Вінницький національний технічний університет Факультет
інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«МІКРОКОМП'ЮТЕРНА СИСТЕМА ОПОВІЩЕННЯ ТА КОНТРОЛЮ ЦІЛІСНОСТІ ОХОРОННОГО ОБ'ЄКТА ЗАСОБАМИ IoT»

Виконав: студент 2 курсу, групи 1КІ-22м
спеціальності 123 Комп'ютерна інженерія
Твердохліб Н.М
Керівник к.т.н., доц. каф. ОТ
Богомолов С.В.
Опонент д.ф. (PhD), доц. каф. МБІС
Салієва О.В.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.
«13» 12 2023 р.

Вінниця 2023

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень другий (магістерський)
Галузь знань — 12 Інформаційні технології
Спеціальність — 123 «Комп'ютерна інженерія»
Освітня програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач каф. ОТ
д.т.н. проф., О. Д. Азаров
«26» вересня 2023 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Твердохлібу Назару Миколайовичу

1 Тема роботи «Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT», керівник роботи Богомолов Сергій Віталійович, к.т.н., доцент,затверджені наказом вищого навчального закладу від 26.09.2023 року № 247.

2 Строк подання студентом роботи 18.12.2023 року.

3 Вихідні дані до роботи: опис бездротових технологій передачі даних, технічний опис Raspberry Pi, та електронних компонентів.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз теоретичних аспектів IoT, проектування апаратної та програмної частин, дослідження та тестування системи, висновки

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) структурна схема, розпінування Raspberry Pi.

6 Консультанти розділів роботи представлено в табл. 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3,4	Богомолов С.В., к.т.н., доцент кафедри ОТ		
5	Небава М.І. к.е.н, Професор кафедри ЕПВМ		

7 Дата видачі завдання 19.09.2023

8 Календарний план наведено в табл.2.

Таблиця 1 — Календарний план

з/п	Назва етапів кваліфікаційної роботи	Термін виконання роботи	Примітка
1	Постановка мети та задач проекту	17.10.23	Вик.
2	Огляд і аналіз	20.10-25.10.23	Вик.
3	Огляд і аналіз теоретичних аспектів IoT	26.10-3.11.23	Вик.
4	Вибір мікрокомп'ютерної платформи	4.11-11.11.23	Вик.
5	Проектування апаратної та програмної частин	12.11-20.11.23	Вик.
6	Дослідження та тестування системи	21.11-26.11.23	Вик.
7	Розрахунок економічної частини роботи	26.11-4.12.23	Вик.
8	Аналіз виконання проекту. Висновки. Додатки	5.12-6.12.23	Вик.
9	Оформлення пояснювальної записки та ілюстративного матеріалу	7.12.23	Вик.
10	Перевірка якості виконання бакалаврського проекту та усунення недоліків	18.12.23	Вик.

Студент _____ Твердохліб Н.М.

Керівник роботи _____ Богомолов С.В.

АНОТАЦІЯ

УДК 654.9

Твердохліб Н. М.

Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT. Магістерська кваліфікаційна робота зі спеціальності 123 — комп’ютерна інженерія, освітня програма — комп’ютерна інженерія. Вінниця: ВНТУ, 2023 — 124 с.

На укр.мові. Бібліогр.: 20 назв, рис. 18, табл. 6.

Дана магістерська кваліфікаційна робота присвячена створенню системи оповіщення та контролю цілісності засобами IoT.

Одним із найбільш актуальних напрямів розвитку охоронних систем в останні роки став Інтернет речей.

Завдяки розвитку технологій у теперішній час можна легко вирішити проблеми безпеки.

Розроблена система допоможе покращити безпеку, шляхом впровадження відеофіксації та ріхноманітних датчиків, автоматизувати передачу отриманої інформації у базу даних що їй відповідає. Таке рішення покращить надійність проектованой системи, та швидкість взаємодії з користувачем.

Ключові слова: модулі, датчики, пристрій, мікрокомп’ютерна система, IoT, ESP 32, Raspberry Pi, MQTT.

ANNOTATION

Tverdokhlib N.M.

Microcomputer-based notification and integrity control system for security objects using IoT tools. Master's qualification work in the field of 123 — computer engineering, educational program - computer engineering. Vinnytsia: VNTU, 2023, 124 p.

In the Ukr. leng. Libr. name 20, figure 18, table 6.

This master's qualification work is dedicated to the development of a notification and integrity control system using IoT tools.

One of the most relevant directions in the development of security systems in recent years is the Internet of Things.

Thanks to the advancement of technologies, contemporary security issues can be easily addressed.

The developed system will enhance security by implementing video surveillance and motion sensors, automating the transfer of received information to the corresponding database. Such a solution will improve the reliability of the designed system and the speed of interaction with the user.

Keywords: modules, sensors, device, microcomputer system, IoT, ESP 32, Raspberry Pi, MQTT.

ЗМІСТ

					08-54.МКР.017.00.000 ПЗ			
		№ докум.	*Підпис*					
Розробив		*Твердохліб Н.М.*			Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT	*Літ.*	*Арк.*	*Аркушів*
Керівник		*Богомолов С.В.*					*6*	
Опонент		*Салієва О.В.*				*ВНТУ, гр. 1КІ-22м*		
Н. Контроль		*Швець С. І.*						
Затверджую		*Азаров О. Д.*			Пояснювальна записка			

ВСТУП

У сучасному світі, який заснований на стрімкому розвитку інформаційних технологій, виникає необхідність ефективного забезпечення безпеки об'єктів з різноманітних сфер життєдіяльності. Інтернет речей (IoT) визначає новий етап в еволюції сучасних технологій, де об'єднуються фізичний та цифровий світи для створення інтелектуальних систем. У цьому контексті, проблема охорони та контролю цілісності об'єктів стає важливою, оскільки зростаюча кількість ризиків та загроз вимагає нових, ефективних рішень для забезпечення безпеки.

Об'єктами охорони можуть бути будь-які простори або території, які вимагають постійного контролю і реагування на непередбачені події. З метою підвищення ефективності цих процесів та забезпечення швидкого реагування на потенційні небезпеки, виникає необхідність у створенні інноваційних мікрокомп'ютерних систем оповіщення та контролю цілісності, які базуються на концепції Інтернету речей.

Дипломна робота присвячена розробці та впровадженню мікрокомп'ютерної системи, яка забезпечує комплексний підхід до забезпечення безпеки об'єктів за допомогою сучасних засобів IoT. Основною метою дослідження є розробка інтелектуальної системи, що поєднує в собі мікрокомп'ютерні технології та засоби збору та обробки даних для надійного виявлення подій, що порушують цілісність об'єкта, та оперативного інформування відповідальних осіб.

На фоні високої динаміки розвитку Інтернету речей і високих вимог до безпеки, дослідження спрямоване на розробку та оптимізацію алгоритмів виявлення інцидентів, а також створення засобів взаємодії мікрокомп'ютерної системи з іншими елементами безпекової інфраструктури. Результати цього дослідження можуть знайти практичне застосування в різних галузях, де важлива проблема забезпечення цілісності та безпеки об'єктів.

Дипломна робота розкриє теоретичні та практичні аспекти розробки мікрокомп'ютерних систем оповіщення та контролю цілісності, надаючи детальний огляд існуючих технологій та розглядаючи їх можливі застосування в контексті вирішення конкретних задач безпеки об'єктів за допомогою Інтернету

речей.

Метою магістерської кваліфікаційної роботи є вдосконалення мікрокомп'ютерної системи оповіщення та контролю цілісності охоронного об'єкта за допомогою функції передачі даних стану бездротовою мережею LoRaWAN.

Задачі дослідження:

— проаналізувати існуючі системи оповіщення та контролю цілісності та методи їхньої реалізації;

— проведення огляд та аналіз методів реалізації мікропроцесорних систем;

— обрати технологію бездротової передачі інформації;

— здійснити підбір комплектуючих елементів;

— спроектувати мікрокомп'ютерну систему оповіщення та контролю цілісності охоронного об'єкта;

— здійснити реалізацію проекту з робочим програмним кодом;

— проведення тестування системи.

Новизна одержаних результатів полягає у вдосконалені системи оповіщення та контролю цілісності охоронного об'єкта за рахунок введення функції передачі даних бездротовою мережею LoRaWAN, що дозволяє забезпечити підвищення дальності спрацювання та зменшити енергоспоживання.

Об’єкт дослідження — процеси, що протікають у система оповіщення та контролю цілісності охоронного об'єкта засобами IoT.

Предмет дослідження — методи та засоби реалізації систем із використанням IoT та бездротових мереж технології LoRaWAN.

Апробацію результатів наукової роботи було проведено на науковій конференції:

«П’ятдесят першій науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії 2022», доповідь на тему

“Автоматизована система управління пропускною охоронною системою”.

1 АНАЛІЗ ТЕОРЕТИЧНИХ АСПЕКТІВ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Огляд історії Інтернету речей

Інтернет речей (IoT) представляє собою мережу взаємодіючих пристроїв, що обмінюються даними, не вимагаючи прямого втручання користувачів. Кевін Ештон, британський технолог, вперше визначив термін "Інтернет речей" в 1999 році, порівнюючи еру гаджетів, які автоматично збирають та передають дані, з попереднім підходом, коли людина вручну завантажувала дані на комп'ютер.

Ідея обміну інформацією між пристроями без участі людини виникла в кінці 70-х років, але повна автоматизація передачі даних взяла свій початок з ініціативи Кевіна Ештона, який використовував технологію Radio Frequency Identification (RFID) для оптимізації виробничого процесу.

Спочатку важко було усвідомити ідею Інтернету речей, але з розвитком технологій та впровадженням RFID, алгоритмів та сенсорів ця концепція стала реальністю. У 2008 році IPSO Alliance був створений для підтримки розвитку технологій Інтернету речей, що визначило новий напрямок в розвитку інформаційних технологій.

У 2010 році Google StreetView став прикладом збору даних для Інтернету речей, а Китай визначив розробку IoT серед пріоритетів наступних п'ять років. Це свідчить про те, що інтерес до збору, обробки та зберігання даних поширюється не тільки на корпорації, але й на уряди всього світу.

У 2011 році, Gartner, визначаючи найбільш перспективні технології, включила Інтернет речей в свій список. З того часу ця концепція почала ширитися, і вже у 2012 році на конференції LeWeb, найбільшій європейській події в сфері Інтернету, була присвячена Інтернету речей. Також в цей період видання Forbes, Fast Company і Wired активно використовували термін "Інтернет речей". Корпорації розпочали гонку за новими технологіями, а дослідження IDC прогнозувало зростання ринку Інтернету речей до 8,9 трильйонів доларів до 2020 року.

У січні 2014 року Google придбав компанію Nest за 3,2 мільярда доларів, яка розробляла смарт-пристрої та системи управління будівлями. Цей крок

підкреслив важливість Інтернету речей на світовому ринку. Того ж року на виставці CES в Лас-Вегасі, найбільшій американській виставці електроніки, гасло було приурочено до Інтернету речей, починаючи еру нових можливостей.

Розвиток Інтернету речей базується на зборі, обробці, передачі та зберіганні даних. Технології M2M (від машини до машини) та розробка протоколів передачі даних визначили напрямок розвитку. Промисловий Інтернет речей, орієнтований на оптимізацію виробничих процесів, враховує як машинні, так і людські аспекти. Зараз важливою є ідея "Загального Інтернету" або Інтернету Всього (IoE), яка не розрізняє між людиною та пристроєм, наголошуючи на обміні даними, незалежно від участі людини.

В рамках Четвертої Промислової Революції Інтернет речей визначається як невід'ємна частина, співпрацюючи з іншими технологіями, такими як віртуальна та доповнена реальність, великі дані, тривимірне друкування, блокчейн та квантові обчислення. Він трансформує не лише виробництво, але й світогляд, викликаючи інтерес та розгортаючи можливості в різних сферах, включаючи медицину та "розумний будинок".

1.2 Інтернет речей в сучасному світі

Технологія Інтернету речей (IoT) стала однією з ключових технологічних тенденцій за останні кілька років, і її вплив охопив багато сфер. Пандемія Covid-19 стала великим викликом для різних галузей, але IoT виявився ключовою силою, допомагаючи вирішувати виклики цього періоду.

Покращення продуктивності роботи з дому завдяки IoT, використання голосових помічників, таких як Siri, Alexa, Google Home, і інших IoT-сумісних додатків, стало необхідною частиною щоденної рутини. Ці пристрої стали не тільки помічниками у побуті, але й частиною нової реальності роботи з дому. Автоматизоване планування та інтелектуальні календарі значно полегшили управління робочими активностями.

Сектор охорони здоров'я повністю використовує переваги технології IoT. Вона дозволяє поліпшити якість обслуговування, забезпечує цілодобовий

моніторинг пацієнтів поза лікарнею, а також сприяє задоволенню клієнтів та доступності послуг.

У роздрібній торгівлі розумні пристрої та підключені гаджети роблять магазини та супермаркети безпечнішими та ефективнішими. Застосування IoT в роздрібній торгівлі дозволяє вдосконалити управління ланцюгами постачання, покращити управління запасами та автоматизувати багато процесів. Технологія також допомагає роздрібним торговцям автоматизувати управління запасами та розробляти персоналізовані рішення для клієнтів.

Розширення розумних міст за допомогою IoT. У найближчі роки очікується збільшення ініціатив для створення розумних міст. IoT великою мірою впливає на функціонування міст, забезпечуючи інтелектуальний моніторинг дорожнього руху, планування громадського транспорту та краще управління муніципальними зручностями. Сенсори і RFID-мітки сприяють управлінню вторинними матеріалами та сортуванню відходів.

Автомобільна промисловість має значний потенціал для використання IoT. Технологія V2X (від автомобіля до всього) розвивається, дозволяючи прогнозуване технічне обслуговування, системи інфотейнменту в автомобілі та відповідність екологічним і транспортним правилам.

Використання IoT для сучасного сільського господарства та екології може покращити його ефективність та стійкість. Застосунки включають сільськогосподарські дрони для картографування та обстеження ферм, а також датчики для точного моніторингу умов в теплицях та для вирощування худоби.

1.3 Особливості роботи Інтернету речей

Інтернет речей (IoT) – це надзвичайно складна система, що забезпечує взаємодію між різнорідними пристроями та програмами. Особливості роботи Інтернету Речей визначаються не лише технічними аспектами, але й враховують соціально-економічні та екологічні фактори. Інновації в цьому напрямку спрямовані на подальше поліпшення функціональності та ефективності систем IoT у всіх їхніх проявах.

Архітектура IoT включає в себе кілька рівнів, що взаємодіють між собою. Рівень обладнання включає датчики, актуатори та вбудовані пристрої, які забезпечують збір даних з навколишнього середовища.

Рівень мережі відповідає за передачу цих даних, використовуючи різноманітні технології, такі як безпровідні мережі та протоколи передачі. Рівень обчислення здійснює аналіз та обробку даних, використовуючи хмарні та розподілені обчислювальні ресурси.

Гетерогенність — одна з основних особливостей є гетерогенність пристроїв, яка означає різноманітність їх обладнання, архітектур та функціоналу. В мережі IoT можуть брати участь різні пристрої, починаючи від малих датчиків і закінчуючи потужними серверами. Ця гетерогенність вимагає розробки універсальних протоколів та стандартів для забезпечення ефективної комунікації між ними.

Багато пристроїв IoT мають мобільні характеристики, що вносять складність у їх управління та обслуговування. Процеси реалізації та відстеження мобільних пристроїв, таких як автомобілі чи носимі гаджети, вимагають додаткових алгоритмів та механізмів для забезпечення постійної доступності та коректності даних.

Деякі застосування IoT вимагають обробки даних в реальному часі. Наприклад, системи моніторингу здоров'я або системи безпеки. Це вимагає швидкої передачі та обробки даних в мережі, а також низької затримки, що представляє додатковий виклик для розробників та архітекторів систем IoT.

Багато пристроїв IoT працюють на обмежених енергетичних ресурсах, таких як батареї або енергоспоживаючі датчики. Тому ефективне управління енергією стає ключовою властивістю для забезпечення тривалої роботи пристроїв та зменшення впливу на навколишнє середовище.

Інтеграція з штучним інтелектом — одна з перспектив розвитку IoT є його інтеграція з штучним інтелектом (ШІ). Використання алгоритмів машинного навчання та аналізу даних може дозволити системам IoT автоматично адаптуватися до змінних умов та оптимізувати свою роботу.

1.4 Бездротові технології Інтернету речей

Важливим аспектом при створенні мережі є питання взаємодії пристроїв. Рішення цього питання обумовлено завданнями об'єкта. Основні критерії вибору проекту включають:

— розгляд дистанції використання – чи буде мережа охоплювати офіс, чи ціле місто;

— визначення оптимальної частоти, яка уникне перешкод і шумів, що можуть впливати на роботу мережі;

— врахування пропускної здатності для передачі даних та частоти оновлення даних;

— визначення джерела енергії – чи подається енергія від мережі чи використовуються акумулятори;

— забезпечення безпеки при передачі, обробці та зберіганні даних.

Кожен бездротовий пристрій для передачі даних має свої параметри, такі як швидкість, радіус дії та енергоефективність. Проте, важливо враховувати, що одночасно відповідати усім вимогам може бути складно (рисунок 1.1).

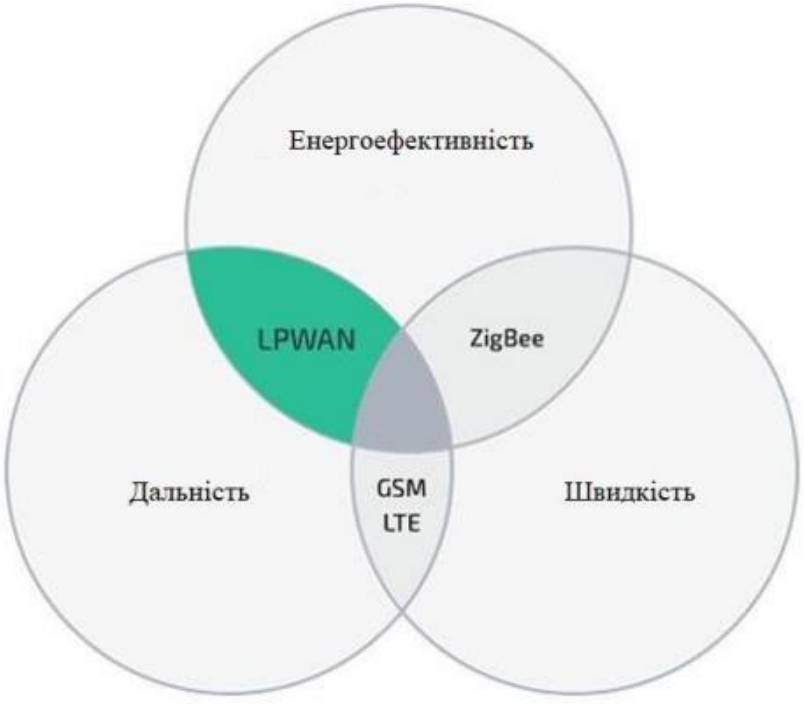

Рисунок 1.1 — Характеристики протоколів передачі

Технологія повинна бути глобальним стандартом, щоб гарантувати доступність та сприяти інноваціям, уникаючи патентованих рішень. Деякі технології ближньої дії, такі як Wi-Fi, Bluetooth і Zigbee, можуть бути більш витратними. Однак, їхні можливості обмежені коротким радіусом, що не відповідає вимогам для додатків, які потребують дальнього покриття, наприклад, автомобілі, датчики стеження та інші (рисунок 1.2). Вони оптимальні для домашніх та офісних умов.

Рисунок 1.2 — Порівняння дальності роботи різних мереж

Технологія Wi-Fi, що походить від англійського вислову "Wireless Fidelity" і буквально перекладається як "висока точність безпроводової передачі даних", постійно розвивається, пропонуючи різні стандарти для передачі цифрових потоків даних через радіоканали. Це передова безпровідна технологія, яка з'єднує пристрої в локальну мережу та забезпечує їхнє підключення до Інтернету. Завдяки цій технології Інтернет стає мобільним, надаючи користувачам волю рухатися як у межах однієї кімнати, так і по всьому світу.

Основний принцип роботи Wi-Fi полягає у використанні радіохвиль з частотою 2,4 МГц і 5 МГц для передачі даних між адаптером та маршрутизатором в мережі. Використовуються 13 частотних каналів у діапазоні 2,4 ГГц та 4 частотних смуги, загалом 23 канали, у діапазоні 5 ГГц. На сучасний момент існує безліч стандартів Wi-Fi з різними технічними характеристиками, такими як

відстань покриття, швидкість тощо. Серед найпопулярніших - стандарти 802.11b, g, n, ac. Стандарт 802.11a не набув широкого поширення в країнах Європи. Стандарти 802.11b та g працюють на частоті 2,4 ГГц зі швидкістю передачі даних 11 Мбіт/с та 54 Мбіт/с відповідно. Стандарт 802.11n, затверджений у 2009 році, може працювати як на частоті 2,4 ГГц, так і 5 ГГц, використовуючи ортогональне частотне 32-мультиплексне утворення сигналу, подібно до 802.11g. Максимальна швидкість передачі становить 600 Мбіт/с (4 приймаючі антени) та 150 Мбіт/с (1 приймаюча антена). Стандарт 802.11ac використовує частоту 5 ГГц і забезпечує передачу даних із швидкістю до 1,3 Гбіт/с.

Проте, розглянемо, як ця технологія забезпечує безпеку в системах. Наприклад, сучасні системи охорони з Wi-Fi сигналізацією нового покоління використовують інноваційну технологію обміну даними через Інтернет між Wi-Fi сигналізацією та смартфонами користувачів. Хоча перші системи Wi-Fi сигналізації з'явилися на ринку охоронного обладнання недавно, їх розширені можливості та зручність у використанні дозволили їм швидко завоювати лідерські позиції серед систем самостійного встановлення.

Звісно, важливо враховувати як позитивні, так і від'ємні аспекти Wi-Fi сигналізації. Цей вид зв'язку працює через Інтернет і встановлює бездротовий зв'язок лише із нашим роутером, який, у свою чергу, зазвичай отримує зв'язок по кабелю.

При розгляді Інтернету речей основною задачею є підключення великої кількості IoT-пристроїв до глобальної мережі. Очевидно, що бездротовий зв'язок стане основою для цього, і питання полягає в тому, яка технологія буде найбільш оптимальною. Зараз існує багато варіантів, таких як Wi-Fi, Bluetooth, LTE на базі стільникового зв'язку. Вибір конкретної технології буде залежати від різних факторів, таких як область застосування, діапазон і смуга частот, пропускна здатність каналу передачі даних і тривалість автономної роботи.

Однією з переваг технології Wi-Fi є її можливість працювати на великій відстані, використовуючи різні частоти та типи модуляції. З'являються постачальники, які використовують цю технологію для забезпечення

безпровідного зв'язку на великій відстані, використовуючи ефективні антени. Також використовується технологія Power over Ethernet (PoE), що дозволяє передавати електроенергію та дані через Ethernet-з'єднання.

Хоча такі системи ідеально підходять для великих відстаней, вони, швидше за все, не підходять для вузлових IoT-пристроїв через споживання енергії. Тим не менш, Wi-Fi залишається популярним в області розумного будинку, де він широко застосовується завдяки своїм перевагам та майже відсутності обмежень щодо енергоспоживання, що є менш критичним для інтелектуальних домашніх пристроїв. Навіть Wi-Fi з малим енергоспоживанням може знайти своє використання в додатках, які вимагають періодичної передачі даних з низькою швидкістю.

Розглядаючи області використання, такі як сенсорне обладнання та лічильники в "розумних будинках", важливо враховувати як переваги, так і недоліки технології Wi-Fi.

Зокрема, варто зазначити, що Wi-Fi має свої обмеження. По-перше, високий рівень енергоспоживання відзначається як один із недоліків. Також існують проблеми, пов'язані з його роботою в перенасичених спектрах частот, що не вимагають ліцензії. Це може викликати підвищений рівень перешкод. Додатково, бездротове підключення до Інтернету, хоча і зручне, створює додаткові виклики в області кібербезпеки, вимагаючи особливої уваги до конфіденційності даних та правильної роботи кінцевих додатків.

З прихідом Інтернету речей і масового підключення IoT-пристроїв, виникають нові виклики. Багато IoT-пристроїв підключаються до Wi-Fi мереж зі спрощеними політиками безпеки, що стає причиною атак типу DDoT (DDoS of Things). Ця проблема залишається невирішеною.

Для вирішення цих питань постачальникам Wi-Fi-обладнання слід акцентувати увагу на розробці програмного забезпечення, яке обмежує підключення IoT-пристроїв лише через дозволені порти та протоколи. Застосування хмарних технологій також ускладнює вирішення цих проблем, адже забезпечення неперервного доступу до хмарного сервера стає ключовим. У цьому

контексті Wi-Fi-мережі повинні здатися на безперебійний моніторинг, управління та самовідновлення.

Технологія Wi-Fi вже готова до підключення мільярдів IoT-пристроїв, забезпечуючи стійкість, гнучкість і придатність для багатоцільового використання. Її схильність до функціональної сумісності пристроїв, які використовують цю технологію, робить Wi-Fi однією з найідеальніших платформ для інновацій в різних сферах технологій Інтернету речей.

Протокол зв'язку Bluetooth, задуманий ще в середині 1990-х років для організації персональних локальних мереж, що об'єднують різноманітні пристрої, від мобільних телефонів до комп'ютерної периферії, залишається активною та постійно розвиваючоюся технологією. Розроблений у рамках стандарту IEEE 802.15.1, Bluetooth пройшов довгий шлях від свого запуску, зокрема завдяки зусиллям альянсу Bluetooth Special Interest Group (Bluetooth SIG), який об'єднує численні компанії, працюючи з цією технологією.

Різноманіття стандартів Bluetooth, таких як Bluetooth Low Energy (BLE) від 2006 року та Bluetooth 5 від 2016 року, розширюють його можливості і функціональність. Проте, наріжною частиною розуміння Bluetooth є його різні протоколи доступу до мультимедіа, розроблені та легалізовані через Bluetooth SIG. Деякі з цих протоколів можуть бути несумісні з іншими протоколами Bluetooth MAC, де MAC вказує на Medium Access Control, що керує доступом до середовища для передачі даних.

Bluetooth використовує різні методи модуляції на фізичному рівні, такі як GFSK-модуляція та FHSS (Frequency-Hopping Spread Spectrum). Останні протоколи Bluetooth включають функції для запобігання завад, забезпечуючи ефективний і безперебійний канал зв'язку.

Bluetooth є популярною технологією і знаходить застосування в різних пристроях, від безпроводових мишей і гарнітур до фітнес-моніторів та інших пристроїв Інтернету речей. Останні версії протоколу забезпечують більший радіус зв'язку і ефективне використання енергії, що робить його привабливим для широкого спектру застосувань, включаючи інтелектуальну рекламу, обмін

ключами безпеки і дистанційне керування.

Профілі Bluetooth, орієнтовані на додатки, пропонують широкий спектр опцій, від обмежених варіантів без встановлення з'єднання до повних протоколів, які забезпечують безпечне та надійне з'єднання для передачі даних. Незалежно від обраного варіанту, Bluetooth залишається важливою та універсальною технологією для бездротового об'єднання різних пристроїв.

Певні пристрої, оснащені Bluetooth, такі як мережеві принтери, які живляться від змінного струму, використовують бездротовий зв'язок для уникнення використання кабелів, а не для обмеження споживання енергії. Тим не менше, для більшості пристроїв, які живляться від батареї, довгий термін автономної роботи є ключовим фактором, а оптимальним є термін служби приблизно 10 років. Це не лише допомагає скоротити сервісні витрати на технічне обслуговування, пов'язане з заміною джерел живлення, але також робить використання таких пристроїв максимально зручним.

Bluetooth Low Energy, також відомий як Bluetooth Smart у сфері Інтернету речей (IoT), переважно використовує протокол BLE, спеціально розроблений для пристроїв із зниженим споживанням енергії. Якщо потрібна велика швидкість передачі, то можна використовувати Bluetooth 5, який забезпечує швидкішу передачу та тривалі сеанси на основі мережевого протоколу без встановлення з'єднання. У такому випадку дані, що передаються, включають повну адресну інформацію відправника та одержувача в кожному пакеті. Проміжні мережеві пристрої читають цю інформацію та вирішують, як маршрутизувати дані. Зменшення витрат енергії на радіозв'язок та оптимізація програмного забезпечення спрямовані на досягнення практичної можливості десятирічного терміну служби без необхідності заміни джерела живлення, що є критичним для Інтернету речей.

Завдяки розмаїттю доступних протоколів та ефективному використанню енергії акумулятора, технологію Bluetooth можна успішно впроваджувати та розглядати як один із ключових стандартів для бездротового зв'язку в пристроях Інтернету речей.

Технологія EnOcean представляє бездротовий зв'язок Інтернету речей у субгігагерцовому діапазоні, і вражає своєю можливістю функціонувати без використання батарей. Живлення пристроїв Інтернету речей забезпечується за рахунок збору вільної енергії, використовуючи технологію "energy harvesting". Ця технологія використовує незначні зміни в русі, тиску, світлі, температурі або вібраціях для перетворення їх у придатну електричну енергію. Пристрої передають інформацію зі швидкістю 120Кбіт/с на відстань до 100 метрів у вигляді пакетів даних з 14 біт, аналогічно тому, як це відбувається в бездротових брелоках для автомобільних замків та системах дистанційного керування гаражними воротами. Частота передачі для цих пристроїв - 868МГц, що входить в неліцензійний частотний діапазон.

Власником патентів та розробником цієї технології є компанія EnOcean GmbH, яка є дочірньою компанією Siemens. EnOcean GmbH виробляє передавачі, приймачі, трансивери та перетворювачі енергії для різних компаній, таких як Thermokon, Wago, Omnio, Osram, Wieland Electric, Eltako, Distech Controls, Zumtobel, Peha, Herga, MK Electric та інші, які розробляють кінцеві продукти.

EnOcean Alliance, консорціум компаній з Європи та Північної Америки, був створений у 2008 році та спочатку включав компанії, такі як Enocean, Texas Instruments, Omnio, Sylvania, Masco та MK Electric. Цей консорціум визначив основу для розвитку бездротових мереж для автоматизації будівель.

Однією з переваг технології EnOcean є те, що вона не вимагає використання батарейок у вимикачах та датчиках. Безобслуговувані бездротові вимикачі та датчики значно знижують вартість володіння системою та підвищують її надійність.

Міжнародна електротехнічна комісія (МЕК) затвердила стандарт Enocean - ISO/IEC 14543-3-10 — для безпроводових додатків із використанням ультранизького енергоспоживання. Це єдиночасно перший бездротовий стандарт, оптимізований для рішень, які збирають енергію з навколишнього середовища. Разом із профілями обладнання EnOcean (EEPs), розробленими альянсом EnOcean Alliance, цей стандарт створює основу для повністю сумісних та відкритих

бездротових технологій, таких як Bluetooth і WiFi.

Стандарт Enocean спрямований на розробку бездротових сенсорів, датчиків та сенсорних мереж із використанням ультранизького енергоспоживання. Він також охоплює сенсорні мережі, які використовують технології добування енергії з навколишнього середовища, такі як рух, світло чи різниця температур. Цей принцип дозволяє сенсорам та їх електронним системам управління працювати незалежно від зовнішніх джерел живлення.

Міжнародна стандартизація сприятиме прискоренню розробки та впровадження енергетично оптимізованих бездротових сенсорів і сенсорних мереж. Міжнародне визнання технологій відкриває нові ринки і сфери застосування для рішень, які отримують енергію з навколишнього середовища. Технологія EnOcean доповнює вже наявні системи автоматизації для дому та промисловості і призведе до розвитку таких областей, як Розумний Дім і Інтелектуальна Будівля, а також до впровадження рішень для промисловості, логістики і транспорту.

Більше 850 сумісних продуктів, що відповідають стандарту Enocean, вже розроблені членами альянсу EnOcean Alliance. Розробники та виробники можуть скористатися обширним практичним досвідом альянсу, різноманітністю продуктів, інсталяційним досвідом та багаторічним використанням користувачами. Альянс EnOcean Alliance розробляє профілі для додатків (EPPs), які гарантують сумісність продуктів різних виробників, оптимізовані для ультранизького споживання енергії і ідеально доповнюють нові стандарти бездротового зв'язку. Це означає, що розумні та енергоефективні рішення в області автоматизації можуть бути впроваджені незалежно від виробника у будь-якій галузі промисловості. Бездротові технології EnOcean вже довели свою ефективність як рішення для екологічних, інтелектуальних будівель та додатків. Альянс EnOcean Alliance вважає ратифікацію міжнародного стандарту ISO/IEC 14543-3-10 ключовою передумовою для розширення вже успішної і швидко розвиваючоїся екосистеми EnOcean.

Технологія ZigBee є інноваційним рішенням, спрямованим на

високоефективну передачу даних при невеликих швидкостях з гарантованою безпекою. Її унікальність полягає в можливості тривалої роботи мережевих пристроїв за рахунок автономних джерел живлення, таких як батареї. Вже в 1998 році, коли протоколи Wi-Fi і Bluetooth не відповідали потребам ряду додатків, мережі, засновані на ZigBee, почали залучати увагу розробників.

Зокрема, технологія ZigBee вирізняється використанням радіочастот, що не вимагають ліцензування, включаючи смугу в районі 2,4 ГГц. Різні регіони і країни використовують різні смуги робочих частот для ZigBee: 915 МГц в США, 784 МГц в Китаї і 868 МГц в Європі. Початково протокол ZigBee підтримує мережеві з'єднання типу "дерево", "зірка" і мережі, що самоорганізовуються з комірчастою топологією.

Здатність пристроїв до самоорганізації без спільної точки доступу утворює тимчасові мережі, де вузли можуть безпосередньо зв'язуватися точка-до-точки. Такий підхід дозволяє передавати дані через канали зв'язку в мережу, що робить технологію ZigBee вельми привабливою для організації мереж з низькою швидкістю передачі даних на великих площах.

Недоліки технології ZigBee виражаються скороченням часу автономної роботи пристроїв, які виступають у ролі репітерів кластерів таких мереж, які використовуються для обміну даними з віддаленими IoT-пристроями. Швидке виснаження енергії батарей пов'язано з необхідністю передачі не лише власних даних та підтверджень між вузлами мережі, але й інформації та підтверджень від інших пристроїв. Щодо завадостійкості, розширена специфікація ZigBee Pro з 2007 року, яка використовує технологію з перескоком частоти, надає можливість подолати завади, але за умови, що при наявності перешкод вся мережа одразу переходить на інший канал.

Щодо швидкості передачі даних, вона може коливатися від 10 до 250 Кбіт/с, залежно від області застосування пристрою. Низькі швидкості можуть бути достатніми для багатьох IoT-пристроїв, але важливо враховувати, що, використовуючи ZigBee, ви отримуєте меншу пропускну здатність каналу, порівняно з протоколами Wi-Fi. Зменшені швидкості, як правило, забезпечують

більш економне використання енергії батареї, оскільки менше енергії споживається процесорами, логічними мікросхемами та, звісно, під час передачі даних.

Хоча низькі швидкості передачі можуть задовольнити багато потреб IoT-пристроїв, важливо враховувати, що використання ZigBee може забезпечити пристрою довший термін служби батареї, особливо при рідкісних оновленнях даних. Зараз технологія ZigBee широко використовується в різних областях, включаючи домашню автоматизацію та промислові мережі, для забезпечення підключення з низьким споживанням енергії. Наприклад, "безключові" замки на вхідних дверях та системи регулювання температур можуть бути ідеальними прикладами пристроїв ZigBee. Профілі додатків, які визначають стандарти взаємодії пристроїв ZigBee, розробляються альянсом ZigBee Alliance, що дозволяє легко і ефективно впроваджувати їх в різноманітні сценарії від різних виробників.

Комунікація в мережі ZigBee реалізується шляхом обміну пакетами даних між пристроями, які можуть бути класифіковані як координатор (ZC), маршрутизатор (ZR) та кінцевий пристрій (ZED).

Координатор відіграє ключову роль в ініціалізації та управлінні мережею. Цей пристрій встановлює і зберігає ключі безпеки для інших пристроїв, встановлює політику безпеки мережі та встановлює зв'язок з іншими мережами. Важливо відзначити, що в кожній мережі ZigBee може існувати лише один координатор.

Щодо недоліків, можна зазначити обмежену швидкість передачі даних, яка становить до 250 кбіт/с. Хоча для завдань домашньої автоматизації це може здатися невеликим недоліком, важливо враховувати, що ця компромісна швидкість обумовлена прагненням до низького енергоспоживання. Такий підхід не стає критичним для багатьох домашніх завдань і не порушує ефективність системи домашньої автоматизації.

Технологія Z-Wave є бездротовим протоколом зв'язку, спеціально розробленим для домашньої автоматизації, включаючи контроль над середовищем та управління житловими будинками та комерційними об'єктами.

Ця технологія дозволяє безпечно обмінюватися короткими фрагментами даних на радіочастотах діапазону ISM до 1 ГГц, а також розширювати діапазон передачі при низькому споживанні енергії. У Z-Wave використовується FSK- або GFSK-модуляція, і хоча ця технологія спочатку була запатентована, в даний час вона є загальнодоступною відкритою специфікацією ITU G.9959.

Розвиток та підтримка технології Z-Wave знаходиться під відповідальністю компанії Sigma Design, відомого виробника напівпровідникових пристроїв і мікросхем. Z-Wave представляє собою набір пропрієтарних протоколів фізичного та логічного рівнів. Фізичні протоколи визначають параметри мережі, такі як частота роботи, рівень сигналу та модуляція. Логічні протоколи, у свою чергу, визначають адресацію пристроїв, команди, послідовність обміну інформацією та інші аспекти, і реалізовані в мікросхемах Sigma Design.

Термін "пропрієтарний" в даному контексті вказує на те, що документація, що детально описує протоколи, залишається конфіденційною, і кожен виробник, який бажає розробляти пристрої, сумісні з технологією Z-Wave, повинен укласти угоду про нерозголошення деталей протоколу. Цей підхід, хоча накладає певні обмеження, гарантує абсолютну сумісність всіх пристроїв з підтримкою Z-Wave. Таким чином, будь-який датчик чи центральний контролер, орієнтований на роботу в мережі Z-Wave, майже завжди буде сумісний з будь-яким іншим пристроєм цього стандарту. Єдиним винятком може бути програмне обмеження підтримки продуктів тільки одного бренду, але такий підхід не є типовим для екосистеми Z-Wave.

Мережа Z-Wave на фізичному рівні побудована на основі mesh-мережі, де кожен компонент завжди виступає в ролі ретранслятора сигналу. У звичайних мережах, таких як Wi-Fi, пристрої пов'язані безпосередньо з центральним контролером, створюючи типову "зіркову" топологію. Якщо сигнал занадто слабкий для прямого зв'язку, пристрій може залишитися непідключеним.

У мережах Mesh кожен компонент служить ретранслятором сигналу, що дозволяє пристроям підключатися через інші компоненти, навіть якщо прямий зв'язок неможливий через слабкий сигнал. Навіть якщо датчик або актуатор не

може прямо підключитися до центрального контролера через слабкий сигнал, інші компоненти мережі можуть допомогти йому бути підключеним, щоб центральний контролер зміг ним керувати. Ця особливість дозволяє мережі Z-Wave покривати значні відстані та об'єкти, такі як багатоповерхові будівлі.

Мережа Z-Wave має декілька інших переваг, включаючи легке підключення до мережі, яке вимагає лише натискання кнопок на контролері та пристрої; високий рівень безпеки, оскільки всі повідомлення шифруються 128-бітним ключем; і діапазон, який офіційно дозволений для використання малопотужнішими пристроями, забезпечуючи ефективність та безпеку мережі Z-Wave.

При розробці інтелектуального житла для окремих квартир, використання "закритих" частот може стати вирішальним питанням. Однак, враховуючи те, що користувач може планувати розширення системи у майбутньому, виникає питання про доступність пристроїв із необхідною частотою. Також можуть виникнути труднощі, пов'язані з імпортними обмеженнями або обмеженнями на продаж пристроїв з визначеними частотами для домашнього використання.

Варто зазначити, що в майбутньому це може стати фактором, який обмежить вибір доступних компонентів. Однак, слід пам'ятати, що при використанні технології Z-Wave, де рівень сигналу залишається високою рекомендацією, користувачеві не загрожує штраф за порушення вимог використання радіочастотного ресурсу, оскільки сигнал практично не виявляється за межами приміщення.

Проте, при створенні більших об'єктів, як от комерційні інсталяції, готелі чи багатоквартирні будинки, рекомендується враховувати частоту роботи встановлюваних пристроїв. Вибір компонентів, орієнтованих на місцевий ринок та відповідний стандартам, забезпечить сумісність і виключить можливі проблеми з органами державного нагляду за використанням радіочастотного ресурсу.

Технологія Jeweller – це інноваційний безпроводовий засіб забезпечення безпеки, розроблений компанією AJAX. Однією з основних переваг системи є її надійність в умовах обмеженого та нестійкого інтернет-з'єднання. Забезпечуючи

безпеку об'єкту, AJAX враховує можливість низької якості зв'язку та забезпечує віддалений контроль і управління системою.

Інноваційний IoT-протокол AJAX дозволяє охоронній системі працювати ефективно при низьких швидкостях Інтернет-з'єднання, таких як GPRS зі швидкістю 0,5 Кбіт/сек. Це робить систему високоефективною та доступною в різних умовах підключення.

AJAX Hub використовує дві антени для аналізу рівня сигналів у реальному часі, гарантуючи вибір найкращого сигналу. У випадку саботажу або виявлення злому датчика Hub негайно активує тривогу, забезпечуючи надійність безпекового заходу.

Система регулярно опитує та перевіряє зв'язок з датчиками кожні 12 секунд, навіть якщо зв'язок втрачено. Алгоритм DeliverAnyway гарантує, що в разі втрати зв'язку тривога буде негайно спрацьована. Захист від несанкціонованого доступу забезпечується шифруванням всіх передаваних даних алгоритмом на основі AES.

Система дозволяє підключити до 100 пристроїв до централі (Hub), що дозволяє розміщувати датчики в кожній кімнаті великого будинку або на кожній двері та вікні. Датчики Ajax працюють на відстані до 2 000 метрів відкритого простору від хаба, забезпечуючи надійний зв'язок.

Оптимізована технологія Jeweller забезпечує стабільну роботу системи при мінімальній вихідній потужності, економляючи заряд батареї датчиків та забезпечуючи їх тривалий термін служби до 7 років.

Ajax Hub завжди знає, на зв'язку чи ні, завдяки двосторонньому зв'язку, адресності та перевірці пристроїв з інтервалами від 12 секунд. Це гарантує цілісність системи, і в разі виявлення неполадок з датчиками, система миттєво повідомляє про проблеми та їх причини.

Запобігаючи глушінню датчиків від накладення радіохвиль, Ajax Hub використовує дві антени для аналізу рівня сигналів в реальному часі та вибору найкращого сигналу, навіть в екстремальних радіоумовах.

Забезпечення надійності безпроводової системи безпеки вимагає вдосконалених технічних рішень, і в цьому Ajax виявляється передовником.

Відзначаючи недостатню надійність використання лише однієї частоти, Ajax імплементує технічний арсенал, базуючись на використанні кількох частот. У випадку глушіння, система автоматично перемикається на чисту частоту, забезпечуючи неперервний зв'язок. Коли вся смуга заглушена, центральний контролер (Hub) спрацює тривогу, миттєво реагуючи на потенційну небезпеку.

Основні технічні характеристики системи Ajax вражають своєю високотехнологічністю:

Основні технічні характеристики системи Ajax:

— потужність радіосигналу 25 мВт;

— двосторонній зв’язок;

— робочі частоти: від 868.0 до 868.6 МГц;

— блочне шифрування засноване на алгоритмі AES;

— дальність зв'язку з датчиками може сягати 2 км на відкритій місцевості;

— опитування датчиків відбувається кожні 12 секунд;

— максимум може бути 100 підключених пристроїв;

— термін роботи датчиків 7 років;

— миттєва доставка сигналу;

— TDMA, захист від глушіння, захист від підробки, захист від збоїв, віддалене налаштування.

Ajax виправдовує свій статус самодостатньої системи, оскільки Hub автоматично слідкує за новими версіями програмного забезпечення, автоматично завантажуючи та встановлюючи оновлення. Інсталяторам не потрібно виїжджати на об'єкт, оскільки конфігурація можлива з будь-якої точки нашої планети.

2 ВИБІР ОПТИМАЛЬНИХ ВАРІАТІВ ТЕХНОЛОГІЙ ТА ЗАСОБІВ ПОБУДОВИ МІКРОКОМП'ТЕРНОЇ СИСТЕМИ

2.1 Технологія LoRaWAN

LoRa (Long Range) представляє собою інноваційний метод модуляції та відповідну мережеву технологію, що активно розвивається за підтримки LoRa Alliance, відкритої некомерційної організації, де об'єдналися провідні учасники галузі Інтернету речей, такі як IBM, Semtech, Cisco, Inmarsat, Swisscom та інші. Що відрізняє технологію LoRa від інших бездротових протоколів малого радіусу дії, це її особливий характер.

Часто під терміном LoRa розуміється саме тип модуляції, в той час як LoRaWAN вказує на відкритий мережевий протокол LoRa. Важливо розрізняти ці терміни, інакше може виникнути плутанина з LPWAN. LoRaWAN використовується для передачі обмежених за обсягом пакетів даних на великі відстані. Ця мережа спеціально створена для потреб розподілених мереж телеметрії, міжмашинної взаємодії та Інтернету речей. Логіка мережі LoRa розкривається в її здатності збирати дані з різноманітного обладнання, такого як датчики, лічильники і сенсори, вирізняючи її серед перспективних бездротових технологій для цілей збору інформації.

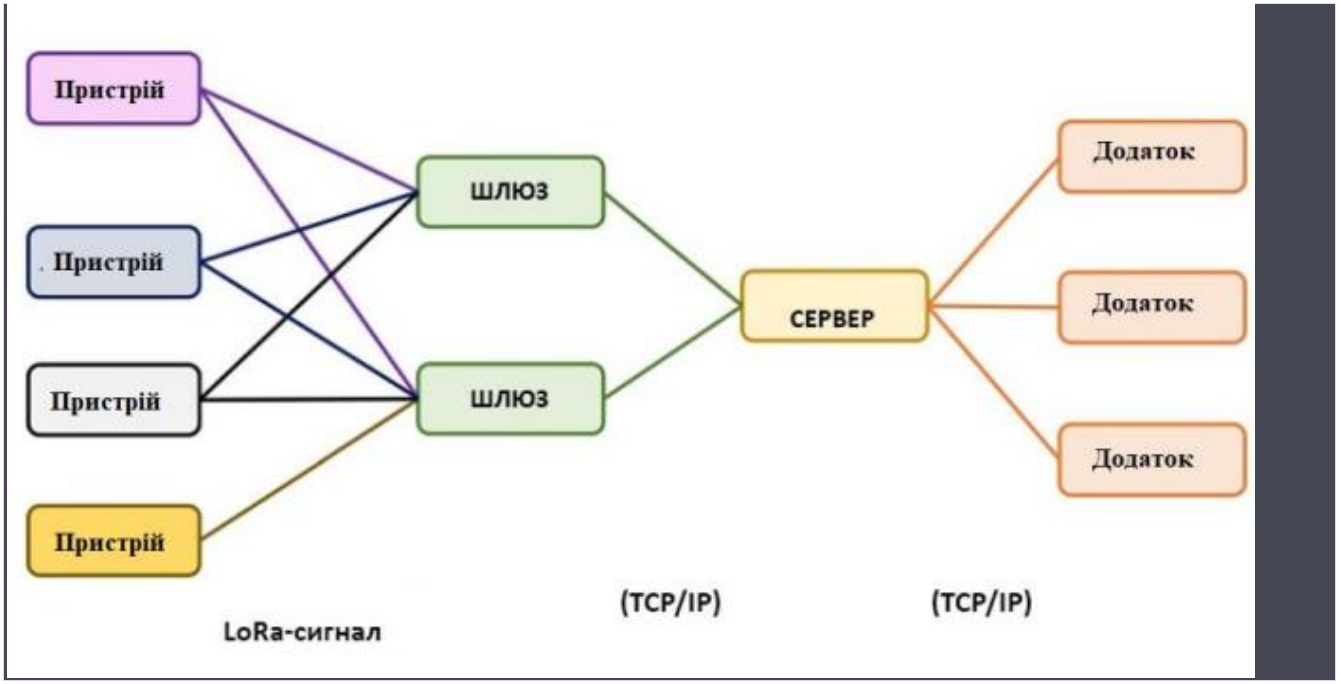

Рисунок 2.1 — Принцип роботи мережі LoRa

В залежності від розподілу по регіонах, мережа, яка використовує технологію LoRa, опирається на радіочастоти субгігагерцового діапазону, які не вимагають ліцензування в спектрах VHF (від 30 до 300 МГц), UHF (від 300 МГц до 3 ГГц) або від 800 до 930 МГц. Оскільки LoRa використовує низькі радіочастоти порівняно із стандартами, що використовують 2,4 або 5 ГГц, вона вирізняється заради своїх радіочастотних характеристик, що дозволяють їй проникати глибоко в будівлі та туди, куди вищі частоти не досягають.

Модуляція LoRa відрізняється від інших типів модуляції, зокрема FSK, OFDM, FHSS або DSSS, завдяки використанню лінійної частотної модуляції - Chirp Spread Spectrum (CSS). Це досягнення в області радіочастотних технологій полягає в перебудові несучої частоти за лінійним законом. Це надає сигналам LoRa високий рівень стійкості до перешкод. Завдяки цій модуляції низькі бітові швидкості (до 300 біт/с) можуть ефективно уникати впливу вузькосмугових перешкод, таких як FSK-сигнали, забезпечуючи успішне відновлення на стороні приймача.

Технологія LoRa дозволяє використовувати різні комбінації швидкості передачі даних і модуляції в залежності від конкретних вимог. Це важливо для досягнення оптимального балансу між швидкістю передачі та дальністю зв'язку в різних умовах. Враховуючи ці аспекти, технологія LoRa виявляється високоефективною та перспективною для різних сценаріїв використання в Інтернеті речей.

Мережа LoRa може бути впроваджена як автономна мережева архітектура або як частина зв'язаної мережі в тих регіонах світу, де існують оператори мереж загального користування. Ці оператори надають можливість пристроям LoRa з'єднуватися через спеціальні шлюзи для передачі даних у хмарне сховище. Хоча мережа на базі технології LoRa вперше була розгорнута в Європі, успішна експансія її охоплення спостерігається і в інших частинах світу.

Крім компанії Semtech, яка виготовляє мікросхеми LoRa, ST Micro і Microchip також пропонують системи на кристалі для розробників. Це відкриває

нові можливості для гнучкості та інновацій при створенні проектів на базі технології LoRa. Такий різноманітний виробничий ландшафт сприяє швидшому поширенню і удосконаленню мережі, роблячи її більш доступною та ефективною для розробників у всьому світі.

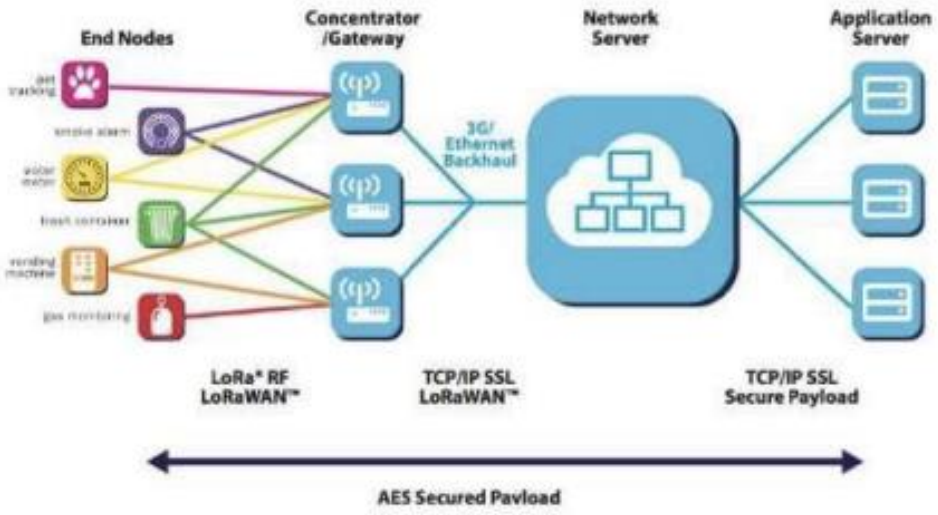

Рисунок 2.2 —Архітектура мережі LoRaWAN

Проте важливо пам'ятати, що застосовуючи технологію LoRa, навіть на використання спектра частот, який не потребує ліцензування, все одно потрібно отримати сертифікацію для пристроїв і підтвердження того, що конкретний пристрій відповідає специфікаціям LoRa. Для отримання сертифікації зазвичай необхідні випробування щодо потужності передавача, девіації частоти, займаної смуги пропускання, гармонік та спектральної щільності потужності. Цей процес сертифікації та передвипробування вже проводяться в авторизованих випробувальних лабораторіях.

Навіть якщо технологія LoRa є відносно новим стандартом для розробників, їм вже доступні мікросхеми, готові модулі та різноманітні тестові інструменти. Це сприяє полегшенню процесу розробки та впровадження пристроїв, забезпечуючи високий стандарт якості і сумісності в мережі LoRa.

2.2 Платформа Arduino Mega 2560

Плата Arduino Mega 2560 представляє собою інтелектуальний пристрій, заснований на мікроконтролері ATmega2560. Це інноваційне рішення включає у себе повний набір можливостей для зручної та ефективної роботи з мікроконтролером. Зокрема, на борту пристрою розташовано 54 цифрових входи/виходи, з яких 15 можуть працювати як виходи з модуляцією ширини імпульсів (ШІМ), а також 16 аналогових входів для точного зчитування аналогових сигналів.

Однією з важливих особливостей є наявність 4 апаратних приймачів UART, що дозволяють реалізувати послідовні інтерфейси. Крім того, в пристрої є кварцовий резонатор, роз'єм USB для комунікації з комп'ютером чи іншими пристроями, а також інші зручності, такі як роз'єм живлення, роз'єм для внутрішньосхемного програмування (ICSP) та кнопка скидання.

Щоб почати працювати з Arduino Mega 2560 потрібно лише надати живлення від AC/DC-адаптера чи батарейки, або підключити до комп'ютера через USB-кабель. Важливо відзначити, що ця плата сумісна з більшістю плат розширення, розроблених для Arduino Duemilanove та Diecimila.

Особливість Arduino Mega 2560 в порівнянні з попередніми моделями полягає в використанні мікроконтролера ATmega16U2 для конвертації інтерфейсів USB-UART, відмінної альтернативи мікросхемі FTDI. У версії R2 плати доданий резистор, який підтягує до землі лінію HWB мікроконтролера 8U2, спрощуючи процес оновлення прошивки та перехід в режим DFU. Це додає платі ще більше гнучкості та зручності в роботі.

Arduino Mega має вбудовану систему живлення, яка дозволяє автоматично вибирати джерело енергії або USB, або зовнішнє джерело. У разі зовнішнього живлення можна використовувати AC/DC-адаптер або акумулятор чи батарея. Для підключення мережного адаптера у якого діаметр штекера - 2.1 мм, а центральний контакт є позитивним, його слід вставити в відповідний роз'єм живлення на платі. Якщо обрано акумулятор/батарею, їхні дроти слід підключити до виводів Gnd і Vin роз'єму POWER.

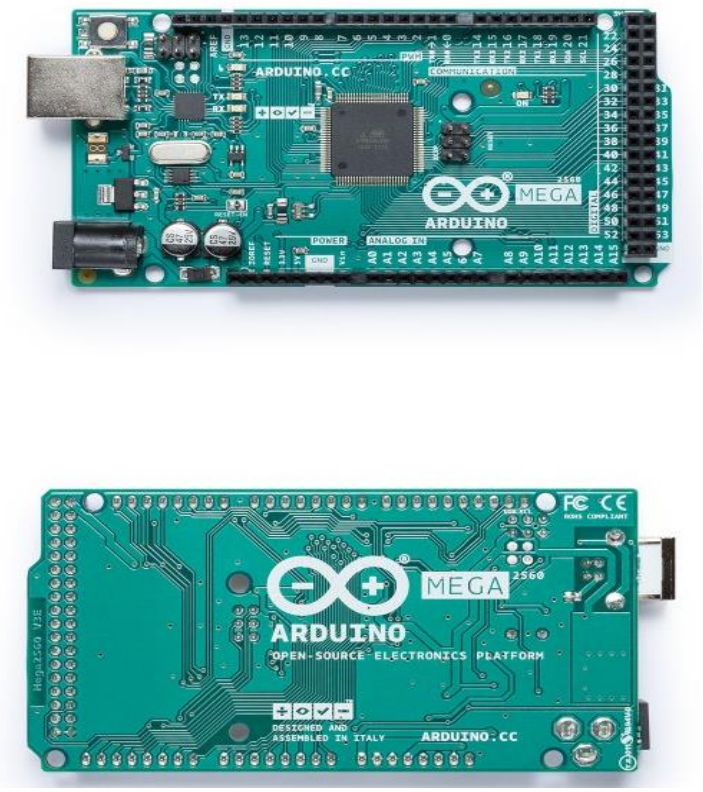

Рисунок 2.3 — Arduino Mega

Важливо враховувати, що напруга зовнішнього джерела живлення повинна знаходитися в межах від 6 до 20 В. Однак зниження напруги нижче показника у 7 В може призвести до зменшення напруги на виведенні 5V, що може вплинути на стабільність роботи пристрою. З врахуванням цього, рекомендується використовувати джерело живлення із напругою від 7 до 12 В, оскільки використання напруги більше 12 В може спричинити перегрів стабілізатора напруги та вивести плату із ладу. Такий підхід забезпечить оптимальну та стабільну роботу Arduino Mega.

Таблиця 2.1 — Технічні характеристики Arduino Mega

Технічні характеристики	Значення
Мікроконтролер	ATmega2560
робоча напруга	5В
Рекомендована напруга живлення	Від 7 до 12В
Напруга живлення (гранична)	Від 6 до 20В

Цифрові входи/виходи	54 (з яких 15 можуть використовуватися як ШИМ-виходи)
Аналогові входи	16 шт.
Максимальний струм одного висновку	40 мА
Максимальний вихідний струм виведення 3.3V	50 мА
Розміри	101.52 мм x 53.3 мм.
Flash-пам'ять	256 КБ, з яких 8 КБ використовуються завантажувачем
SRAM	8 КБ
EEPROM	4 КБ
Тактова частота	16 МГц

Виводи живлення, розташовані на платі: VIN, 5V, 3V3, GND, IOREF.

Вивід VIN (Voltage In) призначений для подачі напруги безпосередньо на Arduino від зовнішнього джерела живлення. Він не пов'язаний із 5V від USB чи іншої стабілізованої напруги. VIN використовується як для живлення пристрою, так і для відведення струму, коли використовується зовнішній адаптер.

Вивід 5V отримує напругу 5В від внутрішнього стабілізатора напруги на платі Arduino. Напруга на цьому виводі не залежить від того, як живиться пристрій - від адаптера (від 7 до 12В), від USB (5В) або через вивід VIN (від 7 до 12В). Не рекомендується подавати живлення на пристрій через виводи 5V або 3V3, оскільки у цьому випадку не використовується внутрішній стабілізатор напруги, що може призвести до поломки.

Вивід 3V3 надає напругу 3.3В від внутрішнього стабілізатора напруги на платі. Максимальний споживаний струм з цього виводу - 50 мА.

Вивід GND (Ground) для з'єднання землі.

Вивід IOREF (Input/Output Reference) надає платам розширення інформацію про напругу мікроконтролера Arduino. Залежно від напруги, зчитаної з виводу IOREF, плата розширення може автоматично переключатися на відповідне

джерело живлення або використовувати рівні перетворювачі, що дозволяє працювати з 5В або 3.3В пристроями.

Arduino Mega 2560 — це мікроконтролерна плата, яка відкриває широкі можливості для встановлення зв'язку з комп'ютером, іншими Arduino або мікроконтролерами. За допомогою ATmega2560 можна реалізувати до чотирьох апаратних приймачів UART для послідовних інтерфейсів з логічним рівнем TTL 5В. Важливою особливістю є наявність мікроконтролера ATmega16U2 (або ATmega8U2 на платах версії R1 і R2), який забезпечує зв'язок одного з приймачів з USB-портом комп'ютера.

При підключенні до комп'ютера Arduino розпізнається як віртуальний COM-порт, що дозволяє взаємодіяти з нею через спеціальну програму SerialMonitor, що входить до пакету програмного забезпечення Arduino. Ця програма дає змогу зчитувати та відправляти прості текстові дані на Arduino.

У разі передачі даних через мікросхему ATmega8U2/ATmega16U2 під час USB-з'єднання, світлодіоди RX та TX на платі вказують на активність передачі даних. Це корисно для відладки та моніторингу даних, і світлодіоди RX та TX світяться під час цього процесу. При використанні послідовних виведень 0 і 1 без USB-перетворювача, ці світлодіоди не використовуються, забезпечуючи гнучкість в залежності від потреб проекту.

Мікроконтролер ATmega2560 володіє апаратною підтримкою послідовних інтерфейсів TWI (Two-Wire Interface) і SPI (Serial Peripheral Interface). У програмному забезпеченні Arduino міститься бібліотека Wire, яка значно полегшує взаємодію з шиною TWI та містить більш детальну інформацію щодо роботи з шиною TWI.

Для взаємодії з інтерфейсом SPI використовується бібліотека SPI. Ця бібліотека дозволяє просто та ефективно взаємодіяти з пристроями, підтримуючи SPI-з'єднання. Вона може бути використана для обміну даними між ATmega2560 та іншими пристроями, такими як сенсори, датчики чи інші мікроконтролери, що підтримують інтерфейс SPI.

Рисунок 2.4 — Мікроконтролер Atmega2560

Використання бібліотек Wire та SPI надає можливість легко інтегрувати функціональність TWI та SPI у свій проект на базі Arduino Mega 2560, роблячи взаємодію з периферійними пристроями більш доступною та зручною.

2.3 Мінікомп’ютер Raspberry Pi Zero

Raspberry Pi Zero - це одноплатний комп'ютер, який розробляється фондом Raspberry Pi Foundation. Він є меншим і менш потужним у порівнянні з більш відомими моделями, такими як Raspberry Pi 3 або Raspberry Pi 4, але в той же час він дуже компактний і має великий потенціал для використання в різних проектах.

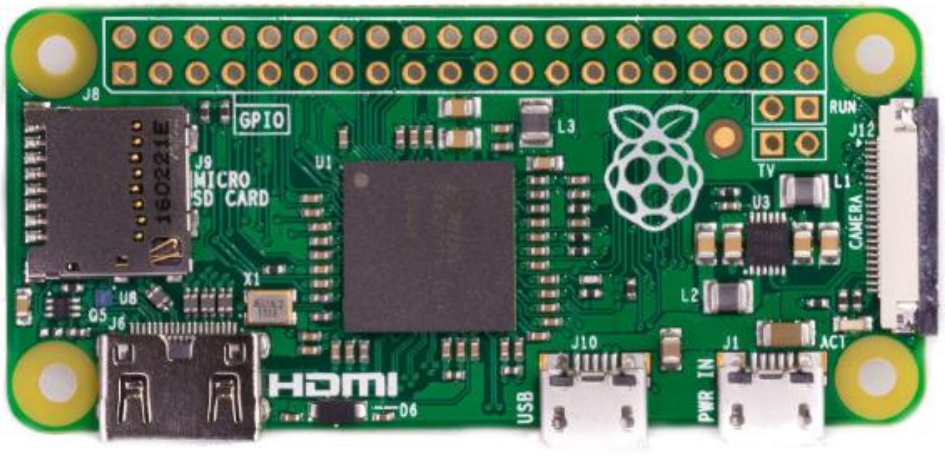

Рисунок 2.5 — Raspberry Pi Zero

Основні характеристики Raspberry Pi Zero:

1) Одноядерний процесор ARM1176JZF-S;

2) 512 МБ оперативної пам'яті (RAM);

3) порти і роз'єми:

— HDMI для підключення до монітору чи телевізора;

— mini HDMI для аудіо і відео виведення;

— мікро USB для живлення та передачі даних;

— мікро USB OTG для підключення до периферійних пристроїв;

— GPIO (загального призначення введення/виведення) для підключення до різних сенсорів, пристроїв і компонентів.

Raspberry Pi Zero може бути використаний з камерами, такими як камера Raspberry Pi (камера моделі v1 або v2), що відкриває безліч можливостей для проектів, пов'язаних із зображенням і відео.

Бездротові можливості Raspberry Pi Zero W (Wireless) включають в себе підтримку Wi-Fi і Bluetooth, що розширює можливості підключення до мережі та бездротових пристроїв.

Плата Raspberry Pi Zero має наступні компоненти:

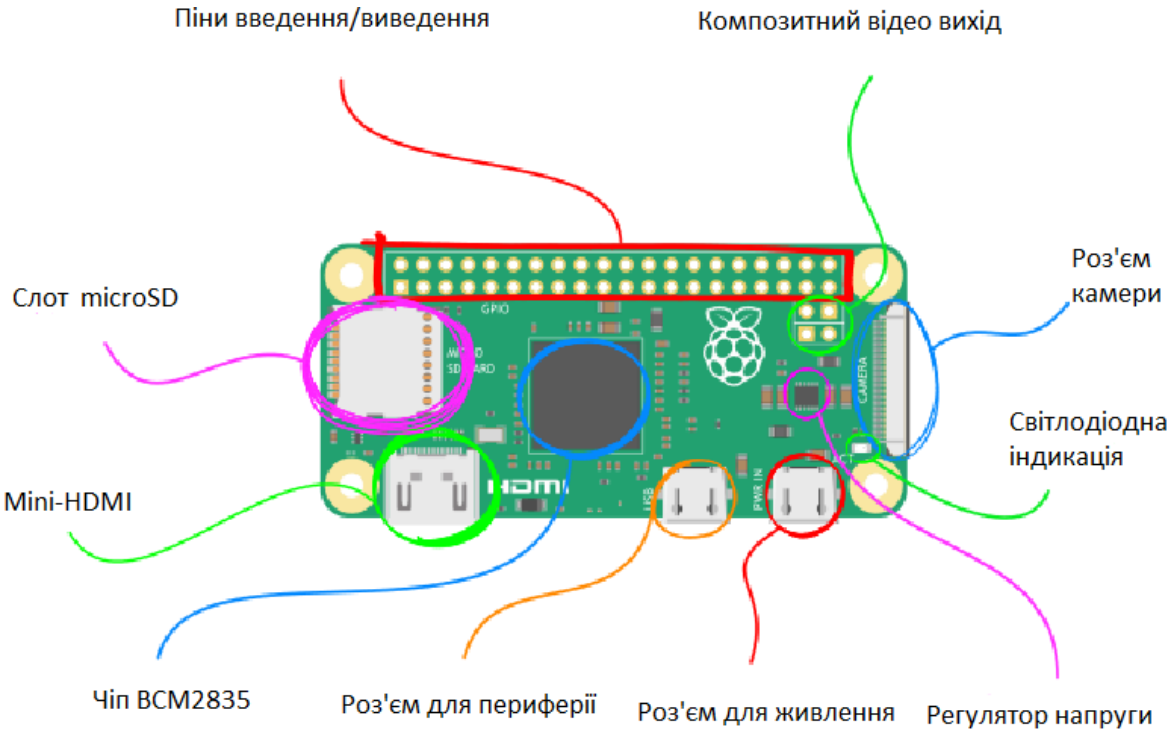

Рисунок 2.6 — Елементи плати

Чіп BCM2835 — основний елемент комп'ютера Raspberry Pi Zero, що базується на технології SoC (System-on-a-Chip — система на кристалі). На кристалі міститься процесор, частота роботи якого становить 1 ГГц і графічний двоядерний співпроцесор частота якого 250 МГц.

Також на даному чіпі розташовується оперативна пам'ять Elpida, об'єм якої 512 МБ, заснована на технології PoP (Package-on-Package — корпус на корпусі)

Mini-HDMI порт — цифровий роз'єм, призначенням якого є виведення зображення та звуку на пристрої.

Роз'єм підключення периферії — порт, формфактором якого є micro-USB, що слугує для підключення мультимедійних пристроїв зі стандартним роз'ємом USB.

Роз'єм живлення — роз'єм, формфактором якого є micro-USB, та використовується для живлення Raspberry Pi.

Слот для microSD — слот для карт пам’яті, які використовуються для встановлення Raspberry Pi OS.

Композитний відео вихід — вихід для аналогових сигналів, який має вигляд двох пінів. Даний відео вихід призначений для підключення до ламповых телевізорів через роз'єм RCA.

Роз'єм камери (CSI) — роз'єм, що використовується для підключення камери Raspberry Pi.

Регулятор напруги — двоканальний імпульсний понижувальний регулятор напруги з виходами 3,3 В і 1,8 В. Максимальний струм кожного каналу 1 А.

На Raspberry Pi Zero розташовані два ряди по 20 контактів у вигляді луджених отворів.

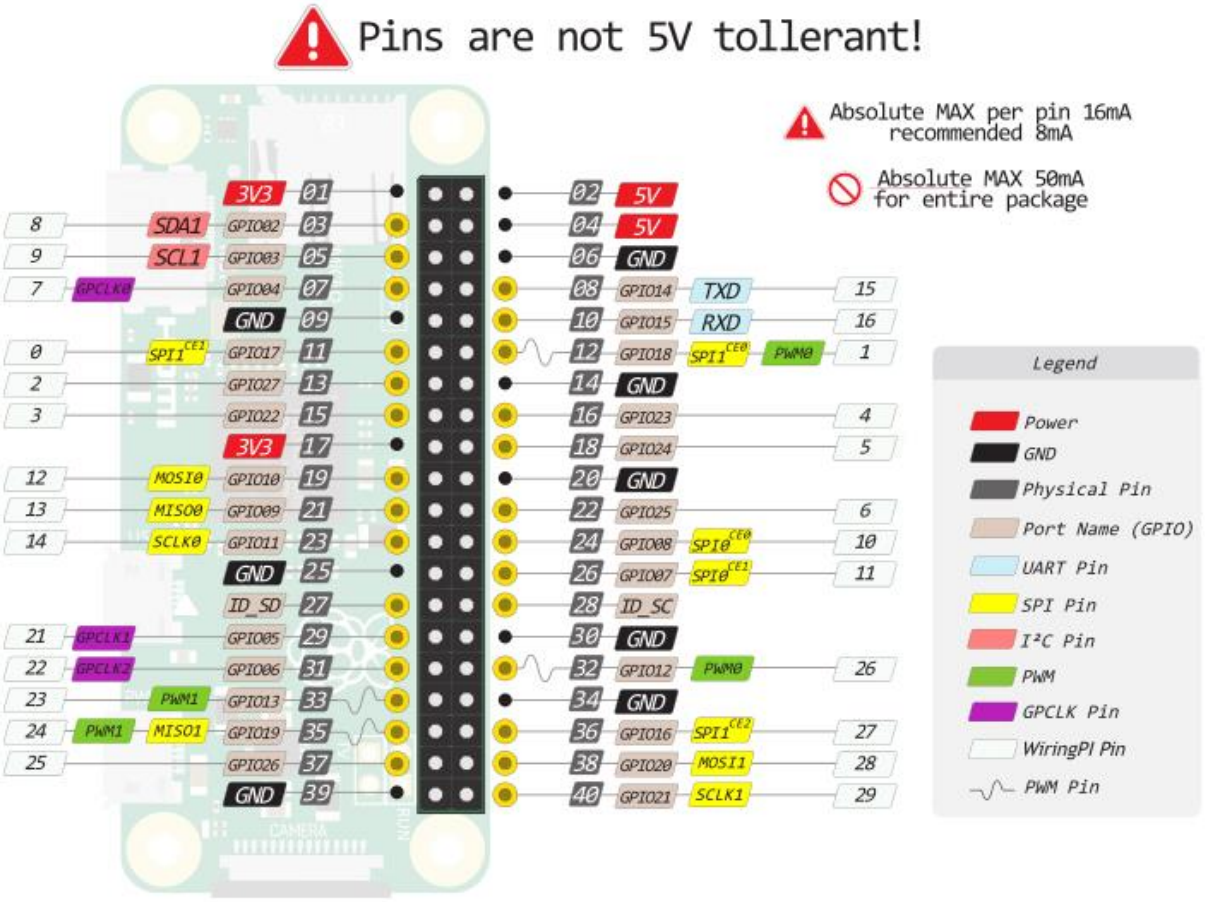

Рисунок 2.7 — Піни плати Raspberry Pi Zero

Піни живлення:

1) напруга 5V надходить на виводи у випадку підключення плати через USB;

2) пін 3V3, від стабілізатора напруги з виходом 3,3 вольта та максимальним струмом 1 А. Регулятор живить всі елементи плати;

3) пін GND виводи землі.

На відміну від платформ з логічною напругою 5, напруга логічних рівнів Raspberry Pi є 3,3 В. Виходи на платі для логічної одиниці видають 3,3 В, а в режимі входу очікують приймати не більше 3,3 В. Висока напруга може пошкодити плату.

На платі Raspberry Pi розміщається 26 контактів пінів введення/виведення GPIO. Логічний рівень одиниці — 3,3 В, нуля — 0 В. Максимальний струм виходу — 16 мА.

2.4 Опис модулів ESP32

Невдовзі після появи мікроконтролер ESP32 став повністю інтегрованим у промислову автоматизацію, головним чином у розгортання вбудованих систем і різноманітних завдань IoT. Його великою перевагою, безсумнівно, є його ціна, структура схеми, можливість підключення периферійних пристроїв і модулів IoT та інших датчиків, а також відмінна підтримка створення додатків.

Чіп ESP32 добре реалізований як веб-сервер, використовуючи бездротовий зв’язок Wi-Fi, Bluetooth і переважно стандарт зв’язку MQTT на рівні обміну повідомленнями з оточенням. Він часто працює з іншим відповідним мікрокомп’ютером, таким як Raspberry Pi.

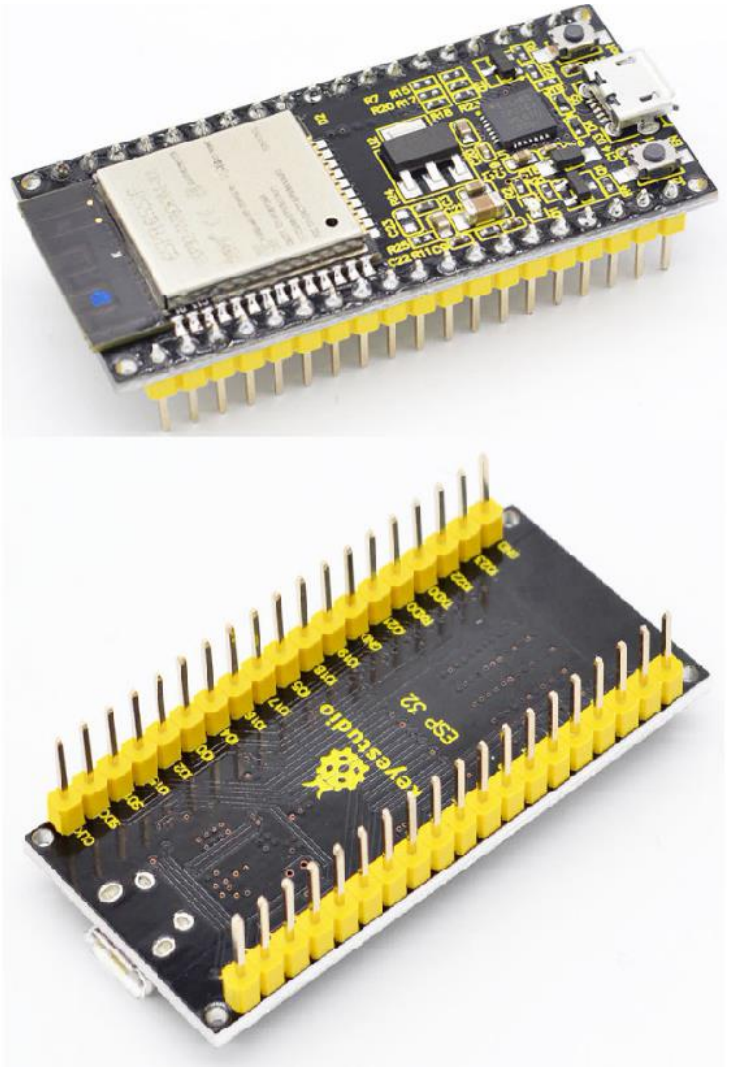

Рисунок 2.8 — мікроконтролер ESP32

Вбудована система на базі мікросхеми ESP32 може виконувати завдання

моніторингу за допомогою Dual-core для її роботи, тоді як одне ядро піклується про отримання та обробку даних з датчики, інше ядро вирішує проблеми зв'язку з оточуючими пристроями. Використання кількох ядер може бути корисним у сфері розгортання ESP32 у сфері машинного навчання та нейронних мереж. В даний час ESP32 реалізовано в багатьох галузях IoT і додатках у сферах.

Мікросхема ESP32 підходить для реалізації в додатках різних завдань моніторингу та безпеки. Загальним елементом впроваджених додатків є високоякісне та недороге рішення в багатьох сферах, таких як сонячна водонасосна система в сільському господарстві, або різні типи систем моніторингу в сільському господарстві. Інші відповідні застосування також можна знайти в області моніторингу систем якості повітря, моніторингу витоків LPG або систем управління відходами. Також ESP32 може бути використано як веб-сервер для системи фотоелектричного моніторингу в реальному часі.

За допомогою інших відповідних периферійних пристроїв, таких як камера, лінійні приводи та драйвери, ми можемо використовувати ESP32 для реалізації системи контролю позиції, або інтелектуальної системи спостереження. Мікроконтролер ESP32 також використовується для виконання завдань Device-Free Passive (DfP), таких як виявлення, локалізація та відстеження людських об’єктів, або використовується за допомогою GPS як внутрішньої системи позиціонування, або системи безпеки.

Системи SCADA також почали зосереджуватися на IoT в останні роки. Архітектури SCADA еволюціонували протягом багатьох років від монолітних (автономних) через розподілені та мережеві архітектури до найновішої архітектури Інтернету речей (IoT). SCADA може використовуватись із локальною серверною платформою IoT, як основних термінальних пристроїв для обробки даних і взаємодії людини з машиною та мікроконтролера ESP32, як віддаленого термінального пристрою для отримання, обробки і надсилання віддалених даних від приладів польових вимірювальних приладів.

Системи IoT несуть ризики з точки зору безпеки та конфіденційності. Без усунення цих ризиків вимоги IoT не виконуються належним чином. Також ESP32

може використовуватись, як система моніторингу навколишнього середовища разом із бездротовою мережею датчиків. Перевагою таких приладів є не тільки його низька вартість, але й система з низьким енергоспоживанням, яка сьогодні є бажаною перевагою при розгортанні системи моніторингу.

Сьогодні невід'ємною частиною Інтернету речей є iHealth і моніторинг здоров'я за допомогою розумних вбудованих систем. Ця сфера підтверджується використанням ESP32 для моніторингу частоти серцевих скорочень, системи розпізнавання зображень для сліпих людей, або інтелектуальної системи моніторингу фізіологічного розчину у внутрішньовенній терапії. ESP32 надає можливість реалізації системи управління за допомогою жестів людини на носимих пристроях. Чіпи ESP32 реалізовані не тільки в системах моніторингу iHealth, але зараз також розробляються розумні системи для комфорту та зручності психічного здоров'я та благополуччя. За допомогою цього модуля можна створювати інтелектуальні системи освітлення відповідно до виявлення людських емоцій. Він використовує, серед іншого, необхідні апаратні компоненти: мікросхеми ESP32, а саме ESP8266, і стандарт зв'язку MQTT, який є найпоширенішим стандартом зв'язку. Можна використовувати зашифрований протокол MQTT із сертифікатами SSL/TLS для захисту зв'язку MQTT. Інші функції безпеки зв'язку, такі як алгоритм вдосконаленого стандарту шифрування (AES), реалізований на ESP32 за допомогою модуля LoRa для захисту бездротового зв'язку.

У сфері вбудованих систем і моніторингу можна використовувати варіанти розробки ESP32 із сенсорними екранами, інтегровані в плату ESP32 Wrover, або підключені зовні.

Мікроконтролерна операційна система FreeRTOS є програмним забезпеченням з відкритим вихідним кодом, яке забезпечує чудову підтримку програм реального часу. Таким чином, ESP32 відіграє важливу роль у розробці систем IoT та вбудованих проектів. Включення додатків IoT на основі ESP32 у системах автоматизації будівель підтверджує придатність використання ESP32 у недорогих додатках із продуктивністю промислового рівня. Це важливо для

розробки дешевих, але не менш надійних промислових рішень на основі SoC.

3 ПРОЕКТУВАННЯ АПАРАТНОЇ ТА ПРОГРАМНОЇ ЧАСТИН МІКРОКОМП'ЮТЕРНОЇ СИСТЕМИ

3.1Головна концепція системи

Основна ідея полягає в тому, щоб використовувати чіп ESP32, який встановлюється з модулем камери, для моніторингу будинку, а також для моніторингу температури окремих приміщень, таких як коридор і котельня. Система містить датчики PIR і датчики температури в приміщеннях, що контролюються, і підключені модулі камер до мікроконтролера ESP32. Дані, зібрані з датчиків, надсилаються по бездротовому зв'язку на блок керування, яким є Raspberry PI Zero. Інші створені апаратні модулі — це панель введення з сенсорним екраном і панель введення з платою ESP32 Wroom і мембранною клавіатурою. Використовується для розблокування та замикання будинку для активації датчика руху та камер.

Блок управління розширюється за допомогою GSM модуля IoT-GA5-B. Він забезпечує надсилання повідомлень на мобільний телефон власника будинку, які інформують його/її про стан безпеки будинку. Інші HW модулі - це сенсорні екрани, що показують поточний стан і температуру в кімнатах, і контрольні світлодіоди, що записують стан всієї системи.

Основні аспекти всієї системи такі:

— виявлення руху за допомогою датчиків PIR;

— захоплення зображення з камери;

— моніторинг фізичних величин домогосподарства, таких як температура, вологість, можливість розширення на інші контрольовані фізичні величини;

— зберігання та моніторинг вимірюваних даних;

— доступ до даних через веб-сервер;

— адаптивні програми для мобільних пристроїв;

— бездротовий зв'язок, зв'язок MQTT, безпека системи;

— GSM зв'язок з системою.

Система включає в себе Raspberry Pi, який виступає у ролі блоку керування. У комплект також увійдуть три модуля камери, оснащені PIR і

датчиком температури. ESP32, оснащений дисплеєм і мембранною клавіатурою для входу в котельню та сенсорним екраном, який також керується ESP32 на головному вході, буде використовуватися для замикання будинку. У комплект також входить один мікропроцесор ESP32, оснащений двома датчиками температури, як і модуль камери.

3.2 Розробка структурної схеми системи

Структурна схема приладу може бути розкладена на кілька ключових блоків, кожен з яких відповідає за конкретну функцію (рис. 3.1).

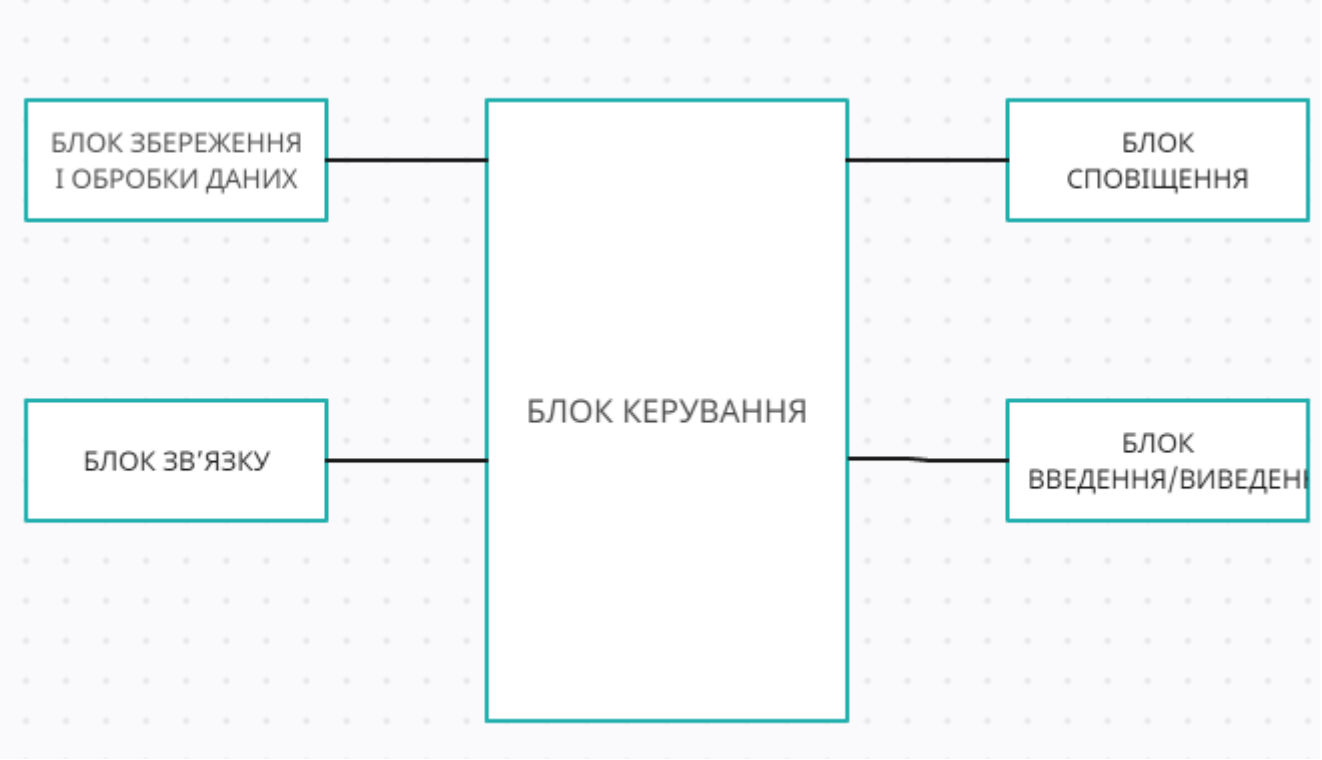

Рисунок 3.1 — Структурна схема системи

Структурна схема пристрою містить такі блоки:

1) блок керування, куди входить основний контролер Raspberry Pi Zero, який відповідає за загальне керування та координацію роботи всіх компонентів та модуль управління дисплеєм;

3) блок збереження і обробки даних, який відповідає за обробку та збереження отриманих від камери зображень або відео;

4) блок зв'язку, призначений для відправки повідомлень чи даних через мобільну мережу, або через Wi-Fi: що забезпечує зв'язок з іншими пристроями чи серверами.

Блок сповіщення містить датчик руху PIR для виявлення руху та сповіщення основного контролера, а також зумер для відтворення аудіосигналів для сигналізації або сповіщення.

Блок введення/виведення куди входить клавіатура для введення інформації або управління і камера, що відповідає за захоплення зображень або відео та їхньої подальшої передачі (рис 3.2).

Рисунок 3.2 — Деякі апаратні компоненти

Система містить такі елементи: ESP-WROOM-32, РК-дисплей, матрична мембранна клавіатура, сигналізація.

ESP-WROOM-32 належить до блоків мікроконтролерів (MCU), які в основному є платформами комп’ютерної плати з центральним процесором, пам’яттю, шинами та вбудованими периферійними пристроями, необхідними для зчитування під’єднаних датчиків або виконавчих механізмів.

РК-дисплей підключається через шину I2C. Це послідовна шина, яка

поділяє підключений пристрій на головний або підпорядкований. Один провід використовується для передачі тактового сигналу (SCL — синхронний годинник) і є каналом даних (SDA - синхронні дані).

Матрична мембранна клавіатура 4x4 для одноплатних комп'ютерів, яка містить символи (1 ÷ 9, A ÷ D і спеціальні символи # і *), і вони підключені до 8 контактів. Кнопки в окремих рядках і стовпцях завжди підключаються до загального проводу. Після натискання кнопки рядок і стовпець завжди з'єднуються в заданій точці. Виробник гарантує термін служби 100 мільйонів натискань.

Сигналізація, яка є останнім компонентом у цьому наборі, робоча напруга якої знаходиться в діапазоні 3-24 В і інтенсивністю 95 дБ. Його функція полягає в тому, щоб викликати гучний сигнал у разі входу зловмисника в будинок, який повинен попередити вас про вторгнення у ваш будинок.

3.3 Вибір електронних компонентів

3.3.1Панель введення з сенсорним екраном

Основним входом є ESP32. Для блоку дисплея буде використовуватися 2,8-дюймовий сенсорний екран USART від NEXTION. Найсуттєвішою перевагою цих дисплеїв є графічний редактор, у якому можна швидко та без проблем визначати графічне та сенсорне середовище та виконувати моделювання. Цей редактор економить багато часу при розробці програм. Роздільна здатність цього дисплея становить 320x240 разом з 65 тисячами кольорів з регульованою яскравістю, що підійде для будь-яких додатків.

Найбільшою перевагою є зв'язок через USART, за допомогою якого створене графічне середовище може бути завантажено безпосередньо на дисплей, у який інтегровано 4 МБ флеш-пам'яті. Другий варіант — це використання карти micro SD, для якої в цьому дисплеї є слот. Інтерфейс USART також використовується для зв’язку з мікропроцесором, через який він отримує змінні та надсилає елементу відповідь на клацання. Для реалізації системи можна використовувати будь-який інший дисплей. Панелі введення показано

на рисунку 3.3.

Рисунок 3.3 — Ввідний щит.

3.3.2 Модулі камери

Композиція містить такі елементи: Ai-Thinker ESP32-CAM, RGB світлодіодний модуль, датчик температури DALLAS DS18B20, PIR-модуль.

Ai-Thinker ESP32-CAM, що являє собою модуль Wi-Fi та BT MCU, розширений 520 КБ SRAM, включаючи зовнішню 4MPSRAM. Він також підтримує камери OV2640 і OV7670 і SD-карти. ESP32-CAM можна широко використовувати в різних додатках IoT. Він підходить для домашніх інтелектуальних пристроїв, промислового бездротового керування, бездротового моніторингу та інших програм IoT.

RGB світлодіодний модуль — це різновид світлодіодів RGB із загальним анодом, тобто окремі виступи перемикаються шляхом підключення до землі. Його функція полягатиме в тому, щоб надавати інформацію про поточний стан модуля камери (чи підключено він до мережі Wi-Fi, чи сталася помилка під час завантаження камери чи інші повідомлення про помилки) індикації різними кольорами.

Датчик температури DALLAS DS18B20 цей датчик дозволяє вимірювати температуру в діапазоні від −55 до +125 градусів за Цельсієм, а в діапазоні від −10 до +85 градусів за Цельсієм має гарантовану точність ±0,5°C. Це доступний у корпусі TO-92, який за розміром подібний до звичайних транзисторів, він також доступний у водонепроникному варіанті, де датчик запаяний у паличку з

нержавіючої сталі. Для зв'язку використовується шина OneWire, яка використовує лише один комунікаційний контакт. Цей датчик також підтримує так званий паразитний режим, коли для підключення датчика до мікроконтролера потрібні лише два дроти.

PIR-модуль, існує багато варіантів датчиків руху, призначених для застосування, подібного до цього. Для реалізаціїї пристрою можна обрати два датчика: модуль HC - SR501 та AM312.

Перший — HC - SR501, напруга живлення якого знаходиться в діапазоні від 4,5 до 20 В, а вихідна логіка становить 0 або 3,3 В. Розмір датчика 32х24 мм, тому він більший датчик, але дозволяє встановіть чутливість датчика та час за допомогою двох потенціометрів. Крім того, його відстань виявлення більш ніж достатня; виробник заявляє про відстань до 7 метрів з кутом чутливості 120°.

Другий — менший модуль під назвою AM312. Робоча напруга знаходиться в діапазоні від 2,7 до 12 В, а вихідна логіка така ж, як у попереднього модуля. Розміри пластини всього 10х8 мм. Однак відстань виявлення також менша, приблизно в діапазоні 3–5 м при куті сканування 110°. На рисунку 3.4 показано встановлені модулі камер в окремих кімнатах.

Рисунок 3.4 — Модулі камер ESP32 в коридорах і вітальні.

3.3.3 Блок керування

Незважаючи на те, що Raspberry Pi вже працює з четвертим поколінням розробки, економного варіанту Raspberry Pi Zero цілком достатньо для такої системи. Також Zero W має вбудований Wi-Fi і Bluetooth. Zero побудовано з одноядерним процесором ARMv6 із тактовою частотою 1 ГГц. Він також має 512 МБ оперативної пам’яті, аудіовізуальний вихід через Mini-HDMI, класичний роз’єм micro-USB 2.0 і живлення через micro-USB. Він також має 40-контактний інтерфейс GPIO.

Блок керування додатково розширюється за допомогою модуля GSM під назвою IOT-GA6-B. Це ідеальне рішення для пристроїв IoT, які спілкуються через послідовну лінію на рівні TTL, і підтримує голосові виклики, SMS, передачу даних GPRS і стандартні AT-команди. Також можна підключити цей модуль до мікропроцесора ESP32. Оскільки потрібно пересилати всі повідомлення з блоку керування на панель введення, а потім на термінал, але така реалізація буде надто складною. Щоб вирішити цю проблему, можна підключити пристрій безпосередньо до блоку управління. Таким чином, архітектура системи IoT є централізованою.

Для системи центральною точкою є блок керування, який також є посередником MQTT, який обмінюється данмим з модулями ESP32 і панелями введення.

3.4 Розробка програмного забезпечення

У цьому розділі описано окремі служби, які використовувалися в цій системі. Також є описи фрагментів вихідного коду, які завантажуються на окремі пристрої. Завдяки цьому плану можна краще зрозуміти, як працює вся збірка. На рисунку 3.5 показано отримане програмне забезпечення, яке описано в цьому підрозділі.

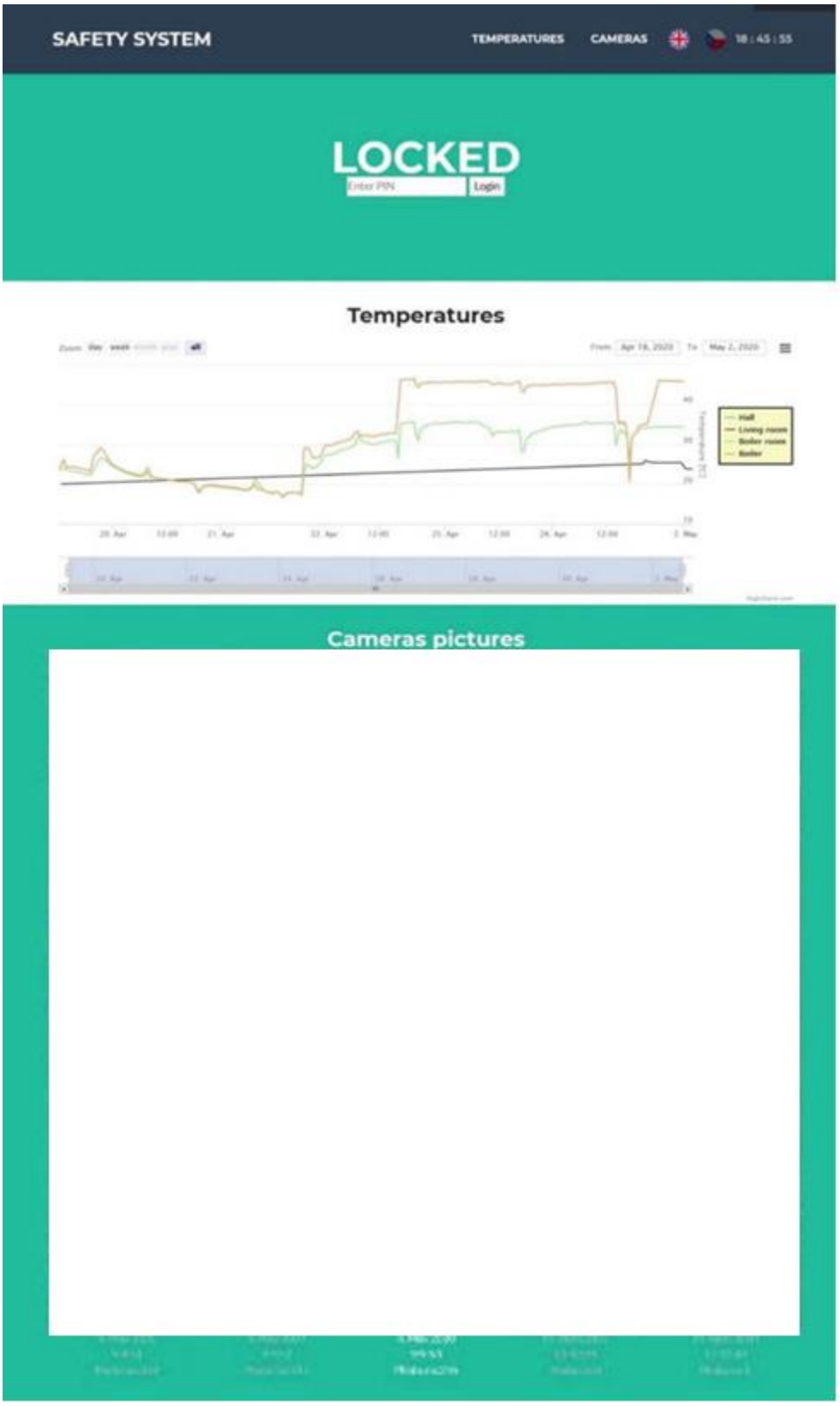

Рисунок 3.5 — Додаток для вимірювання та моніторингу

Операційна система — остання версія Raspbian під назвою Buster і працює на Debian Linux версії 4.16. Це операційна система, розроблена безпосередньо виробником цієї плати, завдяки чому має бути гарантована надійність і

стабільність цієї системи.

Raspberry Pi відповідає за запуск:

1) MQTT Server — сервіс для зв'язку з мікроконтролерами;

2) Apache 2 — веб-сервер;

3) MariaDB — база даних SQL для вимірюваних значень;

4) сценарії Python — охоплює всю логіку зв’язку з мікроконтролерами, а також збереження даних у базі даних.

3.5 Обмін даними

Весь зв'язок відбувається бездротовим способом, за допомогою Wi-Fi. Протокол, за яким взаємодіють окремі пристрої, — MQTT. MQTT (Message Queuing Telemetry Transport) — простий і невимогливий протокол для передачі повідомлень між клієнтами через центральну точку — брокера. Завдяки цій простоті його легко реалізувати навіть у вбудованих пристроях, і він відносно швидко поширюється. Для протоколу MQTT передача виконується за допомогою TCP і використовує шаблон проектування видавець-передплатник. Отже, є одна центральна точка (брокер MQTT), яка відповідає за обмін повідомленнями. У даному випадку у ролі брокера буде Raspberry Pi. На рисунку 3.5 показана блок-схема MQTT Communication. Повідомлення сортуються за так званими темами, і пристрій або публікує в даній темі (публікувати), тобто надсилає дані брокеру, який зберігає та розповсюджує їх на інші пристрої, або підписаний на тему чи кілька тем. Потім брокер надсилає всі повідомлення з заданою темою на пристрій. Звичайно, один пристрій може бути видавцем в одних темах і одночасно передплатником в інших.

Рисунок 3.6 — Зв'язок MQTT

3.6 Панель введення

Панель введення призначена, щоб замикати будинок натисканням ключа, а також відмикати його введенням 4-значного цифрового коду. Водночас він служить елементом для сповіщення зловмисника за допомогою сигналізації. У вихідному коді запрограмовано макроси, які дозволяють визначити, скільки разів можна ввести пін-код, коли будинок замкнено, час затримки спроб, відображення даних на екрані за замовчуванням, читання натискань клавіш і спілкування за допомогою протокол MQTT. Крім того, у вихідному коді, ми виконуємо конфігурацію Wi-Fi і перевіряємо, чи відбулося підключення до Wi-Fi, після чого слідують повторні спроби, світлодіодна сигналізація та реконфігурація пристрою.

Інший функціонал, запрограмований на панелі введення — це функція налаштування клієнта MQTT, яка визначає адресу брокера, порт зв’язку та інструкції з налаштуваннями отримання повідомлень від брокера. Основний цикл програми спочатку викликає функцію, яка перевіряє, чи ESP все ще підключений до мережі Wi-Fi, і якщо ні, він намагається підключитися кілька разів. Якщо це все одно не працює, пристрій перезавантажиться. Згодом викликається функція, яка перевіряє підключення до посередника MQTT і, якщо підключено, читає вхідні повідомлення від посередника. У вхідних повідомленнях міститься статус того, чи замкнений будинок, в цьому випадку цей факт відображається на дисплеї, якщо система у заблокованому стані виявляє зловмисника на одному з модулів камери, то спрацьовує сигналізація у вигляді пасивного динаміка. , який видає постійний тон певної частоти.

Також пристрій перевіряє натискання клавіш через певний проміжок часу, усунення підсвічування дисплея під час введення даних, перевірку неправильно введеного PIN-коду та блокування подальших спроб протягом певного проміжку часу, а також надсилання повідомлення під час спроби зламати пароль. Потім модуль додатково вирішує логіку перевірки статусу замка будинку та зв’язку з блоком керування. PIN-код зберігається в базі даних, розташованій на блоці керування. Блок керування передає статус будинку та поточний PIN-код через безпечний зв’язок.

3.7 Модуль камери

Основна його мета — зафіксувати рух у певній місцевості, якщо модуль був активований заздалегідь. Рух реєструється за допомогою датчика PIR, а потім знімаються фотографії з інтервалом в одну секунду, поки датчик PIR виявляє рух.

Інша мета — контролювати температуру та надсилати ці дані на блок керування. Вихідний код, який представлений у лістингу 3.1, містить макроси, які дозволяють налаштувати тривалість затримки між кожним захопленим зображенням, а також затримку між записом температури. Інші макроси використовуються для визначення контактів, до яких підключено PIR-модуль і датчик температури, а також точності, з якою вимірюється значення на датчику. Крім того, тут визначено катоди RGB LED. З точки зору програмного забезпечення, цей модуль містить загалом 6 бібліотек, які забезпечують надсилання захопленого зображення на сервер, завантаження камери, підключення до мережі Wi-Fi, зв’язок за допомогою протоколу MQTT, а також підключення та роботу з датчик температури.

Лістинг 3.1 — Бібліотеки та визначення макросів

```
#include "esp_http_client.h"
#include " esp_camera.h"
#include <Wi-Fi.h>
#include <PubSubClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define capture_interval 1000
#define temp_interval 300000
#define PIR 15
#define ONE_WIRE_BUS 14
#define TEMPERATURE_PRECISION 12
#define LedR 4
#define LedG 2
```

```
#define LedG 13
```

Програмне забезпечення цього модуля реалізує функції перевірки підключення до мережі Wi-Fi і перевірки підключення до брокера MQTT. Цей фрагмент коду представлено у лістинг 3.2. Іншими функціями є перевірка інтервалу вимірювання, обробка виміряних значень і відправка їх брокеру. Інша перевірка з'ясовує, чи отримав брокер повідомлення про зачинений будинок. Якщо це відбувається, то рух у зоні перевіряється шляхом зчитування значення з датчика PIR. Якщо рух було виявлено, брокеру надсилається звіт про проникнення в будинок. Одночасно перевіряється, чи минув час, необхідний для створення ще одного скріншота з моменту останнього циклу. Якщо так, викликається функція, щоб зробити знімок за допомогою підключеної камери. Потім це зображення надсилається і зберігається в каталозі.

Лістинг 3.2 — Фрагмент вихідного коду модуля камери ESP32

```
void loop()
(
WifiCheck();
mqttRun ();
temp_current_millis=
if (temp_current_millis- temp_last_capture_millis> temp_interval)
{
temp_last_capture_millis= millis();
Temperature ();
millis();
}
if (locked)
(
bool PIRSENSOR = digitalRead (PIR);
if (PIRSENSOR)
{
```

```
if (!last PIR)
{
}
Serial.println("Public Intruder");
MOTTclient.publish (PublishIntruder, MOTTid, MOTTpubQes);
current_millis = millis();
if (current_millis- last_capture_millis> capture_interval) // 1 sec elapsed
{
last_capture_millis= millis();
take_send_photo ();
}
else
last_p
last PIR PIRSENSOR;
```

3.8 Блок керування

Блок керування призначений для здійснення зв’язку за допомогою протоколу MQTT, зберігати вхідні дані в базі даних і надавати доступ до них, а також надавати інтерфейс HMI. У роботі використовується MariaDB для зберігання даних, яка є реляційною базою даних, розробленою спільнотою користувачів MySQL.4 з різними правами на використання бази даних. Таке рішення дозволить максимально захистити доступ до інформації. Основним користувачем є адміністратор із правами, пов’язаними zesp_db база даних, у якій зберігаються всі таблиці, номери телефонів, вимірювані значення, зроблені фотографії та окремі стани закриття та відкриття будинку.

Номери телефонів та імена користувачів, яким ці номери телефонів належать, зберігаються в таблиці Телефони БД. Інші два стовпці цієї таблиці вказують права окремих телефонних номерів, один стовпець вказує на право розблокувати або заблокувати будинок за допомогою SMS, а інший стовпець

вказує на маніпуляції зі шлюзом. Останнім є LOGIN, який зберігає 4-значний PIN-код для розблокування будинку.

3.9 Безпека системи

Дана система, повинна відповідати принципам безпеки. Оскільки це низка взаємопов’язаних технологій, існує кілька рекомендацій щодо заходів безпеки, від цілком звичайних до складних рішень. Система використовує ряд апаратних модулів, комунікаційні елементи, веб-сервер, базу даних, конфігурований доступ до блоку керування, вхідний пін для розблокування системи, і все це має бути захищеним. Основна рекомендація безпеки полягає в тому, щоб відокремити бездротову мережу для модулів IoT системи від бездротової мережі будинку, в якому ми використовуємо ПК, ноутбуки та інші звичайні пристрої. Це може бути зроблено або шляхом повного розділення та роботи двох окремих мереж, або за допомогою різних варіантів VLAN і мікросегментації мережі, залежно від доступного апаратного забезпечення конкретного рішення. У нашому випадку це абсолютно окрема мережа.

Іншим важливим елементом є принцип превентивного захисту блоку керування Raspberry Pi, забезпечення віддаленого доступу SSH, дотримання принципів безпеки та найкращих практик для бази даних MariaDB, таких як запобігання запуску mysqld як root, Limitssh доступ, Limisudo доступ до MariaDB і використання плагінів безпеки. Обмеження кількості спроб введення PIN-коду на панелі приладів вирішується можливістю налаштування вибраного часу затримки після трьох невдалих послідовних спроб, включаючи відправку сповіщення на мобільний телефон на попередньо вибрані телефонні номери.

Захист зв’язку протоколу MQTT важливий, оскільки цей протокол спочатку був розроблений не для безпеки, а для його доступності та невимогливої реалізації. Ми можемо застосувати безпеку цього зв’язку в реалізаціях пізніших брокерів Mosquitto. Це робиться за допомогою механізмів автентифікації та авторизації, які дозволяють додавати плагіни. Це робиться за допомогою механізмів автентифікації та авторизації, які дозволяють додавати плагіни. Це

автентифікація клієнта за допомогою ідентифікаторів клієнта, списку контролю доступу або сертифікатів клієнтів. Щоб захистити вміст ваших повідомлень MQTT, можна використовувати TLS або SSL Security і Payload Encryption.

4 ДОСДІДЖЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

4.1 Встановлення та тестування системи безпеки

Дана система безпеки в першу чергу призначається для приміщень, тому, щоб успішно експлуатувати пристрій, він повинен мати захист від механічних пошкоджень і водночас забезпечувати максимально зручну установку. Тому, щоб виконати ці вимоги, можна використовувати 3D-принтер, на якому можна виготовляти корпуси для модулів камер і панелей введення з клавіатурою і сенсорним екраном.

Процес встановлення системи відбувається наступним чином: у вхідній панелі закріплюється ESP32, LCD, клавіатура, динамік і перетворювач напруги DC-DC, який знизить напругу розподілу 24В постійного струму до необхідних 5В постійного струму.

Корпус призначений для настінного кріплення навколо вхідних дверей, висота якого підходить для користувача. Модуль камери містить такі елементи: ESP32-CAM, датчик температури, PIR-датчик, RGB LED, IR LED, NPN-транзистор, джерело постійного струму, DC-DC перетворювач і антену для Wi-Fi. Оптимальним рішенням буде корпус камери трикутної форми, щоб її можна було встановити в кутку кімнати, щоб вона максимально зливалася з навколишніми стінами і жодним чином не порушувала цілісність приміщення. Після впровадження апаратних модулів, створення корпусу для монтажу і реалізації програмного забезпечення всієї системи відбувається етап монтажу та перевірки функціональності всіх модулів, встановлених у приміщенні. На рисунку 4.1 показано тестування панелі введення, сенсорного екрану, одного модуля камери та відображення моніторингу температури та зроблених фотографій на веб-сервері в окремих режимах, що імітують закритий і незамкнений будинок. Цей тест також стосується перевірки сигналізації та відправки повідомлень модулем GSM на попередньо вибрані номери телефонів.

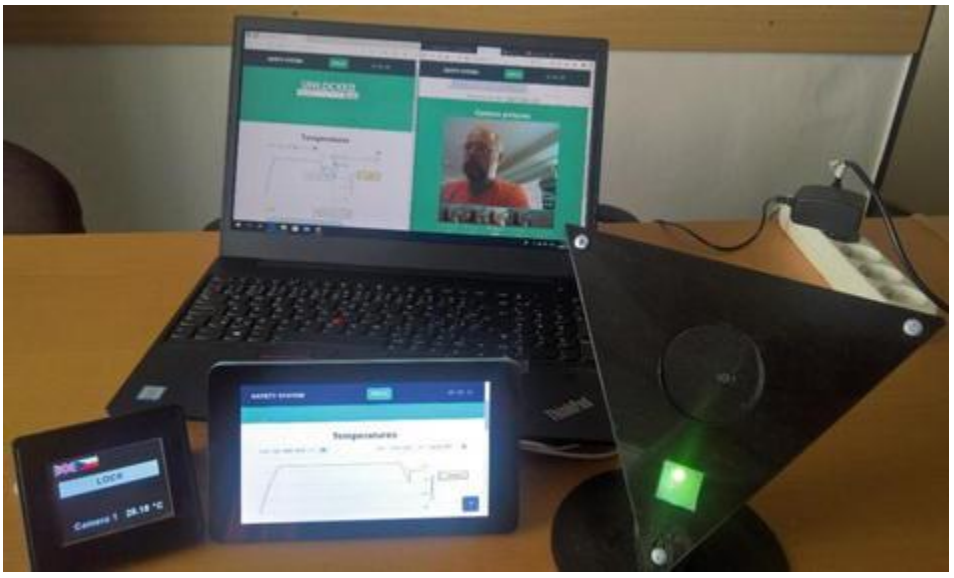

Рисунок 4.1 — Тестування апаратних модулів і функціональності програмного забезпечення

В приміщені може встановлюватись три модулі камер і дві панелі введення, одну з сенсорним екраном, а іншу з мембранною клавіатурою для більш комфортного використання в запилених приміщеннях, один модуль для збору температури і один блок керування Raspberry Pi Zero W.

Усі компоненти пристрою живляться від джерела постійної напруги 12 В, так що навіть у довгостроковій перспективі напруга окремих пристроїв становить не менше 5 В. Під час активації, модуль камери сигналізує про свою ініціалізацію зеленим кольором. Панель введення з мембранною клавіатурою можуть розміщуватись біля головного входу, в той же час панель з сенсорним екраном біля приміщення яке потребує додаткового контролю цілісності. Raspberry Pi Zero W знаходиться в розподільному щитку разом з GSM-модемом і релейним модулем. Після підключення всіх модулів деякі частини коду були налаштовані для забезпечення максимально можливого комфорту використання та надійності.

4.2 Веб-сервер та мобільний додаток

Raspberry PI запускає веб-сервер, який працює через службу Apache 2. Ця служба запускає кілька сценаріїв PHP, доповнених структурою HTML. Потім JavaScript слідкує за динамікою сторінок разом із jQuery.

Екран програми на веб-сервері розділений на три частини. Це надає можливість входу за допомогою введення PIN-коду. Після успішного входу будинок можна розблокувати або заблокувати дистанційно це можна побачити в програмі на мобільному пристрої (рисунок 4.2) .

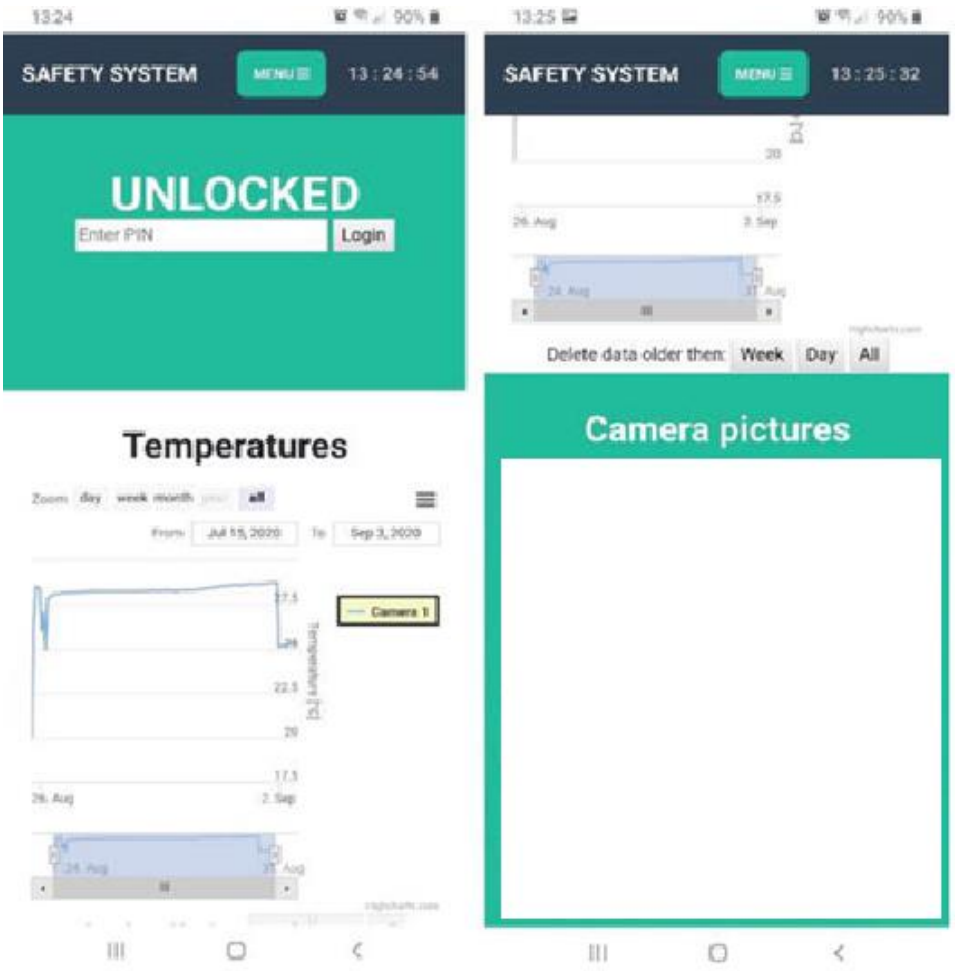

Рисунок 4.2 — Додаток на мобільному пристрої

HighCharts графік чітко показує зміну температури в часі, виміряної на окремих пристроях. У діаграмі можна вибрати конкретний пристрій, з якого ми хочемо бачити зміну температур. Також можна вибрати часовий діапазон для відображення температури. Існують також додаткові параметри, за допомогою яких графіки можна експортувати або в зображення, або, наприклад, на аркуш Excel або подібні файли.

Зображення сортуються відповідно до дати, яка відображається для кожної фотографії, і їм присвоюється порядковий номер для більшої чіткості. Сайт

можна перемикати відповідно до мовних уподобань на місцеве середовище, якщо це необхідно. Цей сайт також було налаштовано для мобільних пристроїв, таких як смартфони.

4.3 Сценарії Python

Ці скрипти забезпечують зв’язок для протоколу MQTT, а також для зберігання даних у базі даних і зв’язку з модулем GSM. На Raspberry є три файли python, один з яких діє як клієнт MQTT. Він прослуховує все, що надсилається брокеру, і виконує певну дію на основі вхідного повідомлення. Другий сценарій python містить функції, які дозволяють отримати доступ до бази даних. Третій зв'язується через послідовну лінію з модулем GSM і забезпечує зв'язок з кінцевим користувачем.

Сценарій Python містить функції, які використовуються для отримання значень із бази даних, наприклад поточного стану будинку (заблоковано/розблоковано). Інша функція використовується для запису значень у таблиці з температурами, які надходять як повідомлення від модулів камери через протокол MQTT.

Перший, on_message, викликається, коли брокер MQTT надсилає повідомлення. Приклад цього коду представлено у лістинг 4.1. У цій функції повідомлення фільтрується за темою. У випадку теми під назвою «Оновити» ми знаємо, що новий пристрій щойно приєднався до спілкування та запитує оновлення даних. Тому до цієї команди надсилається повідомлення, яке викликає функцію, яка зчитує поточний пін-код і статус будинку з бази даних. Він надсилає ці повідомлення після зашифрованого зв’язку.

Лістинг 4.1 — Фрагмент скрипту python для обміну MQTT

```
def on_message(client, userdata, message):
global last_LockState, last IntruderState
if ("Refresh" in message.topic):
print("Requirement for Refresh")
client.publish ("Read/ESP/Lock, payload-last LockState, qos=1, retain=False)
```

```
time.sleep(1)
client.publish ("Read/ESP/Pin, payload-Read FromDB("Read/ESP/Pin"), gos=1, retain=False)
time.sleep (1)
payload-last IntruderState, qos=1, retain=False)
elif ("Write" in message.topic):
print("Topic: + message.topic + Data:" + message.payload.decode('utf-8'))
SaveToDB (message.topic, message.payload.decode('utf-8'))
client.publish('Read/ESP/Intruder",
def
1 while client.connected_flag:
time.sleep (2)
publish_state():
global last LockState, last IntruderState
if ReadFromDB("Read/ESP/Lock") is not last_LockState:
last LockState ReadFromDB("Read/ESP/Lock')
client.publish('Read/ESP/Lock', payload-last_LockState, qos-1, retain=False)
if last LockState and ReadFromDB("Read/ESP/Intruder") is not last IntruderState:
last IntruderState ReadFromDB("Read/ESP/Intruder")
client.publish('Read/ESP/Intruder", payload-ReadFromDB("Read/ESP/Intruder"), qos-1, retain-False)
```

Потім друга функція відповідає за зміну стану будинку (відмикання/замикання). Якщо це станеться, повідомлення буде повторно передано через 5 секунд, щоб бути на 100% впевненим, що всі підключені пристрої отримають це повідомлення. З такою ж частотою також надсилаються повідомлення про стан зловмисника (тобто чи один із пристроїв камери зафіксував рух), якщо будинок знаходиться в замкненому стані.

Сценарій, який забезпечує зв'язок із модулем GSM, спочатку налаштовує цей модуль. Потім він запускає нескінченний цикл, у якому перевіряє, чи не

проник зловмисник у будинок. Якщо так, то на всі телефонні номери, зареєстровані в базі даних, надсилається SMS-повідомлення. У разі набору телефонного номера спочатку перевіряється, чи номер записаний у базі даних і чи має відповідні права для відкриття шлюзу. Потім є контакт на релейному модулі, який підключений до Raspberry Pi, він замикається, і таким чином ворота відкриваються. Надсилаючи SMS «заблокувати» або «розблокувати», можна маніпулювати системою безпеки будинку. Приклад цього кодупредставлено у лістинг 4.2. Відправляючи SMS, система спочатку перевіряє, чи номер, з якого надіслано SMS, зареєстрований у базі разом із відповідними дозволами. Згодом на цей номер телефону приходить SMS-повідомлення. Це повідомлення містить поточний час, стан будинку (розблоковано/заблоковано), останні вимірювані температури всіма датчиками та, що не менш важливо, залишок кредиту на SIM-картці (щоб користувач знав, коли потрібно поповнити мобільний кредит).

Лістинг 4.2 — Фрагмент скрипта python зі зв'язком GSM модуля

```
while True:
reply = bytes.decode(ser.read(ser.inWaiting()))
if reply !="":
print (reply)
if "+CLIP:" in reply: # if calling
phoneNumber = reply [reply.index("+CLIP: ") + 8: reply.index("+CLIP: ") + 28]
for x in range(len(PhoneTable)):
if PhoneTable[x] [e] in phoneNumber and 1 is PhoneTable[x][2]:
print("i am opening the gate")
GPIO.output (17, GPIO. LOW)
time.sleep (2)
GPIO.output (17, GPIO.HIGH)
# Send SMS (phoneNumber)
ser.write(str.encode('ATH\r')) #stop call
if +CMT: in reply: # if SMS
phoneNumber
```

```
print (phoneNumber)
for x in range(len(PhoneTable)):
reply[reply.index("+CMT: ") + 8: reply.index("+CMT: ") +20]
print (phoneNumber)
for x in range(len (Phone Table));
if PhoneTable[x][0] in phoneNumber and 1 is PhoneTable[x][1]:
if "lock" in reply.lower():
print("i am locking the house")
ChangeState("true")
time.sleep (1)
Send SMS (phoneNumber)
elif "unlock" in reply.lower() or "unlocking" in reply.lower():
print("i am unlocking the house")
ChangeState("false")
time.sleep (1)
Send SMS (phoneNumber)
time.sleep (1)
else:
SendSMS (phoneNumber)
# Clear buf
# Clear buf
ser.flushInput()
ser.flushOutput()
time. Sleep(1)
```

5 ЕКОНОМІЧНА ЧАСТИНА

Для успішного впровадження науково-технічної розробки критично важливо, щоб вона відповідала сучасним вимогам науково-технічного прогресу та враховувала економічні аспекти. Надання оцінки економічної ефективності результатів науково-дослідної роботи є важливою частиною цього процесу. Дослідження, яке представлене у магістерській роботі і присвячене розробці та вивченню "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT", віднесено до науково-технічних робіт, спрямованих на виведення на ринок. Рішення про комерціалізацію розробки може бути прийняте протягом проведення самої роботи, відкриваючи можливості для подальшого виведення на ринок. Цей напрямок визначається як пріоритетний, оскільки розроблені результати можуть бути корисними для різних зацікавлених сторін і приносити економічні вигоди. Однак для успішної реалізації цього процесу ключовим є залучення зацікавленого інвестора, який виявить інтерес до втілення даного проекту, і переконання його у доцільності інвестування у цю розробку. З метою досягнення цього завдання були визначені такі етапи виконання робіт:

— проведення комерційного аудиту науково-технічної розробки, включаючи визначення науково-технічного рівня та комерційного потенціалу.

— розрахунок витрат на реалізацію науково-технічної розробки.

— проведення розрахунку економічної ефективності впровадження та комерціалізації науково-технічної розробки для потенційного інвестора, а також обґрунтування економічної доцільності комерціалізації з точки зору інвестора.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" є розробка мікрокомп'ютерної системи оповіщення та контролю цілісності охоронного об'єкта з функцією передачі даних

стану бездротовою мережею.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					

8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці. Для оцінки науково-технічного рівня і комерційного потенціалу розробки експертами було запрошено трьох незалежних експертів Вінницького національного технічного університету кафедри «Обчислювальної техніки»: Мартинюк Тетяна Борисівна доктор технічних наук, професор; Крупельницький Леонід Віталійович кандидат технічних наук, доцент; Дудник Олександр Вікторович кандидат технічних наук, асистент.

Таблиця 5.2 — Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	Мартинюк Т. Б.	Крупельницький Л. В.	Дудник О. В.
	Бали, виставлені експертами:		
1. Технічна здійсненність концепції	4	4	5
2. Ринкові переваги (наявність аналогів)	2	3	3
3. Ринкові переваги (ціна продукту)	4	4	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	3	4	5
12. Практична здійсненність (розробка документів)	4	5	4
Сума балів	$СБ_1$=39	$СБ_2$=44	$СБ_3$=45
Середньоарифметична сума балів $СБ_c$	42,7		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3.

Таблиця 5.3 — Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ ,розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" становить 43 бали, що, відповідно до таблиці 5.3 рівень комерційного потенціалу розробки високий, що свідчить про комерційну важливість проведення даних досліджень.

Магістерська кваліфікаційна робота "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок, тобто при цьому відбувається комерціалізація науково-технічної розробки. Цей напрямок є для нас пріоритетним, оскільки результатами розробки можуть користуватися не тільки самі розробники, а й інші споживачі, отримуючи при цьому суттєвий економічний ефект.

5.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

Одиничний параметричний індекс розраховуємо за формулою:

$$q_i = \frac{P_i}{P_{базі}}. \quad (5.1)$$

де q_i – одиничний параметричний індекс, розрахований за i-м параметром;

P_i – значення i-го параметра виробу;

$P_{базі}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 5.4.

Таблиця 5.4 — Основні техніко-економічні показники аналога та розробки, що проектується

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Дальність зв'язку, км	2	15	7,5	25%
Частота, МГц	868,0-868,6	868	1	25%
Період опитування датчика, с	12	5	2,4	30%
Кут чутливості, °	120	88	1,36	20%

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою:

$$I_{НП} = \prod_{i=1}^{n} q_i \,, \tag{5.2}$$

де $I_{нп}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i-м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{нп} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра.

$$I_{ТП} = \sum_{i=1}^{n} q_i \cdot \alpha_i \,, \tag{5.3}$$

де $I_{тп}$ – груповий параметричний індекс за технічними показниками (порівняно з

виробом-аналогом);

q_i – одиничний параметричний показник *i*-го параметра;

α_i – вагомість *i*-го параметричного показника, $\sum_{i=1}^{n}\alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 5.4.

$$I_{тп} = 7,5 \cdot 0,25 + 1 \cdot 0,15 + 2,4 \cdot 0,30 + 1,36 \cdot 0,2 = 3,02.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою.

$$I_{ЕП} = \sum_{i=1}^{m} q_i \cdot \beta_i \,, \qquad (5.4)$$

де $I_{ЕП}$ – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр *i*-го виду;

β_i – частка *i*-го економічного параметра, $\sum_{i=1}^{m}\beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{ЕП} = 0,75 \cdot 0,5 + 0,86 \cdot 0,5 = 0,80.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою**:**

$$K_{ІНТ} = I_{НП} \cdot \frac{I_{ТП}}{I_{ЕП}}, \quad (5.5)$$

$$К_{ІНТ} = 1 \cdot 3,02 / 0,80 = 3,8.$$

Інтегральний показник конкурентоспроможності $К_{ІНТ} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT", під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 15000 \cdot 5 / 21 = 3409 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
1. Керівник проекту	15000	681,8	5	3409
2. Інженер	9000	409,1	45	18409
Всього				21818

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \quad (5.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} , \quad (5.8)$$

де M_M — розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо Мм=6500 грн;

K_i — коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б);

K_c — мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p — середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ — тривалість зміни, год.

$$C_1 = 6500{,}00 \cdot 1 \cdot 1{,}65 / (21 \cdot 8) = 65{,}8 \text{ грн.}$$

$$З_{р1} = 65{,}8 \cdot 1 = 65{,}8 \text{ грн.}$$

Таблиця 5.6 — Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
1.Підготовчі	1	1	65,8	65,8
2.Монтажні	2	3	88,8	177,7
3.Складальні	1	4	98,7	98,7
4.Налагоджувальні	3	2	72,4	217,2
5.Випробувальні	1	4	59,8	59,8
Всього				619,2

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \qquad (5.9)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод}$ = (21818+ 619,2) · 11 / 100% = 2468,11 грн.

5.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.10)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$Зн = (21818+ 619,2+2468,11) \cdot 22 / 100\% = 5479,19 \text{ грн.}$$

5.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT".

Витрати на матеріали ($М$), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (5.11)$$

де $Н_j$ – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 ... 1,15*);

B_j – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці.

Таблиця 5.7 — Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Вартість витраченого матеріалу, грн
Дріт	89	0,05	4,45
Всього			4,45
З врахуванням коефіцієнта транспортування			4,89

5.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна система підтримки підприємств малого бізнесу у сфері послуг».

Витрати на комплектуючі К, що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$К = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i \quad \text{грн.}, \qquad (5.12)$$

де $Н_i$ – кількість комплектуючих *i*-го виду, шт.;

$Ц_i$ – ціна комплектуючих *i*-го виду, грн.;

$К_i$ – коефіцієнт транспортних витрат, $К_i$ = (1,1…1,15);

n – кількість видів комплектуючих.

Зроблені розрахунки бажано звести до таблиці:

Таблиця 5.8 – Витрати на комлектуючі

Найменування комплектувальних	Кількість	Ціна за штуку, грн.	Сума, грн.
Мікрокомп'ютер	1	1110	1110
Карта пам’яті	1	120	120
Модем	1	450	450
DC – DC перетворювач	2	90	180

RTC модуль	1	40	40
ESP32 – CAM	1	220	220
PIR сенсор	1	22	22
Діод	1	1,5	1,5
ESP32	1	140	140
LCD дисплей	1	220	220
Температурний сенсор	1	35	35
Блок живлення	1	55	55
Витратні матеріали	1	40,0	40,0
Всього з врахування коефіцієнт транспортних витрат			2896,85

5.3.5 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (5.13)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$А_{обл} = (35000 \cdot 2) / (2 \cdot 12) = 2916,67 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер	35000	2	2	2916,67
Робоче місце розробника ПЗ	257000	20	2	2141,67
Всього				5058,33

5.3.6 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (5.14)$$

де W_{yi} — встановлена потужність обладнання на визначеному етапі, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e$ = 7,5 грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,25 \cdot 290,0 \cdot 7,5 \cdot 0,5 / 0,8 = 339,84 \text{ грн.}$$

5.3.7 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов’язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з’їзди, конференції, наради, пов’язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_o + З_p) \cdot \frac{Н_{св}}{100\%}, \quad (5.15)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$$В_{св} = (21818+ 619,2) \cdot 20 / 100\% = 4487,47 \text{ грн.}$$

5.3.8 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (5.16)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$$І_в = (21818+ 619,2) \cdot 50 / 100\% = 11218,67 \text{ грн.}$$

5.3.9 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{H_{нзв}}{100\%}\ , \qquad (5.17)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$$В_{нзв} = (21818 + 619{,}2) \cdot 100 / 100\% = 22437{,}33 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + I_в + B_{нзв}\ . \quad (4.18)$$

$$В_{заг} = 21818 + 619{,}2 + 2468{,}11 + 5479{,}19 + 4{,}89 + 2896{,}85 + 5058{,}33 + 339{,}84 + 4487{,}47$$
$$+ 11218{,}67 + 22437{,}33 = 76828{,}03 \text{ грн.}$$

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}\ , \qquad (5.19)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,5.

$$ЗВ = 76828{,}03 / 0{,}5 = 153656{,}05 \text{ грн.}$$

5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може

отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 1 особа

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 3000,00 грн;

$\pm\Delta Ц_o$ – зміна вартості послуги від впровадження результатів, приймемо зростання на 500,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою:

$$\Delta П_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (5.20)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0{,}8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році ϑ =*18%*;

Збільшення чистого прибутку 1-го року:

$\Delta П_1$ =(1·500+3000·400)·0,83·0,4·(1-0,18/100%)=239242,51 грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2$ =(1·500+3000·(400+300))·0,83·0,4·(1-0,18/100%)=419024,93 грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3$ =(1·500+3000·(400+300+200))·0,83·0,4·(1-0,18/100%)=538603,48 грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (5.21)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =18%;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 239242,51 /(1+0,18)^1+419024,93 /(1+0,18)^2+538603,48 /(1+0,18)^3=$$
$$= 803499,85 \text{ грн.}$$

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \tag{5.22}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 153656,05 грн.

$PV = k_{інв} \cdot ЗВ$ = 2 · 153656,05 = 307312,11 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{5.23}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 803499,85 грн;

PV – теперішня вартість початкових інвестицій, 307312,11 грн.

$E_{абс} = ПП - PV$ = 803499,85 - 307312,11 = 496187,74 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{5.24}$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 3 роки.

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+496187,74\ /307312,11)^{1/3} - 1 = 0,62.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f\ , \qquad (5.25)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,25.

$\tau_{мін}$ = 0,1+0,25 = 0,35 < 0,62 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія онтологічного моделювання бази знань з організації бібліотеки» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}\ , \qquad (5.26)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,62 = 1,6 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу: згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT" становить 43 бали, що, свідчить про комерційну важливість проведення даних досліджень оскільки рівень комерційного потенціалу розробки високий.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 3,02 рази.

Також термін окупності становить 1,6 роки, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою "Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT".

ВИСНОВКИ

У рамках виконання магістерської кваліфікаційної роботи було розроблено систему оповіщення та контролю цілісності засобами IoT.

У першому розділі було розглянуто, історію виникнення IoT, яке місце займає Інтернет речей у сучасному світі, особливості його роботи, бездротові технології IoT, що використовуються для пердачі даних.

У другому розділі було зроблено вибір мікрокомп’ютерної платорми, опираючись на технічні характеристики.

У третьому розділі було спроектовано апаратну та програмну частини мікрокомп’ютерної системи. Описано головну концепцію проектованої системи, розроблено структуну схему системи необхідну для кращого розуміння, як саме влаштована система оповіщення та контролю цілісності, проведено вибір електронних компонентів майбутньої системи. Докладно описано головні вузли приладу, такі як: панель введення з сенсорним екраном, модуль камери, блок керуваня. Також у даному розділі описується розробка програмного забезпечення, процес обміну даних за допомогою бездротової технології LoRaWAN та піднімаються питання безпеки даної системи.

У четвертому розділі було досліджено та протестовано систему оповіщення та контролю цілісності. Запропоновано варіанти встановлення у приміщенні. У даному розділі описується сервер та мобільний додаток у якому можна відслідковувати показники датчиків, а також зображення з камер спостереження, описано сценарії Python, що забезпечують зв’язок для протоколу MQTT.

Проведено економічні розрахунки для проектованої системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Захист інформації в системах IoT, опис захисту та безпеки пристроїв. URL: https://www.anti-malware.ru/practice/solutions/iot-reference-architecture-protection-part-2

2. Опис технології LoRaWAN. URL: https://deps.ua/knowegable-base.ru/spravochnaya-informatsiya/item/66633.html

3. Огляд технологій Інтернету речей. URL: http://ua.automation.com/content/wifi-bluetooth-ili-zigbee-kakoj-standart-luchshe

4. ДобрушськийС.UEBA,або поведінкова аналітика.Базова функція всіх систем безпеки майбутнього // Інформаційна безпека, 2017. - № 4. URL:http://www.itsec.com/articles2/Inf_security/ueba--ili-povedencheskaya-analitika-базова-функція-всіх-систем-безпеки/

5. Інформація щодо призначення систем відеоспостереження. URL: https://www.kp.ru/guide/sistemy-bezopasnosti.html

6. Датчики охоронної системи. URL: https://secur.ua/ajax

7. Miller, DR Security Information and Event Management (SIEM) implementation / DR Information Technology. Information Security. Information Assurancy. URL:http://www.isaca.org.

8. Zimmermann H.-J. Fuzzy Sets, Decision Making and Expert Systems / H.-J. Zimmermann. - Kluwer: Dordrecht, 1987. - 335 p

9. Lora AT COMMANDGUIDE.REYAX TECHNOLOGY CO., LTD, 2018 року. aspberry Pi Documentation. URL:www.raspberrypi.org

10. A Simple Explanation Of 'The Internet Of Things'. URL: https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanationinternet-things-that-anyone-can-understand/?sh=5c7c02081d09

11. Встановлення операційної системи. URL: https://raspberrypi.ru/operating_system_install.

12. Internet of things: все, що потрібно знати про інтернет речей і про майбутнє сучасної цивілізації. URL: https://www.everest.ua/ru/internet-of-things-vse-chto-nuzhno-znat-ob-ynternete-veshhej-y-o-budushhem-sovremennoj-czyvylyzaczyy/.

13. Що таке Arduino. URL:www.arduino.cc

14. ArduinoMEGA. Технічні характеристики. URL: www.arduino.com

15. Hurt A. E. How Bluetooth Works. Cavendish Square Publishing LLC, 2018.

16. Paetz D. C. Z-Wave Essentials. CreateSpace Independent Publishing Platform, 2018. 310 p.

17. Schwartz T. Wi-Fi. Paris : Micro application, 2004. 428 c.

18. Seller O. LoRaWAN Security. Journal of ICT Standardization. 2021. URL: https://doi.org/10.13052/jicts2245-800x.915

19. ZigBee Technology / Vaishali та ін. International Journal of Advanced Research in Science, Communication and Technology. 2022. С. 688–692. URL: https://doi.org/10.48175/ijarsct-7036

20. Hillar G. C. MQTT Essentials - A Lightweight IoT Protocol. Packt Publishing, 2017. 280 c.

ДОДАТОК А

Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

_____проф., д.т.н. О. Д. Азаров

«___» _____20___р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«Мікрокомп'терна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT»

08-54.МКР.017.00.000 ПЗ

Науковий керівник

к.т.н., доц. каф. ОТ

_____ Богомолов С.В.

виконав:

магістрант 2 курсу,

_____Твердохліб Н.М.

Вінниця 2023

1. Підстава виконання магістерської кваліфікаційної роботи

1.1 Одним із найбільш актуальних напрямів розвитку охоронних систем в останні роки став Інтернет речей. Особиста безпека одна з найбільших потреб людини. Завдяки розвитку технологій у теперішній час можна легко вирішити проблеми безпеки.

1.2 Наказ про затвердження теми МКР

2 Мета і призначенням МКР

2.1 Метою роботи є розробка мікрокомп'ютерної системи оповіщення та контролю цілісності охоронного об'єкта з функцією передачі даних стану бездротовою мережею LoRaWAN.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

3 Вихідні дані для виконання МКР

Вихідні дані для виконання МКР: опис бездротових технологій передачі даних, технічний опис Raspberry Pi, та електронних компонентів.

4 Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

— забезпечити постійну взаємодію всіх коспонентів проектованої системи;

— провести та тестування системи;

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в табл.А.1.

6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів,

анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

Таблиця А.1 — Етапи МКР

з/п	Назва етапів виконання бакалаврського проекту	Строк виконання етапів роботи	Примітка
1	Постановка мети та задач проекту	17.10.23	
2	Огляд і аналіз	20.10-25.10.23	
3	Огляд і аналіз теоретичних аспектів IoT	26.10-3.11.23	
4	Вибір мікрокомп'ютерної платформи	4.11-11.11.23	
5	Проектування апаратної та програмної частин	12.11-20.11.23	
6	Дослідження та тестування системи	21.11-26.11.23	
7	Розрахунок економічної частини роботи	26.11-4.12.23	
8	Аналіз виконання проекту. Висновки. Додатки	5.12-6.12.23	
9	Оформлення пояснювальної записки та ілюстративного матеріалу	7.12.23	
10	Перевірка якості виконання бакалаврського проекту та усунення недоліків	18.12.23	

7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

При оформлюванні МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі

спеціальності 123 — «Комп’ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21.

ДОДАТОК Б

Схема розпінування Raspberry Pi

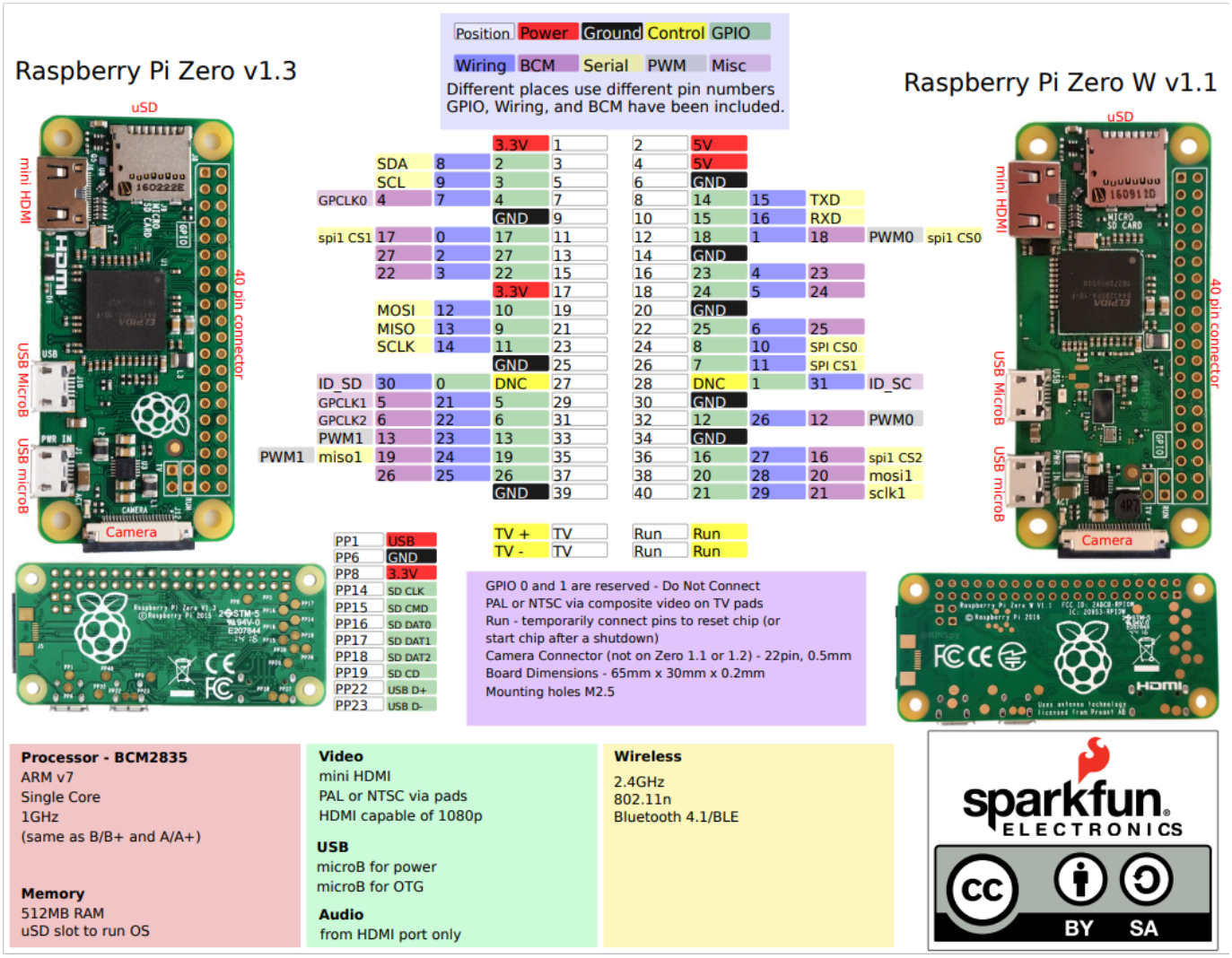

Рисунок Б.1 — Схема розпінування Raspberry Pi

ДОДАТОК В

Код для модуля ESP 32

Лістинг В.1 — Програмний код для модуля ESP 32

```
#include "appGlobals.h"

char camModel[10];

static void prepCam() {

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sccb_sda = SIOD_GPIO_NUM;
  config.pin_sccb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = xclkMhz * 1000000;
  config.pixel_format = PIXFORMAT_JPEG;
  config.grab_mode = CAMERA_GRAB_LATEST;
  // init with high specs to pre-allocate larger buffers
  config.fb_location = CAMERA_FB_IN_PSRAM;
#if CONFIG_IDF_TARGET_ESP32S3
  config.frame_size = FRAMESIZE_QSXGA; // 8M
#else
  config.frame_size = FRAMESIZE_UXGA;  // 4M
#endif
  config.jpeg_quality = 10;
  config.fb_count = FB_BUFFERS;

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
```

```
  pinMode(14, INPUT_PULLUP);
#endif


  if (psramFound()) {
    esp_err_t err = ESP_FAIL;
    uint8_t retries = 2;
    while (retries && err != ESP_OK) {
      err = esp_camera_init(&config);
      if (err != ESP_OK) {

        digitalWrite(PWDN_GPIO_NUM, 1);
        delay(100);
        digitalWrite(PWDN_GPIO_NUM, 0);
        delay(100);
        retries--;
      }
    }
    if (err != ESP_OK) snprintf(startupFailure, SF_LEN, "Startup Failure: Camera init error
0x%x", err);
    else {
      sensor_t * s = esp_camera_sensor_get();
      switch (s->id.PID) {
        case (OV2640_PID):
          strcpy(camModel, "OV2640");
        break;
        case (OV3660_PID):
          strcpy(camModel, "OV3660");
        break;
        case (OV5640_PID):
          strcpy(camModel, "OV5640");
        break;
        default:
          strcpy(camModel, "Other");
        break;
      }
      LOG_INF("Camera init OK for model %s on board %s", camModel, CAM_BOARD);

      if (s->id.PID == OV3660_PID) {

        s->set_vflip(s, 1);//flip it back
        s->set_brightness(s, 1);//up the brightness just a bit
        s->set_saturation(s, -2);//lower the saturation
      }
```

```
      char fsizePtr[4];
      if (retrieveConfigVal("framesize", fsizePtr)) s->set_framesize(s,
(framesize_t)(atoi(fsizePtr)));
      else s->set_framesize(s, FRAMESIZE_SVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE)
      s->set_vflip(s, 1);
      s->set_hmirror(s, 1);
#endif

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
    s->set_vflip(s, 1);
    s->set_hmirror(s, 1);
#endif

#if defined(CAMERA_MODEL_ESP32S3_EYE)
    s->set_vflip(s, 1);
#endif
    }
  }
  debugMemory("prepCam");
}

void setup() {
  logSetup();

  startStorage();

  loadConfig();

  if (psramFound()) prepCam();
  else snprintf(startupFailure, SF_LEN, "Startup Failure: Need PSRAM to be enabled");

#ifdef DEV_ONLY
  devSetup();
#endif

  startWifi();

  startWebServer();
  if (strlen(startupFailure)) LOG_ERR("%s", startupFailure);
  else {

    startSustainTasks();
```

```
    prepSMTP();
    prepUpload();
    prepPeripherals();
    prepMic();
    prepTelemetry();
    prepTelegram();
    prepRecording();
    LOG_INF("Camera model %s on board %s ready @ %uMHz", camModel,
CAM_BOARD, xclkMhz);
    checkMemory();
  }
}

void loop() {

  LOG_INF("================ Total tasks: %u ===============\n",
uxTaskGetNumberOfTasks() - 1);
  delay(1000);
  vTaskDelete(NULL);
}
```

ДОДАТОК Г

Лістинг протоколу передачі даних

Лістинг Г.1 — Лістинг протоколу MQTT

```
#include "appGlobals.h"
#include "mqtt_client.h"

char mqtt_broker[MAX_HOST_LEN] = "";
char mqtt_port[5] = "";
char mqtt_user[MAX_HOST_LEN] = "";
char mqtt_user_Pass[MAX_PWD_LEN] = "";
char mqtt_topic_prefix[FILE_NAME_LEN / 2] = "";

#define MQTT_LWT_QOS 2
#define MQTT_LWT_RETAIN 1
#define MQTT_RETAIN 0
#define MQTT_QOS 1

bool mqtt_active = false;
bool mqttRunning = false;
bool mqttConnected = false;
esp_mqtt_client_handle_t mqtt_client = nullptr;
TaskHandle_t mqttTaskHandle = NULL;
static char remoteQuery[FILE_NAME_LEN * 2] = "";
static char lwt_topic[FILE_NAME_LEN / 2];
static char cmd_topic[FILE_NAME_LEN / 2];
static int mqttTaskDelay = 0;
static char mqttPublishTopic[FILE_NAME_LEN] = "";

void mqtt_client_publish(const char* topic, const char* payload){
  if (!mqtt_client || !mqttConnected) return;
```

```
  int id = esp_mqtt_client_publish(mqtt_client, topic, payload, strlen(payload), MQTT_QOS,
MQTT_RETAIN);
  LOG_DBG("Mqtt pub, topic:%s, ID:%d, length:%i", topic, id, strlen(payload));
  LOG_DBG("Mqtt pub, payload:%s", payload);
}

void mqttPublish(const char* payload) {
  if (!strlen(mqtt_topic_prefix)) return; //Called before load config?
  if (!strlen(mqttPublishTopic)) snprintf(mqttPublishTopic, FILE_NAME_LEN,
"%s%s/status", mqtt_topic_prefix, hostName);
  mqtt_client_publish(mqttPublishTopic, payload);
}

static void mqtt_connected_handler(void *handler_args, esp_event_base_t base, int32_t
event_id, void *event_data) {
  LOG_INF("Mqtt connected");
  esp_mqtt_client_publish(mqtt_client, lwt_topic, "online", 0, MQTT_LWT_QOS,
MQTT_LWT_RETAIN);
  mqttConnected = true;
}

static void mqtt_disconnected_handler(void *handler_args, esp_event_base_t base, int32_t
event_id, void *event_data) {
  LOG_INF("Mqtt disconnect");
  mqttConnected = false;
  xTaskNotifyGive(mqttTaskHandle);
}

static void mqtt_data_handler(void *handler_args, esp_event_base_t base, int32_t event_id,
void *event_data) {
```

```
  esp_mqtt_event_handle_t event = (esp_mqtt_event_handle_t)event_data;
  LOG_DBG("Mqtt topic=%.*s ", event->topic_len, event->topic);
  LOG_DBG("Mqtt data=%.*s ", event->data_len, event->data);
  if (strlen(remoteQuery) == 0) sprintf(remoteQuery, "%.*s", event->data_len, (char*)event->data);
  mqttConnected = true;
  LOG_DBG("Resuming mqtt thread..");
  xTaskNotifyGive(mqttTaskHandle);
}

static void mqtt_error_handler(void *handler_args, esp_event_base_t base, int32_t event_id, void *event_data) {
  LOG_DBG("Event base=%s, event_id=%d", base, event_id);
  esp_mqtt_event_handle_t event = (esp_mqtt_event_handle_t)event_data;
  LOG_DBG("Mqtt event error %i", event->msg_id);
  if (event->error_handle->error_type == MQTT_ERROR_TYPE_TCP_TRANSPORT) {

    LOG_ERR("Last err string (%s)", strerror(event->error_handle->esp_transport_sock_errno));
    mqttConnected = false;
  }
}
void checkForRemoteQuerry() {

  if (strlen(remoteQuery) > 0) {
    char* query = strtok(remoteQuery, ";");
    while (query != NULL) {
      char* value = strchr(query, '=');
      if (value != NULL) {
        *value = 0;
```

```
        value++;
        LOG_DBG("Mqtt exec q: %s v: %s", query, value);

        if (!strcmp(query, "restart"))
{
           doRestart("Mqtt remote restart");
        } else if (!strcmp(query, "clockUTC")) {

        } else {
#ifdef ISCAM

          if (!strcmp(query, "fps")) setFPS(atoi(value));
          else if (!strcmp(query, "framesize"))  setFPSlookup(fsizePtr);
#endif
          updateStatus(query, value);
        }
      } else { //No params command
        LOG_DBG("Execute cmd: %s", query);
        if (!strcmp(query, "status")) {
          buildJsonString(false);
          mqttPublish(jsonBuff);
        } else if (!strcmp(query, "status?q")) {
          buildJsonString(true);
          mqttPublish(jsonBuff);
        }
      }
      query = strtok(NULL, ";");
    }
    remoteQuery[0] = '\0';
  }
```

```
}

static void mqttTask(void* parameter) {
  LOG_DBG("Mqtt task start");
  while (mqtt_active) {

    ulTaskNotifyTake(pdTRUE, portMAX_DELAY);

    if (mqttConnected) {

      checkForRemoteQuerry();
      if (mqttTaskDelay > 0 ) vTaskDelay(mqttTaskDelay / portTICK_RATE_MS);
    } else {
      LOG_ERR("Disconnected wait..");
      vTaskDelay(2000 / portTICK_RATE_MS);
    }

  }
  mqttRunning = false;
  LOG_DBG("Mqtt Task exiting..");
  vTaskDelete(NULL);
}

void stopMqttClient() {
  if (mqtt_client == nullptr) return;
  if (mqttConnected){
    esp_mqtt_client_publish(mqtt_client, lwt_topic, "offline", 0, MQTT_LWT_QOS,
MQTT_LWT_RETAIN);
    vTaskDelay(1000 / portTICK_RATE_MS);
  }
```

```
  ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_disconnect(mqtt_client));
  ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_stop(mqtt_client));
  ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_destroy(mqtt_client));
  LOG_DBG("Checking task..%u", mqttTaskHandle);
  if ( mqttTaskHandle != NULL ) {
    LOG_DBG("Unlock task..");
    xTaskNotifyGive(mqttTaskHandle); //Unblock task
    vTaskDelay(1500 / portTICK_RATE_MS);
    LOG_DBG("Deleted task..?");
  }
  LOG_DBG("Exiting..");
  mqttConnected = false;
  mqtt_client = nullptr;
}

void startMqttClient(void){
  if (!mqtt_active) {
    LOG_DBG("MQTT not active..");
    return;
  }

  if (mqttConnected) {
    LOG_DBG("MQTT already running.. Exiting");
    return;
  }

  if (WiFi.status() != WL_CONNECTED) {
    mqttConnected = false;
    LOG_DBG("Wifi disconnected.. Retry mqtt on connect");
    return;
```

```
  }

  char mqtt_uri[FILE_NAME_LEN];
  sprintf(mqtt_uri, "mqtt://%s:%s", mqtt_broker, mqtt_port);

  esp_mqtt_client_config_t mqtt_cfg{.event_handle = NULL, .host = "", .uri = mqtt_uri,
.disable_auto_reconnect = false};
  mqtt_cfg.username = mqtt_user;
  mqtt_cfg.password = mqtt_user_Pass;
  mqtt_cfg.client_id = hostName;
  mqtt_cfg.lwt_qos = MQTT_LWT_QOS;
  mqtt_cfg.lwt_msg = "offline";
  mqtt_cfg.lwt_retain = MQTT_LWT_RETAIN;

  snprintf(lwt_topic, FILE_NAME_LEN, "%s%s/lwt", mqtt_topic_prefix, hostName);
  snprintf(cmd_topic, FILE_NAME_LEN, "%s%s/cmd", mqtt_topic_prefix, hostName);
  mqtt_cfg.lwt_topic = lwt_topic;

  mqtt_client = esp_mqtt_client_init(&mqtt_cfg);
  LOG_INF("Mqtt connect to %s...", mqtt_uri);
  if (mqtt_client != NULL) {

ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_register_event(mqtt_client,
esp_mqtt_event_id_t::MQTT_EVENT_CONNECTED, mqtt_connected_handler, NULL));

ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_register_event(mqtt_client,
esp_mqtt_event_id_t::MQTT_EVENT_DISCONNECTED, mqtt_disconnected_handler,
NULL));

ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_register_event(mqtt_client,
```

```
esp_mqtt_event_id_t::MQTT_EVENT_DATA, mqtt_data_handler, NULL));

ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_register_event(mqtt_client,
esp_mqtt_event_id_t::MQTT_EVENT_ERROR, mqtt_error_handler, NULL));
   if (ESP_ERROR_CHECK_WITHOUT_ABORT(esp_mqtt_client_start(mqtt_client)) !=
ESP_OK) {
     LOG_ERR("Mqtt start failed");
   } else {
     LOG_DBG("Mqtt started");
     int id = esp_mqtt_client_subscribe(mqtt_client, cmd_topic, 1);
     if (id == -1){
       LOG_ERR("Mqtt failed to subscribe: %s", cmd_topic );
       stopMqttClient();
       return;
     }
     else LOG_DBG("Mqtt subscribed: %s", cmd_topic );

     BaseType_t xReturned = xTaskCreate(&mqttTask, "mqttTask", MQTT_STACK_SIZE,
NULL, 1, &mqttTaskHandle);
     LOG_INF("Created mqtt task: %u", xReturned );
     mqttRunning = true;
   }
  }
}
```

ДОДАТОК Д

Лістинг користувацького веб-серверу

Лістинг Д.1 — Лістинг користувацького веб-серверу

```
#include "appGlobals.h"

#define MAX_PAYLOAD_LEN 1000

char inFileName[IN_FILE_NAME_LEN];
static char variable[FILE_NAME_LEN];
static char value[FILE_NAME_LEN];
static char retainAction[2];
int refreshVal = 5000;

static httpd_handle_t httpServer = NULL;
static int fdWs = -1;
static httpd_ws_frame_t wsPkt;
bool useHttps = false;
bool useSecure = false;

static fs::FS fp = STORAGE;
static byte* chunk;

esp_err_t sendChunks(File df, httpd_req_t *req, bool endChunking) {

  size_t chunksize = 0;
  while ((chunksize = df.read(chunk, CHUNKSIZE))) {
    if (httpd_resp_send_chunk(req, (char*)chunk, chunksize) != ESP_OK) break;

  }
  if (endChunking) {
```

```
    df.close();
    httpd_resp_sendstr_chunk(req, NULL);
  }
  if (chunksize) {
    LOG_ERR("Failed to send %s to browser", inFileName);
    httpd_resp_set_status(req, "500 Failed to send file");
    httpd_resp_sendstr(req, NULL);
  }
  return chunksize ? ESP_FAIL : ESP_OK;
}

esp_err_t fileHandler(httpd_req_t* req, bool download) {

  httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
  if (!strcmp(inFileName, LOG_FILE_PATH)) flush_log(false);
  File df = fp.open(inFileName);
  if (!df) {
    df.close();
    LOG_ERR("File does not exist or cannot be opened: %s", inFileName);
    httpd_resp_set_status(req, "404 File Not Found");
    httpd_resp_sendstr(req, NULL);
    return ESP_FAIL;
  }
  return (download) ? downloadFile(df, req) : sendChunks(df, req);
}

static void displayLog(httpd_req_t *req) {

  if (ramLog) {
    int startPtr, endPtr;
```

```
    startPtr = endPtr = mlogEnd;
    httpd_resp_set_type(req, "text/plain");

    do {
      int maxChunk = startPtr < endPtr ? endPtr - startPtr : RAM_LOG_LEN - startPtr;
      size_t chunkSize = std::min(CHUNKSIZE, maxChunk);
      if (chunkSize > 0) httpd_resp_send_chunk(req, messageLog + startPtr, chunkSize);
      startPtr += chunkSize;
      if (startPtr >= RAM_LOG_LEN) startPtr = 0;
    } while (startPtr != endPtr);
    httpd_resp_sendstr_chunk(req, NULL);
  } else {
    LOG_WRN("RAM Log not enabled");
    httpd_resp_sendstr(req, "400 RAM Log not enabled");
  }
}

static esp_err_t indexHandler(httpd_req_t* req) {
  strcpy(inFileName, INDEX_PAGE_PATH);

  if (strlen(startupFailure)) {
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_sendstr(req, startupFailure);
  }

  if (!fp.exists(INDEX_PAGE_PATH) && WiFi.status() != WL_CONNECTED) {

    httpd_resp_set_type(req, "text/html");
    return httpd_resp_sendstr(req, setupPage_html);
  } else {
```

```
    if (strlen(Auth_Name)) {

      size_t credLen = strlen(Auth_Name) + strlen(Auth_Pass) + 2; // +2 for colon &
terminator
      char credentials[credLen];
      snprintf(credentials, credLen, "%s:%s", Auth_Name, Auth_Pass);
      size_t authLen = httpd_req_get_hdr_value_len(req, "Authorization") + 1;
      if (authLen) {

        char auth[authLen];
        httpd_req_get_hdr_value_str(req, "Authorization", auth, authLen);
        if (!strstr(auth, encode64(credentials))) authLen = 0; // credentials not valid
      }
      if (!authLen) {

        httpd_resp_set_hdr(req, "WWW-Authenticate", "Basic");
        httpd_resp_set_status(req, "401 Unauthorised");
        return httpd_resp_sendstr(req, NULL);
      }
    }
  }
  return fileHandler(req);
}

esp_err_t extractHeaderVal(httpd_req_t *req, const char* variable, char* value) {

  esp_err_t res = ESP_FAIL;
  size_t hdrFieldLen = httpd_req_get_hdr_value_len(req, variable);
  if (!hdrFieldLen) LOG_WRN("Field %s not present", variable);
```

```
  else if (hdrFieldLen >= IN_FILE_NAME_LEN - 1) LOG_WRN("Field %s value too long
(%d)", variable, hdrFieldLen);
  else {
    res = httpd_req_get_hdr_value_str(req, variable, value, hdrFieldLen + 1);
    if (res != ESP_OK) LOG_ERR("Value for %s could not be retrieved: %s", variable,
espErrMsg(res));
  }
  return res;
}

esp_err_t extractQueryKeyVal(httpd_req_t *req, char* variable, char* value) {

  size_t queryLen = httpd_req_get_url_query_len(req) + 1;
  httpd_req_get_url_query_str(req, variable, queryLen);
  urlDecode(variable);
  char* endPtr = strchr(variable, '=');
  if (endPtr != NULL) {
    *endPtr = 0; // split variable into 2 strings, first is key name
    strcpy(value, variable + strlen(variable) + 1); // value is now second part of string
  } else {
    LOG_ERR("Invalid query string %s", variable);
    httpd_resp_set_status(req, "400 Invalid query string");
    httpd_resp_sendstr(req, NULL);
    return ESP_FAIL;
  }
  return ESP_OK;
}

static esp_err_t webHandler(httpd_req_t* req) {
```

```
size_t queryLen = httpd_req_get_url_query_len(req) + 1;
httpd_req_get_url_query_str(req, variable, queryLen);
urlDecode(variable);

if (!strcmp(variable, "OTA.htm")) {

  httpd_resp_set_type(req, "text/html");
  return httpd_resp_sendstr(req, otaPage_html);
} else if (!strcmp(HTML_EXT, variable+(strlen(variable)-strlen(HTML_EXT)))) {

  httpd_resp_set_type(req, "text/html");
} else if (!strcmp(JS_EXT, variable+(strlen(variable)-strlen(JS_EXT)))) {

  httpd_resp_set_type(req, "text/javascript");
  httpd_resp_set_hdr(req, "Cache-Control", "public, max-age=604800");
} else if (!strcmp(CSS_EXT, variable+(strlen(variable)-strlen(CSS_EXT)))) {

  httpd_resp_set_type(req, "text/css");
  httpd_resp_set_hdr(req, "Cache-Control", "max-age=604800");
} else if (!strcmp(TEXT_EXT, variable+(strlen(variable)-strlen(TEXT_EXT)))) {

  httpd_resp_set_type(req, "text/plain");
} else if (!strcmp(ICO_EXT, variable+(strlen(variable)-strlen(ICO_EXT)))) {

  httpd_resp_set_type(req, "image/x-icon");
} else if (!strcmp(SVG_EXT, variable+(strlen(variable)-strlen(SVG_EXT)))) {

  httpd_resp_set_type(req, "image/svg+xml");
} else LOG_WRN("Unknown file type %s", variable);
int dlen = snprintf(inFileName, IN_FILE_NAME_LEN - 1, "%s/%s", DATA_DIR,
```

```
variable);
  if (dlen >= IN_FILE_NAME_LEN) LOG_WRN("file name truncated");
  return fileHandler(req);
}

static esp_err_t controlHandler(httpd_req_t *req) {

  if (extractQueryKeyVal(req, variable, value) != ESP_OK) return ESP_FAIL;
  if (!strcmp(variable, "displayLog")) displayLog(req);
  else {
    strcpy(value, variable + strlen(variable) + 1); // value points to second part of string
    if (!strcmp(variable, "reset")) {
      httpd_resp_sendstr(req, NULL); // stop browser resending reset
      doRestart("user requested restart");
      return ESP_OK;
    }
    if (!strcmp(variable, "startOTA")) snprintf(inFileName, IN_FILE_NAME_LEN - 1,
"%s/%s", DATA_DIR, value);
    else {
      updateStatus(variable, value);
      appSpecificWebHandler(req, variable, value);
    }
  }
  httpd_resp_sendstr(req, NULL);
  return ESP_OK;
}

static esp_err_t statusHandler(httpd_req_t *req) {
  uint8_t filter = (uint8_t)httpd_req_get_url_query_len(req); // filter number is length of
query string
```

```
  buildJsonString(filter);
  httpd_resp_set_type(req, "application/json");
  httpd_resp_sendstr(req, jsonBuff);
  return ESP_OK;
}

bool parseJson(int rxSize) {

  jsonBuff[rxSize - 1] = ','; // replace final '}'
  jsonBuff[rxSize] = 0; // terminator
  char* ptr = jsonBuff + 1; // skip over initial '{'
  size_t itemLen = 0;
  bool retAction = false;
  do {

    char* endItem = strchr(ptr += itemLen, ':');
    itemLen = endItem - ptr;
    memcpy(variable, ptr, itemLen);
    variable[itemLen] = 0;
    removeChar(variable, "");
    ptr++;
    endItem = strchr(ptr += itemLen, ',');
    itemLen = endItem - ptr;
    memcpy(value, ptr, itemLen);
    value[itemLen] = 0;
    removeChar(value, "");
    ptr++;
    if (!strcmp(variable, "action")) {
      strcpy(retainAction, value);
      retAction = true;
```

```
    } else updateStatus(variable, value);
  } while (ptr + itemLen - jsonBuff < rxSize);
  return retAction;
}

static esp_err_t updateHandler(httpd_req_t *req) {

  size_t rxSize = min(req->content_len, (size_t)JSON_BUFF_LEN);
  int ret = 0;

  do {
    ret = httpd_req_recv(req, jsonBuff, rxSize);
    if (ret < 0) {
      if (ret == HTTPD_SOCK_ERR_TIMEOUT) continue;
      else {
        LOG_ERR("Update request failed with status %i", ret);
      }
    }
  } while (ret > 0);
  httpd_resp_sendstr(req, NULL);
  if (ret >= 0 && parseJson(rxSize)) appSpecificWebHandler(req, "action", retainAction);
  return ret < 0 ? ESP_FAIL : ESP_OK;
}

void progress(size_t prg, size_t sz) {
  static uint8_t pcProgress = 0;
  if (calcProgress(prg, sz, 5, pcProgress)) LOG_INF("OTA uploaded %d%%", pcProgress);
}

static esp_err_t uploadHandler(httpd_req_t *req) {
```

```
  esp_err_t res = appSpecificHeaderHandler(req);
if (res == ESP_OK) {
  size_t fileSize = req->content_len;
  size_t rxSize = min(fileSize, (size_t)JSON_BUFF_LEN);
  int bytesRead = -1;
  LOG_INF("Upload file %s", inFileName);

  if (strstr(inFileName, ".bin") != NULL) {

    LOG_INF("Firmware update using file %s", inFileName);
    OTAprereq();
    if (fdWs >= 0) httpd_sess_trigger_close(httpServer, fdWs);

    int cmd = (strstr(inFileName, "spiffs") != NULL) ? U_SPIFFS : U_FLASH;
    if (cmd == U_SPIFFS) STORAGE.end(); // close relevant file system
    if (Update.begin(UPDATE_SIZE_UNKNOWN, cmd)) {
      do {
        bytesRead = httpd_req_recv(req, jsonBuff, rxSize);
        if (bytesRead < 0) {
          if (bytesRead == HTTPD_SOCK_ERR_TIMEOUT) {
            delay(10);
            continue;
          } else {
            LOG_ERR("Upload request failed with status %i", bytesRead);
            break;
          }
        }
        Update.write((uint8_t*)jsonBuff, (size_t)bytesRead);
        Update.onProgress(progress);
        fileSize -= bytesRead;
```

```
      } while (bytesRead > 0);
      if (!fileSize) Update.end(true); // true to set the size to the current progress
    }
    if (Update.hasError()) LOG_ERR("OTA failed with error: %s", Update.errorString());
    else LOG_INF("OTA update complete for %s", cmd == U_FLASH ? "Sketch" :
"SPIFFS");
    httpd_resp_set_hdr(req, "Connection", "close");
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    httpd_resp_sendstr(req, Update.hasError() ? "OTA update failed, restarting ..." : "OTA
update complete, restarting ...");
    doRestart("Restart after OTA");

  } else {

    File uf = fp.open(inFileName, FILE_WRITE);
    if (!uf) LOG_ERR("Failed to open %s on storage", inFileName);
    else {

      do {
        bytesRead = httpd_req_recv(req, jsonBuff, rxSize);
        if (bytesRead < 0) {
          if (bytesRead == HTTPD_SOCK_ERR_TIMEOUT) {
            delay(10);
            continue;
          } else {
            LOG_ERR("Upload request failed with status %i", bytesRead);
            break;
          }
        }
        uf.write((const uint8_t*)jsonBuff, bytesRead);
```

```
        } while (bytesRead > 0);
        uf.close();
        res = bytesRead < 0 ? ESP_FAIL : ESP_OK;
        httpd_resp_sendstr(req, res == ESP_OK ? "Completed upload file" : "Failed to upload
file, retry");
        if (res == ESP_OK) LOG_INF("Uploaded file %s", inFileName);
        else LOG_ERR("Failed to upload file %s", inFileName);
      }
    }
  }
  return res;
}

static esp_err_t sendCrossOriginHeader(httpd_req_t *req) {

  httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
  httpd_resp_set_hdr(req, "Access-Control-Max-Age", "600");
  httpd_resp_set_hdr(req, "Access-Control-Allow-Methods",
"POST,GET,HEAD,OPTIONS");
  httpd_resp_set_hdr(req, "Access-Control-Allow-Headers", "*");
  httpd_resp_set_status(req, "204");
  httpd_resp_sendstr(req, NULL);
  return ESP_OK;
}

void wsAsyncSend(const char* wsData) {

  if (fdWs >= 0) {

    memset(&wsPkt, 0, sizeof(httpd_ws_frame_t));
```

```
    wsPkt.payload = (uint8_t*)(wsData);
    wsPkt.len = strlen(wsData);
    wsPkt.type = HTTPD_WS_TYPE_TEXT;
    wsPkt.final = true;
    esp_err_t ret = httpd_ws_send_frame_async(httpServer, fdWs, &wsPkt);
    if (ret != ESP_OK) LOG_ERR("websocket send failed with %s", esp_err_to_name(ret));
  }
}

static esp_err_t wsHandler(httpd_req_t *req) {

  if (req->method == HTTP_GET) {

    if (fdWs != -1) {
      if (fdWs != httpd_req_to_sockfd(req)) {

        LOG_WRN("closing connection, as newer Websocket on %u",
httpd_req_to_sockfd(req));
        httpd_sess_trigger_close(httpServer, fdWs);
      }
    }
    fdWs = httpd_req_to_sockfd(req);
    if (fdWs < 0) {
      LOG_ERR("failed to get socket number");
      return ESP_FAIL;
    }
    LOG_INF("Websocket connection: %d", fdWs);
  } else {

    uint8_t wsMsg[MAX_PAYLOAD_LEN];
```

```
    memset(&wsPkt, 0, sizeof(httpd_ws_frame_t));
    wsPkt.type = HTTPD_WS_TYPE_TEXT;
    wsPkt.payload = wsMsg;
    esp_err_t ret = httpd_ws_recv_frame(req, &wsPkt, MAX_PAYLOAD_LEN);
    if (ret != ESP_OK) {
      LOG_ERR("websocket receive failed with %s", esp_err_to_name(ret));
      return ret;
    }
    wsMsg[wsPkt.len] = 0;
    if (wsPkt.type == HTTPD_WS_TYPE_TEXT) appSpecificWsHandler((char*)wsMsg);
  }
  return ESP_OK;
}

void killWebSocket() {
  // user requested
  if (fdWs >= 0) {
    httpd_sess_trigger_close(httpServer, fdWs);
    fdWs = -1;
  }
}

void startWebServer() {
  esp_err_t res = ESP_FAIL;
  chunk = psramFound() ? (byte*)ps_malloc(CHUNKSIZE) : (byte*)malloc(CHUNKSIZE);
  size_t prvtkey_len = strlen(prvtkey_pem);
  size_t cacert_len = strlen(cacert_pem);
  if (useHttps && (!cacert_len || !prvtkey_len)) {
    useHttps = false;
    LOG_ALT("HTTPS not available as server keys not defined, using HTTP");
```

```
  }
  if (useHttps) {

    httpd_ssl_config_t config = HTTPD_SSL_CONFIG_DEFAULT();
#if CONFIG_IDF_TARGET_ESP32S3
    config.httpd.stack_size = SERVER_STACK_SIZE;
#endif
    config.cacert_pem = (const uint8_t*)cacert_pem;
    config.cacert_len = cacert_len + 1;
    config.prvtkey_pem = (const uint8_t*)prvtkey_pem;
    config.prvtkey_len = prvtkey_len + 1;
    config.httpd.server_port = HTTPS_PORT;
    config.httpd.ctrl_port = HTTPS_PORT;
    config.httpd.lru_purge_enable = true; // close least used socket
    config.httpd.max_uri_handlers = 10;
    config.httpd.max_open_sockets = HTTP_CLIENTS + MAX_STREAMS;
    res = httpd_ssl_start(&httpServer, &config);
  } else {

    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
#if CONFIG_IDF_TARGET_ESP32S3
    config.stack_size = SERVER_STACK_SIZE;
#endif
    config.server_port = HTTP_PORT;
    config.ctrl_port = HTTP_PORT;
    config.lru_purge_enable = true;
    config.max_uri_handlers = 10;
    config.max_open_sockets = HTTP_CLIENTS + MAX_STREAMS;
    res = httpd_start(&httpServer, &config);
  }
```

```
  httpd_uri_t indexUri = {.uri = "/", .method = HTTP_GET, .handler = indexHandler,
.user_ctx = NULL};
  httpd_uri_t webUri = {.uri = "/web", .method = HTTP_GET, .handler = webHandler,
.user_ctx = NULL};
  httpd_uri_t controlUri = {.uri = "/control", .method = HTTP_GET, .handler =
controlHandler, .user_ctx = NULL};
  httpd_uri_t updateUri = {.uri = "/update", .method = HTTP_POST, .handler =
updateHandler, .user_ctx = NULL};
  httpd_uri_t statusUri = {.uri = "/status", .method = HTTP_GET, .handler = statusHandler,
.user_ctx = NULL};
  httpd_uri_t wsUri = {.uri = "/ws", .method = HTTP_GET, .handler = wsHandler, .user_ctx
= NULL, .is_websocket = true};
  httpd_uri_t uploadUri = {.uri = "/upload", .method = HTTP_POST, .handler =
uploadHandler, .user_ctx = NULL};
  httpd_uri_t optionsUri = {.uri = "/upload", .method = HTTP_OPTIONS, .handler =
sendCrossOriginHeader, .user_ctx = NULL};
  httpd_uri_t sustainUri = {.uri = "/sustain", .method = HTTP_GET, .handler =
appSpecificSustainHandler, .user_ctx = NULL};
  httpd_uri_t checkUri = {.uri = "/sustain", .method = HTTP_HEAD, .handler =
appSpecificSustainHandler, .user_ctx = NULL};

  if (res == ESP_OK) {
    httpd_register_uri_handler(httpServer, &indexUri);
    httpd_register_uri_handler(httpServer, &webUri);
    httpd_register_uri_handler(httpServer, &controlUri);
    httpd_register_uri_handler(httpServer, &updateUri);
    httpd_register_uri_handler(httpServer, &statusUri);
    httpd_register_uri_handler(httpServer, &wsUri);
    httpd_register_uri_handler(httpServer, &uploadUri);
```

```
    httpd_register_uri_handler(httpServer, &optionsUri);
    httpd_register_uri_handler(httpServer, &sustainUri);
    httpd_register_uri_handler(httpServer, &checkUri);
    LOG_INF("Starting web server on port: %u", useHttps ? HTTPS_PORT : HTTP_PORT);
    LOG_INF("Remote server certificates %s checked", useSecure ? "are" : "not");
    if (DEBUG_MEM) {
      uint32_t freeStack = (uint32_t)uxTaskGetStackHighWaterMark(NULL);
      LOG_INF("Task httpServer stack space %u", freeStack);
    }
  } else LOG_ERR("Failed to start web server");

  debugMemory("startWebserver");
}
```

ДОДАТОК Е

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Мікрокомп'ютерна система оповіщення та контролю цілісності охоронного об'єкта засобами IoT

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 94,8% Схожість 5,2%

Аналіз звіту подібності (відмітити потрібне):

✓ Запозичення, виявлення у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Твердохліб Н. М.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Богомолов С.В.
(підпис) (прізвище, ініціали)