

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІНЖЕНЕРНО-ТЕХНІЧНИЙ ФАКУЛЬТЕТ  
КАФЕДРА ПРИЛАДОБУДУВАННЯ

ДО ЗАХИСТУ ДОПУЩЕНО

Завідувач кафедри

\_\_\_\_\_

Ігор ЧИЧУРА

«\_\_\_» \_\_\_\_\_ 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної бакалаврської роботи

на тему:

**АВТОМАТИЗОВАНИЙ ПРИСТРІЙ ЗБОРУ ПАРАМЕТРІВ  
ЕЛЕКТРОМЕРЕЖІ З МОЖЛИВІСТЮ ІНТЕГРУВАННЯ У SMART  
GRID**

Виконав:

Василь КОЗАК

(ім'я та прізвище)

\_\_\_\_\_

(підпис)

Керівник:

ст. викл. Роман МЕШКО

(вчене звання, ім'я та прізвище)

\_\_\_\_\_

(підпис)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної бакалаврської роботи: 65 сторінки, 32 рисунків, 1 таблиць, 2 додатки, 18 джерел посилань.

### **ІНТЕРНЕТ РЕЧЕЙ, ЕНЕРГОЗБЕРЕЖЕННЯ, ARDUINO, MQTT**

Об'єкт дослідження – автоматизований пристрій збору параметрів електромережі з можливістю інтегрування у Smart Grid.

Мета роботи – розробити автоматизований пристрій збору параметрів електромережі, яка працює в автономному режимі із можливістю реєстрації показів для формування даних Smart Grid.

Методи дослідження – технічний аналіз пристроїв розумний лічильник; аналіз аналогічних розробок на платформі ардуіно; проектування структурної і принципової схеми; розробка алгоритму роботи системи.

Проведено огляд аналогів на основі ардуіно і готових комплексів. За результатами аналізу спроектовано автоматизований пристрій збору параметрів електромережі і підібрано її компоненти. Розроблено структурну та принципову схеми, а також алгоритм роботи.

## ABSTRACT

Explanatory note of the qualifying bachelor thesis: 65 pages, 32 figures, 1 table, 2 appendices, 18 reference sources.

INTERNET OF THINGS, ENERGY SAVING, ARDUINO, MQTT

The object of research is an automated device for collecting power grid parameters with the possibility of integration into Smart Grid.

The purpose of the work is to develop an automated device for collecting power grid parameters that works in offline mode with the ability to register displays for Smart Grid data generation.

Research methods – technical analysis of smart meter devices; analysis of similar developments on the Arduino platform; design of a structural and principle scheme; development of the system operation algorithm.

An overview of analogues based on Arduino and ready-made complexes was conducted. Based on the results of the analysis, an automated device for collecting power network parameters was designed and its components were selected. The structural and principle schemes, as well as the work algorithm, have been developed.

# Ужгородський національний університет

Інженерно-технічний факультет  
Кафедра приладобудування  
Освітньо-кваліфікаційний рівень "БАКАЛАВР"  
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
ст.. викл. Ігор ЧИЧУРА

"\_\_" \_\_\_\_\_ 2024 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ

студенту Козак Василь Васильович  
(прізвище, ім'я, по батькові)

1. Тема КБР: Автоматизований пристрій збору параметрів електромережі з  
можливістю інтегрування у Smart Grid  
керівник роботи: Мешко Роман Олексійович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені Розпорядженням по ІТФ від "\_\_" \_\_\_\_\_ 2024 року № \_\_\_\_.

2. Строк подання студентом роботи "\_\_" \_\_\_\_\_ 2024 року.

3. Вихідні дані до роботи: об'єктом розробки є автоматизований пристрій збору параметрів електромережі для моніторингу споживання енергії, і якості наданих послуг, можливістю інтегрування у Smart Grid для формування попиту.

Умови експлуатації пристрою:

- температура навколишнього середовища: від 0°C до +60 °C;
- атмосферний тиск: від 720 до 780 мм.рт.ст.;
- відносна вологість повітря: до 95 %;
- габарити і маса: мінімально можливі.

Характеристики системи:

- збір параметрів: струм, напруга;
- моніторинг 1 фази;
- комунікація за допомогою Інтернету;
- робота в автоматизованому режимі.

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

Огляд та аналіз аналогів об'єкту проектування (аналіз сучасних автоматизованих пристроїв розумний лічильник; огляд аналогічних рішень на основі мікропроцесорної схемотехніки; принципи функціонування автоматизованих пристроїв). Проектно - конструкторський розділ (розробка структурної, принципової схем; розрахунок параметрів модулів та елементів схеми; підбір датчиків).

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

- структурна або функціональна схема пристрою (1 аркуш А3);
- принципова електрична схема (1 аркуш А3).
- алгоритм функціонування контролера(1 аркуш А3);

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання: “ \_\_\_\_\_ ” “ \_\_\_\_\_ ” 2024 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітки
1	Аналіз сучасних систем розумний лічильник.	30.03.2024	
2	Огляд аналогів.	20.04.2024	
3	Розробка структурної і принципової схеми.	30.04.2024	
4	Підбір елементів	10.05.2024	
5	Написання програми для контролера.	20.05.2024	
6	Розробка схем.	25.05.2024	
7	Написання пояснювальної записки.	05.06.2024	
8	Оформлення роботи та креслень.	10.06.2024	

Студент

\_\_\_\_\_ / \_\_\_\_\_  
(підпис)

\_\_\_\_\_ / \_\_\_\_\_  
(Ім'я ПРІЗВИЩЕ)

Керівник КБР

\_\_\_\_\_ / \_\_\_\_\_  
(підпис)

\_\_\_\_\_ / \_\_\_\_\_  
(Ім'я ПРІЗВИЩЕ)

## Зміст

ВСТУП.....	9
1 ОГЛЯД І АНАЛІЗ АНАЛОГІВ ОБ’ЄКТУ ПРОЕКТУВАННЯ.....	10
1.1 Моніторинг Smart Grid на основі ІОТ за допомогою контролера Arduino .....	10
1.2 Розумний вузол навантаження для нерозумного навантаження відповідно до парадигми Smart Grid: Нова система управління енергією будинку .....	13
1.3 Моніторинг і керування розумною електромережею на основі ІоТ за допомогою Arduino .....	18
1.4 Gravity: цифровий ватметр I2C .....	20
1.5 Системи моніторингу якості електричної енергії SMART GRID: метрологічний компас PQ5000CL .....	22
2 ПРОЕКТНО - КОНСТРУКТОРСЬКИЙ РОЗДІЛ.....	20
2.1 Теоретичні основи Інтернет речей .....	25
2.1.1 Зв'язок між машинами (M2M) .....	25
2.1.2 Приклади протоколів і архітектур .....	26
2.1.3 MQTT .....	27
2.2 Теоретичні основи вимірювання .....	35
2.2.1 Трансформатори струму .....	35
2.2.2 Активне значення (RMS значення) форми сигналу змінного струму.....	37

КБР.АКІТ.20050051.01.000 ПЗ								
Зм.	Арк.	№ докум.	Підпис	Дата	Автоматизований пристрій збору параметрів електромережі з можливістю інтегрування у Smart Grid Пояснювальна записка	Літера	Аркуш	Аркушів
Розробив		Козак В.В.				У	6	65
Перевірив		Мешко Р.О				ІТФ, кафедра ПБ, 4 курс бакалаври денна форма		
Т. контр.								
Н.Контр.								
Затв.		Чичура І.І.						

2.2.3. Аналого-цифрові перетворювачі - АЦП .....	38
2.2.3. Дільник напруги .....	38
2.2.3. Аналого-цифрові перетворювачі - АЦП .....	28
2.3 Матеріали .....	40
2.3.1 Arduino Uno .....	40
2.3.2 Wemos D1 Mini .....	42
2.3.3 Датчик SCT-013-030.....	42
2.3.4 Модульна плата каналу реле 5 В-1.....	44
2.4 Синтез структурної та принципової схеми .....	47
3 АЛГОРИТМ РОБОТИ ТА ПРОГРАМУВАННЯ КОНТРОЛЕРА .....	49
3.1 Архітектура .....	51
3.2 Організація бази даних .....	52
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	56
ДОДАТКИ.....	58

## ВСТУП

Smart Grid (розумна мережа) — набір технологій, що перетворюють енергетичну інфраструктуру старого типу на сучасну цифрову систему. Тільки на основі розумних мереж можливий розвиток сучасної енергетики.

Енергозбереження та розробка методів її ефективного використання були предметом вивчення значної частини дослідницького співтовариства протягом багатьох років. Одним із найважливіших секторів споживання енергії є будівельний сектор, з особливим акцентом на житлових будинках. Швидкий технологічний розвиток в останні роки в області Інтернету речей (IoT) з розвитком мереж і взаємодією різних пристроїв і систем робить можливим розробку додатків у сфері «Розумних будинків», який може значною мірою допомогти в управлінні енергією в домашніх установках.

Нами розроблена система, яка реалізує запис енергетичних даних, з метою розробки методів для покращення енергетичних звичок споживачів і зменшення непотрібного споживання електроенергії. Реалізація ідеї базувалася на надійній системі дистанційного керування навантаженнями та запису цих вимірювань. Це можливо із використанням розумних лічильників. Вони передають дані про споживання електроенергії практично в режимі реального часу. Також допомагають споживачеві приймати обґрунтовані рішення про те, скільки енергії використовувати і в який час доби.

З кожним роком в наших будинках з'являється все більше електроприладів: малопотужні планшети та побутові прилади, кондиціонери. Люди переходять на опалення на електриці, купують електрокари. Це веде до надмірного зростання споживання електроенергії і як наслідок появи пікових навантажень на мережу навіть в літню пору.

Вироблення додаткових потужностей із точним навантаженням сприятиме заощадливому використанні електроенергії. Саме моніторинг споживання і буде тою цеглинкою у побудові великих мереж типу Smart Grid у всій країні.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

# 1 ОГЛЯД І АНАЛІЗ АНАЛОГІВ ОБ'ЄКТУ ПРОЕКТУВАННЯ

## 1.1 Моніторинг Smart Grid на основі ІОТ за допомогою контролера Arduino

Інтелектуальний лічильник електроенергії, що використовує систему Wi-Fi, розроблений на основі трьох основних цілей. Вони є: 1. Негайно забезпечити автоматичне зчитування енергії навантаження; 2. Оптимально використовувати електроенергію; 3. Зменште втрату електроенергії [1].

Дані системи відображаються на веб-сторінці, до якої може отримати доступ споживач.

Система розроблена на мікроконтролері Arduino. Його можна структурно розділити на три частини, а саме: контролер, схему виявлення крадіжки та модуль Wi-Fi. Контролер виконує основні розрахунки та обробляє інформацію. Схема виявлення крадіжки надає інформацію про будь-яке зчитування енергії додаткового або крадіжного навантаження, і найважливішу роль відіграє модуль Wi-Fi для надсилання інформації з контролера через Інтернет.

У цьому документі описано розробку та впровадження лічильника енергії на основі мікроконтролера Arduino з використанням концепції ІоТ. Запропонована конструкція системи виключає участь людини в обслуговуванні електроенергії. Покупець зобов'язаний сплачувати за споживання електроенергії за графіком, у випадку, якщо він не зміг оплатити, передачу електроенергії можна відключити автономно з віддаленого сервера. Користувач може контролювати споживання енергії в одиницях з веб-сторінки, надавши ІР-адресу пристрою. Блок виявлення крадіжки, підключений до лічильника енергії, повідомить сторону компанії, коли відбудеться втручання в лічильник електроенергії, і він надішле інформацію про виявлення крадіжки через модем PLC, а виявлена крадіжка відобразиться у вікні терміналу на стороні компанії. Пристрій Wi-Fi виконує операцію ІоТ, надсилаючи дані лічильника енергії на веб-сторінку, доступ до якої можна отримати через ІР-адресу.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Інтелектуальний лічильник електроенергії з використанням модуля Wi-Fi можна легко описати в двох частинах. Перша частина, яка є фізичною частиною, а друга – веб-сторінкою, складається з плати Arduino Nano, модуля Wi-Fi ESP 8266 (Node MCU), РК-дисплея 16\*2, зумера та джерела живлення. Блок-схема показана на рисунку 1.

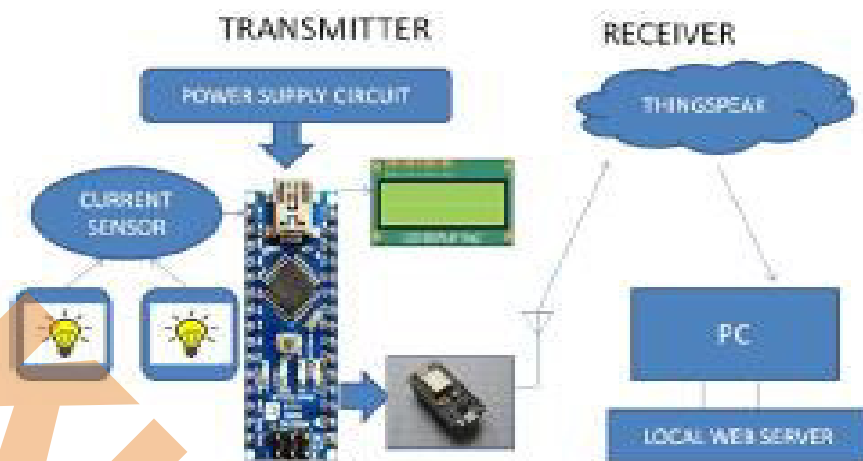


Рисунок 1.1 - Блок-схема системи [1]

Повна система зондування на основі IoT пропонується для застосування автоматизації підстанції в середовищі Smart Grid. Детально обговорюються різні частини системи разом з можливістю їх застосування разом із існуючими системами автоматизації підстанцій. Також обговорюється загальна реалізація системи, включаючи топології мережі, мережі бездротового зв'язку та безпечний зв'язок з іншими частинами енергосистеми. Протокол MQTT, запропонований у запропонованій системі, актуальний для середовища підстанції та забезпечує надійний і безпечний спосіб передачі даних датчиків великій аудиторії. Конструкція системи економічно ефективна та проста у впровадженні та налаштуванні. Він забезпечує новий підхід до моніторингу, зберігання, візуалізації та передачі даних датчиків за допомогою Інтернету речей у середовищі підстанції. Запропонована система в даний час знаходиться на стадії впровадження, і прототип сенсорного вузла Arduino був випробуваний як її частина.

У проєкті використано дві принципові схеми. Схематична діаграма (рис. 1.2. ) та принципова схема блоку живлення (рис. 1.3).

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

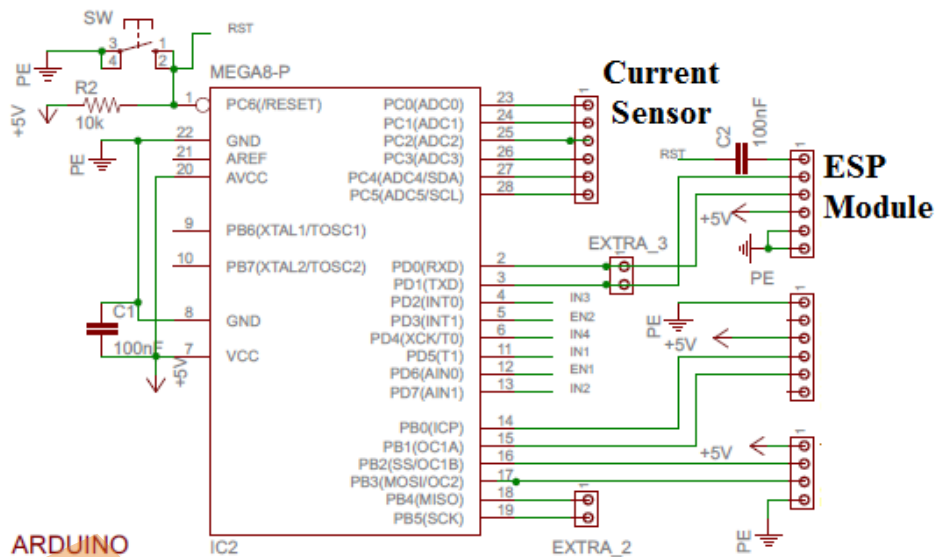


Рисунок 1.2 - Схематична діаграма [1]

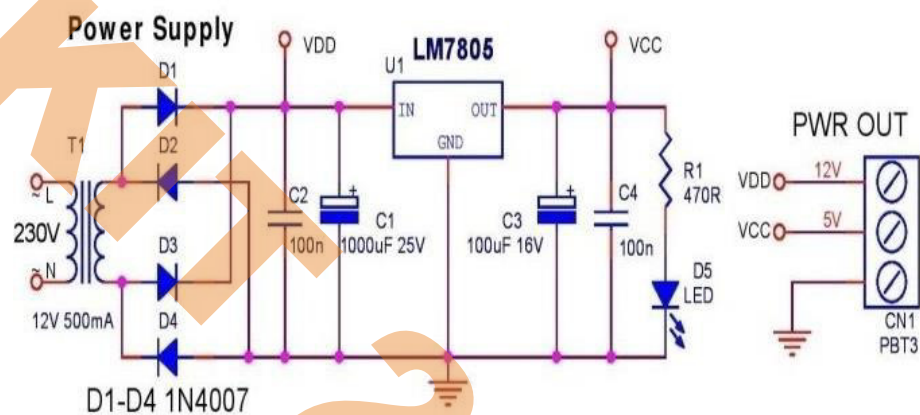


Рисунок 1.3 - Принципова схема блоку живлення [1]

Запропоновану систему можна використовувати для відображення показників споживання енергії навантаженням у Ватах. Кожен користувач матиме доступ до інформації з будь-якої точки землі. Thingspeak.com є однією з таких веб-сторінок, яка використовує аналітику Math Works MATLAB для представлення інформації про пристрій у більш детальному аналізі як в описі, так і у візуалізації. Thingspeak.com надає користувачеві можливість додавати будь-яку кількість каналів до одного облікового запису, і в кожному обліковому записі інформацію можна вводити у 8 полів. Обліковий запис можна призначити одній частині області, а n каналів можна створити для набору n метрів у цьому населеному пункті. Аналітику можуть переглядати як споживач, так і постачальник послуг.

Вим.	Арк.	№ докум.	Підпис	Дата

## 1.2 Розумний вузол навантаження для нерозумного навантаження відповідно до парадигми Smart Grid: Нова система управління енергією будинку

Розумна мережа – це інтелектуальна мережа, яка включає технології інтелектуального вимірювання, сучасні перетворювачі електроенергії, інфраструктуру швидкого зв'язку, автоматизацію та участь споживачів, що забезпечує ефективну та надійну роботу енергосистеми [2].

Постійне збільшення споживання електроенергії в житлових будинках привернуло більше уваги до розробки систем управління енергією вдома (HEMS).

Для встановлення зв'язку між побутовою технікою та LMU було запропоновано два підходи. Перший підхід полягає в тому, щоб зробити прилад інтелектуальним, включивши засоби зв'язку та обробки інформації. Другий запропонований підхід полягає у впровадженні інтелекту в розетки, що вимагає модифікації електричної інфраструктури побудованого будинку.

SLN призначений для підключення між наявною розеткою живлення та навантаженням. Кожен SLN має інтерфейс користувача, який допомагає споживачеві налаштувати роботу навантаження. Після конфігурації споживачем SLN надсилає параметри навантаження до LMU, спеціального одноплатного комп'ютера, який виконує алгоритм керування навантаженням для інтелектуального планування навантажень для участі в програмах DR. Потік інформації та живлення в пропонованій HAN зображено на рисунку 1.4.

Новими аспектами SLN є: - Не вимагає жодних модифікацій в існуючій електричній інфраструктурі будинку. Не потребує модифікації побутової техніки на етапі виробництва та у споживача. Загальне рішення для всіх типів нерозумних навантажень. Здатність включати різноманітні неінтелектуальні плановані завантаження в будь-який тип алгоритму DR. Мобільний характер. У разі зміни будинку можна використовувати той самий HAN, що складається з SLN, без додаткових витрат на компоненти. Діє як вимірювач бездротової потужності, коефіцієнта потужності, напруги та струму.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

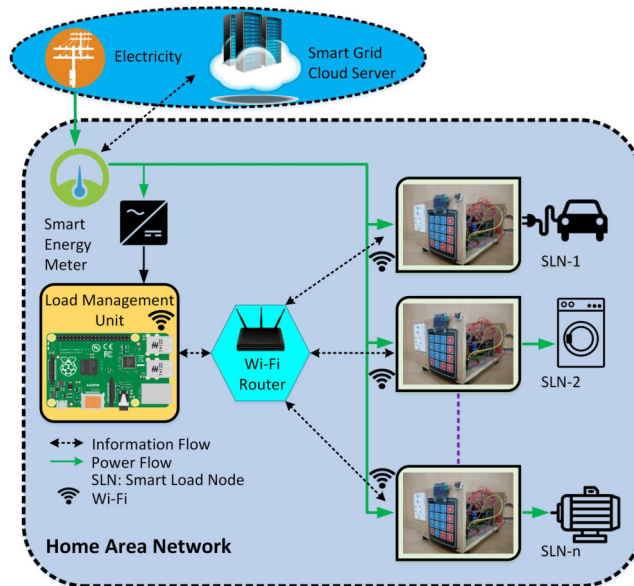


Рисунок 1.4 - LMU та SLN, підключені через Wi-Fi [2]

Основними елементами SLN є плата Arduino UNO R3 (Arduino), блоки вимірювання напруги та струму, модуль трансивера Wi-Fi ESP8266, реле, 0,96-дюймовий OLED-дисплей і матрична клавіатура. Блок-схема SLN зображена на рисунку 1.5.

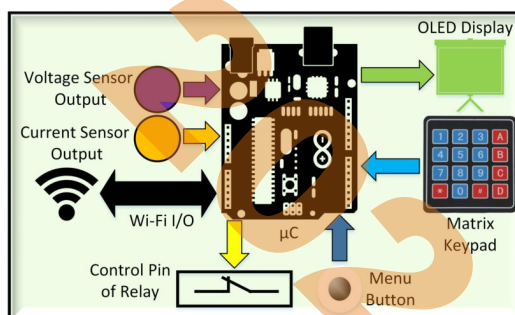


Рисунок 1.5 - Arduino з кількома пристроями введення/виведення [2]

Навантаження має бути підключено до розетки через SLN. Напруга живлення, яка відбирається від настінної розетки, знижується трансформатором і додається зі зміщенням постійного струму перед подачею на контакт A0 Arduino.

Датчик струму ACS712 підключається послідовно з фазою живлення змінного струму. Вихід датчика струму подається на контакт A2 Arduino. Отримані дані про напругу та струм від контактів A0 і A2 обробляються для розрахунку електричних параметрів навантаження. Спеціальний трансформатор служить для живлення різних елементів, присутніх у SLN. Трансивер Wi-Fi SLN

Вим.	Арк.	№ докум.	Підпис	Дата

має статичну IP-адресу. Він підключений до Arduino через послідовний порт для надсилання параметрів навантаження та конфігурацій навантаження, встановлених користувачем (zareestrovanih za dopomogoyu matrichnoyi klavivatury), до LMU та отримання інструкцій щодо роботи навантаження. Один із розроблених SLN, показаних на рис. 1.6a, має три піддисплеї в одному OLED-дисплеї. Інформація, яка відображається на рис. 1.6b, є стандартним відображенням SLN. Конфігурацію навантаження можна змінити, натиснувши кнопку меню. Вибір конфігурації вузла (рис. 1.6c) призводить споживача до екрана реєстрації даних (рис. 1.6d). У запиті на реєстрацію даних споживач повинен надати час початку ( $\_$ , інтервал запуску), час зупинки ( $\_$ , інтервал, за який навантаження має завершити роботу) та інтервал ( $\_$ , тривалість роботи навантаження в хвилинах). Вираз (1) передає, що після вибору режиму NINSL SLN не запитуватиме введення від користувача.

Якщо споживач вибирає один із режимів NISL або ISL, тоді потрібно надати як вхідні дані 1. LMU видає команду УВІМКНЕННЯ та ВИМКНЕННЯ між та з інтервалами, враховуючи тривалість роботи навантаження. У разі режиму NISL навантаження не повинно перериватися під час його роботи між 1. Після отримання команди ON завантаження NISL працюватиме безперервно протягом періоду.

У разі режиму ISL навантаження може бути перервано під час його роботи за допомогою кількох команд увімкнення/вимкнення, продиктованих алгоритмом DR, однак сума періодичного часу виконання навантаження має дорівнювати.

$$(\alpha, \beta, \gamma) = \begin{cases} (0,0,0) & \text{if } NINSL \\ (1,1,1) & \text{if } NISL \\ (1,1,1) & \text{if } ISL \end{cases} \quad (1.1)$$

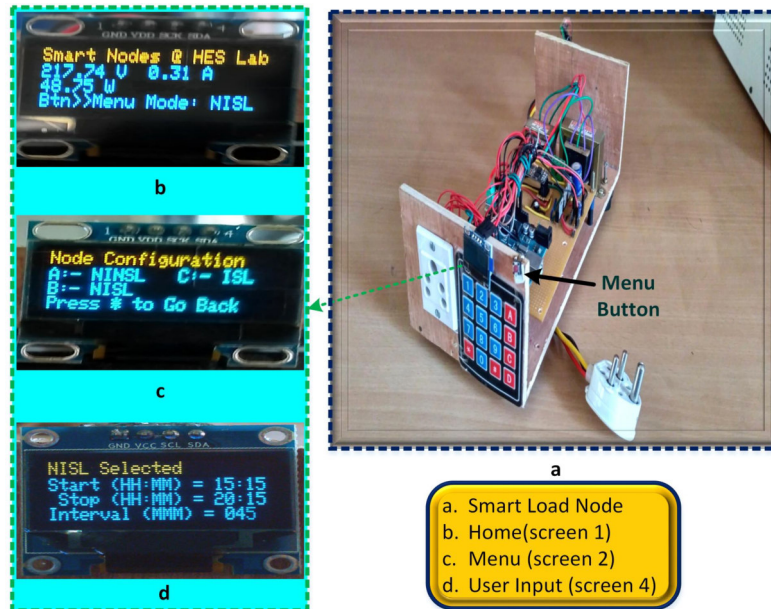


Рисунок 1.6 - . Розроблений SLN з інтерфейсом користувача [2]

LMU — це процесор Raspberry Pi 3 Model B, який виконує алгоритм керування навантаженням. LMU підключено до інтелектуального лічильника енергії (SEM), а SEM підключено до інтелектуальної мережі через Інтернет. SEM отримує значення параметрів, необхідних для включення DR в алгоритм керування навантаженням, і передає ці значення в LMU. На основі конфігурації, встановленої споживачем, SLN форматує зареєстровані вхідні дані та надсилає їх до LMU. Після отримання LMU розділяє дані за допомогою специфікаторів формату та зберігає відповідні значення окремо у своїй пам'яті. LMU синхронізовано з GMT +5:30 (індійський стандартний час). Сигнали ON і OFF генеруються LMU, які запускають вбудоване реле відповідного SLN.

На рисунок 1.7 зображено обробку даних і маршрутизацію з наступною операцією завантаження. Отримання даних LMU та запуск завантаження видно з оболонки Python LMU у HAN. На рисунок 1.8 - Ілюстрація алгоритму планування навантаження на основі динамічного пріоритету

Вим.	Арк.	№ докум.	Підпис	Дата

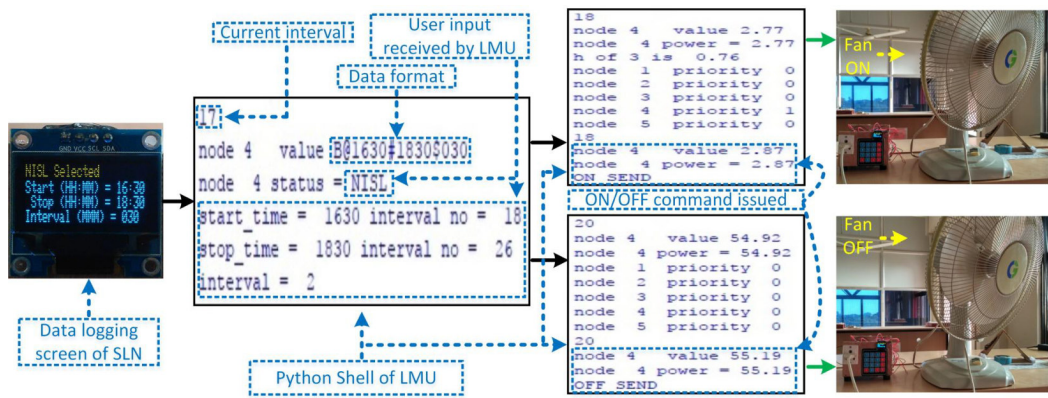


Рисунок 1.7 - Огляд роботи HAN, від реєстрації даних до операції завантаження [2]

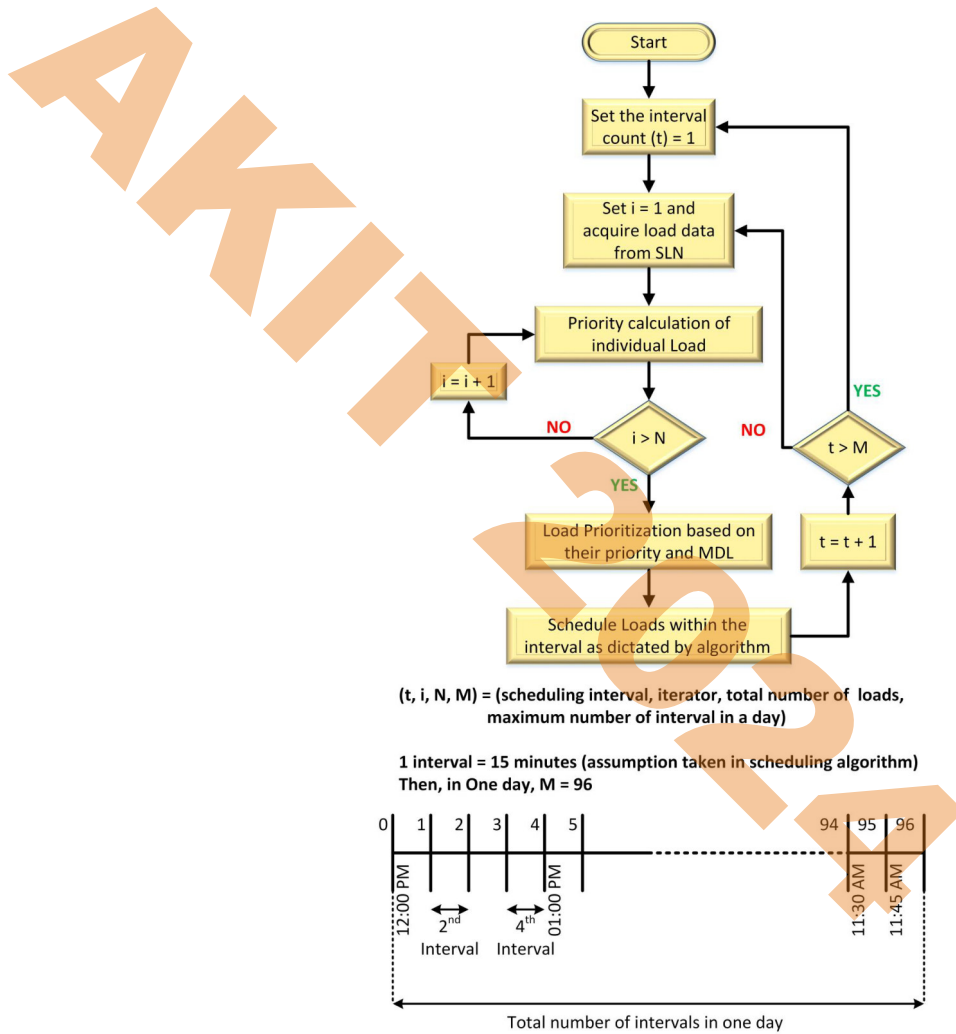
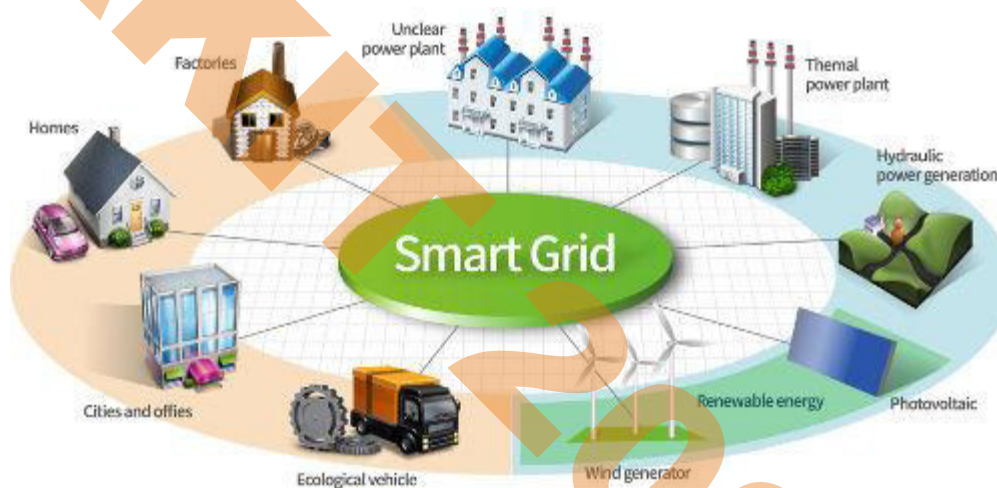


Рисунок 1.8 - Ілюстрація алгоритму планування навантаження на основі динамічного пріоритету [2]

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

### 1.3 Моніторинг і керування розумною електромережею на основі IoT за допомогою Arduino

Енергогенеруючі компанії постачають електроенергію всім домогосподарствам через проміжні контрольовані вузли передачі електроенергії, відомі як електрична мережа. Іноді проблеми виникають через збій в електромережі, що призводить до знеструмлення цілої території, яка одержувала живлення від цієї конкретної мережі. Проект має на меті вирішити цю проблему, використовуючи ІОТ як засіб зв'язку, а також вирішувати різні інші проблеми, з якими може впоратися інтелектуальна система, щоб уникнути непотрібних втрат для енергетичних процедур. На рисунку 1.9 зображено розумну сітку [3].



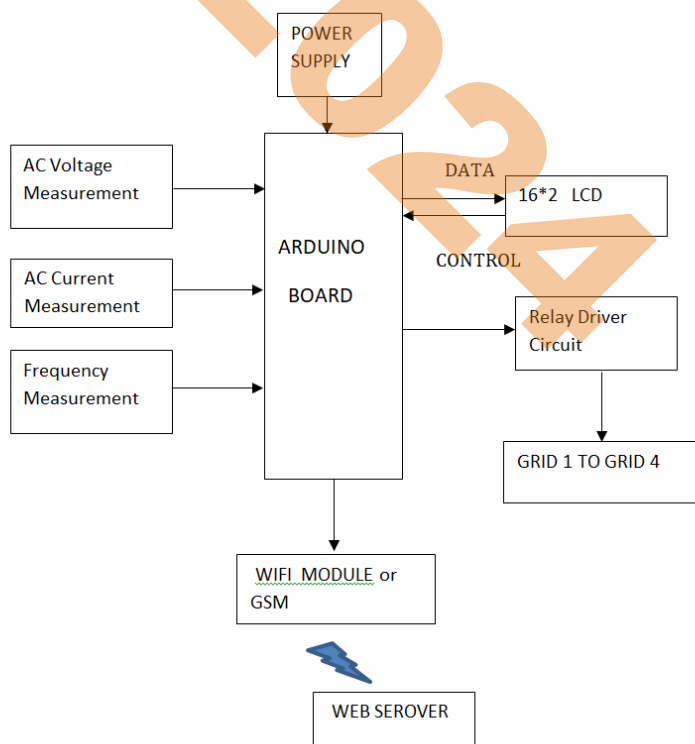
Рисунку 1.9 – Розумна сітка [3]

Інтелектуальна енергетична мережа ІОТ базується на контролері сімейства mega АТ, який контролює різні види діяльності системи. Система спілкується через Інтернет за допомогою технології Wi-Fi. Лампочка використовується в цьому проекті для демонстрації як дійсного споживача, а лампочка для демонстрації недійсного споживача. Найголовніше, чому сприяє цей проект, це перепідключення ЛЕП діючої мережі. Якщо енергетична мережа стає несправною та існує інша енергетична мережа, система перемикає лінії передачі на цю мережу, таким чином полегшуючи припинення постачання електроенергії до того конкретного регіону, енергомережа якого була ВИМКНЕНА. І ця інформація про те, яка сітка активна, оновлюється через веб-сторінку ІОТ Gesco, де органи влади

Вим.	Арк.	№ докум.	Підпис	Дата

можуть увійти в систему та можуть бути оновлення. Окрім моніторингу мережі, цей проект має розширені можливості моніторингу споживання енергії та навіть виявляє крадіжки електроенергії. Обсяг спожитої електроенергії та орієнтовна вартість споживання оновлюється на веб-сторінці IOT Gecko разом із інформацією про енергетичну мережу. Умови крадіжки імітуються в системі за допомогою двох перемикачів.

Огляд системи. Arduino UNO R3: на додаток до всіх функцій попередньої плати, Uno тепер використовує ATmega16U2 замість 8U2, який є в Uno (або FTDI, який є в попередніх поколіннях). Це забезпечує швидшу швидкість передачі та більше пам'яті. Не потрібні драйвери для Linux або Mac (потрібен файл inf для Windows, який включено в Arduino IDE), і можливість відображати Uno як клавіатуру, мишу, джойстик тощо. Uno R3 також додає контакти SDA та SCL поруч з AREF. Крім того, поруч із контактом RESET є два нових контакти. Одним з них є IOREF, який дозволяє екранам адаптуватися до напруги, що подається від плати. Інший не підключений і зарезервований для майбутніх цілей. Uno R3 працює з усіма існуючими екранами, але може адаптуватися до нових екранів, які використовують ці додаткові шпильки.



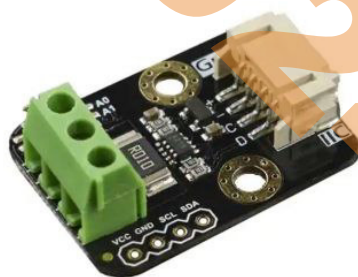
Рисунку 1.10 – Блок-схема системи [3]

Вим.	Арк.	№ докум.	Підпис	Дата

Arduino Uno має 14 цифрових входів/виходів (з яких 6 можна використовувати як ШІМ-виходи), 6 аналогових входів, кристалічний генератор 16 МГц, USB-з'єднання, роз'єм живлення, роз'єм ICSP і кнопку скидання. Містить все необхідне для підтримки мікроконтролера; просто підключіть його до комп'ютера за допомогою USB-кабелю або живіть за допомогою адаптера змінного струму в постійний струм або акумулятора, щоб розпочати роботу. Arduino Uno відрізняється від усіх попередніх плат тим, що не використовує чіп драйвера FTDI USB-to-serial. Натомість він оснащений чіпом мікроконтролера Atmega8U2, запрограмованим як перетворювач USB-последовний.

#### 1.4 Gravity: цифровий ватметр I2C

Gravity: Цифровий ватметр I2C — це великомасштабний вимірювальний прилад із високою роздільною здатністю, високою точністю, який може вимірювати напругу, струм і потужність різних електронних модулів і електричного обладнання в межах 26 В 8 А, а максимальна відносна похибка становить не більше  $\pm 0,2\%$  (Перед використанням потрібна проста ручна калібрування). Його можна використовувати для оцінки енергоспоживання або терміну служби батареї систем сонячної енергії, акумуляторів, двигунів, контролерів або електронних модулів [4].

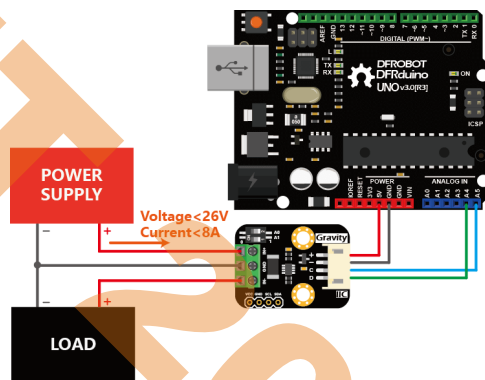


Рисунку 1.11 – Цифровий ватметр Gravity I2C [4]

Модуль використовує TI INA219 мікросхему моніторингу струму/потужності нульового зміщення температури та високопотужний резистор із сплаву 2 Вт 10 мОм. Роздільна здатність напруги та струму може досягати 4 мВ та 1 мА відповідно. За умов натурального вимірювання максимальна відносна похибка вимірювання напруги та струму може перевищувати  $\pm 0,2\%$ . Він також

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

забезпечує чотири адреси I2C, які можна налаштувати за допомогою 2P DIP-перемикача. Пристрій точно вимірює двонаправлені струми високої сторони (струм, що протікає через джерело живлення або плюс батареї), що особливо корисно в програмах вимірювання сонячних батарей або батареї, де батарею потрібно заряджати та розряджати. Цей стан можна просто визначити за позитивними чи негативними показниками струму. У двигунах струм можна контролювати в режимі реального часу шляхом відстеження, якщо струм двигуна занадто великий через перевантаження. Крім того, ви можете використовувати цей пристрій для вимірювання енергоспоживання різних електронних модулів або всього проекту для оцінки терміну служби батареї за схемою рис. 1.12. Також ще одне застосування ватметра це – контроль енергоспоживання сонячної панелі, батареї та виходів сонячної системи рис. 1.13.



Рисунку 1.12 – Схема підключення ватметра I2C (Arduino UNO) [4]

#### Характеристики Gravity

Тип датчика: поточний;

Типова вхідна напруга: 3,3 В постійного струму;

5В постійного струму;

Робочий струм: 1 мА;

Інтерфейс: цифровий;

Протокол зв'язку: I2C;

Роздільна здатність по напрузі: 4 мВ;

Відносна похибка напруги:  $\leq \pm 0,2\%$  (типова);

Діапазон струму: 0 ~  $\pm 8$  А (двонаправлений струм);

Роздільна здатність струму: 1 мА;

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Поточна відносна похибка:  $\leq \pm 0,2\%$  (типово, потрібне ручне калібрування);

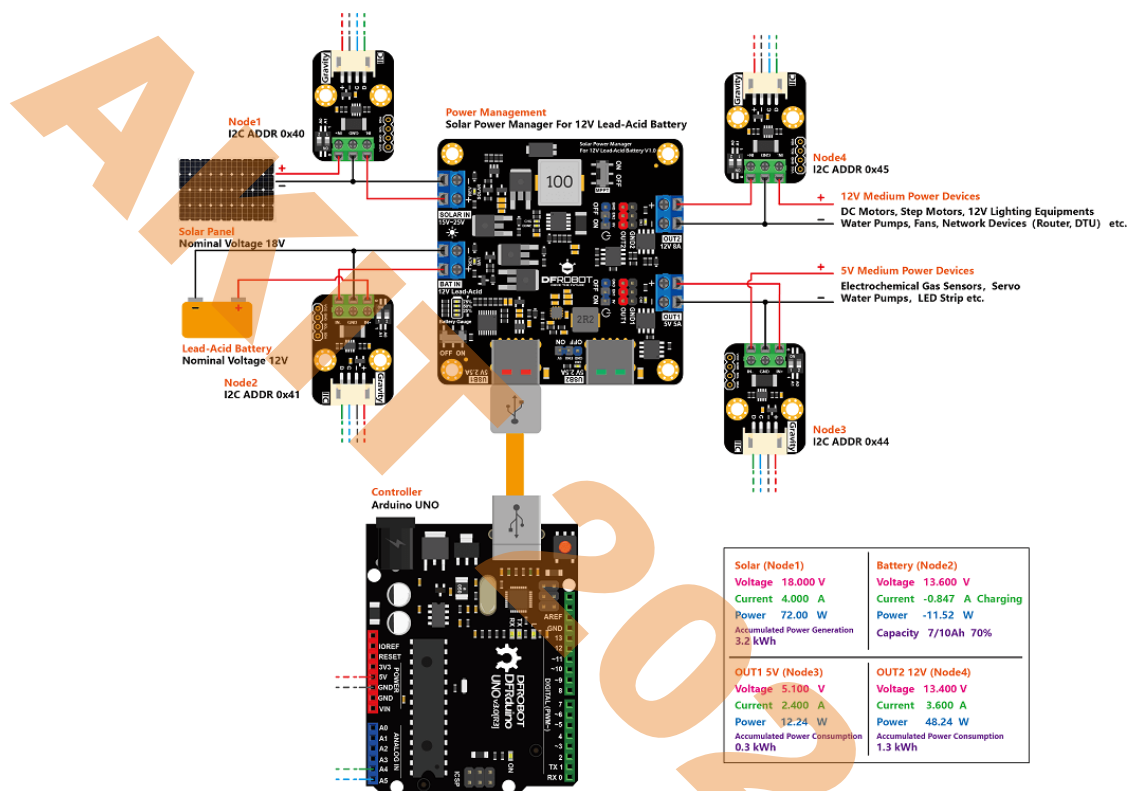
Діапазон потужності: 0~206 Вт;

Роздільна здатність потужності: 20 мВт (апаратне забезпечення) / 4 мВт (програмне забезпечення);

Струм спокою: 0,7 мА;

Інтерфейс: Gravity I2C (логічний рівень: 0-3,3 В);

I2C-адреса: чотири варіанти 0x40, 0x41, 0x44, 0x45.



Рисунку 1.13 – Схема контролю енергоспоживання сонячної панелі, батареї та виходів сонячної системи [4]

## 1.5 Системи моніторингу якості електричної енергії SMART GRID: метрологічний компас PQ5000CL

ТОВ «Електровимір» пропонує сертифіковане вимірювання якості електроенергії та аналіз потужності до 32 каналів в розподільному цеху. Обробка сигналу реалізована за допомогою вимірювального пристрою - багатофункціонального аналізатора якості електроенергії серії LINAX®

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

КБР.АКІТ.20050051.01.000 ПЗ

Арк.

22

PQ5000CL, в якому обробляються відповідні виміряні значення струму блоків живлення Current Link [5].

Завдяки технології Current Link окремі струмові модулі та їхні датчики Rogiv об'єднані в мережу за допомогою сигнального шлейфу з коаксіальними кабелями. Це спрощує витрати на введення в експлуатацію та забезпечує правильне встановлення датчика. Крім того, ця система визначення якості електроенергії та потоків навантаження є надзвичайно економічною та метрологічно сертифікованою. Таким чином, масштабований вимірювальний пристрій ефективно поєднує в собі перетворювачі відповідно до IEC 60688, пристрої вимірювання та контролю потужності згідно IEC 61557-12 і пристрої якості електроенергії згідно IEC 62586-1. LINAX® PQ5000CL це:

- метрологічно сертифікований PQ-аналізатор класу А відповідно до IEC61000-4-30.
- масштабована система для сертифікованих зон якості електроенергії, а також управління навантаженням і продуктивністю до 10 джерел живлення.
- додаткове базове вимірювання струму (наприклад, відразу після трансформатора) з високою точністю завдяки датчикам трансформатора струму.
- встановлення 3-х або 4-х канальних блоків Current Link на фідери (максимальний струм 32).
- синхронізований за часом процес вимірювання для кількох фідерів, на відміну від традиційного процесу вимірювання (1 пристрій на 1 фідер).
- миттєва звітність про відповідність та відображення подій через веб-браузер (згідно ДСТУ EN50160).
- синхронне керування навантаженням для  $U / I / P / Q / \cos\varphi$ ?
- Вимірювання струму на канал зв'язку до 1000 А і надструму до 20000 А.
- модернізація завдань управління в розумній мережі - розумна мережа (наприклад, з використанням мережі PQ).
- Немає необхідності припиняти роботу для встановлення вимірювальної системи завдяки знімним датчикам Роговського.
- дуже висока міцність завдяки перевіреному коаксіальному принципу.
- значення струму і напруги (IEC61000-4-30).

Вим.	Арк.	№ докум.	Підпис	Дата

КБР.АКІТ.20050051.01.000 ПЗ

Арк.

23

- різні комунікаційні інтерфейси (Modbus TCP/IP, Modbus RTU, REST API, IEC61850, хмара з MQTT, веб-браузер), які забезпечують високу гнучкість підключення до паралельних систем і систем вищого рівня.
- швидке введення в експлуатацію з надійною технологією вимірювання.

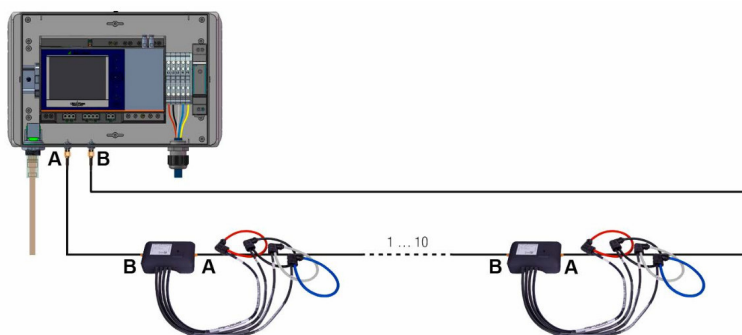


Рисунок 1.14 - LINAX® PQ5000CL-3 у польовому корпусі з підключеними силовими модулями 3PN [5]

Таблиця 1.1 - Характеристики PQ5000CL

Якість електроенергії	PQ5000CL - на DIN рейку	PQ5000CL - польовий корпус
Входи напруги	4	4
Струміві входи базового приладу	4 (опція)	
Струміві входи струмових модулів	до 32	до 32
Клас згідно з IEC 61000-4-30	A	A
Тип пристрою згідно з IEC 62586-1	PQI-A FI1	PQI-A FI1
<b>Похибка вимірювання</b>		
Напруга	±0,1%	±0,1%
Струм (базовий пристрій)	±0,1%	-
Потужність (базовий пристрій)	±0,5%	-
Активна енергія (базовий пристрій)	клас 0.5S (IEC 62053-22)	-
Модуль струму 3P/3PN	±0,5%	±0,5%
Модуль потужності 3P/3PN	±2.0%	±2.0%
Модуль активної енергії 3P/3PN	клас 3	клас 3
<b>Інтерфейси зв'язку</b>		
Ethernet: Modbus/TCP, Webserver, NTP	стандарт	стандарт
IEC 61850	опція	опція
MQTT (за спеціальним запитом)	опція	опція
<b>Живлення</b>		
Споживання енергії	100...230V AC 50/60Hz / DC ±15% окреме джерело живлення 24 VDC для кожного модуля ≤ 27VA, ≤ 12W	100...230V AC 50/60Hz /(internal) ≤ 60VA
Кольоровий дисплей (опціонально)	TFT 3,5" (320x240px)	TFT 3,5" (320x240px)
Розміри	160 x 110 x 70 мм	271 x 170 x 90 мм
Спосіб монтажу	на DIN рейку	настінне кріплення

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

КБР.АКІТ.20050051.01.000 ПЗ

Арк.

24

## 2 ПРОЕКТНО - КОНСТРУКТОРСЬКИЙ РОЗДІЛ

### 2.1 Теоретичні основи Інтернет речей

Термін «Інтернет речей» (IoT) був вперше використаний у 1999 році британським піонером технологій Кевіном Ештоном для опису системи, у якій об'єкти у фізичному світі можуть бути підключені до Інтернету [6]. Ашон мав на меті підкреслити потужність систем радіочастотної ідентифікації, відомих як RFID (Радіочастотна ідентифікація), які використовуються корпоративними ланцюгами поставок для вимірювання та відстеження товарів без потреби втручання людини.

Сьогодні «Інтернет речей» став популярним терміном для опису сценаріїв, у яких підключення до Інтернету та обчислювальні можливості розширюються, щоб сформувати мережу фізичних об'єктів, пристроїв, датчиків і взагалі будь-якого об'єкта штучного людського середовища. Кожен із цих об'єктів містить вбудовану технологію, щоб він міг взаємодіяти з внутрішніми станами чи зовнішнім середовищем, збирати та обмінюватися даними. Підключення до локальної мережі або Інтернету дозволяє користувачеві контролювати датчики, а також дистанційно керувати пристроєм.

Інтернет речей можна розглядати як ідею максимального використання та розвитку взаємопов'язаних технологій, зокрема Інтернету. Це дедалі більше проникнення мікрокомп'ютерів і телекомунікацій у людське та природне середовище.

#### 2.1.1 Зв'язок між машинами (M2M)

Термін зв'язок між машинами (M2M) означає прямий зв'язок між пристроями через будь-який канал зв'язку (наприклад, дротовий або бездротовий). Комунікація M2M тісно пов'язана з Інтернетом речей, оскільки пристрої повинні спілкуватися один з одним для автоматичного обміну даними. Підтримка M2M базується на реалізаціях, розроблених різними організаціями. Більш відомими організаціями є робоча група Eclipse machine to machine, OASIS MQTT, XMPP і Open Mobile Alliance [7].

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

## 2.1.2 Приклади протоколів і архітектур

Необхідною умовою для створення мережі Інтернету речей є спроба комплексної інтеграції безлічі закритих мереж і різних протоколів зв'язку, що застосовуються на кожному рівні. Ці рівні, визначені для кращого дизайну мережі, можна включити до наступних основних категорій:

- Фізичний рівень передачі даних
- Протокол передачі даних
- Протокол роботи мережі (IoT).

Перша категорія відноситься до найнижчого рівня мережевого стеку. Він визначає фізичне середовище передачі та конфігурацію інформації на ньому. Особливий інтерес в Інтернеті речей представляють бездротові технології [8], оскільки більшість програм засновані на бездротовому зв'язку пристроїв.

Протокол передачі даних є другою базовою категорією і належить до проміжного рівня мережевого стеку IoT, він визначає правила, процедури та способи, якими ця мережа визначається та працює. На цьому рівні передбачається використання Інтернет-протоколу (IP). Цей протокол є основою сучасного Інтернету та домінуючої взаємопов'язаної технології. Його 6-та версія (IPv6) зможе одночасно підтримувати 2<sup>128</sup>, тобто 3,4 x 10<sup>38</sup> різних адрес і, як наслідок, пристроїв.

Третя категорія належить до найвищого рівня мережевого стеку, де зв'язок тепер є передачею даних між вузлами/пристроями IoT. Ця категорія включає протокол зв'язку, який визначає правила, за якими працює мережа IoT. З розвитком існуючих протоколів і розробкою нових потрібно вирішити різноманітні проблеми щодо функціональності, продуктивності, безпеки тощо. Розглянемо один із них.

MQTT (телеметричний транспорт черги повідомлень)

Цей протокол, розроблений для вузлів/пристроїв з обмеженою пропускнуою здатністю. Архітектура тут також клієнт-сервер, однак використовується комунікаційна модель публікація-підписки. Сервер — це проксі-сервер, до якого всі вузли мережі підключаються як клієнти. Розумні об'єкти мережі (речі) — це

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

видавці, які надсилають дані посереднику, тоді як клієнти, зацікавлені в даних цих речей – це передплатники.

Ця модель зв'язку має велике значення в мережах IoT. Особливо якщо ми врахуємо, що більшість речей мають обмежені можливості обробки, тоді як вузли, зацікавлені в даних речей, є більш потужними пристроями, тоді важливо, щоб речі надсилали єдине повідомлення вузлам для кожного оновлення даних своїх датчиків, а не сповіщати кожен зацікавлену сторону окремо, можливість, передбачена протоколом MQTT. У наступному підрозділі аналізується режим його роботи.

### 2.1.3 MQTT

MQTT [9], як уже згадувалося, є протоколом публікації та підписки для використання через TCP/IP і за замовчуванням використовує порт 1883. Він легкий, простий і простий у реалізації, функції, які роблять його ідеальним для додатків IoT, де пропускна здатність мережі обмежена. і потрібен «невеликий слід коду». Навіть низьке енергоспоживання є ще однією причиною, чому його часто вибирають для мобільних додатків.

Одна з тем дебатів обертається навколо значення акроніму MQTT. Коротка відповідь полягає в тому, що абревіатура більше не має визначення, а протокол називається просто MQTT. У більш широкому сенсі, як абревіатура була MQ Telemetry Transport, де MQ відноситься до продукту IBM (International Business Machines), серії MQ, який підтримував MQTT і, отже, назва протоколу. В Інтернеті ми помилково помічаємо, що він також характеризується як послідовний протокол повідомлень (протокол черги повідомлень). Хоча протокол підтримує серіалізацію, він не відповідає традиційній концепції послідовного протоколу, як в інших традиційних системах черги повідомлень, оскільки черг немає.

#### 2.1.3.1 Модель публікації/підписки

Філософія MQTT відрізняється від класичного підходу клієнт-сервер, коли клієнт спілкується безпосередньо з сервером і навпаки. Модель публікації-підписки є альтернативою традиційній моделі клієнт-сервер, оскільки тут клієнт,

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

який надсилає повідомлення, не знає про існування одержувача, а зв'язок базується на існуванні проміжної сутності.

Комунікуючими суб'єктами є видавці, передплатники або обидва. Комунікація відбувається, коли видавці створюють повідомлення, які підписники отримують (споживають).

Кожне повідомлення належить до певної теми. Темі — це рядки в ієрархічній структурі, розділені символом косої риски (/). Абонент підписується на одну або кілька цікавих тем і отримує повідомлення лише з цих тем. Відповідно, видавець, який створює повідомлення, класифікує його в тему, тому його отримують лише ті, хто підписався на цю тему.

Здійснення комунікації здійснюється проміжною організацією, яка називається брокером і відома як видавцям, так і передплатникам. Отримує повідомлення від видавців і пересилає їх зацікавленим передплатникам.

По суті, за допомогою цієї моделі ми досягаємо 3 видів роз'єднання.

- Просторова роз'єднаність. Видавець і передплатник не знають про існування один одного.
- Тимчасове відключення. Видавцеві та передплатнику не потрібно запускати одночасно.
- Розрив синхронізації. Усі сторони (видавець, передплатник, брокер) не переривають свою роботу під час надсилання чи отримання повідомлень.

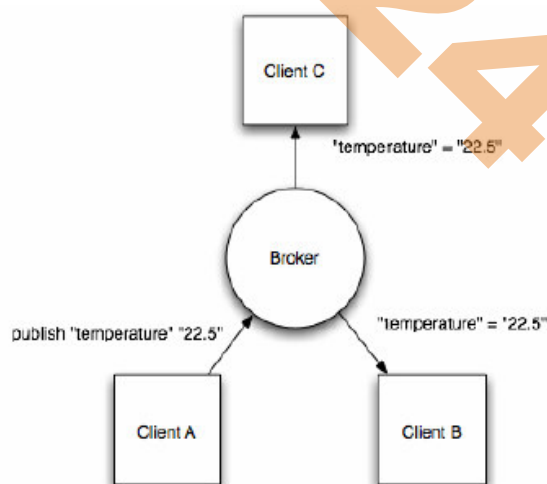


Рисунок 2.1 - Модель публікації/підписки [10]

Вим.	Арк.	№ докум.	Підпис	Дата

Припустімо, що клієнт А на малюнку 2.1 знаходиться у вітальні та хоче оновити значення температури. Можливий сценарій - надсилання значення в тему Дім/вітальня/температура, і кожен, хто хоче отримувати інформацію про цю температуру, повинен підписатися на відповідну тему.

Ця структура дозволяє абоненту підписуватися на кілька тем одночасно за допомогою символів підстановки (#, +). Символ «#» розміщується строго в кінці запитуваної теми та відповідає всім рівням, що залишилися, що дозволяє нам підписатися на все піддерево. Зокрема, підписавшись на тему Home/livingroom/#, дані як датчика температури, так і будь-якого іншого датчика пристрою, який надсилає дані в розділі Home/livingroom/, наприклад Home/livingroom/humidity, Home/livingroom/light\_switch тощо.

Іншим важливим символом підстановки є символ «+». На відміну від «#», «+» замінює один рівень. Замість «+» буде замінено все, що відомо Брокеру, щоб завершити тему. Наприклад, Home+/light\_switch відповідатиме як Home/bedroom/light\_switch, так і Home/livingroom/light\_switch, створюючи тут сценарій домашнього освітлення, який впливає як на спальню, так і на вітальню.

Великою перевагою моделі Pub/Sub перед традиційним клієнт-сервером є те, що процеси в посереднику є високорозпаралелізованими, що дозволяє нашим програмам значно масштабуватися. Крім того, багаторівнева структура тем з належним дизайном дозволяє плавно розширювати систему, коли це необхідно. Недоліком може бути те, що як видавець, так і передплатник повинні бути знайомі зі структурою тем.

### 2.1.3.2 Підключення клієнта та брокера

#### Клієнти

Клієнт MQTT може бути будь-яким типом пристрою (від простого мікроконтролера до сучасної та потужної серверної машини), який містить бібліотеку реалізації MQTT і може бути підключений через будь-яку мережу до брокера. У випадку Інтернет-середовища (TCP/IP) бібліотека MQTT реалізована на прикладному рівні, як і інші подібні протоколи, наприклад HTTP (рис. 2.2).

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

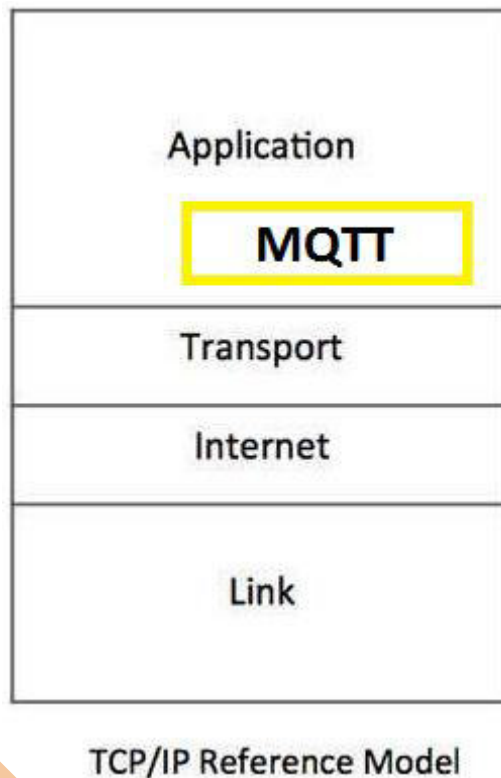


Рисунок 2.2 - Місце протоколу MQTT в ієрархії рівня TCP/IP [10]

### Брокер

Серцем будь-якої моделі pub/sub є брокер, який отримує всі повідомлення, фільтрує їх і вирішує, яким передплатникам їх пересилати. Він також автентифікує всіх клієнтів за допомогою методів автентифікації та перевіряє авторизованих абонентів, якщо така реалізація була додана. Крім того, він зберігає всі дані про клієнтів, які встановили «постійне» з'єднання (постійні клієнти). Нижче наведено більше інформації про це.

Брокер, будучи сервером системи, повинен дозволити легке масштабування (розширюваність), щоб він міг впоратися з робочим навантаженням. Таким чином, залежно від реалізації, він може керувати тисячами підключень одночасно.

### Налаштування підключення

З'єднання MQTT завжди встановлюється між клієнтом і посередником і ініціалізується, коли клієнт надсилає посереднику повідомлення `CONNECT`. Якщо це можливо, після відповіді брокера надсилається повідомлення `CONNACK` (підтвердження з'єднання) і статус з'єднання. Посередник підтримуватиме з'єднання, доки не отримає від клієнта команду на роз'єднання. `CONNECT` містить таку інформацію:

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

- `clientId` – унікальний ідентифікатор для кожного клієнта MQTT. Якщо клієнт не вкаже, брокер надасть йому випадкове число.

- Чистий сеанс – ця змінна визначає, чи буде підключення клієнта постійним (постійний сеанс). Таке підключення (`Clean Session = false`) означає, що всі підписки та втрачені повідомлення (з `QoS>0`), які стосувалися цього клієнта і з якихось причин не були доставлені (можливе відключення), будуть зберігатися в брокері.

- Ім'я користувача/пароль – вони дозволяють ідентифікувати клієнта, якщо щось подібне потрібно.

- `Will Message` – ця функція дозволяє оновлювати клієнти після раптового відключення клієнта від системи. Коли брокер розуміє, що клієнт-емітент припинив роботу, він пересилає повідомлення відповідним клієнтам.

`KeepAlive` – це період часу, протягом якого клієнт зобов'язується надіслати запит PING, а брокер – відповідь PING, щоб обидві сторони підтвердили, що з'єднання активне.

### 2.1.3.3 Якість послуг (QoS)

Якість послуг використовується в системах зв'язку для вказівки продуктивності системи, як її сприймають її користувачі. У MQTT QoS — це угода між відправником і одержувачем про те, як гарантовано повідомлення буде доставлено до місця призначення.

Публікація повідомлення клієнтом-видавцем, як обговорювалося, передбачає, що це повідомлення досягне клієнтів-передплатників у два етапи: (а) від клієнта-емітента до брокера та (б) від брокера до клієнта-передплатника.

Для кроку (а) QoS вказується видавцем (команда публікації) і визначається для кожного повідомлення. Для (б) QoS визначається шляхом підписки підписника на конкретну тему (команда підписки). Таким чином, QoS може бути знижено для одержувача, який підписався з нижчим QoS. У MQTT є три рівні QoS:

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

QoS(0): мінімальний рівень QoS, повідомлення надсилається один раз, підтвердження доставки не отримується та не зберігається. Зазвичай він настільки надійний, наскільки надійний протокол TCP, через який він надсилається.

Цей рівень вибирається, коли є фіксований посередник і клієнтське підключення (зазвичай дротове).

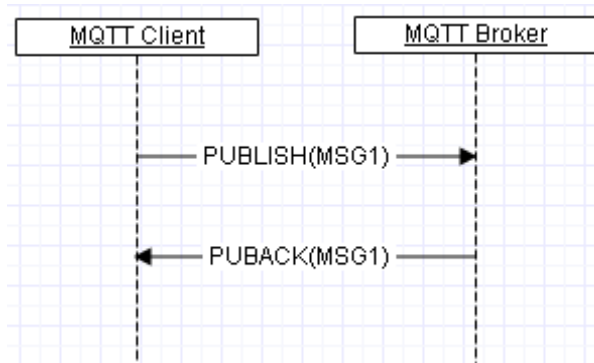


Рисунок 2.3 - Публікація повідомлення з QoS = 0

QoS(1): Варіант 1 гарантує, що повідомлення буде доставлено одержувачу принаймні один раз. Для кожного повідомлення, яке отримує брокер, він надсилає повідомлення підтвердження (puback) відправнику, що означає, що повідомлення досягло місця призначення. Якщо певний проміжок часу минає, а повідомлення PUBACK не отримано, відправник намагається повторно надіслати те саме повідомлення. Таким чином існує висока ймовірність того, що повідомлення дійде до одержувача більше разів. Цей рівень вибирається, коли мета полягає в тому, щоб забезпечити надходження інформації, і користувач повинен обробити додаткову інформацію (дублікати).

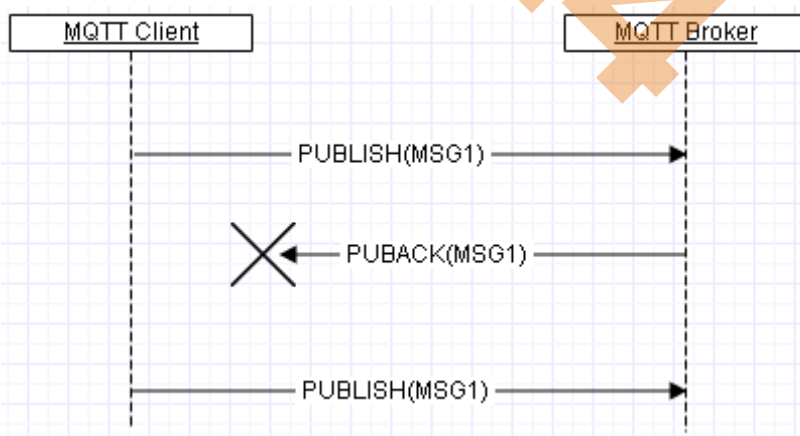


Рисунок 2.4 - Публікація повідомлення з QoS = 1

QoS (2): цей рівень гарантує, що повідомлення буде доставлено одержувачу рівно один раз. Гарантія походить від двох потоків повідомлень (подвійний

зв'язок) між відправником і одержувачем із загалом чотирма етапами (4-етапне рукописання), як показано на рис. 2.5.

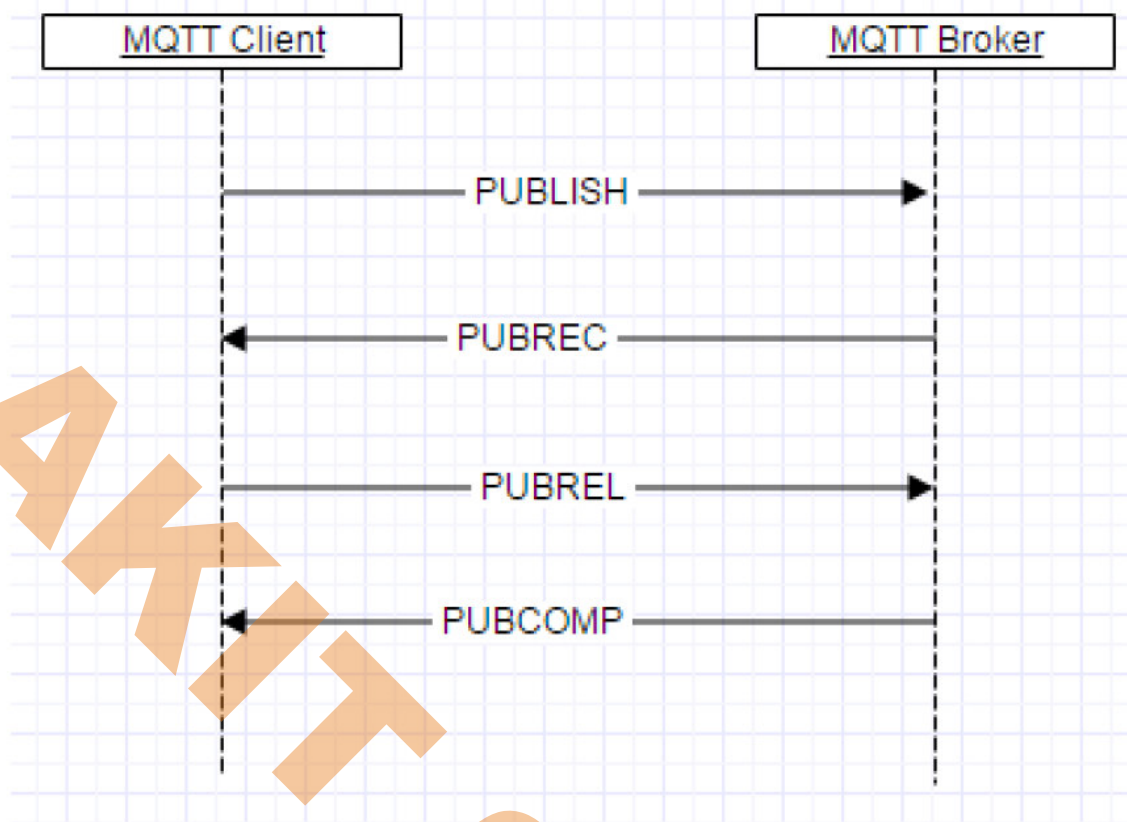


Рисунок 2.5 - Публікація повідомлення з QoS = 2

Коли брокер отримує повідомлення PUBLISH, він утримує його, доки не надішле пакет PUBCOMP. Якщо клієнт надсилає повідомлення вдруге, тому що відповідь була затримана, брокер перешле лише одне. Цей рівень вибирається, коли програмі дуже важливо, щоб усі повідомлення отримувалися рівно один раз. Він найповільніший і найбільш ресурсомісткий, оскільки вимагає більше зв'язку та використання пам'яті до завершення дії, але ці накладні витрати від подвійної перевірки не є таким недоліком.

#### 2.1.3.4 Постійне з'єднання та серіалізація повідомлень

Коли клієнт підключається до брокера, він повинен підписатися на всі теми, які його цікавлять. Під час цього повторного підключення процес потрібно повторити, якщо з'єднання не є «постійним». Щоб зробити з'єднання постійним (постійним сеансом), змінна cleanSession повинна бути встановлена на false під

час з'єднання. Якщо цей клієнт уже збережено, це повторне підключення. У цьому випадку йому буде надіслано всю інформацію, яку зберіг для нього брокер. Інформація включає наявність підключення, попередні підписки та всі повідомлення з QoS 1 або 2, які не були підтвержені як отримані клієнтом. Щоб очистити постійне з'єднання від посередника, той самий клієнт (clientID) має встановити з'єднання з trueSession.

Зі сказаного вище можна зробити висновок, що MQTT можна використовувати як класичний послідовний протокол повідомлень, оскільки клієнт має можливість зберігати всі повідомлення, коли він не підключений, і отримувати їх, коли він знову підключається.

У класичному протоколі послідовних повідомлень повідомлення серіалізуються до тих пір, поки споживач-клієнт не отримає їх. Якщо самовивозу немає, вони залишаються «застряглими» в черзі в очікуванні споживання. Навпаки, це обмеження не існує в протоколі MQTT. У випадку, якщо на цю тему не підписаний клієнт, повідомлення нікому не буде надіслано, оскільки немає зацікавленого одержувача. Крім того, велика різниця полягає в тому, що в класичному протоколі послідовних повідомлень кожне повідомлення має лише одного одержувача, тоді як, як ми бачили в MQTT, одержувачами можуть бути ті, хто підписався на відповідну тему.

### 2.1.3.5 Безпечне підключення до MQTT

#### Атестація

Протокол MQTT надає можливість аутентифікації при підключенні до брокера (поля імені користувача, пароля). Надсилання імені користувача та пароля брокеру передбачає належне налаштування останнього з реалізацією файлу з дозволеними користувачами, щоб їхню інформацію можна було перевірити під час підключення. Однак слід зазначити, що дані надсилаються у вигляді звичайного тексту. Щоб запобігти будь-якій можливості крадіжки під час передачі, дані можна зашифрувати.

Іншим методом є використання сертифіката, такого як X.509, який надається брокеру під час рукоштовання TLS. Кілька брокерів дозволяють використовувати

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

інформацію з сертифіката на рівні програми, але лише після успішного рукостискання.

### Шифрування

Сертифікація — це практика безпеки на рівні програми. На рівні передачі даних у MQTT можна реалізувати протоколи шифрування TLS (Transport Layer Security) і SSL (Secure Sockets Layer), використовуючи порт 8883. Протоколи TLS/SSL використовують механізми рукостискання для встановлення відповідних параметрів для створення безпечного з'єднання між клієнт і сервер. У зв'язку з цим дані зараз передаються в зашифрованому вигляді, і з сучасною технологією їх дуже важко прочитати.

Це особливо рекомендується, коли використовується ім'я користувача та пароль. Якщо пристрій не підтримує з'єднання TLS, альтернативним методом є шифрування повідомлень (поле корисного навантаження в команді публікації) і коду (у команді підключення) на рівні програми.

## 2.2 Теоретичні основи вимірювання

### 2.2.1 Трансформатори струму

Найбільш поширеним способом вимірювання струму в енергосистемах є трансформатор струму (Current Transformer – СТ). Трансформатор струму призначений для перетворення первинного струму енергосистеми в сигнал, який можна використовувати електромеханічними та електронними пристроями, такими як захисні реле, перетворювачі тощо. Через те, що існує широкий спектр пристроїв, які можна підключити, характеристики кожного трансформатора відіграють вирішальну роль у виборі правильного для кожного застосування [11].

Трансформатор з'єднаний послідовно з пристроєм, який створює вимірюваний струм і перетворює струм первинної обмотки у вторинну відповідно до співвідношення:

$$I_p = n \cdot I_s$$

2.2

де

$I_p$  - первинний струм;

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

$I_s$  - струм вторинної обмотки;

$n$  - співвідношення обмоток між первинною та вторинною обмотками.

Тобто: у вторинній обмотці відповідно до співвідношення:

$$n = \frac{N_s}{N_p}$$

Розміщуючи опір навантаження  $R_B$  (опір навантаження) у вторинній обмотці, вибір якого визначає амплітуду вихідної напруги, яку має створити трансформатор, із закону Ома випливає, що:

$$V_O = R_B \cdot I_s \quad (2) \quad 2.2$$

Комбінуючи співвідношення (2) із (1), що з'єднує первинний і вторинний струм, ми отримуємо:

$$V_O = R_B \cdot \frac{I_P}{n}$$

З наведеної вище формули ми можемо визначити значення вихідної напруги в залежності від потрібного нам застосування.

Перетворювачі на ефекті Холла

Перетворювачі на ефекті Холла [12] засновані на ефекті Холла, який викликається силами Лоренца, що діють на рухомі заряди в магнітному полі. Сила дорівнює:

$$F_L = q \cdot (V \times B)$$

Представлення ефекту Холла показано на малюнку 2.6.

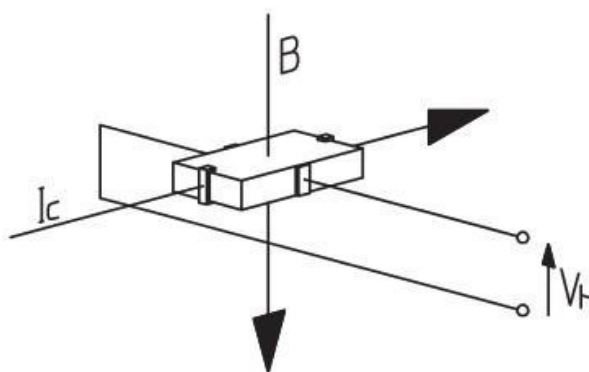


Рисунок 2-6: Ефект Холла [12]

Вим.	Арк.	№ докум.	Підпис	Дата

Тонкий лист індуктивного матеріалу протікає через струм  $I_c$ . На рухомі електричні заряди струму впливає зовнішній магнітний потік  $B$ , який створює силу Лоренца  $FL$ , перпендикулярну до напрямку струму. Результатом відхилення струму є створення більшої кількості електричних зарядів, які накопичуються на краю листа, створюючи різницю потенціалів, напругу Холла,  $V_H$ . Для наведеного вище розташування ми отримуємо співвідношення:

$$V_H = \frac{K}{d} \cdot I_c \cdot B \cdot V_{OH}$$

де

$K$  - постійна Холла індукваного матеріалу

$d$  - товщина листа

$I_c$  - струм, що проходить через лист

$B$  – напруженість магнітного поля

$V_{OH}$  - напруга Холла за відсутності зовнішнього магнітного поля

### 2.2.2 Активне значення (RMS значення) форми сигналу змінного струму

Середньоквадратичне значення сигналу або середньоквадратичне значення є показником амплітуди сигналу. Він в основному застосовується до змінних величин, тобто величин, які іноді є від'ємними, а іноді додатними.

Датчики струму можуть вимірювати постійні (DC) і змінні (AC) струми. У безперервному сигналі немає сенсу вимірювати активні значення і потужність розраховується з середніх значень величин. На відміну від цього, у сигналі змінного струму під час вимірювання вихід датчика дає негативні та позитивні значення, а форма вихідного сигналу пропорційна вхідному сигналу змінного струму, який вимірюється. У цьому випадку потрібно розрахувати середньоквадратичне значення форми вихідного сигналу, щоб отримати правильні показання [13].

Ефективне значення змінної напруги або середньоквадратичне значення (середньоквадратичне значення) визначається як значення постійної напруги, яке виробляє таку саму середню потужність, що й змінна напруга.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Загалом, для будь-якої періодичної форми сигналу  $V(t)$  ефективне значення  $V_{RMS}$  залежить від періоду  $T$  через співвідношення:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T V^2(t) dt}$$

У випадку синусоїдальної форми (змінний струм змінного струму) середньоквадратичне значення змінної напруги визначається співвідношенням:

$$V_{RMS} = V_{pk} \frac{1}{\sqrt{2}} = V_{pk} \times 0.7071$$

### 2.2.3 Аналого-цифрові перетворювачі - АЦП

У системах збору та перетворення даних необхідно вводити сигнали з одного або кількох джерел і перетворювати їх у цифрові сигнали для аналізу чи передачі на термінальний пристрій або відображення на цифровому дисплеї. У більшості випадків аналогові сигнали надходять у систему через датчики, які перетворюють фактичні величини в електричні сигнали, які, у свою чергу, оцифровуються за допомогою перетворювача.

Аналого-цифрові перетворювачі (АЦП) перетворюють аналоговий сигнал або величину, характерну для явищ реального світу, на цифрову мову, яка використовується в обробці інформації, обчисленні, передачі даних, а також системах керування. Зворотний процес виконується цифро-аналоговими перетворювачами (DAC), які використовуються для перетворення, передачі та збереження результатів цифрової обробки назад у реальний світ.

Найважливішим у аналого-цифрових перетворювачах є те, що вихідний сигнал є цифровим, тому сигнал виходить із перетворювача квантованим. Іншими словами,  $N$ -розрядне слово може мати лише  $2^N$  можливих станів. Тому  $N$ -розрядний перетворювач має лише  $2^N$  можливих цифрових виходів.

### 2.2.4 Дільник напруги

Підключення виходу датчика до входу системи зазвичай виконується за допомогою дільника напруги, який регулює вихідну напругу датчика до бажаних меж вхідної напруги перетворювача.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Дільник напруги є найпоширенішим методом вимірювання постійної та змінної напруги в цифрових вимірювальних приладах. Це проста схема, що складається з двох послідовно з'єднаних резисторів, на кінцях яких подається вхідна напруга. За вихідну напругу приймається різниця потенціалів між висновками одного з двох резисторів.

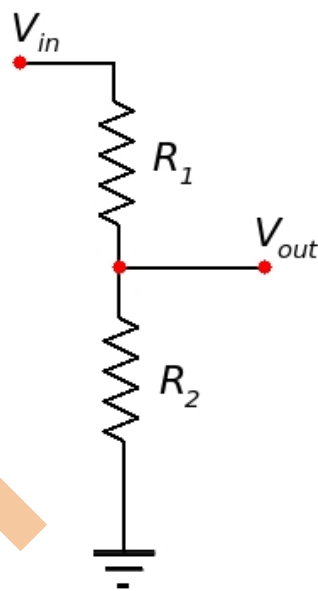


Рисунок 2.7 - Принципова схема дільника напруги

Відповідно до теорії електричних кіл напруга, що протікає через ланцюг двох резисторів, становить:

$$I = \frac{V_{in}}{R_1 + R_2}$$

Вихідна напруга буде:

$$V_{out} = I \cdot R_2 \Rightarrow V_{out} = R_2 R_1 + R_2 \cdot V_{in}$$

З наведеного вище аналізу випливає, що при відповідному виборі значень електричних опорів вихідна напруга може бути в межах вхідного діапазону інтеграторів, що використовуються для кожного вимірювання величин. Таким чином ми регулюємо вимірювану вхідну напругу до допустимих меж для перетворювача.

Вим.	Арк.	№ докум.	Підпис	Дата

## 2.3 Матеріали

Початковою метою було отримати необхідний матеріал для реалізації пристрою. Тому на цьому етапі було визнано необхідним визначити технічні характеристики та характеристики компонентів, які забезпечать виконання поставленої задачі.

### 2.3.1 Arduino Uno

Arduino [14] — це обчислювальна платформа, заснована на простій материнській платі з відкритим вихідним кодом із вбудованим мікроконтролером і середовищем розробки для легкого та простого написання програмного забезпечення, яке можна використовувати для створення цифрових пристроїв та інтерактивних об'єктів.

На платах використовуються різні версії 8-розрядних мікроконтролерів Atmel AVR або 32-розрядних процесорів Atmel ARM. Ці системи забезпечують набори цифрових і аналогових терміналів вводу/виводу, до яких можна підключати різні плати розширення та інші схеми. Arduino має послідовний інтерфейс, а мікроконтролер ATmega підтримує послідовний зв'язок, який Arduino передає через контролер Serial-over-USB для підключення до комп'ютера через USB. Це з'єднання використовується для передачі розроблених комп'ютером програм на Arduino, а також для двостороннього зв'язку між Arduino та комп'ютером через програму під час її роботи.

Для програмування мікроконтролерів платформа Arduino забезпечує інтегроване середовище розробки (IDE) відповідно до програми Processing, яка може підтримуватися мовами програмування C і C++.

Мікроконтролер запрограмовано на заводі за допомогою «завантажувача», тобто внутрішнього коду, тому зовнішній програматор не потрібен. Arduino доступний у багатьох версіях залежно від потреб і можливостей кожного програміста. Однак зазвичай усі плати програмуються через послідовне з'єднання RS-232, але те, як це реалізовано, залежить від версії. Послідовні плати Arduino містять просту інвертуючу схему для перетворення між сигналами рівня "RS-232" і "TTL" [15].

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Контакт Arduino Uno може спілкуватися з комп'ютером, іншим Arduino або іншими мікроконтролерами. Мікроконтролер ATmega328 забезпечує послідовний зв'язок за допомогою UART TTL (5 В), який доступний з контактів 0 (отримання RX) і 1 (передача TX).

Arduino з підключенням через порт USB відображається як віртуальний послідовний COM-порт у програмному забезпеченні комп'ютера, оскільки мікроконтролер ATmega16U2 перетворює послідовний зв'язок у формат USB. Крім того, Arduino містить сегментний контролер, який дозволяє простий текстовий формат даних, що надсилаються на плату Arduino та з неї. Світлодіоди RX і TX на платі блиматимуть, коли дані передаються через USB-порт (USB-послідовне підключення до комп'ютера).

Arduino Uno Rev 3. Що стосується архітектури Arduino Uno Rev 3 [16], то це платформа, заснована на ATmega 328 від Atmel. Він містить 14 цифрових входів-виходів ("I/O"), 6 аналогових входів для зчитування аналогових значень із зовнішніх компонентів (потенціометрів, аналогових датчиків) і 6 цифрових виходів, які можна використовувати як ШІМ (широтно-імпульсна модуляція), тобто мати здатність виробляти імпульси змінної амплітуди. Він також включає керамічний кристалічний генератор на 16 МГц.

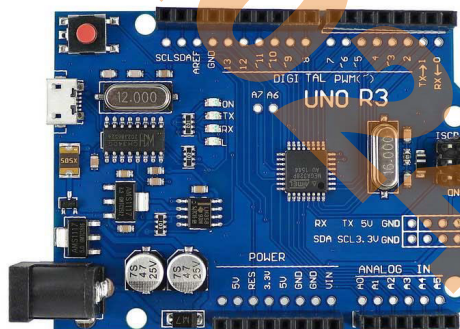


Рисунок 2.8 - Arduino Uno R3 [16]

Підключення здійснюється за допомогою кабелю USB (Universal Serial Port), при цьому також є розетка для підключення до живлення, а також можливість внутрішньосхемного послідовного програмування-ICSP ("In-Circuit Serial Programming") і скидання кнопка на випадок короткого замикання платформи. Загальні характеристики платформи представлені в таблиці.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

### 2.3.2 Wemos D1 Mini

Wemos D1 mini [17] — це мікроконтролер ESP-8266 із можливостями Wi-Fi. Це плата, яка має вбудований міст USB-UART (CH340) і роз'єм micro-USB у поєднанні з регулятором потужності 3,3 Вольта для живлення.

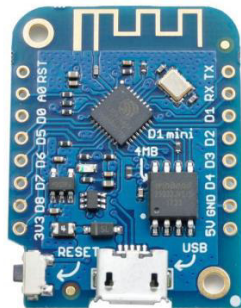


Рисунок 2.9 - Wemos D1 Mini [17]

Цей блок ESP8266 утворює функціональну систему розвитку, не потребуючи додаткових компонентів, що робить його більш економічним та енергоефективним, оскільки він споживає менше енергії. Крім того, він сумісний з Arduino на програмному рівні (підтримка Arduino IDE), забезпечує можливість створення програм на MicroPython, а також використання прошивки NodeMCU (Lua scripts).

### 2.3.3 Датчик SCT-013-030

Датчиком струму, який використовується в даній роботі, є трансформатор струму SCT-013-030 [18] фірми УНДС.



Рисунок 2.10 - SCT-013-030

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Це неінвазивний датчик струму, який можна закріпити навколо лінії живлення електричного навантаження та зчитувати силу струму, що проходить через нього. Його сердцевина розділена, і таким чином датчик реагує на магнітне поле навколо навколишнього провідника струму. Такий тип датчиків ідеально підходить для ідентичних застосувань.

Як і будь-який трансформатор, він має первинну обмотку, феритовий сердечник і вторинну обмотку. Первинна котушка буде фазою або нейтраллю дроту, що проходить через порожнечу датчика. Слід зазначити, що нейтраль і фаза не повинні проходити через датчик разом, оскільки в цьому випадку ми мали б витік однакових струмів, але в різних напрямках, тобто два рівних і протилежних магнітних поля, які гасили б одне одного і, отже, існували б не буде виводу на другу котушку. На кожні 30 Ампер в первинній котушці він індукуює 1 Вольт у вторинній, яка складається з 1800 витків. Вихідна напруга, яку він дає нам, становить 0-1 Вольт з відповідними значеннями струму 0-30 Ампер, що протікає через первинну обмотку. Цей конкретний датчик додатково містить резистор 62 Ом.

Вихідний сигнал трансформатора є майже синусоїдальною формою хвилі, яка має зміну -1 +1 Вольт, що відповідає максимальному вимірюваному значенню датчика (30 Ампер). Оскільки всі АЦП вимірюють лише позитивні напруги, вихідна напруга повинна бути зміщена в позитивну сторону, щоб усунути негативні значення. Для цього на виході трансформатора реалізована така схема (див. рис. 2.11):

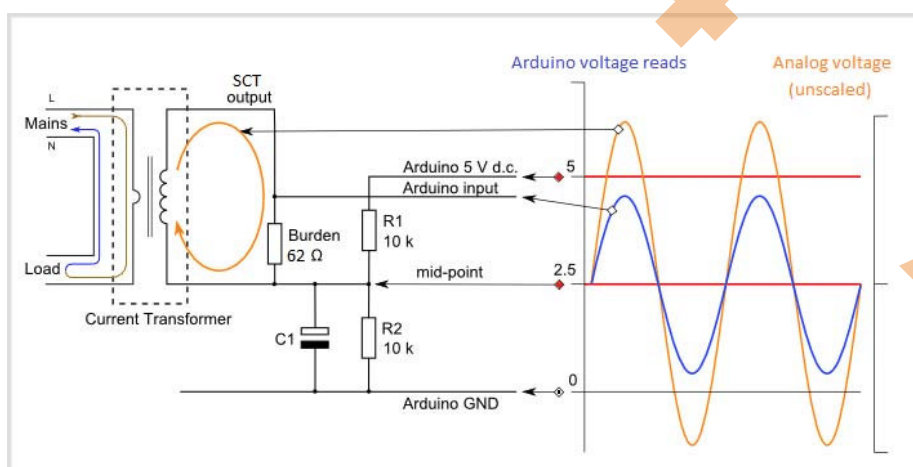


Рисунок 2.11 - Схема резистора навантаження та дільника напруги

Вим.	Арк.	№ докум.	Підпис	Дата

Резистори R1, R2 утворюють дільник напруги та додають зміщення, створюючи компонент постійного струму 2,5 В, тоді як конденсатор С1 забезпечує низький імпеданс шляху до землі для сигналу змінного струму (тут обрано 10 мкФ). Таким чином, щоб гарантувати, що сигнал напруги трансформатора буде вимірний перетворювачем АЦП, резистори R1, R2 вибираються з орієнтовним значенням 10 кОм, як показано на схемі. Резистори повинні мати відносно великі номінали, щоб мати якомога менший витік струму в ланцюзі, особливо у випадках, коли наші системи живляться від акумуляторних електричних акумуляторів.

### 2.3.4 Модульна плата каналу реле 5 В-1

Релейний модуль, який використовується в системі для організації керування, є релейною моделлю 5V-1 Channel. Це пристрій, який містить реле та деякі активні елементи, завдяки яким ним можна керувати за допомогою мікроконтролера. Він також містить два фотодіоди для індикаторів роботи та може керувати електричними пристроями потужністю до 2,2 кВт (рис.2.12). Загальні характеристики реле: максимальне навантаження - АС 250V / 10А, DC 30V / 10А; робоча напруга - 5 В; робочий струм - 15-20 мА; контрольний струм -  $\geq 4$  мА.



Рисунок 2.12 - Реле 5V-1 Channel

Щоб легко з'єднати лічильник SCT-013 100А з рештою схеми, було вибрано використання роз'єму TRRS діаметром 3,5 мм, показаного на малюнку 19, щоб скористатися перевагами кінцевої частини кабелю лічильника, що дозволяє уникнути будь-яких перешкод із кабелем, але також дає можливість легкої заміни

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

лічильника, не вимагаючи тривалої процедури втручання в схему. Гніздо TRRS є формою інтерфейсу з трьома провідниками та заземленням. У деяких пристроях заземлення використовується для інших функцій, наприклад для мікрофона в навушниках (рис. 2.13).

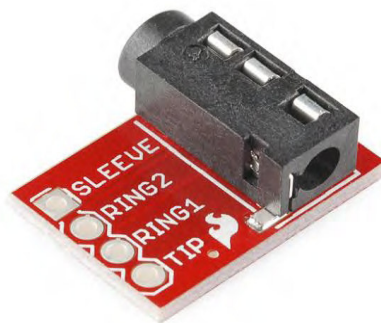


Рисунок 2.13 - Роз'єм TRRS діаметром 3,5 мм.

Області, на які поділена така заглушка, це наконечник, кільце 1, кільце 2 та гільза, дивлячись від кінчика пробки до її основи. У випадку інтерфейсу лічильника двох крайніх зон (наконечника та рукава) достатньо для повної функціональності системи.

У проекті ми використовуємо електrolітичний конденсатор ємністю 10 мкФ із полярністю, максимальною робочою напругою 16 вольт і максимальною робочою температурою 105 °С, який діє як шумовий фільтр і є шляхом для проходження струму в обхід опору. Форма такого конденсатора показана на малюнку 2.14, і видно, що на його оболонці з'являються різні позначки, які є його ємністю та робочою напругою. Також на конденсаторі є позначки його полярності, які також можна зрозуміти по довжині двох його виводів. Вивід із найбільшою довжиною є позитивним, а інший – негативним.



Рисунок 2.14 - Типовий електrolітичний конденсатор

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Щоб підключити датчик струму до Arduino, окрім конденсатора, також потрібні деякі резистори. Резистори — це електронні елементи з певним і незмінним значенням електричного опору, які вони застосовують як елемент у ланцюзі, і їх основною характеристикою є обмеження потоку електронів. Вони є пасивними компонентами, що означає, що вони споживають лише електроенергію, і зазвичай зустрічаються в схемах з активними компонентами, такими як мікроконтролер. У схемах вони зазвичай знаходяться у двох з'єднаннях, де кожне пропонує щось своє для потреб схеми. Перше з'єднання – це послідовне з'єднання резисторів, що допомагає створити дільник напруги, а друге – це паралельне з'єднання резисторів, яке, по суті, створює дільник струму. Характерним розміром резисторів є електричний опір, який позначається англійською літерою «R» і вимірюється в Омах (позначається грецькою літерою «Ω»). Простий приклад інтерпретації електричного опору полягає в тому, що якщо між двома клемами з різницею потенціалів 1 В існує опір 1 Ом, то через них протікає струм 1 А. Для проекту відповідним типом резисторів є наскрізні резистори, які підходять для мікроелектронних схем. Такий резистор має два висновки для прямого підключення, він не має полярності, на відміну від електролітичного конденсатора, а на його оболонці є різнокольорові смужки, які свідчать про величину його електричного опору та відхилення фактичного значення від зображених.

Color	Value	Multiplier	Tolerance
Black	0	$\times 10^0$	$\pm 20\%$
Brown	1	$\times 10^1$	$\pm 1\%$
Red	2	$\times 10^2$	$\pm 2\%$
Orange	3	$\times 10^3$	$\pm 3\%$
Yellow	4	$\times 10^4$	- 0, + 100%
Green	5	$\times 10^5$	$\pm 0.5\%$
Blue	6	$\times 10^6$	$\pm 0.25\%$
Violet	7	$\times 10^7$	$\pm 0.10\%$
Gray	8	$\times 10^8$	$\pm 0.05\%$
White	9	$\times 10^9$	$\pm 10\%$
Gold	–	$\times 10^{-1}$	$\pm 5\%$
Silver	–	$\times 10^{-2}$	$\pm 10\%$

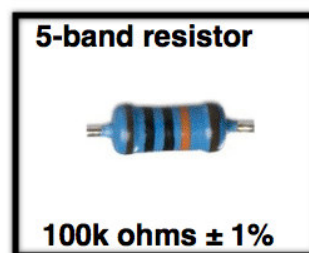
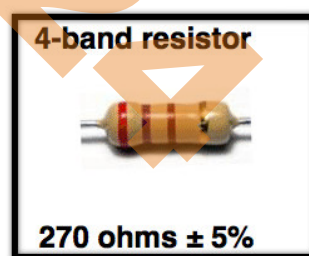


Рисунок 2.15 - Кольоровий список резисторів і розрахунок резисторів.

Вим.	Арк.	№ докум.	Підпис	Дата

На малюнку 2.15 ліворуч показано таблицю для розрахунку номінального електричного опору та відхилення резистора, а праворуч ми бачимо два приклади резисторів з різним кодом кольору. У 4-смугових резисторах рядки «зчитуються» зліва направо, причому перші два беруть значення зі стовпця «Значення», створюючи двозначне число, третє — це ступінь числа 10, на який це число множиться, і четвертий рядок показує відхилення значення опору. Наприклад, для верхнього та правого резистора на малюнку 2.15 перші два рядки (де червоний = 2 і фіолетовий = 7) дають число 27, яке потім множиться на потужність третього рядка (коричневий =  $10^1$ ), щоб отримати, що питомий опір 270 Ом з відхиленням 5%, так як четверта лінія - золото. У категорії резисторів із 5-діпазонним кодом (5-діпазонні резистори) значення резистора отримується подібним чином, з тією різницею, що число, яке виходить зі стовпця Значення, витягується з перших трьох рядків, четвертий — це множник, а п'ятий — відхилення.

## 2.4 Синтез структурної та принципової схеми

Для проектування пристрою, потрібно окреслити, його основні функціональні блок згідно ТЗ і на основі цього побудувати структурну схему. На рисунку 2.14. представлена структурна схема.

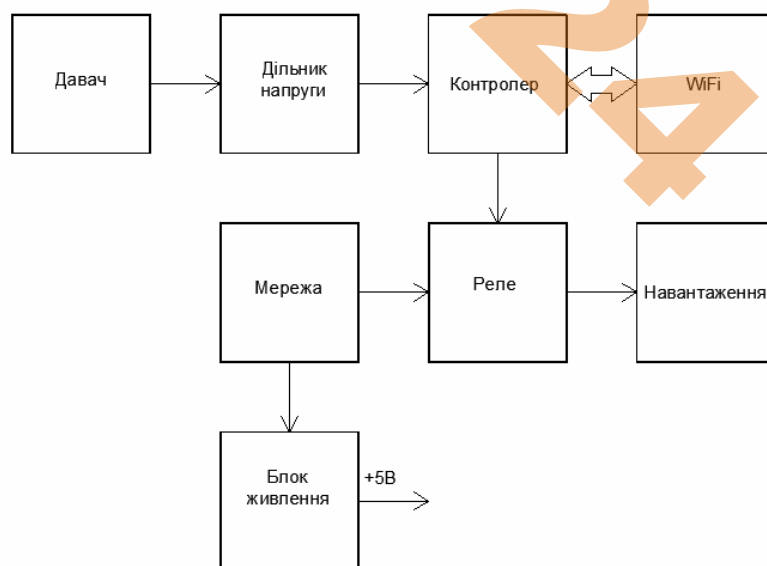


Рисунок 2.16 - Структурна схема автоматизованого пристрою

Для виконання поставленої задачі, оперуючись на підібрані компоненти, спроектуємо принципову схему. На рисунку 2.15. представлена принципова схема автоматизованого пристрою.

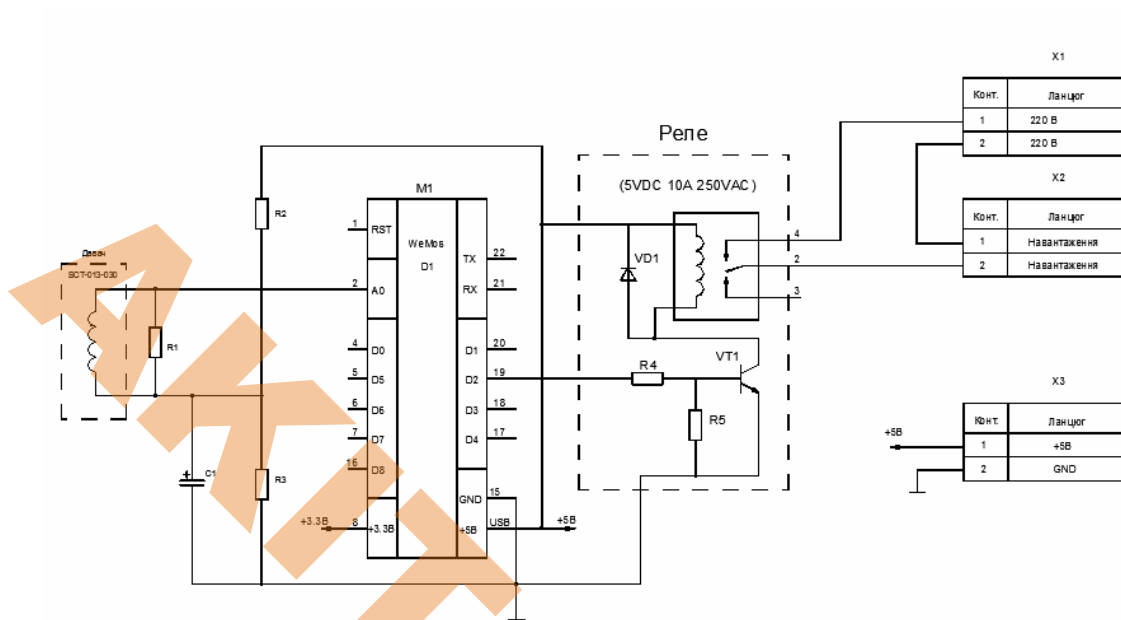


Рисунок 2.17 - Принципова схема автоматизованого пристрою

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

### 3 АЛГОРИТМ РОБОТИ ТА ПРОГРАМУВАННЯ КОНТРОЛЕРА

Зв'язок кінцевої системи базується на протоколі MQTT і архітектурі публікації-підписки. Як було проаналізовано у 2-му розділі, посередницька організація, посередницький сервер, Брокер, відповідає за розповсюдження повідомлень зацікавленим підписникам (клієнтам). Для використання протоколу необхідна установка сервера. У нашому випадку був обраний Mosquitto Broker.

#### Від мікроконтролера до брокера і навпаки

Усі функції нашої системи, такі як вимірювання, отримані від трансформатора струму, мають бути передані брокеру MQTT, який відповідає за збереження значень у темі та оновлення цих полів через публікацію/підписку.

Спочатку нам потрібно включити бібліотеку PubSubClient.h у wemos і оголосити деякі основні параметри, щоб увімкнути її. Далі ми створюємо функцію, за допомогою якої наша система підключається до брокера. На першому етапі ми створюємо клієнт mqtt, оголошуємо брокера, з яким він буде підключатися. Далі ми повинні оголосити нашу реєстрацію, яка приймає три аргументи, перший — це клієнт, який ми створили, другий — шлях, де ми будемо реєструватися, і третій — QoS.

Функція налаштування void містить ініціалізацію значень і встановлення підключення нашого пристрою до мережі.

У функції void loop() реалізований фрагмент коду для виклику функцій, відповідальних за підключення до Брокера, відправку значень Брокеру, а також ідентифікацію записів.

Відразу після того, як ми створюємо функцію void callback(), яка відповідає за процес планування ввімкнення/вимкнення пристроїв, функцію calcWatt для оновлення посередника поточними записами про споживання пристроїв, а також функцію checkbound, яка перевіряє варіацію між двома послідовними вимірюваннями. При невеликих коливаннях потужності (зазначених змінною dif\_watts) значення не надсилаються Брокеру, щоб уникнути сценарію перевантаження мережі.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Нарешті, ми використовуємо ще одну функцію `void MQTT_reconnect()`, яка відповідає за підтримку з'єднання з Посередником відкритим і викликається, якщо з'єднання з Посередником втрачено, доки воно не буде відновлено.

Як приклад наведено код зворотного виклику та `getWatts()`.

```
void callback(char* topic, byte* payload, unsigned int length)
```

```
{
```

```
Serial.print("Command from MQTT broker is : [");
```

```
Serial.print(topic);
```

```
Serial.print(", ");
```

```
payload[length] = '\0';
```

```
String message = (char*)payload;
```

```
if (message == "0")
```

```
{
```

```
digitalWrite(CTRL_PIN, LOW);
```

```
ACT_CTRL_PIN = false;
```

```
Serial.println(" Turn Off CONTROLLER! ");
```

```
}
```

```
if (message == "1")
```

```
{
```

```
digitalWrite(CTRL_PIN, HIGH);
```

```
ACT_CTRL_PIN = true;
```

```
Serial.println(" Turn On CONTROLLER! ");
```

```
}
```

```
Serial.println();
```

```
}
```

Змінні, пов'язані з методом `getWatts()`.

```
float watts = 0.0;
```

```
float dif_watts = 4.0;
```

```
unsigned long previousMillis = 0;
```

```
const long delayMillis = 15000; // 15 сек
```

```
char data[80];
```

Вим.	Арк.	№ докум.	Підпис	Дата

```

void calcWatt ()
{
// float or double watts; client.publish(topic, String(watts).c_str(), true);
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= delayMillis)
{
previousMillis = currentMillis;
float Irms = emon1.calcIrms(1480);
float newWatts = Irms * 220;
if (checkBound(newWatts, watts, dif_watts))
{
watts = newWatts;
snprintf(data, 80, "%ld", (int)watts);
Serial.print("New Power:");
Serial.println(String((int)watts).c_str());
client.publish(pubTopic, data, true);
}
}
}

bool checkBound(float newValue, float prevValue, float maxDiff)
{
return !isnan(newValue) && (newValue < prevValue - maxDiff || newValue >
prevValue + maxDiff);
}

```

### 3.1 Архітектура

Бездротова форма зв'язку між розетками (контролерами) і системним додатком у поєднанні з реалізацією протоколу MQTT є фундаментальними елементами остаточної архітектури платформи.

На наведеному нижче ескізі показано серверний комп'ютер системи, на якому запущено веб-сервер для роботи веб-додатку. Він також запускає брокера MQTT.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

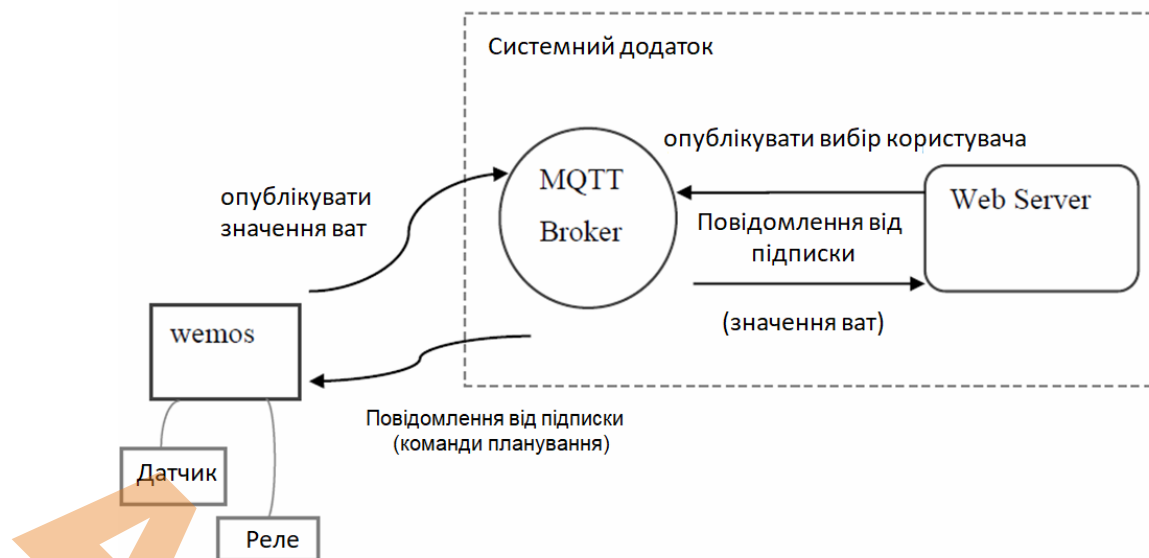


Рисунок 3.1 - Остаточна архітектура системи

Протокол визначає, як клієнти (мікроконтролера та веб-сервера) повинні реєструватися в темі (темі) брокера MQTT, щоб існував двосторонній обмін даними для запису вимірювань, а також для керування мікроконтролерів. Назва цього суб'єкта також повинна поєднувати серійний номер кожного домашнього контролера для унікальності його зв'язку з Брокером.

Веб-сервер записує на шлях, який ми попередньо визначили для поточних значень датчиків від клієнта (мікроконтролера wemos). Крім того, він публікує (publish) параметри користувача для управління електропостачанням (скрипт планування).

Зліва ми бачимо, що мікроселектор wemos підписується на отримання команд планування користувача, а також виконує процес публікації для сценарію надсилання значень вимірювань енергоспоживання.

### 3.2 Організація бази даних

Структура сутності та код спочатку. База даних будь-якої системи є її ядром. У цій програмі для організації, реалізації та доступу до даних бази даних використовувався Entity Framework від Microsoft. Також було дотримано методології підходу Code First, де не було необхідності створювати базу даних, на

Вим.	Арк.	№ докум.	Підпис	Дата

якій буде пов'язано додаток, замість цього базу даних створювали за допомогою вищезазначеного підходу з коду.

Перший підхід коду. Міграції. Як уже згадувалося, використовуючи техніку першого коду, кожного разу, коли ми змінюємо наші базові класи, які складають нашу програму, схема, або іншим чином структура нашої бази, також змінюється. Папка міграцій містить усі різні стани, через які пройде наша база.

Програма використовує попередньо визначену бібліотеку Microsoft для автентифікації користувачів та інших різноманітних функцій, пов'язаних із ідентифікацією користувачів ASP.NET. З цією бібліотекою є готові контролери, представлення та моделі, які реалізують пов'язані з користувачем функції. Під час створення 1-ї міграції, хоча інші класи не були визначені, у базі даних створюються таблиці, такі як:

- AspNetRoles
- AspNetUserClaims
- AspNetUserLogins
- AspNetUserRoles
- AspNetUsers

Команда, яку ми вводимо в консолі менеджера пакунків, щоб додати міграцію: `add-migration ExampleMigrationName`

Результатом команди є створення класу `csharp`, який визначає, створює вищезазначені таблиці.

Кожна міграція складається з методів `Up` і `Down`. По суті, ці два методи містять абсолютно протилежні команди. У методі `Up` створюються таблиці SQL, які оголошуються автоматично, як ми згадували вище, через `Identity`, тоді як у методі `down` вони видаляються.

### Оновлення Бази

Наступним кроком після кожної міграції є оновлення бази даних і фактичне виконання команд, згенерованих міграцією. Команда має такий вигляд: `Update-database`

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Під час виконання цієї команди створюється файл .mdf, який, по суті, становить базу, і папку \_MigrationsHistory, яка містить останню міграцію, тобто поточний стан бази в будь-який час.

Після того, як усі класи сутностей, пов'язані з програмою, будуть скомпільовані, буде виконано створення остаточної міграції, і, таким чином, база даних буде оновлено. Остаточна версія бази даних організована в таблиці, пов'язані з записом значень лічильників споживання, частиною планування та рештою з керуванням користувачами (як зазначено вище). Вони не є незалежними, а радше пов'язані відносинами, вираженими в реляційних моделях.

АКІТ  
2024

АКІТ

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

## ВИСНОВКИ

Автоматизовані пристрої збору параметрів електромережі значно підвищують ефективність управління електричними мережами. Вони дозволяють реєструвати і аналізувати основні параметри, такі як напруга, струм, потужність, що сприяє швидкому виявленню та вирішенню проблем, зменшенню втрат і оптимізації розподілу енергії.

Можливість інтеграції цих пристроїв у інфраструктуру Smart Grid є ключовою перевагою. Smart Grid технології дозволяють автоматизувати процеси моніторингу та управління, забезпечуючи більш стабільне і надійне постачання електроенергії, а також створюючи передумови для гнучкого реагування на зміни у попиті та пропозиції енергії.

Автоматизація збору параметрів забезпечує високу точність і оперативність отримання даних. Це дозволяє енергетичним компаніям краще розуміти стан мережі в режимі реального часу, зменшуючи час реакції на аварійні ситуації та покращуючи планування технічного обслуговування.

Завдяки постійному моніторингу та аналізу даних, автоматизовані системи сприяють покращенню безпеки електромережі, запобігаючи перегріву, перевантаженням та іншим аварійним ситуаціям. Це, в свою чергу, підвищує надійність постачання електроенергії кінцевим споживачам.

Впровадження автоматизованих пристроїв збору параметрів електромережі з можливістю інтеграції у Smart Grid є важливим кроком на шляху до модернізації енергетичної інфраструктури. Це не тільки підвищує ефективність, надійність і безпеку електромереж, але й сприяє розвитку стійких і екологічно чистих енергетичних систем, забезпечуючи стабільне і ефективне постачання електроенергії в умовах зростаючого попиту.

У результаті виконання бакалаврської роботи було проведено аналіз систем розумний лічильник. Підібрано компоненти для нашої системи. Розроблено структурну та принципову схему, написано код для мікроконтролера.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

### Список використаної літератури

1. IOT Based Smart Grid Monitoring Using Arduino Controller// P.Shanmukha.V.Sai1,Y.Gopichand2,K.Manohar3,V.Nagamani /International Journal of Research in Advent Technology, Special Issue, March 2019. E-ISSN: 2321-9637. 517-522 P
2. Smart Load Node for Non–Smart Load under Smart Grid Paradigm: A New Home Energy Management System// Shashank Singh, Student Member, IEEE, Amit Roy, and Selvan M.P., Member, IEEE/ Published: iee consumer electronics magazine. Indian patent application, APP NO: 201741039083, 2017-NOV-2 1
3. IoT based Smart Power Grid Monitoring and Control using Arduino// Anil Jaybhaye, Ashutosh Sonawane / International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 04 | Apr 2020
4. Gravity I2C [Електронний ресурс]. – Режим доступу: <https://grobotronics.com/gravity-i2c-digital-wattmeter.html> Дата звернення 01.03.2024
5. Системи моніторингу якості електричної енергії SMART GRID: метрологічний компас [Електронний ресурс]. – Режим доступу: <https://www.evm.ua/novosti/sistemy-monitoryngu-yakosyi-ee-smart-grid>. Дата звернення 01.03.2024
6. Towards a Definition of the Internet of Things (IoT).// Minerva, R., Biru, A. and Rotondi, D. / (2015) IEEE Internet Initiative, 1, 1-86.
7. C. Pereira та A. Aguiar, «Towards Efficient Mobile M2M Communications: Survey and Open Challenges», Sensors, vol. 14, стор. 19582-19608, 2014.
8. C. Saad, B. Mostafa, E. Ahmadi and H. Abderrahmane, "Comparative analysis of the productivity of wireless communication protocols for intelligent sensors and their applications", International journal of advanced computer science and applications (IJACSA), vol. 5, № 4, p. 76-85, 2014.
9. MQTT Version 3.1.1 - OASIS Standard [Електронний ресурс]. – Режим доступу: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf>. Дата звернення 01.03.2024.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

10. MQTT i CoAP, IoT Protocols. [Електронний ресурс]. – Режим доступу: [https://www.eclipse.org/community/eclipse\\_newsletter/2014/february/article2.php](https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php) Дата звернення 01.03.2024.

11. Selection of current transformers & wire sizing in substations [Електронний ресурс]. – Режим доступу: <https://guggenmosales.com/documents/SelectionOfCurrentTransformersWireSizingInSubstations.pdf>. Дата звернення 01.03.2024.

12. Characteristics of isolated current and voltage converters – applications. [Електронний ресурс]. – Режим доступу: <https://www.lem.com/en/file/5630/download>. Дата звернення 01.03.2024.

13. "Power in alternating current circuits". [Електронний ресурс]. – Режим доступу: <https://www.electronics-tutorials.ws/accircuits/power-in-ac-circuits.html>. Дата звернення 01.03.2024.

14. Вікіпедія «Arduino», вільна енциклопедія. [Електронний ресурс]. – Режим доступу: <https://en.wikipedia.org/wiki/Arduino> Дата звернення 01.03.2024.

15. Getting Started with Arduino, Second Edition [Електронний ресурс]. – Режим доступу: <https://docs.arduino.cc/retired/other/getting-started-with-arduino/> Дата звернення 01.03.2024.

16. Arduino Uno Rev 3. [Електронний ресурс]. – Режим доступу: <https://store.arduino.cc/arduino-uno-rev3>. Дата звернення 01.03.2024.

17. D1 mini [Wemos Electronics] [Електронний ресурс]. – Режим доступу: [https://geekmatic.in.ua/ua/nodemcu\\_esp8266\\_wemos](https://geekmatic.in.ua/ua/nodemcu_esp8266_wemos). Дата звернення 01.03.2024.

18. SCT-013 Datasheet [Електронний ресурс]. – Режим доступу: [https://www.mcielectronics.cl/website\\_MCI/static/documents/Datasheet\\_SCT013.pdf](https://www.mcielectronics.cl/website_MCI/static/documents/Datasheet_SCT013.pdf) Дата звернення 01.03.2024.

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

# ДОДАТКИ

АКТИВ 2024

АКТИВ

Вим.	Арк.	№ докум.	Підпис	Дата

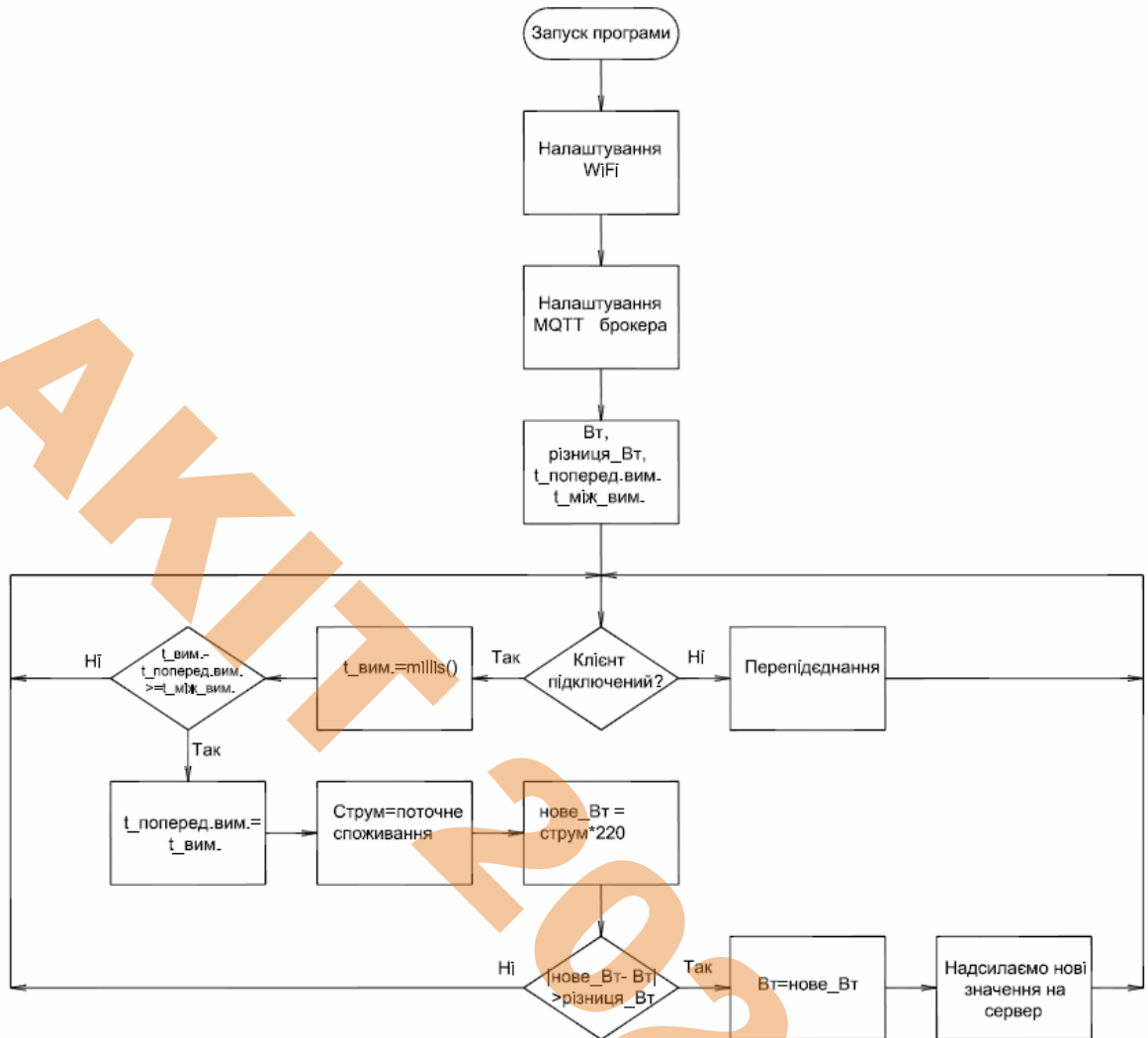
КБР.АКІТ.20050051.01.000 ПЗ

Арк.

58

## Додаток А

### Алгоритм



Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

## Додаток Б

### Лістинг програми

```
Клієнт на C++
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <EmonLib.h> // Підключення бібліотеки для вимірювання
споживаної потужності (приклад бібліотеки)

// Налаштування WiFi
const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";

// Налаштування MQTT брокера
const char* mqtt_server = "mqtt_server_address";
const int mqtt_port = 1883; // Порт MQTT брокера
const char* mqtt_user = "mqtt_username";
const char* mqtt_password = "mqtt_password";

// Налаштування PIN для керування контролером
const int CTRL_PIN = D1; // Приклад PIN для керування контролером

// Об'єкт для вимірювання споживаної потужності (приклад використання
бібліотеки)
EnergyMonitor emon1;

// Оголошення об'єкту клієнта MQTT
WiFiClient espClient;
PubSubClient client(espClient);

// Глобальні змінні для вимірювання споживаної потужності
```

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

```

float watts = 0.0;
float dif_watts = 4.0;
unsigned long previousMillis = 0;
const long delayMillis = 15000; // Інтервал вимірювання споживаної
потужності (15 секунд)
char data[80];

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Command from MQTT broker is : [");
  Serial.print(topic);
  Serial.print("], ");
  payload[length] = '\0';
  String message = (char*)payload;

  if (message == "0") {

```

Вим.	Арк.	№ докум.	Підпис	Дата

```

digitalWrite(CTRL_PIN, LOW);
Serial.println("Turn Off CONTROLLER!");
}
else if (message == "1") {
    digitalWrite(CTRL_PIN, HIGH);
    Serial.println("Turn On CONTROLLER!");
}
else {
    Serial.println("Unknown command received.");
}
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
            Serial.println("connected");
            client.subscribe("controller/command");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

void setup() {
    pinMode(CTRL_PIN, OUTPUT);
    Serial.begin(115200);
    setup_wifi();
}

```

Вим.	Арк.	№ докум.	Підпис	Дата

```

client.setServer(mqtt_server, mqtt_port);
client.setCallback(callback);
emon1.current(0, 111.1); // Приклад налаштування об'єкта вимірювання
(потрібно відповідно до вашого датчика)
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  calcWatt();
}

void calcWatt() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= delayMillis) {
    previousMillis = currentMillis;
    float Irms = emon1.calcIrms(1480); // Розрахунок поточного споживання
    float newWatts = Irms * 220; // Приклад розрахунку потужності (залежить
від вашої конфігурації)
    if (checkBound(newWatts, watts, dif_watts)) {
      watts = newWatts;
      snprintf(data, 80, "%ld", (int)watts);
      Serial.print("New Power:");
      Serial.println(String((int)watts).c_str());
      client.publish("controller/power", data, true);
    }
  }
}
}

```

Вим.	Арк.	№ докум.	Підпис	Дата

```
bool checkBound(float newValue, float prevValue, float maxDiff) {  
    return !isnan(newValue) && (newValue < prevValue - maxDiff || newValue >  
prevValue + maxDiff);  
}
```

Сервер на NodeJS

```
const mqtt = require('mqtt');  
const mysql = require('mysql');  
  
// Параметри підключення до MQTT брокера  
const mqtt_host = 'mqtt://localhost'; // Адреса MQTT брокера  
const mqtt_port = 1883; // Порт MQTT брокера  
const mqtt_topic = 'controller/#'; // Тема для підписки  
  
// Параметри підключення до бази даних MySQL  
const mysql_host = 'localhost';  
const mysql_user = 'your_mysql_username';  
const mysql_password = 'your_mysql_password';  
const mysql_database = 'your_mysql_database';  
  
// Підключення до MQTT брокера  
const mqttClient = mqtt.connect(mqtt_host + ':' + mqtt_port);  
  
// Підключення до бази даних MySQL  
const dbConnection = mysql.createConnection({  
    host: mysql_host,  
    user: mysql_user,  
    password: mysql_password,  
    database: mysql_database
```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР.АКІТ.20050051.01.000 ПЗ

Арк.

64

```

});

dbConnection.connect((err) => {
  if (err) {
    console.error('Error connecting to MySQL database: ' + err.stack);
    return;
  }
  console.log('Connected to MySQL database as id ' + dbConnection.threadId);
});

mqttClient.on('connect', () => {
  console.log('Server connected to MQTT broker');
  mqttClient.subscribe(mqtt_topic);
});

mqttClient.on('message', (topic, message) => {
  // Отримання повідомлення від MQTT брокера
  console.log('Message received:', topic, message.toString());

  // Збереження отриманого повідомлення в базі даних
  const sql = 'INSERT INTO mqtt_data (topic, message) VALUES (?, ?)';
  const values = [topic, message.toString()];

  dbConnection.query(sql, values, (err, result) => {
    if (err) {
      console.error('Error inserting data into MySQL:', err);
      return;
    }
    console.log('Data inserted into MySQL successfully');
  });
});

```

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

```
mqttClient.on('error', (error) => {
  console.error('Error occurred:', error);
});

mqttClient.on('close', () => {
  console.log('Disconnected from MQTT broker');
});
```

Структура БД

```
CREATE TABLE mqtt_data (
  id INT AUTO_INCREMENT PRIMARY KEY,
  topic VARCHAR(255) NOT NULL,
  message TEXT NOT NULL,
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

					КБР.АКІТ.20050051.01.000 ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66