255.

3. Б. Ю. Жураковський, І.О. Зенів, Технології інтернету речей : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 271 с.

4. G. C. Hillar, MQTT Essentials–A Lightweight IoT Protocol. Birmingham, U.K. : Packt Publishing Ltd., 2017.

5. C. Son, B. Ziaie, "Electret Based Wireless Micro Ionizing Radiation Dosimeter," in Proc. 19th IEEE Int. Conf. on Micro Electro Mechanical Systems, Istanbul, Turkey, 2006, pp. 610–613, doi: 10.1109/MEMSYS.2006.1627873.

6. E. Heijne, T. Koi, C. Leroy, H. Oberlack, S. Pospisil, et al., "Comparison of measurement and simulation of ATLAS cavern radiation background," Journal of Instrumentation, vol. 17, no. 01, pp. 1–33, Jan. 2022, Art. no. P01027, doi: 10.1088/1748-0221/17/01/P01027.

7. V. I. Vytko, L. I. Honcharova, V. V. Kartashev, H. D. Kovalenko, S. A. Seheda, and S. V. Barbashev, "Automated radiation control system as the main component of radiation safety of the population," Nuclear and radiation safety, vol. 3, no. (59), pp. 33–37, 2013. (In Ukrainian).

8. L. Devell and B. L. Riso, Radiological emergency monitoring systems in the Nordic and Baltic Sea countries. Roskilde: NKS, 2002.

9. S. I. Voronov, E. V. Popov, V. A. Sednev, and O. S Voronov, "Public safety conditions under radiological emergencies monitoring comprehensive system mobile facilities application," IOP Conf. Series: Earth and Environmental Science 843, 2021, 012049, doi:10.1088/1755-1315/843/1/012049.

10. M. I. Ahmad, M. H. Ab. Rahim, R. Nordin, et al., “Ionizing radiation monitoring technology at the verge of Internet of Things,” Sensors 21(22):7629, 2021, doi: 10.3390/s21227629.

11. UK Cabinet Office. National Risk Register of Civil Emergencies. (2012). [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/211858/CO_NationalRiskRegister_2012_acc.pdf

12. N. M. Remon, C. M. Hassan, M. Hassan, and M. Zakir, “Build very simple design and cost effective Geiger-Muller counter,” Journal of Recent Advances in Applied Sciences, vol. 31, no. 1, pp. 1−10, Apr. 2018.

13. В. Л. Терьохін, М. Г. Стервоєдов, О. В. Рідозуб, “Застосування технологій IoT та хмарних сервісів для радіаційного моніторингу,” Системи керування та комп'ютери, № 2–3, С. 60−68, 2021, doi: 10.15407/csc.2021.02.060.

14. Module Wi-Fi ESP32-CAM with camera 2MP. Accessed: Apr. 23, 2023. [Online]. Available: https://arduino.ua/prod3458-modyl-wi-fi-esp32-s-kameroi-2mp

15. B. R. Sveinbjornsson and S. Gizurarson, “Radioactive materials,” in Handbook for Laboratory Safety Handbook for Laboratory Safety. Elsevier, 2022, pp. 101−111, doi: 10.1016/b978-0-323-99320-3.00014-8.

16. EcoCity − мережа громадського моніторингу якості повітря за допомогою пристроїв на Arduino. Публ. 9 липня 2019 р. [Онлайн]. URL: https://dou.ua/lenta/articles/dou-projector-ecocity/.

17. P. S. Akshatha, S. M. D. Kumar, and K. R. Venugopal, “MQTT implementations, open issues, and challenges: A detailed comparison and survey,” International Journal of Sensors, Wireless Communications and Control, vol. 12, no. 8, pp. 553–576, 2022, doi: 10.2174/2210327913666221216152446.

18. NAS Synology DS-220+. Accessed: Apr. 23, 2023. [Online]. Available: https://www.synology.com/en-eu/products/DS220+

ДОДАТОК А

Довідка про антиплагіат

ЗВІТ

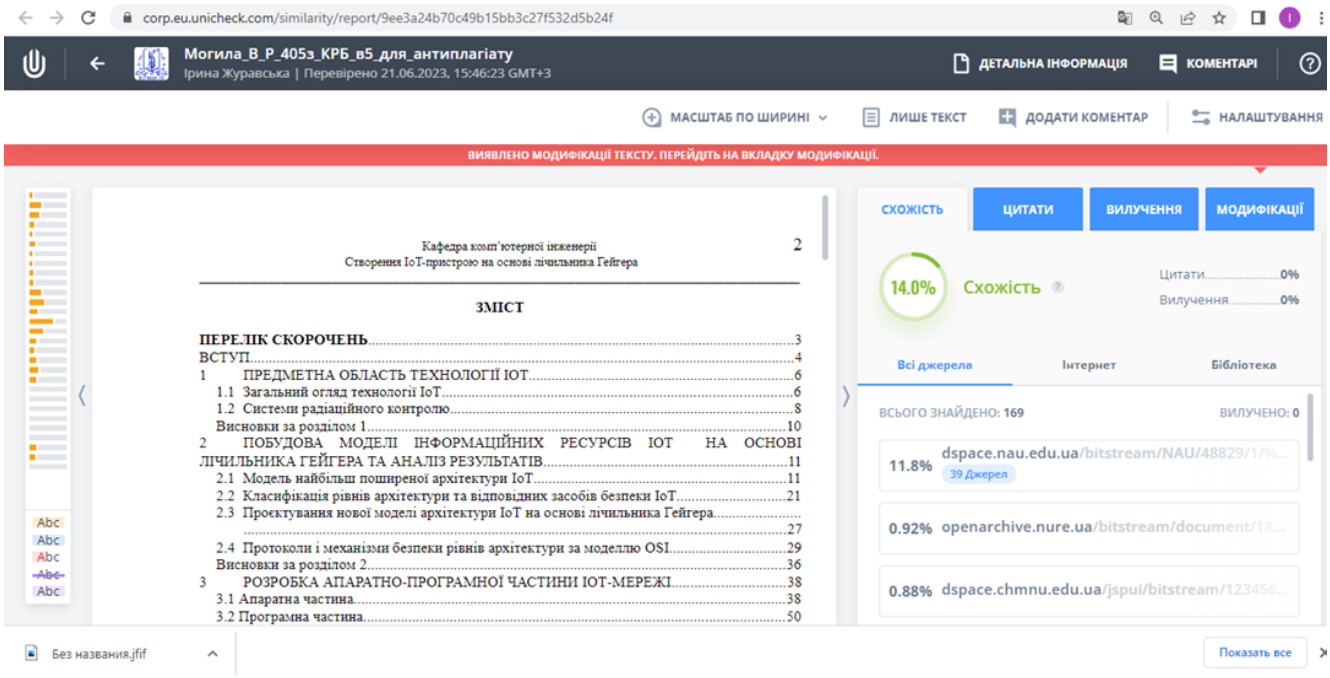
про унікальність пояснювальної записки
бакалаврської кваліфікаційної роботи на тему:
«Створення IoT-пристрою на основі лічильника Гейгера»

студента спеціальності 123 «Комп'ютерна інженерія», групи 405з

Могили Владислава Романовича
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту бакалаврської кваліфікаційної роботи: схожість складає 14,0 %.



Студент:

_____ В. Р. Могила
підпис ініціали, прізвище

Керівник:

Ст. викладач

_____ Б. Г. Салтовський
підпис ініціали, прізвище

Дата: «21» червня 2023 р.

ДОДАТОК Б

Матеріали апробації роботи