

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень


Програмно-технічний засіб вимірювання та моніторингу параметрів  
електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0  
Назва теми

КВРКІ 210116.21.01.44 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»  
Назва

Виконав: студент IV курсу, група КІ2-21-1  Денис МАЗУРИК  
Підпис Ініціали, прізвище

Керівник  Андрій ПИЧЕНОРУК  
Підпис, дата Ініціали, прізвище

Нормоконтролер  Тетяна КИСІЛЬ  
Підпис, дата Ініціали, прізвище

До захисту допускаю:  
зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

 Ольга ПАВЛОВА  
Підпис Ініціали, прізвище

«2» червня 2025 р.

Хмельницький 2025

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛЮВА



“ 10 ” 01 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Денису МАЗУРИКУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0

Керівник проекту (роботи) Андрій НЕЧИПОРУК, к.т.н., доцент.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Аналіз відомих програмно-технічних рішень вимірювання та моніторингу параметрів електроспоживання на основі IoT 6

Проектування програмно-технічного пристрою вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0

Програмна реалізація програмно-технічного пристрою вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Схема електрична принципова

Структура програмно-технічного засобу та загальний алгоритм

Монтажна схема проекту

6. Консультанти розділів дипломного проекту (роботи)

| Розділ        | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|---------------|---|----------------|------------------|
|               |   | завдання видав | завдання прийняв |
| Нормоконтроль | Тетяна КИСІЛЬ, доцент кафедри КПС         |                |                  |
| Антиплагіат   | Андрій ПІЧЕПОРУК, доцент кафедри КПС      |                |                  |

7. Дата видачі завдання « 10 » 01 2025 р.

**КАЛЕНДАРНИЙ ПЛАН**

| №з/п | Назва етапів (розділів) дипломного проекту (роботи)   | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1    | Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником                                 | 10.01.2025                               | виконано |
| 2    | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2025                               | виконано |
| 3    | Робота над розділом 1 – дослідження предметної області та постановка задачі   | 01.03.2025                               | виконано |
| 4    | Робота над розділом 2 – проектування системи моніторингу електроспоживання  | 01.04.2025                               | виконано |
| 5    | Робота над розділом 3 – програмна реалізація системи моніторингу електроспоживання                                    | 29.04.2025                               | виконано |
| 6    | Оформлення пояснювальної записки згідно вимог   | 22.05.2025                               | виконано |
| 7    | Попередній захист ВКР   | 23.05.2025                               | виконано |
| 8    | Захист ВКР на засіданні ЕК  | Червень 2025 року                        |          |

Студент

Підпис

Денис МАЗУРИК

Ініціали, прізвище

Керівник роботи

Підпис

Андрій ПІЧЕПОРУК

Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0».

Автор роботи: Денис МАЗУРИК.

Керівник роботи: Нічепорук Андрій Олександрович.

Пояснювальна записка: 74 с., 29 рис., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

ЛІЧИЛЬНИК, ПРОГРАМНО-ТЕХНІЧНИЙ ЗАСІБ, АРХІТЕКТУРА, МОНІТОРИНГ, БАЗА ДАНИХ.

Метою дипломної роботи є розробка та реалізація програмно-технічного засобу для вимірювання та моніторингу параметрів електроспоживання на основі технологій Інтернету речей (IoT) із використанням мікроконтролера ESP32 та платформи Blynk 2.0. Основна увага приділяється забезпеченню точного збору, обробки та візуалізації даних про споживання електроенергії в режимі реального часу.

Об'єктом дослідження є процес вимірювання та моніторингу електроспоживання за допомогою IoT-пристроїв.

Предметом дослідження є програмно-апаратні засоби та методи реалізації систем енергомоніторингу з використанням мікроконтролера ESP32 і платформи Blynk 2.0.

Під час виконання даного дослідження було використано метод системного аналізу для вивчення існуючих рішень у сфері IoT-моніторингу електроспоживання, а також експериментальний підхід для розробки та тестування власної реалізації системи.





  
Підпис студента

30.05.2025

Дата

## ЗМІСТ

|   |           |
|---|-----------|
| ВСТУП.....  | 4         |
| <b>1 АНАЛІЗ ВІДОМИХ ПРОГРАМНО-ТЕХНІЧНИХ РІШЕНЬ<br/>ВИМІРЮВАННЯ ТА МОНІТОРИНГУ ПАРАМЕТРІВ<br/>ЕЛЕКТРОСПОЖИВАННЯ НА ОСНОВІ ІОТ .....</b>                    | <b>6</b>  |
| 1.1 Концепція Інтернету речей.....  | 6         |
| 1.2 Огляд відомих рішень та засобів для вимірювання та моніторингу<br>параметрів електроспоживання на основі ІоТ .....                                    | 12        |
| 1.3 Висновки до першого розділу.....  | 18        |
| <b>2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ВИМІРЮВАННЯ<br/>ТА МОНІТОРИНГУ ЕЛЕКТРОСПОЖИВАННЯ НА ОСНОВІ ІОТ З<br/>ВИКОРИСТАННЯМ ПЛАТФОРМИ BLYNK 2.0 .....</b>   | <b>20</b> |
| 2.1 Формулювання вимог до програмно-технічного пристрою.....  | 20        |
| 2.2 Узагальнена структура програмно-технічного засобу .....   | 21        |
| 2.3 Схема електрична принципова.....  | 23        |
| 2.4 Монтажна схема проєктованого програмно-технічного пристрою.....   | 26        |
| 2.5 Концептуальна модель бази даних для збереження показників<br>енергоспоживання.....  | 27        |
| 2.6 Аналіз обраних рішень .....   | 29        |
| 2.6.1 Аналіз обраних апаратних рішень.....  | 29        |
| 2.6.2 Аналіз обраних програмних рішень.....   | 37        |
| 2.7 Перспективи розвитку системи .....  | 38        |
| 2.7.1 Використання хмарних баз даних(Firebase, InfluxDB) .....  | 38        |
| 2.7.2 Аналіз та візуалізація даних через графіки(Grafana, Power BI) .....   | 39        |
| 2.7.3 Розширення функціоналу на систему пристроїв .....   | 40        |
| 2.7.4 Підтримка декількох зон або пристроїв .....   | 40        |
| 2.8 Висновки до другого розділу.....  | 41        |
| <b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНО-ТЕХНІЧНОГО<br/>ПРИСТРОЮ ВИМІРЮВАННЯ ТА МОНІТОРИНГУ ПАРАМЕТРІВ<br/>ЕЛЕКТРОСПОЖИВАННЯ НА ОСНОВІ ІОТ З ВИКОРИСТАННЯМ</b> |           |

|  |      |                 |   |            |
|--|------|-----------------|---|------------|
| КвРКІ.210116.19.01.08 ПЗ   |      |                 |   |            |
| Зм.  | Арк. | Надокум.        | Підпис  | Дата       |
| Виконав  |      | Денис МАЗУРИК   |  | 20.08.2023 |
| Перевір.   |      | Андрій ПИЩЕГОРЬ |  | 20.08.2023 |
| Н.контр.   |      | Тетяна КИСІЛЬ   |  | 22.08.2023 |
| Затвер.  |      | Ольга ПАВЛОВА   |  | 21.08.2023 |
| Програмно-технічний засіб вимірювання та моніторингу параметрів електроспоживання на основі ІоТ з використанням ESP32 і платформи Blynk 2.0 Пояснювальна записка |      |                 |   |            |
|  |      |                 | Літера  | Аркуш      |
|  |      |                 | у   | 2          |
|  |      |                 | Аркушів   |            |
|  |      |                 | 79  |            |
| ХНУ КІ2-21-1   |      |                 |   |            |

|  |           |
|--|-----------|
| <b>ESP32 І ПЛАТФОРМИ BLYNK .....</b>                     | <b>41</b> |
| 3.1 Підготовка серверу та бази даних .....               | 41        |
| 3.2 Розгортання та налаштування платформи Blynk 2.0..... | 47        |
| 3.3 Програмна реалізація програми керування ESP32 .....  | 56        |
| 3.4. Висновки до третього розділу.....                   | 62        |
| <b>ВИСНОВКИ .....</b>                                    | <b>64</b> |
| <b>ДОДАТОК А.....</b>                                    | <b>71</b> |
| <b>ДОДАТОК Б .....</b>                                   | <b>72</b> |
| <b>ДОДАТОК В .....</b>                                   | <b>73</b> |
| <b>ДОДАТОК Г .....</b>                                   | <b>74</b> |
| <b>ДОДАТОК Д.....</b>                                    | <b>77</b> |

## ВСТУП

За останні роки автоматизація інформаційних технологій у значній мірі увійшла в повсякдення, зачепивши майже всі сфери діяльності людини, від простих справ до складних виробничих процесів і наукових розвідок. Її вплив помітний у транспорті, медицині, освіті, фінансах та багатьох інших галузях, де вона сприяє оптимізації процесів, збільшенню ефективності та розширенню горизонтів. Однією з основних сфер, де автоматизація демонструє надзвичайну ефективність, є енергоменеджмент, а точніше - контроль та облік споживання електроенергії. Ця сфера вимагає опрацювання великих обсягів інформації, точного вимірювання та оперативного реагування, що робить її ідеальною для впровадження автоматизованих систем.

Сучасні технології дозволяють не просто частково, а й повністю автоматизувати процес моніторингу енергоспоживання, від збору даних з лічильників до їх аналізу та представлення у зручному вигляді. Це передбачає встановлення інтелектуальних приладів обліку (смарт-лічильників), використання датчиків та контролерів, а також програмного забезпечення для управління та візуалізації даних. Такий підхід відкриває нові перспективи для економії ресурсів, як фінансових, так і природних, зменшення витрат на експлуатацію та оплату комунальних послуг, а також значного підвищення енергоефективності будівель і підприємств.

Не дивно, що системи обліку спожитої електроенергії, зокрема автоматизовані системи комерційного обліку (АСКОЕ) або локальне обладнання для збирання та обробки інформації (ЛУЗОД), стають все поширенішими в багатоквартирних та приватних оселях, офісних будівлях, торгових центрах, виробничих цехах та на інших об'єктах інфраструктури. Вони забезпечують можливість моніторингу споживання електроенергії по кожному вузлу обліку в режимі реального часу. Також ідентифікують незаконні підключення або помітні витоки, вивчають структуру споживання за різними сегментами, окремими

|     |      |          |        |      |                          |           |
|-----|------|----------|--------|------|--------------------------|-----------|
|     |      |          |        |      | КвРКІ.210116.21.01.44 ПЗ | Арк.<br>4 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |           |

приладами чи виробничими процесами, а також швидко реагують на надзвичайні ситуації, як-от короткі замикання або скачки напруги, уникаючи ушкодження техніки. Крім того, такі системи дають змогу впроваджувати гнучкі тарифні плани (наприклад, нічний тариф) та оптимізувати режими роботи енерговитратного обладнання.

У цьому контексті тематика автоматизованого вимірювання споживання електроенергії стає особливо актуальною на тлі глобальних проблем, зумовлених кліматичними змінами та потребою в ефективному використанні енергоресурсів. Провідні світові та українські компанії активно розробляють та вводять в дію сучасні рішення, що базуються на передових технологіях, таких як Інтернет речей (IoT), хмарні обчислення, штучний інтелект та машинне навчання для попереднього аналізу. Ці інновації можуть зробити контроль енергоспоживання не лише доступним та точним, але й максимально зручним та інтуїтивно зрозумілим для кінцевого споживача, надаючи йому інструменти для свідомого керування споживанням, сприяючи формуванню культури заощадження електроенергії.

|     |      |          |        |      |                          |      |
|-----|------|----------|--------|------|--------------------------|------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк. |
|     |      |          |        |      |                          | 5    |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |      |

# 1 АНАЛІЗ ВІДОМИХ ПРОГРАМНО-ТЕХНІЧНИХ РІШЕНЬ ВИМІРЮВАННЯ ТА МОНІТОРИНГУ ПАРАМЕТРІВ ЕЛЕКТРОСПОЖИВАННЯ НА ОСНОВІ ІОТ

## 1.1 Концепція Інтернету речей

Інтернет речей (Internet of Things, IoT) – це концепція, що описує взаємопов'язану мережу "речей", кожна з яких оснащена технологіями для взаємодії між собою та з навколишнім світом. Сенс цієї концепції полягає у формуванні інтелектуального середовища, де фізичні об'єкти можуть збирати, обробляти, обмінюватися даними та діяти на основі отриманої інформації.

На відміну від традиційних комп'ютерних мереж, що з'єднують переважно цифрові пристрої, IoT розширює можливості підключення до широкого спектру фізичних об'єктів від предметів повсякденного вжитку до промислового обладнання та інфраструктури. Ці об'єкти перетворюються з простих елементів на активних учасників цифрової екосистеми, дозволяючи збирати дані про їх стан та оточення в режимі реального часу.

Ключова ідея полягає у наданні фізичним об'єктам можливості бути "розпізнаваними" та "підключеними". Це реалізується шляхом інтегрування в них датчиків (для сприйняття), актуаторів (для дії), обчислювальних модулів (для обробки) та комунікаційних інтерфейсів (для зв'язку). Таким чином, "речі" можуть бути як джерелами інформації, так і об'єктами для керування.

Важливість IoT в технічному плані полягає у створенні розподілених систем, здатних автономно функціонувати, оптимізувати процеси та надавати дані для аналізу. Це відкриває нові перспективи для автоматизації, моніторингу, контролю та оптимізації в масштабах, що раніше були недоступні.

Будь-яка ефективна система Інтернету речей (IoT) ґрунтується на взаємодії кількох ключових технічних компонентів. Перш за все, це пристрої (things) – фізичні елементи, які є кінцевими точками в системі. Вони включають апаратні компоненти для взаємодії з фізичним і цифровим середовищем, а саме: сенсорні

|     |      |          |        |      |                          |      |
|-----|------|----------|--------|------|--------------------------|------|
|     |      |          |        |      | КвРКІ.210116.21.01.44 ПЗ | Арк. |
|     |      |          |        |      |                          | 6    |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |      |

модулі, що складаються з різноманітних датчиків (температури, вологості, тиску, світла, руху, газу, зображення тощо) для перетворення фізичних показників на сигнали; виконавчі пристрої (актуатори), такі як двигуни, реле, клапани чи дисплеї, що виконують дії у фізичному світі у відповідь на команди; обчислювальні модулі (мікроконтролери, мікропроцесори) для локальної обробки даних та управління; модулі зв'язку (радіомодулі, мережеві карти) для налагодження з'єднання та передачі даних. Вибір технологій зв'язку залежить від потреб у дальності дії, пропускній здатності, енергоспоживанні, топології мережі та кількості пристроїв [2].

Ключові технології зв'язку для IoT підбираються, виходячи з потреб у дальності, швидкості передачі даних та енергоефективності. Для невеликих дистанцій та з низьким енергоспоживанням часто використовуються персональні мережі PAN (Bluetooth/BLE, Zigbee, Z-Wave), що ідеально підходять для носіїв та систем розумного будинку. Для локальних мереж, де потрібна більша пропускна здатність, застосовуються LAN (Wi-Fi, Ethernet), що є зручним для будівель, але потребують більше енергії або дротового з'єднання. На великі відстані та з мінімальним енергоспоживанням для передачі невеликих обсягів даних орієнтовані LPWAN (LoRaWAN, Sigfox, NB-IoT, LTE-M), які є основою для функціонування розумних міст, сільськогосподарського сектору та лічильників. Широке та глобальне покриття забезпечують WAN (стільниковий зв'язок 2G-5G, супутниковий зв'язок), що необхідно для мобільних пристроїв та віддалених місцевостей. Взаємодію між цими мережами забезпечують протоколи зв'язку, включаючи базові TCP/IP та UDP, прикладні HTTP та його полегшений варіант CoAP, а також популярні для телеметрії протоколи обміну повідомленнями, на кшталт MQTT.

Шлюзи IoT (IoT Gateways) – це проміжні пристрої, які збирають дані з групи пристроїв IoT, виконують їх попередню обробку (фільтрація, нормалізація), перетворюють протоколи зв'язку та гарантують безпечне підключення до більш високорівневих мереж або хмарних платформ. Вони можуть виконувати функції

|     |      |          |        |      |                          |           |
|-----|------|----------|--------|------|--------------------------|-----------|
|     |      |          |        |      | КвРКІ.210116.21.01.44 ПЗ | Арк.<br>7 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |           |

граничних обчислень (Edge Computing), зменшуючи затримку та навантаження на центральну систему, обробляючи критичні дані локально. Шлюзи є важливим рівнем абстракції між різноманітними пристроями та основною інфраструктурою.

Платформи IoT - це комплекси програмно-апаратних рішень, котрі надають інфраструктуру та інструменти для управління життєвим циклом пристроїв (під'єднання, автентифікація, оновлення), збору та зберігання даних, виконання аналітики, візуалізації даних та інтеграції з іншими системами. Платформи можуть бути хмарними, локальними або гібридними, пропонуючи масштабованість та гнучкість для різних сценаріїв розгортання IoT, а також централізоване управління та моніторинг.

Для обробки та аналізу даних використовуються механізми для перетворення сирих даних з пристроїв у цінну інформацію. Це передбачає зберігання даних за рахунок бази даних (реляційних, NoSQL, часових рядів), сховищ даних (data lakes), оптимізованих для великих обсягів різнорідних даних, що надходять від IoT-пристроїв. Також це вимагає потокової обробки, що має на меті аналіз даних "на льоту" в міру їх надходження (наприклад, з використанням Kafka, Spark Streaming) для реагування в реальному часі на події та аномалії. Крім того, використовується технологія пакетної обробки накопичених даних для глибшого аналізу, формування звітів та виявлення довгострокових тенденцій. Для аналітики використовується статистичний аналіз, машинне навчання (ML), штучний інтелект (AI) для виявлення закономірностей, аномалій, прогнозування поведінки системи або пристроїв, що перетворює дані на корисні інсайти.

Програмне забезпечення та додатки функціонують на різних рівнях системи від вбудованого ПЗ на пристроях до мобільних додатків та корпоративних інформаційних систем. Вони реалізують бізнес-логіку, надають інтерфейси користувача та інтегрують IoT-функціональність в інші системи, забезпечуючи взаємодію користувачів із системою та автоматизацію процесів на основі даних з IoT [4].

|     |      |          |        |      |                          |           |
|-----|------|----------|--------|------|--------------------------|-----------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>8 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |           |

Типова технічна архітектура системи IoT здебільшого описується як багатошарова модель, що забезпечує структурований підхід до проектування й реалізації складних систем. З технічного погляду можна виділити такі основні компоненти: шар пристроїв, мережевий шар, шар платформи, шар застосунків, а також, у деяких випадках, шар аналітики та наскрізний шар безпеки.

Розглянемо детальніше шар пристроїв, також відомий як периферійний або сенсорний. Саме тут відбувається безпосередня взаємодія з фізичним середовищем. Він включає всі апаратні складові IoT-системи, зокрема мікроконтролери, сенсори, виконавчі пристрої та модулі зв'язку. Технічна особливість цього шару полягає в необхідності оптимального вибору елементної бази з огляду на споживання енергії, розміри, вартість та умови експлуатації. На цьому етапі також розробляється вбудоване програмне забезпечення для збору та початкової обробки даних, керування пристроями й налагодження їх комунікації. Успішна реалізація взаємодії між компонентами досягається за допомогою внутрішніх протоколів, як-от SPI або I2C, а також зовнішніх протоколів зв'язку низького рівня. Забезпечення безпеки на цьому рівні здійснюється через механізми захищеного завантаження, локальне шифрування й надійну аутентифікацію пристроїв у мережі.

Наступним проаналізуємо мережевий шар. Його роль полягає в забезпеченні надійної й безпечної передачі даних від пристроїв до центральної платформи, а також у зворотному напрямку для управління. Технічно цей шар охоплює вибір відповідної мережевої технології, зокрема Wi-Fi, Bluetooth, стільникові мережі або LPWAN, що визначається вимогами до енергоспоживання, дальності дії та пропускної здатності. Важливою також є побудова відповідної топології мережі: зірки, дерева або mesh. На цьому рівні активно використовуються мережеві протоколи IP (IPv4, IPv6), транспортні протоколи TCP та UDP, а також полегшені протоколи прикладного рівня, як-от MQTT, CoAP або HTTP, що спеціально адаптовані до умов IoT. Захист комунікацій забезпечується через VPN, шифрування TLS/SSL, фаєрволи та системи виявлення загроз.

|     |      |          |        |      |                          |           |
|-----|------|----------|--------|------|--------------------------|-----------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>9 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |           |

Тепер перейдемо до третього рівня, що представляє собою шар платформи, котрий інколи називають проміжним або рівнем обробки сервісів. Він є основою для централізованого управління всією IoT-системою та обробки отриманих даних. На цьому рівні зосереджено керування життєвим циклом пристроїв, первинне зберігання та фільтрація інформації, а також інтеграція з аналітичними або зовнішніми системами. Саме тут реалізується логіка обробки потоків даних, моніторинг, масштабування та інші платформи як сервіси (PaaS), що підтримують ефективне функціонування всієї екосистеми.

Четвертим шаром виступає шар застосунків. Його функція полягає в наданні кінцевим користувачам або зовнішнім системам зручного інтерфейсу для доступу до функціональності IoT. Технічно він охоплює розробку мобільних, веб або десктопних застосунків, які взаємодіють з платформою через API, візуалізують дані у вигляді панелей приладів, діаграм або звітів, а також підтримують інтеграцію з корпоративними системами, такими як ERP, CRM або MES. Важливою частиною цього рівня є реалізація бізнес-логіки, зокрема автоматизації процесів на основі подій або аналітики, отриманої з нижчих рівнів.

Окремі архітектурні моделі виокремлюють шар аналітики або бізнес-логіки, що розміщується між платформою та застосунками. Його мета полягає у глибокому аналізі зібраних даних, використанні машинного навчання, предиктивного моделювання або алгоритмів прийняття рішень, що дає змогу реалізовувати складніші сценарії автоматизації.

Слід також зауважити, що шар безпеки розглядається як наскрізний, тобто такий, що охоплює всі рівні системи. Його задача полягає в гарантії цілісності, конфіденційності та доступності даних на кожному етапі від фізичних пристроїв до кінцевих застосунків. Реалізація безпечного IoT передбачає комплексну роботу з шифруванням, аутентифікацією, керуванням правами доступу, аудитом подій та відповідністю вимогам стандартів безпеки [5].

Технічний процес роботи IoT-системи являє собою структурований ланцюжок операцій, де кожен крок виконує конкретну функцію у загальній

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>10 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

взаємодії пристроїв, даних та користувача. Все починається зі збору інформації, коли фізичне середовище фіксується за допомогою датчиків. Це можуть бути показники температури, руху, тиску, вологості або інші параметри. Дані, отримані від сенсорів, є сирими, тому виникає потреба в попередній обробці безпосередньо на місці. Вбудовані мікроконтролери або edge-процесори на пристроях здатні виконувати фільтрацію, агрегацію або елементарний аналіз, наприклад, визначати порогові значення. Такий підхід зменшує навантаження на мережу та забезпечує швидке реагування без необхідності постійного звернення до віддаленого сервера.

Після локальної обробки настає фаза передачі даних, коли сформовані пакети інформації надсилаються через модуль зв'язку. Вибір протоколів і технологій зв'язку залежить відстані, енергоспоживання та частоти відправки даних. Інформація може передаватися напряму до хмари або через шлюз, який забезпечує додаткову обробку, шифрування або буферизацію даних.

Коли дані досягають центральної платформи, відбувається їхнє отримання та інжест. Це включає в себе автентифікацію джерела, дешифрування, перевірку формату, а також, при необхідності, повторну передачу у випадку втрати. Потім інформація переходить на етап централізованої обробки, де застосовуються алгоритми нормалізації, видалення шуму, конвертації форматів і збагачення даних. Оброблені дані зберігаються у спеціалізованих базах даних, часто орієнтованих на часові ряди, що дозволяють оперативно проводити аналіз в реальному часі. Потокова обробка забезпечує миттєве реагування на події, а пакетна обробка використовується для більш глибокого аналізу, навчання моделей та прогнозування.

Наступний етап включає аналітику та прийняття рішень. У цьому процесі використовуються інструменти машинного навчання або систем штучного інтелекту, які можуть виявляти шаблони, аномалії чи передбачати поведінку об'єктів. В результаті отримується інформація, на основі якої система або оператор може здійснити певну дію. Наприклад, виявивши аномальне підвищення

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>11 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

температури, система може надіслати команду на активацію охолоджувального пристрою.

Це приводить до фази виконання дій. Команди, сформовані на основі аналітики або вручну ініційовані користувачем, передаються назад до пристрою. Механізм актуатора, наприклад, реле або мотор, виконує відповідну фізичну дію. Ці команди використовують ту саму інфраструктуру зв'язку, що й дані, забезпечуючи повний цикл зворотного зв'язку.

Завершальним елементом цього потоку є візуалізація та взаємодія з користувачем. Через мобільний додаток, веб-панель або інші інтерфейси надається доступ до візуалізованих даних, графіків, статусу пристроїв та налаштувань. Інтерфейс може бути як пасивним (для моніторингу), так і інтерактивним з можливістю вручну запускати процеси, змінювати параметри або отримувати повідомлення у разі подій. Разом це утворює замкнений та безперервний цикл, де технічні компоненти та логіка взаємодіють в режимі реального часу [7].

## 1.2 Огляд відомих рішень та засобів для вимірювання та моніторингу параметрів електроспоживання на основі IoT

Електричні лічильники – це пристрої, що функціонують на основі різноманітних технологій (механічних, електронних, інтелектуальних) і призначені для вимірювання обсягу спожитої електричної енергії. Принцип їх роботи охоплює кілька ключових етапів: спочатку відбувається вимірювання потоку електричної енергії, що проходить через проводи, на основі різних принципів (електромагнітних, індукційних або електронних). Далі лічильник перетворює виміряні параметри електричного струму (напруга, сила струму, потужність) в кількісне значення енергії, що витрачається за певний проміжок часу. Нарешті, здійснюється облік енергії – лічильник зберігає дані про спожиту енергію, які потім можуть передаватися для подальшої обробки або відображатися на дисплеї.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>12 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

На даний момент відбувається активне впровадження інтелектуальних (розумних) лічильників. Вони є найсучаснішим типом і відрізняються можливістю автоматизованого дистанційного зчитування та передачі даних, а також здатністю інтегруватися в системи "Розумного будинку". Розумні лічильники надають розширений функціонал, включаючи моніторинг споживання електроенергії в реальному часі, автоматичну передачу даних постачальнику енергії для точного нарахування оплат, а також можливість надання детальної інформації про енергоспоживання користувачеві через мобільні додатки або веб-інтерфейси.

Ключові функції електричних лічильників (особливо інтелектуальних) включають у себе: облік спожитої енергії, показники споживання в реальному часі, дистанційне зчитування та передача даних, а також можливість підключення до інших систем. Розглянемо ці функції детальніше. Облік спожитої енергії є основною функцією, яка полягає у точному фіксуванні кількості спожитої електричної енергії в кВт·год для цілей розрахунків між споживачем та постачальником. Для інтелектуальних лічильників додається можливість надавати показники споживання в реальному часі, що дає користувачам змогу миттєво бачити своє поточне енергоспоживання. Важливою сучасною функцією є дистанційне зчитування та передача даних, що дозволяє автоматизовано надсилати зібрану інформацію про споживання до систем постачальника енергії або інших платформ без необхідності фізичного візиту інспектора(рис.1.1). Крім того, вони можливість підключення до інших систем, інтегруючись у розумну інфраструктуру будинку або підприємства (такі як системи управління енергоспоживанням або "Розумний будинок") для оптимізації використання енергії та зниження витрат.

Переваги інтелектуальних лічильників: можливість інтеграції з автоматизованими системами обліку та управління, а також зниження потреби в регулярних візитах інспекторів для зняття показників.

Недоліки інтелектуальних лічильників: їх вища початкова вартість порівняно з традиційними лічильниками та необхідність спеціалізованого обладнання та інфраструктури для їх зчитування та функціонування.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>13 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

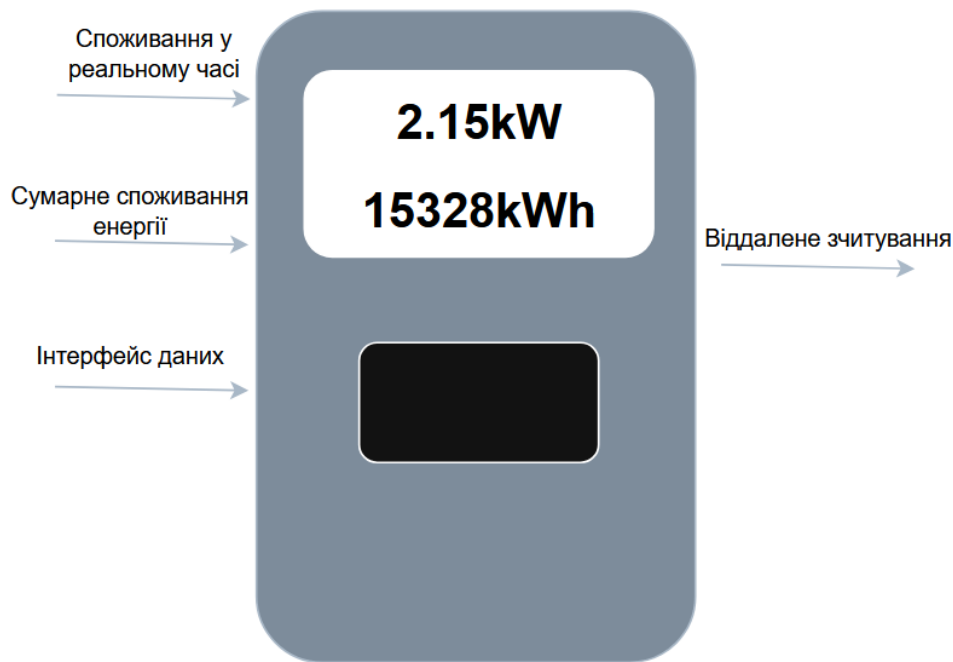


Рисунок 1.1 – Схематичне зображення електронного лічильника

На сьогодні відзначається тенденція до використання багатофункціональних електронних лічильників з дистанційною передачею даних, що забезпечують не лише точний облік ресурсів, а й можливість моніторингу споживання в реальному часі, виявлення витоків та інтеграції з системами автоматизованого управління будинком або промисловим об'єктом. Найбільш популярні лічильники на ринку наочно демонструють вищеописані тенденції.

### 1. Shelly EM

Це лічильник енергії Wi-Fi від Shelly, призначений для моніторингу двох окремих каналів споживання струму за допомогою струмових трансформаторів (рис.1.2). До того ж, він здатний підраховувати загальне енергоспоживання. Shelly EM легко інтегрується з різними системами автоматизації будинку, включно з Home Assistant, MQTT, Domoticz, а також надає підтримку як для локального доступу (через HTTP API), так і для хмарного (за допомогою Shelly Cloud). Серед переваг можна виділити його компактність, легкість монтажу в електричному щитку, наявність локального API, адаптивну інтеграцію з іншими системами. До недоліків можна віднести відсутність фізичного дисплею, необхідність

використання якісних струмових трансформаторів для точної роботи, доступність хмарних функцій лише після реєстрації. Крім того, Shelly EM здатний керувати контакторами з навантаженням до 2А, має вбудований захист від перегрівання, підтримує функцію тижневого планування та має робочу температуру в межах від -20°C до 40°C [9].



Рисунок 1.2 – Лічильник Shelly EM

## 2. Sonoff POW Elite (20A/16A)

Це Wi-Fi смарт-реле з функцією вимірювання електроспоживання. Має вбудований LCD-дисплей, який показує поточні значення напруги, струму, потужності, загального споживання (рис.1.3). Працює через хмару eWeLink, але підтримує перепрошивку (наприклад, на Tasmota) для локальної інтеграції. Також сумісний з Google Assistant та Alexa. З плюсів цієї моделі можна виділити просте в користуванні рішення з дисплеєм, віддалене керування живленням, ведення статистики, можливість інтеграції з автоматизаціями. На противагу це лічильник має обмеження по струму (максимум 20 ампер), без перепрошивки залежить від хмари, не підтримує багатофазну мережу. Також пристрій має доступні версії на 16А та 20А, монтується на DIN-рейку та використовує Wi-Fi 2.4 ГГц [10].

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>15 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |



Рисунок 1.3 – Wi-Fi смарт-реле з функцією вимірювання електроспоживання Sonoff POW Elite (20A/16A)

### 3. Iammeter WEM3080T (3-фазний)

Цей пристрій розроблено для вимірювання споживання електричної енергії у трифазній електромережі(рис.1.5). Вбудований Wi-Fi забезпечує передавання даних у хмарне сховище Iammeter, а також дає змогу інтегрувати прилад з локальними системами за допомогою MQTT, Modbus TCP, Home Assistant або HTTP API. Працює як з однофазною, так і з трифазною мережею. Підтримує напругу  $80V \sim 277VAC$  (фазна) /  $140V \sim 480 VAC$  (лінійна). Доступні додаткові трансформатори струму на 150A, 250A або 500A. Точність вимірювання напруги, струму та активної потужності становить  $\pm 1.0\%$ . Споживає менше 2W при напрузі 230VAC. Найвизначнішими рисами, що можуть привернути увагу для використання є підтримка трьох фаз, локальний збір даних, наявність веб-інтерфейсу, інтерфейс для відображення графіків та історичних даних, якісна підтримка. З іншого боку даний комплект має вищу ціну й великі габарити, потребує встановлення трансформаторів струму (вони входять до комплекту), необхідно монтувати у захищеному місці [11].



Рисунок 1.4 – Iammeter WEM3080T (3-фазний)

#### 4. ZMAi-90

Це Wi-Fi енергометр, призначений для встановлення на DIN-рейку в електричному щитку. Він оснащений РК-дисплеєм для безпосереднього відображення інформації, а також забезпечує інтеграцію з хмарним сервісом TuYa Smart, що базується в Китаї. В залежності від модифікації, існує можливість перепрошивки альтернативним програмним забезпеченням (наприклад, ESPHome або Tasmota). Передає інформацію про напругу, силу струму, потужність та споживання електроенергії. Переваги: невеликий розмір, наявність екрану, монтаж як стандартний автоматичний вимикач, інтеграція з мобільними програмами, гнучкість перепрошивки для локального використання. Недоліки: залежність від хмари TuYa, обмежений доступ до локального API без перепрошивки, певні труднощі з інтеграцією в локальні мережі без китайського застосунку. Додаткові відомості: зазвичай адаптовано для однофазних мереж з напругою до 250 В та струмом до 60 А. У деяких моделях застосовується чіп ESP8266 [12].

|     |      |          |        |      |  |  |  |  |  |                          |            |
|-----|------|----------|--------|------|--|--|--|--|--|--------------------------|------------|
|     |      |          |        |      |  |  |  |  |  | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>17 |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |                          |            |



Рисунок 1.6 – ZMAi-90 Wi-Fi енергометр

### 1.3 Висновки до першого розділу

У першому розділі було представлено концепцію Інтернету речей (IoT) як ключової технології для побудови сучасних систем моніторингу енергоспоживання. Здійснено аналіз наявних програмно-технічних рішень, які ілюструють широкий потенціал IoT для автоматизованого збирання, транспортування та аналізу даних у режимі реального часу. Окрему увагу зосереджено на мікроконтролері ESP32, який завдяки своїм технічним характеристикам (вбудований Wi-Fi/Bluetooth, висока продуктивність, підтримка різних периферійних пристроїв) виглядає оптимальним рішенням для створення систем енергомоніторингу. Платформа Blynk 2.0, в свою чергу, пропонує зручний інструментарій для візуалізації даних, дистанційного доступу та інтеграції з хмарними сервісами, що робить її ефективним рішенням для реалізації інтерфейсної складової проекту. Отже, на основі проведеного дослідження сформульовано загальну задачу розробки програмно-технічного рішення для вимірювання та моніторингу показників споживання електроенергії з використанням мікроконтролера ESP32 та платформи Blynk 2.0. Таке рішення має гарантувати високоточні вимірювання ключових електротехнічних параметрів, автоматичний збір і передачу даних, а також ефективну візуалізацію та обробку результатів. Це дозволить оптимізувати витрати споживачів і покращити ефективність управління енергоресурсами.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>18 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

## 2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ВИМІРЮВАННЯ ТА МОНІТОРИНГУ ЕЛЕКТРОСПОЖИВАННЯ НА ОСНОВІ ІОТ З ВИКОРИСТАННЯМ ПЛАТФОРМИ VLYNK 2.0

### 2.1 Формулювання вимог до програмно-технічного пристрою

Розробка системних вимог є важливим етапом створення будь-якого програмного або апаратного засобу, в тому числі засобу для вимірювання та моніторингу параметрів енергоспоживання. Поставлення вимог допомагає зрозуміти, якій функціональності повинна відповідати дана система. Це все виконується з метою задоволення потреб та очікувань користувачів. Точний список поставлених вимог на початковій стадії проекту запобігає непотрібній розробці та переробці системи на більш пізніх стадіях процесу розробки, що призводить до марної трати часу та ресурсів.

Програмно-апаратні комплекси моніторингу електроенергії мають фіксувати ключові електричні характеристики: напругу, силу струму, потужність та сукупне енергоспоживання, виражене в кіловат-годинах. Система повинна автоматично зчитувати дані, використовуючи відповідні сенсори струму та напруги, та відправляти їх через мікроконтролер ESP32 за допомогою Wi-Fi з'єднання до хмарного сервісу, наприклад, до програми Vlynk 2.0. Користувачу потрібно надати можливість дистанційного перегляду показників в режимі реального часу з будь-якої локації за допомогою мобільного додатку. Щоб уникнути втрати даних в разі відключення електроенергії, система повинна використовувати EEPROM ESP32 для зберігання останніх отриманих значень. Зафіксовані дані також мають бути збережені в базі даних для подальшого аналізу та порівняльного дослідження. Важливим є забезпечення енергоефективності, що включає в себе низьке споживання енергії самим пристроєм та здатність до тривалої роботи від батареї. Інтерфейс користувача мусить бути інтуїтивно зрозумілим та зручним у використанні, що сприятиме легкому налаштуванню та

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>19 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

керуванню системою. Паралельно, вартість обладнання та компонентів повинна бути мінімальною для забезпечення доступності реалізації.

Тому вимоги до програмно-апаратного засобу вимірювання та моніторингу параметрів енергоспоживання на основі IoT диктують функції, які повинен виконувати розроблений пристрій.

## 2.2 Узагальнена структура програмно-технічного засобу

Першим кроком у проектуванні пристрою є визначення компонентів пристрою, а також розгляд та узагальненої структури програмно-апаратного засобу для вимірювання та моніторингу параметрів енергоспоживання на основі Інтернету речей.

У проектованому програмно-апаратному засобі використовується мікроконтролер ESP32 WiFi Module для збору та обробки інформації з датчиків для збору даних при споживанні пристроєм електроенергії, таких як струму або напруги, а також застосунок Blink 2.0 для отримання всіх необхідних даних про моніторинг споживання електроживлення в реальному часі з будь-якої точки світу, де є інтернет живлення. Дана система надає можливість забезпечити дистанційний моніторинг та контроль за показниками електроживлення через смартфон або девайс який має доступ до Blink. Схему, яка узагальнює програмно-технічний засіб вимірювання та моніторингу параметрів електроспоживання на основі IoT наведено на рисунку 2.1. Проведемо розгляд та аналіз компонентів, які входять до складу програмно-апаратного засобу.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КвРКІ.210116.21.01.44 ПЗ | Арк.<br>20 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

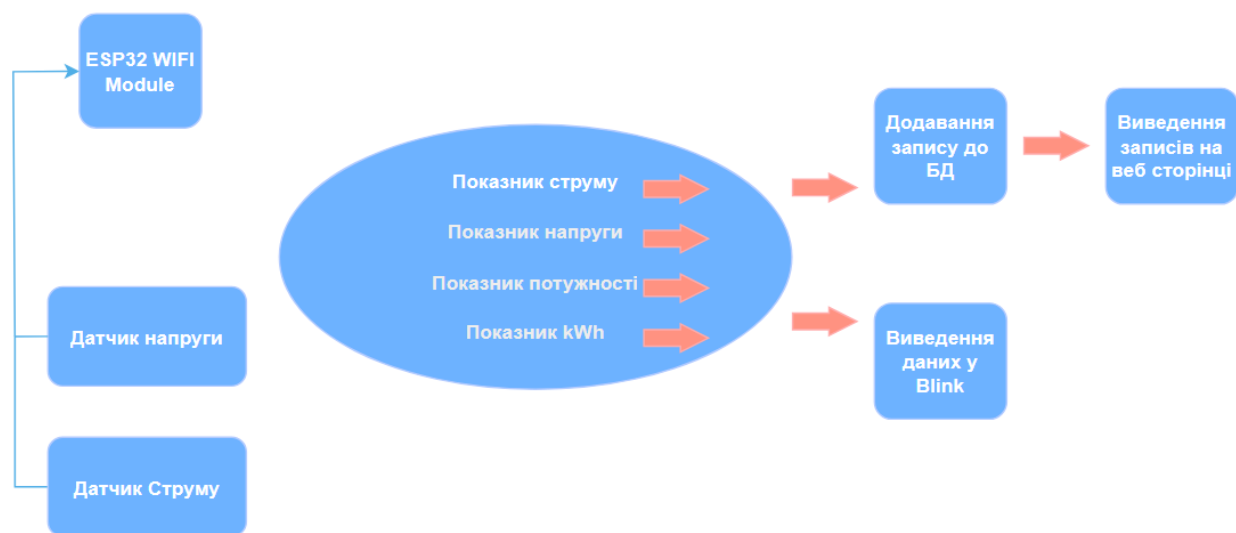


Рисунок 2.1 – Узагальнена схема пристрою моніторингу електроспоживання

Мікроконтролерна система ESP32 є основним компонентом для вимірювання електроспоживання, а також для передачі та отримання даних через Blynk 2.0. Вибір даної мікроконтролерної системи обумовлений її потужними обчислювальними можливостями, вбудованим Wi-Fi модулем, а також енергоефективністю, що робить її ідеальним рішенням для проекту енергомоніторингу.

Під час проектування програмно-технічного засобу IoT Smart Electricity Energy Meter були визначені його головні функціональні можливості, серед котрих вимірювання напруги, струму, потужності та загального енергоспоживання в кіловат-годинах, а також передавання цих даних у реальному часі до хмарного сервісу Blynk 2.0. Одержані показники можуть відображатися як у веб, так і в мобільному додатку Blynk задля зручного моніторингу. Задля забезпечення збереження даних при збої електропостачання застосовується вбудована EEPROM-пам'ять мікроконтролера ESP32. Крім того, передбачена змога використання додаткового дисплея LCD 20×4 з інтерфейсом I2C для локального відображення вимірюваних параметрів. До складу основних апаратних компонентів входить мікроконтролер ESP32, що відповідає за

вимірювання та передачу даних, датчик струму SCT-013 для фіксації сили струму, датчик напруги ZMPT101B для вимірювання рівня напруги, EEPROM для збереження енергетичних даних, опціональний LCD-дисплей для візуалізації показників, а також резистори (10К, 100 Ом) і конденсатор (10uF), котрі формують точне вимірювальне коло.

Blynk 2.0 є основним інтерфейсом користувача для моніторингу енергоспоживання. Через Blynk-додаток користувач може отримувати поточні дані про використання електроенергії, налаштовувати пороги потужності та отримувати попередження про перевищення встановлених меж. Взаємодія між Blynk 2.0 та ESP32 забезпечує миттєвий доступ до інформації з будь-якої точки світу, що робить систему зручною та ефективною.

Таким чином, створена система IoT Smart Electricity Energy Meter дозволяє автоматизувати процес моніторингу електроспоживання, підвищуючи ефективність використання електроенергії, знижуючи витрати та забезпечуючи надійний контроль у режимі реального часу.

### 2.3 Схема електрична принципова

Суть поняття схеми електричної принципової лежить у тому, що це креслення, яке демонструє найважливіші компоненти, які складають нашу систему та схематично показують наявні з'єднання між ними за рахунок схематичних позначень та з'єднань. Вона не показує конкретні компоненти та їх розташування, але допомагає зрозуміти принципи та функції системи. Це важливий інструмент для розробників при проектуванні електричних систем, що дозволяє їм ідентифікувати компоненти та з'єднання між компонентами.

У даному проєкті реалізовано розумний лічильник електроенергії IoT з використанням мікроконтролера ESP32 та Blynk 2.0. Основною метою є автоматизація моніторингу споживання електроенергії, що дозволяє зробити цей процес зручнішим та ефективнішим.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>22 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

Для вимірювання змінного струму використовується безконтактний струмовий датчик SCT-013, який є трансформатором струму. Він дозволяє проводити вимірювання без прямого підключення до високовольтних ланцюгів, що робить його безпечним та простим у використанні. Конструкція датчика включає первинну обмотку, магнітне осердя та вторинну обмотку з великою кількістю витків.

Традиційні методи контролю електроспоживання передбачають ручне зняття показників, що є трудомістким та незручним процесом. Впровадження технології Інтернету речей (IoT) дозволяє автоматизувати збір даних, що сприяє економії часу та ресурсів. Саме тому концепція розумного лічильника електроенергії набуває популярності у світі.

У проєкті використано високоточний датчик напруги ZMPT101B, який побудований на основі відповідного трансформатора напруги. Даний модуль забезпечує точне вимірювання змінної напруги в межах 250В, а його аналоговий вихід можна налаштувати за допомогою багатооборотного потенціометра для калібрування сигналу.

Живлення всіх компонентів здійснюється від ESP32, який забезпечує 5В та 3.3В. Виводи VCC і GND датчиків SCT-013 та ZMPT101B підключені до Vin та GND ESP32, відповідно. Аналогові виходи датчиків підключені до GPIO34 (для SCT-013) та GPIO35 (для ZMPT101B). Для коректної роботи схеми додатково використано два резистори 10 кОм, один резистор 100 Ом та конденсатор 10 мкФ.

Для відображення інформації можна використовувати додатковий 20×4 LCD-дисплей з I2C-інтерфейсом, підключивши його до GPIO21 (SDA) та GPIO22 (SCL) ESP32. Контрастність дисплея регулюється за допомогою потенціометра на I2C-модулі. Однак, основний спосіб моніторингу – через мобільний додаток Blynk 2.0, що дозволяє переглядати дані про спожиту енергію з будь-якої точки світу.

Таке рішення дозволяє ефективно відстежувати споживання електроенергії в реальному часі, що є важливим для оптимізації витрат та контролю енергоспоживання.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>23 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

На рисунку 2.2 наведено схему електричну принципову програмно-апаратного пристрою розумний лічильник електроенергії IoT з ESP32 і Blynk 2.0.

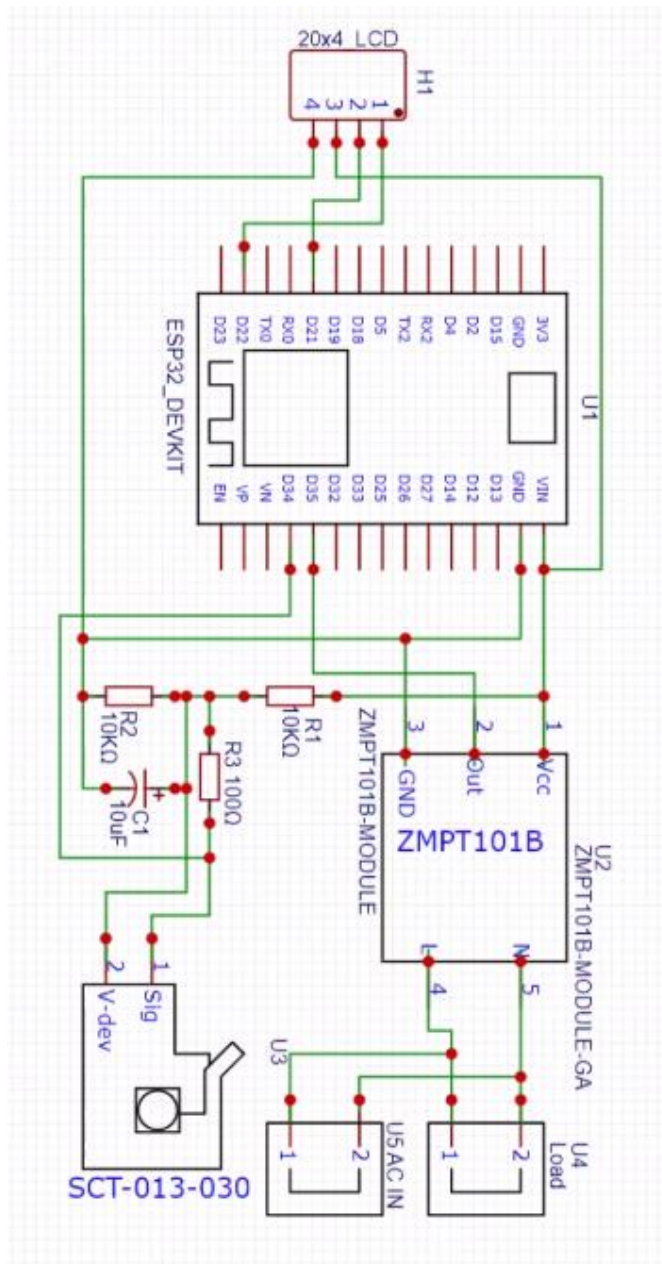


Рисунок 2.2 – Схема електрична принципова програмно-апаратного пристрою розумний лічильник електроенергії IoT

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

## 2.4 Монтажна схема проектованого програмно-технічного пристрою

Монтажна схема – це графічне зображення розташування та підключення компонентів системи на реальному об’єкті (у цьому випадку – в системі “Розумний лічильник електроенергії IoT з ESP32”). Вона вказує на конкретне розташування кожного компонента, джерела живлення, з’єднання між ними, а також інші важливі деталі.

Основне призначення монтажною схемою полягає в тому, щоб забезпечити коректний та ефективний монтаж системи, дозволяючи адміністраторам чітко розуміти, де і як слід розмішувати кожен компонент, як підключати їх до живлення та інших пристроїв, та забезпечити відповідні умови для їх функціонування. Це допомагає забезпечити правильну інсталяцію системи та підвищити ефективність її роботи.

Монтажну схему для розумного лічильника електроенергії IoT на основі ESP32 реалізовано таким чином: пини VCC і GND датчиків струму SCT-013 та напруги ZMPT101B під’єднані до пінів Vin і GND мікроконтролера ESP32, що забезпечують живлення 5 В. Аналоговий вихідний пін датчика напруги ZMPT101B під’єднано до пину GPIO35 на ESP32, а аналоговий вихід датчика струму SCT-013 до пину GPIO34. Для завершення схеми також використовуються два резистори на 10 кОм, один резистор на 100 Ом та конденсатор на 10 мкФ. Для вимірювання струму та напруги необхідно підключити АС дроти до вхідного терміналу датчика напруги. Для датчика струму достатньо під’єднати один провід (житловий або нейтральний) до кліпси датчика.

Додатково можна підключити I2C LCD дисплей 20×4 для відображення даних. Для цього необхідно під’єднати пини VCC, GND, SDA та SCL дисплея до пінів 5V, GND, GPIO21 та GPIO22 мікроконтролера ESP32 відповідно. Для налаштування контрасту LCD дисплея використовується потенціометр 10К, що знаходиться на задній частині I2C модуля. Якщо LCD дисплей не

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>25 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

використовується, можна моніторити дані про енергоспоживання через мобільний додаток Vlynk без підключення дисплея.

Монтажну схему програмно-апаратного засобу розумний лічильник електроенергії IoT подано на рисунку 2.3.

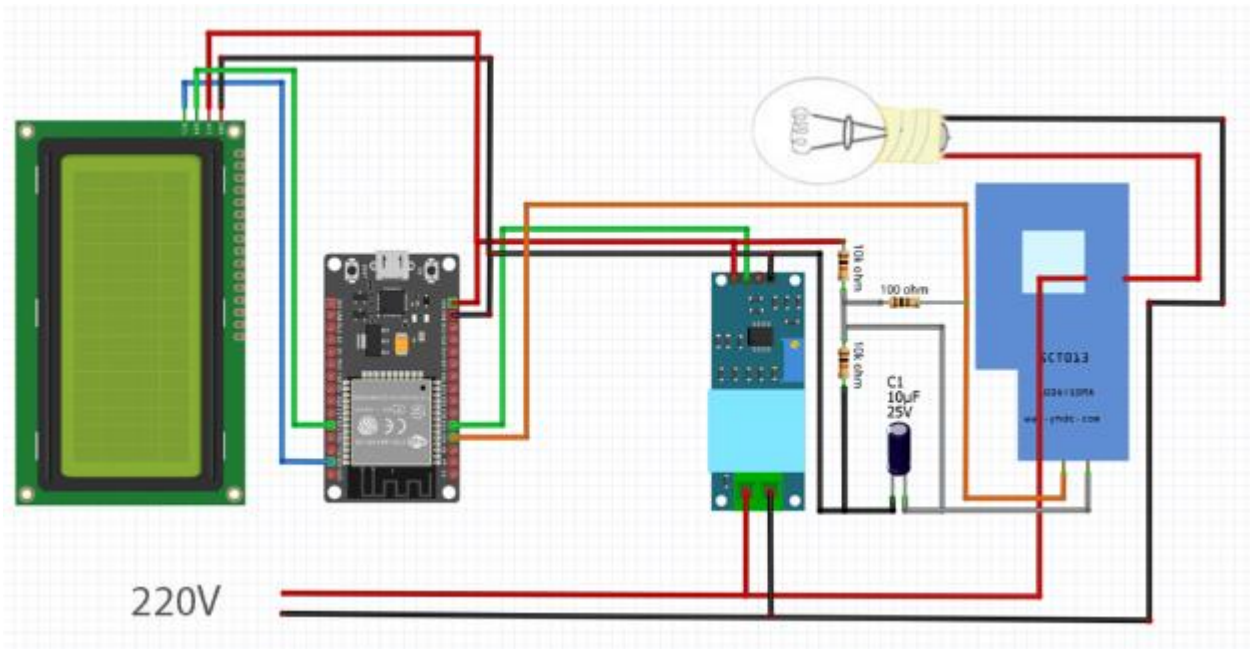


Рисунок 2.3 – Монтажна схема програмно-апаратного засобу розумний лічильник електроенергії IoT

## 2.5 Концептуальна модель бази даних для збереження показників енергоспоживання

Концептуальна модель бази даних призначена для фіксації, обробки та подальшого аналізу показників споживання енергії. Вона являє собою логічне представлення структури даних, що не залежить від конкретної мови програмування, але в даному випадку буде реалізована з використанням реляційної СУБД SQLite, котра є оптимальним рішенням для вбудованих чи легких локальних додатків.

Основою цієї моделі є запис споживання енергії, що включає в себе всю необхідну інформацію про стан енергоспоживання в конкретний момент часу.

Кожен запис обов'язково містить мітку часу, яка точно фіксує момент проведення вимірювання. Це дає змогу будувати часові ряди даних, аналізувати зміни у споживанні енергії з плином часу, виявляти пікові навантаження та періоди мінімального споживання.

На додаток до мітки часу, важливим параметром є напруга ( $V_{rms}$ ), тобто ефективне значення напруги в мережі на момент вимірювання. Цей параметр використовується для оцінки стабільності електропостачання, а також для виявлення потенційних відхилень чи проблем у мережі.

Також фіксується сила струму ( $I_{rms}$ ), що є ефективним значенням електричного струму. Вона дає змогу оцінити навантаження на систему та виявити нераціональне або аномальне споживання електроенергії. Разом з напругою ці два показники використовуються для обчислення потужності.

Потужність (Power) у записі відображає миттєве споживання електроенергії. Вона розраховується як добуток напруги на силу струму та дозволяє оцінити інтенсивність споживання енергії в конкретний момент часу. Цей параметр є критично важливим для виявлення перевантажень у мережі та для формування звітів про споживання енергії.

Додатково, модель включає параметр накопиченого споживання електроенергії (kWh), котрий дозволяє фіксувати загальний обсяг спожитої енергії з часом, що критично важливо для довгострокової статистики, обліку та виставлення рахунків.

Завдяки такій структурі, база даних дозволяє ефективно збирати, зберігати та аналізувати дані, що надходять з систем моніторингу енергоспоживання. Вона може слугувати основою для створення аналітичних звітів, візуалізацій, систем оповіщення та прогнозування енергетичних потреб.

## 2.6 Аналіз обраних рішень

Проведемо більш детальний аналіз апаратних та програмних рішень у системі розумного лічильника електроенергії.

### 2.6.1 Аналіз обраних апаратних рішень

Під час проектування програмно-апаратного засобу було включено такі апаратні компоненти: мікроконтролерна система ESP32 WIFI Module, модуль датчика вимірювання струму SCT-013, модуль датчика вимірювання напруги ZMPT-101B, дисплей для виведення інформації 20x4 I2C LCD Display. Проведемо більш деталізований аналіз обраних апаратних рішень.

Важливою складовою програмно-апаратного пристрою, що розробляється, є система мікроконтролерів WiFi модуля ESP32. ESP32 призначена для створення компактного модуля на базі мікросхеми ESP32 - пристрою з підтримкою Wi-Fi та Bluetooth зв'язку. Пристрій є зручним рішенням для створення систем IoT, розумних будинків, бездротових датчиків, систем автоматичного управління та інших проектів, які потребують стабільного бездротового зв'язку.

Мікроконтролер ESP32 – чудовий вибір для розробки IoT-пристроїв завдяки інтегрованим модулям Wi-Fi 802.11 b/g/n та Bluetooth 4.2 (як класичним, так і BLE), які дозволяють йому функціонувати в режимах точки доступу, клієнта або обох одночасно, забезпечуючи стабільне підключення до мережі(рис.2.5). Він обладнаний потужним двоядерним процесором Tensilica Xtensa LX6 з частотою до 240 МГц, що дає можливість ефективно опрацьовувати великі масиви даних, здійснювати складні обчислення, працювати з аудіо- та відеосигналами, а також реалізовувати алгоритми штучного інтелекту. ESP32 підтримує різні режими енергозаощадження: від активного до глибокого сну та гібернації, що робить його відповідним для автономних систем з мінімальним енергоспоживанням. Широкі можливості підключення периферії дозволяють ESP32 легко інтегруватися з різними датчиками, дисплеями, реле та іншими модулями через інтерфейси GPIO,

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КвРКІ.210116.21.01.44 ПЗ | Арк.<br>28 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

UART, SPI, I2C, PWM, ADC і DAC. Крім того, він підтримує програмування в різних середовищах розробки, як-от Arduino IDE, ESP-IDF, MicroPython, CircuitPython та PlatformIO, що забезпечує гнучкість у розробці як для початківців, так і для професіоналів. Завдяки компактним габаритам та низькій ціні ESP32 ідеально підходить для мініатюрних і бюджетних проєктів, таких як смарт-гаджети, сенсори та портативні пристрої. З метою забезпечення безпеки ESP32 підтримує сучасні протоколи захисту, зокрема WPA/WPA2, апаратне шифрування AES, SHA, RSA, а також функції Secure Boot та Flash Encryption для захисту коду та внутрішньої пам'яті мікроконтролера від несанкціонованого доступу [13]

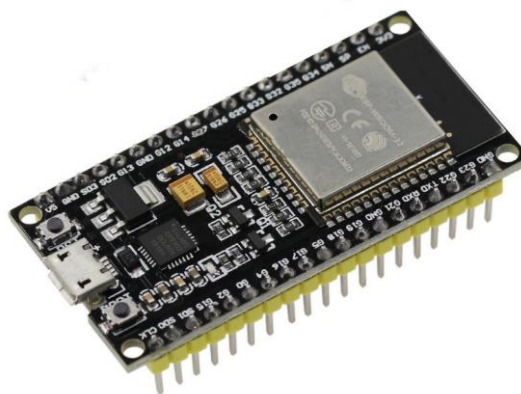


Рисунок 2.5 – Мікроконтролерна система ESP32 WIFI module

До основних компонентів, що складають мікроконтролер ESP32 Wi-Fi module, належить центральний чип ESP32-S, який включає два 32-розрядні високопродуктивні процесори LX6 з 7-ступінчастою конвеєрною архітектурою, здатних працювати на частоті до 240 МГц, що забезпечує високу обчислювальну потужність, управління периферією та підтримку бездротового зв'язку. Для покращення якості сигналу й збільшення радіусу дії використовується IPEX-роз'єм, який дозволяє підключити зовнішню антену, що особливо актуально в умовах слабого сигналу або на великих відстанях. У модулі також застосовується танталовий конденсатор для стабілізації напруги живлення та зменшення електричних шумів, а також стабілізатор напруги, що забезпечує постійне

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>29 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

живлення 3,3 В незалежно від коливань вхідної напруги. Кнопка скидання дозволяє перезапустити пристрій і повторно виконати програму з початку, що зручно для тестування або усунення помилок. У деяких версіях модуля присутня додаткова оперативна пам'ять PSRAM обсягом 4–8 МБ, що розширює можливості роботи з великими обсягами даних або ресурсоемними процесами. Також може бути інтегровано роз'єм для TF-карти пам'яті, який дозволяє зберігати дані на зовнішньому накопичувачі через SPI-інтерфейс із швидким доступом. Велику гнучкість забезпечують GPIO універсальні входи/виходи, які можуть використовуватись для підключення різноманітної периферії, включно з підтримкою таких протоколів, як PWM, I2C, SPI, UART, I2S, а також сенсорного введення. У деяких версіях модуля додатково вбудований світлодіод, який можна використовувати як індикатор стану або для сигналізації різних подій у роботі пристрою ESP32 WiFi Module підтримує два рівні напруги живлення: 3,3В та 5В.

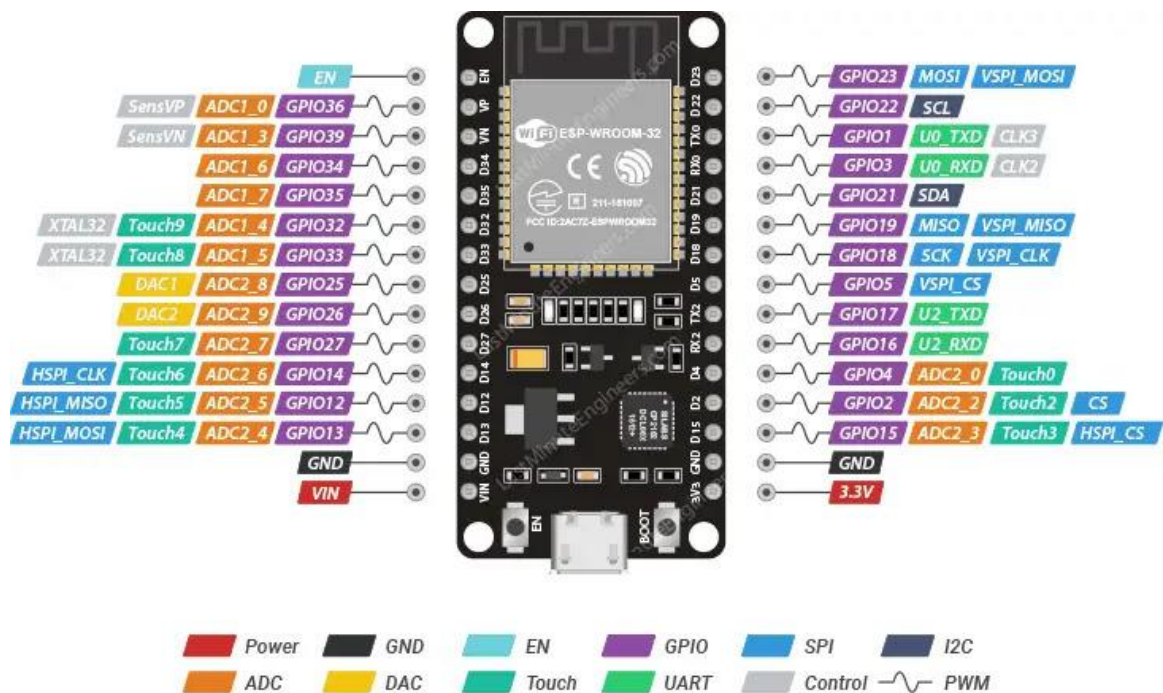
Для стабільної роботи рекомендується подавати 5В, оскільки при 3,3В можливі випадкові перезавантаження через недостатню стабільність напруги.

На модулі передбачено спеціальний пін VCC, який можна використовувати для живлення зовнішніх пристроїв. Його максимальний вихідний струм 40 мА, що достатньо для роботи датчиків або невеликих сенсорів. Окрім основного режиму роботи, ESP32 підтримує кілька енергозберігаючих режимів:

Модуль ESP32 WiFi пропонує декілька варіантів економії енергії. Серед них можна виділити такі як: Light Sleep, де процесор активний, а периферійні пристрої здебільшого вимкнені; Deep Sleep, який занурює мікроконтролер у стан мінімального енергоспоживання з активацією лише таймера або зовнішнього переривання; та Hibernation, що забезпечує найменше споживання енергії, з можливістю збереження ключових даних. Ці режими роблять ESP32 корисним для пристроїв, що працюють від батарей, як-от сенсори або мобільні рішення для Інтернету речей. Модуль підтримує дві основні бездротові технології: Wi-Fi 802.11 b/g/n, що функціонує на частоті 2,4 ГГц, забезпечуючи швидкий і надійний зв'язок у локальних мережах, та Bluetooth 4.2 / BLE – технологія з низьким

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>30 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

енергоспоживанням для з'єднання зі смартфонами, смартгодинниками та датчиками. Вбудована Wi-Fi антена має потужність приблизно 4 дБ, але за потреби можна приєднати зовнішню антену через IPEX-роз'єм. Для цього на платі є спеціальна перемичка, що дозволяє вибрати між внутрішньою та зовнішньою антенами. Для обміну даними з іншими пристроями ESP32 має підтримку UART (з GPIO1 (TX) для передавання та GPIO3 (RX) для отримання даних), SPI для швидкої взаємодії з периферією, I2C для підключення сенсорів і дисплеїв, а також I2S, що дозволяє працювати з аудіообладнанням, як-от мікрофони та динаміки. Для програмування ESP32 потрібен зовнішній USB-UART перетворювач, оскільки сам модуль не має вбудованого програматора. Код завантажується через UART-інтерфейс за допомогою середовища розробки, наприклад Arduino IDE, PlatformIO або ESP-IDF. На рисунку 2.6 зображено розпіновку ESP32 WIFI Module [14].



моніторингу електроспоживання в житлових і промислових приміщеннях. Завдяки своєму конструктивному рішенню, цей датчик не потребує безпосереднього контакту з провідниками під високою напругою, що забезпечує безпечне та зручне використання.

Принцип роботи SCT-013 полягає у виявленні магнітного поля навколо провідника зі струмом. Датчик містить первинну обмотку, магнітне осердя та вторинну обмотку, що складається з великої кількості витків тонкого дроту. Це дозволяє датчику генерувати пропорційний вихідний сигнал, який можна зчитувати за допомогою мікроконтролерів, таких як Arduino, Raspberry Pi та інших.

SCT-013 широко застосовується у системах енергомоніторингу, розумних будинках, автоматизованих системах керування електроспоживанням та інших електротехнічних пристроях. Його використовують для вимірювання струму в побутових електромережах, промислового обладнанні, системах розподілу енергії та для аналізу споживання електроенергії.

Цей модуль повністю корелюється з контролерами Arduino всіх лінійок та також добре функціонує з мікроконтролерами Arduino, ESP8266, ESP32 та аналогічними платформами. Він ефективно обробляє вхідний змінний струм від 0 до 30 ампер, перетворюючи його у вихідний сигнал постійного струму в межах від 0 до 1 вольт. Модуль має нелінійність у межах 2-3%, оснащений вбудованим шунтовим резистором на 62 Ом та коефіцієнтом трансформації 1800:1. Клас точності В, а температурний діапазон його експлуатації становить від -25 °С до +70 °С. Діелектрична міцність модуля гарантується на рівні 1000 вольт змінного струму протягом однієї хвилини, за умови, що струм не перевищує 5 мА [16].

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>32 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |



Рисунок 2.7 – Модуль датчика вимірювання струму SCT-013

Для відстеження точних показників напруги було обрано датчик ZMPT101B AC (рис.2.8). Це високоточний модуль, побудований на основі трансформатора напруги ZMPT101B, який забезпечує точне вимірювання змінної напруги. Завдяки своїй чутливості та компактним розмірам, він широко використовується для моніторингу напруги в електромережах, аналізу навантаження та створення інтелектуальних систем керування електроенергією.

Модуль підтримує вимірювання змінної напруги до 250В, що робить його ідеальним вибором для використання з мікроконтролерами Arduino, ESP32, Raspberry Pi та іншими. Він має регульований аналоговий вихід, який можна підключити до АЦП (ADC) мікроконтролера для отримання точних цифрових значень напруги.

Однією з головних особливостей цього датчика є наявність багатооборотного підлаштувального резистора, який дозволяє точно налаштувати рівень вихідного сигналу відповідно до потреб конкретного застосування. Це робить ZMPT101B зручним для калібрування в системах контролю електромережі та енергомоніторингу.

Датчик ZMPT101B знайшов широке застосування в системах моніторингу напруги, як вдома, так і на виробництвах, для вимірювання показників електроживлення в системах "Розумний дім", для аналізу якості електроенергії, реалізації автоматизованих систем управління навантаженням та вбудовуваних

рішень для енергозбереження. Модуль бездоганно інтегрується з мікроконтролерами, серед яких Arduino, ESP8266, ESP32, Raspberry Pi та інші. Він дає змогу вимірювати напругу змінного струму до 250 В і генерує аналоговий вихідний сигнал, що відповідає вхідній напрузі. Пристрій відзначається низьким рівнем нелінійності ( $\leq 0,2\%$ ) та надає можливість точного налаштування вихідного сигналу за допомогою багатооборотного резистора. Робочий температурний діапазон становить від  $-40^{\circ}\text{C}$  до  $+70^{\circ}\text{C}$ , живлення здійснюється від джерела постійного струму 5 В, а також забезпечується висока електрична ізоляція між первинною та вторинною сторонами, що гарантує безпеку використання.

Модуль ZMPT101B є ефективним рішенням для проєктів, що потребують надійного та безпечного вимірювання змінної напруги, і може бути використаний як у домашніх умовах, так і в промислових застосуваннях [19].



Рисунок 2.8 – Модуль датчика змінного струму ZMPT101B AC

В якості дисплею для моментального виведення інформації про моніторинг було використано 20x4 I2C LCD Display (рис.2.9). Даний пристрій це рідкокристалічний дисплей з 20 символами в рядку та 4 рядками, який використовує інтерфейс I2C для зручного підключення до мікроконтролерів. Завдяки використанню I2C адаптера, дисплей потребує лише двох сигналів для зв'язку з мікроконтролером (SDA та SCL), що значно зменшує кількість задіяних

виводів у порівнянні зі стандартним 20x4 LCD, який працює через паралельний інтерфейс.

Цей дисплей є чудовим вибором для вбудованих систем, де важливо зберегти вільні пін-контакти для інших компонентів. Він підтримується багатьма популярними платформами, такими як Arduino, ESP32, Raspberry Pi, STM32, та має сумісність із бібліотекою LiquidCrystal\_I2C, що значно спрощує програмування та виведення інформації.



Рисунок 2.9 – 20x4 I2C LCD Display

Рідкокристалічний дисплей 20x4 з інтерфейсом I2C вирізняється простотою підключення, використовуючи лише два дроти (SDA, SCL), що дозволяє значно зекономити піни мікроконтролера (замість 6-10), при цьому забезпечуючи статичне енергоспоживання менше ніж 0,1 мА і час блокування близько 2 секунд. Високий контраст і чіткість символів роблять його зручним для відображення текстової інформації, а можливість регулювання підсвічування дозволяє адаптувати яскравість до умов використання. Завдяки широкій підтримці бібліотек дисплей легко інтегрується з Arduino, ESP32, Raspberry Pi, STM32 та іншими платформами. Живлення здійснюється від 5 В, що є стандартним для більшості мікроконтролерів, а також підтримується відображення кирилиці, включно з українським алфавітом. Загальні габарити об'єктива дисплея становлять 12 мм на 25 мм [20].

## 2.6.2 Аналіз обраних програмних рішень

З метою автоматизованого моніторингу споживання електроенергії та віддаленого контролю параметрів енергоспоживання використано платформу Blynk 2.0 у поєднанні з контролером ESP32.

Blynk 2.0 – це сучасна IoT-платформа, яка надає можливість зручного збору, аналізу та відображення даних у реальному часі. Вона підтримує інтеграцію з мікроконтролерами, забезпечує віддалене керування пристроями та надає зручний інтерфейс для користувачів через мобільні додатки та вебпанель.

Основними перевагами використання Blynk 2.0 в цьому пристрої є зручність його впровадження, адже ця платформа пропонує гнучкий API та SDK для розробників, що суттєво полегшує інтеграцію з мікроконтролерами, зокрема, з ESP32. Завдяки цьому з'являється можливість автоматично передавати дані про споживання електроенергії та отримувати команди від користувача. Платформа також гарантує високий рівень захисту, підтримуючи шифрування та автентифікацію, що забезпечує захист даних під час передачі. Додатково Blynk 2.0 доступний на мобільних пристроях та через вебінтерфейс, що дозволяє відстежувати та керувати показниками споживання електроенергії з будь-якої точки. Платформа пропонує функціональність для візуалізації даних у вигляді графіків, віджетів і таблиць, а також дає змогу налаштовувати автоматичні сповіщення у разі перевищення встановлених лімітів споживання енергії. Blynk 2.0 також відрізняється крос-платформеністю, функціонуючи на iOS, Android та у веб-браузерах, що дозволяє користувачам управляти енергоспоживанням зі свого смартфона, планшета чи комп'ютера [24].

Завдяки інтеграції ESP32, датчиків напруги (ZMPT101B) та струму (SCT-013), система може вимірювати основні електричні параметри: напругу, силу струму, потужність та загальне споживання енергії в кВт·год. У разі відключення живлення дані зберігаються у пам'яті ESP32, що забезпечує безперервний контроль за енергоспоживанням.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>36 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

Такий пристрій дозволяє спростити моніторинг електроенергії, зменшити витрати та зробити процес керування більш автоматизованим та ефективним.

## 2.7 Перспективи розвитку системи

Після втілення у життя базової версії системи контролю споживання електроенергії відкривається перспектива значного покращення її функціональності, можливість масштабування потенціалу аналізу та інтеграції з передовими цифровими рішеннями. У сучасних умовах, коли зростають потреби у енергоефективності, автоматизації та аналітиці, додавання нових технологій є не просто бажаним, а й вкрай необхідним рішенням. Нижче представлено ключові напрями удосконалення.

### 2.7.1 Використання хмарних баз даних(Firebase, InfluxDB)

У початковій версії системи дані про електроспоживання зберігаються локально, використовуючи вбудовану реляційну базу даних SQLite. Такий підхід є достатнім для локального тестування, налагодження та роботи в умовах обмеженого доступу до мережі. Проте в умовах реального багатокористувацького застосування виникає потреба у централізованому зберіганні інформації, що забезпечить кращу доступність, збереження історичних даних та можливість роботи з великим обсягом інформації. Для цього доцільним є перехід до використання хмарних баз даних.

Firebase Realtime Database та Firestore дозволяють зберігати і синхронізувати дані в режимі реального часу з різними пристроями через інтернет. Це особливо корисно у випадках, коли важливо бачити останні зміни миттєво, наприклад, при візуалізації показників електроспоживання в онлайн-режимі. Крім того, дані зберігаються у хмарному середовищі з вбудованим резервним копіюванням, що значно підвищує надійність системи [27].

Іншим варіантом є використання InfluxDB – спеціалізованої бази даних для роботи з часовими рядами. Такий підхід дозволяє оптимізувати зберігання великої кількості показників, які фіксуються з високою частотою, та полегшує подальший аналіз і агрегування інформації за певними інтервалами. Таким чином, перехід до хмарних баз даних не лише забезпечить доступність даних з будь-якої точки світу, але й створить основу для подальшої інтеграції з аналітичними платформами та інтелектуальними системами прогнозування [30].

### 2.7.2 Аналіз та візуалізація даних через графіки(Grafana, Power BI)

Одним з ключових моментів в обробці даних є їхня результативна візуалізація й аналіз. Навіть значна кількість зібраної інформації про електроспоживання без відповідного набору інструментів, ускладнює аналіз та знижує його ефективність. Зважаючи на це, впровадження систем візуалізації даних, як-от Grafana або Power BI, виглядає перспективним шляхом розвитку.

Grafana є платформою для аналізу з відкритим кодом, що чудово взаємодіє з базами даних часових рядів, наприклад, InfluxDB. Вона дозволяє формувати інтерактивні інформаційні панелі, відображати зміни показників в реальному часі, а також зіставляти різні часові проміжки чи пристрої. Завдяки великому вибору налаштувань та підтримці сповіщень, Grafana може використовуватись як потужний інструмент для моніторингу важливих систем [33].

Іншим вагомим варіантом є Microsoft Power BI, який дозволяє будувати поглиблені аналітичні звіти, об'єднувати кілька джерел даних та використовувати вбудовані алгоритми для прогнозування. Це надає можливість виявляти тенденції у споживанні енергії, аналізувати пікові навантаження, оцінювати вплив окремих приладів на загальне споживання, а також формувати рекомендації для оптимізації витрат. Застосування таких інструментів зробить систему не тільки інформативною, а й аналітично корисною для кінцевих споживачів [35].

### 2.7.3 Розширення функціоналу на систему пристроїв

Первинний варіант втілює моніторинг одного пристрою, що цілком достатньо для первинної перевірки працездатності системи. Проте, у реаліях енергоспоживання розподіляється між різними ділянками-приміщеннями, офісами, виробничими цехами чи окремими приладами. Зважаючи на це, важливим етапом розвитку системи є її розширення для забезпечення одночасного моніторингу кількох ділянок або пристроїв.

Це може включати роздільне відстеження енерговитрат у різних кімнатах житла, таких як кухня, ванна, вітальня, або ж окремий облік для потужних споживачів, як, наприклад, кондиціонери, бойлери, пральні машини. У промислових умовах можливо впровадження системи, яка контролює кожен електроприлад окремо, що дає змогу точно локалізувати втрати енергії чи виявити несправності в обладнанні.

Технічно це можна реалізувати шляхом приєднання кількох контролерів ESP32, кожен з яких надсилає свої дані на центральний сервер або у хмарну базу даних. Відповідно, виникає потреба у масштабуванні програмного забезпечення для ідентифікації, опрацювання та зберігання даних з кожного джерела. Підсумком такого підходу буде значне підвищення точності контролю, що дасть змогу користувачу ухвалювати обґрунтовані рішення стосовно енергоефективності.

### 2.7.4 Підтримка декількох зон або пристроїв

Сучасні інформаційні системи дедалі більше використовують алгоритми штучного інтелекту та машинного навчання для автоматизації аналізу даних і прийняття рішень. Система моніторингу електроспоживання має великі можливості для застосування цих технологій. На основі зібраної історичної інформації можна формувати моделі, котрі здатні не тільки аналізувати структуру споживання, але й прогнозувати її в майбутньому, виявляючи відхилення від нормального перебігу подій, або, простіше кажучи, аномалії.

Виявлення аномалій дає змогу системі самостійно інформувати про незвичайні ситуації, такі як раптове збільшення енергоспоживання, зниження напруги чи поломка обладнання. Це надзвичайно важливо на критичних об'єктах, де швидке реагування на проблеми допомагає уникнути серйозних негативних наслідків. Прогнозування енергоспоживання, в свою чергу, сприятиме оптимізації розподілу навантаження, плануванню графіків роботи устаткування та зменшенню пікових навантажень.

Впровадити ці функції можна за допомогою фреймворків машинного навчання, зокрема TensorFlow або Scikit-learn. Також існують хмарні платформи з підтримкою AutoML, такі як Google Cloud AutoML або Microsoft Azure ML, які дозволяють налаштовувати процес навчання моделей без необхідності глибокого знання в галузі Data Science. Відтак, інтеграція AI-модулів у систему моніторингу забезпечує значне збільшення її інтелектуального рівня та функціональності.

## 2.8 Висновки до другого розділу

У підсумку цей проект реалізує IoT-систему, яка автоматично контролює споживання електроенергії. Визначено основні вимоги до пристрою, проаналізовано його компоненти та запропоновано ефективну структуру. Система надає дані про споживання електроенергії в режимі реального часу, значно спрощуючи управління витратами та пропонуючи простоту використання. Інформація надається користувачеві через мобільний додаток або веб-інтерфейс, а зберігання даних забезпечує безперервний моніторинг навіть при перервах у роботі. Таким чином, запропоноване рішення є ефективним інструментом для автоматизації процесу збору даних про споживання електроенергії і може використовуватися як у побутових, так і в промислових умовах.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>40 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНО-ТЕХНІЧНОГО ПРИСТРОЮ ВИМІРЮВАННЯ ТА МОНІТОРИНГУ ПАРАМЕТРІВ ЕЛЕКТРОСПОЖИВАННЯ НА ОСНОВІ ІОТ З ВИКОРИСТАННЯМ ESP32 І ПЛАТФОРМИ VLYNK

#### 3.1 Підготовка серверу та бази даних

Для гарантування безпечного зберігання вимірних показників електроспоживання, отриманих з пристрою на базі ESP32, постала необхідність створити серверну частину, що здатна обробляти та приймати вхідні запити за протоколом HTTP. Також сервер мусить забезпечувати можливість доступу до збережених даних з метою подальшого аналізу чи відображення на веб-ресурсі.

У рамках цього проєкту було вирішено використовувати Python як мову програмування для реалізації серверної частини, зокрема фреймворк Flask. Вибір Flask зумовлений його простотою, легкістю інтеграції, лаконічністю та швидкістю розгортання. Цей фреймворк надає можливість створити повноцінний REST API для передавання та отримання даних, уникнувши зайвих налаштувань [42].

Для зберігання інформації про виміряні значення (напруга, сила струму, потужність, спожита енергія) було вирішено використовувати базу даних SQLite. Основна перевага SQLite полягає у відсутності потреби у окремому сервері для роботи, що дозволяє зберігати дані у вигляді єдиного файлу, безпосередньо в межах проєкту. Це спрощує налаштування та значно зменшує складність у порівнянні з традиційними СУБД (наприклад, MySQL або PostgreSQL), особливо на етапі розробки та тестування.

Крім того, середовище розробки PyCharm пропонує зручний графічний інтерфейс для роботи з базами даних, що дає змогу оперативно переглядати, змінювати або видаляти дані, не вдаючись до написання окремих SQL-запитів вручну. Це значно спрощує процес налагодження та обслуговування проєкту.

Отже сервер надає такі функції як: надання запитів для відправки та отримання даних з бази даних, збереження даних до бази даних, виведення записів у веб сторінці.

Для того щоб розпочати роботу над сервером, спершу необхідно ініціалізувати базу даних, для цього напишемо функцію яка буде створювати встановлювати зв'язок з нашою базою даних(energy\_data.db) та яка буде створювати в ній таблицю energy з полями timestamp, vrms, irms, power, kwh, якщо дана таблиця ще не існує [38].

```
def init_db():
    conn = sqlite3.connect("energy_data.db")
    c = conn.cursor()
    c.execute('''
        CREATE TABLE IF NOT EXISTS energy (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            timestamp TEXT,
            vrms REAL,
            irms REAL,
            power REAL,
            kwh REAL
        )
    ''')
    conn.commit()
    conn.close()
```

Для реалізації запису серверу створимо функцію run\_app()

```
def run_app():
    app.run(host='0.0.0.0', port=5000)
```

Для того щоб здійснити запуск програми, використовуючи наші обидві функції, ми застосуємо багатопоточність за допомогою модулю threading. Ми створимо два потоки і запустимо їхнє виконання один за одним.

```
if __name__ == '__main__':  
    first_thread = threading.Thread(target=run_app)  
    second_thread = threading.Thread(target=init_db)  
    first_thread.start()  
    second_thread.start()
```

Наступним кроком для зручного перегляду інформації в базі даних буде додавання її файлу до Database and sources нашого проекту. Для цього необхідно створити новий Data Source та обрати Sql lite [40](рис.3.1)

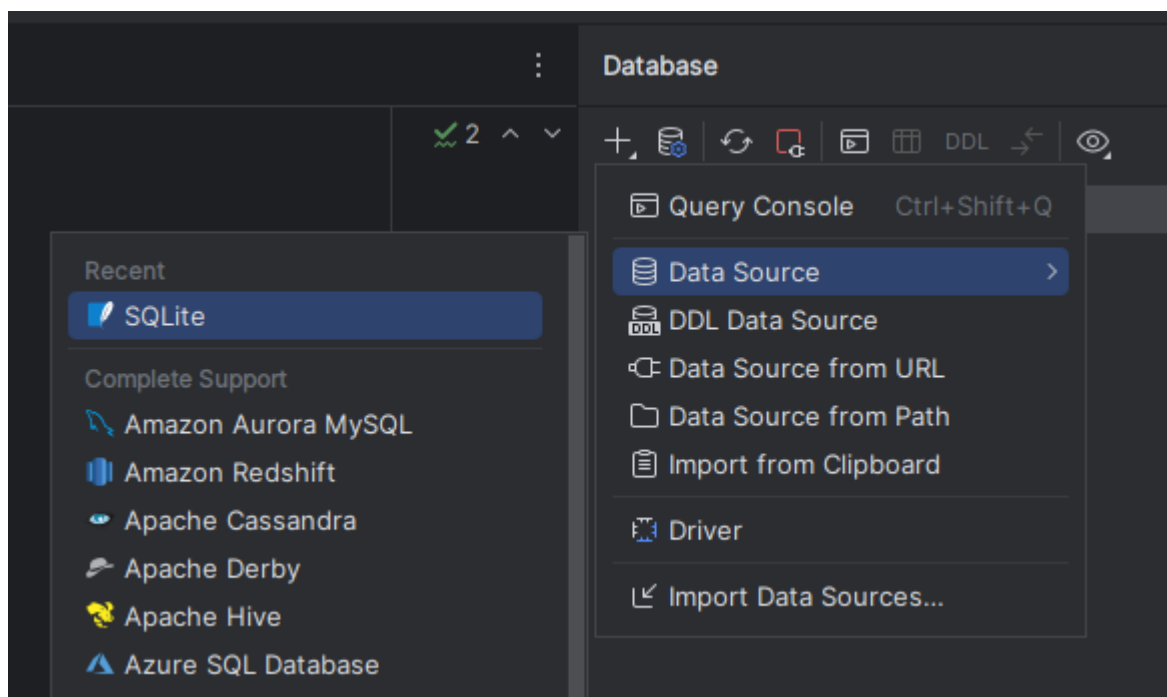


Рисунок 3.1 – Додавання бази даних до Data Source

Наступним кроком буде вибір нашого файлу бази даних у проекті(рис.3.2)

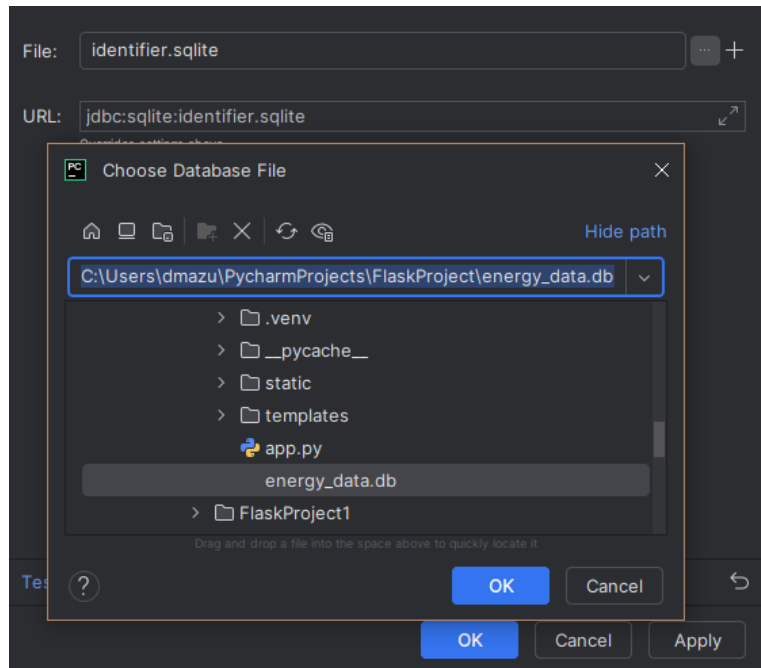


Рисунок 3.2 – Вибір нашої створеної БД

Результатом буде можливість перегляду таблиці нашої БД у форматі зображеному на рисунку 3.3.

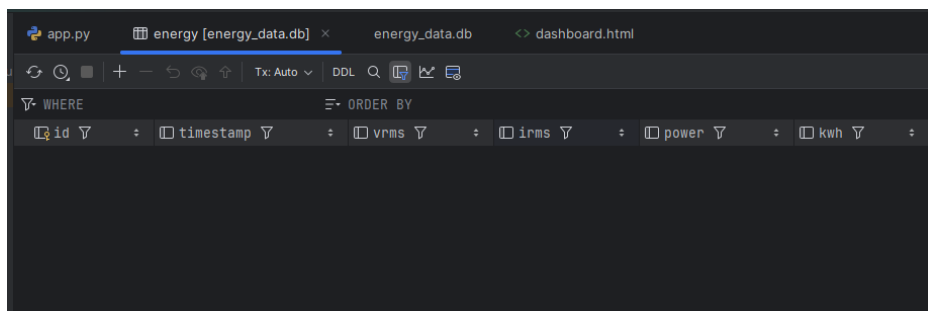


Рисунок 3.3 – Перегляд таблиці за допомогою інтерфейсу IDE

Тепер створимо два ендпоїнти та відповідні функції для запису даних до БД та їх зчитування з неї (функції `get_data()` та `receive_data()` відповідно). 1. Кінцева точка для отримання даних: `/api/data`

Перший ендпоїнт підтримує HTTP-метод GET. Якщо надіслати запит до `/api/data`, функція спочатку відкриває з'єднання з базою даних `energy_data.db`, а потім робить запит на вибірку останніх 20 записів з таблиці `energy`. Ці дані впорядковуються за часовою міткою у зворотньому порядку (від свіжих до старих).

Кожен рядок результату перетворюється на словник, що спрощує конвертацію у формат JSON. Після зчитування та форматування даних, з'єднання з базою даних закривається, а оброблені дані повертаються у вигляді відповіді JSON. Завдяки цьому, клієнти мають змогу зручно отримувати найсвіжіші дані про споживання енергії.

Другий ендпоінт призначений для внесення нових даних до бази даних. Він використовує HTTP-метод POST. Якщо отримується POST-запит на /data, функція очікує, що дані будуть надіслані у форматі JSON. Вона проводить перевірку на наявність даних; у випадку їх відсутності, повертається помилка 400 "No data received". Якщо дані присутні, встановлюється з'єднання з базою даних energy\_data.db, та значення vrms, irms, power і kwh (імовірно, напруга, сила струму, потужність та використана електроенергія) вставляються в таблицю energy разом з поточною позначкою часу. Після успішного запису даних, зміни фіксуються, з'єднання закривається, і надсилається успішний HTTP-статус 200 з повідомленням "Data saved successfully". Це дозволяє зовнішнім пристроям або додатковим програмам передавати нові дані про споживання енергії для їх подальшого зберігання [43].

Останнім кроком буде виведення даних з бази даних у простій html сторінці dashboard.html(рис.3.5). Таблиця відображає час передавання даних серед яких дата у форматі mm/dd/yyyy та час у дванадцятигодинному форматі.

| IoT Energy Meter Dashboard |          |          |           |         |
|----------------------------|----------|----------|-----------|---------|
| Timestamp                  | Vrms (V) | Irms (A) | Power (W) | kWh     |
| May 09, 2025, 10:47:27 AM  | 241.05   | 0.2900   | 69.9000   | 0.00117 |
| May 09, 2025, 10:47:22 AM  | 218.46   | 0.8800   | 192.2400  | 0.00320 |

Рисунок 3.4 – Вигляд таблиці для виведення даних

Алгоритм функціонування серверу стартує з моменту його ввімкнення, після чого здійснюється початкове налаштування бази даних, приводячи її у стан готовності для роботи. Далі сервер переходить у режим спостереження, постійно

перевіряючи наявність нових запитів. У момент отримання запиту, відбувається його аналіз, а потім, в залежності від типу запиту, виконуються певні дії: у випадку звернення до "api/data", запускається функція get\_data для збору необхідних відомостей; якщо запит містить "data", активується функція receive\_data для обробки отриманої інформації; у випадку запиту, пов'язаного з дашбордом, сервер обробляє його згідно з логікою відображення інформаційної панелі. Після виконання однієї з цих операцій, обробка поточного запиту завершується. Загальний алгоритм роботи зображено на рисунку 3.5

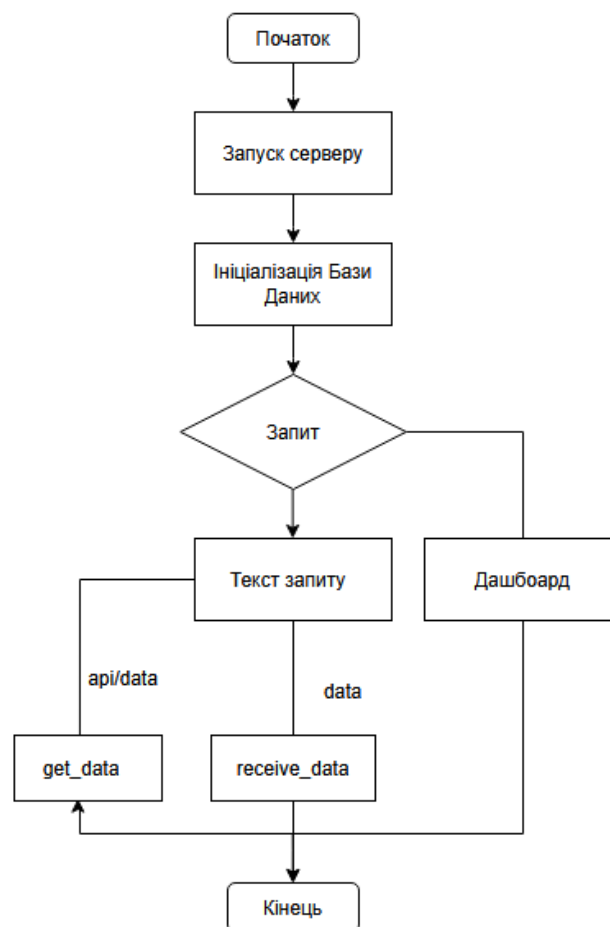


Рисунок 3.5 – Блоксхема роботи серверу

### 3.2 Розгортання та налаштування платформи Blynk 2.0

Для результативного відстеження енерговитрат у системі Інтернету речей (IoT), ключовим рішенням стала інтеграція з платформою Blynk 2.0, що дозволяє демонструвати життєво важливі показники споживання з будь-якого куточку світу. Ця платформа є сучасною та потужною інструментарієм для швидкого розгортання IoT-додатків, спрощуючи процес з'єднання мікроконтролерів з хмарним середовищем та мобільним додатком. Завдяки цій інтеграції, збір, моніторинг та представлення даних відбуваються в режимі реального часу, забезпечуючи негайний доступ до актуальної інформації.

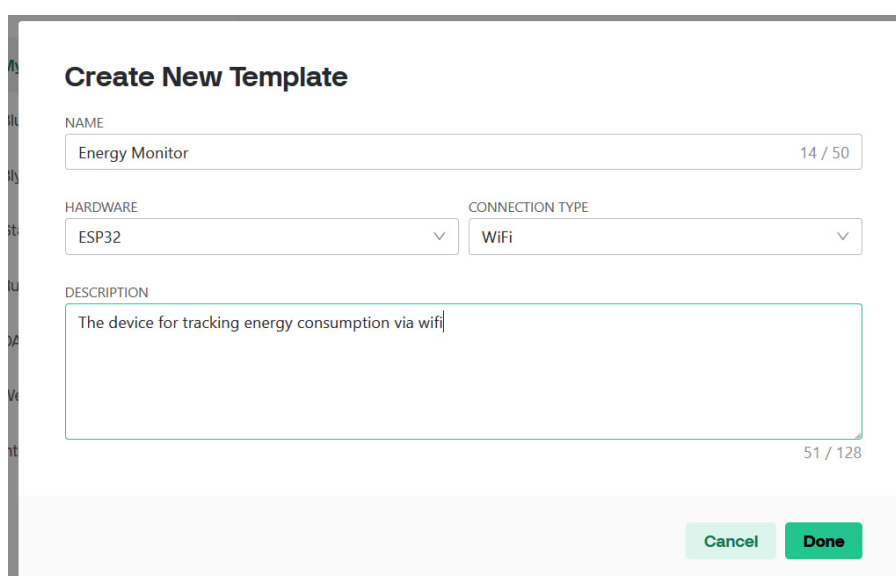
Blynk 2.0 вирізняється низкою суттєвих переваг, які зробили його оптимальним вибором для цього проекту. Перш за все, його хмарна інфраструктура повністю бере на себе завдання з підключення та передачі даних, звільняючи від потреби в налаштуванні власного серверу, що значно пришвидшує розробку та розгортання. Мобільний додаток платформи дає змогу користувачам інтуїтивно створювати власний інтерфейс для взаємодії з пристроєм, не потребуючи глибоких знань з програмування. Повна сумісність з платою ESP32, яка є ключовим компонентом проекту, забезпечує безшовну інтеграцію апаратного та програмного забезпечення. До того ж, Blynk 2.0 надає широкі можливості для візуалізації даних за допомогою різноманітних графіків та віджетів, що значно полегшує аналіз показників споживання. Безпечне з'єднання з аутентифікацією гарантує конфіденційність і цілісність переданих даних, що критично важливо для будь-якої IoT-системи.

Вибір Blynk 2.0 для цього проекту був обумовлений його можливістю забезпечити швидкий запуск системи моніторингу без необхідності розробки власних мобільних або веб-додатків. Платформа дає змогу миттєво відображати виміряні значення, підтримувати двосторонній зв'язок для керування пристроями та легко масштабується для підтримки збільшення кількості пристроїв або обсягів даних. Окрім мобільного додатку, Blynk 2.0 також передбачає виведення

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>47 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |



вказати назву шаблону, вибрати модель плати, яка використовується в проєкті. У нашому випадку це ESP32. Далі необхідно вказати тип з'єднання (Wi-Fi), а також за бажанням надати опис платформи, зазначивши, що пристрій базується на ESP32, використовує датчик струму SCT-013 та LCD-дисплей для локального виведення даних. Після створення шаблону система створить ідентифікатор пристрою, токен авторизації та інші технічні параметри, які необхідно буде внести в програмний код пристрою. Саме завдяки цим даним пристрій матиме змогу передавати виміряні показники на сервер Blynk та відобразити їх у мобільному додатку у вигляді графіків, лічильників або віджетів реального часу.



The screenshot shows a web form titled "Create New Template". It has the following fields:

- NAME:** A text input field containing "Energy Monitor" with a character count of "14 / 50".
- HARDWARE:** A dropdown menu with "ESP32" selected.
- CONNECTION TYPE:** A dropdown menu with "WiFi" selected.
- DESCRIPTION:** A text area containing "The device for tracking energy consumption via wifi" with a character count of "51 / 128".

At the bottom right, there are two buttons: "Cancel" (light green) and "Done" (dark green).

Рисунок 3.7 – Створення темплейту для системи

Для даного проєкту було обрано назву Energy Monitor. Також апаратне забезпечення ESP32 та тип з'єднання WiFi. Після підтвердження створення нового середовища ми переходимо на вкладку Web Dashboard та розміщуємо на ній чотири Gauge віджети які й будуть відображати наші показники(рис.3.8)



зчитувати дані й подавати їх у зручному інтерфейсі, що не перевантажує користувача надмірною або неактуальною інформацією. Далі буде детальніше розглянуто конфігурацію інших віджетів, які показують напругу, потужність, енергію та інші параметри електромережі.

The screenshot shows a configuration window titled 'Virtual Pin Datastream'. At the top, there is a 'TITLE (OPTIONAL)' field containing 'Irms'. Below this is the 'Datastream' section, which is divided into 'General' and 'Expose to Automations' tabs. The 'General' tab is active. It contains several fields: 'NAME' (Irms), 'ALIAS' (Irms), 'PIN' (V1), 'DATA TYPE' (Double), 'UNITS' (Ampere, A), 'MIN' (0), 'MAX' (10), 'DECIMALS' (###), and 'DEFAULT VALUE' (0). There is also a checkbox for 'Enable history data' which is currently unchecked. At the bottom, there is a section for 'ADVANCED SETTINGS' and two buttons: 'Cancel' and 'Create'.

Рисунок 3.9 – Налаштування віджету для Irms

Наступним кроком буде конфігурація віджету, призначеного для показу даних про активну потужність, яку споживає електричне навантаження. Цей віджет надзвичайно важливий, адже саме він відповідатиме за візуалізацію інформації, необхідної для розуміння рівня енергоспоживання певного пристрою чи системи в режимі реального часу. Варто надати йому ім'я Power, оскільки воно є прозорим та загальновизнаним терміном у сфері енергетики. За одиницю вимірювання визначаємо Watts (Вт), адже це стандартна одиниця для позначення активної потужності в міжнародній системі одиниць SI. Для належного масштабування графічного представлення значень, мінімальне значення встановлюємо на 0, а максимальне на 50. Це прийнятний діапазон для приладів домашнього використання

або низької потужності, які можуть бути під'єднані до системи моніторингу. Такий діапазон забезпечує можливість візуалізувати коливання потужності з необхідною точністю, зберігаючи деталізацію навіть при низькому навантаженні. Застосування типу даних double гарантує точне відображення значень з десятковими дробами, що дозволяє оперативно реагувати на зміни у споживанні. Правильне налаштування цього віджету дасть змогу користувачеві без проблем відслідковувати споживання електроенергії у ватах, виявляти пікові навантаження та планувати заходи з метою оптимізації енергоспоживання. На подальших етапах буде здійснено налаштування інших віджетів для більш детального та інформативного представлення параметрів енергосистеми.

The image shows a configuration window for a 'Virtual Pin Datastream'. The title is 'Power'. The 'Datastream' section is titled 'Virtual Pin Datastream'. There are two tabs: 'General' (selected) and 'Expose to Automations'. Under 'General', there are fields for 'NAME' (Power), 'ALIAS' (Power), 'PIN' (V2), 'DATA TYPE' (Double), and 'UNITS' (Watts, W). There are also fields for 'MIN' (0), 'MAX' (50), 'DECIMALS' (###), and 'DEFAULT VALUE' (Default Value). A checkbox for 'Enable history data' is checked. At the bottom, there is a section for 'ADVANCED SETTINGS' and two buttons: 'Cancel' and 'Create'.

Рисунок 3.10 – Налаштування віджету для Power

Наступним кроком у процесі налаштування інтерфейсу в системі моніторингу енергоспоживання є конфігурація віджету, призначеного для відображення значення напруги. Цей параметр є одним із ключових у будь-якій електротехнічній системі, оскільки дозволяє визначити стабільність та якість електроживлення. Для зручності ідентифікації віджет доцільно назвати  $V_{rms}$ , що відповідає позначенню діючого

(середньоквадратичного) значення напруги, широко вживаному в електроенергетиці та електроніці. В якості одиниці вимірювання встановлюється Volts (Вольти), що є загально визнаним стандартом для напруги у міжнародній системі одиниць. З метою забезпечення коректного масштабування та охоплення широкого діапазону можливих значень, мінімальне значення задається на рівні 0, а максимальне 500. Це дозволяє використовувати систему як у побутових, так і в промислових умовах, де рівень напруги може бути вищим за стандартні 220–240 В, зокрема для трифазних мереж або спеціалізованих електротехнічних установок. Тип даних double забезпечує відображення значень з високою точністю, що особливо важливо для виявлення навіть незначних відхилень від норми, які можуть свідчити про перевантаження, збої у мережі або несправності обладнання. Завдяки точному й інформативному налаштуванню цього віджету користувач отримує змогу в реальному часі відстежувати рівень напруги на об'єкті, оперативно реагувати на потенційно небезпечні коливання та забезпечити стабільну й безпечну роботу підключених пристроїв.

### Gauge Settings

TITLE (OPTIONAL)  
Vrms

Datastream  
Virtual Pin Datastream

General Expose to Automations

NAME Vrms ALIAS Vrms

PIN V0 DATA TYPE Double

UNITS Volt, V

MIN 0 MAX 500 DECIMALS ### DEFAULT VALUE 0

Enable history data

ADVANCED SETTINGS

Cancel Create

Рисунок 3.11 – Налаштування віджету для Vrms

Останнім штрихом в створенні інтерфейсу моніторингу споживання електрики є віджет, що візуалізує споживану потужність. Цей показник надзвичайно важливий для користувача, адже дає змогу оцінювати поточне споживання електроенергії та контролювати ефективність її використання. Віджет варто назвати KWs, що відображає потужність у кіловатах (Kilowatts) – загальноприйнятій одиниці вимірювання активної потужності в системах обліку електроенергії.

Для забезпечення точності та відповідного діапазону відображення, мінімальне значення визначається як 0, а максимальне 10. Це дозволяє ефективно контролювати споживання як в домашніх умовах, так і для невеликих комерційних об'єктів. Тип даних double, застосований у цьому віджеті, гарантує високу точність вимірювань, враховуючи дробові значення, що надзвичайно важливо при аналізі змін потужності та оптимізації споживання енергії.

Крім цього, коректно налаштований віджет KWs дає змогу виявляти пікові навантаження, прогнозувати витрати на електрику, а також оперативно реагувати на перевищення допустимого рівня споживання. Це сприяє зменшенню енергетичних витрат та підвищенню енергоефективності. В інтеграції з платформою Blynk, даний віджет надає користувачеві зручний та візуально зрозумілий інструмент для щоденного контролю енергоспоживання, що робить систему корисною як з точки зору інформативності, так і практично для оптимізації ресурсів.

TITLE (OPTIONAL)  
KWh

Datastream

Virtual Pin Datastream

**General** Expose to Automations

NAME: KWh1 ALIAS: KWh1

PIN: V4 DATA TYPE: Double

UNITS: Kilowatts, kW

MIN: 0 MAX: 10 DECIMALS: ### DEFAULT VALUE: Default Value

Enable history data

**ADVANCED SETTINGS**

Cancel Create

Рисунок 3.12 – Налаштування віджету для KWh

У підсумку ми можемо побачити таку панель на дашборді, яка відображає всі створені нами Gauge віджети(рис.3.13)

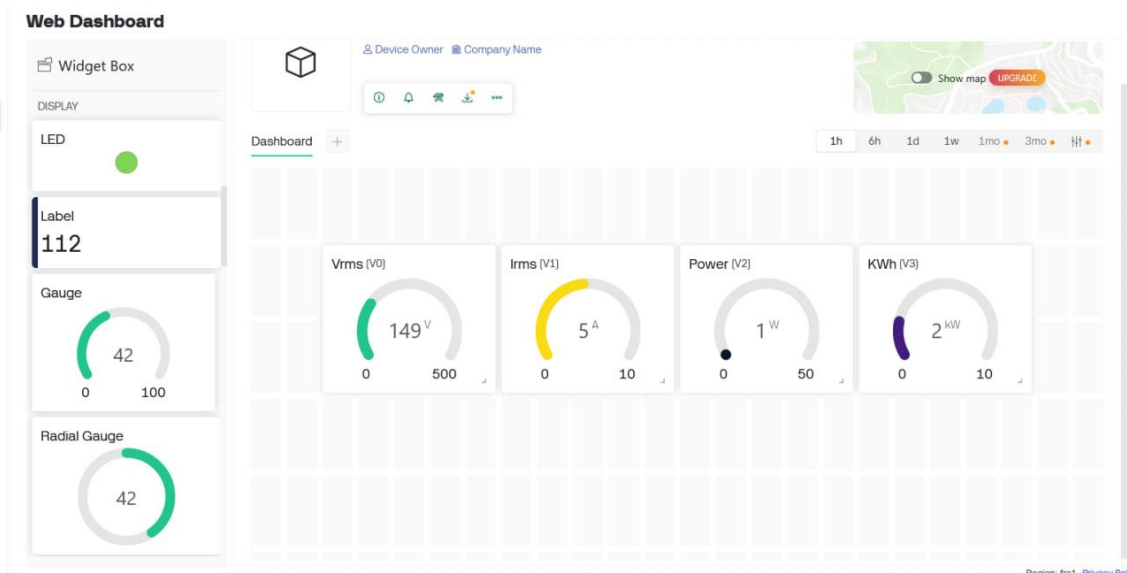


Рисунок 3.13 – фінальний вигляд панелі

Крім того, додамо інтерфейс з такою самою метою для нашого мобільного додатку. Так як є можливість підключення до одного акаунту як і веб версії, так і

мобільного застосунку, всі налаштування для віджетів збереглися в профілі і тепер для їх додавання необхідно просто обрати відповідне налаштування для кожного із доданих віджетів(рис. 3.14)

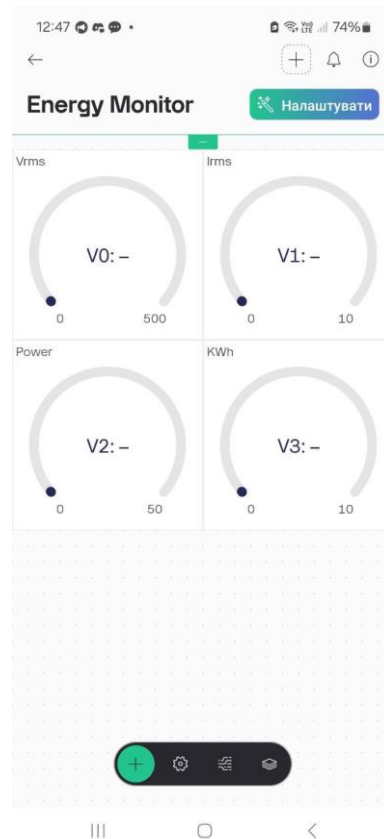


Рисунок 3.14 – вигляд інтерфейсу в мобільному додатку

Таким чином створений проект в Blink 2.0 дозволяє в реальному часі спостерігати за моніторингом показників електроенергії з підключеного до ESP32 пристрою.

### 3.3 Програмна реалізація програми керування ESP32

Для завантаження нашої програми до мікроконтролера ESP32 у рамках цього проекту була застосована інтегрована середовище розробки Arduino IDE. Ця платформа по праву вважається однією з найбільш популярних для програмування мікроконтролерів, що виділяється зрозумілим інтерфейсом, активною спільнотою

користувачів, великим вибором бібліотек та підтримкою різних плат. Arduino IDE забезпечує підтримку мікроконтролерів ESP32 через офіційний пакет, який легко встановити за допомогою менеджера плат. Важливою перевагою Arduino IDE є кросплатформенність (працює на Windows, Linux та macOS), а також можливість швидкого створення прототипів завдяки простому синтаксису, численним прикладам та автоматичній компіляції та завантаженню прошивки на пристрій.

Крім того, Arduino IDE надає зручні можливості керування бібліотеками через вбудований менеджер бібліотек, спрощує інтеграцію нових компонентів (сенсорів, комунікаційних модулів, дисплеїв та інше) та має корисні засоби налагодження, зокрема монітор порту для перегляду даних в реальному часі. З огляду на це, Arduino IDE є оптимальним рішенням для освітніх та дослідницьких проєктів, де важливим є досягнення швидкого результату з мінімальними затратами часу на налаштування.

Початкова частина програми включає підключення всіх необхідних бібліотек. Зокрема, це бібліотека EmonLib, яка використовується для вимірювання напруги, струму та обчислення активної потужності та енергії. Бібліотека EEPROM дозволяє зчитувати та зберігати дані у вбудованій пам'яті мікроконтролера. Підключаються бібліотеки WiFi і Blynk для бездротового підключення до інтернету та інтеграції з платформою Blynk. LiquidCrystal\_I2C використовується для керування LCD дисплеєм через інтерфейс I2C, а HTTPClient - для надсилання даних на зовнішній сервер через протокол HTTP.

На самому початку коду оголошуються необхідні константи. Це ідентифікатор шаблону Blynk, назва шаблону, облікові дані WiFi та токен авторизації Blynk. Далі ініціалізується об'єкт LiquidCrystal\_I2C, який дозволяє працювати з дисплеєм, що має адресу 0x27 та розмір 20 символів на 4 рядки. Також оголошуються коефіцієнти калібрування для напруги та струму - vCalibration і currCalibration відповідно. Ці значення підібрані для конкретного трансформатора струму (наприклад, типового SCT-013), і вони визначають точність вимірювання. Імплементацию цього коду можемо побачити нижче:

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>57 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
const float vCalibration = 41.5;
const float currCalibration = 0.15;
const char auth[] = "*****";
const char ssid[] = "*****";
const char pass[] = "*****",
```

Далі оголошується глобальний об'єкт типу EnergyMonitor під назвою emon. Цей об'єкт дозволяє використовувати функції бібліотеки EmonLib для вимірювання електричних параметрів. Оголошується також таймер BlynkTimer, який використовується для періодичного виклику функцій без блокування основного циклу loop. У змінній kWh зберігається значення спожитої енергії у кіловат-годинах, яка розраховується під час роботи пристрою. Змінна lastMillis використовується для обчислення часу між двома вимірюваннями, що необхідно для точного обчислення спожитої енергії. EEPROM адреса, в якій зберігається значення kWh, визначена як addrKWh = 12. Також визначені прототипи функцій та інших значень для зберігання даних: Імplementацію цього функціоналу можна побачити в коді нижче:

```
float kWh = 0.0;
unsigned long lastMillis = millis();
const int recVrms = 0;
const int recIrms = 4;
const int recPower = 8;
const int reckWh = 12;
void sendRecordedEnergyToBlynk();
void readRecordedEnergyFromEEPROM();
void saveRecordedEnergyToEEPROM();
```

Функція setup() виконується один раз при запуску пристрою. У ній спочатку ініціалізується серійний порт для налагодження через Serial.begin(115200). Далі

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>58 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

виконується підключення до Blynk сервера за допомогою функції Blynk.begin, яка використовує задані WiFi облікові дані. Відбувається ініціалізація LCD дисплея, включаючи підсвітку. EEPROM ініціалізується командою EEPROM.begin(32) з об'ємом у 32 байти. Значення енергії, накопичене у попередній сесії, зчитується з EEPROM за допомогою функції readEnergyDataFromEEPROM. Імплементацію цього функціоналу можна побачити нижче:

```
Serial.begin(115200);  
  Blynk.begin(auth, ssid, pass);  
  lcd.init(); lcd.backlight();  
  EEPROM.begin(32);  
  readRecordedEnergyFromEEPROM();
```

Далі налаштовуються входи для вимірювання: на GPIO 35 підключений трансформатор напруги, а на GPIO 34 — трансформатор струму. Параметри калібрування передаються функціям emon.voltage та emon.current. Таймер налаштовується так, щоб кожні 5 секунд викликалась функція sendEnergyDataToBlynk, яка відповідає за виконання вимірювань та передачу даних. Це можна побачити в наступному коді:

```
emon.voltage(35, vCalibration, 1.7);  
  emon.current(34, currCalibration);  
  timer.setInterval(5000L, sendRecordedEnergyToBlynk);  
  delay(1000);
```

У головному циклі loop() знаходиться стандартний виклик Blynk.run() для підтримки з'єднання з Blynk сервером, а також timer.run() для обробки таймерів.

```
void loop() {  
  Blynk.run();  
  timer.run();  
}
```

Функція `sendEnergyDataToBlynk()` виконує ключову логіку пристрою. Спочатку вона викликає метод `emon.calcVI(20, 2000)`, який виконує аналіз сигналів з аналогових входів і обчислює значення напруги, струму та потужності. Метод використовує 20 півперіодів змінного струму та частоту дискретизації 2000 точок.

Після цього виконується обчислення накопиченої енергії у кіловат-годинах за формулою: спожита потужність, помножена на час, що минув від попереднього вимірювання. Щоб перевести мілісекунди в години і отримати енергію в кВт·год, необхідно поділити на 3,600,000,000 (оскільки потужність вважається у Вт, і мілісекунди потрібно перевести в години).

Результати вимірювання виводяться в серіальний порт за допомогою `Serial.printf`. Далі відбувається збереження значення kWh у EEPROM за допомогою функції `saveEnergyDataToEEPROM`.

Наступним кроком є передача даних на платформу Blynk. Значення напруги, струму, потужності та енергії надсилаються на віртуальні піни V0–V3 відповідно. Користувач у мобільному застосунку Blynk може бачити відповідні дані у віджетах для їхнього перегляду у реальному часі.

Після цього формується HTTP-запит. Використовується клієнт `HTTPClient`, який встановлює з'єднання із сервером за адресою `http://localhost:5000/data`. Заголовок `Content-Type` встановлюється як `application/json`. Далі формується JSON-повідомлення, яке містить значення `vrms`, `irms`, `power` і `kwh`. Відправлення відбувається за допомогою методу `POST`, після чого перевіряється код відповіді. Якщо відповідь є успішною (наприклад, код 200 або 201), виводиться відповідне повідомлення в серійний порт. У разі помилки також виводиться повідомлення з поясненням.

На завершення функції, виміряні значення відображаються на LCD дисплеї. Кожен рядок містить певне значення: напруга, струм, потужність та загальна енергія. Це дозволяє користувачеві бачити результати вимірювань навіть без підключення до мобільного застосунку чи сервера.

Функція `readEnergyDataFromEEPROM()` зчитує збережене значення спожитої енергії. Якщо зчитане значення є некоректним (наприклад, NaN), воно скидається до нуля, і одразу викликається функція `saveEnergyDataToEEPROM()` для збереження коректного значення. Її код наведено в наступному фрагменті:

```
void readRecordedEnergyFromEEPROM() {
    EEPROM.get(reckWh, kWh);
    if (isnan(kWh)) {
        kWh = 0.0;
        saveRecordedEnergyToEEPROM();
    }
}
```

Функція `saveEnergyDataToEEPROM()` використовує метод `EEPROM.put` для запису значення `kWh` у пам'ять. Після запису необхідно викликати `EEPROM.commit()`, щоб зміни були збережені у флеш-пам'яті. Її код наведено нижче:

```
void saveRecordedEnergyToEEPROM() {
    EEPROM.put(reckWh, kWh);
    EEPROM.commit();
}
```

Загальний алгоритм програми(рис.3.15) розпочинається із запуску та базового налаштування всіх потрібних фізичних складових та налагодження зв'язків, що забезпечують її працездатність. Далі програма переходить до безперервного циклу, в якому вона регулярно виконує набір операцій. Наприклад, відбувається взаємодія з хмарною платформою для інтернету речей, що дає змогу віддалено управляти пристроями та отримувати інформацію, а також активується внутрішній таймер для синхронізації дій. Потім система проводить вимірювання енергетичних параметрів, оновлює нагромаджені дані про спожиті кіловат-години та надійно зберігає ці дані в спеціальній енергонезалежній пам'яті. Після цього

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>61 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

робиться спроба відправити зібрану інформацію. У випадку успішної відправки дані відображаються на підключеному дисплеї; якщо виникає помилка під час відправлення, система це фіксує, але потім повертається на початок циклу для подальшого моніторингу та обробки даних.

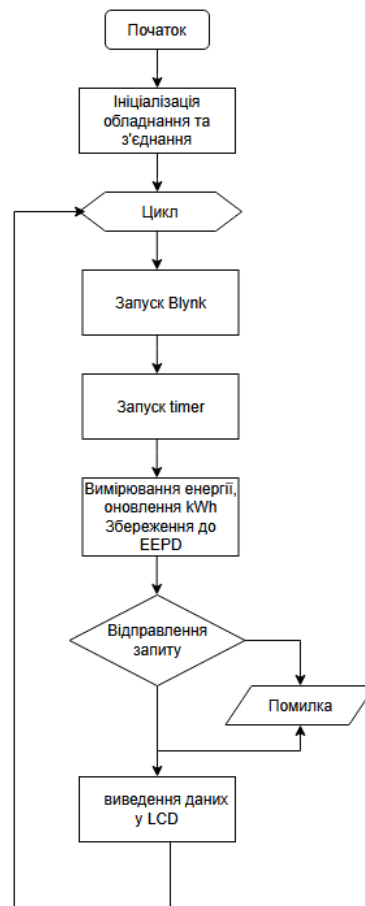


Рисунок 3.15 – блок-схема головної програми

### 3.4. Висновки до третього розділу

У третьому розділі було повністю реалізовано програмне забезпечення програмно-технічного комплексу, розробленого для вимірювання і моніторингу параметрів електроспоживання на основі технології IoT. В якості ядра було використано мікроконтролер ESP32 та платформу Blynk 2.0. Спочатку було підготовлено серверне середовище і базу даних, що забезпечують збереження та подальший аналіз зібраних показників електроспоживання. Після цього було

розгорнуто і налаштовано платформу Blynk 2.0, котра надає інтуїтивно зрозумілий інтерфейс для візуалізації даних в режимі реального часу, а також можливість дистанційного управління пристроєм. Особливу увагу було приділено програмуванню мікроконтролера ESP32, який забезпечує зчитування електричних параметрів (напруги, сили струму, потужності і т.д.) з допомогою підключених сенсорів, обробку даних і їх відправку на хмарну платформу. Програмне забезпечення було розроблено з врахуванням вимог до точності, стабільності і безперебійної роботи пристрою. Підсумовуючи, реалізація програмної частини підтвердила ефективність обраної архітектури системи. Спільна робота апаратної частини на базі ESP32, платформи Blynk 2.0 та серверної інфраструктури гарантує надійний збір, передачу та візуалізацію даних про електроспоживання. Це є фундаментом для подальшого аналізу, оптимізації споживання електроенергії та підвищення загальної енергоефективності.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>63 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

## ВИСНОВКИ

У цій роботі, спираючись на результати виконаних теоретичних та практичних досліджень, було розроблено програмно-технічний засіб для вимірювання та моніторингу параметрів електроспоживання. В основу покладено технологію Інтернету речей (IoT) з використанням мікроконтролера ESP32 та платформи Blynk 2.0. Отримані результати підтверджують ефективність запропонованого рішення для автоматизованого збору, передачі та аналізу даних про споживання енергії в режимі реального часу, що дозволяє підвищити енергоефективність та оптимізувати витрати.

У першому розділі було представлено концепцію Інтернету речей (IoT) як ключової технології для створення сучасних систем моніторингу енергоспоживання. Здійснено аналіз наявних рішень, технічних можливостей мікроконтролера ESP32 та функціоналу платформи Blynk 2.0. Це дало змогу сформулювати загальну задачу створення ефективного та точного пристрою для енергомоніторингу з можливістю інтеграції з хмарними сервісами та інтерфейсами візуалізації.

У другому розділі визначено функціональні та технічні вимоги до майбутнього пристрою, обґрунтовано вибір апаратних компонентів та сенсорів. Крім того, було спроектовано архітектуру програмного забезпечення та електричні схеми підключення. Особливу увагу приділено забезпеченню точності вимірювань, стабільності функціонування пристрою та ефективній взаємодії між складовими системи. Запропонована структура передбачає безперервний збір даних, їх локальну обробку та подальшу передачу на платформу візуалізації.

У третьому розділі було повністю реалізовано програмне забезпечення системи. Зокрема, підготовлено серверне середовище та базу даних, налаштовано платформу Blynk 2.0, що забезпечує візуалізацію даних у реальному часі, а також запрограмовано мікроконтролер ESP32 для зчитування та обробки даних із

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>64 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

сенсорів. Програмна реалізація підтвердила ефективність обраної архітектури та можливість стабільної роботи пристрою в умовах практичного застосування.

Таким чином, розроблений пристрій є сучасним, надійним та масштабованим рішенням для моніторингу енергоспоживання. Його використання забезпечує точний контроль за споживанням електроенергії, надає зручний інтерфейс для кінцевого користувача та відкриває можливості для інтеграції з іншими елементами розумної інфраструктури, як у побутових, так і промислових умовах.

|     |      |          |        |      |                          |      |
|-----|------|----------|--------|------|--------------------------|------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                          | 65   |

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Appinventiv. IoT Energy Management – 10 Powerful Applications & Advantages. URL: <https://appinventiv.com/blog/iot-in-energy-management/> (дата звернення: 19.05.2025).
2. Internet of Things: a comprehensive overview, architectures, technologies, and applications. *Discover Internet of Things*. URL: <https://link.springer.com/article/10.1007/s43926-024-00084-3> (дата звернення: 18.05.2025)
3. El-Khozondar H. J. et al. A smart energy monitoring system using ESP32 microcontroller . *Advances in Electrical Engineering, Electronics and Energy*. 2024. Vol. 9. Article number: 100666.
4. Петросян А.Р., Петросян Р.В., Колос К.Р. Розробка платформи віддаленого управління інфраструктурою Інтернет речей. *Технічні науки та технології*, 2021, № 1(87), С. 73-80. URL: [https://www.researchgate.net/publication/354184727\\_Rozrobka\\_platforni\\_viddalenogo\\_upravlinna\\_infrastrukturou\\_Internet\\_recej](https://www.researchgate.net/publication/354184727_Rozrobka_platforni_viddalenogo_upravlinna_infrastrukturou_Internet_recej) (дата звернення: 17.05.2025).
5. Domínguez-Bolaño T., Alayón F., Hernández-Alemán J. A., Pérez-Martín J., Brito-Sanjuan A. An overview of IoT architectures, technologies, and existing open-source projects. *Internet of Things*. 2022. Vol. 20. P. 100626.
6. Haider, M. U., et al. Smart energy meters in renewable-energy-based power networks: An extensive review *Engineering Proceedings*. 2022. Vol. 20, No. 1. P. 23.
7. Tanasiev V., et al. Enhancing environmental and energy monitoring of residential buildings through IoT. *Automation in Construction* : journal. Vol. 126. 2021. Article ID 103662.
8. HRMARS. IOT-Based Cost-Effective Energy Meter Monitoring System for Smart Home Applications: Design, Implementation, and Analysis. 2024. URL: [https://hrmars.com/papers\\_submitted/24398/iot-based-cost-effective-energy-meter-monitoring-system-for-smart-home-applications-design-implementation-and-analysis.pdf](https://hrmars.com/papers_submitted/24398/iot-based-cost-effective-energy-meter-monitoring-system-for-smart-home-applications-design-implementation-and-analysis.pdf) (дата звернення: 19.05.2025).

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>66 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

9. Shelly EM – Wi-Fi-operated energy meter. URL: <https://kb.shelly.cloud/knowledge-base/shelly-em> (дата звернення: 26.05.2025).
10. SONOFF POW Elite – Power Meter Switch. URL: <https://sonoff.tech/product/diy-smart-switches/pow-elite/> (дата звернення: 18.05.2025).
11. Iammeter – Wi-Fi Three Phase Energy Meter. URL: <https://www.iammeter.com/products/three-phase-meter> (дата звернення: 19.05.2025).
12. Smart WiFi Switch Energy Monitor ZMAi-90: User Manual. URL: <https://henractech.co.za/wp-content/uploads/2022/03/Smart-Wifi-Switch-Energy-Monitor-ZMAi-90-user-manual.pdf> (дата звернення: 17.05.2025).
13. ESP32 – Wi-Fi & Bluetooth SoC. URL: <https://www.espressif.com/en/products/socs/esp32> (дата звернення: 17.05.2025).
14. Посібник з мікроконтролера ESP32. URL: <https://ua.ariat-tech.com/blog/ESP32-Microcontroller-Guide.html> (дата звернення: 18.05.2025).
15. SCT-013 – датчик змінного струму. URL: <https://justdoelectronics.com/sct-013-sensor/> (дата звернення: 19.05.2025).
16. SCT-013-000 Current Transducer/Sensor. URL: [https://www.researchgate.net/figure/SCT-013-000-Current-Transducer-Sensor-Some-other-single-input-channeled-Hall-effect\\_fig4\\_317305006](https://www.researchgate.net/figure/SCT-013-000-Current-Transducer-Sensor-Some-other-single-input-channeled-Hall-effect_fig4_317305006) (дата звернення: 19.05.2025).
17. ZMPT101B AC Voltage Transformer Sensor Module. URL: <https://datacapturecontrol.com/articles/io-components/sensors/voltage/zmpt101b-ac-voltage-transformer-sensor-module> (дата звернення: 26.05.2025).
18. Датчик напруги ZMPT101B. URL: <https://itmaster.biz.ua/directory/sensors/zmpt101b.html> (дата звернення: 16.05.2025).
19. Voltage Sensor. URL: <https://www.edn.com/voltage-sensor/> (дата звернення: 17.05.2025).
20. 20x4 I2C LCD Panel. URL: <https://docs.circuitdesigner.com/component/3f29b96d-1be1-4030-98de-a9e587eda8b7/20x4-i2c-lcd-panel> (дата звернення: 19.05.2025).

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>67 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

21. I2C Serial Interface 20x4 LCD Module. URL: [https://www.handsontec.com/dataspecs/I2C\\_2004\\_LCD.pdf](https://www.handsontec.com/dataspecs/I2C_2004_LCD.pdf) (дата звернення: 16.05.2025).

22. 20x4 I2C LCD Display setup. URL: <https://forum.arduino.cc/t/20x4-i2c-lcd-display-setup/475117> (дата звернення: 26.05.2025).

23. Blynk Documentation. URL: <https://docs.blynk.io/en> (дата звернення: 18.05.2025).

24. Документація Blynk. URL: <https://shoorik007.github.io/> (дата звернення: 17.05.2025).

25. What is Blynk 2.0. URL: <https://github.com/blynkkk/docs/blob/main/What-is-Blynk-2.0.md> (дата звернення: 19.05.2025).

26. Firebase Documentation. URL: <https://firebase.google.com/docs> (дата звернення: 26.05.2025).

27. What is Firebase? The complete story, abridged. URL: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0> (дата звернення: 17.05.2025).

28. What is Google Firebase? URL: <https://www.techtarget.com/searchmobilecomputing/definition/Google-Firebase> (дата звернення: 18.05.2025).

29. InfluxDB Documentation. URL: <https://docs.influxdata.com/> (дата звернення: 19.05.2025).

30. InfluxDB data sourceGrafana documentation. URL: <https://grafana.com/docs/grafana/latest/datasources/influxdb/> (дата звернення: 19.05.2025).

31. InfluxDB OSS v2 Documentation. URL: <https://docs.influxdata.com/influxdb/v2/> (дата звернення: 19.05.2025).

32. Grafana Documentation. URL: <https://grafana.com/docs/grafana/latest/introduction/> (дата звернення: 19.05.2025).

33. What is Grafana. URL: <https://www.redhat.com/en/topics/data-services/what-is-grafana> (дата звернення: 18.05.2025).

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>68 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

34. Microsoft Power BI. URL: <https://www.techtarget.com/searchcontentmanagement/definition/Microsoft-Power-BI> (дата звернення: 19.05.2025).

35. What is Power BI. URL: <https://www.coursera.org/articles/what-is-power-bi> (дата звернення: 20.05.2025).

36. What is Power BI Tutorial. URL: <https://www.simplilearn.com/tutorials/power-bi-tutorial/what-is-power-bi> (дата звернення: 19.05.2025).

37. About SQLite. URL: <https://www.sqlite.org/about.html> (дата звернення: 21.05.2025).

38. What is SQLite. URL: <https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite> (дата звернення: 19.05.2025).

39. Pawlaszczyk, Dirk. "SQLite." Mobile Forensics—The File Format Handbook: Common File Formats and File Systems Used in Mobile Devices. Cham: Springer International Publishing, 2022. 129-155.

40. Configuring Database Connections. URL: <https://www.jetbrains.com/help/pycharm/configuring-database-connections.html#connection-modes> (дата звернення: 20.05.2025).

41. Create and Modify Dialogs. URL: <https://www.jetbrains.com/help/pycharm/create-and-modify-dialogs.html> (дата звернення: 19.05.2025).

42. Flask Documentation. URL: <https://flask.palletsprojects.com/en/stable/> (дата звернення: 20.05.2025).

43. Flask. URL: <https://flask.palletsprojects.com/en/stable/> (дата звернення: 26.05.2025).

44. Flask Tutorial. URL: <https://www.geeksforgeeks.org/flask-tutorial/> (дата звернення: 20.05.2025).

45. Flask on PyPI. URL: <https://pypi.org/project/Flask/> (дата звернення: 20.05.2025).

46. Set up Web Dashboard. URL: <https://docs.blynk.io/en/getting-started/template-quick-setup/set-up-web-dashboard> (дата звернення: 20.05.2025).

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>69 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

47. Step-by-step Video Guide to Making Your First Project on the New Blynk.  
URL: <https://blynk.io/blog/step-by-step-video-guide-to-making-your-first-project-on-the-new-blynk> (дата звернення: 19.05.2025).

48. Beginner Tutorial: IoT Dashboards with Blynk. URL:  
[https://www.reddit.com/r/raspberryDIY/comments/15v6myd/beginner\\_tutorial\\_iot\\_dashboards\\_with\\_blynk/](https://www.reddit.com/r/raspberryDIY/comments/15v6myd/beginner_tutorial_iot_dashboards_with_blynk/) (дата звернення: 21.05.2025).

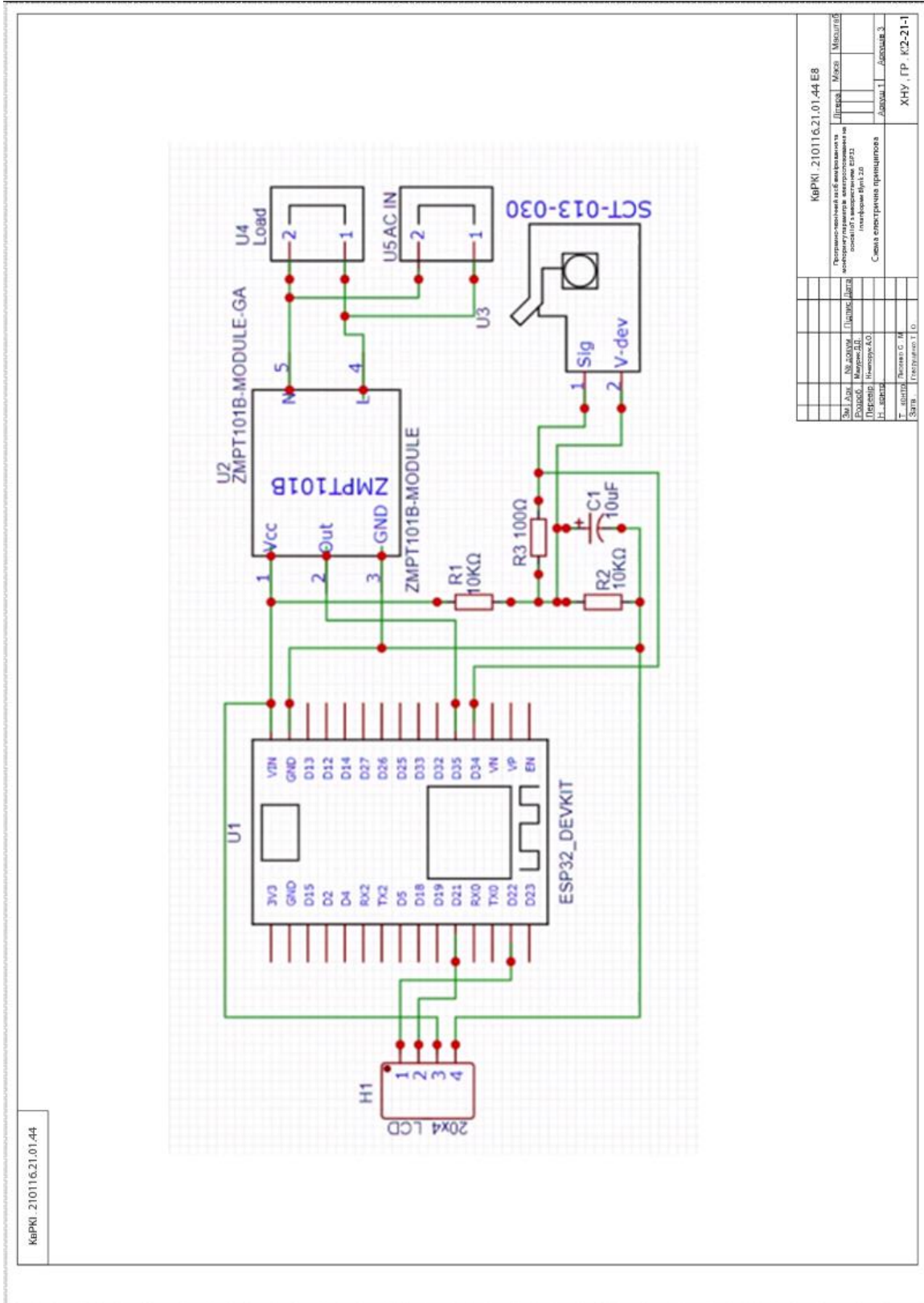
49. Koohang, Alex, et al. "Internet of Things (IoT): From awareness to continued use." *International Journal of Information Management* 62 (2022): 102442.

50. Zikria, Yousaf Bin, et al. "Next-generation internet of things (iot): Opportunities, challenges, and solutions." *Sensors* 21.4 (2021): 1174.

|     |      |          |        |      |                          |            |
|-----|------|----------|--------|------|--------------------------|------------|
|     |      |          |        |      | КВРКІ.210116.21.01.44 ПЗ | Арк.<br>70 |
| Зм. | Арк. | № докум. | Підпис | Дата |                          |            |

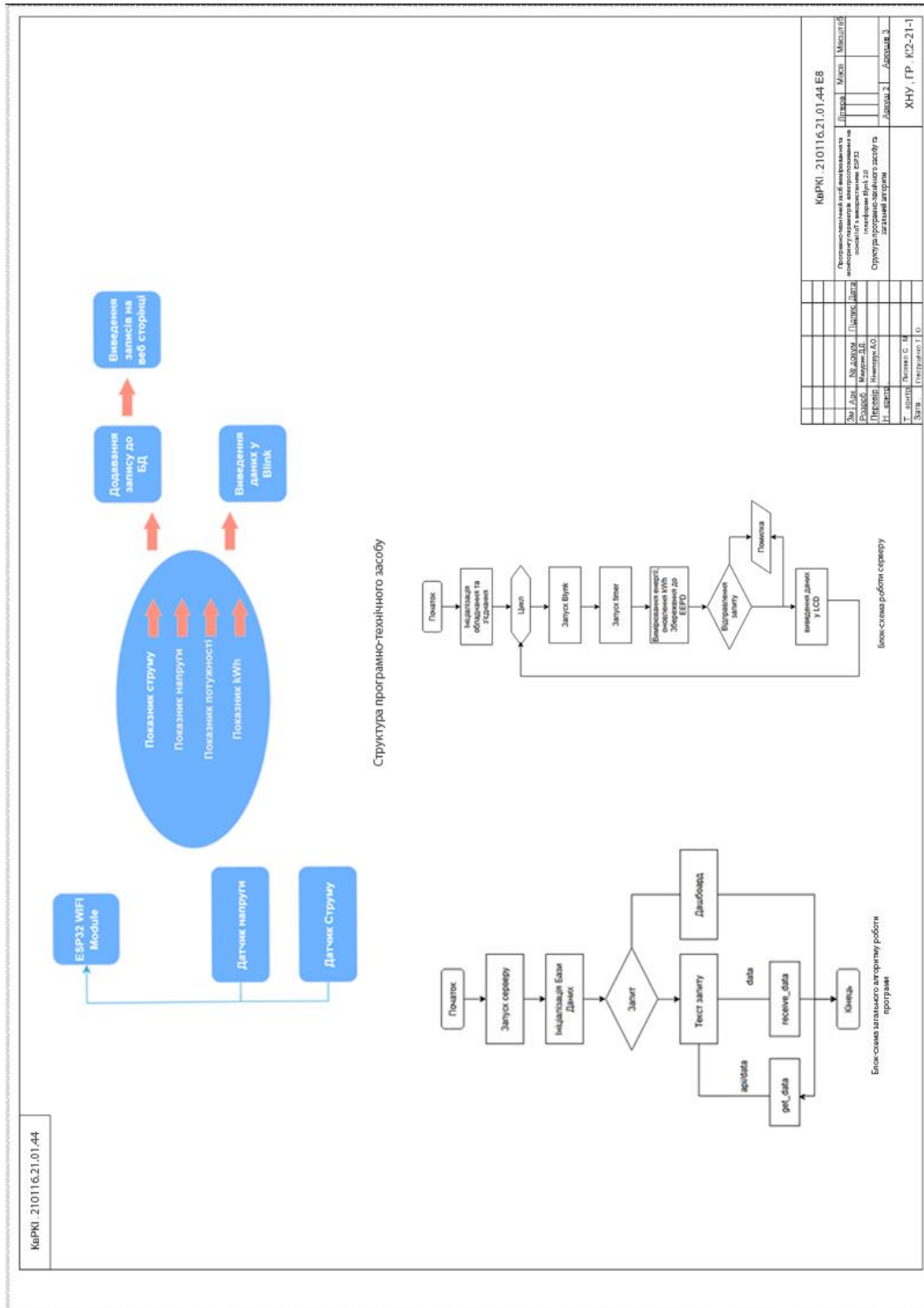
# Додаток А (обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «СХЕМА ЕЛЕКТРИЧНА ПРИНЦИПОВА»



## Додаток Б (обов'язковий)

# КОПІЯ КРЕСЛЕННЯ «СТРУКТУРА ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ТА ЗАГАЛЬНИЙ АЛГОРИТМ»





**Додаток Г**  
**(обов'язковий)**

**КОПІЯ КОДУ «DASHBOARD.HTML»**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>IoT Energy Dashboard</title>
  <style>
    body {
      font-family: sans-serif;
      padding: 20px;
      background: #f7f7f7;
    }
    h1 {
      text-align: center;
    }
    table {
      margin: auto;
      border-collapse: collapse;
      width: 90%;
      background-color: white;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    th, td {
      padding: 10px;
      border: 1px solid #ccc;
      text-align: center;
    }
    th {
      background-color: #4CAF50;
      color: white;
    }
  </style>
</head>
<body>
  <h1>IoT Energy Dashboard</h1>
  <table border="1">
    <thead>
      <tr>
        <th>Device ID</th>
        <th>Power Consumption (W)</th>
        <th>Temperature (°C)</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>123456789</td>
        <td>150</td>
        <td>25.5</td>
      </tr>
      <tr>
        <td>987654321</td>
        <td>200</td>
        <td>30.2</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

```

    </style>
</head>
<body>
  <h1>IoT Energy Meter Dashboard</h1>
  <table id="data-table">
    <thead>
      <tr>
        <th>Timestamp</th>
        <th>Vrms (V)</th>
        <th>Irms (A)</th>
        <th>Power (W)</th>
        <th>kWh</th>
      </tr>
    </thead>
    <tbody></tbody>
  </table>
  <script>
    function formatTimestamp(isoString) {
      const date = new Date(isoString);
      const options = {
        year: 'numeric',
        month: 'short',
        day: '2-digit',
        hour: '2-digit',
        minute: '2-digit',
        second: '2-digit',
      };
      return date.toLocaleString(undefined, options);
    }
    async function fetchData() {
      const res = await fetch('/api/data');
      const data = await res.json();
      const tbody = document.querySelector('#data-table tbody');
      tbody.innerHTML = '';
    }
  </script>

```

```
data.forEach(row => {
  const tr = document.createElement('tr');
  tr.innerHTML = `
    <td>${formatTimestamp(row.timestamp)}</td>
    <td>${row.vrms.toFixed(2)}</td>
    <td>${row.irms.toFixed(4)}</td>
    <td>${row.power.toFixed(4)}</td>
    <td>${row.kwh.toFixed(5)}</td>
  `;
  tbody.appendChild(tr);
});
}

fetchData();
setInterval(fetchData, 5000);
</script>
</body>
</html>
```

**Додаток Д**  
**(обов'язковий)**

**КОПІЯ КОДУ ОСНОВНОЇ ФУНКЦІЇ**

```
void sendRecordedEnergyToBlynk() {
    emon.calcVI(20, 2000);
    unsigned long currentMillis = millis();
    kWh += emon.apparentPower * (currentMillis - lastMillis) /
3600000000.0;
    lastMillis = currentMillis;
    Serial.printf("Vrms: %.2fV\tIrms: %.4fA\tPower: %.4fW\tkWh:
%.5fkWh\n",
                emon.Vrms, emon.Irms, emon.apparentPower, kWh);
    saveEnergyDataToEEPROM();
    Blynk.virtualWrite(V0, emon.Vrms);
    Blynk.virtualWrite(V1, emon.Irms);
    Blynk.virtualWrite(V2, emon.apparentPower);
    Blynk.virtualWrite(V3, kWh);
    HTTPClient http;
    http.begin("http://localhost:5000/data");
    http.addHeader("Content-Type", "application/json");
    String jsonPayload = "{";
    jsonPayload += "\"vrms\":" + String(emon.Vrms, 2) + ",";
    jsonPayload += "\"irms\":" + String(emon.Irms, 4) + ",";
    jsonPayload += "\"power\":" + String(emon.apparentPower, 4) +
",,";
    jsonPayload += "\"kwh\":" + String(kWh, 5);
    jsonPayload += "}";
    int httpResponseCode = http.POST(jsonPayload);
    if (httpResponseCode > 0) {
        Serial.printf("HTTP Response code: %d\n",
httpResponseCode);
    } else {
```

```
        Serial.printf("HTTPRequestfailed:%s\n",
http.errorToString(httpResponseCode).c_str());
    }

    http.end();

    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("Vrms: ");
    lcd.print(emon.Vrms, 2); lcd.print(" V");
    lcd.setCursor(0, 1); lcd.print("Irms: ");
    lcd.print(emon.Irms, 4); lcd.print(" A");
    lcd.setCursor(0, 2); lcd.print("Power: ");
    lcd.print(emon.apparentPower, 4); lcd.print(" W");
    lcd.setCursor(0, 3); lcd.print("kWh: "); lcd.print(kWh, 5);
    lcd.print(" kWh");
}
```

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Денис МАЗУРИК

**Співавтор:**

**Назва:** Мазурик Програмно-технічний засіб вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 3.9%

**Коефіцієнт подібності 2:** 1.3%

**Мікропробіли:** 28

**Заміна букв:** 1

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-05-28 10:01:23.0

**Після аналізу Звіту подібності констатую наступне:**

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-05-28

Дата

Доцент Андрій Нічепорук

експерт

Wed May 28 09:49:47 EEST 2025, Мелзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism v-15.274 Educational

**The maximum coincidence with one document 5.0%**

Dictionary check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 11%

|   |         |                 |            |                                      |         |
|---|---------|-----------------|------------|--------------------------------------|---------|
| <p>ID: 242247<br/>         Title: БКР Програмно-технічний засіб<br/>         вимірювання та моніторингу параметрів<br/>         електроспоживання на основі IoT з<br/>         використанням ESP32 і платформи Блуток 2.0<br/>         Added in a DV: 2025-05-28<br/>         Authors: Денис МАЗУРІК<br/>         Heads: Андрій ПІЧЕПОРУК<br/>         Consultants:<br/>         Opponents:</p> |         | <p>Document</p> |            | <p>Sum coincidence on<br/>the DV</p> |         |
| Symbols   | Lexemes | Symbols         | Lexemes    | Symbols                              | Lexemes |
| 91882   | 667     | 5770<br>(6%)    | 46<br>(7%) |                                      |         |

### Plagiarism sources

| ID | Description | Plagiarism presence in the document |         |
|----|-------------|-------------------------------------|---------|
|    |             | Symbols                             | Lexemes |
|    |             |                                     |         |

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Мазурик Денис Дмитович

Тема: Програмно-технічний засіб вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень  3  Кількість сторінок записки  79

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є створення програмно-технічного засобу вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. У першому розділі кваліфікаційної роботи представлено розбір тематичної сфери, що охоплює сучасні підходи до вимірювання та моніторингу споживання електроенергії на базі технологій Інтернету речей (IoT). Досліджено концепцію IoT, її структуру, позитивні та негативні сторони, а також здійснено огляд існуючих апаратних і програмних рішень у цій галузі. Проаналізовано ключові розробки, які застосовують мікроконтролери, датчики та хмарні сервіси для збору та обробки енергетичних даних. На основі здобутих результатів сформульовано основні вимоги до пристрою, який розробляється, та обґрунтовано вибір компонентів. У другому розділі виконано проектування програмно-технічного засобу вимірювання та моніторингу споживання електроенергії з використанням ESP32 та платформи Blynk 2.0. Створено загальну структурну модель пристрою, розроблено електричну принципову та монтажну схеми. Розроблено концептуальну модель бази даних для зберігання даних про споживання електроенергії. Здійснено детальний аналіз обраних апаратних компонентів, зокрема мікроконтролера ESP32, сенсорів струму, модуля живлення тощо. Розглянуто програмні рішення, зокрема можливості платформи Blynk 2.0 для візуалізації та віддаленого управління. Вивчено перспективи подальшого

розвитку системи, включаючи інтеграцію з хмарними базами даних (Firebase, InfluxDB), інструментами для візуалізації (Grafana, Power BI), а також розширення рішення на декілька зон або пристроїв. У третьому розділі здійснено безпосередню програмну реалізацію розробленого пристрою. Здійснено підготовку серверного середовища та розгортання бази даних для збереження та обробки даних про споживання електроенергії. Налаштовано платформу Blynk 2.0, створено інтерфейси користувача для перегляду даних у реальному часі та управління пристроєм. Реалізовано програмне забезпечення для ESP32, що забезпечує зчитування параметрів, їх обробку та передачу до хмарного сервісу. Забезпечено інтеграцію апаратної частини з мобільним додатком та серверною інфраструктурою, що дозволило досягти функціонального завершення системи та продемонструвати її працездатність у тестовому середовищі.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага до розширення можливостей схеми.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

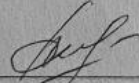
8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Корсунська Л.О., заступник кафедри АКТІТ та Р, ХНУ

"2" травня 2025 р.

 (підпис)

Завідувачу кафедри КПС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Дениса МАЗУРИКА

ГПБ здобувача вищої освіти


ФІТ, 4 курсу, групи КІ2-21-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

 2025 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА  
ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-технічний засіб вимірювання та моніторингу параметрів електроспоживання на основі IoT з використанням ESP32 і платформи Blynk 2.0

Автор Денис МАЗУРИК

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський) рівень

Спеціальність 123– Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент, Андрій Нічепорук

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

| №   | Висновок  | Позначка про відповідність |
|-----|---|----------------------------|
| 1   | Ознаки академічного плагіату  |                            |
| 1.1 | Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.   | Відповідає                 |
| 1.2 | Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.   |                            |
| 1.3 | Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат. |                            |
| 1.4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.  |                            |
| 2   | Інші види порушень академічної доброчесності  | Не виявлено                |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 3.86% і адресується до 14 першоджерела; та системою Anti-Plagiarism складає 5%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



Андрій Нічепорук

Гарант ОП

Андрій Нічепорук

Завідувач кафедри КІС

Ольга Павлова