

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ ТА ВІЗУАЛІЗАЦІЇ
ВИТРАТ ЕЛЕКТРОЕНЕРГІЇ НА ЖИТЛОВИХ ОБ'ЄКТАХ»**

на здобуття освітнього ступеня бакалавр

за спеціальності 126 Інформаційні системи та технології

(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології

(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Андрій ШЕЛЕСТ

(ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти гр.ІСД-42

Андрій ШЕЛЕСТ

(ім'я, ПРІЗВИЩЕ)

Керівник:

PhD

Віктор САГАЙДАК

(ім'я, ПРІЗВИЩЕ)

Рецензент:

(ім'я, ПРІЗВИЩЕ)

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

_____ Каміла СТОРЧАК

“ ____ ” _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Шелеста Андрія Олександровича

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Інформаційна система моніторингу та візуалізації витрат електроенергії на житлових об'єктах

керівник кваліфікаційної роботи Віктор САГАЙДАК, PhD

(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20” лютого 2026 р. № 51

2. Строк подання кваліфікаційної роботи “01” червня 2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Енергетична система будівель;
2. Програмно-інструментальне забезпечення;
3. Інформаційної системи моніторингу електроспоживання;
4. Науково-технічна література.

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

1. Огляд інфраструктур інтернету речей для енергетичних систем будівель.
2. Аналіз вимог і програмно-інструментальне забезпечення IoT-системи.
3. Реалізація та тестування інформаційної системи моніторингу електроспоживання.

5. Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання “20” лютого 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	21.02.26-03.03.26	
2.	Огляд інфраструктур інтернету речей для енергетичних систем будівель	05.03.26-18.03.26	
3.	Аналіз вимог і програмно-інструментальне забезпечення IoT-системи	18.03.26-30.03.26	
4.	Реалізація та тестування інформаційної системи моніторингу електроспоживання.	02.04.26-25.04.26	
5.	Оцінка результатів роботи, розробка висновків.	25.04.26-18.05.26	
6.	Формування демонстраційних матеріалів та доповіді	19.05.26-22.05.26	
7.	Оформлення роботи	28.05.26-30.05.26	

Здобувач вищої освіти

(підпис)

Андрій ШЕЛЕСТ

(ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи

(підпис)

Віктор САГАЙДАК

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступня бакалавр: 72 стор., 46 рис., 4 табл., 36 джерел.

Мета роботи - реалізація інформаційної IoT-системи моніторингу електроспоживання та візуалізації даних у режимі реального часу.

Об'єкт дослідження – процес моніторингу та контролю електроспоживання в електричних мережах із використанням технологій IoT.

Предмет дослідження – методи та засоби побудови IoT-системи для вимірювання електричних параметрів.

Короткий зміст роботи. У першому розділі досліджено сучасні підходи до використання Інтернету речей в енергетичних системах будівель, зокрема архітектуру IoT, типи датчиків, методи їх живлення та способи збору й обробки даних. Також розглянуто методи прогнозного й адаптивного керування енергоспоживанням та основні мережеві технології, що забезпечують взаємодію компонентів IoT-систем.

У другому розділі досліджено апаратні та програмні компоненти, необхідні для побудови IoT-системи інтелектуального обліку електроенергії. Також проаналізовано середовище розробки Arduino IDE та сучасні платформи для створення мобільних застосунків у системах Інтернету речей.

У третьому розділі досліджено процес проєктування, реалізації та тестування IoT-системи моніторингу електроспоживання на базі ESP32 та платформи Blynk 2.0. Також розглянуто особливості підключення апаратних компонентів, налаштування програмного середовища та візуалізації даних у реальному часі.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ, МОНІТОРИНГ, КОНТРОЛЬ, СИСТЕМА, ПРИСТРІЙ, МЕТОД, ІНТЕГРОВАНЕ СЕРЕДОВИЩЕ, ДАТЧИК, АПАРАТНЕ ЗАБЕЗМЕЧЕННЯ, МОДУЛЬ.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ОГЛЯД ІНФРАСТРУКТУР ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ ЕНЕРГЕТИЧНИХ СИСТЕМ БУДІВЕЛЬ.....	10
1.1 Загальні принципи та розвиток Інтернету речей і його застосування в енергетичних системах будівель.....	10
1.2 Класифікація датчиків та методи їх енергозабезпечення.....	15
1.3 Методи прогнозного та адаптивного керування в IoT-системах енергоменеджменту.....	19
1.4 Мережеві архітектури та протоколи зв'язку в системах IoT.....	23
1.5 Моніторинг та керування енергоспоживанням у IoT-системах будівель...	26
РОЗДІЛ 2 АНАЛІЗ ВИМОГ І ПРОГРАМНО-ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ ІОТ-СИСТЕМИ.....	37
2.1 Аналіз технічної специфікації та вибір складових проекту.....	37
2.2 Інтегроване середовище розробки Arduino IDE.....	45
2.3 Програмні інструменти для створення мобільних застосунків в IoT- системах.....	51
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ЕЛЕКТРОСПОЖИВАННЯ.....	63
3.1 Проєктування електричної схеми та налаштування апаратного забезпечення.....	63
3.2 Інтеграція та налаштування користувацького інтерфейсу в Blynk 2.0.....	67
3.3 Підготовка програмного середовища та калібрування датчика напруги....	71
3.4 Оцінка працездатності системи збору та візуалізації даних електроспоживання.....	75
ВИСНОВКИ.....	78
ПЕРЕЛІК ПОСИЛАНЬ.....	80
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	84

ВСТУП

Актуальність теми. Сучасний етап розвитку інформаційних технологій, зокрема концепції Інтернету речей, характеризується інтенсивним впровадженням інтелектуальних систем обліку та управління енергоспоживанням у різних сферах господарської діяльності. Інтеграція апаратних і програмних засобів для автоматизованого збору, обробки та передачі даних забезпечує підвищення ефективності використання електроенергії, сприяє оптимізації витрат і створює умови для постійного моніторингу параметрів енергоспоживання в режимі реального часу. Крім того, застосування таких підходів відкриває можливості для аналітичної обробки отриманих даних, що є основою для прийняття обґрунтованих управлінських рішень.

Водночас традиційні методи обліку електроенергії, що передбачають періодичне ручне зняття показників, поступово втрачають свою ефективність через значну трудомісткість, низьку оперативність та відсутність можливості безперервного контролю. Це зумовлює потребу у впровадженні більш сучасних, автоматизованих рішень, здатних забезпечити точність, надійність і зручність доступу до інформації.

Отже, актуальність даної роботи визначається необхідністю розробки та впровадження інтелектуальних IoT-систем для підвищення рівня енергоефективності, а також потребою адаптації існуючих технічних рішень до сучасних програмних платформ і технологічних стандартів, що відповідає загальним тенденціям цифрової трансформації та розвитку енергетичної галузі.

Мета роботи - реалізація інформаційної IoT-системи моніторингу електроспоживання та візуалізації даних у режимі реального часу.

Для досягнення мети, у бакалаврській роботі успішно виконано наступні завдання.

1. Огляд інфраструктур інтернету речей для енергетичних систем будівель.
2. Аналіз вимог і програмно-інструментальне забезпечення IoT-системи.

3. Реалізація та тестування інформаційної системи моніторингу електроспоживання.

Об'єкт дослідження – процес моніторингу та контролю електроспоживання в електричних мережах із використанням технологій IoT.

Предмет дослідження – методи та засоби побудови IoT-системи для вимірювання електричних параметрів.

Методи дослідження. У процесі виконання бакалаврської роботи було застосовано методи системного аналізу, метод проєктування програмно-апаратних систем, метод калібрування вимірювальних пристроїв та методи візуалізації даних.

Наукова новизна одержаних результатів. Запропоновано підхід до побудови IoT-системи моніторингу електроспоживання на базі ESP32 із поєднанням аналогових датчиків струму та напруги та хмарної платформи Blynk 2.0, що забезпечує синхронне вимірювання, обробку та віддалену візуалізацію енергетичних параметрів у реальному часі.

Практична значущість одержаних результатів. Розроблена система дозволяє здійснювати дистанційний контроль електроспоживання в побутових та малих промислових об'єктах, забезпечує автоматизований збір даних та може бути використана як готове рішення або прототип для впровадження у системах енергомоніторингу та «розумного будинку».

РОЗДІЛ 1 ОГЛЯД ІНФРАСТРУКТУР ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ ЕНЕРГЕТИЧНИХ СИСТЕМ БУДІВЕЛЬ

У даному розділі здійснено аналіз застосування технологій Інтернету речей (IoT) для підвищення ефективності використання енергетичних ресурсів. Розглянуто сучасні наукові підходи та практичні рішення щодо побудови IoT-систем, орієнтованих на підвищення енергоефективності в житловому секторі.

Проаналізовано основні аспекти проектування IoT-рішень, зокрема вибір датчиків і виконавчих пристроїв, способи їх енергоживлення, алгоритми збору та обробки даних, методи прогнозування енергоспоживання, а також технології обміну інформацією між компонентами системи. Окремо розглянуто приклади застосування IoT у будівельних енергетичних системах.

Значну увагу приділено ролі IoT у моніторингу та контролі енергоспоживання, а також можливостям автоматизації процесів керування енергетичними ресурсами. Описано підходи до інтеграції сенсорних мереж і хмарних сервісів для збору та візуалізації даних.

Також розглянуто фактори, що впливають на ефективність проектування IoT-систем, включаючи індивідуальні параметри користувачів та умови експлуатації. Узагальнено основні напрями розвитку технологій IoT у сфері енергетичного менеджменту будівель.

1.1 Загальні принципи та розвиток Інтернету речей і його застосування в енергетичних системах будівель

Інтернет речей (IoT) є сучасною технологічною концепцією, що активно розвивається та охоплює все більше сфер застосування. Її витоки пов'язують із початком 2000-х років, коли технологія RFID почала використовуватися для

автоматизації процесів обліку товарів і відстеження логістичних операцій [1]. Сутність IoT полягає у створенні мережі взаємопов'язаних обчислювальних пристроїв, які здатні автоматизувати різноманітні процеси та забезпечувати більш ефективне використання ресурсів у повсякденній діяльності людини.

Хоча ідея підключених пристроїв обговорюється в науковій спільноті ще з кінця 1990-х років, практичний розвиток цієї технології все ще триває та перебуває на етапі активного розширення [2]. Очікується, що кількість пристроїв, підключених до мережі, у найближчі роки зростатиме експоненційно та досягне десятків мільярдів одиниць, що свідчить про подальше масштабне впровадження IoT-рішень [3]. За оцінками, вже до 2025 року їх кількість становила близько 50 мільярдів пристроїв.

Основним фактором такого стрімкого розвитку є широке коло можливих застосувань IoT. Технології Інтернету речей дозволяють автоматизовано збирати та аналізувати дані, які раніше отримувалися вручну або взагалі не фіксувалися. Використання цих даних сприяє підвищенню зручності та безпеки життєдіяльності, а також оптимізації бізнес-процесів і підвищенню загальної ефективності використання ресурсів.

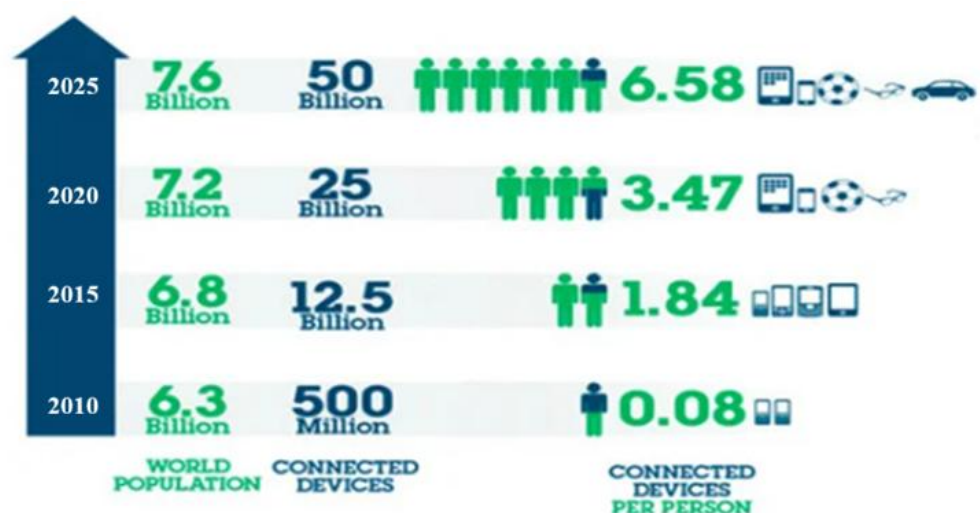


Рис.1.1 Загальна чисельність населення світу та підключені пристрої до 2025року

З точки зору архітектури, системи Інтернету речей базуються на інтеграції різноманітних датчиків та інтелектуальних пристроїв, які підключаються до локальних або хмарних контролерів. Сенсорні пристрої здійснюють безперервний збір і передачу даних про стан навколишнього середовища в режимі реального часу. Отримана інформація використовується керуючими системами для формування як оперативних, так і довгострокових рішень. Використання прогностичних та адаптивних алгоритмів дозволяє реалізовувати керуючі дії різного рівня складності - від простих сценаріїв, таких як автоматичне вмикання освітлення на основі сигналів датчиків присутності, до комплексних систем оптимізації, наприклад, керування системами опалення, вентиляції та кондиціонування повітря (HVAC) з урахуванням статистики заповненості приміщень та кліматичних умов.

Архітектура IoT-систем зазвичай є багаторівневою та включає три основні складові: апаратний (сенсорний) рівень, рівень керуючого програмного забезпечення та прикладний рівень. Апаратний рівень охоплює фізичні пристрої з унікальними ідентифікаторами, які здатні передавати дані через мережу без участі людини в процесі взаємодії. До цього рівня належать датчики для моніторингу параметрів довкілля, а також виконавчі механізми, що забезпечують керування інженерними системами будівель, зокрема освітленням або системами опалення. Сучасні пристрої часто мають функціональність прямого обміну даними між собою, утворюючи елементи інтелектуальної мережі. Масштаби IoT-систем можуть суттєво варіюватися - від невеликої кількості датчиків у приватному будинку до мільйонів пристроїв у великих промислових або комерційних комплексах [4]. Для забезпечення їхньої взаємодії необхідно розробляти надійні програмні рівні, які координують обмін даними між вузлами та реалізують відповідні алгоритми керування, наприклад, активацію або вимкнення обладнання.

IoT-системи повинні бути орієнтовані на покращення функціональності тих застосувань, у яких вони використовуються, при цьому кінцеві користувачі суттєво впливають на формування вимог до їхньої архітектури. Водночас адаптація таких систем до поведінкових особливостей користувачів призводить до постійного

збору даних про їхню активність і моделі поведінки, що може створювати ризики щодо конфіденційності та безпеки. Саме тому питання захисту даних і забезпечення приватності є ключовими для стабільного та масштабованого впровадження IoT-рішень. Узагальнена структура основних компонентів функціональної IoT-системи наведена на рис. 1.2.

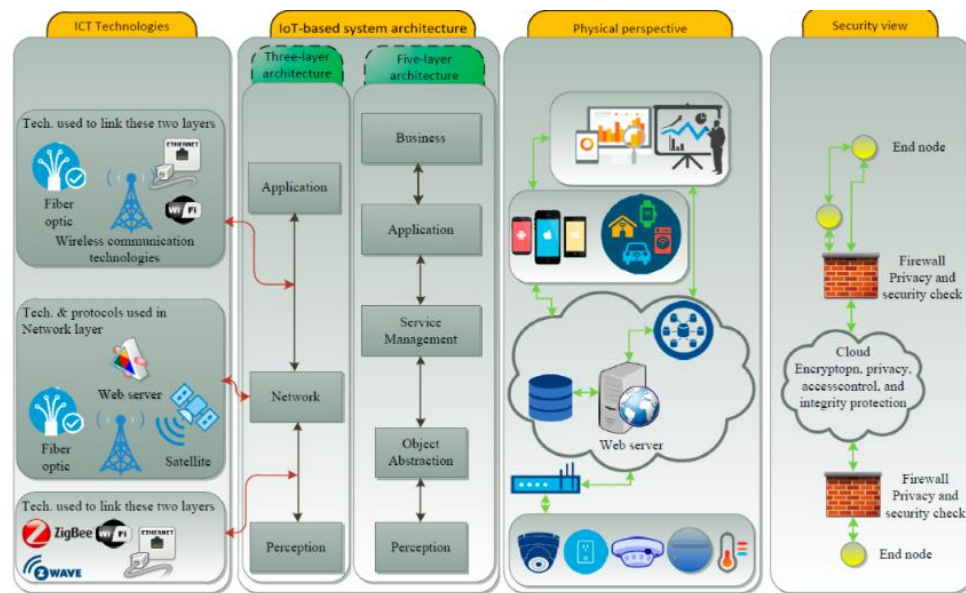


Рис.1.2 Архітектура системи на основі Інтернету речей

Управління енергоспоживанням у будівлях характеризується низкою специфічних вимог, обмежень і підходів до їх вирішення. У сучасних умовах усе більшого поширення набуває концепція «розумної будівлі». Під цим поняттям розуміють об'єкт, оснащений інноваційними технологічними рішеннями, які забезпечують автоматизацію різних процесів, зокрема керування системами опалення, вентиляції та кондиціонування повітря, а також контроль енергоспоживання, наприклад, через інтелектуальні розетки. Попри те, що комфорт користувачів залишається важливим аспектом, ключовим завданням є підвищення енергоефективності, мінімізація втрат і забезпечення використання енергоресурсів лише за реальної потреби.

Функціонування розумних будівель базується на аналізі різномірних даних, які визначають ефективність роботи як окремих компонентів, так і всієї системи в

цілому. До таких даних належать як історичні показники (наприклад, профілі споживання енергії та моделі присутності користувачів), так і поточна інформація в режимі реального часу (температурні умови, фактична присутність людей у приміщеннях тощо). Один із підходів передбачає використання систем керування енергоспоживанням на основі місцезнаходження, що трансформують традиційні централізовані та статичні механізми в більш гнучкі — динамічні та розподілені. Типова архітектура Інтернету речей у цьому контексті включає чотири ключові складові: мобільні засоби моніторингу та керування, автоматизовані механізми, орієнтовані на місцезнаходження, хмарну платформу для обробки й зберігання даних, а також прикладні сервіси, що реалізують алгоритми енергозбереження [5].

Згідно з науковими джерелами, на сьогодні проведено значну кількість оглядових досліджень, присвячених системам Інтернету речей [6]. Водночас відсутні ґрунтовні роботи, які б комплексно розглядали інфраструктуру IoT саме в контексті енергетичних систем будівель. Це свідчить про новизну та актуальність даного напрямку досліджень. Представлений огляд не дублює вже існуючі напрацювання у сфері промислових IoT-рішень, а зосереджується на більш вузькій, проте глибшій аналітиці сучасних підходів до застосування IoT у будівельній енергетиці.

Доцільність проведення такого дослідження обґрунтовується кількома ключовими чинниками. По-перше, енергетичні системи будівель є важливим і відносно доступним напрямом для досягнення глобальних цілей сталого розвитку. За даними міжнародних організацій, зокрема Управління енергетичної інформації (EIA), понад третина загального енергоспоживання припадає на житлові та комерційні будівлі. Відповідно, навіть незначне підвищення ефективності використання енергії може забезпечити вагомі економічні та екологічні результати. По-друге, концепція енергетичного Інтернету будівель базується на специфічному баченні «розумних будівель», що суттєво відрізняється від традиційних промислових IoT-рішень і потребує спеціалізованих підходів. Такі системи зазвичай ґрунтуються на мережах датчиків і виконавчих пристроїв, інтегрованих із

підсистемами будівлі, включаючи освітлення, HVAC-системи, ліфти та системи відеоспостереження. Використання адаптивних і прогнозних алгоритмів керування дозволяє не лише зменшити енергоспоживання, а й підвищити рівень комфорту для користувачів. По-третє, реалізація таких систем ускладнюється через значну різноманітність даних і варіативність конфігурацій керування, які можуть бути як централізованими, так і децентралізованими.

На рисунку 1.3 наведено узагальнення основних наукових досліджень, присвячених інфраструктурам Інтернету речей у системах енергозабезпечення будівель.

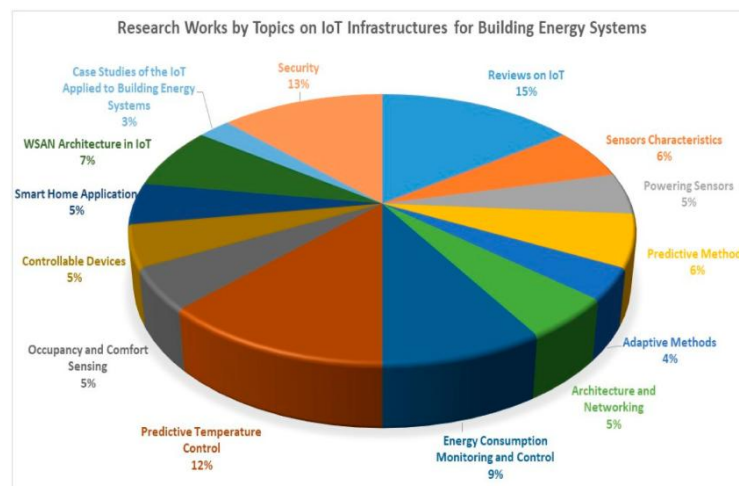


Рис.1.3 Дослідницькі роботи за темами інфраструктури Інтернету речей для енергетичних систем будівництва

1.2 Класифікація датчиків та методи їх енергозабезпечення

Типи датчиків. Значна кількість наукових досліджень спрямована на обґрунтування ефективності застосування технологій Інтернету речей для підвищення енергоефективності та зниження рівня споживання енергії.

У роботі [7] проведено оцінку доцільності впровадження низки інтелектуальних рішень, орієнтованих на оптимізацію енергоспоживання. До таких рішень віднесено автоматизоване керування системами опалення, вентиляції та кондиціонування повітря, освітленням, енергоємними побутовими приладами, а

також системами циркуляції гарячої води. Результати дослідження засвідчили, що всі зазначені процеси можуть бути реалізовані з можливістю дистанційного керування, що робить їх придатними для інтеграції в IoT-системи управління енергією.

Аналогічні висновки наведено у роботі [8], де додатково підтверджено ефективність використання інтелектуальних технологій для організації енергоменеджменту в житлових умовах.

Подальший розвиток цієї тематики представлено у дослідженні [9], де виконано кількісну оцінку потенційної економії енергії залежно від типу використовуваних датчиків Інтернету речей. Розглянуті сенсори суттєво відрізняються як за функціональним призначенням, так і за типом вихідних даних — від датчиків наближення, що передають дискретні сигнали, до сенсорів якості повітря, які генерують багатовимірні аналогові дані.

Рівень досягнутої економії енергії при цьому може суттєво варіюватися. Узагальнені результати щодо типів датчиків, відповідних інтелектуальних технологій та очікуваного ефекту наведено в таблиці 1.1 [9].

Таблиця 1.1

Очікуване зниження енергоспоживання для різних датчиків

Система	Технології	Зниження енергоспоживання
Аналітика	Хмарна енергетична інформаційна система	5–10% всієї будівлі
Автоматизація будівель	Система автоматизації будівель	10–25% всієї будівлі
Освітлення	Розширене керування освітленням	45%
Освітлення	Веб-система керування освітленням	Зниження на 20–30% порівняно з контрольним рівнем
Затінення вікон	Автоматизована система затінення	21–38%
Затінення вікон	Розумне скло	20–30%
Затінення вікон	Плівка, яка перемикається	32–43%

Зокрема, модернізація окремого елементу енергоспоживання в житлі дозволяє знизити витрати енергії приблизно на 5–15%, тоді як комплексне впровадження інтелектуальних рішень у межах енергоавтономної будівлі може забезпечити економію до 50%.

У роботі [10] запропоновано альтернативний підхід до організації живлення сенсорних пристроїв, що дозволяє суттєво знизити енергетичні вимоги при побудові систем Інтернету речей. Дослідження базується на реалізації, де декілька датчиків для моніторингу якості повітря інтегруються в єдину платформу на базі Arduino, яка виконує функції керування та обробки даних. Використані сенсори включають аналогові пристрої, підключені до відповідних входів/виходів контролера, а також датчики, що формують дискретні (булеві) та цифрові сигнали. Завдяки застосуванню енергоефективного контролера та організації живлення через його контакти, загальне енергоспоживання системи може становити близько 100 мВт навіть при максимальному навантаженні.

Живлення датчиків. Однією з ключових проблем при розгортанні сенсорних мереж у системах енергоменеджменту будівель є забезпечення ефективного розподілу живлення та управління ресурсом роботи пристроїв. Незважаючи на те, що більшість IoT-датчиків характеризуються низьким рівнем енергоспоживання, складність виникає через їхню велику кількість і розподілений характер розміщення. Використання автономних джерел живлення, зокрема батарей, зазвичай є достатнім рішенням, однак їх регулярне обслуговування та заміна створюють додаткові експлуатаційні труднощі. У зв'язку з цим активно досліджуються альтернативні підходи до живлення сенсорних систем.

Зокрема, у роботі [9] розглянуто застосування технології Power over Ethernet (PoE) для живлення датчиків у системах інтелектуального освітлення. Даний підхід дозволяє одночасно передавати електроживлення та забезпечувати канал зв'язку між датчиками і керуючими пристроями, що спрощує архітектуру системи. Використання PoE дає змогу жити як сенсори, так і освітлювальні елементи через єдину інфраструктуру. Водночас цей метод має обмеження: необхідність

прокладання кабельних з'єднань, а також обмеження напруги (до 57 В постійного струму), що може бути недостатнім для живлення деяких сучасних систем освітлення. З цієї причини РоЕ менш ефективний для модернізації існуючих систем, але добре підходить для впровадження нових рішень.

У дослідженні [11] проаналізовано можливість використання відновлюваних джерел енергії, таких як вітер і сонячне випромінювання, для живлення сенсорів на рівні міліватних потужностей. Розглянуто особливості мініатюризації енергогенеруючих пристроїв, зокрема залежність між розмірами мікротурбін і їхньою надійністю в умовах експлуатації. Отримані результати показали, що за сприятливих умов такі системи здатні генерувати до 10 мВт потужності. Крім того, досліджено підходи до управління енергією, які передбачають перенаправлення надлишкової енергії від сенсорних вузлів до центральних контролерів. Подібні висновки також підтверджено у роботі [12].

Схожі результати отримано у дослідженні [13], де проведено порівняльний аналіз різних відновлюваних джерел енергії для живлення IoT-датчиків. Узагальнені результати представлені в таблиці 1.2.

Таблиця 1.2

Виробництво електроенергії з різних відновлюваних джерел енергії

Система збору	Вихідна потужність	Система збору	Вихідна потужність
Фотоелектричні (зовнішні)	50 мВт/см ²	Рух повітря	6 Вт/см ²
Фотоелектричні (для приміщень)	50 мкВт/см ²	Перетворення зміни тиску на електрику	15 мкВт/см ²
Фотосинтез (лабораторія)	10–40 мкВт/см ²	П'єзоелектрики	12,5 мкВт/см ²
Термоелектрика	20 мкВт/см ²	Трибоелектрика	3 мВт/см ²
Піроелектрики	8,64 мкВт/см ²	Електростатика	12 мВт/см ²
Хімічний потенціал	3 мВт	Індукція	70+ мкВт/см ²

1.3 Методи прогнозного та адаптивного керування в IoT-системах енергоменеджменту

Методи прогнозування.

Підходи прогнозного керування орієнтовані на оцінювання майбутнього рівня енергоспоживання на основі аналізу сукупності факторів, серед яких історичні дані використання енергії, кліматичні умови та поведінкові особливості користувачів. Важливою передумовою розроблення таких алгоритмів є те, що попри варіативність миттєвих показників (наприклад, погодних змін або короткострокових коливань споживання), довгострокові тенденції в історичних даних залишаються відносно стабільними та піддаються прогнозуванню.

Зокрема, аналіз присутності мешканців у приміщенні протягом кількох днів дозволяє виявити характерні шаблони використання житлового простору як на рівні домогосподарства, так і окремих осіб. Хоча щоденні відхилення можливі, дослідження свідчать, що такі зміни зазвичай не перевищують близько 10% від загальної тенденції [9]. Подібні закономірності можуть бути екстрапольовані на більш тривалі часові інтервали. Наприклад, типова поведінка мешканця, який працює у будні дні та проводить вихідні вдома, формує повторювані моделі енергоспоживання протягом тижнів, а також демонструє сезонну варіативність упродовж року. У деяких випадках подібні патерни можуть зберігатися навіть протягом кількох років.

Порівняльний аналіз даних може здійснюватися на основі тривалих часових інтервалів — від місяців до років. Кліматичні показники, накопичені за тривалий період, дають змогу визначити узагальнені тенденції та сформувати базу для довгострокового прогнозування. Незважаючи на складність точного передбачення погодних умов, середньострокові (тижневі або місячні) закономірності, отримані з багаторічних спостережень, є більш стабільними та придатними для аналізу. Як зазначено у роботі [14], використання усереднених історичних метеоданих є доцільним орієнтиром при побудові прогнозних моделей.

Для обробки та інтеграції даних з метою зменшення похибок і впливу аномальних значень застосовуються різноманітні методи математичного та статистичного моделювання. До них належать множинна лінійна регресія, адаптивні фільтри, алгоритми нормалізованих і рекурсивних найменших квадратів, а також моделі на основі гаусових сумішей. Вибір конкретного методу визначається специфікою задачі та умовами застосування.

Отримані в результаті аналізу дані використовуються для формування алгоритмів керування енергоспоживанням, що підтверджено у роботі [15]. Зокрема, у системах опалення та кондиціонування історичні дані дозволяють визначити цільові температурні режими та відповідні рівні енергоспоживання. Водночас ці параметри можуть змінюватися залежно від зовнішніх і внутрішніх умов. Для запобігання надмірним коливанням роботи систем застосовується зона нечутливості (гістерезис), яка зазвичай становить 4–6 °C. У межах цього діапазону система не реагує на незначні зміни температури, що сприяє зниженню енергоспоживання та підвищенню стабільності роботи. На основі зазначених принципів було запропоновано відповідний метод керування.

У дослідженні [16] основну увагу зосереджено на аналізі та класифікації типових поведінкових моделей мешканців у динаміці часу. Авторами було розглянуто запатентований алгоритм DeepHare, який базується на обробці даних, отриманих із різних джерел. Зокрема, використовувалися відеодані з камер, підключених через Wi-Fi, інформація з датчиків наближення, а також показники акселерометрів мобільних пристроїв для ідентифікації повсякденних дій користувачів.

Сформований масив даних надалі піддавався обробці за допомогою спеціалізованої моделі вилучення ознак — довгострокової рекурентної згорткової автоенкодерної мережі. У межах цієї моделі враховувалися як безпосередні сигнали від сенсорів, так і узагальнені історичні дані, сформовані шляхом об'єднання кількох часових послідовностей у єдине представлення для подальшого аналізу.

Отримані в результаті обробки закономірності дозволили ефективно ідентифікувати поведінкові патерни користувачів. За результатами експериментальної перевірки запропонований алгоритм продемонстрував високий рівень точності, що досягав 97%.

Адаптивні методи.

На відміну від прогнозних підходів, адаптивні методи орієнтовані на коригування рівня енергоспоживання будівлі на основі зворотного зв'язку, що надходить у режимі реального часу.

У дослідженні [17] проведено аналіз ефективності різних стратегій керування вентиляційними системами із застосуванням механізмів зворотного зв'язку. Було розглянуто три підходи: керування типу «увімкнено/вимкнено», ПД-регулювання та метод керування за станом із перемиканням режимів. Результати показали, що саме керування за станом забезпечує найкращі показники — мінімальний час встановлення (менше 0,05 с), високу стабільність і надійність роботи системи. Натомість простий алгоритм «увімкнено/вимкнено», який фактично реалізує змінний робочий цикл, характеризується нижчою стійкістю та більшою чутливістю до збурень.

У роботі [18] досліджено можливість використання зворотного зв'язку від користувачів для регулювання температурного режиму з урахуванням їхнього комфорту. Перед встановленням цільової температури визначався індекс теплового комфорту — прогнозоване середнє значення, яке змінюється в діапазоні від -3 (відчуття холоду) до +3 (відчуття перегріву). Цей показник формується з урахуванням температури повітря та теплового випромінювання, швидкості повітряних потоків, вологості, а також індивідуальних факторів, таких як рівень одягу та метаболічна активність користувачів. У подальшому розрахункові моделі коригувалися в реальному часі для отримання уточненого показника — спостережуваного середнього значення. Додатково мешканці мали можливість через веб-інтерфейс оцінювати власні відчуття мікроклімату, що дозволяло визначати фактичне середнє значення. Експериментальні результати показали, що

розрахунковий рівень комфорту часто відрізнявся від суб'єктивного сприйняття користувачів, що вказує на необхідність додаткового коригування температурних параметрів, запропонованих алгоритмом.

У роботі [19] розглянуто інтеграцію адаптивних і прогнозних підходів у єдиній системі керування енергоспоживанням. Для цього було застосовано оптимізовану прогностичну модель, яка оцінює динаміку споживання енергії залежно від параметрів мікроклімату, якості повітря та рівня природного освітлення. Отримані результати використовувалися як вхідні дані для нечіткого контролера, який оперує не дискретними значеннями, а ступенями належності (наприклад, «холодно», «тепло», «спекотно»). Порівнюючи прогнозовані та фактичні значення параметрів і враховуючи похибку, система здатна більш точно визначати оптимальний рівень енергоспоживання. Схематичне представлення такої системи наведено на рис. 1.4.

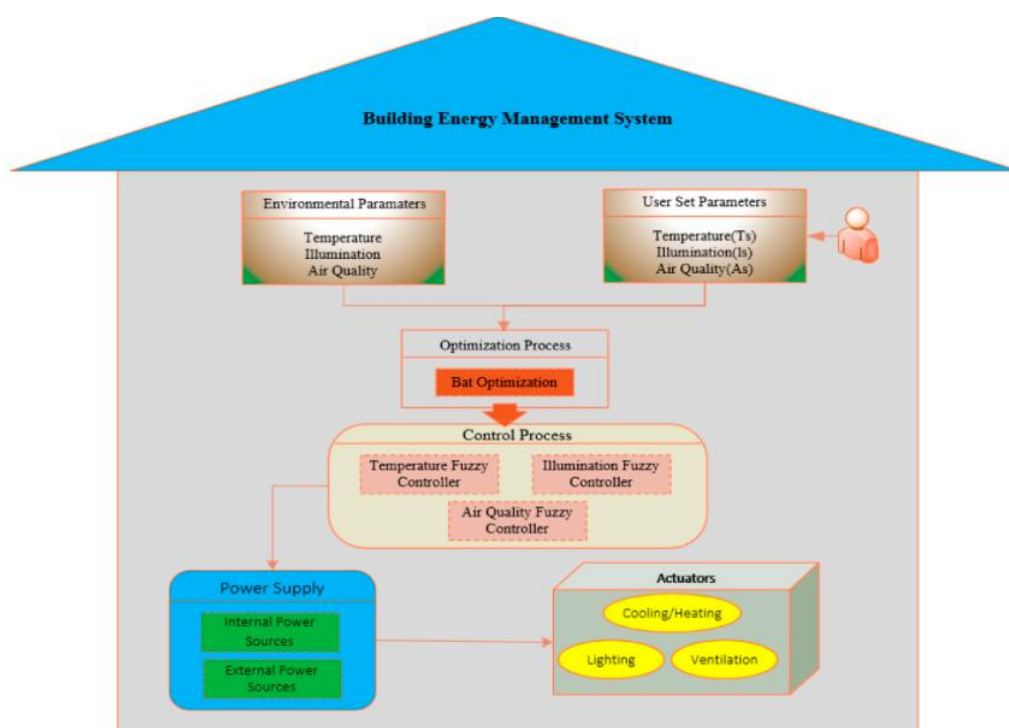


Рис.1.4 Система енергоменеджменту будівлі з використанням підходу нечіткої логіки

1.4 Мережеві архітектури та протоколи зв'язку в системах IoT

Протягом останніх двадцяти років було розроблено значну кількість мережевих архітектур, які забезпечують надійний та безпечний обмін даними між пристроями й керуючими елементами. Особливості функціонування різних комунікаційних технологій, що застосовуються в системах Інтернету речей (зокрема Bluetooth, LoRaWAN, Wi-Fi, ZigBee та Z-Wave), детально розглянуто у [15]. Важливо зазначити, що значна частина цих протоколів залишається актуальною і широко використовується в сучасних IoT-рішеннях. Нижче наведено приклади досліджень, які підтверджують ефективність застосування існуючих мережевих технологій.

У роботі [20] досліджено використання бездротових сенсорів, що функціонують у діапазоні 2,4 ГГц, у межах житлової IoT-архітектури для підвищення ефективності керування системами опалення, вентиляції та кондиціонування повітря (HVAC). Вузли бездротової сенсорної мережі (WSN) взаємодіють між собою через сервер, передаючи дані безпосередньо до прикладного IoT-сервера, минаючи локальні модеми. Виконавчі пристрої HVAC підключені до відповідних сенсорів і забезпечують фізичне регулювання систем.

Загальна архітектура включає кілька ключових компонентів: модулі HVAC, виконавчі механізми для їх керування, бездротову сенсорну мережу для збору даних про температуру та енергоспоживання, шлюз (GW), який виконує функції інтеграції та обробки інформації, а також клієнтські пристрої (наприклад, смартфони чи планшети) для взаємодії користувача з системою. Шлюз поєднує локальну мережу з Інтернетом і містить веб-сервер та базу даних для накопичення інформації, отриманої як від сенсорів, так і з зовнішніх джерел.

Принцип роботи системи полягає в наступному: сенсори вимірюють температурні показники у різних зонах приміщення та передають ці дані до шлюзу. На основі отриманої інформації застосовується алгоритм оптимізації енергоспоживання, який визначає оптимальний режим роботи HVAC з

урахуванням заданих параметрів комфорту та вартості енергії. Відповідні керуючі сигнали передаються до виконавчих пристроїв, які активують або регулюють роботу обладнання, забезпечуючи необхідний температурний режим.

Усі зібрані дані зберігаються в базі даних шлюзу та можуть бути доступні користувачам віддалено через веб-інтерфейс. Це дає змогу не лише здійснювати моніторинг, але й коригувати параметри системи в межах встановлених температурних обмежень для забезпечення комфортних умов.

У роботі [15] досліджено можливості використання мережі The Things Network (TTN), яка є відкритою платформою з відкритим вихідним кодом для IoT-застосувань. Дана технологія забезпечує інтеграцію простих апаратних рішень, таких як Arduino або Raspberry Pi, із надійними програмними сервісами. Використання TTN дозволяє перетворювати базові мікроконтролери на функціональні шлюзи, які збирають дані від сенсорів, обробляють їх і передають керуючі сигнали назад до пристроїв.

Ключовою особливістю цієї архітектури є використання централізованих алгоритмічних сервісів, що забезпечують стабільність і надійність роботи системи. Основна логіка функціонування реалізується безпосередньо на мікроконтролерах, тоді як обмін даними між пристроями та контролерами здійснюється через стандартні мережі Wi-Fi із застосуванням протоколу TCP/IP (рис. 1.5).

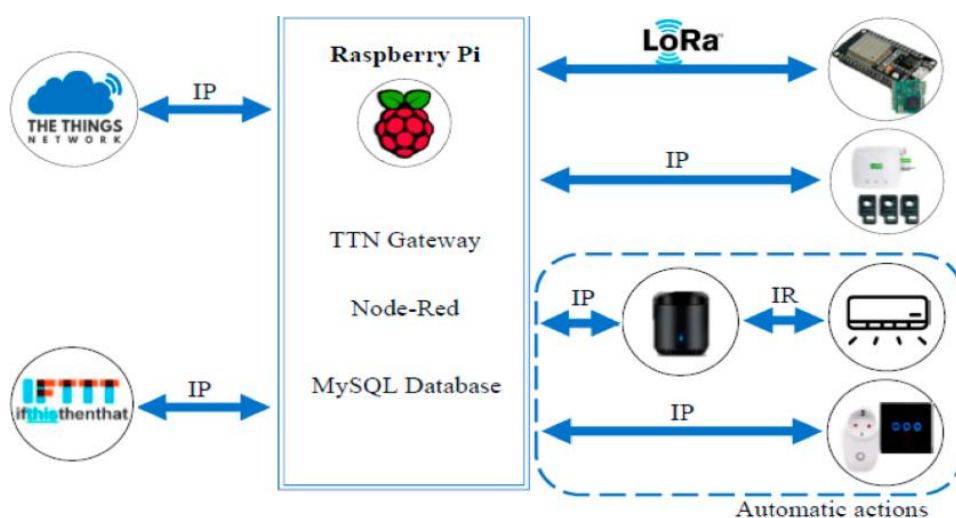


Рис.1.5 Архітектура системи на основі The Things Network

У дослідженні [21] розглянуто варіанти реалізації бездротової мережі для системи опалення житлового будинку. Замість традиційних провідних рішень було використано бездротові термостати та датчики концентрації CO₂, які розміщувалися як у внутрішніх приміщеннях, так і на зовнішніх ділянках будівлі.

Сенсорні вузли були організовані у кластери за принципом територіальної близькості. Для обміну даними всередині таких кластерів застосовувався спеціалізований датчик із обмеженим радіусом дії, який працював у частотному діапазоні 447 МГц. Використання нижчої пропускну здатності було обумовлене необхідністю зменшення енергоспоживання, оскільки передавані дані мали невеликий обсяг і не вимагали високої швидкості передачі.

Комунікація між окремими кластерами та центральним шлюзом здійснювалася на частоті 917 МГц, що забезпечувало більшу пропускну здатність і дозволяло передавати ширший обсяг інформації. На завершальному рівні архітектури для взаємодії між центральним шлюзом і сервером системи домашньої автоматизації було використано дротове підключення через Ethernet.

Отримані від сенсорів дані надходили до серверної частини системи, де виконувалася їх обробка, аналіз і формування керуючих рішень на рівні всієї системи. Взаємодію компонентів і загальну структуру запропонованої архітектури ілюстровано на рис. 1.6.

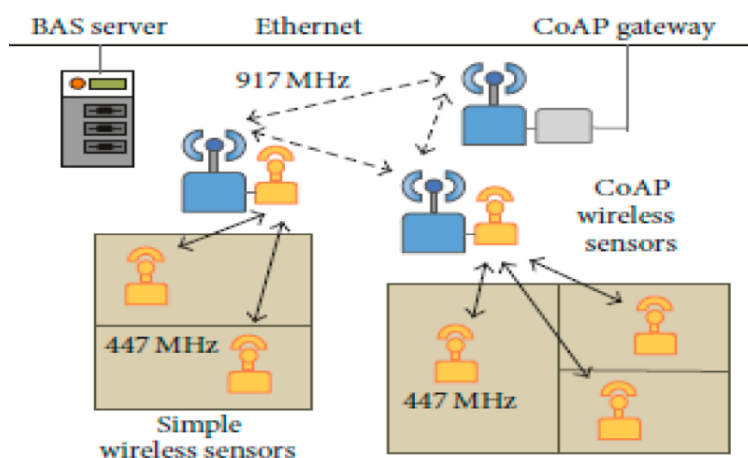


Рис.1.6 Мережа з використанням датчиків протоколу обмежених програм (CoAP)

Результати дослідження [22] підтвердили доцільність впровадження уніфікованих підходів і стандартів у системах Інтернету речей, що є необхідною умовою для їх ефективного використання в кіберфізичних системах, які оперують значними обсягами даних.

У роботі [23] розглянуто функціонування IoT-мереж у межах концепції розумних міст, де взаємодіють численні об'єкти, зокрема будівлі та транспортні засоби. Особливу увагу приділено задачам управління ресурсами в умовах одночасної роботи великої кількості сенсорних пристроїв. Авторами запропоновано дві алгоритмічні стратегії: адаптивне регулювання пропускної здатності та потокову передачу з урахуванням затримок.

Перший підхід передбачає динамічне налаштування частоти передавання даних сенсорами з метою зменшення енергоспоживання, що, однак, може призводити до збільшення тривалості передачі. Другий метод реалізує протилежний принцип: у випадку зростання затримок система підвищує потужність передачі, збільшуючи пропускну здатність і, відповідно, скорочуючи час передавання даних.

Для забезпечення стабільної роботи мережі необхідним є використання додаткових алгоритмів, які дозволяють оптимально балансувати між енергоспоживанням і затримками передачі. Застосування запропонованих підходів продемонструвало можливість зниження енергетичних витрат приблизно на 37% порівняно з іншими аналогічними рішеннями.

1.5 Моніторинг та керування енергоспоживанням у IoT-системах будівель

Інтернет речей виступає ефективним засобом для здійснення моніторингу та керування енергоспоживанням у будівлях. IoT-системи можуть бути налаштовані таким чином, щоб забезпечувати вимірювання енерговитрат безпосередньо на рівні окремих джерел, що дозволяє аналізувати роботу кожного пристрою окремо. Крім

того, такі системи дають змогу здійснювати деталізований облік споживання енергії за окремими приміщеннями або поверхами будівлі.

Зібрані дані мають важливе практичне значення. По-перше, вони дозволяють ідентифікувати обладнання або системи з підвищеним рівнем енергоспоживання, що відкриває можливості для їх оптимізації. Це також сприяє кращому розумінню впливу різних факторів і дій користувачів на загальне енергоспоживання, що є корисним для оцінки ефективності впроваджених заходів енергозбереження. По-друге, IoT-рішення можуть виконувати не лише функцію спостереження, а й активного керування, наприклад, автоматично відключаючи пристрої для усунення споживання в режимі очікування. По-третє, система здатна регулювати доступ обладнання до електроживлення, обмежуючи його використання для некритичних задач або зміщуючи навантаження на періоди з нижчим рівнем споживання в мережі. Такий підхід є економічно вигідним і сприяє більш рівномірному розподілу енергетичного навантаження.

У дослідженні [23] описано впровадження IoT-системи моніторингу енергоспоживання у будівлі навчального закладу. Оскільки подібні об'єкти є поширеними, запропоноване рішення може бути масштабоване та використане в інших установах. Додатковою перевагою є можливість ознайомлення учнів із сучасними технологіями енергоменеджменту, що може мати позитивний освітній ефект у довгостроковій перспективі.

Метою реалізації системи було створення платформи для збору та аналізу даних у реальному часі, що охоплює як параметри енергоспоживання, так і характеристики внутрішнього та зовнішнього середовища. Архітектура базується на використанні доступної сенсорної мережі, розподіленої по всій будівлі, що забезпечує економічно ефективний моніторинг.

Сенсорні пристрої підключалися до вузлів на базі Arduino, які здійснювали обмін даними через бездротовий інтерфейс XBee із шлюзом IoT. Шлюз, у свою чергу, мав дротове підключення до хмарної інфраструктури. Для визначення енергоспоживання використовувалися датчики струму, встановлені на

провідниках, що дозволяло обчислювати споживану потужність з урахуванням напруги та коефіцієнта потужності.

Загальна структура системи передбачала використання декількох інсталяцій у різних частинах будівлі. Кожна з них складалася з набору IoT-вузлів, які взаємодіяли з хмарними сервісами через шлюз. У залежності від конфігурації, кожне приміщення могло містити один або кілька таких вузлів, оснащених сенсорами для вимірювання параметрів середовища.

До складу системи входили:

- пристрої вимірювання енергоспоживання, що забезпечували контроль як на рівні всієї будівлі, так і окремих зон;
- сенсори навколишнього середовища для оцінки температури, вологості, рівня шуму та активності у приміщеннях;
- шлюзові вузли, розташовані у ключових точках будівлі, які забезпечували взаємодію між локальною мережею та Інтернетом.

Для організації зв'язку між вузлами використовувався стандарт IEEE 802.15.4, реалізований через модулі XBee та відповідні програмні бібліотеки. Усі IoT-пристрої формували децентралізовану мережу, передаючи дані через шлюзи до хмарних сервісів, а також отримуючи керуючі команди та оновлення.

У дослідженні [24] розглянуто впровадження IoT-орієнтованої системи моніторингу енергоспоживання в умовах промислової будівлі. Як центральний елемент керування було обрано платформу Raspberry Pi, що обумовлено її доступністю, надійністю та достатньою обчислювальною потужністю для виконання поставлених задач.

Для реалізації функцій збору та обробки даних використовувалося середовище на базі Node.js, яке забезпечувало отримання інформації з лічильників електроенергії та її локальне збереження. Подальший доступ до накопичених даних здійснювався з використанням клієнтських пристроїв (ноутбуків або мобільних пристроїв) через систему візуалізації Grafana.

Запропонована система сприяла підвищенню ефективності енергоспоживання за рахунок забезпечення постійного моніторингу та аналізу даних. Її архітектура включала існуючі вимірювальні прилади, обчислювальний модуль на базі Raspberry Pi, хмарну інфраструктуру (зокрема базу даних, веб-сервіси обробки інформації, а також додатки для керування і моніторингу) та засоби візуалізації.

Загалом для реалізації IoT-систем можуть використовуватися різні апаратні платформи, такі як Arduino, Intel Edison, MediaTek LinkIt ONE або NVIDIA Jetson Nano. Проте в межах даного дослідження перевагу було надано саме Raspberry Pi як найбільш оптимальному рішення з точки зору функціональності та вартості.

Схематичне представлення системи моніторингу енергоспоживання наведено на рисунку 1.7 [24].



Рис.1.7 Принципова блок-схема системи моніторингу енергії Інтернету речей

У дослідженні [25] розглянуто створення економічно ефективною IoT-системи для ідентифікації електричних приладів за характером їхнього енергоспоживання. Актуальність цієї задачі зростає разом із розвитком систем автоматизації будівель і розумних домів, де виникає потреба у детальному контролі та аналізі роботи окремих пристроїв. Запропоновані підходи можуть бути використані в різних прикладних сферах, зокрема для управління попитом на електроенергію, організації прогностного технічного обслуговування, підвищення прозорості енергоспоживання, а також у медичних застосуваннях, де аналіз профілів споживання дозволяє робити висновки про поведінку користувачів.

У межах роботи було представлено підхід до проектування системи розпізнавання приладів на основі аналізу їхніх енергетичних характеристик. Для

підвищення точності ідентифікації проведено порівняння різних методів машинного навчання, здатних працювати зі змінними профілями споживання. Отримані результати демонструють можливість здійснення точного моніторингу енергоспоживання без необхідності встановлення окремих вимірювальних пристроїв для кожного електроприладу.

У роботі [26] представлено систему моніторингу енергоспоживання з використанням технології далекого зв'язку LoRa під назвою EnerMon. Реалізація базується на застосуванні Arduino, датчиків струму трансформаторного типу, бездротового зв'язку LoRa та сервера на основі Raspberry Pi. Запропоноване рішення відзначається низькою вартістю впровадження (близько 80 євро за один сенсорний вузол) і забезпечує отримання аналітичних даних у режимі реального часу для виявлення енергетичних втрат.

З огляду на обмежену швидкість передачі даних у мережах LoRa, у системі було використано підхід периферійних обчислень, що дозволяє виконувати первинну обробку інформації безпосередньо на рівні сенсорних вузлів. Це рішення спрощує розгортання системи та зменшує залежність від пропускної здатності каналу зв'язку. Перевагами запропонованого підходу є простота встановлення, відсутність суттєвих обмежень щодо дальності передачі сигналу та можливість використання в різних умовах — від великих будівель до невеликих об'єктів, таких як приватні будинки чи туристичні комплекси.

Архітектура системи відповідає типовій моделі IoT і включає чотири основні функціональні модулі, які наведено на рисунку 1.8.

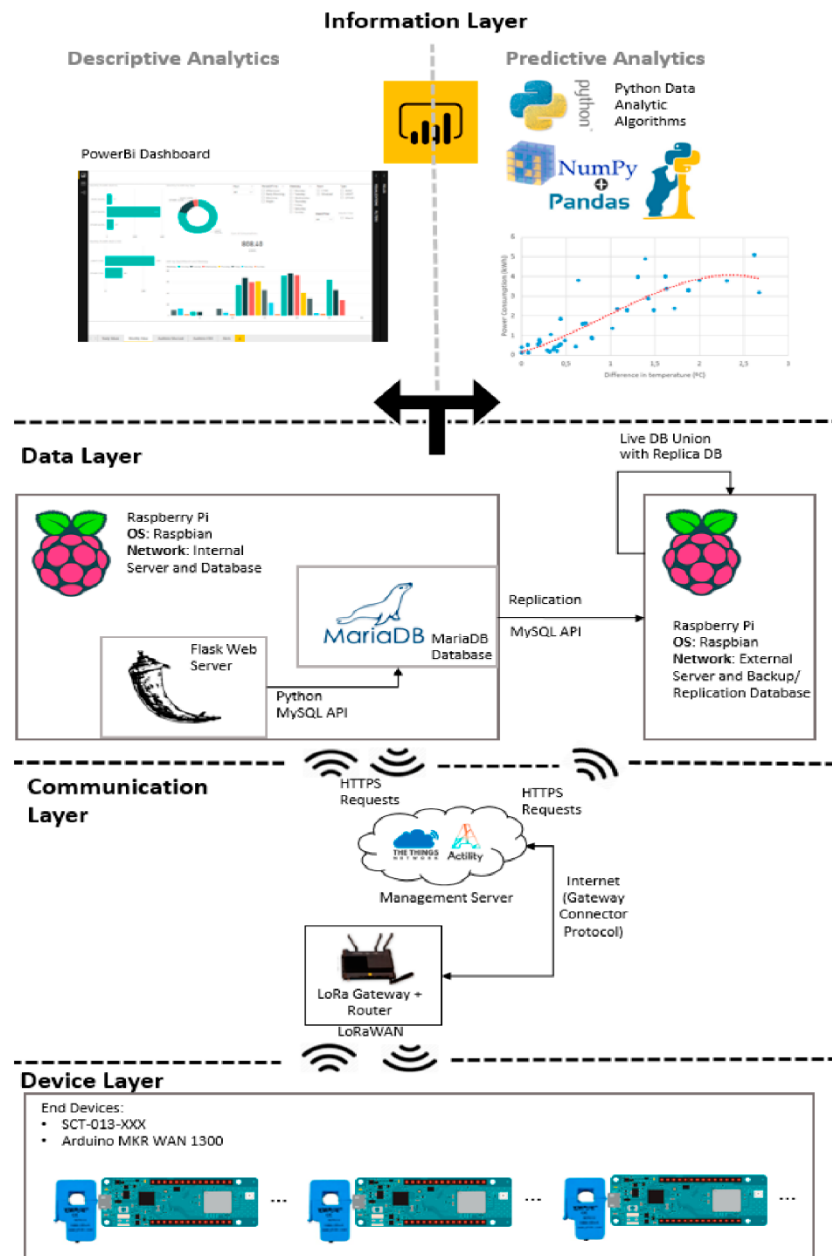


Рис.1.8 Проектування системи Інтернету речей далекого радіуса дії (LoRa) від EnerMon

Архітектура системи Інтернету речей на базі технології LoRa включає кілька функціональних рівнів, кожен з яких виконує визначені завдання:

1. Рівень пристроїв (кінцеві вузли LoRa). На цьому рівні функціонують кінцеві сенсорні пристрої, які здійснюють збір даних та їх первинну обробку із застосуванням підходів периферійних обчислень. Отримана інформація

передається безпосередньо до LoRa-шлюзу, що дозволяє уникнути використання локальних мереж короткого радіусу дії, таких як Wi-Fi.

2. Комунікаційний рівень (LoRa-шлюз). Шлюз виконує роль проміжного вузла, який приймає висхідні повідомлення від кінцевих пристроїв і передає їх до серверної частини системи. Окрім цього, він забезпечує передачу низхідних повідомлень — керуючих команд від сервера до сенсорних вузлів.

3. Рівень обробки даних (сервер додатків і база даних). На цьому етапі відбувається прийом інформації від шлюзів, її подальша обробка та збереження у базах даних. Взаємодія з кінцевими пристроями реалізується через LoRa-шлюз, що забезпечує централізоване управління системою.

4. Інформаційно-аналітичний рівень (візуалізація та аналіз). Накопичені дані використовуються для формування інтерфейсів моніторингу у вигляді інформаційних панелей із графіками та діаграмами, що налаштовуються за відповідними параметрами. Для аналітичної обробки застосовуються інструменти, такі як Power BI у поєднанні з мовою програмування Python, що дозволяє отримувати більш глибоке розуміння зібраних даних.

У таблиці 1.3 наведено порівняльний аналіз основних комунікаційних протоколів, які використовуються в IoT-системах. Як зазначено у [26], не існує універсального протоколу, який був би оптимальним для всіх випадків. Вибір конкретного рішення залежить від умов застосування, характеристик мережі та вимог до системи.

Активний розвиток технологій Інтернету речей сприяє інтеграції інтелектуальних пристроїв у житлові будівлі в межах концепції «розумних» будинків і міст. Такі пристрої здатні не лише здійснювати моніторинг, а й брати участь у процесах управління енергоспоживанням, зокрема через механізми реагування на попит (Demand Response, DR). Це дозволяє більш ефективно розподіляти енергетичні ресурси, знижувати витрати та адаптувати споживання до умов ринку електроенергії.

Таблиця 1.3

Огляд поширених парадигм комунікації Інтернету речей

Бездротові технології	Швидкість передачі даних	Максимальна тривалість зарядки	Дальність зв'язку	Безпека	Переваги
Bluetooth 5	125 кбіт/с (далека дальність S = 7) 2 Мбіт/с–500 кбіт/с (далека дальність S = 2)	255 байтів	До 200 м +200 м (BLE)	L1–Без безпеки L2–AES 128 L3–AES та сполучення L4–ECDHE	Просте обладнання; легкий доступ та експлуатація; безпечний; низьке енергоспоживання (BLE)
LoRaWAN	50 кбіт/с	243 байти	~5 км місто ~15 км–20 км село	т АЕС	Доступність; велика дальність зв'язку; безпека; низьке енергоспоживання; низька вартість
NB-IoT	200 кбіт/с	1600 байт	~1 км у місті ~10 км у сільській місцевості	Шифрування LTE	широка максимальна довжина завантаження (покриття 4G); безпечний
Cigfox	100 кбіт/с	12 байтів	~10 км у місті ~40 км у сільській місцевості	Без шифрування або адаптації для кожного випадку	Велика дальність зв'язку; низьке енергоспоживання
Wi-Fi	1 Гбіт/с IEEE 802.11ac	2034 байти	1–100 м	WPA/WPA2	Розширений стандарт; висока швидкість
ZigBee	20 кбіт/с при 868 МГц 40 кбіт/с при 915 МГц 250 кбіт/с при 2,4 ГГц	255 байтів	10–300 м у зоні прямої видимості, 75–100 м у приміщенні	128-бітний AES	Велика кількість вузлів; низьке енергоспоживання; низька вартість; безпека
Z-хвиля	100 кбіт/с	64 байти	~100 м (може змінюватися залежно від кількості вузлів) (до 4 переходів)	Безпека 2 (S2) (включаючи AES-128, ECDHE, безпечний тунель TLS)	Просте встановлення; сумісність між пристроями різних виробників; низьке енергоспоживання; безпека

У дослідженні [27] запропоновано підхід до управління енергоспоживанням житлових будівель із використанням цінової моделі DR. Авторами розроблено

алгоритм WBFA (Wind-Driven Bacterial Foraging Algorithm), який поєднує принципи оптимізації на основі вітрових процесів та поведінки бактерій під час пошуку їжі. Запропонований метод дозволяє оптимізувати роботу інтелектуальних пристроїв, зменшувати пікові навантаження, мінімізувати витрати на електроенергію та водночас підтримувати комфортні умови для користувачів. Такий підхід сприяє підвищенню стійкості IoT-орієнтованих енергетичних систем.

У роботі [28] розглянуто використання IoT для одночасного вимірювання температури внутрішнього середовища та рівня енергоспоживання систем опалення, вентиляції та кондиціонування повітря. На основі отриманих даних визначалися теплові характеристики будівлі та формувалися прогностичні моделі зміни температури. Експериментальні результати показали, що система здатна передбачати температурні тенденції на 24 години вперед із похибкою близько 2 °C.

У дослідженні [29] проаналізовано розгортання IoT-системи, призначеної для збору, обробки та передачі даних про енергоспоживання з урахуванням температурних параметрів внутрішнього середовища. Основною метою системи є підвищення поінформованості власників нерухомості та орендарів щодо використання енергії, а також забезпечення більш прозорого формування рахунків за енергоресурси. Додатково система дозволяє оцінювати умови проживання, які забезпечують орендодавці.

Архітектура запропонованої системи складається з трьох рівнів. Перший рівень включає вимірювальні та субвимірювальні пристрої, які збирають дані про споживання електричної, теплової та газової енергії. Другий рівень відповідає за агрегацію та передачу інформації, використовуючи бездротові персональні мережі на базі протоколу ZigBee, а також маршрутизатори, підключені до Інтернету. Крім того, розумні лічильники можуть передавати дані безпосередньо до хмарної інфраструктури. Третій рівень реалізує функції управління даними, включаючи їх зберігання, обробку, аналіз та візуалізацію через веб-інтерфейси.

На цьому рівні користувачі отримують доступ до інформації у вигляді звітів і графічних відображень. У межах системи передбачено два ключові функціональні компоненти:

- модуль оцінювання, який акумулює дані з різних джерел, зокрема щодо споживання та генерації енергії;
- модуль конфігурації, що зберігає інформацію з зовнішніх джерел, таких як тарифи на енергоресурси, метеорологічні показники та поведінкові характеристики споживачів.

Загальна структура інтегрованої IoT-системи наведена на рисунку 1.9.

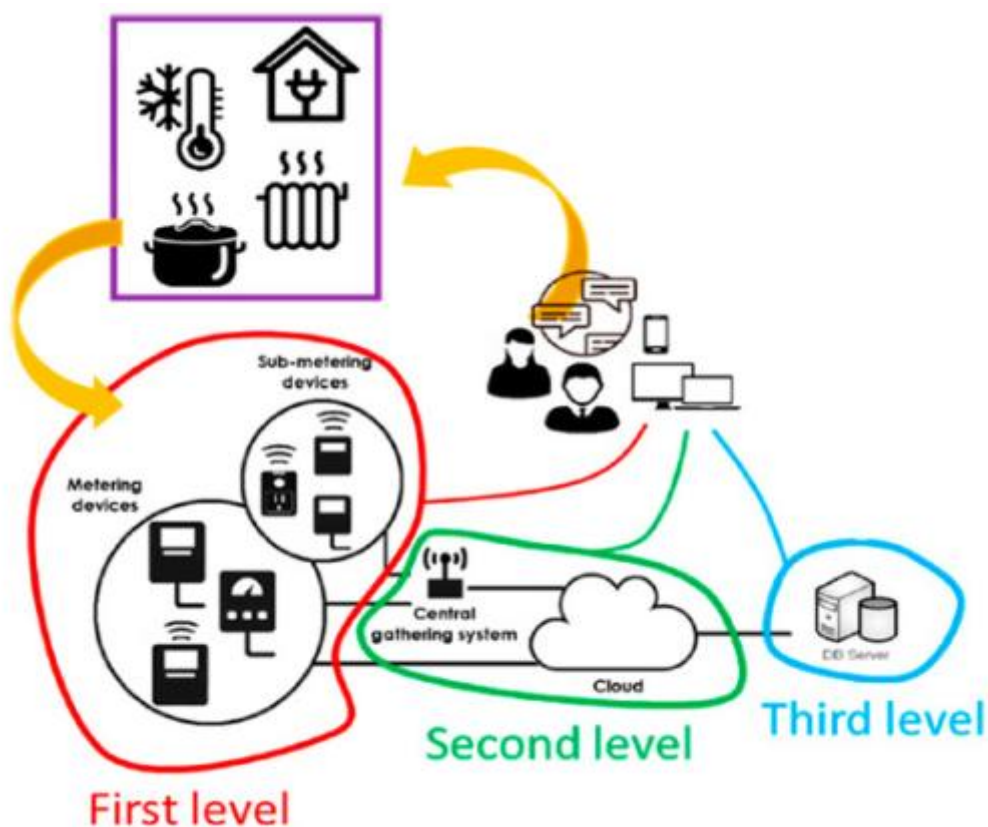


Рис.1.9 Схема інструменту, інтегрованого з Інтернетом речей

У дослідженні [30] наведено результати експериментального впровадження IoT-системи моніторингу енергоспоживання в університетській будівлі з метою підвищення енергоефективності. Було встановлено, що ключову роль у досягненні

економії відіграють як кількість точок вимірювання, так і рівень деталізації зібраних даних. Саме ці фактори суттєво впливають на можливість виявлення змін у поведінці користувачів, які, у свою чергу, визначають рівень споживання енергії.

У межах дослідження також проаналізовано методи оцінювання поведінкових змін в умовах дії неконтрольованих факторів, характерних для університетського середовища. Отримані результати продемонстрували, що під час активного функціонування системи рівень економії енергії досягав 25,4%. Після завершення експерименту було зафіксовано збереження ефекту у вигляді зниження енергоспоживання приблизно на 10% протягом наступних 44 днів.

На рисунку 1.10 представлено приклад щоденного звіту, сформованого на основі даних, отриманих від IoT-системи моніторингу.

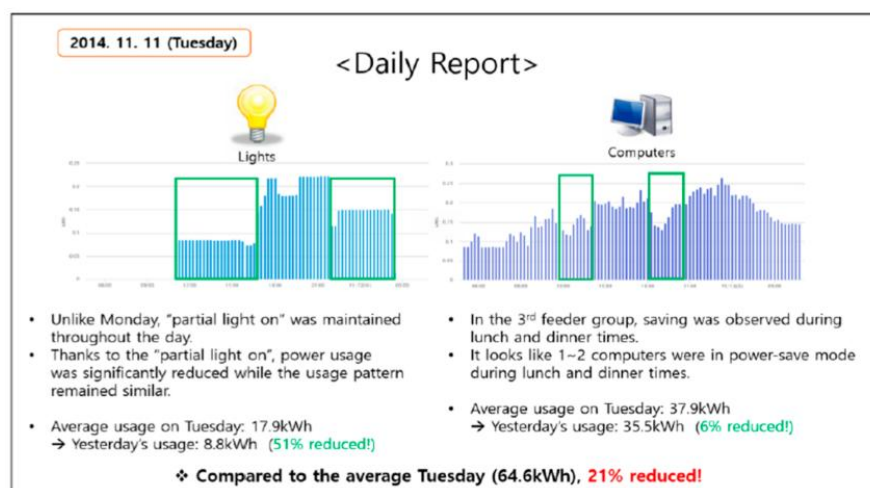


Рис.1.10 Приклад щоденного звіту, наданого в результаті аналізу системи Інтернету речей

РОЗДІЛ 2 АНАЛІЗ ВИМОГ І ПРОГРАМНО-ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ ІОТ-СИСТЕМИ

2.1 Аналіз технічної специфікації та вибір складових проекту

У таблиці 2.1 наведено узагальнений перелік компонентів, що застосовуються при створенні інтелектуального лічильника електроенергії на основі технологій Інтернету речей. До складу системи входять як апаратні засоби (мікроконтролер, сенсорні модулі вимірювання, релейні елементи, джерела живлення), так і програмне забезпечення, яке забезпечує обробку та передачу отриманих даних.

Інтеграція зазначених компонентів дозволяє реалізувати основні функціональні можливості системи, зокрема вимірювання електричних параметрів, аналіз зібраної інформації, її передавання до віддалених сервісів, а також візуалізацію результатів і дистанційне керування навантаженням. Такий підхід забезпечує ефективний моніторинг і аналіз енергоспоживання в режимі реального часу [10].

Таблиця 2.1

Перелік складових реалізації проекту

№	Компоненти	Кількість
1	ESP32	1
2	Модуль датчика змінної напруги ZMPT101B	1
3	SCT-0130030 неінвазивний датчик змінного струму	1
4	20x4 I2C РК - датчик	1
5	Кабель Micro-USB	1
6	Резистор 10 кОм	2
7	Резистор 100 кОм	1
8	Конденсатор 10 мкФ	1
9	З'єднувальні дроти	10
10	Макетна плата	1

У подальшому було проведено аналіз конструктивних особливостей і технічних характеристик окремих компонентів системи.

Датчик струму SCT-013 є неінвазивним вимірювальним пристроєм із роз'ємним (розщепленим) магнітопроводом, призначеним для контролю змінного струму до 100 А. Такий тип сенсора належить до класу трансформаторів струму (СТ) і широко застосовується для вимірювання електричних параметрів у будівлях. Основною перевагою SCT-013 є простота встановлення, оскільки він дозволяє виконувати підключення до фазного або нульового проводу без втручання в електричне коло та без необхідності проведення робіт із високою напругою, рис. 2.1.



Рис. 2.1 Датчик струму SCT-013

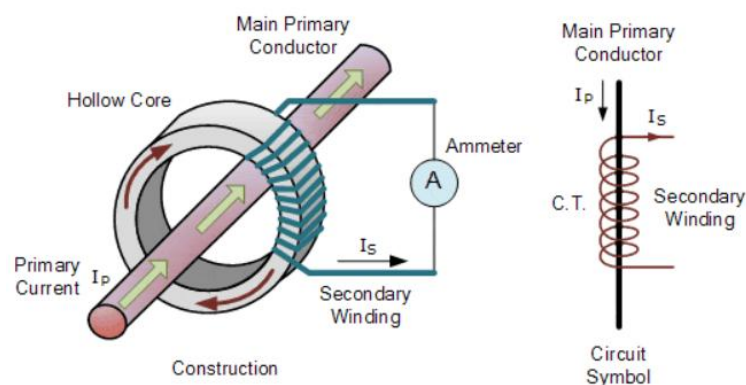


Рис.2.2 Принцип функціонування датчика струму

Датчик має первинну обмотку, магнітний сердечник і вторинну обмотку, яка складається з багатьох витків тонкого дроту, укладених у корпус трансформатора.

Модуль вимірювання змінної напруги ZMPT101B є високоточним сенсорним пристроєм, що функціонує на основі однойменного трансформатора напруги. Завдяки своїй конструкції він забезпечує точне визначення параметрів змінної напруги, що робить його доцільним для використання у системах на базі Arduino або ESP32, рис. 2.3.



Рис.2.3 Модуль датчика однофазної напруги змінного

Даний модуль забезпечує вимірювання змінної напруги в межах до 250 В і формує аналоговий вихідний сигнал із можливістю регулювання. Конструкція пристрою передбачає простоту експлуатації та оснащена багатооборотним потенціометром, який використовується для точного налаштування й калібрування сигналу, що подається на вхід аналого-цифрового перетворювача.

ESP32 є мікроконтролерною платформою з інтегрованою підтримкою бездротових технологій, зокрема Wi-Fi, а в окремих версіях — і Bluetooth. Завдяки цим можливостям вона активно використовується у системах Інтернету речей, де необхідна передача даних між пристроями або забезпечення підключення до мережі.

Незважаючи на те, що ESP32 є безпосередньо мікросхемою, у практичному застосуванні цей термін часто використовується також для позначення готових модулів і відлагоджувальних плат, створених на її основі, рис. 2.4 [12].

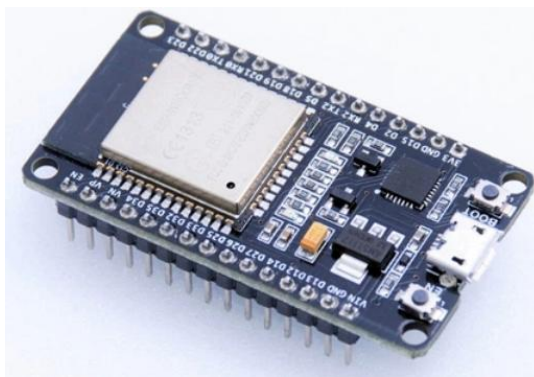


Рис.2.4 Мікроконтролерна платформа

Перші модифікації ESP32 базувалися на одноядерному процесорі Tensilica Xtensa LX6, тактова частота якого перевищувала 240 МГц, що забезпечувало достатній рівень продуктивності для задач обробки даних у вбудованих системах. Надалі було представлено нові покоління мікроконтролерів, зокрема серії ESP32-C та ESP32-S, які підтримують як одноядерні, так і двоядерні конфігурації. У цих версіях застосовується архітектура RISC-V, яка відрізняється відкритою специфікацією та більш гнучкими можливостями адаптації порівняно з Xtensa. Крім того, RISC-V добре інтегрується з сучасними компіляторами GNU, тоді як Xtensa потребує спеціалізованих інструментів підтримки.

Типова конфігурація ESP32 передбачає використання 32-бітного процесора Tensilica Xtensa LX6, зазвичай у двоядерному виконанні, за винятком окремих модифікацій, таких як ESP32-S0WD, що мають одне ядро. Максимальна тактова частота досягає 240 МГц, а обчислювальна продуктивність може становити до 600 DMIPS. Важливою перевагою платформи є енергоефективність, яка дозволяє виконувати окремі операції, зокрема аналого-цифрове перетворення та контроль порогових значень, навіть у режимі глибокого сну.

У сфері бездротового зв'язку ESP32 підтримує стандарти IEEE 802.11 (b/g/n/e/i), що забезпечує стабільне підключення до мережі. Також реалізовано підтримку Bluetooth версії 4.2 (BR/EDR), а в окремих варіантах — Bluetooth Low Energy (BLE), що робить платформу придатною для використання в енергоефективних IoT-системах [10].

Архітектура пам'яті ESP32 включає кілька типів внутрішніх ресурсів: постійну пам'ять (ROM) обсягом 448 КБ для завантажувальних функцій, оперативну пам'ять (SRAM) обсягом 520 КБ для виконання програм і зберігання даних, а також спеціалізовані області RTC Fast SRAM (8 КБ) і RTC Slow SRAM (8 КБ), призначені для збереження даних у режимах енергозбереження та роботи співпроцесора. Додатково використовується eFuse-пам'ять обсягом 1 Кбіт, яка містить системні параметри (наприклад, MAC-адресу) та може бути використана для користувацьких налаштувань. Деякі модифікації, такі як ESP32-D2WD і ESP32-PICO-D4, оснащені вбудованою флеш-пам'яттю.

Платформа також підтримує підключення зовнішньої QSPI флеш-пам'яті та SRAM обсягом до 16 МБ. Для забезпечення захисту даних реалізовано апаратне шифрування за допомогою алгоритму AES encryption, а доступ до зовнішньої пам'яті здійснюється через високошвидкісні кеш-механізми.

З погляду безпеки ESP32 відповідає сучасним вимогам, підтримуючи стандарти бездротового захисту, зокрема WPA, WPA/WPA2 та WAPI. Додатково реалізовано механізми безпечного завантаження (Secure Boot) і шифрування флеш-пам'яті, що дозволяє захистити програмний код і конфіденційні дані розробника.

Hi-Link HLK-PM01 — це компактний імпульсний модуль живлення, призначений для перетворення змінної мережевої напруги в діапазоні 100–240 В AC у стабілізовану постійну напругу 5 В DC.

Номінальна вихідна потужність модуля становить приблизно 3 Вт, що дозволяє використовувати його для живлення малопотужних електронних пристроїв, зокрема мікроконтролерів і вбудованих систем. Завдяки високому коефіцієнту корисної дії та компактним розмірам цей модуль широко застосовується у проєктах, де критичними є обмеження габаритів і стабільність електроживлення [10].

Модуль HLK-PM01 характеризується зручністю інтеграції у складі електронних схем та забезпечує надійне формування постійної напруги навіть за умов обмеженого простору для монтажу, рис. 2.5.



Рис.2.5 Імпульсний модуль живлення

Типові напрями використання Hi-Link HLK-PM01 охоплюють різноманітні електронні та вбудовані системи, у яких необхідне стабільне джерело постійної напруги.

Зокрема, даний модуль широко застосовується для живлення мікроконтролерних платформ, таких як Arduino, ESP8266, ESP32 та Raspberry Pi Pico. Він також активно використовується у пристроях Інтернету речей і системах «розумного дому», де ключове значення мають компактність конструкції та енергоефективність.

Крім того, модуль знаходить застосування у промислових системах автоматизації, схемах керування світлодіодним освітленням, а також для живлення малопотужних двигунів постійного струму. У окремих реалізаціях він може використовуватися в схемах заряджання акумуляторних батарей.

Завдяки малим габаритам і високому коефіцієнту корисної дії HLK-PM01 часто застосовується як альтернатива традиційним лінійним джерелам живлення, особливо в пристроях із обмеженим простором для монтажу.

Одноканальний релейний модуль є універсальним комутаційним елементом, призначеним для керування електричними колами з підвищеною напругою або струмом за допомогою низьковольтних керуючих сигналів, що надходять від мікроконтролерів чи інших систем керування.

Модуль працює від джерела постійної напруги 12 В і забезпечує керування одним каналом навантаження. Це дозволяє здійснювати ввімкнення або вимкнення електричних пристроїв відповідно до заданих умов або команд, рис. 2.6.

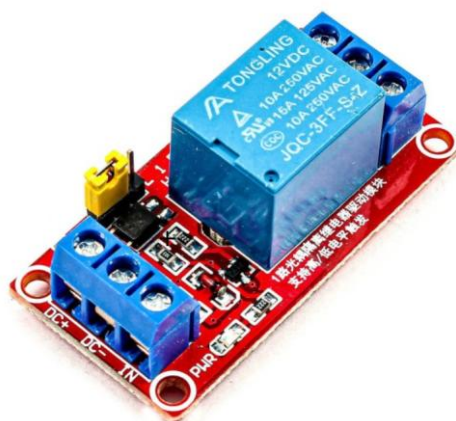


Рис.2.6 Релейний модуль

Однією з важливих функціональних характеристик модуля є можливість вибору логіки спрацювання — за високим або низьким рівнем керуючого сигналу, що реалізується за допомогою перемички на платі. Така особливість забезпечує гнучкість у використанні та сумісність із різними мікроконтролерними платформами, дозволяючи адаптувати пристрій до конкретних умов керування.

Релейний модуль здатний комутувати навантаження зі струмом до 10 А при напрузі 250 В змінного струму або до 30 В постійного струму. Це робить його придатним для керування широким спектром електричних пристроїв, зокрема системами освітлення, вентиляції, електродвигунами та іншими побутовими або промисловими установками.

З конструктивної точки зору модуль, як правило, оснащений клемною колодкою для підключення зовнішніх електричних ланцюгів, світлодіодним індикатором для відображення стану реле, а також захисним діодом, який компенсує вплив зворотної електрорушійної сили під час комутації індуктивних навантажень.

Завдяки своїм технічним характеристикам такі релейні модулі широко використовуються в системах автоматизації, робототехнічних рішеннях, «розумних» будинках та інших електронних проєктах, де необхідне дистанційне або автоматизоване керування електроприладами [10].

16×2 I2C РК-дисплей є символьним індикатором, призначеним для відображення текстової інформації у двох рядках по 16 символів у кожному. Використання інтерфейсу I2C значно спрощує підключення до мікроконтролера, оскільки передача даних здійснюється через обмежену кількість сигнальних ліній, рис. 2.7.

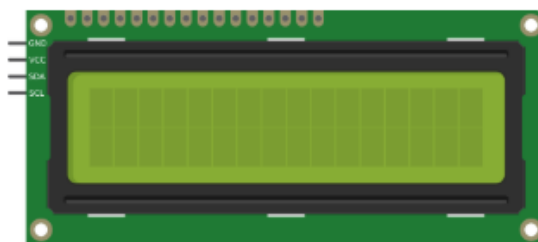


Рис. 2.7 Дисплей

Дисплеї даного типу широко використовуються у вбудованих системах і мікроконтролерних проєктах для візуалізації текстових повідомлень, показників сенсорів, службової інформації та інших даних у режимі реального часу. Простота підключення разом із універсальністю функціонування зумовлює їхню популярність як зручного засобу відображення інформації в різноманітних електронних пристроях.

PZEM-015 power meter є цифровим вимірювальним пристроєм, призначеним для моніторингу параметрів електричних кіл змінного струму в реальному часі. Модуль забезпечує визначення ключових електричних величин, зокрема напруги, сили струму, активної потужності, спожитої електроенергії та частоти мережі.

Завдяки своїм функціональним можливостям пристрій широко застосовується в системах енергоменеджменту, засобах промислової автоматизації, а також для контролю енергоспоживання у побутових умовах. Використання такого модуля дозволяє отримувати комплексну інформацію про стан електричних навантажень і здійснювати ефективний аналіз витрат електроенергії.

Компактність конструкції та простота інтеграції роблять PZEM-015 зручним рішенням для різноманітних застосувань, включаючи системи відновлюваної енергетики, електричні розподільчі щити та випробувальні стенди електрообладнання [10].



Рис.2.8 Цифровий вимірювальний модуль

2.2 Інтегроване середовище розробки Arduino IDE для програмування мікроконтролерів

На наступному етапі використовується компілятор, який виконує перетворення вихідного програмного коду у машинні інструкції або двійковий файл, придатний для виконання мікроконтролером. Окрім цього, необхідне спеціалізоване програмне забезпечення, що забезпечує запис сформованої прошивки безпосередньо в пам'ять пристрою [31].

Сукупність зазначених інструментів разом із додатковими можливостями, такими як засоби налагодження, робота з консоллю та інші сервісні функції, утворює інтегроване середовище розробки (IDE). Одним із таких рішень є Arduino IDE — програмний комплекс, призначений для створення, компіляції та завантаження програмного забезпечення на мікроконтролерні плати Arduino. До його складу входять редактор коду з розширеним функціоналом, компілятор, інструменти прошивки, а також засоби взаємодії через послідовний інтерфейс,

зокрема монітор порту та графічний плотер. Середовище характеризується простотою використання та доступністю для широкого кола користувачів.

Arduino IDE є кросплатформним програмним забезпеченням і підтримує роботу в різних операційних системах, зокрема Microsoft Windows, Linux та macOS. Розробка програм здійснюється мовою Arduino, синтаксис якої базується на мовах C/C++.

Процес встановлення середовища є відносно простим і не потребує спеціальних навичок. Спочатку необхідно завантажити актуальну версію програмного забезпечення з офіційного вебресурсу Arduino, обравши варіант, що відповідає використовуваній операційній системі. Наразі доступні дві основні версії середовища: класична гілка 1.x та оновлена 2.x.

Інсталяція виконується поетапно: після завантаження інсталяційного файлу його необхідно запустити, ознайомитися з умовами ліцензійної угоди та підтвердити їх, обрати параметри встановлення (для одного користувача або для всієї системи), а також визначити каталог розміщення програми. Після завершення інсталяції середовище стає готовим до подальшого використання, рис. 2.9.

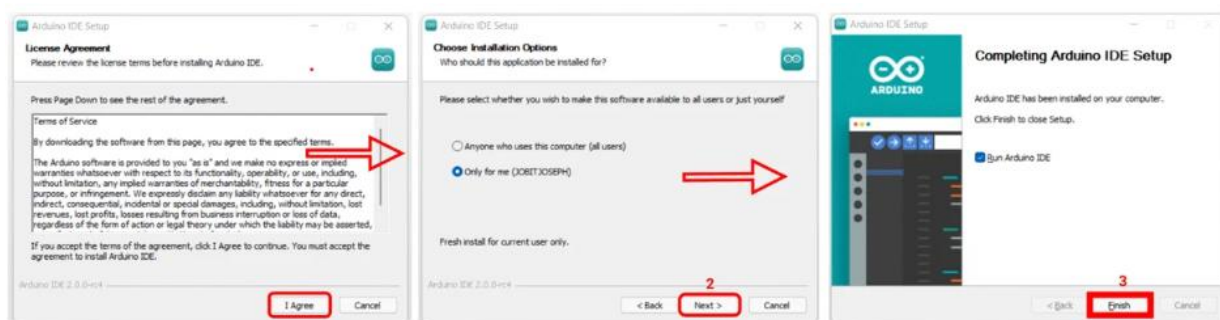


Рис.2.9 Процес інсталяції середовища розробки Arduino IDE

Для встановлення зв'язку між платою Arduino та персональним комп'ютером використовується стандартне USB-з'єднання: один кінець кабелю підключається до плати, інший — до відповідного порту ПК. Після підключення активується індикатор живлення, що свідчить про подачу напруги та коректну ініціалізацію

пристрою. У більшості випадків операційна система автоматично розпізнає підключене обладнання та встановлює необхідні драйвери.

Як приклад на рисунку 2.10 наведено підключення плати Arduino Uno. Для забезпечення коректної роботи з нею доцільно попередньо ознайомитися з її структурною організацією, зокрема з призначенням контактів і основними функціональними можливостями, що сприятиме кращому розумінню принципів взаємодії з апаратною частиною [31].

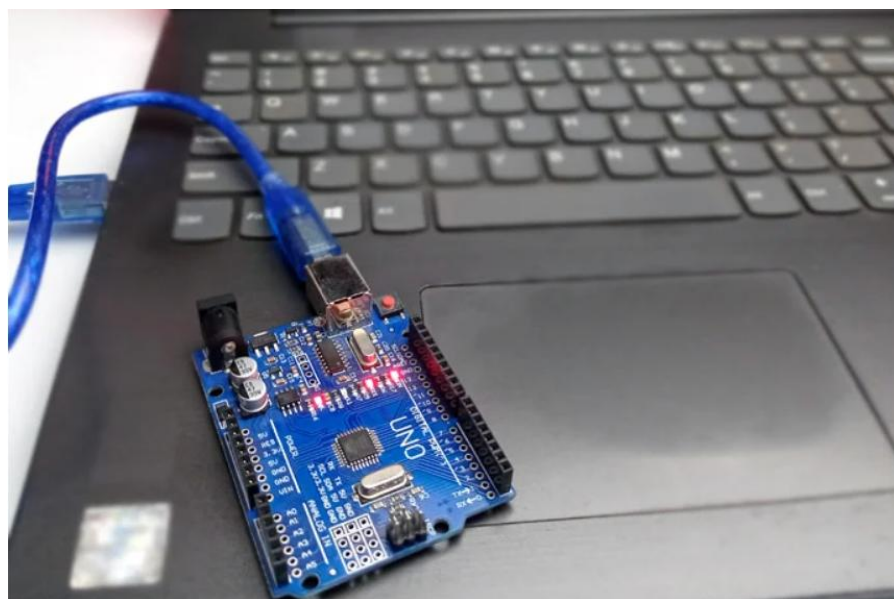


Рис.2.10 Підключення плати Arduino Uno

Після інсталяції Arduino IDE його запуск виконується стандартним способом - шляхом відкриття ярлика програми на робочому столі або через меню «Пуск». Після запуску користувачу стає доступним головне вікно середовища, яке забезпечує доступ до основних інструментів для створення, редагування та завантаження програмного коду на мікроконтролерні плати.

Інтерфейс Arduino IDE включає панель інструментів, область редагування коду, консоль повідомлень, а також інші елементи, необхідні для написання програм і взаємодії з підключеним пристроєм. Зовнішній вигляд середовища наведено на відповідному рисунку.

Одним із ключових етапів підготовки до роботи є вибір цільової апаратної платформи. Ця процедура є необхідною перед компіляцією, оскільки від обраного типу плати залежать параметри збірки та визначається архітектура, для якої буде згенеровано виконуваний код.

Для налаштування слід у меню «Інструменти» обрати пункт «Плата», перейти до розділу «Плати Arduino AVR» і зі списку доступних варіантів вибрати відповідну модель, наприклад Arduino Uno. Коректне встановлення цього параметра забезпечує правильну компіляцію програми та її подальше завантаження на обрану плату, рис. 2.11.

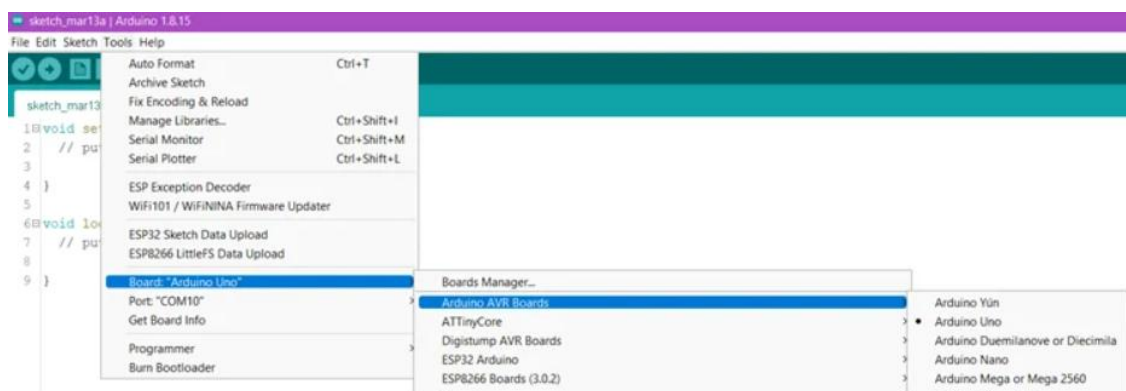


Рис.2.11 Вибір плати

Важливим етапом конфігурації є правильний вибір послідовного порту, через який здійснюється обмін даними між комп'ютером і платою Arduino. У разі некоректного визначення цього параметра передавання програмного коду на мікроконтролер стає неможливим.

Після відкриття прикладу в середовищі Arduino IDE користувач отримує доступ до типової структури програми, зокрема до прикладу Blink. У редакторі коду відображаються дві базові функції, які лежать в основі більшості програм для плат Arduino [31].

Перша з них — функція `void setup()`, яка виконується один раз після подачі живлення або перезапуску мікроконтролера Arduino Uno. У цьому блоці зазвичай

здійснюється початкова ініціалізація, включаючи налаштування режимів роботи виводів, запуск периферійних модулів та конфігурацію системних параметрів.

Друга — функція `void loop()`, яка працює у безперервному циклі протягом усього часу функціонування пристрою. Саме в цьому блоці реалізується основна логіка програми: зчитування даних із сенсорів, керування виконавчими елементами та виконання інших операцій, що потребують циклічного повторення, рис. 2.12.

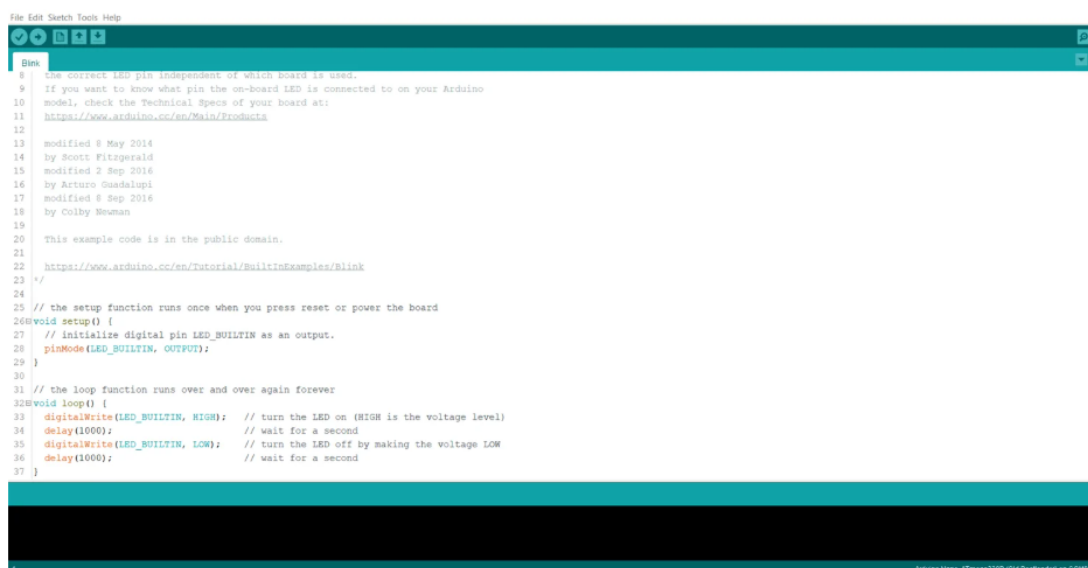


Рис. 2.12 Прикладом коду blink

Як зазначалося раніше, функція `void setup()` призначена для виконання початкової ініціалізації програми. У цьому блоці задаються всі необхідні параметри, що забезпечують коректний старт роботи мікроконтролера Arduino Uno.

Зокрема, у функції `setup()` здійснюється конфігурація режимів роботи виводів, ініціалізація підключених периферійних модулів, а також виконуються інші початкові налаштування, необхідні для стабільного функціонування системи, рис. 2.13.

```
pinMode(LED_BUILTIN, OUTPUT);
```

Рис.2.13 Налаштування виводів

У наведеному фрагменті коду за допомогою функції `pinMode()` виконується конфігурація режиму роботи визначеного виводу мікроконтролера, до якого підключений вбудований світлодіод, у режим виходу. Дана функція використовується для встановлення ролі піна — як вхідного (INPUT) або вихідного (OUTPUT).

Загальний формат виклику має вигляд: `pinMode(pin, mode)`, де параметр `pin` визначає номер відповідного контакту мікроконтролера, а `mode` задає режим його функціонування. У даному випадку використовується контакт №13, який на платі Arduino Uno відповідає вбудованому світлодіоду. Значення цього піна вже визначене у системних конфігураційних файлах плати, що дозволяє використовувати його без додаткових налаштувань.

```
void loop () {  
  digitalWrite (LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)  
  delay(1000);                        //wait for a second  
  digitalWrite(LED_BUILTIN, LOW);    //turn the LED off by making the voltage Low  
  delay(1000);                        // wait for a second  
}
```

Рис.2.14 Приклад функції `loop()` в Arduino

Для запису програмного коду в пам'ять мікроконтролера та перевірки правильності налаштувань (зокрема підключення плати, вибору моделі та послідовного порту) необхідно виконати компіляцію і передачу прошивки через середовище Arduino IDE.

Запуск цієї процедури можливий кількома способами: за допомогою кнопки завантаження на панелі інструментів, через меню «Ескіз» → «Завантажити» або використовуючи комбінацію клавіш `Ctrl+U`. Після ініціації процесу програмний код спочатку проходить етап компіляції, після чого автоматично передається на підключену плату Arduino Uno через обраний COM-порт.

У разі коректного виконання всіх налаштувань прошивка успішно завантажується, після чого програма починає виконуватися безпосередньо на пристрої [31].



Рис. 2.15 Завантаження програмного коду

Після успішного завершення всіх етапів завантаження у рядку стану середовища Arduino IDE відображається повідомлення «Завантаження завершено», що свідчить про коректну передачу програмного коду на мікроконтролер. Після цього на платі Arduino Uno починає виконуватися завантажений скетч, у результаті чого, відповідно до прикладу, вбудований світлодіод переходить у режим періодичного миготіння.

Таким чином, користувач отримує підтвердження правильності виконаних налаштувань, зокрема вибору плати, визначення послідовного порту та успішної компіляції, що свідчить про коректне функціонування системи в цілому.

2.3 Програмні інструменти для створення мобільних застосунків в IoT-системах

У сфері сучасної електроніки поширеною практикою є створення мобільних застосунків для керування пристроями, що підключаються через бездротові інтерфейси, зокрема Wi-Fi або Bluetooth. Такі рішення дозволяють користувачу здійснювати контроль і моніторинг обладнання безпосередньо зі смартфона. Для розробки подібних застосунків використовуються як нативні, так і кросплатформні

інструменти, серед яких React Native, Flutter, Xamarin, Ionic, Apache Cordova та інші.

Разом з тим, використання таких фреймворків передбачає повноцінну розробку програмного забезпечення, що вимагає значних витрат часу, відповідних технічних знань і практичного досвіду. Як альтернатива традиційному програмуванню застосовуються готові візуальні середовища, інтегровані в IoT-платформи. У цьому випадку створення інтерфейсу здійснюється без написання коду, шляхом налаштування готових елементів і визначення логіки їх взаємодії з апаратною частиною системи, що значно спрощує процес розробки та прискорює впровадження рішення.

Подібні візуальні інструменти, незважаючи на відносно обмежений функціонал, є достатніми для реалізації прикладних рішень у сфері Інтернету речей, домашньої автоматизації та інших електронних систем, що взаємодіють через Wi-Fi або Bluetooth і керуються зі смартфона. Вони забезпечують можливість швидкого створення інтерфейсу користувача, а також налаштування основної логіки роботи застосунку без необхідності повноцінного програмування.

Завдяки цьому розробник може зосередитися на організації взаємодії між апаратною та програмною частинами системи, використовуючи готові елементи для формування інтерфейсу та визначення функціональних можливостей застосунку.

У цьому підрозділі проаналізовано поширені інструменти, призначені для створення візуальних мобільних застосунків, що активно використовуються розробниками електронних систем.

1. MIT App Inventor є одним із найбільш популярних середовищ розробки мобільних додатків, яке базується на візуальній моделі програмування. Спочатку платформа була створена компанією Google, а на сьогодні її розвиток і підтримка здійснюються Massachusetts Institute of Technology. Основною особливістю є використання блокового підходу до програмування, що дозволяє розробляти повнофункціональні застосунки для платформ Android та iOS без написання

традиційного текстового коду. Програмне забезпечення є безкоштовним і має відкритий вихідний код, який поширюється за ліцензіями Apache License 2.0 та Creative Commons Attribution ShareAlike 3.0 [32].

Процес створення застосунків у MIT App Inventor характеризується простотою та інтуїтивністю, що робить його доступним навіть для користувачів без значного досвіду програмування. Основи роботи з платформою можна опанувати за короткий час. Для оперативного тестування застосунків використовується допоміжний інструмент App Inventor Companion, який дозволяє перевіряти функціональність без встановлення програмного продукту на пристрій. Середовище реалізоване у вигляді веб-платформи та підтримує роботу через емулятори в операційних системах Windows, macOS і Linux.

Функціональні можливості MIT App Inventor охоплюють широкий спектр застосувань. Зокрема, платформа може використовуватися для розробки рішень у сфері Інтернету речей і домашньої автоматизації, керування електронними пристроями через мережу Інтернет, створення геоінформаційних сервісів із використанням картографічних даних, розробки мобільних ігор, а також реалізації задач обробки інформації, включаючи використання методів штучного інтелекту для аналізу текстових, графічних і аудіоданих.

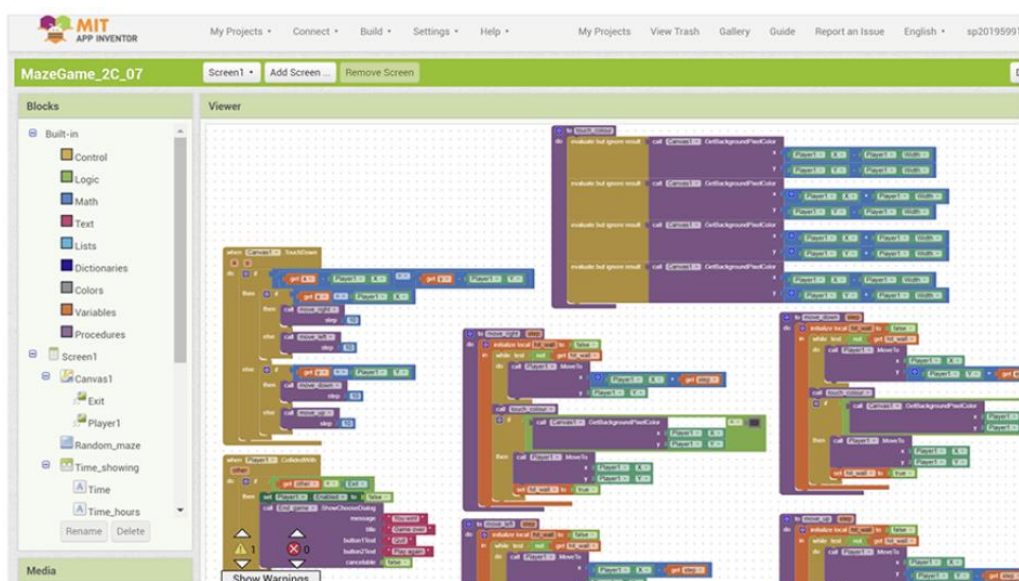


Рис.2.16 MIT App Inventor

Високий рівень популярності платформи підтверджується значною кількістю реалізованих проєктів і активною спільнотою користувачів: за допомогою MIT App Inventor створено мільйони застосунків, а кількість користувачів, які регулярно працюють із середовищем, становить сотні тисяч щомісяця, що свідчить про її актуальність і практичну значущість у сучасній мобільній розробці [32].

RAD Studio — це програмне середовище розробки, створене компанією Embarcadero Technologies, яке призначене для побудови кросплатформних застосунків для мобільних і настільних систем. Воно дозволяє створювати програмні рішення, що функціонують на платформах iOS, Android, macOS, Windows і Linux, використовуючи єдину кодову базу [33].

Однією з ключових особливостей середовища є підтримка візуальної розробки з використанням бібліотек VCL та FireMonkey. Завдяки вбудованим майстрам і графічним інструментам користувач може формувати адаптивні інтерфейси, оптимізовані як для мобільних пристроїв, так і для персональних комп'ютерів. Попри наявність елементів низькорівневої розробки, для ефективного використання платформи необхідні базові знання мов програмування C++ або Delphi, хоча загальний процес створення застосунків є значно спрощеним порівняно з традиційним підходом.

Платформа забезпечує широку підтримку сучасних технологій і стандартів. Зокрема, для Android реалізована сумісність із відповідними API та бібліотеками, а для iOS і macOS використовується 64-бітна ARM-архітектура. У середовищі Windows підтримуються VCL-застосунки з можливістю роботи з WinRT API та технологіями високої роздільної здатності (HighDPI). Для Linux передбачена підтримка графічних інтерфейсів і бібліотек FireMonkey.

Окремою перевагою є інтеграція з різноманітними системами керування базами даних, серед яких MySQL, PostgreSQL, SQLite, SQL Server і MongoDB, а також можливість роботи з REST-сервісами та хмарними платформами через HTTPS. Це дозволяє створювати складні інформаційні системи з використанням мережевих сервісів та розподіленої архітектури.

Завдяки використанню єдиної кодової бази та широкому набору інструментів, RAD Studio є ефективним рішенням для розробки продуктивних мобільних застосунків, у тому числі для систем Інтернету речей, де необхідна інтеграція з базами даних, веб-сервісами та різноманітними електронними пристроями [33].

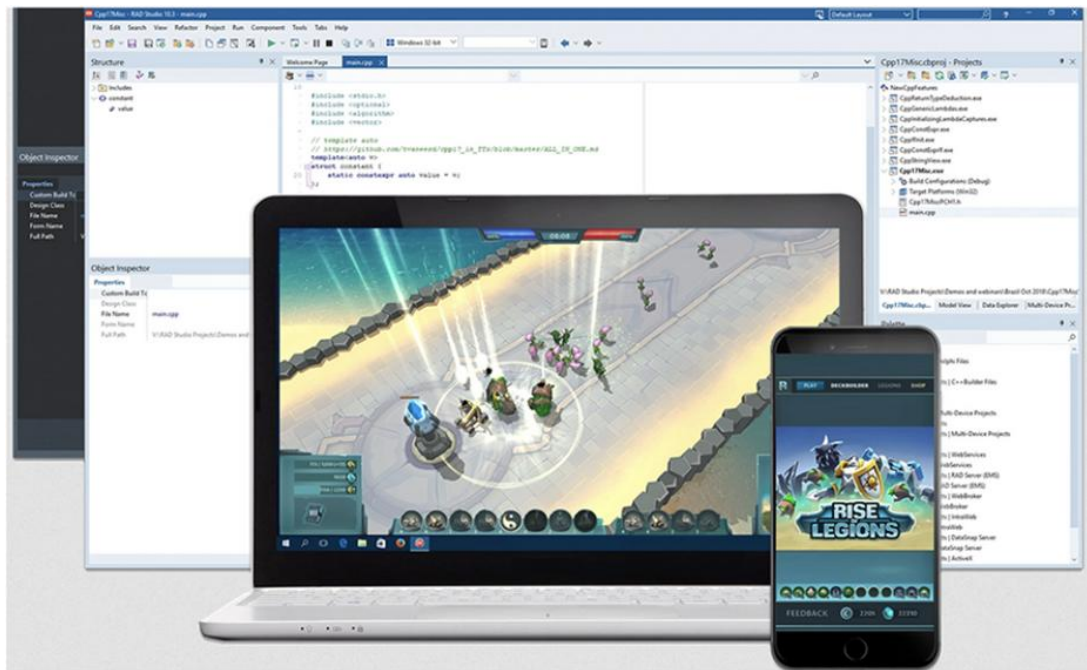


Рис.2.17 RAD Studio

Cayenne IoT platform від myDevices позиціонується як інструмент для швидкої розробки рішень у сфері Інтернету речей із використанням принципу візуального конструювання. Платформа орієнтована на створення IoT-систем різного рівня складності та забезпечує сумісність із широким спектром пристроїв, включаючи мікроконтролери та одноплатні комп'ютери [34].

Cayenne реалізована у вигляді веб-середовища з можливістю налаштування інтерфейсу користувача. Вона підтримує функції моніторингу, керування та автоматизації, включаючи роботу з тригерами, системою сповіщень, відстеженням об'єктів, а також інструменти для візуалізації даних. Додатково платформа інтегрує протоколи обміну даними, зокрема MQTT, і підтримує технології зв'язку, такі як LoRaWAN, що розширює можливості її застосування у розподілених системах.

Користувачі можуть створювати індивідуальні панелі керування за допомогою віджетів, використовуючи механізм «перетягування» (drag-and-drop). Це дозволяє організувати зручний інтерфейс для відображення інформації та взаємодії з підключеними пристроями без необхідності глибокого програмування. Доступ до панелей керування здійснюється як через веб-інтерфейс, так і за допомогою мобільних застосунків.

Залежно від функціональних можливостей конкретного обладнання, кожен пристрій або датчик може бути представлений на панелі керування кількома віджетами, що забезпечує гнучкість у налаштуванні системи моніторингу та керування.

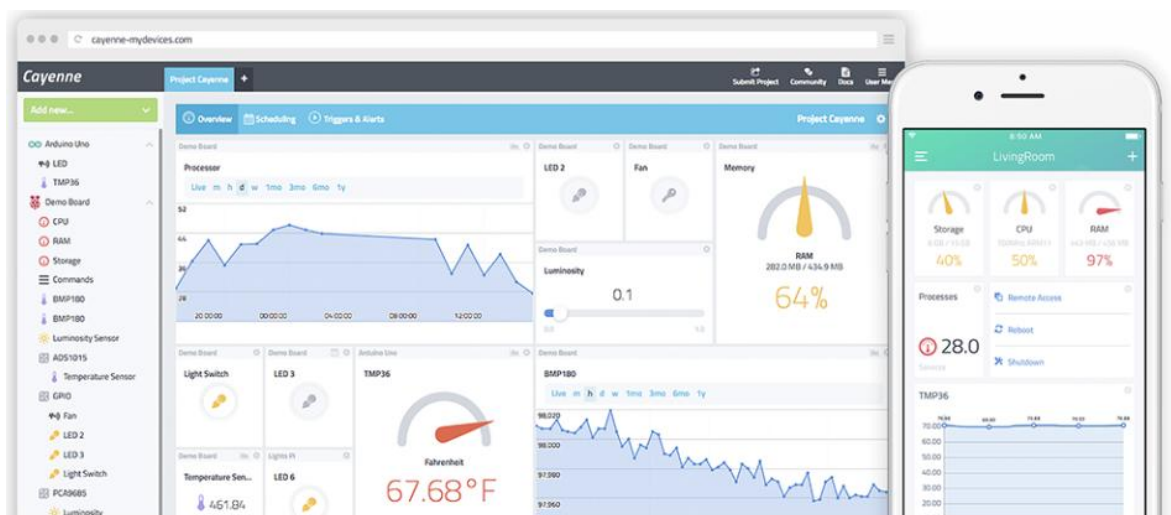


Рис.2.18 Cayenne IoT platform від myDevices

Blynk IoT — це комплексна платформа Інтернету речей, яка забезпечує можливість дистанційного керування та моніторингу електронних пристроїв на базі мікроконтролерів, зокрема Arduino, ESP та Raspberry Pi. Вона являє собою інтегроване програмне рішення, що включає мобільний застосунок, хмарну інфраструктуру та інструменти для розробки й керування пристроями [35].

Для взаємодії з апаратною частиною користувачеві необхідно встановити мобільний застосунок Blynk на пристрій під керуванням iOS або Android. Через нього здійснюється візуалізація даних, отриманих із датчиків, а також передача

команд до підключених пристроїв. Після реєстрації облікового запису користувач отримує можливість створювати власні панелі керування, використовуючи віджети з механізмом «перетягування», що значно спрощує процес налаштування інтерфейсу.

Обмін даними між мобільним застосунком і пристроями відбувається через серверну частину платформи, яка може бути реалізована як у вигляді хмарного сервісу, так і локально розгорнутого сервера. Така архітектура дозволяє керувати великою кількістю пристроїв із єдиного інтерфейсу. Для забезпечення зв'язку використовуються різні канали передачі даних, зокрема Wi-Fi, Ethernet, Bluetooth або мобільні мережі, що реалізуються за допомогою відповідних бібліотек.

Платформа належить до класу low-code рішень: вона значно спрощує розробку, проте потребує написання програмного коду для реалізації логіки роботи пристроїв. Це робить її більш придатною для створення функціонально складних або масштабованих IoT-проектів. Для задач, що не потребують програмування, можуть застосовуватися альтернативні платформи з повністю візуальним підходом до розробки, такі як Cayenne IoT platform [35].

Хоча платформа Blynk забезпечує надійний та зручний інтерфейс для систем управління енергією, необхідно врахувати кілька обмежень:

1. Масштабованість: Blynk розроблений в першу чергу для малих та середніх проектів. Зі збільшенням масштабу системи управління енергією, наприклад, у великих промислових умовах або міській інфраструктурі, платформа може зіткнутися з проблемами продуктивності та стабільності. Інфраструктура, необхідна для підтримки великої кількості пристроїв та пов'язаного з ними трафіку даних, може перевищувати поточні можливості платформи.

2. Проблеми безпеки. Хоча Blynk пропонує основні функції безпеки, такі як SSL-шифрування, він може не відповідати суворим вимогам безпеки, необхідним для критичної інфраструктури. Системи управління енергоспоживанням часто стають мішенями для кібератак, і заходи безпеки платформи можуть потребувати посилення додатковими рівнями захисту для захисту від складних загроз.

3. Інтеграція з існуючими системами. Інтеграція Blynk з існуючими системами управління енергією та автоматизації може бути складною. Застарілі системи можуть використовувати різні протоколи та стандарти, які не підтримуються Blynk безпосередньо, що вимагає розробки користувацьких інтерфейсів або проміжного програмного забезпечення, що може бути як трудомістким, так і дорогим.

4. Обробка даних у режимі реального часу. Хоча Blynk забезпечує можливості моніторингу в режимі реального часу, швидкість та ефективність обробки даних можуть бути обмеженням для програм, що потребують миттєвої реакції. Високочастотні потоки даних можуть зіткнутися з проблемами затримки, що впливає на ефективність процесів прийняття рішень у режимі реального часу в управлінні енергією.

5. Обмежена можливість налаштування для досвідчених користувачів. Інтерфейс Blynk розроблений з урахуванням зручності використання, що іноді обмежує можливості налаштування для досвідчених користувачів. Розширені функції або складні алгоритми, необхідні для складних рішень з управління енергією, можуть бути складними для реалізації в рамках попередньо визначеної структури платформи.

6. Залежність від підключення до Інтернету. Платформа Blynk значною мірою залежить від стабільного підключення до Інтернету. У середовищах, де доступ до Інтернету ненадійний або переривчастий, продуктивність системи управління енергоспоживанням може серйозно постраждати. Ця залежність створює значний ризик для постійного моніторингу та контролю у віддалених або слаборозвинених районах.

7. Міркування щодо вартості. Для масштабних впроваджень витрати, пов'язані з використанням Blynk, включаючи хмарні сервіси та абонентську плату, можуть стати значними. Це може обмежити доцільність впровадження Blynk для масштабних систем управління енергією, особливо у секторах, чутливих до витрат.

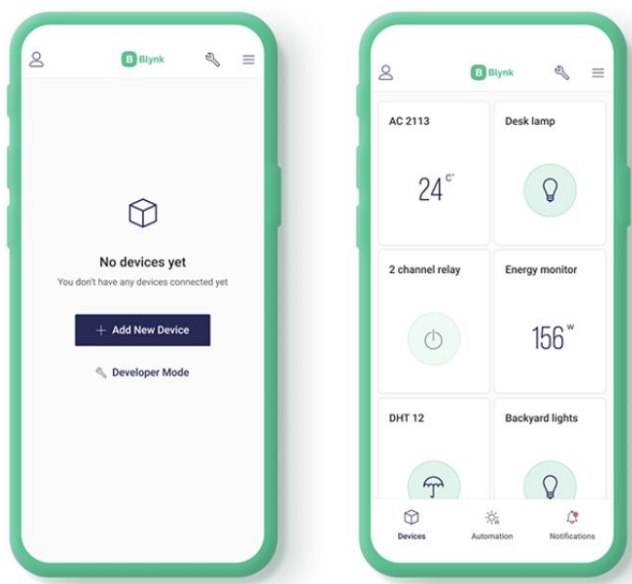


Рис.2.19 Blynk IoT

ThingsBoard - це відкрита платформа Інтернету речей, призначена для керування підключеними пристроями, збору телеметричних даних та їх подальшої обробки й візуалізації. Вона надає можливість створювати інтерактивні панелі моніторингу в режимі реального часу, що дозволяє як відображати стан системи, так і здійснювати дистанційне керування обладнанням. Для побудови інтерфейсу доступний набір віджетів із підтримкою механізму «перетягування», що спрощує процес налаштування панелей керування [36].

Платформа підтримує роботу з різними типами баз даних, включаючи реляційні (SQL), нереляційні (NoSQL) та комбіновані рішення, що забезпечує гнучкість у зберіганні та обробці великих обсягів даних. Однією з важливих функціональних можливостей є механізм обробки подій, який дозволяє створювати складні логічні правила для аналізу даних, що надходять від підключених пристроїв.

Передача інформації в системі здійснюється з використанням захищених протоколів, зокрема HTTPS та MQTT, що гарантує належний рівень безпеки. Також реалізовано механізми автентифікації пристроїв і керування обліковими даними. Для взаємодії з обладнанням застосовується модель віддаленого виклику процедур (RPC), яка забезпечує керування пристроями в реальному часі.

Серед ключових можливостей платформи варто відзначити збір і обробку телеметрії, візуалізацію даних, підтримку багатокористувацького доступу, горизонтальне масштабування, управління пристроями та активами, систему сповіщень і тривог, а також засоби забезпечення безпеки та шифрування. Використання мікросервісної архітектури дозволяє ефективно масштабувати систему та забезпечує стабільну роботу навіть при значному зростанні кількості підключених пристроїв.

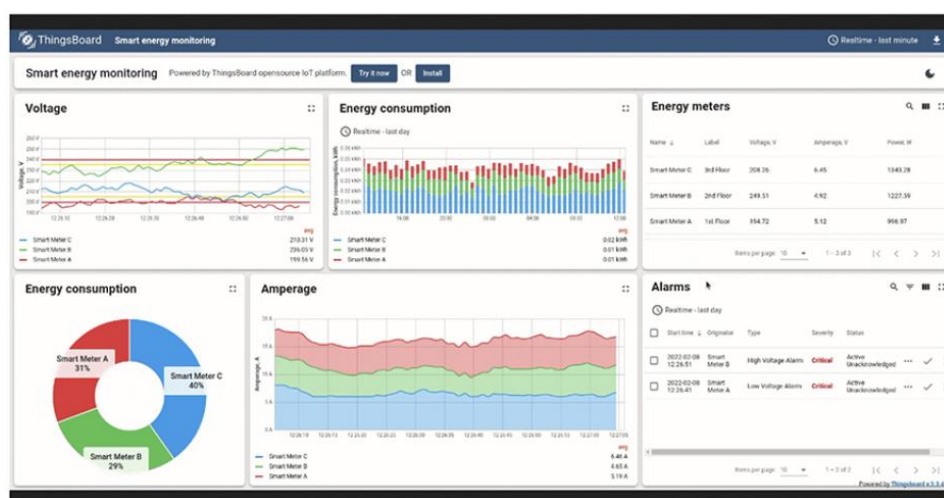


Рис.2.20 ThingsBoard

Сучасні платформи для створення мобільних застосунків дедалі частіше орієнтуються на спрощення процесу розробки та зменшення потреби у традиційному програмуванні. Одним із прикладів є Appy Pie, яка використовує технології штучного інтелекту для формування кросплатформних застосунків без написання коду. Вона базується на візуальному конструюванні інтерфейсу за допомогою елементів «перетягування», готових шаблонів та налаштовуваних стилів оформлення. Додатково платформа дозволяє задавати логіку роботи застосунку через систему тригерів, що робить її зручною для користувачів без технічної підготовки.

Іншим представником low-code підходу є Zoho Creator. Це середовище надає можливість створювати веб- та мобільні застосунки за допомогою візуальних

інструментів і майстрів налаштування. Окрім побудови інтерфейсу, платформа підтримує автоматизацію бізнес-процесів та інтеграцію елементів штучного інтелекту без необхідності глибокого програмування. Важливою особливістю є можливість створення власних моделей даних та їх візуалізації, а також виконання аналітики з різних джерел інформації.

До подібних рішень також належать BuildFire та Adalo, які орієнтовані на швидке створення мобільних застосунків для iOS та Android. Вони використовують концепцію візуальної розробки з готовими компонентами та шаблонами, що дозволяє швидко формувати інтерфейс користувача та тестувати результат у реальному часі. Зокрема, Adalo додатково підтримує роботу з базами даних у вигляді електронних таблиць, що спрощує організацію інформаційної структури застосунку.

Узагальнюючи проведений аналіз, слід відзначити, що одним із найпоширеніших підходів до керування пристроями Інтернету речей є використання мобільних застосунків, які забезпечують зручний інтерфейс взаємодії між користувачем та підключеним обладнанням. Водночас процес розробки таких застосунків для виробників електроніки часто є складним і ресурсомістким, оскільки традиційні підходи передбачають значний обсяг програмування та необхідність володіння спеціалізованими технологіями мобільної розробки.

Сучасні IoT-платформи частково вирішують цю проблему, пропонуючи інструменти візуальної розробки, які дозволяють створювати кросплатформні мобільні застосунки без необхідності глибокого програмування. Зокрема, рішення на зразок MIT App Inventor, Blynk IoT, Cayenne IoT platform та ThingsBoard є ефективними для реалізації систем, що базуються на хмарній інфраструктурі та віддаленому моніторингу пристроїв.

Для більш складних програмних рішень, які потребують глибшої інтеграції з апаратною частиною або підтримки різних операційних систем, доцільним є

використання середовищ типу RAD Studio, що забезпечують створення кросплатформних застосунків із єдиною кодовою базою.

Окрему групу становлять універсальні no-code та low-code платформи, такі як Appy Pie, BuildFire, Zoho Creator та Adalo, які орієнтовані на швидке створення мобільних застосунків без необхідності програмування. Вони є доцільними для реалізації відносно простих рішень, прототипів або локальних систем керування, де не потрібна складна мережева взаємодія.

Таким чином, сучасні інструменти розробки мобільних застосунків для IoT демонструють чітку тенденцію до спрощення процесу створення програмного забезпечення, розширення доступності технологій та зменшення вимог до рівня програмної підготовки розробника.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ЕЛЕКТРОСПОЖИВАННЯ

У даному розділі проведено дослідження процесу розроблення інтелектуального лічильника електроенергії на основі технологій Інтернету речей із використанням ESP32 та сучасного мобільного застосунку Blynk 2.0.

Традиційні методи контролю енергоспоживання передбачають ручне зчитування показників лічильника, що є трудомістким і не завжди зручним. Використання IoT-технологій дозволяє автоматизувати процес збору даних і забезпечити їх дистанційну передачу, що сприяє економії часу та ресурсів. У зв'язку з цим концепція інтелектуальних лічильників електроенергії останніми роками набула широкого поширення у світі, відкриваючи можливості для створення ефективних власних рішень. Для реалізації системи використовуються датчик струму SCT-013 та датчик напруги ZMPT101B, що забезпечують вимірювання основних електричних параметрів, зокрема напруги, сили струму, потужності та сумарного енергоспоживання (кВт·год). Отримані дані передаються до застосунку Blynk 2.0, де вони відображаються у вигляді інформаційної панелі з можливістю доступу з будь-якого місця. З метою забезпечення надійності системи передбачено збереження даних у внутрішній пам'яті EEPROM мікроконтролера ESP32 у разі втрати живлення, що дозволяє уникнути втрати накопичених показників. На початковому етапі реалізації системи передбачається автоматизація процесу моніторингу споживання електроенергії.

3.1 Проєктування електричної схеми та налаштування апаратного забезпечення

На рис. 3.1 наведено принципову електричну схему інтелектуального лічильника електроенергії, реалізованого на основі технологій Інтернету речей із

використанням ESP32. Схема була розроблена з використанням програмного середовища Fritzing та характеризується відносною простотою реалізації й підключення компонентів.

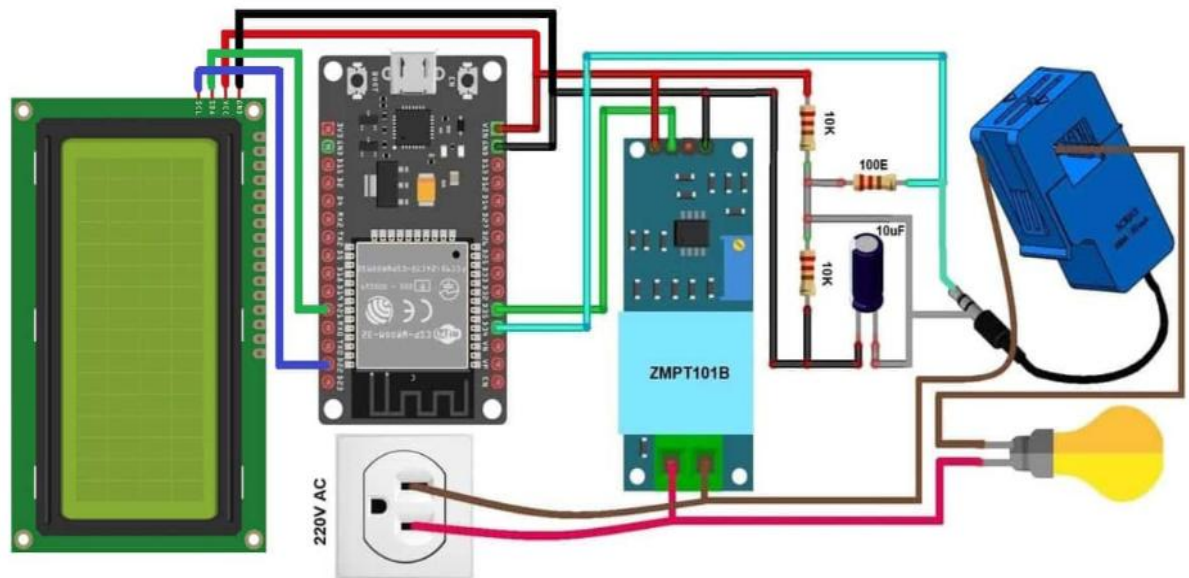


Рис. 3.1 Схема лічильника електроенергії

Контакти живлення VCC і GND датчиків струму SCT-013 та напруги ZMPT101B підключаються до виводів Vin і GND мікроконтролера ESP32, який забезпечує напругу живлення 5 В.

Аналоговий вихід датчика напруги ZMPT101B під'єднується до входу GPIO35, тоді як аналоговий вихід датчика струму SCT-013 з'єднується з входом GPIO34 мікроконтролера.

Для коректної роботи схеми також використовуються допоміжні пасивні компоненти, зокрема два резистори номіналом 10 кОм, один резистор на 100 Ом та конденсатор ємністю 10 мкФ.

На рисунку 3.2 представлено модель, яка описує принцип роботи системи.

Алгоритм функціонування розумного лічильника

1. Після подачі живлення виконується запуск ESP32.
2. Мікроконтролер ініціалізує:

- LCD дисплей;
 - Wi-Fi модуль;
 - ADC;
 - датчики напруги та струму.
3. ESP32 підключається до бездротової мережі Wi-Fi.
 4. Датчик ZMPT101B вимірює напругу мережі.
 5. SCT-013 вимірює струм навантаження.
 6. Аналогові сигнали надходять на ADC ESP32.
 7. Виконується цифрова обробка сигналів:
 - обчислення RMS напруги;
 - обчислення RMS струму.
 8. Розраховується миттєва потужність:

$$P = U \times I \quad (3.1)$$

9. Обчислюється спожита електроенергія:

$$E = P \times t \quad (3.2)$$

10. Дані відображаються на LCD дисплеї.
11. Інформація передається через Wi-Fi у Blynk; Web Server або хмарну базу даних.
12. Система аналізує значення споживання.
13. Якщо значення перевищує допустимий поріг надсилається повідомлення користувачу та/або активується звукова або світлова сигналізація.
14. Після завершення циклу система повторює вимірювання.

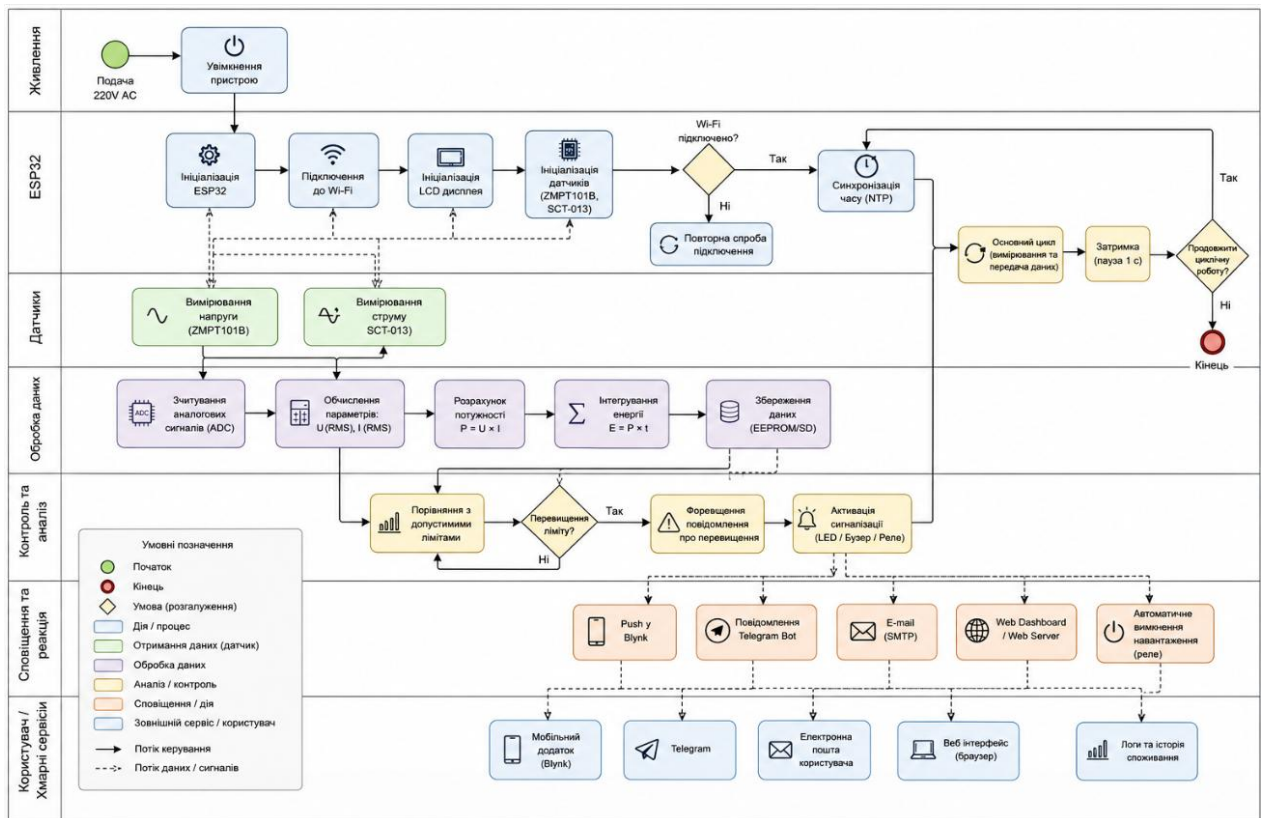


Рис.3.2 Модель, яка описує принцип роботи системи

Для здійснення вимірювання параметрів змінного струму провідники мережі необхідно під'єднати до вхідних клем датчика напруги. У випадку використання датчика струму достатньо пропустити лише один провід (фазний або нульовий) через вимірювальний затискач.

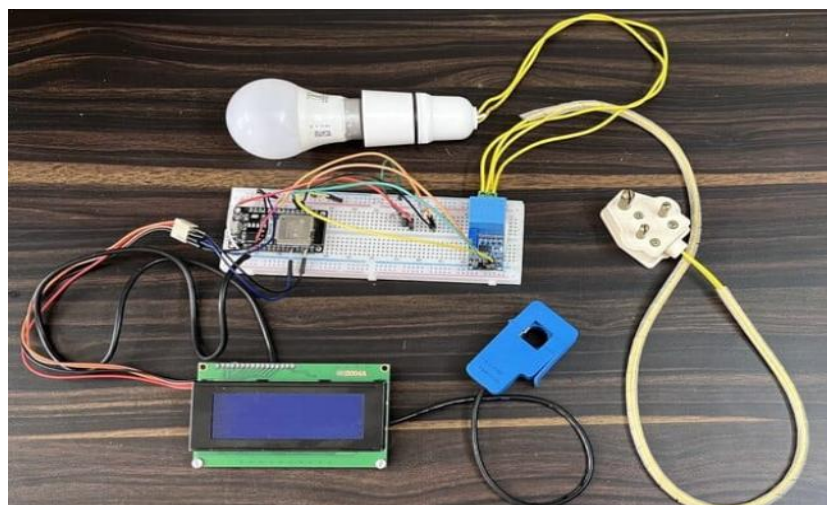


Рис. 3.3 Під'єднання обладнання

За необхідності до системи може бути додано символьний РК-дисплей із інтерфейсом I2C формату 20×4, що потребує додаткових з'єднань. Для його підключення виводи VCC і GND дисплея з'єднуються з відповідними контактами 5V і GND мікроконтролера ESP32, а сигнальні лінії SDA та SCL підключаються до GPIO21 і GPIO22 відповідно.

Регулювання контрастності зображення здійснюється за допомогою підлаштувального потенціометра номіналом 10 кОм, розташованого на платі I2C-адаптера дисплея.

Водночас варто зазначити, що моніторинг даних, отриманих від системи на базі ESP32, датчиків SCT-013 і ZMPT101B, може виконуватися безпосередньо через застосунок Blynk 2.0, що дозволяє відмовитися від використання локального дисплея.

3.2 Інтеграція та налаштування користувацького інтерфейсу в Blynk 2.0

Як згадувалось раніше, Blynk є мобільний застосунок для пристроїв на базі Android та iOS, який використовується для віддаленого керування IoT-рішеннями через смартфон. Платформа дає змогу розробляти власний графічний інтерфейс користувача для Інтернету речей та взаємодіяти з підключеними пристроями в режимі реального часу.

У межах даного проєкту передбачається відображення показників лічильника електроенергії в IoT-системі як у веб-інтерфейсі, так і в мобільному застосунку Blynk.

Для початку роботи необхідно перейти на офіційний сайт blynk.io та пройти реєстрацію, використовуючи електронну пошту користувача. Після створення облікового запису виконується створення нового шаблону, який надалі буде використовуватися для налаштування та відображення даних системи.

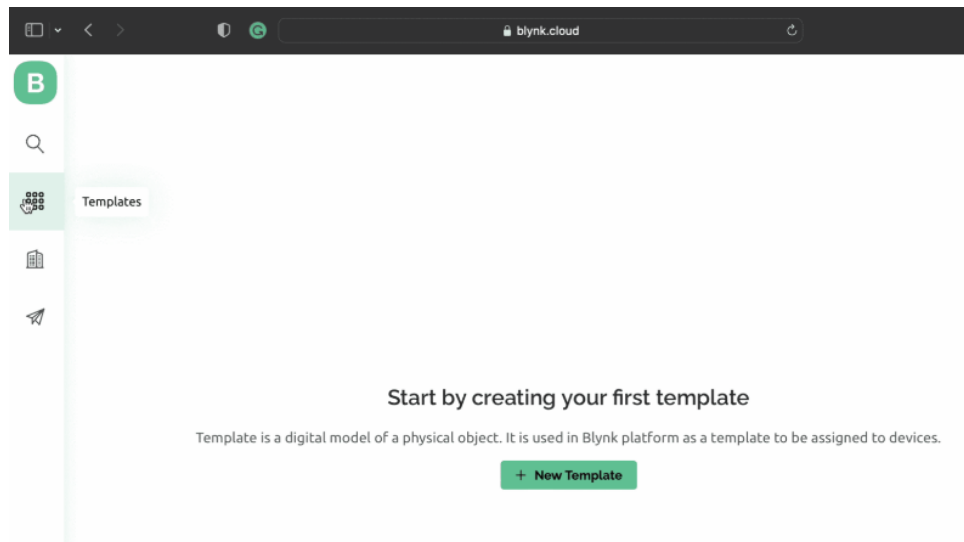


Рис.3.4 Створення шаблону

Присвоюємо назву проєкту, обираємо відповідне обладнання та визначаємо тип підключення.

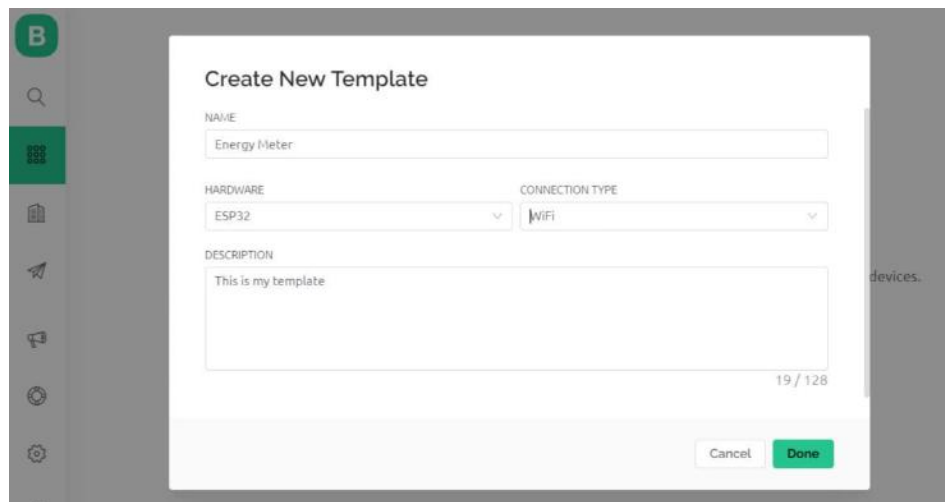


Рис.3.5 Призначення назви, обладнання та тип підключення

На веб-панелі керування створюємо 4 віджети Gauge.

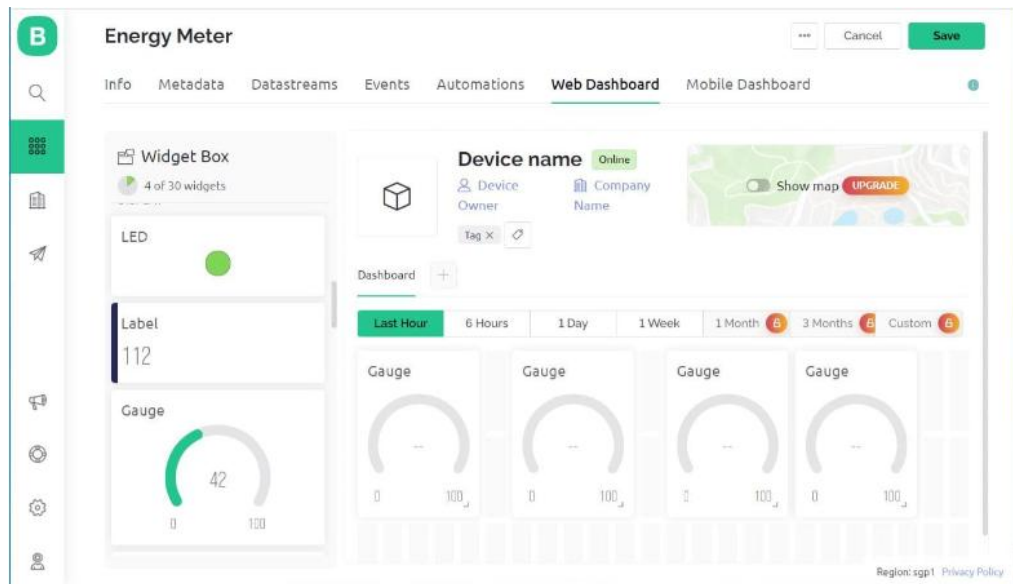


Рис.3.6 Створення віджету Gauge

Тут розташовані 4 віджети для відображення значень V_{rms} , I_{rms} , Power та KWh. Далі виконуємо налаштування, як показано на рис.3.7.

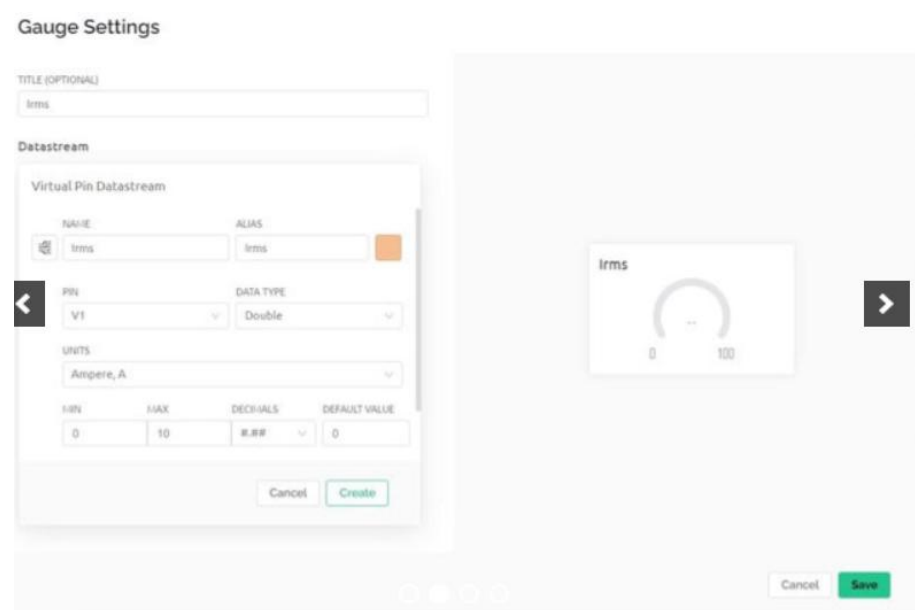


Рис.3.7 Налаштування віджету

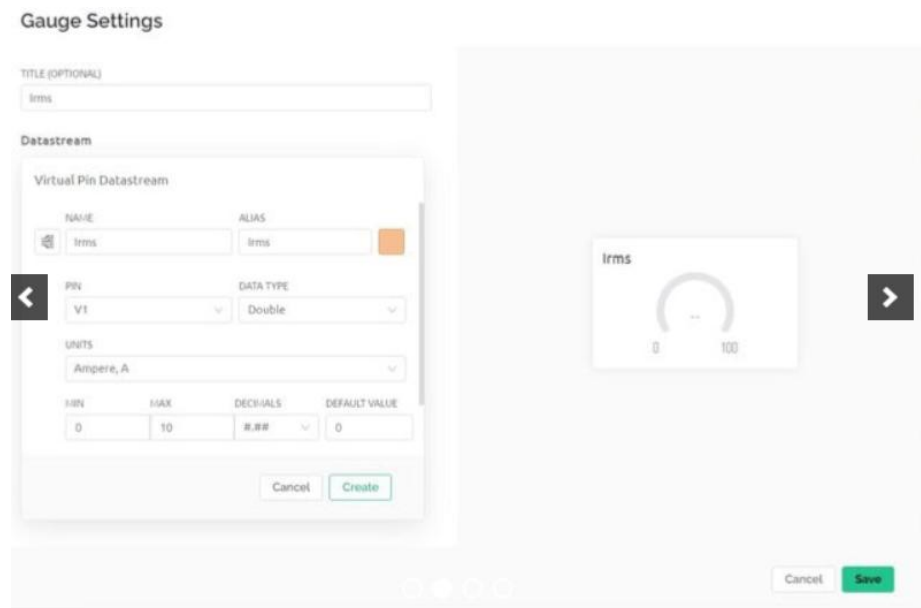


Рис.3.8 Налаштування віджету

У підсумку веб-панель керування набуває вигляду, наведеного на рисунку 3.9, і готова до приймання та відображення даних інтелектуального лічильника електроенергії, що передаються з мікроконтролера ESP32.

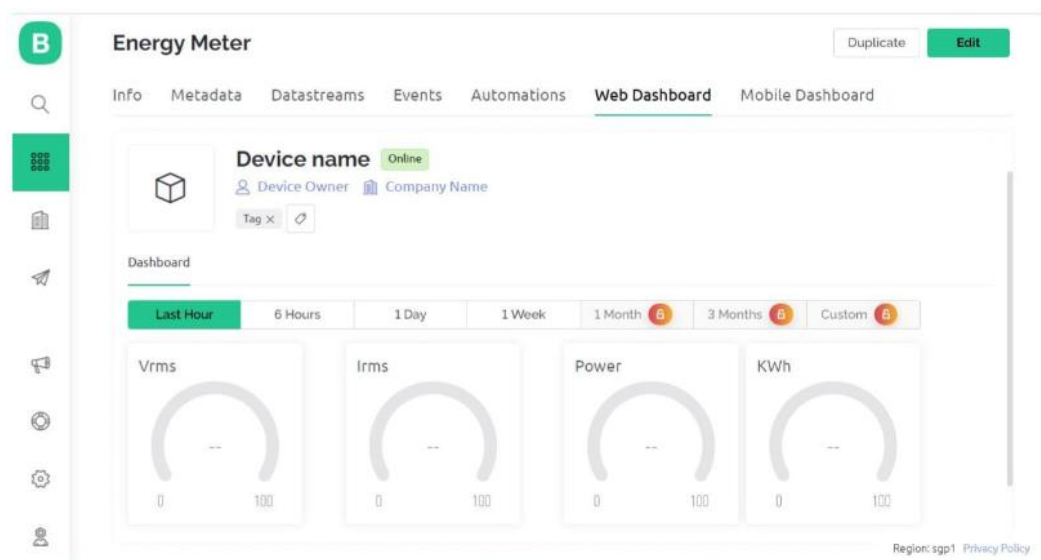


Рис.3.9 Вигляд створеної веб-панелі

Окрім веб-панелі керування, можливо налаштувати панель керування мобільним додатком.

Для цього необхідно завантажити та встановити додаток Blynk з [Google Play Store](#) . Користувачі iOS можуть завантажити його з [App Store](#) .

Після завершення встановлення відкриваємо програму та реєструємось, використовуючи свою електронну адресу та пароль. Потім виконуємо налаштування програми подібно до веб-панелі керування.

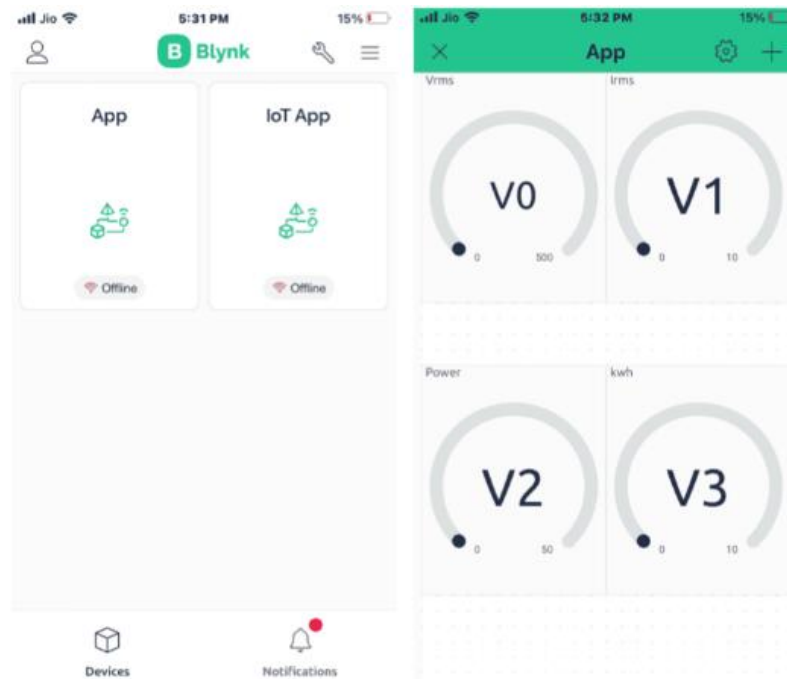


Рис.3.10 Зображення панелі керування мобільним додатком

3.3 Підготовка програмного середовища та калібрування датчика напруги

Встановлення бібліотек.

1. Бібліотека EmonLib.

Бібліотека EmonLib застосовується для реалізації функцій вимірювання електричних параметрів у системах енергомоніторингу. Вона забезпечує безперервний контроль напруги та струму, виконуючи циклічні вимірювання з інтервалом приблизно 5–10 секунд. У фоновому режимі бібліотека здійснює обробку отриманих сигналів, визначає їх середньоквадратичні значення та передає результати до основної програми для подальшого аналізу й обробки.

2. Бібліотека Blynk.

Бібліотека Blynk 2.0 використовується для інтеграції пристроїв із хмарною платформою та організації дистанційного керування. Вона дозволяє підключати різноманітні апаратні платформи, зокрема Arduino, ESP8266 та ESP32, до хмарного середовища Blynk. Це забезпечує можливість створення мобільних застосунків для моніторингу та керування пристроями, а також масштабування IoT-рішень у межах розгорнутих систем.

Калібрування датчика напруги ZMPT101B. Перед початком експлуатації датчик напруги ZMPT101B потребує калібрування, оскільки він не має попередньо встановлених параметрів точності. Для виконання цієї процедури доцільно використовувати плату Arduino Uno або Nano.

Завдяки достатньо лінійним характеристикам аналого-цифрового перетворювача такі плати забезпечують коректне зчитування сигналу, що робить їх придатними для калібрування. Для підключення датчика використовується аналоговий вхід A0, через який здійснюється отримання вимірюваного сигналу та подальше налаштування параметрів датчика.

Тепер завантажуюмо наступний код на плату Arduino.

```

1 void setup()
2 {
3   Serial.begin(9600);
4 }
5
6 void loop()
7 {
8   Serial.println(analogRead(A0));
9   delay(100);
10 }
```

Рис.3.11 Код для калібрування датчика

Після завантаження програмного коду необхідно відкрити послідовний плотер у середовищі Arduino IDE для візуалізації сигналу. Якщо на графіку не відображається коректна синусоїдальна форма, слід відрегулювати підлаштувальний потенціометр датчика.

Калібрування вважається завершеним у той момент, коли на екрані відображається стабільна синусоїда правильної форми, що свідчить про коректне налаштування вимірювального модуля.

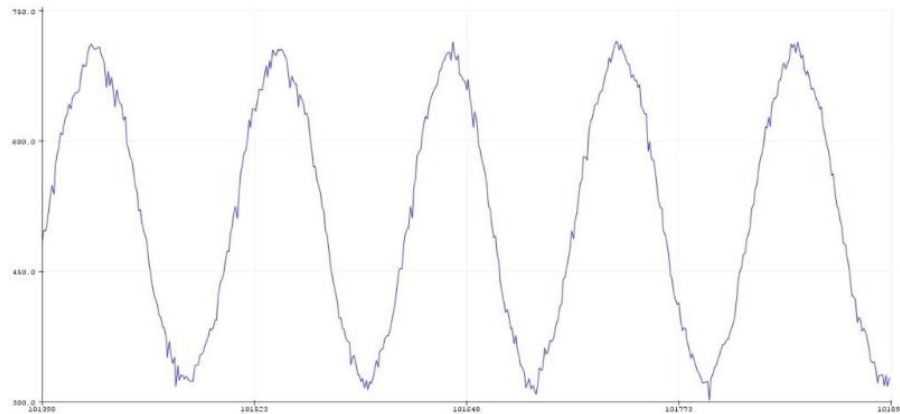


Рис.3.12 Зображення відкаліброваного датчику

Програмна реалізація інтелектуального лічильника енергії IoT. Наведений програмний код являє собою скетч для платформи ESP32, який реалізує функціональність інтелектуального лічильника електроенергії на основі технологій Інтернету речей. У програмі використовуються бібліотеки LiquidCrystal_I2C, EmonLib, EEPROM, WiFi, WiFiClient та BlynkSimpleEsp32, що забезпечують роботу з дисплеєм, обробку вимірювань, збереження даних, мережеву взаємодію та інтеграцію з платформою Blynk 2.0.

Для початку роботи необхідно скопіювати наведений код і вставити його в середовище Arduino IDE.

Перед завантаженням програми слід відредагувати відповідні параметри конфігурації, зокрема вказати власні дані для підключення до мережі Wi-Fi (SSID і пароль), а також унікальний токен автентифікації Blynk. Замінені значення забезпечать коректне підключення пристрою до мережі та взаємодію з хмарною платформою.

```

1 // Blynk and WiFi credentials
2 const char auth[] = "*****";
3 const char ssid[] = "*****";
4 const char pass[] = "*****";

```

Рис.3.13 Вихідний код/програма для інтелектуального лічильника енергії IoT

Після внесення необхідних змін у програмний код можна переходити до його завантаження на плату ESP32. Для цього в середовищі розробки потрібно обрати відповідну модель плати, а також вказати правильний СОМ-порт.

Після виконання цих налаштувань слід ініціювати процес завантаження, натиснувши кнопку «Upload», у результаті чого програмний код буде передано на мікроконтролер, рис. 3.14.

```

1 #define BLYNK_TEMPLATE_ID "*****"
2 #define BLYNK_TEMPLATE_NAME "IoT Energy Meter"
3 #define BLYNK_PRINT Serial
4
5 #include "EmonLib.h"
6 #include <EEPROM.h>
7 #include <WiFi.h>
8 #include <WiFiClient.h>
9 #include <BlynkSimpleEsp32.h>
10 #include <Wire.h>
11 #include <LiquidCrystal_I2C.h>
12 LiquidCrystal_I2C lcd(0x27, 20, 4);
13
14 // Constants for calibration
15 const float vCalibration = 41.5;
16 const float currCalibration = 0.15;
17
18 // Blynk and WiFi credentials
19 const char auth[] = "*****";
20 const char ssid[] = "*****";
21 const char pass[] = "*****";
22
23 // EnergyMonitor instance
24 EnergyMonitor emon;
25
26 // Timer for regular updates
27 BlynkTimer timer;
28
29 // Variables for energy calculation
30 float kWh = 0.0;
31 unsigned long lastMillis = millis();
32
33 // EEPROM addresses for each variable
34 const int addrVrms = 0;
35 const int addrIrms = 4;
36 const int addrPower = 8;
37 const int addrKWh = 12;
38
39 // Function prototypes
40 void sendEnergyDataToBlynk();
41 void readEnergyDataFromEEPROM();
42 void saveEnergyDataToEEPROM();
43
44
45 void setup()
46 {
47   Serial.begin(115200);
48   Blynk.begin(auth, ssid, pass);
49
50   // Initialize the LCD
51   lcd.init();
52   lcd.backlight();
53
54   // Initialize EEPROM with the size of the data to be stored
55   EEPROM.begin(32); // Allocate 32 bytes for float values (4 bytes each) and some extra space
56
57   // Read the stored energy data from EEPROM
58   readEnergyDataFromEEPROM();
59
60   // Setup voltage and current inputs
61   emon.voltage(35, vCalibration, 1.7); // Voltage: input pin, calibration, phase_shift
62   emon.current(34, currCalibration); // Current: input pin, calibration
63
64   // Setup a timer for sending data every 5 seconds
65   timer.setInterval(5000L, sendEnergyDataToBlynk);
66
67   // A small delay for system to stabilize
68   delay(1000);
69 }
70
71 void loop()
72 {
73   Blynk.run();
74   timer.run();
75 }
76
77
78 void sendEnergyDataToBlynk()
79 {
80   emon.calcVI(20, 2000); // Calculate all. No. of half wavelengths (crossings), time-out
81
82   // Calculate energy consumed in kWh
83   unsigned long currentMillis = millis();
84   kWh += emon.apparentPower * (currentMillis - lastMillis) / 3600000000.0;
85   lastMillis = currentMillis;
86
87   // Print data to Serial for debugging
88   Serial.printf("Vrms: %.2fV\tIrms: %.4fA\tPower: %.4fW\tkWh: %.5fKWh\n",
89               emon.Vrms, emon.Irms, emon.apparentPower, kWh);
90
91   // Save the latest values to EEPROM
92   saveEnergyDataToEEPROM();
93
94   // Send data to Blynk
95   Blynk.virtualWrite(V0, emon.Vrms);
96   Blynk.virtualWrite(V1, emon.Irms);
97   Blynk.virtualWrite(V2, emon.apparentPower);
98   Blynk.virtualWrite(V3, kWh);
99
100   // Update the LCD with the new values
101   lcd.clear();
102   lcd.setCursor(0, 0);
103   lcd.print("Vrms: ");
104   lcd.print(emon.Vrms, 2);
105   lcd.print(" V");
106
107   lcd.setCursor(0, 1);
108   lcd.print("Irms: ");
109   lcd.print(emon.Irms, 4);
110   lcd.print(" A");
111
112   lcd.setCursor(0, 2);
113   lcd.print("Power: ");
114   lcd.print(emon.apparentPower, 4);
115   lcd.print(" W");
116
117   lcd.setCursor(0, 3);
118   lcd.print("kWh: ");
119   lcd.print(kWh, 5);
120   lcd.print(" kWh");
121 }
122
123
124 void readEnergyDataFromEEPROM()
125 {
126   // Read the stored kWh value from EEPROM
127   EEPROM.get(addrKWh, kWh);
128
129   // Check if the read value is a valid float. If not, initialize it to zero
130   {
131     if (!isnan(kWh))
132     {
133       kWh = 0.0;
134       saveEnergyDataToEEPROM(); // Save initialized value to EEPROM
135     }
136   }
137 }
138
139 void saveEnergyDataToEEPROM()
140 {
141   // Write the current kWh value to EEPROM
142   EEPROM.put(addrKWh, kWh);
143
144   // Commit changes to EEPROM
145   EEPROM.commit();
146 }

```

Рис.3.14 Код для плати ESP32

Псевдокод для розрахунку споживання та сповіщення про перевантаження START.

```

Ініціалізація LCD
Ініціалізація Wi-Fi
Ініціалізація ADC
Ініціалізація датчиків
WHILE (true)
    voltage_raw = ADC_Read(ZMPT101B)
    current_raw = ADC_Read(SCT013)
    voltage = Calculate_RMS(voltage_raw)
    current = Calculate_RMS(current_raw)
    power = voltage * current
    energy = energy + (power * delta_time)
    LCD_Print(voltage, current, power, energy)
    Send_To_Blynk(voltage, current, power, energy)
    IF power > POWER_LIMIT THEN
        Send_Notification()
        Activate_Buzzer()
    END IF
    Delay(1000)
END WHILE
END

```

3.4 Оцінка працездатності системи збору та візуалізації даних електроспоживання

Після запуску плата ESP32 ініціює спробу підключення до бездротової мережі Wi-Fi, використовуючи попередньо задані параметри SSID та пароль. На

початковому етапі роботи на РК-дисплеї відображається відповідне службове повідомлення, яке інформує про процес встановлення з'єднання.

У разі відсутності підключеного навантаження або коли воно вимкнене, значення струму та напруги повинні наближатися до нуля. Якщо ж спостерігаються відмінні від нуля показники, необхідно виконати коригування калібрувального коефіцієнта у програмному коді для забезпечення точності вимірювань.

```
1 #define vCalibration 41.5
2 #define currCalibration 0.15
```

Рис.3.15 Калібрувальний коефіцієнт у коді

Після підключення навантаження на РК-дисплеї починають відображатися виміряні значення напруги та струму, а також розрахована спожита потужність і сумарне енергоспоживання у кВт·год.

Показники лічильника електроенергії передаються до платформи Blynk з інтервалом кожні 5 секунд. Отримані дані також доступні для перегляду через веб-панель керування, що дозволяє здійснювати моніторинг параметрів у режимі реального часу.

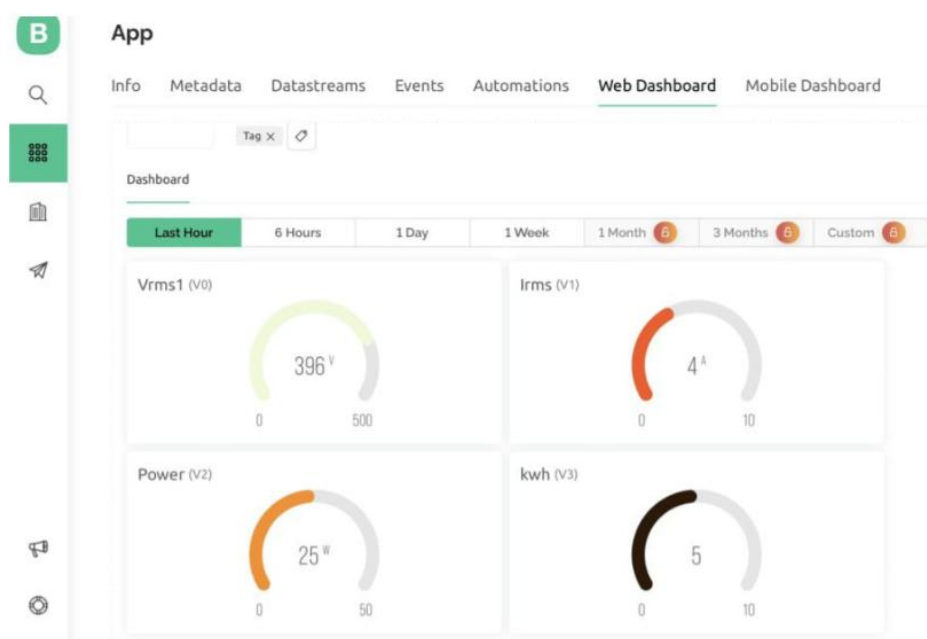


Рис.3.16 Веб-панелі керування та моніторингу Blynk

Так само можливо також одночасно спостерігати за даними на панелі керування мобільним веб-сайтом.

Таким чином можливо використовувати цей IoT-лічильник енергії ESP32 з Blynk 2.0 для контролю споживання електроенергії будинком.

Існує вдосконалена реалізація проєкту лічильника електроенергії, яка використовує HDMI-дисплей для розширеної візуалізації даних. Зокрема, у проєкті IoT AC Energy Meter з ESP32 та HDMI/HMI дисплеєм додатково передбачено відображення параметрів споживання електроенергії на великому екранному інтерфейсі, що дозволяє зручніше контролювати показники в режимі реального часу, рис.3.17.

Окрім локальної візуалізації, система також підтримує віддалений моніторинг даних через хмарні сервіси, зокрема Blynk, де інформація оновлюється автоматично та доступна через веб-панель керування.

Таким чином, використання HDMI-дисплея у поєднанні з IoT-платформою дозволяє розширити функціональність системи та підвищити зручність контролю енергоспоживання.

ВИСНОВКИ

У першому розділі проведено комплексний огляд інфраструктур Інтернету речей, що застосовуються в енергетичних системах будівель. Проаналізовано сучасні наукові підходи до побудови IoT-систем для «розумних будівель», включаючи архітектуру, склад апаратних і програмних компонентів, методи збору, обробки та передачі даних, а також принципи інтеграції сенсорних мереж із хмарними платформами. Розглянуто основні типи датчиків, підходи до їх енергозабезпечення, зокрема використання батарейних рішень, енергозбирання та PoE, а також їх переваги й обмеження.

Виконано огляд методів прогнозного та адаптивного керування енергоспоживанням, які базуються на аналізі історичних і реальних даних для оптимізації роботи інженерних систем будівель. Також проаналізовано мережеві архітектури та протоколи зв'язку IoT (Wi-Fi, ZigBee, LoRaWAN тощо), що забезпечують взаємодію пристроїв у різних сценаріях застосування. Узагальнено практичні приклади впровадження IoT-систем у житлових, освітніх та комерційних будівлях, що підтверджують їхню ефективність у зниженні енергоспоживання та підвищенні керованості енергетичних процесів.

У другому розділі проведено аналіз технічної специфікації та обґрунтовано вибір апаратних і програмних складових для реалізації IoT-системи інтелектуального обліку електроенергії. Розглянуто основні елементи системи, зокрема мікроконтролер ESP32, вимірювальні модулі струму та напруги, модуль живлення, релейний модуль, дисплей та інші компоненти, що забезпечують функціонування системи.

Проаналізовано інтегроване середовище розробки Arduino IDE, яке використовується для програмування мікроконтролерів, та визначено його основні функціональні можливості. Окремо розглянуто сучасні програмні платформи для

створення мобільних застосунків у IoT-системах, включаючи як візуальні no-code/low-code рішення, так і професійні середовища розробки.

У результаті встановлено, що використання поєднання апаратних компонентів на базі ESP32 та сучасних програмних інструментів дозволяє створити ефективну, масштабовану та зручну в керуванні IoT-систему для моніторингу та аналізу енергоспоживання в режимі реального часу.

У третьому розділі виконано практичну реалізацію інформаційної системи моніторингу електроспоживання на базі технологій Інтернету речей. Розроблено апаратну частину системи з використанням мікроконтролера ESP32, датчиків струму та напруги, а також додаткових компонентів для вимірювання та обробки електричних параметрів.

Проведено проєктування електричної схеми, монтаж апаратного забезпечення та налаштування взаємодії між усіма компонентами системи. Реалізовано програмну частину проєкту, включаючи калібрування датчиків, підключення бібліотек та розробку алгоритму збору і передачі даних. Виконано інтеграцію системи з платформою Blynk 2.0, створено веб- та мобільний інтерфейси для візуалізації показників у режимі реального часу. Проведене тестування підтвердило коректність роботи системи, стабільність передачі даних та точність вимірювання електричних параметрів.

Таким чином, розроблена IoT-система забезпечує автоматизований моніторинг електроспоживання, дистанційний доступ до даних та можливість їх візуального аналізу, що підтверджує її практичну ефективність та придатність для використання в реальних умовах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Lee, H.; Kim, M. The Internet of Things in a Smart Connected World; IntechOpen: London, UK, 2020; pp. 91–104.
2. Khajenasiri, I.; Estebasari, A.; Verhelst, M.; Gielen, G. A Review on Internet of Things solutions for intelligent energy control in buildings for smart city applications. *Energy Procedia* 2019, 11, 770–779.
3. Mahbub, M.; Hossain, M.; Gazi, M. IoT-cognizant cloud-assisted energy efficient embedded system for indoor intelligent lighting, air quality monitoring, and ventilation. *Internet Things* 2020, 11, 1–26.
4. Ray, P.P. A survey on Internet of Things architectures. *J. King Saud Univ. Comput. Inf. Sci.* 2018, 30, 291–319.
5. Pan, J.; Jain, R.; Paul, S.; Vu, T.; Saifullah, A.; Sha, M. An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments. *IEEE Internet Things J.* 2019, 2, 527–537.
6. Ray, P.P. A survey of IoT cloud platforms. *Futur. Comput. Inform. J.* 2021, 1, 35–46.
7. Casini, M. Internet of Things for energy efficiency of buildings. *J. Archit. Eng.* 2022, 9, 1–5.
8. Lork, C.; Choudhary, V.; Hassan, N.U.; Tushar, W.; Yuen, C.; Ng, B.K.K.; Wang, X.; Liu, X. An Ontology-Based Framework for Building Energy Management with IoT. *Electronics* 2019, 8, 485.
9. King, J.; Perry, C. Smart buildings: Using smart technology to save energy in existing buildings. *American Counc. Energy-Efficient Econ.* 2017, Report A170, 1–46.
10. Png, E.; Srinivasan, S.; Bekiroglu, K.; Chaoyang, J.; Su, R.; Poolla, K. An internet of things upgrade for smart and scalable heating, ventilation and air-conditioning control in commercial buildings. *Appl. Energy* 2019, 239, 408–424.

11. Prauzek, M.; Konecny, J.; Borova, M.; Janosova, K.; Hlavica, J.; Musilek, P. Energy harvesting sources, storage devices and system topologies for environmental wireless sensor networks: A review. *Sensors* 2018.
12. Gawali, S.K.; Deshmukh, M.K. Energy Autonomy in IoT Technologies. *Energy Procedia* 2019, 156, 222–226.
13. Curry, J.; Harris, N. Powering the Environmental Internet of Things. *Sensors* 2019, 19, 1940.
14. Matsui, K.; Yamagata, Y.; Kawakubo, S. Real-time sensing in residential area using IoT technology for finding usage patterns to suggest action plan to conserve energy. *Energy Procedia* 2019, 158, 6438–6445.
15. Mataloto, B.; Ferreira, J.C.; Cruz, N. LoBEMS—IOT for Building and Energy Management Systems. *Electronics* 2019, 8, 763.
16. Zou, H.; Zhou, Y.; Yang, J.; Spanos, C. Towards occupant activity driven smart buildings via WiFi-enabled IoT devices and deep learning. *Energy Build.* 2018, 177, 12–22.
17. Lachhab, F.; Bakhouya, M.; Ouladsine, R.; Essaaidi, M. Towards an Intelligent Approach for Ventilation Systems Control using IoT and Big Data Technologies. *Procedia Comput. Sci.* 2018, 130, 926–931.
18. Marinakis, V.; Doukas, H.; Spiliotis, E.; Papastamatiou, I. Decision support for intelligent energy management in buildings using the thermal comfort model. *Int. J. Comput. Intell. Syst.* 2023.
19. Fayaz, M.; Kim, D. Energy consumption optimization and user comfort management in residential buildings using a bat algorithm and fuzzy logic. *Energies* 2018.
20. Serra, J.; Pubill, D.; Antonopoulos, A.; Verikoukis, C. Smart HVAC Control in IoT: Energy Consumption Minimization with User Comfort Constraints. *Sci. World J.* 2024.

21. Kim, D.S.; Chung, B.J.; Son, S.-Y. Implementation of a Low-Cost Energy and Environment Monitoring System Based on a Hybrid Wireless Sensor Network. *J. Sens.* 2017.
22. Mehta, R.; Sahni, J.; Khanna, K. Internet of Things: Vision, Applications and Challenges. *Procedia Comput. Sci.* 2021.
23. Sodhro, A.; Pirbhulal, S.; Luo, Z.; Albuquerque, V. Towards an optimal resource management for IoT Based green and sustainable smart cities. *Clean. Prod.* 2019.
24. Mudaliar, M.D.; Sivakumar, N. IoT based real time energy monitoring system using Raspberry Pi. *Internet Things* 2020.
25. Mihailescu, R.-C.; Hurtig, D.; Olsson, C. End-to-end anytime solution for appliance recognition based on high-resolution current sensing with few-shot learning. *Internet Things* 2020.
26. Santos, D.; Ferreira, J.C. IoT Power Monitoring System for Smart Environments. *Sustain. J. Rec.* 2019
27. Hafeez, G.; Wadud, Z.; Khan, I.; Khan, I.; Shafiq, Z.; Usman, M.; Khan, M. Efficient energy management of IoT-enabled smart homes under price-based demand response program in smart grid. *Sensors* 2020.
28. Massano, M.; Patti, E.; Macii, E.; Acquaviva, A.; Bottaccioli, L. An Online Grey-Box Model Based on Unscented Kalman Filter to Predict Temperature Profiles in Smart Buildings. *Energies* 2020.
29. Dell’Isola, M.; Ficco, G.; Canale, L.; Palella, B.; Puglisi, G. An IoT integrated tool to enhance user awareness on energy consumption in residential buildings. *Atmosphere* 2019.
30. Han, J.; Lee, E.; Cho, H.; Yoon, Y.; Lee, H.; Rhee, W. Improving the Energy Saving Process with High-Resolution Data: A Case Study in a University Building. *Sensors* 2018
31. Monk, S. *Programming Arduino: Getting Started with Sketches*. – 2nd ed. – New York: McGraw-Hill Education, 2016. – 192 p.

32. MIT App Inventor. Get Started [Электронный ресурс]. – Режим доступа: <https://appinventor.mit.edu/explore/get-started.html> – Дата звернення: 02.05.2026.

33. RAD Studio [Электронный ресурс]. – Embarcadero Technologies. – Режим доступа: <https://www.embarcadero.com/products/rad-studio> – Дата звернення: 02.05.2026.

34. Cayenne IoT Platform [Электронный ресурс]. – myDevices. – Режим доступа: <https://cayenne.mydevices.com/> – Дата звернення: 02.05.2026.

35. Blynk IoT Platform [Электронный ресурс]. – Blynk Inc. – Режим доступа: <https://blynk.io/> – Дата звернення: 02.05.2026.

36. ThingsBoard – Open-source IoT Platform [Электронный ресурс]. – ThingsBoard, Inc. – Режим доступа: <https://thingsboard.io/> – Дата звернення: 02.05.2026.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)



МЕТА ТА ЗАВДАННЯ

2

Мета роботи - реалізація інформаційної IoT-системи моніторингу електроспоживання та візуалізації даних у режимі реального часу.

Об'єкт дослідження – процес моніторингу та контролю електроспоживання в електричних мережах із використанням технологій IoT.

Предмет дослідження – методи та засоби побудови IoT-системи для вимірювання електричних параметрів.

Завдання роботи:

1. Опгляд інфраструктур інтернету речей для енергетичних систем будівель.
2. Аналіз вимог і програмно-інструментальне забезпечення IoT-системи.
3. Реалізація та тестування інформаційної системи моніторингу.

РОЗВИТОК ІНТЕРНЕТУ РЕЧЕЙ І ЙОГО ЗАСТОСУВАННЯ В ЕНЕРГЕТИЧНИХ СИСТЕМАХ БУДІВЕЛЬ

3

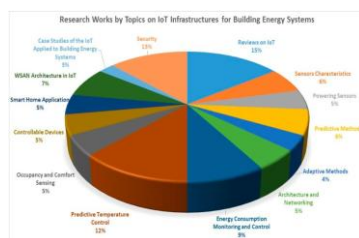


Рис.1 Дослідницькі роботи за темами інфраструктури Інтернету речей для енергетичних систем будівництва

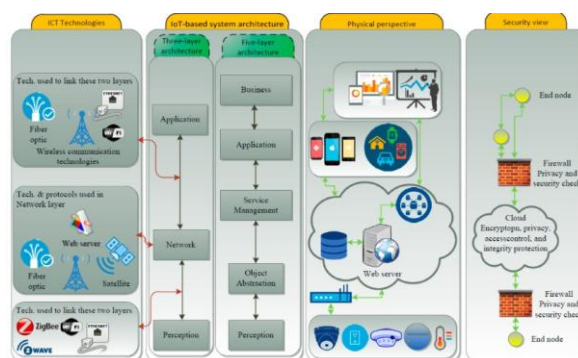


Рис.2 Архітектура системи на основі Інтернету речей

КЛАСИФІКАЦІЯ ДАТЧИКІВ ТА МЕТОДИ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ

4

Табл.1. Очікуване зниження енергоспоживання для різних датчиків

Система	Технології	Зниження енергоспоживання
Аналітика	Хмарна енергетична інформаційна система	5–10% всієї будівлі
Автоматизація будівель	Система автоматизації будівель	10–25% всієї будівлі
Освітлення	Розширене керування освітленням	45%
Освітлення	Веб-система керування освітленням	Зниження на 20–30% порівняно з контрольним рівнем
Затінення вікон	Автоматизована система затінення	21–38%
Затінення вікон	Розумне скло	20–30%
Затінення вікон	Перемикається плівка	32–43%

Табл.2. Виробництво електроенергії з різних відновлюваних джерел енергії

Система збору	Вихідна потужність	Система збору	Вихідна потужність
Фотоелектричні (зовнішні)	50 мВт/см ²	Рух повітря	6 Вт/см ²
Фотоелектричні (для приміщень)	50 мкВт/см ²	Перетворення зміни тиску на електрику	15 мкВт/см ²
Фотосинтез (лабораторія)	10–40 мкВт/см ²	Пієзоелектрики	12,5 мкВт/см ²
Термоелектрика	20 мкВт/см ²	Трибоелектрика	3 мВт/см ²
Піроелектрики	8,64 мкВт/см ²	Електростатика	12 мВт/см ²
Хімічний потенціал	3 мВт	Індукція	70+ мкВт/см ²

МЕРЕЖЕВІ ТЕХНОЛОГІЇ ТА ЕНЕРГОМЕНЕДЖМЕНТ В ІОТ-СИСТЕМАХ

5

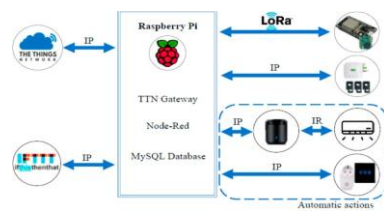


Рис.3. Архітектура системи на основі The Things Network

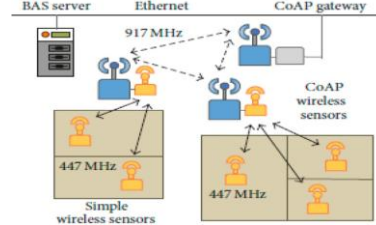


Рис.4. Мережа з використанням датчиків протоколу обмежених програм

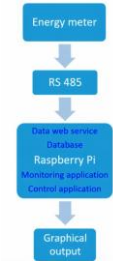


Рис.5. Принципова блок-схема системи моніторингу енергії Інтернету речей

Табл.3. Огляд поширених протоколів Інтернету речей

Бездротові технології	Швидкість передачі даних	Максимальна тривалість зарядки	Дальність зв'язку	Безпека	Переваги
Bluetooth S	125 кбіт/с (далека дальність S = 7) 2 Мбіт/с–500 кбіт/с (далека дальність S = 2)	255 байтів	До 200 м +200 м (BLE)	L1–Без безпеки, L2–AES 128 L3–AES та сполучення L4–ECDHE	Просте обладнання; легкий доступ та експлуатація; безпечний, низьке енергоспоживання (BLE)
LoRaWAN	50 кбіт/с	243 байти	~5 км місто ~15 км–20 км село ~1 км у місті ~10 км у сільській місцевості	т АЕС	Доступність; велика дальність зв'язку; безпека; низьке енергоспоживання; низька вартість
NB-IoT	200 кбіт/с	1600 байт	~10 км у місті ~40 км у сільській місцевості	Шифрування LTE	широка максимальна довжина завантаження (покривта 4G); безпечний
Сігмакс	100 кбіт/с	12 байтів	~10 км у місті ~40 км у сільській місцевості	Без шифрування або адаптації для кожного випуску	Велика дальність зв'язку; низьке енергоспоживання
Wi-Fi	1 Гбіт/с IEEE 802.11ac	2034 байти	1–100 м	WPA/WPA2	Розширений стандарт; висока швидкість
ZigBee	20 кбіт/с при 868 МГц 40 кбіт/с при 915 МГц 250 кбіт/с при 2,4 ГГц	255 байтів	10–300 м у зоні прямої видимості, 75–100 м у приміщенні	128-бітний AES	Велика кількість вузлів; низьке енергоспоживання; низька вартість; безпека
Z-хвиля	100 кбіт/с	64 байти	~100 м (може змінюватися залежно від кількості вузлів) (до 4 переходів)	Безпека 2 (S2) (включаючи AES-128, ECDHE, безпечний тунель TLS)	Просте встановлення; сумісність між пристроями різних виробників; низьке енергоспоживання; безпека

АНАЛІЗ ТЕХНІЧНОЇ СПЕЦИФІКАЦІЇ ТА ВИБІР СКЛАДОВИХ ПРОЕКТУ

6

Табл.4. Складові реалізації проекту

№	Компоненти	Кількість
1	ESP32	1
2	Модуль датчика змінної напруги ZMPT101B	1
3	SCT-0130030 неінвазивний датчик змінного струму	1
4	20x4 I2C ПК - датчик	1
5	Кабель Micro-USB	1
6	Резистор 10 кОм	2
7	Резистор 100 кОм	1
8	Конденсатор 10 мкФ	1
9	3' сднувальні дроти	10
10	Макетна плата	1



Рис.6. Датчик струму SCT-013



Рис.7. Модуль датчика однофазної напруги змінного струму

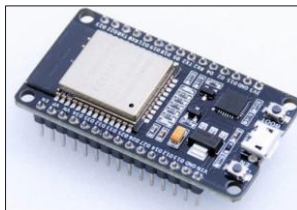


Рис.8. Мікроконтролерна платформа ESP32

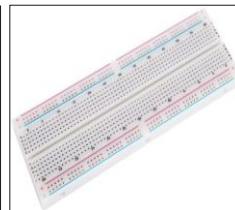


Рис.9. Макетна плата

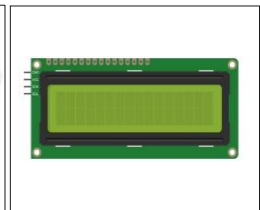


Рис.10. Дисплей

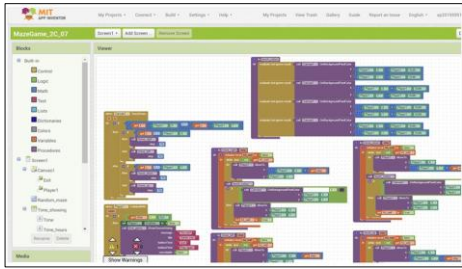


Рис.11 MIT App Inventor



Рис.12 RAD Studio



Рис.13 Cayenne IoT platform від myDevices



Рис.14 Blynk IoT

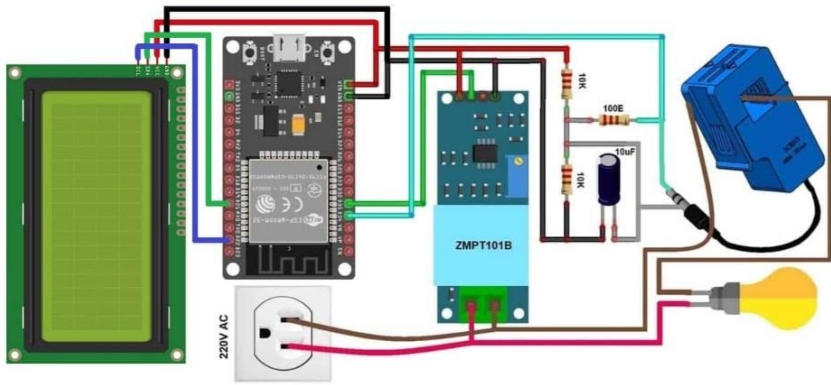


Рис. 15 Схема лічильника електроенергії

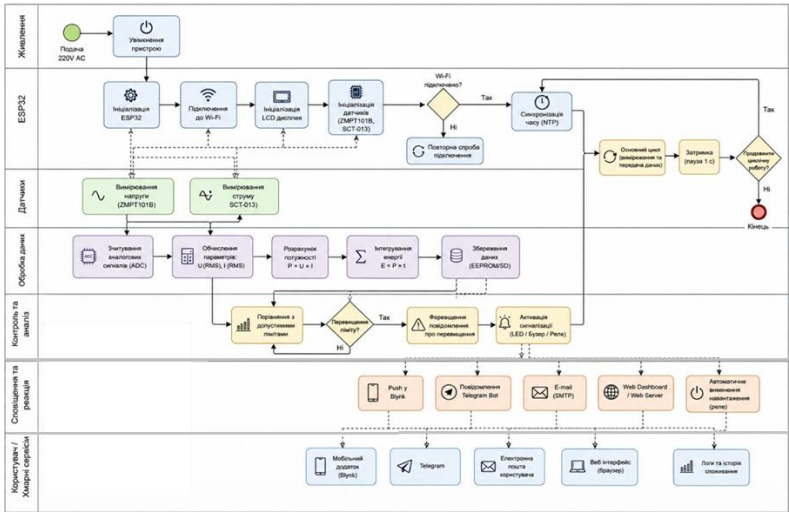




Рис.16 Створення шаблону



Рис.17 Призначення назви, обладнання та тип підключення



Рис.18 Створення віджету Gauge

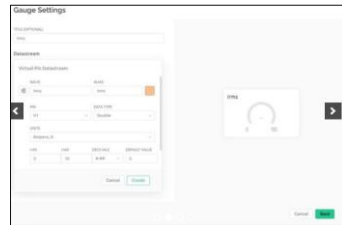


Рис.19 Налаштування віджету

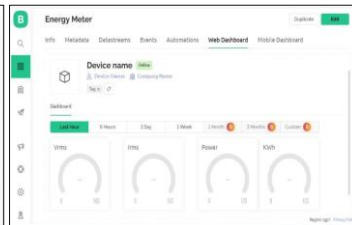


Рис.20 Вигляд створеної веб-панелі

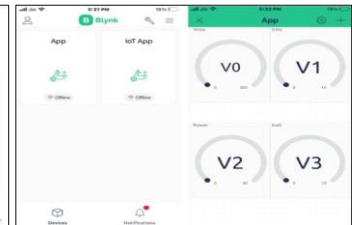


Рис.21 Зображення панелі керування мобільним додатком

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА КАЛІБРУВАННЯ СИСТЕМИ



Рис.22 Код для калібрування датчика

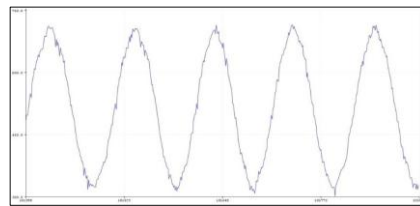


Рис.23 Зображення відкаліброваного датчику

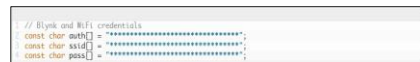


Рис.24 Вихідний код/програма для інтелектуального лічильника енергії IoT



Рис.25 Код для плати ESP32

ВИСНОВКИ

У ході виконання роботи здійснено комплексний аналіз сучасних підходів до побудови IoT-рішень та досліджено особливості їх застосування в енергетичних системах будівель. На основі проведеного аналізу обґрунтовано вибір апаратних і програмних засобів для реалізації інформаційної системи моніторингу та візуалізації витрат електроенергії на житлових об'єктах на базі ESP32.

У межах практичної частини виконано проектування електричної схеми, реалізовано апаратну складову системи із застосуванням датчиків струму та напруги, а також розроблено програмне забезпечення для збору, обробки та передачі даних. Додатково проведено калібрування вимірювальних модулів, що забезпечило підвищення точності отриманих результатів. Інтеграція системи здійснено з хмарною платформою Blynk 2.0, що дозволило реалізувати функції віддаленого моніторингу та візуалізації параметрів електроспоживання в режимі реального часу через веб- та мобільний інтерфейси.

У результаті експериментальної перевірки підтверджено працездатність розробленої системи, стабільність передачі даних та достатню точність вимірювання електричних параметрів. Отримані результати свідчать про ефективність запропонованого підходу та можливість його практичного застосування для контролю, аналізу та оптимізації енергоспоживання в житлових будівлях.

Таким чином, поставлена мета роботи досягнута, а розроблена інформаційна система може бути використана як основа для подальшого розвитку інтелектуальних рішень у сфері енергомоніторингу та підвищення енергоефективності житлових об'єктів.