

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

Факультет електроніки  
(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації  
(повна назва кафедри)

«На правах рукопису»  
УДК 004.738.5.057

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Г.Г.Власюк  
(підпис) (ініціали, прізвище)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2019 р.

## Магістерська дисертація

зі спеціальності 171 Електроніка  
(код і назва спеціальності)

на тему: Система моніторингу стану помешкань за технологією IoT

Виконав: студент II курсу, групи ДВ-81мп  
(шифр групи)

\_\_\_\_\_ Парадзінський Олег Олегович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент каф. ЗТРІ, к. т. н., доц. Швайченко В.Б.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**Національний технічний університет України  
«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Інститут (факультет) \_\_\_\_\_ Факультет електроніки \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ звукотехніки та реєстрації інформації \_\_\_\_\_  
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (освітня програма) 171 Електроніка (Електронні системи мультимедіа та засоби Інтернету речей) \_\_\_\_\_  
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Г.Г.Власюк \_\_\_\_\_  
(підпис) (ініціали, прізвище)

«\_\_» \_\_\_\_\_ 2018 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

Парадзінському Олегу Олеговичу  
(прізвище, ім'я, по батькові)

1. Тема дисертації Система моніторингу стану помешкань за технологією IoT, \_\_\_\_\_  
науковий керівник дисертації Швайченко Володимир Борисович, доц. \_\_\_\_\_  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)  
затверджені наказом по університету від «07» листопада 2019 р. № 3859-с
2. Строк подання студентом дисертації 9.12.2019 р.
3. Об'єкт дослідження: Процеси за технологією Інтернету речей
4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою): методи та технології виміру параметрів електричної мережі, підходи до створення додатків інтернету речей, протоколи передачі даних.

5. Перелік завдань, які потрібно розробити: 1) Проаналізувати існуючі методи виміру параметрів електричної мережі, дослідження протоколу передачі даних, провести огляд існуючих рішень. 2) Проаналізувати програмні та апаратні засоби створення систем інтернету речей, запропонувати рішення, що дозволяють підвищити якісні характеристики моніторингу та керування IoT системою. 3) Розробити та дослідити IoT систему моніторингу та керування «розумним будинком», що може бути використана для навчальних потреб
6. Перелік графічного (ілюстративного) матеріалу: 1) 97 рис, 21 табл., 1 презентація, 10 слайдів.
7. Орієнтовний перелік публікацій: 1) Аналіз обчислювальних мереж для застосування в IoT системах// XXXVI міжнародна науково-практична інтернет-конференція «Інновації науки XXI», 2019 р., -С. 18-20 2) Система керування кліматом у теплицях// XX міжнародна науково-практична інтернет-конференція «Інновації XXI століття», 2018 р.,- С.65-68
8. Дата видачі завдання 10. 09. 2018 р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Написання першого розділу: Огляд стану предметної області	15.12.2018	
2	Дослідження існуючих методів виміру характеристик мережі. Дослідження існуючих платформ. Порівняльний аналіз протоколів передачі даних	30.05.2019	
3	Написання третього розділу: Проектування архітектури системи. Реалізація прототипу та тестування	10.10.2019	
4	Написання четвертого розділу: Дослідження безмаркерної системи тривимірного захоплення руху людини з використанням кількох видів з камер	09.11.2019	
5	Підготовка матеріалів до друку та оформлення пояснювальної записки	30.11.2019	
6	Підготовка та оформлення плакатів для доповіді	03.12.2019	

Студент

\_\_\_\_\_ (підпис)

О.О. Парадзінський

\_\_\_\_\_ (ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_ (підпис)

В.Б. Швайченко

\_\_\_\_\_ (ініціали, прізвище)

## **РЕФЕРАТ**

Магістерська дисертація: 97 с., 21 рис., 11 табл., 18 джерел, 2 додатка.

**ІНТЕРНЕТ РЕЧЕЙ, ПРОТОКОЛ ПРИКЛАДНОГО РІВНЯ,  
ПЛАТФОРМА ІНТЕРНЕТУ РЕЧЕЙ, ХМАРНА СИСТЕМА.**

### **Актуальність теми**

Область використання розумних будинків обмежена локальними мережами для взаємодії з клієнтськими пристроями. В останні роки набувають все більшої популярності альтернативні системи, спроектовані для інтернету речей та пристроїв на їх базі, чим обумовлена необхідність розширення систем розумних будинків для додавання підтримки даних протоколів і пристроїв.

### **Мета та задачі дослідження**

Аналіз підходів до розширення області використання платформи інтернету речей з подальшим забезпеченням можливості використання існуючих рішень, які взаємодіють, використовуючи існуючі протоколи передачі даних. Головним завданням є дослідження, що полягає у проектуванні системи для забезпечення взаємодії функціоналу реалізованого на базі МК, з користувачем через існуючі протоколи передачі даних.

### **Вирішення поставлених завдань та досягнуті результати**

В результаті виконання роботи була реалізована система, що керує та аналізує характеристики електричної мережі. Дана реалізація також включає можливість встановлення правил, за якими відбувається автоматичне керування системою. Для керування електричної мережі було розроблено додаток з графічним інтерфейсом адміністратора. Система проста у розгортанні та може бути використана як доповнення до існуючої системи.

### **Об'єкт дослідження**

Процеси технології Інтернету речей

### **Предмет дослідження**

Система моніторингу та керування, яка виконує функцію керування та аналізу мережі помешкань.

### **Методи дослідження**

Для вирішення проблеми в даній роботі використовуються методи аналізу, синтезу, системного аналізу, порівняння та логічного узагальнення результатів.

### **Наукова новизна**

Наукова новизна роботи полягає у аналізі та удосконаленні методів, що дозволяють керувати електричною мережею, виконувати аналіз мережі, та обирати автоматизовані сценарії роботи.

### **Практичне значення одержаних результатів**

Розроблена система може бути розгорнута на кафедральній інфраструктурі. Вона включає в собі зручний користувацький інтерфейс для встановлення конфігурацій та може бути використана для інтеграції в існуючі системи з підтримкою обраного протоколу.

Апробації результатів дисертації

### **Публікації**

Аналіз обчислювальних мереж для застосування в IoT системах. XXXVI міжнародна науково-практична інтернет-конференція «Інновації науки XXI», 2019 р.

Система керування кліматом у теплицях. XX міжнародна науково-практична інтернет-конференція «Інновації XXI століття», 2018 р.

## **SUMMARY**

Master's dissertation: 97 p., 21 pic., 11 tabl., 18 sources, 2 supplement.

THE INTERNET OF THINGS, THE APPLICATION LEVEL PROTOCOL,  
THE THINGS 'INTERNET PLATFORM, THE CLOUD SYSTEM.

### **Actuality of work**

The scope of smart homes is limited by local area networks to interact with client devices. In recent years, alternative systems designed for the Internet of Things and devices based on them have become increasingly popular, which necessitates the expansion of smart home systems to add support for these protocols and devices.

### **The purpose and objectives of the study**

An analysis of approaches to expanding the scope of the Internet of Things platform, further enabling existing interoperable solutions to be used using existing data communication protocols. The main task is to design a system to ensure that the functionality implemented on the basis of MK, with the user through existing data transmission protocols.

### **Problem solving and results achieved**

As a result of the work, a system was implemented that manages and analyzes the characteristics of the electrical network. This implementation also includes the ability to set rules that automatically control the system. An administrative GUI application was developed to control the power grid. The system is easy to deploy and can be used to complement an existing system.

### **Object of study**

Processes of Internet of Things technology

### **Subject of study**

A monitoring and control system that performs the function of managing and analyzing a network of premises.

### **Research methods**

Methods of analysis, synthesis, system analysis, comparison and logical generalization of results are used to solve the problem.

### **Scientific novelty**

The scientific novelty of the work is to analyze and improve methods that

allow you to manage the electrical network, perform network analysis, and choose automated scenarios.

### **The practical significance of the results obtained**

The developed system can be deployed on the cathedral infrastructure. The tub includes a user-friendly interface for configuring and can be used to integrate into existing systems with support for the selected protocol.

## ЗМІСТ

ВСТУП .....	8
1. АНАЛІЗ ТЕХНОЛОГІЇ ІНТЕРНЕТУ РЕЧЕЙ.....	12
1.1 Базові поняття Інтернету речей.....	12
1.2. Засоби взаємодії пристроїв на прикладному рівні .....	14
1.2.1 RESTFUL Services.....	14
1.2.2 Протокол MQTT .....	15
1.2.3 Протокол CoAP: Constrained Application Protocol .....	18
1.2.4 Протокол XMPP.....	19
1.2.5 Протокол AMQP.....	20
1.2.6 Протокол Websockets.....	21
1.3 Огляд та аналіз способу передачі сигналів в IoT системах.....	23
1.4. Технології для вимірювання напруги .....	28
1.4.1. Дільник напруги .....	28
1.4.2. Трансформатор напруги .....	30
1.4.3. Електронний ізольований датчик.....	31
1.5.1. Шунтовий резистор.....	32
1.5.2. Датчики Холла.....	35
1.5.3. Трансформатор струму .....	35
2. АНАЛІЗ МОНІТОРИНГУ СПОЖИВАННЯ ЕЛЕКТРИЧНОЇ МЕРЕЖІ .....	41
2.1. Огляд характеристик електричної мережі.....	41
2.2. Вимір діючої напруги .....	47
2.3. Вимір різниці фаз .....	49

2.4.	Вимір активної потужності .....	51
3.	<b>ТЕХНІЧНІ АСПЕКТИ СТВОРЕННЯ ТА РЕАЛІЗАЦІЇ ВИМІРЮВАННЯ ПОТУЖНОСТІ ЕЛЕКТРИЧНОЇ МЕРЕЖІ.....</b>	<b>52</b>
3.1.	Розробка структурної схеми.....	52
3.2.	Вибір елементної бази .....	53
3.2.1.	Вибір мікроконтролера для IoT системи.....	53
3.2.2.	Вибір резисторів .....	56
3.2.3.	Вибір датчику напруги. ....	56
3.2.4.	Вибір датчику струму. ....	56
4.	<b>РОЗРОБКА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ. ....</b>	<b>59</b>
4.1.	Розробка REST API серверу.....	59
4.2.	Підключення МК до MQTT серверу та мережі Інтернет.....	60
5.	<b>РОЗРОБКА СТАРТАП-ПРОЕКТУ. ....</b>	<b>59</b>
5.1	Опис ідеї проекту .....	64
5.2	Технологічний аудит ідеї проекту.....	64
5.3.	Аналіз ринкових можливостей запуску стартап-проекту.....	65
5.4.	Розроблення ринкової стратегії проекту .....	69
5.5.	Розроблення маркетингової програми стартап-проекту .....	71
	<b>ВИСНОВКИ .....</b>	<b>75</b>
	<b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>	<b>76</b>
	<b>ДОДАТОК А.....</b>	<b>753</b>
	<b>ДОДАТОК Б.....</b>	<b>764</b>

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

МК	–	мікроконтролер;
HTTP	–	Hypertext transfer protocol (протокол передачі гіпер-текстових документів);
IoT	–	Internet of Things (інтернет речей);
IP	–	Internet Protocol (інтернет протокол);
JSON	–	JavaScript Object Notation;
MQTT	–	Message queue telemetry transport;
QoS	–	Quality of service (якість обслуговування);
RFC	–	Request for Comments (запит коментарів);
RPC	–	Remote procedure call;
SMD	–	Surface mount device (пристрій поверхневого монтажу);
XML	–	Extensible Markup Language (розширювана мова розмітки);
TCP	–	Transmission Control Protocol (протокол керування передачею);
UDP	–	User Datagram Protocol (протокол датаграм користувача);
YAML	–	YAML Ain't Markup Language (YAML — не мова розмітки);

## ВСТУП

Останнім часом приріст споживання електроенергії значно збільшився і, таким чином, потребує значного збільшення постачання енергії в найближчі десятиліття через збільшення кількості населення та економічного розвитку. Це призводить до дефіциту попиту та пропозиції. У багатьох розвинених країнах на рівні домогосподарств впроваджено автоматичне зчитування лічильників (AMR), розширену інфраструктуру вимірювання (AMI) або розумний лічильник енергії з звітом про інформацію про енергію в режимі реального часу. Таким чином, споживачі зможуть побачити їх використання в режимі реального часу, що буде заохочувати їх використовувати менше енергії для економії коштів. Крім того, дослідження [1] припустили, що більше енергії можна заощадити або зменшити на рівні домогосподарств за допомогою отримання звітів щодо споживання енергії в режимі реального часу порівняно зі звичайними звітами, такими як щомісячні рахунки. Однак такі розумні лічильники зазвичай мають високу вартість і потребують великих інвестицій в інфраструктуру комунікаційного середовища. Отже, у багатьох країнах, що розвиваються, це може бути не ефективним та не доступним рішенням.

Взагалі, інтернет речей - це середовище, в якому до Мережі підключено практично все, що може представляти хоча б найменший інтерес для того, хто користується «речами». Інтерес цей може виражатися в можливості «річчю» управляти та дізнаватися її стан. Причому, робити це можна в будь-якому місці і в будь-який час. Особливе становище в концепції IoT займають наші оселі. Це й не дивно. Будинок - це центр життя людини і цілком природно те, що йому потрібно постійне відчуття впевненості в тому, що вдома все добре.

## **1. АНАЛІЗ ТЕХНОЛОГІЇ ІНТЕРНЕТУ РЕЧЕЙ.**

У цьому розділі починається опис архітектурних елементів Інтернету речей в його нинішньому стані. Описується декілька конкуруючих ключових протоколів, що були розроблені для забезпечення роботи Інтернету речей. Розглянуто існуючі мікроконтролери, які сьогодні зазвичай використовуються в системах Інтернету речей. Розглянуто способи виміру параметрів та аналізу домашніх електричних мереж.

### **1.1 Базові поняття Інтернету речей.**

Інтернет речей (IoT, також відомий як Інтернет об'єктів) - концепція, вперше створена Кевіном Ештоном у 1999 р. у контексті управління мережею постачання, була описана система, в якій фізичний світ, через датчики, підключений до Інтернету. Протягом багатьох років концепція розвивалася та поширювалася у межах декількох галузей, впливаючи на здатність сутностей Інтернету речей передавати інформацію, змінювати стан одне одного, взаємодіяти один з одним без втручання людини та виконувати дії, які мають вплив на фізичний світ.

Інтернет речей набуває все більшої популярності в наші дні. IoT використовують в різних областях, таких як автомобільна промисловість, охорона здоров'я, логістика, агрономія та розумні міста, і, в останні роки, інтеграція об'єктів, датчиків та хмарних обчислень, були активною областю досліджень. Проте розвиток цих технологій ще перебуває у фазі становлення, і перед ними постають багато проблем. Наразі продовжуються дослідження щодо стратегій стандартизації, інфраструктури, безпеки, інтерфейсів та протоколів зв'язку, з кількома конкуруючими організаціями та реалізаціями.

Бачення Інтернету речей має кілька глибоких наслідків для нашого суспільства. Більше того, очікується, що переваги будуть не тільки економічними, а також підвищать якість життя в таких аспектах, як охорона здоров'я або життя в містах.

Деякі області Інтернету речей також включають у собі аспекти сталого розвитку. Зокрема, сектор інформаційно-комунікаційних технологій (ІКТ) та Інтернет-речей мають потенціал для підвищення ефективності економіки шляхом надання споживачам та підприємствам можливості вимірювати та оптимізувати свої дії. Сучасний стан Інтернету речей характеризується різноманітним набором стандартів, ініціатив та реалізацій. Стандартизація та взаємодія залишаються проблемою.

Опишемо найважливіші компоненти та архітектурні елементи в Інтернеті речей. На найнижчому рівні існують сенсорні елементи, котрі беруть початок з самих "речей": виконавчих механізмів, датчиків та інших пристроїв, що є частиною Інтернету речей. Очікується, що сотні мільйонів таких пристроїв стануть частиною Інтернету речей, а спеціальні схеми ідентифікації та адресації необхідні завдяки великому масштабу їх розгортання та їх частому обмеженому характеру з точки зору енергії та обчислювальних ресурсів.

### **Проблеми та недоліки**

Найголовнішою проблемою на сьогоднішній день є відсутність стандартів в цій галузі, що ускладнює можливість інтеграцій пропонованих рішень на ринку і багато в чому стримує появу нових.

Також для повноцінного функціонування такої мережі необхідна автономність усіх «речей», тобто датчики повинні не працювати від батарейок чи мережі, а навчитися отримувати енергію з навколишнього середовища.

## 1.2. Засоби взаємодії пристроїв на прикладному рівні

### 1.2.1 RESTFUL Services

The Representational State Transfer (REST) - набір архітектурних стилів побудови сервісно-орієнтованих систем [2]. Щоб використовувати дизайном REST API не потрібно встановлювати зовнішні бібліотеки чи додаткове програмне забезпечення. Це відрізняється неймовірним гнучким шаром, оскільки дані не прив'язані до ресурсів та методів, REST може обробляти різні типи запитів, повертати відповідь у різних форматах даних.

Ця гнучкість та свобода, притаманна дизайну REST API, надає можливість створювати API, що відповідає потребам розробника, а також задовольняє потреби клієнтів. На відміну від SOAP REST не обмежується XML, а може повертати YAML, JSON, XML або ж інші формати в залежності від того, що запитує клієнт. І на відміну від RPC, користувачі можуть не знати конкретні параметри в певному порядку або імена процедур [2].

Архітектура REST умовно складається з клієнтів і серверів. Клієнти ініціюють запити до серверів. Сервери обробляють запити і повертають відповідні відповіді. Запити та відповіді будуються навколо передачі уявлень ресурсів. Ресурс може бути по суті будь-яким послідовним і змістовним поняттям, що може бути адресовано. Подання ресурсу, як правило, це документ, який фіксує поточний стан або призначене ресурсу. REST був спочатку описаний в контексті HTTP, але не обмежується цим протоколом. RESTful архітектура може базуватися на інших протоколах прикладного рівня, якщо вони вже забезпечують багатий і рівномірний словник для додатків, заснований на передачі значущої репрезентативною інформації стану.

## 1.2.2 Протокол MQTT

Telemetry Transport Queue Message (MQTT) – протокол, розроблений IBM, та призначений для забезпечення легких комунікацій M2M. Це асинхронний протокол публікації/підписки, який працює поверх стеку TCP. Використовується для обміну повідомленнями між пристроями за принципом видавець-отримувач. Видавець і отримувач не передають один одному повідомлення безпосередньо, не встановлюють прямий контакт, можуть не знати про існування один одного. Координує і керує передачею повідомлень від видавця до отримувача і від отримувача до видавця брокер (Broker) [3].

Основні риси протоколу MQTT:

- обмін повідомленнями відбувається за принципом "видавець-отримувач" (Pub-Sub);
- розмір заголовка повідомлення становить 2 байта, а корисне навантаження може варіюватися від 1 байта до 260 Мбайт;
- в протоколі закладена можливість вибору одного з трьох рівнів обслуговування.

Розглянемо більш детально процес встановлення з'єднання, відправлення і прийому повідомлень (рис. 1.1).

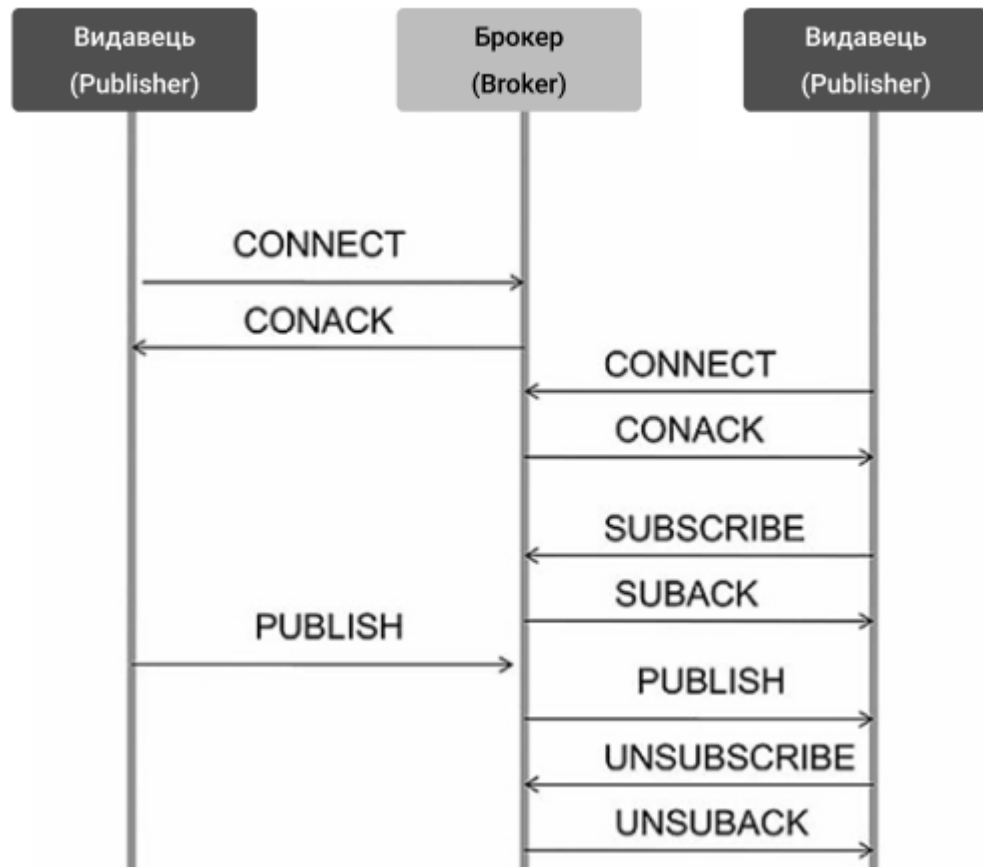


Рисунок 1.1 – Принципи роботи MQTT протоколу

Встановлення з'єднання починається з передачі від клієнта брокеру повідомлення CONNECT, в якому зазначаються:

- `ClientId` – унікальний ідентифікатор для кожного клієнта, що підключається до брокеру;
- `CleanSession` – прапор видалення збережених повідомлень з попередніх сесій для даного клієнта;
- `Username / Password` – ім'я користувача та пароль для ідентифікації та авторизації клієнта;
- `KeepAlive` – регулює передачу ping-запитів і ping-відповідей для контролю відключення однієї зі сторін.

Брокер у відповідь посилає клієнту повідомлення CONACK, що складається з:

- Session Present Flag - вказує чи існують для даного клієнта діючі сесії від попередніх підключень;
- Connect Acknowledge Flag - повідомляє клієнту про успішне підключення або про будь-яку ваду.

Після того, як MQTT клієнт підключений до брокера, він може публікувати повідомлення. Публікація відбувається шляхом відправки брокеру від клієнта повідомлення PUBLISH, де вказується:

- Topic Name - назва теми, до якої відноситься дане повідомлення. Дане поле є обов'язковим, так як MQTT-брокер приймає рішення про пересилку того чи іншого повідомлення клієнту, виходячи з того, на які теми клієнт підписаний;
- спеціальні прапори - QoS, DUP і RETAIN;
- корисне навантаження, де передаються самі дані.

Таким чином, після отримання повідомлення PUBLISH брокер відправляє підтвердження прийому публікації (якщо це задано QoS) і пересилає отримане повідомлення всім клієнтам, які підписані на дану тему[4].

Щоб отримувати повідомлення з необхідними даними, MQTT-клієнт повинен спочатку підписатися на їх отримання за допомогою повідомлення SUBSCRIBE. Дане повідомлення складається з двох частин:

- Packet Identifier – необхідно для QoS 1 і QoS 2;
- List of Subscriptions - назви тим, на які клієнт хоче підписатися, і необхідне значення QoS.

Варто відзначити, що в протоколі MQTT прийнята ієрархічна структура побудови тем, тому для зручності застосовуються «wildcard»-символи, завдяки яким отримувач може підписатися на всі підтеми даної теми (символ #) або теми певного рівня (символ +).

У відповідь при повідомленні SUBSCRIBE брокер відправляє клієнту

підтвердження SUBACK, в якому повідомляє про результат підписки (успішна чи ні).

Також клієнт може відписатися від теми відправивши брокеру повідомлення UNSUBSCRIBE, в якому буде вказана дана тема.

Брокер підтверджує відмову від інформації по цій темі повідомленням UNSUBACK.

### **1.2.3 Протокол CoAP: Constrained Application Protocol**

Simple Object Access Protocol (SOAP) - це протокол на базі мови XML, який задає правила передачі повідомлень по Internet між різними прикладними системами. Він використовується, в основному, для віддаленого виклику процедур. Протокол SOAP спочатку розроблявся з тим розрахунком, що він буде функціонувати «понад» HTTP (для того щоб спростити інтеграцію SOAP в Web-додатки), однак тепер можуть бути використанні інші транспортні протоколи, наприклад SMTP. SOAP був розроблений з урахуванням вимог до малопотужних пристроїв, з можливими невеликими обсягами пам'яті, в мережесередовищах, які можуть бути малопотужними та мати втрати. Протокол розроблений із метою підтримки існуючих M2M додатків, що виконуються поверх існуючих архітектур і взаємодії з існуючими веб-стандартами [5].

CoAP є протоколом передачі документів, який був розроблений для використання з дуже простими електронними пристроями, що дозволяє їм спілкуватися через Інтернет. IETF CORE – робоча група, в даний час працює над стандартизацією CoAP. CoAP призначений для невеликих малопотужних датчиків, вимикачів, клапанів та інтернетпристроїв, таких як бездротові мережі датчиків (WSNs) і призначений для легкої трансляції на

HTTP для спрощення RESTful веб-інтеграції. CoAP легкий, простий і використовує UDP з підтримкою групової адресації (multicast). Він часто використовується в поєднанні з WSNs використовуються IPv6 IETF через 6LoWPAN стандарт. Цей новий стандарт дозволяє використовувати IPv6 в малопотужних і з втратами мережах (LLNs), таких як ті, які засновані на IEEE 802.15.4. (WirelessHART). CoAP підтримує модель програмування клієнт / сервер на основі RESTful архітектури, в якій ресурси є сервером контролюючим абстракції, що надають прикладні процеси і ідентифікуються за допомогою Universal Resource ідентифікаторів (URI). Клієнти можуть маніпулювати ресурсами, використовуючи HTTP: GET, PUT, POST і DELETE методи. Він також забезпечує в побудові підтримку "дискавері" ресурсів в рамках протоколу.

Відображення між CoAP і HTTP також визначається, що дозволяє проксі, щоб бути збудованим, щоб забезпечити доступ до ресурсів в CoAP одноманітно через HTTP.

#### **1.2.4 Протокол XMPP.**

XMPP - є протоколом для передачі поточних XML елементів для того, щоб обмінюватися повідомленнями і інформацією про присутність в близькому до реальному часі. Цей протокол передачі документів був розроблений для використання з дуже простих електронних пристроїв, таких як мобільний телефон. Він є розширюваним, щоб використовуватися для реалізації обміну миттєвими повідомленнями, легкого middleware, голосових і відео дзвінків, передачі файлів, ігор, соціальних мереж і додатків IoT.

XMPP працює над TCP і забезпечує публікацію/підписку (асинхронно), а також повторного запиту/відповіді (синхронно) систем обміну повідомленнями.

Він призначений для операцій зв'язку в режимі реального часу, і, таким чином, він підтримує невеликий розмір повідомлення і обмін повідомлень з низькою затримкою. Як видно з назви, XMPP розширюваний і дозволяє специфікації XMPP Extension Protocols (XEP) збільшити його функціональність [5].

XMPP підтримує архітектуру публікації/підписки, яка більш підходить для IP, на відміну від підходу запиту/відповіді CoAP. Крім того, це вже встановлений протокол, який підтримується в усьому Інтернеті як плюс відносно нового MQTT. XMPP має TLS/SSL захист, побудований в ядрі специфікації. Однак він не надає параметрів QoS, які роблять його непрактичним для M2M-зв'язків.

Однак XMPP використовує XML-повідомлення (eXtensible Markup Language), які створюють додаткові витрати через непотрібні теги, і вимагають розбору XML, що потребує додаткової обчислювальної здатності, що збільшує споживання енергії.

### **1.2.5 Протокол AMQP.**

AMQP (Advanced Message Queuing Protocol) забезпечує взаємодію між клієнтами та брокерами (проміжним ПО для обміну повідомленнями). Він створений для того, щоб шляхом стандартизації повідомлень надати можливість широкому колу різних додатків і систем працювати разом незалежно від їх внутрішньої структури [6].

AMQP - широкий термін, який має на увазі як спосіб організації мереж, так роботу брокерів повідомлень. AMQP також відноситься:

- до маршрутизації і зберігання повідомлень брокерами і наборам правил для визначення поведінки компонентів;

- до протоколу для реалізації зв'язків між клієнтами і брокерами, які виконують вищевказані операції.

Його головна перевага - це функція зберігання та передачі даних, яка забезпечує надійність навіть після збоїв в роботі мережі.

AMQP можна використовувати в будь-якій ситуації, коли існує потреба у високоякісній і безпечної доставки повідомлень між додатками і процесами.

AMQP забезпечує:

- надійність доставки повідомлень;
- високу швидкість доставки повідомлень;
- підтвердження прийому повідомлень.

Протокол AMQP ідеально підходить для:

- моніторингу та спільного використання оновлень;
- налаштування взаємодії між різними системами;
- забезпечення швидкої реакції сервера на запити і передачі трудомістких завдань для подальшої обробки;
- поширення повідомлень декільком одержувачам;
- підключення офлайн клієнтів для подальшого вилучення даних;
- забезпечення асинхронної функціональності системи;
- підвищення надійності і безперебійної роботи додатків.

### **1.2.6 Протокол Websockets.**

Протокол WebSocket був розроблений як частина проекту HTML 5 по

розширенню каналів зв'язку через TCP. Веб-сайт не є протоколом типу запит / відповідь, ні протоколом типу публікація / підпис. В WebSocket клієнт ініціює рукостискання з сервером для створення сеансу. Рукостискання нагадує HTTP, тому веб-сервери можуть обробляти веб-сеанси та HTTP-з'єднання через один порт. Однак, вже після встановлення сеансу, рукостискання не відповідає HTTP [7]. Фактично, під час сеансу HTTP-заголовки видаляються, а клієнти та сервери можуть обмінюватися повідомленнями в асинхронному двонаправленому з'єднанні. Сеанс може бути зупинений, якщо він більше не потрібен ні для серверу, ні для клієнту. WebSocket був створений, щоб зменшити накладні витрати на зв'язок з Інтернетом, забезпечуючи при цьому повнодуплексний зв'язок в режимі реального часу. Існує також підпротокол WebSocket, що називається протокол повідомлень для вебзастосунків (WAMP), який надає системні повідомлення / підписки.

WebSocket працює над надійним TCP і самостійно не використовує механізми надійності. При необхідності сеанси можуть бути захищені за допомогою веб-сайту через TLS / SSL.

Під час сеанса повідомлень WebSocket мають лише 2 байта службових даних. Як повідомляється в відповідних дослідженнях, HTTP-запит (в REST) повторює інформацію заголовка, коли швидкість передачі даних збільшується, що збільшує затримку. За оцінками, WebSocket забезпечує зниження затримок в три рази у порівнянні з половиною дуплексного HTTP-опросу. WebSocket не призначений для пристроїв з обмеженими ресурсами, оскільки попередні протоколи та його клієнт-серверна архітектура не підходять для застосунків IoT. Однак він призначений для обміну в реальному часі, він безпечний, він мінімізує поточні витрати і, використовуючи WAMP, може забезпечити ефективні системи обміну повідомленнями. Таким чином, він може конкурувати з будь-яким іншим протоколом, що працює через TCP.

## 1.2 Огляд та аналіз способу передачі сигналів в IoT системах

В залежності від зони охоплення і деяких інших характеристик мереж для Інтернету речей, бездротові мережі діляться на персональні (PAN), міські мережі (MAN), локальні (LAN) і глобальні (WAN). Для будь-якої мережі важливими властивостями є адаптивність, надійність, можливість самоорганізації, тривала автономна робота і ефективність.

До протоколів, обслуговуючим PAN-мережі, відносяться 6LoWPAN, ZigBee, Thread, BLE 4.2, Z-Wave і інші. Мережі, що базуються на цих протоколах, характеризуються низьким споживанням енергії і малим радіусом прийому сигналів [8].

До LAN-мереж для Інтернету речей можна віднести Wi-Fi. Мережі стандарту IEEE 802.11 мають безліч переваг, серед яких висока швидкість, повсюдна поширеність і сумісність. Але деякі недоліки, такі як енергоємність, перешкоди, вразливість безпеки, обмежують використання протоколу в області IoT, тому спеціально для Інтернет речей був розроблений новий стандарт Wi-Fi HaLow. Wi-Fi HaLow збільшить дальність поширення сигналу до 1000 м, буде працювати на частоті 900 МГц, але зменшить пропускну здатність у порівнянні з передостанній версією протоколу. Але цей стандарт новий, і кількість пристроїв які його підтримують дуже обмежена.

Використання дротового з'єднання не розглядаємо, так як воно не рентабельне та впливає на універсальність приладу, що суперечить технічному завданню.

На даний час найбільш розповсюдженими технологіями для приладів IoT є передача даних за допомогою Wi-Fi або Bluetooth, також набирає популярність ZigBee. Розглянемо їх переваги та недоліки більш детально.

## Передача даних за допомогою Wi-Fi

Wi-Fi - це безпроводна технологія локальної мережі з пристроями, що підтримують стандарт 802.11 IEEE [9]. Wi-Fi - це технологія, яка використовує радіохвилі для надсилання та отримання сигналів пристроям поблизу, забезпечує підключеним пристроям доступ до Інтернету. Wi-Fi встановлює бездротове підключення до пристроїв, випромінюючи частоти від 2,4 ГГц до 5 ГГц, виходячи зі стандарту, що використовується.

Розглянемо основні стандарти IEEE 802.11:

1. 802.11b використовує частоту 2,4 ГГц, досягає швидкості передачі даних до 11 Мбіт/с;
2. 802.11a використовує частоту 5 ГГц зі швидкістю до 54 Мбіт/с, один з перших високошвидкісних стандартів;
3. 802.11g використовує частоту 2,4 ГГц зі швидкістю до 54 Мбіт/с, має високу швидкість і гарантує сумісність з 802.11b, тому і майже витіснив його.

Варто відмітити, що деякі побутові пристрої теж працюють в діапазоні 2,4 ГГц і можуть створювати завади.

Опишемо основні переваги стандарту Wi-Fi.

- найшвидша швидкість передавання даних;
- багато готових рішень і готових вбудованих модулів Wi-Fi;
- гарна стандартизація і сумісність з різними пристроями;

Основними недоліками ж є:

- мала енергоефективність;
- висока вартість модулів (порівнюючи з ZigBee чи Bluetooth).

### **Передача за допомогою Bluetooth.**

Стандарт Bluetooth можна назвати компромісом в співвідношенні параметрів економічності, дальності та швидкості. Bluetooth - це стандарт безпроводного зв'язку, призначений в першу чергу для малого енергоспоживання, роботи на невеликих відстанях, на базі недорогих мікросхем.

Передача даних за допомогою Bluetooth є найпростіша та захищена від перехвату та передачі чужих пакетів, що захищає прилад від втручання сторонніх людей та спотворення результатів. Зазвичай передача даних за допомогою Bluetooth забезпечує надійний зв'язок між двома функціональними модулями на відстані до 200 м.

Варто відмітити, що сучасні версії Bluetooth 4.0 і більше, мають низьке енергоспоживання та можуть використовуватися для електронних датчиків

Передбачається, що Bluetooth буде конкурувати і витіснити ZigBee в класі малоспоживаючих мобільних пристроїв з підтримкою складних мереж.

Проаналізувавши сучасний стан технології Bluetooth, можна позначити плюси і мінуси. До переваг стандарту відносяться:

- стандартизація технології і сумісність з різними пристроями;
- безпека підключення;
- доступність;
- універсальність та наявність вбудованих модулів.

Серед недоліків відзначимо:

- низька енергоефективність (не актуально для нових версій);
- відносно мала швидкість передачі даних (до 2 Мбіт/с для п'ятої версії Bluetooth).

### **Передача за допомогою ZigBee.**

ZigBee технологія працююча на стандарті IEEE 802.15.4, що використовується для безпроводної персональної мережі. Розглядається як альтернатива Wi-Fi і Bluetooth для малопотужних пристроїв, які не вимагають великої пропускної здатності - як розумні домашні датчики. Швидкість обміну даними порівняно з Wi-Fi та Bluetooth невелика - 250 кбіт/с, але технологія і призначена для роботи з датчиками та малопотужними пристроями, де швидкість не настільки важлива.

Вкрай низьке енергоспоживання – одна з ключових особливостей мережевих пристроїв Zigbee. Основну частину роботи пристрій проводить в режимі сну, вмикаючись лише в моменти прийому або передачі даних [10]. Час виходу із сплячого режиму вкрай малий - 15 мс або менше, на відміну від, наприклад, пристроїв Bluetooth, яким для цього може знадобитися кілька секунд. Економічність пристроїв ZigBee дозволяє їм довго працювати з живленням від батарей.

Виходячи з особливостей стандарту ZigBee, сформулюємо його переваги:

- безпека передачі даних;
- висока енергоефективність.

Серед недоліків відзначимо:

- низьку стандартизацію;
- відсутність вбудованих модулів та готових рішень;
- мала швидкість передачі даних.

З приведених даних можливо зробити порівняльну таблицю технологій безпроводного зв'язку.

Таблиця 1.1 – Порівняльна таблиця технологій безпроводного зв'язку

	<b>Bluetooth</b>	<b>Wi-Fi</b>
<b>Частота</b>	2.4 GHz	2.4, 3.6, 5 GHz
<b>Вартість</b>	Низька	Висока
<b>Пропускна здатність</b>	800 Кілобіт	11 Мегабіт
<b>Технічні характеристики</b>	Bluetooth SIG	IEEE, WESA
<b>Безпека</b>	Середня	Середня
<b>Основні пристрої</b>	Мобільні телефони, миші, клавіатури, пристрою для офісної та промислової автоматизації.	Ноутбуки, настільні комп'ютери, сервери, телевізор, мобільні телефони.
<b>Вимоги до обладнання</b>	Bluetooth-адаптер на всіх пристроях, що підключаються один до одного	Бездротові адаптери на всіх пристроях мережі, бездротової маршрутизаторі та / або бездротових точках доступу
<b>Діапазон</b>	5-30 метрів	3 802.11 b / g типовий діапазон складає 32 метри у приміщенні і 95 метрів (300 футів) на відкритому повітрі. 802.11 n має більший діапазон.
<b>Споживання енергії</b>	Низька	Висока
<b>Простота використання</b>	Дуже простий у використанні. Можна використовувати для підключення до семи пристроїв одночасно. Легко переключатися між пристроями або знаходити та підключитися до будь-якого пристрою.	Складніше і вимагає конфігурації апаратного та програмного забезпечення.
<b>Затримка</b>	200мс	150мс
<b>Швидкість передачі даних</b>	2.1 Мегабіт	600 Мегабіт

Bluetooth не надає можливості моніторингу, контролю і керування на великих відстанях між користувачем та блоку керування і прийому/передачі даних.

### **Висновки аналізу існуючих методів передачі даних**

Вимогам технічного завдання задовольняє бездротова технологія. Проаналізувавши бездротові методи передачі даних, можна сказати, що кожна технологія має свої переваги та недоліки. Кожна з них має свою область

застосування, і, відповідно, певні особливості.

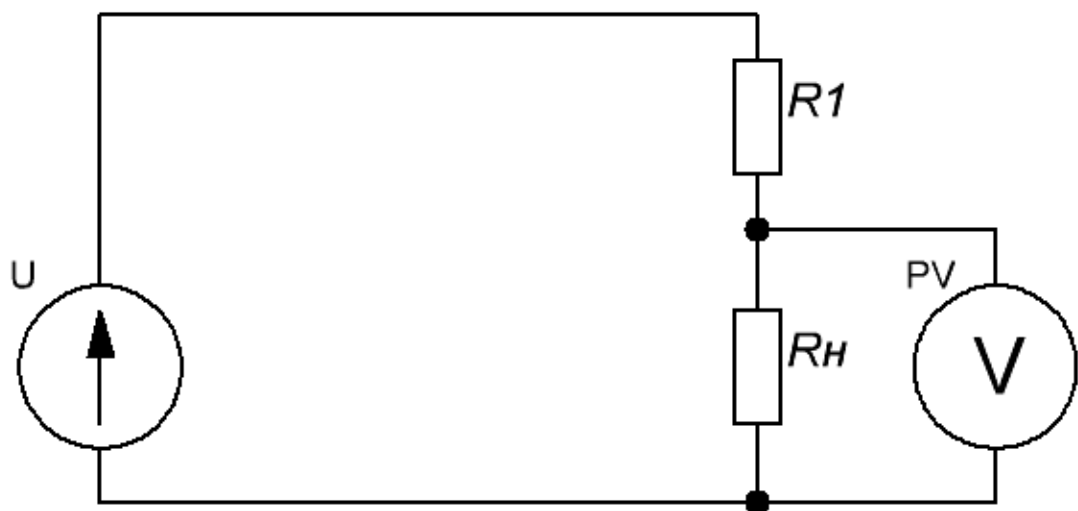
- Bluetooth не працює на достатньо великих відстанях;
- ZigBee не поширена технологія, у якої відсутні вбудовані модулі;
- Wi-Fi не енергоефективний, але поширений та має вбудовані модулі.

Отже, після аналізу існуючих методів передачі даних, можна зробити висновок, що технологія Wi-Fi задовольняє умовам технічного завдання, і може бути взята за основу передачі даних від МК до веб-серверу.

## 1.4. Технології для вимірювання напруги

### 1.4.1. Дільник напруги

Найпростіший спосіб вимірювати напругу – використовувати дільник напруги (рисунок 1.2). Напруга на нашому вимірювальному приладі буде визначатися як вхідна напруга, помножена на коефіцієнт дільника, що залежить від номіналу резисторів.



## Рисунок 1.2 – Схема дільника напруги

При розробці дільника напруги на великі робочі напруги необхідно враховувати гранично допустимі напруги для використовуваних резисторів.

Наприклад номінальна робоча напруга для SMD резисторів - 15 В (0201); 50 В (0402, 0603); 150 В (0805); 200 В (1206, 2010 2512),

А гранично-допустима - 50 В (0201); 100 В (0402, 0603); 200 В (0805); 400 В (1206, 2010 2512).

Саме тому краще використовувати три послідовно-з'єднаних резистора типорозміру 1206 - вони укладаються в 555 В робочого і 1000 В гранично-допустимого.

Зрозуміло, всі ці резистори повинні бути високоточними, так як від їх опору і термостабільності цього опору залежить точність вимірювання.

На базі RC кола, можна зібрати фільтр низьких частот, щоб ніякі перешкоди нам не були страшні.

Переваги дільника напруги:

- широкий діапазон напруги і частот, який визначається номіналами резисторів;
- висока точність, обумовлена точністю і термостабільністю резисторів;
- вимірює постійну і змінну напругу.

Недоліки:

- відсутня гальванічна розв'язка - при взаємодії з мережею необхідно передбачити захист користувача від електричних кіл, або використовувати гальванічну розв'язку;
- низький ККД - весь струм дільника йде в тепло.

### 1.4.2. Трансформатор напруги

Для випадків, коли потрібно виміряти дуже високі напруги, 6 - 10кВ і вище, використовується трансформатор напруги. Фактично, він являє собою звичайний трансформатор, схема якого зображена на рисунку 1.3, основним режимом роботи якого є режим холостого ходу.

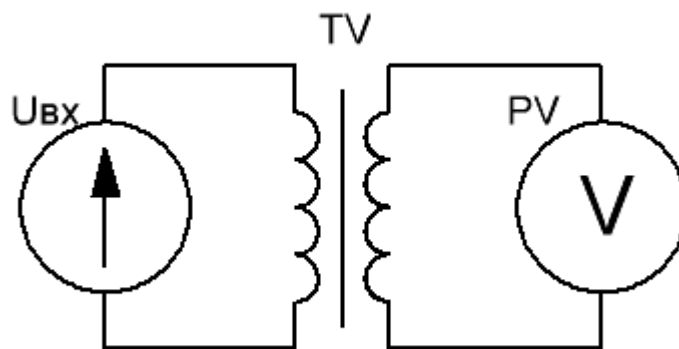


Рисунок 1.3 – Схема трансформатора напруги

Клас точності такого трансформатора залежить від робочої ділянки характеристики намагнічування. Адже нам треба пропустити через нього не просто сигнал з певною амплітудою, але і не зіпсувати його форму. Тут якраз проблема - більшість трансформаторів напруги практично не пропускає гармоніки. Вся справа в металевому сердечнику і втрати на перемагнічування. При цьому, чим тонше пластини сердечника, тим краще його частотні характеристики.

Переваги трансформатора напруги:

- величезний діапазон робочої напруги - до сотень кіловольт і вище;
- така необхідна гальванічна розв'язка.

Недоліки:

- працює на певній смузі частот;
- працює тільки зі змінним напругою

### 1.4.3. Електронний ізольований датчик

Недоліків попередніх схем позбавлений електронний ізольований датчик. Фактично, він являє собою закінчений пристрій. Всередині якого є і дільник напруги, і операційні підсилювачі, і блок гальванічного відокремлення і схема ізольованого живлення всього пристрою. Схему пристрою наведено на рисунку 1.4.

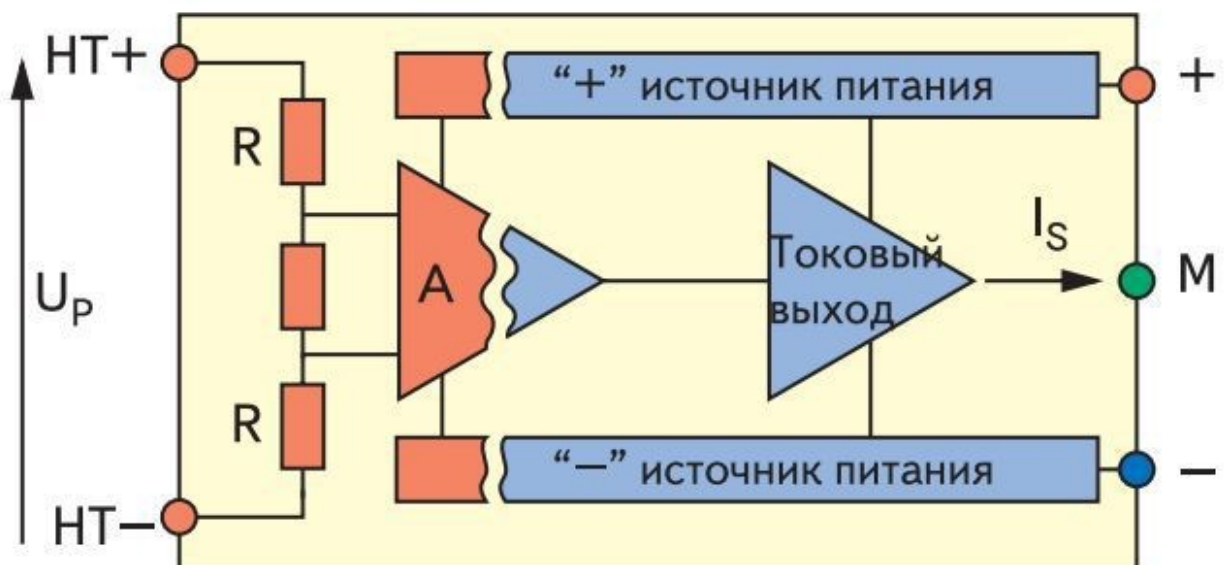


Рисунок 1.4. – Структурна схема електронного ізольованого датчика

Переваги:

- гальванічна розв'язка;

- висока точність;
- широкий діапазон напруги і частот;
- вимірює постійне і змінне напруга.

Недоліки:

- порівняно велика вартість.

## **1.5. Технології для вимірювання струму**

Існують різні способи вимірювання струму, напруги та споживання енергії. Їх відносні властивості, такі як вимога мати гальванічне з'єднання з об'єктом вимірювання, тип вихідного сигналу, точність та робоча температура, важливий для проектування функціональної схеми входу для датчика мережі.

### **1.5.1. Шунтовий резистор**

Найпростіший і найбільш точний спосіб вимірювання струму. Як відомо, при протіканні струму через активний опір, на ньому відбувається падіння напруги, пропорційне вимірюваному струму. Для виміру струму, потрібно помістити резистор в розрив вимірюваного кола, як показано на рисунку 1.5.

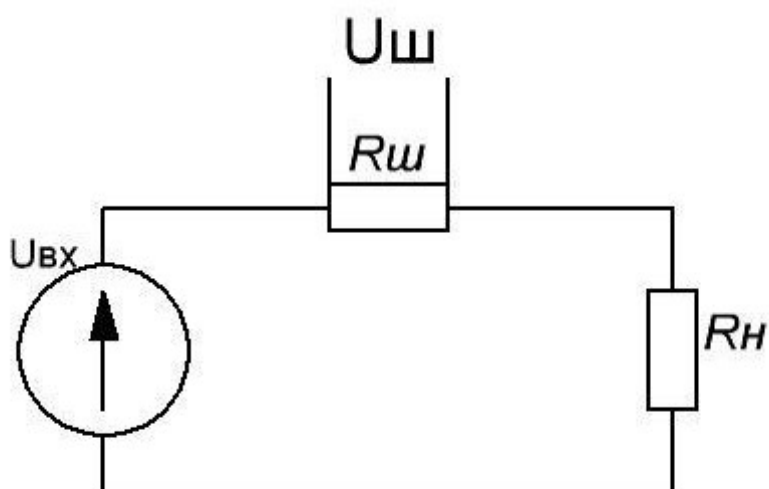


Рисунок 1.5 – Схема датчику струму на основі струмового шунта

Падіння напруги на шунт пропорційне струму, який пропускають через нього.

Відповідно в залежності від необхідного напруги на виході датчика підбираємо потрібне опір шунта. Але, падіння напруги на шунт призведе до втрат потужності, які виділяються в тепло, відповідно при великих струмах ми змушені задовольнятися малими значеннями напруги з датчика, щоб обмежити втрати.

Наприклад, шунти типу ШСМ, зображений на рисунку 1.6, забезпечують стандартну вихідну напругу в 75 мВ при номінальному струмі:



Рисунок 1.6 – Струмівий шунт типу ШСМ

На напругу в 75 мВ відкалібрована більшість вимірювальних головок для шунтів.

Для вимірювання струму за допомогою таких шунтів потрібно використовувати операційні підсилювачі. При цьому, середній коефіцієнт підсилення становить 20-40, що під силу широко-поширеним операційним підсилювачам.

Переваги використання струмового шунта:

- висока точність;
- широкий діапазон напруги та частот;
- вимірює постійний і змінний струм.

Недоліки використання:

- відсутнє гальванічне відокремлення;
- низький ККД.

### 1.5.2. Датчики Холла

Датчики цього типу використовують ефект створення різниці потенціалів при розміщенні провідника зі струмом у магнітне поле. Різниця напруги спостерігається паралельно відхиленню. Принцип датчика Холла показано на рисунку 1.7.

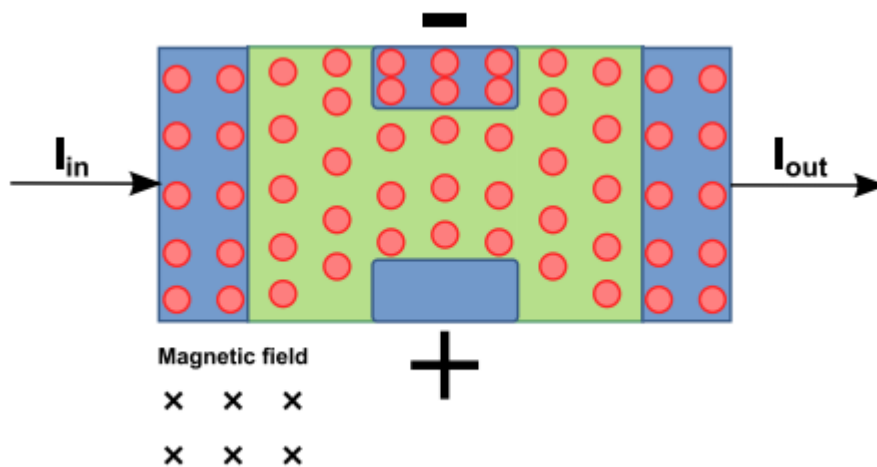


Рисунок 1.7 – Принцип роботи датчика Холла: магнітне поле змушує електрони відхилятися від прямої лінії між верхом та низом провідної підкладки, що викликає різницю напруги, яка прямо пропорційна силі струму.

До переваг датчиків Холла належать:

- широкий діапазон вимірюваних струмів з частотою до 50-100 кГц і вище;
- вимірює постійний і змінний струм;
- гальванічна розв'язка.

### 1.5.3. Трансформатор струму

Сучасні вимірювальні трансформатори струму засновані на тому самому принципі вимірювання магнітного поля навколо провідника струму, що і в

датчиках Холла. Вихідний сигнал трансформатора може бути напругою або струмом. Для напруги вимірювання проводиться по паралельному резистору.

Вимірювальний трансформатор струму являє собою трансформатор, первинна обмотка якого підключається до джерела струму, а вторинна замикається на вимірювальні прилади або пристрої захисної автоматики. Трансформатори з вбудованою первинною обмоткою зображені на рисунку 1.8.



Рисунок 1.8 – Трансформатор струму серії CS2106L від Coilcraft

Трансформатори струму використовуються для вимірювання струмів в потужнострумівих ланцюгах, часто з високим потенціалом. Основна проблема трансформаторів струму полягає в тому, що вони вмiють вимірювати тільки змінну напругу.

Трансформатор струму завжди навантажується. Якщо вторинна обмотка трансформатора струму виявиться розімкнутою, то на ній виникне потенціал в пару тисяч кіловольт, який може вивести з ладу прилад, пробивши його ізоляцію.

Основний недолік трансформатора струму - це робота тільки на певній частоті 50, 60 або 400 Гц через металевого осердя. Зрозуміло, трансформатор

струму здатний передати сигнал і більшої частоти, проте з набагато меншою точністю. При цьому, слід звернути увагу на товщину листа використовуваного осердя - чим тонше залізо в трансформаторі струму, тим вище гранично допустима робоча частота.

А ось якщо ми видалимо сердечник, то отримаємо повітряний трансформатор, або, так звану котушку Роговського, схема якого зображена на рисунку 1.9.

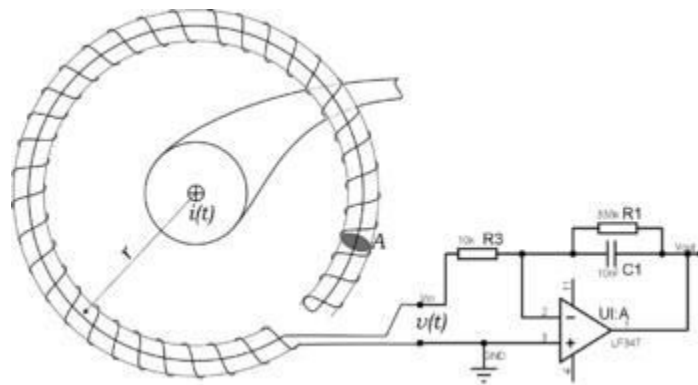


Рисунок 1.9 – Схема підключення котушки Роговського

На відміну від інших датчиків, які потребують взаємодії з вимірюваної ланцюгом, котушку Роговського можна встановити поверх проводів вимірюваної ланцюга як хомут.

Переваги використання трансформаторів:

- є гальванічне відокремлення;
- робота з великими струмами в тисячі ампер.

Недоліки:

- вимірює тільки змінний струм в певному діапазоні частот (крім котушки Роговського);

- змінює фазу сигналу і вимагає компенсації.

### **1.5. Огляд існуючих систем та патентний пошук**

Було здійснено патентний пошук аналогів та прототипів пристрою, що розробляється. Винахід належить до області інтернет речей та може відноситись до технології автоматичного поливу чи бути елементом системи «smart house».

#### **Система управління інфраструктурою будинку “розумний дім” [11]**

Номер патенту: 81054. Автор Калінін Дмитрій Юрійович. Система управління інфраструктурою будинку, що містить датчики, відеокамери і виконавчі пристрої, з'єднані з сервером управління, яка відрізняється тим, що як сервер управління використаний принаймні один шлюз, що містить безпроводний приймально-передавальний елемент і забезпечує вихід у мережу Ethernet, а мережею Internet зв'язаний з хмаровим сервером, при цьому система виконана з можливістю реєстрації пристроїв і конфігурування алгоритмів їх взаємодії за допомогою WEB-сервісу у хмаровому сервері з подальшою передачею конфігурації у шлюз.

1. Система виконана з можливістю працювати на одній з найрозповсюджених частот і приймати найбільш розповсюджені протоколи безпроводного зв'язку.
2. Шлюз виконаний з можливістю підтримки керування системою за допомогою Android планшета або смартфона через домашню систему WI-FI.
3. Система виконана з можливістю за допомогою програми "бутлоадер" постійно оновлювати внутрішні програми з хмарового серверу, вносячи інші стандарти інтелектуальних можливостей самого шлюзу

по зв'язку з зовнішніми пристроями керування.

### **Пристрій керування системами будинку [12].**

Номер патенту 62854. Автор: Панченко Денис Юрійович. Пристрій керування системами будинку, який включає модуль прийому сигналів і модулі управління, які встановлені в місцях збору інформації і виконання команди з модуля управління, який відрізняється тим, що як модуль прийому сигналів використана як мінімум одна IP-камера, яка по засобах wi-fi або ethernet-мережі зв'язана із відеосервером, причому пристрій містить як мінімум одну сенсорну панель, а також телефон, який через GSM-мережу зв'язаний з GSM-модулем і з сервером управління, причому як модуль управління використаний комп'ютер, зв'язаний з веб-сервером, який розташований на сервері управління, медіа-сервер, який містить медіа-бібліотеку систем мультимедіа, а також точку доступу wi-fi, причому GSM-модуль зв'язаний з мобільними телефонами користувача, причому до складу пристрою також додатково введені модулі управління CAN-BUS мережею.

#### **Відмінності**

1. У складі пристрою - шлюз GW4X2, модем, а також модуль управління опалюванням HCX8, пов'язаний з опалювальними елементами.
2. Система мультимедіа зв'язана з медіа-сервером через ethernet-мережу

### **Висновки до розділу**

1. На підставі базових принципів технології інтернету речей та систем ця технологія обрана для подальшого впровадження.
2. Аналіз сучасних стандартів опису та протоколів передачі даних між елементами Інтернету речей показали, що доцільно використовувати протокол MQTT, для обміну даних між МК та сервером. Проаналізовані типові архітектури рішень на базі протоколу MQTT, підтверджують

доцільність використання протоколу, для постійного відправлення однотипних даних, без підтвердження отримання.

3. На підставі недоліків та переваг використання основних технологій для вимірювання струму та напруги, обрано використання електронного ізольованого датчику для виміру напруги та датчику на основі ефекту Холла для вимірювання струму.

## **2. АНАЛІЗ МОНІТОРИНГУ СПОЖИВАННЯ ЕЛЕКТРИЧНОЇ МЕРЕЖІ**

### **2.1. Огляд характеристик електричної мережі.**

Моніторинг характеристик мережі цілого будинку вимірює характеристики електричної мережі, які використовуються приладами, підключеними до будинку. Щоб зрозуміти, особливості цього процесу, доцільно визначити, як прилади взаємодіють з електричною системою.

#### **Резистивні навантаження**

Не всі прилади взаємодіють з електричною системою однаково. Спочатку буде розглянуто резистивні навантаження та спосіб розрахунку потужності, яку вони використовують. Далі розглянемо реактивні навантаження і трохи про нелінійні навантаження. Нарешті, розглянемо, як ми вимірюємо напрям потоку енергії, що важливо, якщо енергія генерується та споживається.

Лампи розжарювання, чайники, праски, електричні водонагрівачі, електроплити - прості. Вони використовують всю надану їм енергію. Вони є резистивними навантаженнями, а значить, струм їх дорівнює напрузі, поділеній на їх опір (Закон Ома). Чисто резистивне навантаження дає вигляд напруги та струму, подібний до зображеного на рисунку 2.1:

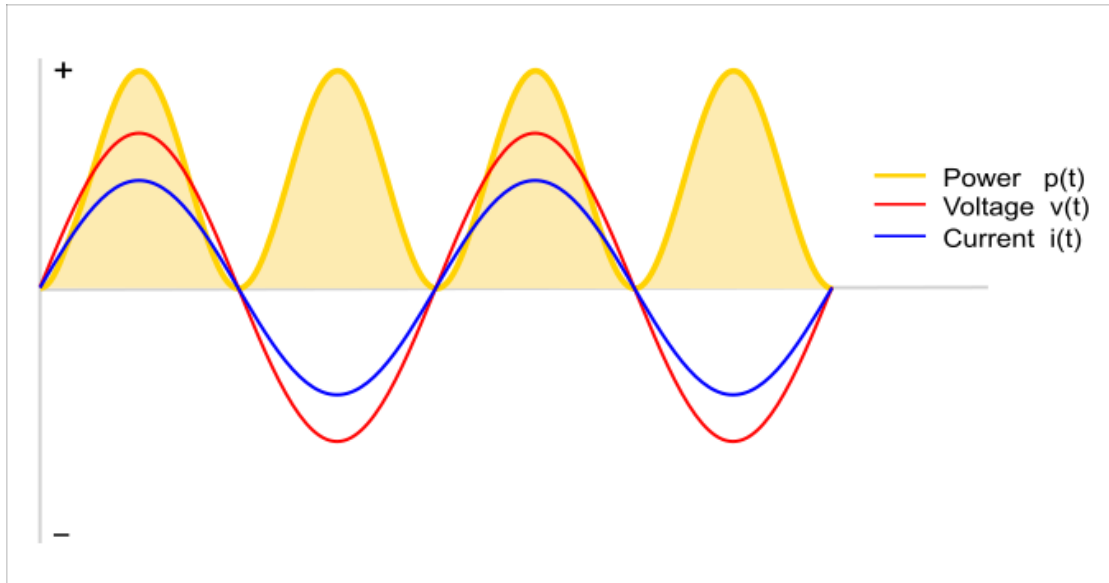


Рисунок 2.1. — Залежність напруги та фази струму при активному навантаженні

Жовта лінія - це потужність у даний момент часу (у заданий момент часу її називають діючою потужністю), яка дорівнює добутку напруги та струму в даний момент часу. Можна побачити, що потужність завжди позитивна.

### **Частково реактивне навантаження**

Однак такі речі, як холодильники, пральні машини, електро-свердла та дугові зварювачі не настільки прості, оскільки ці прилади забирають певну кількість енергії, а потім відпускають частину енергії назад у електромережу.

На додаток до резистивного компонента вони мають індуктивні (наприклад, двигуни) або ємнісні (наприклад, дугові зварювачі). Частково індуктивне навантаження дає вигляд напруги та струму, подібний зображеному на рисунку 2.2.

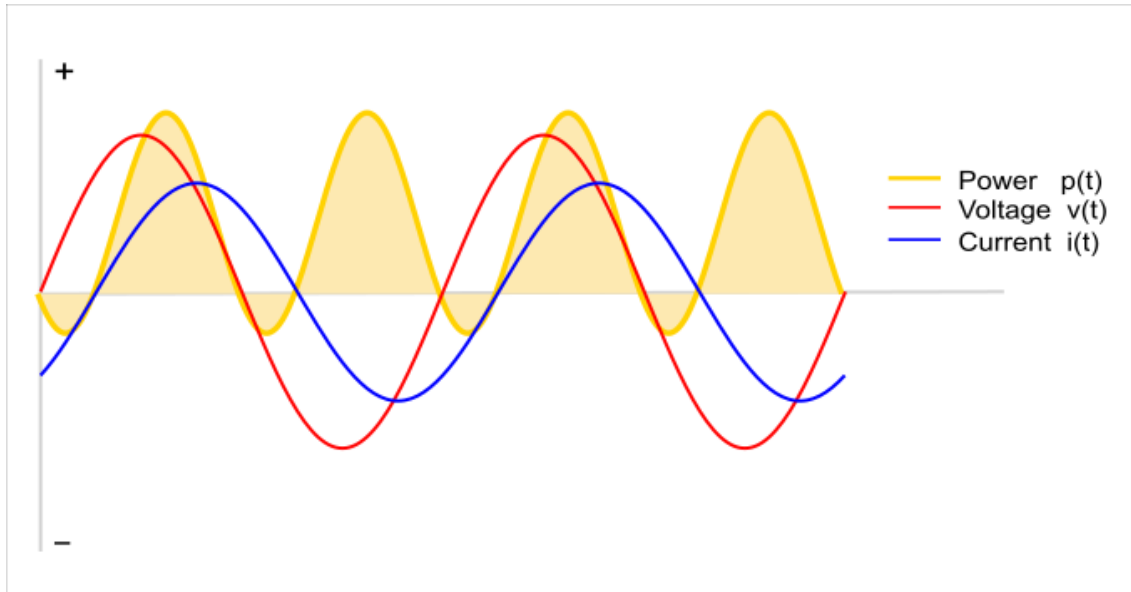


Рисунок 2.2 — Залежність напруги та фази струму при частковому реактивному навантаженні

Можна побачити, що жовта лінія на певний проміжок часу стає негативною, позитивна частина - це енергія, що надходить до навантаження, а негативна - енергія, що стікає назад від навантаження.

Інша річ, яку слід врахувати, - це те, що форми напруги та струму зміщені. Як приклад, розглянемо, що заряджаємо досить великий конденсатор з послідовним резистором (щоб він не міг заряджатися миттєво). Для початку конденсатор розряджається. Напруга живлення зростає і перевищує напругу на конденсаторі, тому струм впадає в конденсатор (позитивна частина на графіку), через що напруга конденсатора зростає. Напруга живлення падає. Тепер напруга в зарядженому конденсаторі вище напруги живлення. Струм починає текти назад у напрямку подачі (негативна частина на графіку). Це призводить до того, що поточна форма хвилі виглядає так, ніби вона зміщена, як зображено на рисунку 2.2., це називається зсувом фази.

### Активна потужність, реактивна потужність і повна потужність

Активна потужність часто визначається як потужність, яка використовується пристроєм для виробництва корисної роботи [13]. Посилаючись на рисунок 2.2 вище, позитивна частина - це потужність, що надходить на навантаження від живлення, а негативна частина - потужність, що повертається до живлення, від навантаження. Потужність, яка фактично була використана навантаженням, тобто потужність, що йде, мінус потужність, яка повертається, - це активна потужність.

Реактивна або уявна потужність - це міра потужності, що рухається вперед і назад між навантаженням і подачею, яка не робить корисної роботи.

Іншим корисним показником потужності є «повна потужність», яка є добутком діючого значення напруги та струму. Для чисто резистивних навантажень активна потужність дорівнює повній потужності. Але для всіх інших навантажень активна потужність менше, ніж повна потужність. Повна потужність - це міра активної та реактивної потужності, але це не сума двох, оскільки сума двох не враховує фазових різниць.

Зв'язок між активною, реактивною та повною потужностями для ідеальних синусоїдальних навантажень, представлений формулами (2.1, 2.2) [13]:

$$P = V I \cos \phi, \quad (2.1)$$

$$S = V I, \quad (2.2)$$

де  $P$  — це активна потужність, Вт,  $S$  — повна потужність, В-А,  $Q$  — реактивна потужність, В-А-Р,  $\cos \phi$  — коефіцієнт потужності.

Однак, ця залежність коефіцієнта потужності справедлива лише для

лінійних синусоїдальних навантажень. Більшість джерел живлення для пристроїв постійного струму, таких як портативні комп'ютери, представляють нелінійне навантаження на електромережу. Їх графічне зображення часто виглядає, як зображено на рисунку 2.3.

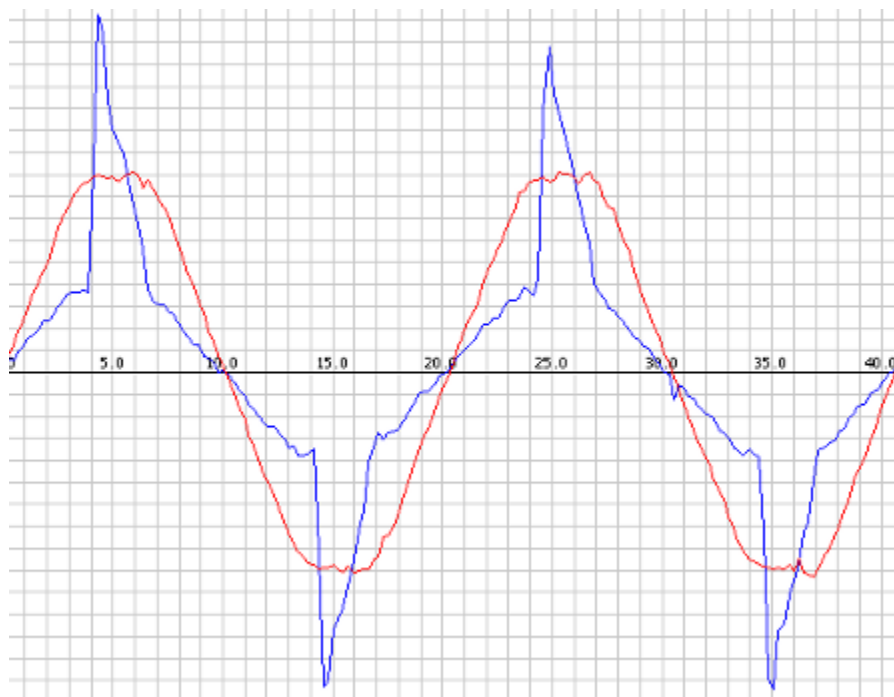


Рисунок 2.3 — Вигляд споживаючого струму при не лінійному навантаженні

Обчислюємо коефіцієнт потужності з наступного рівняння:  $P = UI \cos \phi$ , де  $\cos \phi$  — коефіцієнт потужності, але залежність  $\cos \phi$  від  $\phi$ , що вірна для чистих синусоїд, більше не є правильною. Коефіцієнт потужності також не є  $\cos \phi$ , оскільки слід враховувати ефекти гармонік вищого порядку як у напрузі, так і у струмі.

### Визначення напрямку потоку живлення

Якщо будинок генерує енергію, напрям струму змінюється в зворотному порівнянні з іншими попередніми прикладами. Тепер, коли напруга позитивна, (вище осі X) струм негативний (нижче осі X), а коли напруга від'ємна, струм позитивний. Потужність завжди негативна - вся крива потужності знаходиться нижче осі X (рис. 2.4).

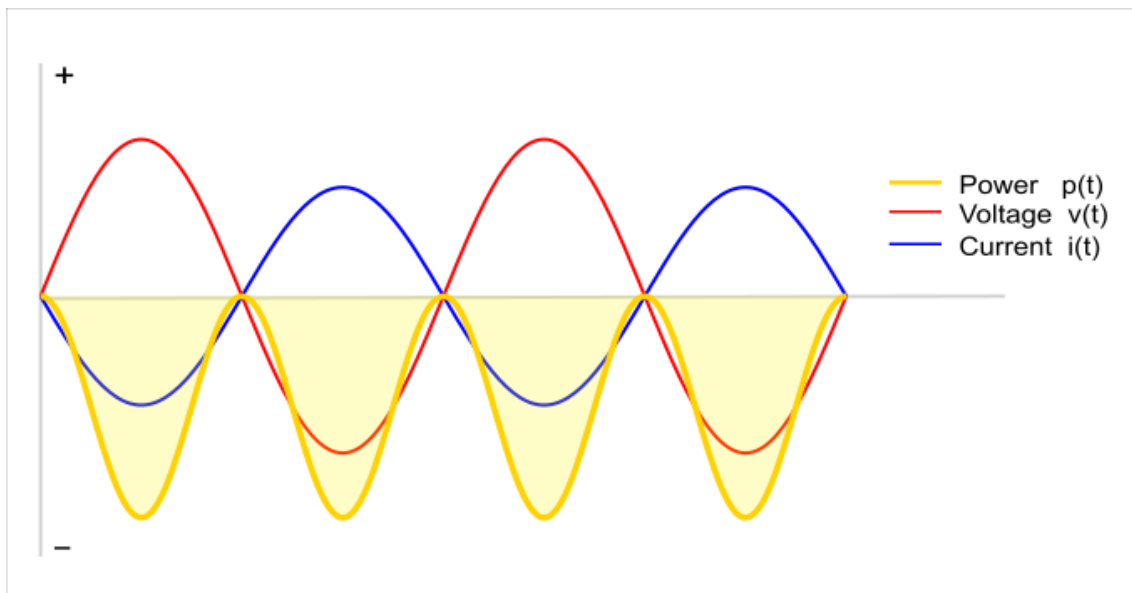


Рисунок 2.4 — Залежність напруги та фази струму при генерації енергії.

Таким чином, знак потужності вказує, споживається чи виробляється енергія.

Можна вимірювати багато параметрів, що характеризують використання енергії в мережах змінного струму. У кожного є свої переваги. Для побутового обліку енергії активна потужність, ймовірно, буде найбільш корисною, оскільки вона говорить про те, скільки енергії споживають усі ваші прилади, і це те, за що сплачуються комунальні послуги.

## 2.2. Вимір діючої напруги

Діюче значення періодичного сигналу, зображеного на рисунку 2.5, розраховують за формулою (2.3)

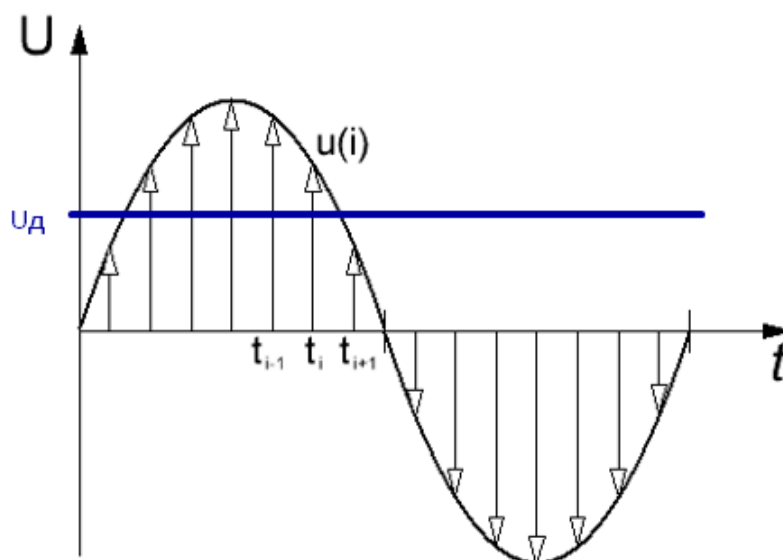


Рисунок 2.5 — Графік залежності напруги від часу

$$U_d = \sqrt{\frac{1}{T} \int_0^T u^2(t) dt} \quad (2.3)$$

де  $T$  — період сигналу,  $U$  — амплітудна напруга, на відріжку часу  $T$ ,  
— діюча напруга.

Інтеграл  $\int_0^T u^2(t) dt$ , являє собою площу змінного сигналу за один період  $T$ . Діюче значення періодичного сигналу, Якщо розглядати цифрову систему виміру сигналу, то потрібно провести дискретизацію сигналу, де площу змінного сигналу за один період  $T$ , буде представляти сума прямокутників, де площа

кожного прямокутника, це добуток амплітудної напруги та , де – малий проміжок часу, і розраховується по формулі 2.4,

$$- , \quad (2.4)$$

де  $T$  – період сигналу, – кількість вимірів за період.

Для мікроконтролера MSP430, що використовується в електролічильниках Меркурій 200, використовує 4096 вимірів за 1 секунду, що можна буде взяти для розрахунків. Таким чином можна перейти від інтеграла до суми

$$\frac{\text{-----}}{- ,} \quad (2.5)$$

де  $T$  – період сигналу, - амплітудна напруга, на відрізок часу , – діюча напруга.

Графік зображений на рисунку 2.2 був зроблений за допомогою виміру напруги та струму мережі на частоті 50 Гц.

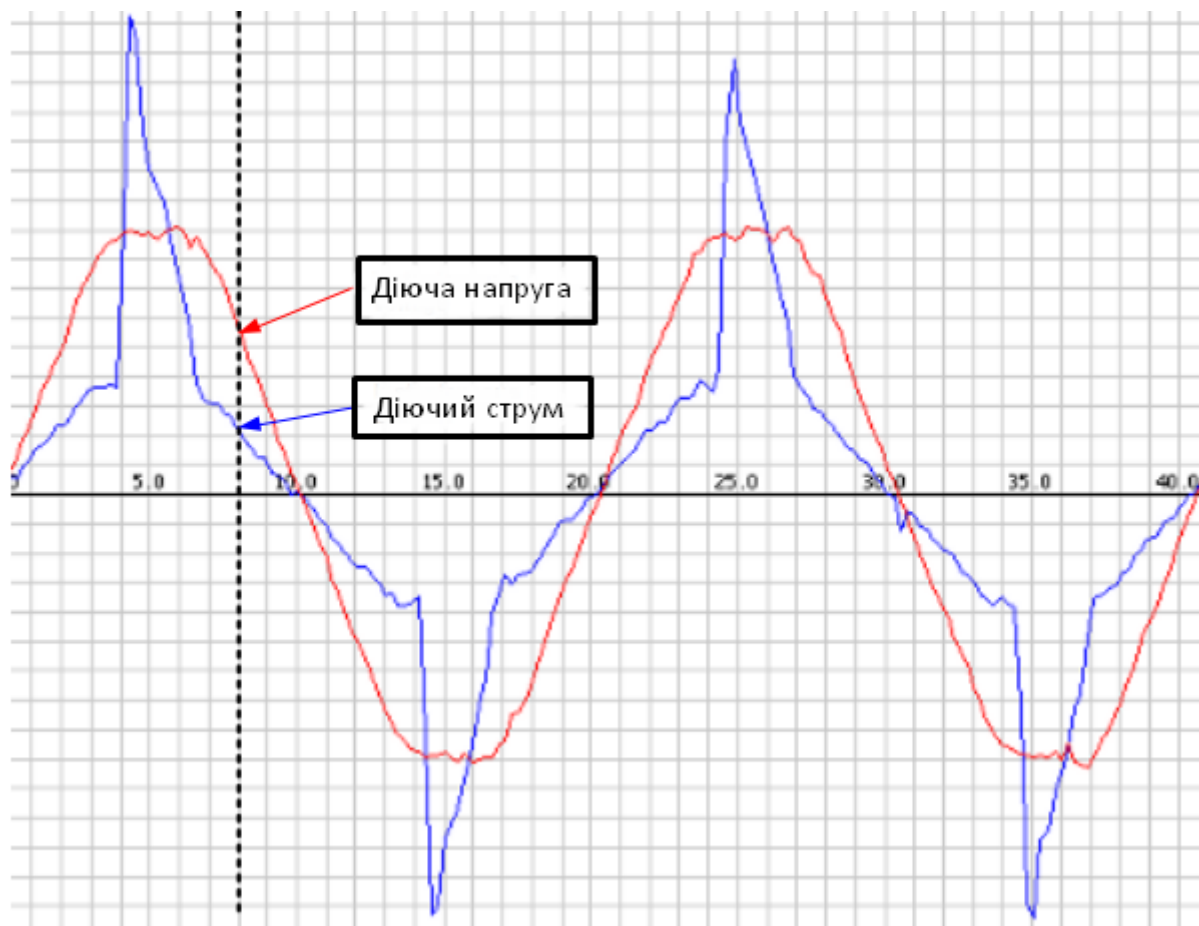


Рисунок 2.6. — Графік виміряної напруги та струму

Було зроблено від 50 до 100 вимірювань кожні 20 мс.

- 100, якщо вимірюється тільки струм;
- 50, якщо вимірюється напруга та струм.

Виміри обмежені аналоговою командою та швидкістю обчислення аналогового сигналу МК.

### 2.3. Вимір різниці фаз

Якщо у плати наявний лише один аналого-цифровий перетворювач, але з великою кількістю каналів то для того, щоб залишити частоту дискретизації в

4096 Гц, що використовують у стандартних лічильників, необхідно цю частоту збільшити вдвічі і кожного разу переключати канали, таким чином в заданий момент часу, можна вимірювати силу струму, перемикаючи канал і в наступному семплі вимірювати значення діючої напруги, таким чином після 8192 семплів, буде 4096 наборів по струму та напрузі, по яким можна вимірювати діючу потужність.

Для виміру коефіцієнта потужності, на основі одного періоду, використаємо метод детектора переходу через нуль [14]. Суть метода полягає в вимірі часу при якому діюче значення напруги та струму дорівнюють нулю, їх різниця буде час між цими двома моментами, що визначає кут зсуву між двома синусоїдами, цей час розраховують як:

$$\tau = \frac{t}{N} \quad (2.6)$$

де  $t$  – час між нулями сигналів,  $N$  – кількість вимірів, на відрізок часу  $\tau$ ,  
—, де  $f$  – частота АЦП для одного каналу.

Кут зсуву буде розраховуватися через кругову частоту мережі помножену на час між нулями (2.7).

$$\varphi = 2\pi f \tau \quad (2.7)$$

Кругова частота мережі обчислюють за частотою мережі (2.8)

$$\omega = 2\pi f \quad (2.8)$$

Частоту мережі можна обрахувати, використовуючи метод переходу через нуль. Використовуючи лише одну синусоїду, тоді час між нулями сигналу буде період цього сигналу  $T$ .

$$T = \frac{1}{f} \quad (2.9)$$

де  $N$  – кількість вимірів за один період сигналу напруги чи струму.

Таким чином кругову частоту можна описати через частоту АЦП

(2.10)

## 2.4. Вимір активної потужності

Для розрахунку активної потужності потрібно обчислити повну потужність, помноживши діючу напругу на діючий струм та розрахувати коефіцієнт потужності.

Знаючи кут зсуву, та діючі значення напруги та струму, розраховувати активну потужність не є проблемою.

(2.11)

У будь-якому електричному колі як синусоїдального, так і несинусоїдального струму активна потужність всього ланцюга дорівнює сумі активних потужностей окремих частин ланцюга, для трифазних ланцюгів електрична потужність визначається як сума потужностей окремих фаз.

### Висновки до розділу

1. Для отримання характеристик електричної мережі, потрібно вимірювати діючу напругу та струм, на всьому періоді синусоїдального сигналу.
2. Повна потужність — це міра активної та реактивної потужності. Активна потужність, визначає, скільки енергії споживають усі ваші прилади, реактивна — скільки енергії вони повертають назад до мережі. Їх співвідношення характеризує коефіцієнт потужності.
3. Використовуючи аналого-цифровий перетворювач, можна вимірювати, діючу напругу та струм, визначаючи різницю в часі між переходами цих значень нуля, можна обрахувати коефіцієнт потужності, для розрахунку активної чи реактивної потужності

### 3. ТЕХНІЧНІ АСПЕКТИ СТВОРЕННЯ ТА РЕАЛІЗАЦІЇ ВИМІРЮВАННЯ ПОТУЖНОСТІ ЕЛЕКТРИЧНОЇ МЕРЕЖІ

#### 3.1. Розробка структурної схеми

Структурна схема пристрою складається з МК з підключеною периферією та веб-додатку (рис. 3.1), для віддаленого моніторингу та керування. Схема складається з наступних блоків:

- блок індикації;
- блок живлення;
- блок віддаленого сервера і клієнту веб-додатка;
- мікроконтролер;
- датчик струму;
- датчик наруги;
- чотирьох-канального реле для підключення виконавчих блоків.

Центральним блоком структурної схеми є МК, що відповідає за обробку даних, вміщує модуль Wi-Fi, порти вводу-виводу, інтерфейси SPI, I2C, UART та інші.

Для відображення даних з датчиків та ручного управління реле, використовується веб-сервер [15]. Сервер отримує показання датчиків від МК через протокол MQTT. Клієнт, через запити побудовані по принципу REST API, отримує показання датчиків, та надсилає їх користувачеві, також клієнт має змогу відправляти запити на сервер для ручного керування реле.

За допомогою реле комутують зовнішні джерела живлення для виконавчих блоків.

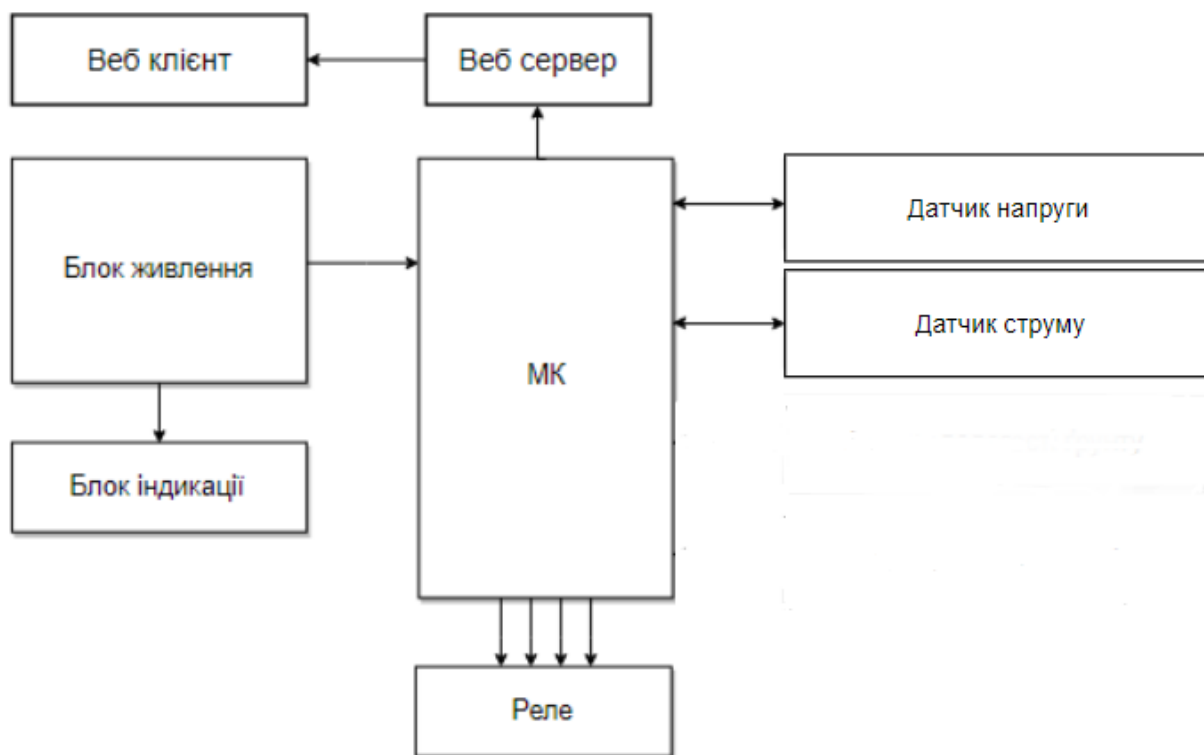


Рисунок 3.1 — Структурна схема системи

## 3.2. Вибір елементної бази

Варто зазначити, що так як вимоги до вартості доволі низькі, то можна зменшити вимоги для вартості елементної бази.

### 3.2.1. Вибір мікроконтролеру для IoT системи.

Для реалізації IoT системи, а саме побудови блока керування та обробки даних потрібно використовувати платформу, що поєднує наступні характеристики.

Модуль Wi-Fi, який буде вбудований як окремий контролер на платі (не

буде виникати необхідність у використанні додаткових плат/плат розширень до головної плати керування).

Розгалужену GPIO, тобто достатня кількість цифрових та аналогових портів для підключення зовнішніх елементів периферії (датчик напруги, датчиків струму та температури і т.д).

Низьке енергоспоживання є головним аспектом при розробці та проектуванні сучасних пристроїв (їх прототипів). При розробці конкурентоспроможних систем енергоефективний аспект є головуючим. Розглянемо основні рішення представлені на ринку.

### **ESP8266**

Мікроконтролер приваблює невеликою ціною, здатністю підтримувати режими зниженого споживання, вбудованим модулем Wi-Fi. Але доступно лише 11 портів вводу-виводу (1 - SPI, 1 - I<sup>2</sup>C, 1 - UART), що є його основним недоліком.

### **chipKIT Wi-FIRE**

Достатньо потужна та доступна плата, має вбудований модуль Wi-Fi, розгалужене GPIO(43 доступних портів входу-виходу, 6 - SPI, 6 - UART, 5 - I2C). Недоліком є застаріле програмне забезпечення фреймворку, останнє оновлення якого було в 2012 році.

### **ESP32**

Оновлення мікроконтролера ESP8266. В ньому виправлені недоліки попередника та додані нові функції. Основні відмінності – це двоядерний процесор, збільшена кількість портів вводу-виводу (до 36 портів, 4 - SPI, 2 - I2S, 2 - I2C, 3 - UART), збільшення пам'яті та потужності, доданий Bluetooth. Збільшилась і ціна, але економічні показники в технічному завданні ми не враховуємо.

## Платформа Arduino.

Великий спектр можливостей за рахунок легкого створення прототипів та великої кількості ком'юніті які створюють бібліотеки для більш простого використання модулів та датчиків. Але на платах не має вбудованого модуля Wi-Fi (потребує використання додаткової периферії).

## Microchip Technology PIC32

Сімейство мікроконтролерів PIC32 має низьке енергоспоживання та доступність і в сукупності із великою кількістю готових рішень являється гарним варіантом. Також цей модуль має вбудований Wi-Fi, але цей модуль працює через спеціальний фреймворк від Microchip який вже не підтримується, тому використовувати ці мікроконтролери недоцільно.

Таблиця 2.1 – Порівняльна таблиця мікроконтролерів

МК	Характеристики				
	Напруга живлення	Входи GPIO	Вбудовані модулі Wi-fi	Розрядність АЦП	CPU
ESP8266	3,3 – 3.7 V	16	+	10-bit	80 або 160 МГц
chipKIT Wi-FIRE	3,3 - 5,2V	43	+	10-bit	200 МГц
ESP32	3.3 – 3.7V	32	+	12-bit	160 або 240 МГц
Arduino Uno	3.3 - 9V	22	-	10-bit	16 МГц
PIC32	3,5 - 5,5V	28	-	10-bit	200 МГц

На основі таблиці, переваг та недоліків, обираємо ESP32.

### 3.2.2. Вибір резисторів

Резистори дозволяють контролювати значення струмів і напруги в електричному ланцюгу. Електричні характеристики резистора значною мірою визначаються матеріалом, з якого він виготовлений, і його конструкцією.

При виборі типу резистора враховуємо наступні параметри:

- нормований допуск;
- розсіювання потужності;
- робочу температуру;
- тип резистора;
- максимальну робочу напругу;
- спосіб монтажу.

Будемо використовувати резистори SMD 0,125Вт 0805 5% Nitano. У пристрої використання високоточних резисторів недоцільно, тому допуск  $\pm 5\%$  підходить. Потужність 0,125Вт. Резистор керамічний.

Робоча температура  $-55 \dots + 125 \text{ }^\circ\text{C}$ , ці величини з великим допуском покривають температурний режим роботи резистора в пристрої. Максимальна робоча напруга 150 В є достатньою для коректної роботи пристрою.

### 3.2.3. Вибір датчику напруги.

Для вимірювання напруги та довговічної роботи датчика, потрібно вибирати герметичні, захищені від зовнішнього впливу датчики. Для заданих умов підходить електронний ізольований датчик. Для виміру напруги,

підходить перетворювач напруги ZMPT101B, він вимірює напруго до 250 В, має вихідний аналоговий сигнал 0 – 5 В, має бортовий точний перетворювач напруги. Може регулювати вихідний сигнал, загалом дуже добре підходить для виміру напруги та потужності.

#### 3.2.4. Вибір датчику струму.

Для вимірювання струму найбільше підходить неінвазивний датчик SCT013-000 (рис. 3.2). Він може вимірювати струм до 100 А, має вихід до 50мА, для підключення потрібно використовувати схему підключення, що дозволяє підібрати номінали, для конкретної мережі, зменшуючи похибки вимірювань. А головне використовує гальванічне роз'єднання для зручного використання.

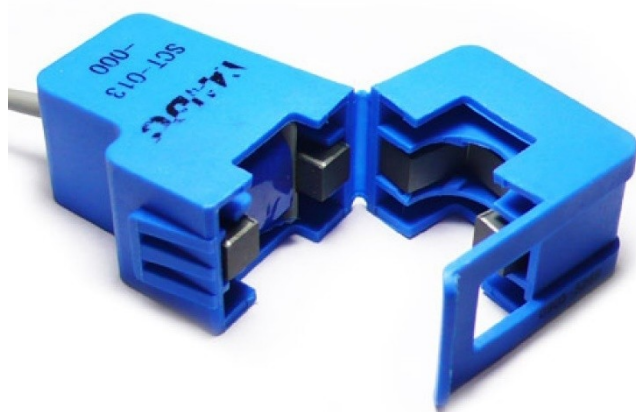


Рисунок 3.2 — Вигляд датчика SCT013-000

### **Висновки до розділу**

1. Структурна схема включає МК, датчики, індикацію, виконавчі блоки, та розділену архітектуру додатку, що включає сервер, що має REST архітектуру та клієнт.
2. Проаналізувавши наявну елементну базу, обрано ключові елементи схеми, а саме МК, резистори, датчик напруги та струму.

## **4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ**

### **4.1. Розробка REST API серверу.**

Для розробки серверу будемо використовувати платформу `node.js`, що виконує програми які написані на мові `javascript`. Платформа популярна, має багато готових рішень та легка в масштабуванні та відладці.

Процес відправки запитів від клієнту до серверу представлена на рисунку 4.1. При підключенні нового клієнту, відбувається запит на сервер для отримання показань від датчиків [16]. Сервер підключається до MQTT серверу (брокеру) та підписується на отримання повідомлень з темою «`/data`». Це повідомлення, які відправляє МК. При отриманні повідомлення та його обробки, сервер відправляє покази, що отримує від датчиків, клієнту у `json`-форматі. Отримувачи їх, клієнт десериалізує дані та відображує користувачу на веб-сторінці. Приклад програмної реалізації можна побачити у приведеному до роботи додатку Б.

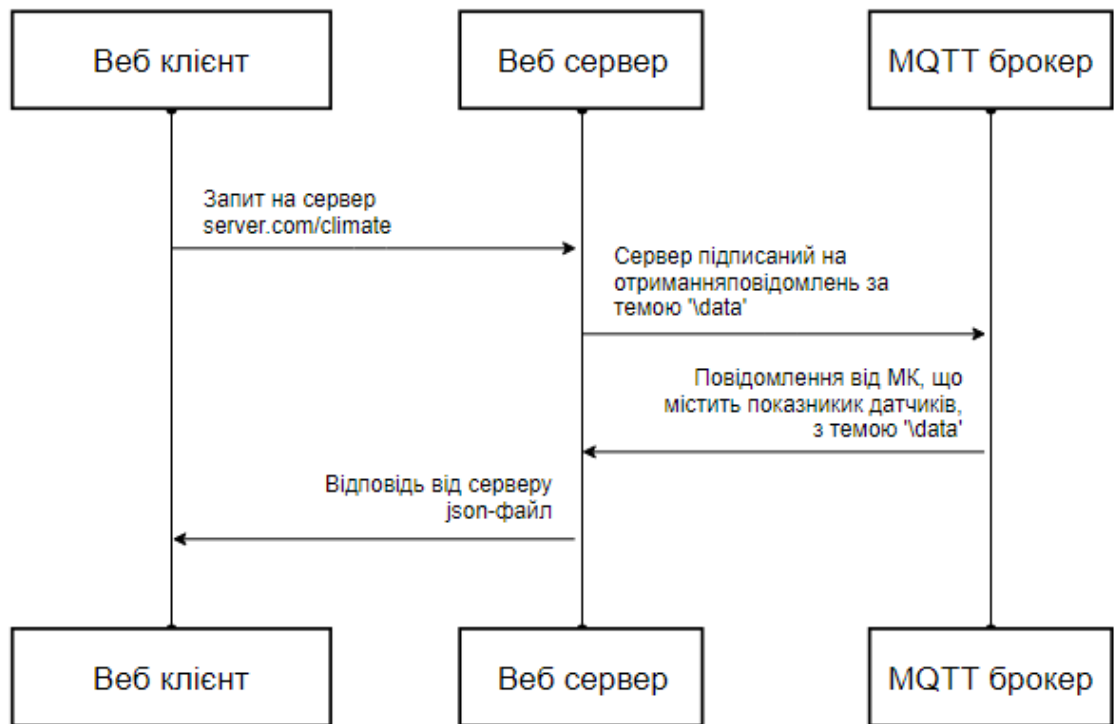


Рисунок 4.1 — Схема відправлення запитів від клієнту до серверу

#### 4.2. Підключення МК до MQTT серверу та мережі Інтернет.

Для створення програмного забезпечення для ESP32, будемо використовувати програмне середовище розробки Arduino IDE, де можна вибрати потрібний пристрій для завантаження прошивки.

Для підключення до інтернету через мережу Wi-Fi, використаємо готові рішення для підключення до мережі Wi-Fi. Для підключення обов'язкові параметри це назва мережі та пароль до неї. Потрібно в блоці установок створити клієнт для Wi-Fi.

Для підключення до MQTT брокеру, створимо клієнта, використовуючи підключення через Wi-Fi. Для підключення обов'язкові параметри: адреса серверу, порт, якщо використовується приватне підключення - необхідно

задати ім'я користувача та пароль доступу. Також потрібно задати функцію оберненого виклику, що буде викликатися коли до клієнта прийде повідомлення, після цього можна підключитися до MQTT серверу. У функції оберненого виклику описуємо обробку даних, що отримали від серверу. Важливо дізнатися тему повідомлення, корисну інформацію, що було відправлено, і відповідно до отриманих даних виконувати контроль виконавчими блоками через МК. Тут використовуються блоки порівняння, або ж конструкції switch-case. Для відправки повідомлення на сервер, потрібно задати тему повідомлення, та дані які потрібно відправити. В нашому випадку ми відправляємо відразу всі отримані показники від датчиків. Блок-схема роботи приведена на рис. 5.2. Для детального ознайомлення з програмою див. додаток Б.

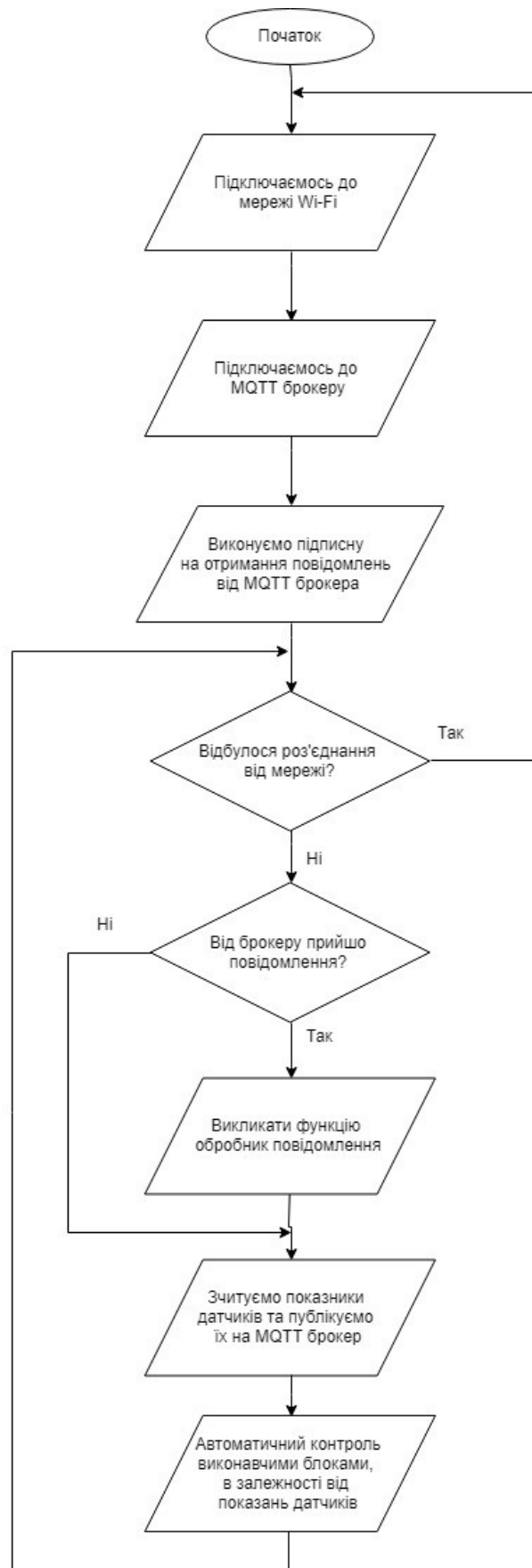


Рисунок 4.2 — Блок-схема роботи МК

**Висновки до розділу:**

1. На базі існуючих бібліотек та найкращих практик при роботі з протоколом MQTT, розроблено програмне забезпечення підключення МК до брокеру. Програма включає перевірку роз'єднання з мережею, підписку на отримання повідомлень від брокеру, відправку повідомлень, та автоматичні сценарії роботи, при недоступності мережі.
2. Розроблено відкриту документацію, для подальшого впровадження серверу до клієнтських застосунків.
3. На основі API серверу розроблено клієнтський веб додаток для користувачів

## 5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

### 5.1 Опис ідеї проекту

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Використання системи у побутових електричних мережах.	1. Будинок	Можливість моніторингу та керування електричної мережею в режимі реального часу. Корикування споживання електроенергії, для економії коштів.
	2. Розгортання	Використання зручног осервісу для перегляду споживчої електроенергії.

Таблиця 5.2 – Визначення характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Запропонований метод	NANSENSE	Moscar One			
1.	Пропозиція продажу	Дає змогу	Дає змогу	Дає змогу		+	
2.	Створення застосунку на мобільні платформи та вебу	Дає змогу	Не дає змогу	Дає змогу			+
3.	Вартість встановлення	100\$	200-1000\$	300-500\$			+

### 5.2 Технологічний аудит ідеї проекту.

У таблиці 5.3 показано оцінку технологічної здійсненності ідеї проекту та наведено технології, що можуть бути використані для реалізації проекту.

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Універсальне рішення для моніторингу та керування електричною мережею.	Спеціалізоване обладнання для виміру характеристик мережі	Наявна	Доступна
2		Застосування апаратних систем	Необхідно розробити	Доступна
3		Розробка власних апаратно-програмних рішень	Наявна	При обмеженому бюджеті недоступна

Обрана технологія реалізації ідеї проекту: застосування спеціалізованого обладнання для виміру характеристик мережі.

### 5.3. Аналіз ринкових можливостей запуску стартап-проекту

У таблиці 5.4 показано попередню характеристику потенційного ринку стартап-проекту.

Таблиця 5.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн/ум.од	500 000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Зацікавлення потенційних клієнтів
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	$500000/202000 = 266\%$

У таблиці 5.5 показано характеристику потенційних клієнтів стартап-проекту.

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Здешевлення процесу створення IoT систем	Будівельні компанії	Рівень очікування до зручності використання	Легкість використання, точність
2	Потребність моніторингу та керування в режимі реального часу	Будівельні компанії	Кожна з потенційних цільових груп має свої вимоги до мереж	Забезпечення моніторингу та керування в залежності від рівня потреб споживача

У табл. 5.6 показані фактори загроз реалізації стартап-проекту.

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Незацікавленість клієнтів	Внаслідок невдалого маркетингу клієнт може не зацікавитись послугами	Внесення додаткових сервісних послуг, демонстрація можливостей
2	Втрата конкуренції	Втрата рангу надійного поставника	Якісне та кількісне нарощування інтенсивності та грамотна цінова політика

У табл.5.7 показано фактори можливостей при реалізації стартап-проекту.

Таблиця 5.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Перехід до домінування на ринку вбудованих послуг	Зростання попиту	Якісне та кількісне нарощування потужностей
2	Імплементация технологій в існуючі системи	Зростання попиту внаслідок зростання клієнтів	Якісне та кількісне нарощування потужностей

У таблиці 5.8 визначено особливості конкурентного середовища та його вплив на впровадження проекту [17].

Таблиця 5.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Чиста конкуренція	Використання схожих технологій	Стандартизація на високому рівні
2. Локальний	Відсутність єдиного національного постачальника послуг	Окремий підхід до кожної локальної ділянки
3. Міжгалузєва	Відсутня	Відсутня
4. Товарно-видова	Застосування стандартизованих технологій	За необхідності, використання загальноживаних апаратних та програмних засобів
5. Цінова	Застосування спеціалізованих комплексів, які мають значну ціну	Можливість заощадити за допомогою застосування загальноживаних апаратних засобів
6. Марочна	Кожна діагностика має бути стандартизованою	Отримання переваги на ринку медійних послуг

У таблиці 5.9 показано аналіз конкуренції проекту в галузі за М. Портером

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Постачальники маркерних технологій	Необхідність пошуку постачальників	Залучення малопопулярних постачальників	Незалежність у прийнятті клієнтських рішень	Надання переваги більш авторитетним апаратним рішенням
Висновки:	Середня	Можливість виходу на ринок є	Постачальники диктують цінову політику на обладнання	Клієнти диктують вимоги до якості	Обмеження існують лише у разі відмови від діагностики

У табл. 5.10 показано фактори конкурентноспроможності та їх обґрунтування.

Таблиця 5.10. Обґрунтування факторів конкурентноспроможності

№ п/п	Фактор конкурентноспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Раціональніший ціновий показник	Можливість більш раціонально використати ресурси на покращення якості захоплення руху
2	Надання сервісних послуг	Сервісна підтримка програмної частини

У табл. 5.11 наведено сильні та слабкі сторони проекту.

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні						
			-3	-2	-1	0	+1	+2	+3
1	Раціональніший ціновий показник	17	+						
2	Надання сервісних послуг	12		+					
3	Періодична діагностика	4				+			
4	Необхідність залучення висококваліфікованих кадрів	7							+

У табл.5.12 наведено SWOT-аналіз стартап-проекту.

Таблиця 5.12. SWOT- аналіз стартап-проекту

Сильні сторони: раціональний ціновий показник, надання сервісних послуг	Слабкі сторони: надійність, можливості погрішностей при виміру характеристик мережі
Можливості: Перехід до ексклюзивного застосування нового методу, Імплементация методу в існуючі комплекси захоплення руху	Загрози: Незацікавленість клієнтів, втрата авторитету

Альтернативи ринкового впровадження стартап-проекту наведені у табл.5.13.

Таблиця 5.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Укладення договорів з будівельними компаніями та швидке захоплення ринку при використанні нового рішення	висока	незначні
2	Використання приладів загального вжитку для підвищення конкурентноспроможності	середня	незначні

Обрана альтернатива - укладення договорів з будівельними компаніями та швидке захоплення ринку при використанні нового рішення

#### 5.4. Розроблення ринкової стратегії проекту

Обґрунтування вибору цільових груп потенційних споживачів наведено у табл. 5.14 [17].

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Будівельні компанії	Середня	Високий	Середня	Висока
2	Ремонтні агенства	Висока	Високий	Середня	Низька

Визначення базової стратегії розвитку наведено у табл. 5.15.

Таблиця 5.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Використання альтернативних технологій та пристроїв	Встановлення нового стандарту якості	Зацікавлення та залучення гігантів у галузі телебачення та кіно	Стратегія диференціації
2	Дешевизна проекту	Раціональніші витрати на обладнання, та послуги	Застосування загальноживаних апаратних рішень замість спеціалізованих комплексів	Стратегія лідерства по витратах

Визначення базової стратегії конкурентної поведінки наведено у табл.5.16.

Таблиця 5.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки *
1	Ні	Забирати існуючих та шукати нових	Не буде	Стратегія виклику лідера

Визначення стратегії позиціонування наведено у табл. 5.17.

Таблиця 5.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока якість послуг	Стратегія диференціації	Новизна, гарант якості, точність дослідження	Якість, надійність, точність
2	Мінімальні витрати	Стратегія лідерства по витратах	Універсальність запропонованого рішення	Дешевизна, універсальність

### 5.5. Розроблення маркетингової програми стартап-проекту

Ключові переваги концепції потенційного товару наведено у табл. 5.18.

Таблиця 5.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість	Висока якість, надійність	Надійність
2	Дешевизна	Раціональне використання коштів, дешевше обладнання	Дешевизна

Визначено три рівні моделі товару. Сутність та складові рівнів товару, що включає якісні послуги, стандартизовану якість послуг та обладнання наведено у табл. 5.19.

Таблиця 5.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Якісні послуги, стандартизована якість послуг та обладнання		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Гл/Е/Ор
	1)Вартість обслуговування,	1) М	1)Е
	2)Кількість комплектів	2) М	2) Пр
	обладнання	3) М	3)Нд
	3)Строк безвідмовної праці	4) М	4)Тх
4)Технологічна собівартість товару			
	Якість: міжнародні стандарти якості, постійна підтримка обладнання		
	Доставка, встановлення та налаштування		
	Марка: Кіновиробництво		
III. Товар із підкріпленням	До продажу – обладнання, встановлення		
	Після продажу – сервісна підтримка		

За рахунок чого потенційний товар буде захищено від копіювання: специфічна методика захоплення та обробка даних.

Визначення меж встановлення ціни на послугу наведено у табл. 5.20.

Таблиця 5.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	250 у.о./од.	180 у. о./од	Високий	Н.500 у.о. – В.1000 у.о. (Товар) Н.200 у.о. – В.500 у.о. (Послуга)

Формування системи збуту послуги наведено у табл. 5.21.

Таблиця 5.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Орієнтована на отримання максимальної якості та точності захоплення руху	Поставки якісного, точного та надійного товару	Значна	Договірна система збуту

Концепції маркетингових комунікацій наведено у табл. 5.22.

Таблиця 5.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Зацікавленість в якісному та точному продукті з раціональним використанням ресурсів	Медіа ресурси	Гарантованість якості та стандартизація, політика сервісності	Зацікавити у покращеннях пов'язаних із зростаючою популярністю послуг	Представлення центру створення реалістично рухомих тривимірних моделей
2	Зацікавленість у великій кількості продукту із дотриманням умов якості	Медіа ресурси	Глибина каналу постачальників, гарант якості	Зацікавити у позитивних сторонах первісності та в глибині каналу постачання	Представлення центру створення реалістично рухомих 3D та 2D моделей

### Висновки до розділу

1. Комерціалізацію стартап-проекту щодо розвитку та впровадження запропонованого апаратно-програмного рішення для створення системи моніторингу та керування розумним будинком, можна вважати доцільною. На дану пропозицію на ринку присутній попит, наразі він задовольняється товарами заміниками та більш дорогими рішеннями, саме тому важливо зайняти нішу конкурента у якості поставника вигідного продукту, порівнюючи з конкурентами. Рентабельність на ринку послуг насамперед обумовлена заміною повної апаратної залежності на універсальність, що обумовлена використанням не спеціалізованих комплексів, а загальнозживаного програмного та апаратного забезпечення.

2. Впровадження є перспективним, адже основними групами клієнтів є

масштабні будівельні та ремонтні компанії, і після набуття достатньої авторитетності можливе охоплення у масштабах міжнаціональних ринків. Конкурентноспроможність проекту обумовлена меншою ціною на повний продукт та високою якістю в умовах, коли конкуренти за цим параметром у даних умовах програють. Це вигідно вирізняє запропоноване рішення, власне, і є основним критерієм входження на ринок.

3. Альтернативою впровадження було обрано – пошук альтернативних технологій та пристроїв для побудови системи. Імплементация проекту доцільна, оскільки рентабельність та зацікавленість потенційних груп клієнтів створює досить сприятливі умови для розвитку проекту.

## ВИСНОВКИ

У магістерській дисертації запропоновано варіант реалізації системи моніторингу та керування електричною мережею будинку, через веб застосунок. Реалізація зрозуміла для звичайних користувачів і може мати практичну цінність для побутових потреб, зокрема, для реалізації моніторингу споживчої потужності.

В рамках магістерської дисертації було проведено дослідження методів та технологій виміру характеристик мережі. На основі проведених досліджень отримано наступні результати:

1. Аналіз сучасних стандартів опису та протоколів передачі даних між елементами Інтернету речей показали, що доцільно використовувати протокол MQTT, для обміну даних між МК та сервером. Проаналізовані типові архітектури рішень на базі протоколу MQTT, підтверджують доцільність використання протоколу, для постійного відправлення однотипних даних, без підтвердження отримання.
2. На підставі недоліків та переваг використання основних технологій для вимірювання струму та напруги, обрано використання електронного ізольованого датчику для виміру напруги та датчику на основі ефекту Холла для вимірювання струму.
3. Розроблена структурна схема включає МК, датчики, індикацію, виконавчі блоки, та розділену архітекту додатку, що включає сервер, що має REST архітектуру та клієнт.
4. Розроблено відкриту документацію, для подальшого впровадження серверу до клієнтських застосунків.
5. Розроблено стартап-проект, який базується на просуванні на ринок системи розумного будинку, з використанням технологій виміру характеристик електричної мережі. Проведено дослідження доцільності та рентабельності даного бізнес-проекту та визначено, що комерціалізація проекту є доцільною.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Стаття «Комплексне дослідження результативності та економічної ефективності державної цільової економічної програми енергоефективності і розвитку»,  
URL: <http://saee.gov.ua/sites/default/files/Ocinka.pdf>
2. Lange, Kenneth. The Little Book on REST Services. 2016. — с. 11–14.
3. Стаття «Arduino: MQTT to Serial Bridge Using ESP32 »,  
URL: <http://icircuit.net/arduino-getting-started-mqtt-using-esp32/2138> (дата звернення 14.01.2019 р.)
4. Стаття «MQTT Protocol. Features, applications and basic procedures »,  
URL: [http://www.tssonline.ru/articles2/fix-corp/protokol-mqtt-osobennosti-varianty-primeneniya-osnovnye-protsedury-mqtt-protocol./](http://www.tssonline.ru/articles2/fix-corp/protokol-mqtt-osobennosti-varianty-primeneniya-osnovnye-protsedury-mqtt-protocol/) (дата звернення 14.01.2019 р.)
5. P. Saint-Andre, K. Smith, and R. Troncon. Xmpp: The definitive guide, 2009. — с. 11–14.
6. Emrah Ayanoglu, Yusuf Aytas, Dotan Nahum. Mastering RabbitMQ. — Packt Publishing, 2016. — с. 286.
7. Шестаков В. С., Сагидуллин А. С. Применение технологии websocket в web-приложениях технологического назначения, 2015. — с. 16–17.
8. Браїловський В. В., Рождественська М. Г. Багатоканальні системи передачі інформації, 2017. — с. 7-8.
9. Стаття «Стандарти бездротових стандартів Wi-Fi »,  
URL: <https://qpcs.ru/standarty-besprovodnyh-setei.html> (дата звернення 11.11.2018 р.)
10. Козлов А. Промышленные стандарты беспроводной передачи данных. Chip News Украина, 2008. — с. 18-21.
11. Стаття «Український інститут інтелектуальної власності»,  
URL: <http://base.uipv.org//search.php?action=viewdetails&IdClaim=117204> (дата звернення 23.12.2019 р.)
12. Стаття «Український інститут інтелектуальної власності»,

URL: <http://base.uipv.org//search.php?action=viewdetails&IdClaim=102335> (дата звернення 23.12.2019 р.)

13. Бессонов Л. А. Теоретические основы электротехники. — М: Высшая школа, 1984 — с. 45-47.
14. Schneier, Bruce. AC Power Theory - Arduino Maths, 2016. — 65 с.
15. Стаття «IoT: Mobile App Controlled Relays »,  
URL: <https://steemit.com/technology/@vyncl/mobile-app-controlled-relays-through-esp8266-via-mqtt-http> (дата звернення 14.01.2019 р.)
16. Fowler, Martin. Home Energy Monitoring System, 2014. — с. 12-20
17. J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 2000. — с. 49–53.

ДОДАТОК А  
ABSTRACT

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

The Internet of Things (IoT) is becoming more widely used technology nowadays. It is often used to refer to the growing network for connected devices, or “things”, that are capable of exchange data over on a low bandwidth network. IoT is being used in various areas, such as automotive industry, logistics, healthcare, smart grid and smart cities. Recently, electric energy consumption growth has risen significantly and thus, needed greatly increased energy supply in the coming decades due to increasing population and economic development. This is leading to a demand-supply deficiency. In many developed countries, automatic meter reading (AMR), advanced metering infrastructure (AMI) or smart energy meter with real-time energy information report have been implemented at the household level. Thus, consumers will be able to see their usage in real-time, eventually encouraging them to use less energy to save money. In addition, studies have suggested that more energy can be saved or decreased in household level with real-time energy consumption feedback as compared to conventional indirect feedback like monthly bills. However, those smart meters are usually high cost and require large amounts of investments on communication medium infrastructure; hence in many developing countries, these might not be an efficient and affordable solutions.

The day by day innovation improved from automatic machine to customer products. IoT is another pattern advancement that empowers us to screen and control hardware devices through the web. Here we propose to use IoT in order to screen and control home apparatuses, in this way computerizing present day homes through a web. This proposed framework enables a consumer to effortlessly control these home apparatuses through the web. The undertaking proposes a capable usage for IoT utilized for checking and controlling the home apparatuses by means of World Wide Web. Home robotization framework utilizes the reasonable gadgets as a UI. They can likewise speak with home computerization organize through an Internet access, by strategies for low power correspondence traditions like Zigbee, Wi-Fi, etc. This endeavor goes for controlling home machines by methods for Smartphone using Wi-Fi as correspondence tradition and

raspberry pi as server structure. The IoT based Monitoring and Controlling System for home is a progression which can control and screen gadgets not just for home mechanization but any real life appliances remotely. It provides facility to have control over a wide range of home appliances and ensure securities. In this project we presented implementation of home automation system through the Wi-Fi module, Massachusetts Institute of Technology (MIT) app. And Webpage server using ESP32. This project monitors all home appliances such as light, fan and controlled based on the threshold value programmed in the ESP 32. This system is low cost, allowing additional home appliances. Home monitoring and device controlling without involvement of humans the system can be control the devices, So that we can save the time and energy levels. So now we have introducing this system. The fig.1 shows the architecture of the system here we are using the four sensors and four devices for the home appliances for monitoring and controlling devices for the home respectively. The data can be sending to open source cloud storage for the store the data and use the MIT app Inventor for display the status of the sensors.

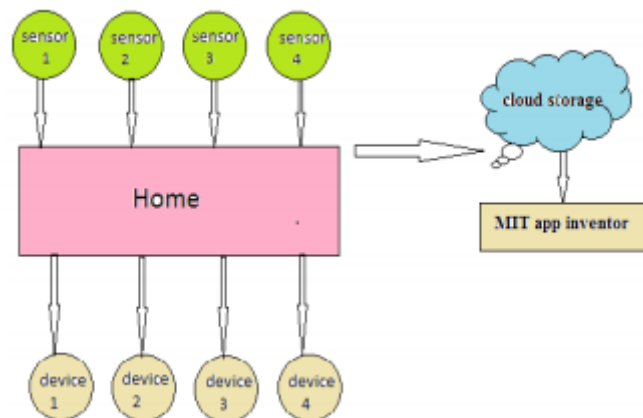


Figure 1. — Architecture of IoT based home monitoring and device control

In the literature different solutions proposed to this problem, but all of these involve complex circuitry and high-level knowledge required to operate these systems. But the proposed project does not require any profound knowledge of the hardware or software. By summarizing all the previous completed work, there are some drawbacks in them. To overcome those drawbacks, we are using an efficient way by employing esp32 and connecting them to cloud and retrieving the data to MIT app inventor. There are no

systems which exactly do the same work. But the present system easily to monitoring and controlling the home appliances and easily updates to open source cloud and then it automatically updates those same values to the MIT app inventor. The system consist of a four sensors temperature sensor, PIR sensor, gas sensor, water level sensor and using the ESP32 module. By using an IoT the ESP32 send the commands to cloud storage things speak. The cloud storage things speak it will receive the commands and shows the response. The system it is also using the MIT app inventor it shows the sensor status. When the internet is not available the system uses the buzzer it will give audio signaling when the water level reaches the threshold value and when the person entering into the home. The planned framework screen the sensor information, similar to temperature, gas, light, movement sensors, yet additionally activates a procedure as per the prerequisite, for instance exchanging on the globule when it gets dull. It additionally stores the sensor information in the cloud. It will assist the client with knowing the states of different parameters in the home at anytime anywhere. The proposed system block diagram as shown in fig.2. The system consists of two major parts: hardware implementation and software implementation.

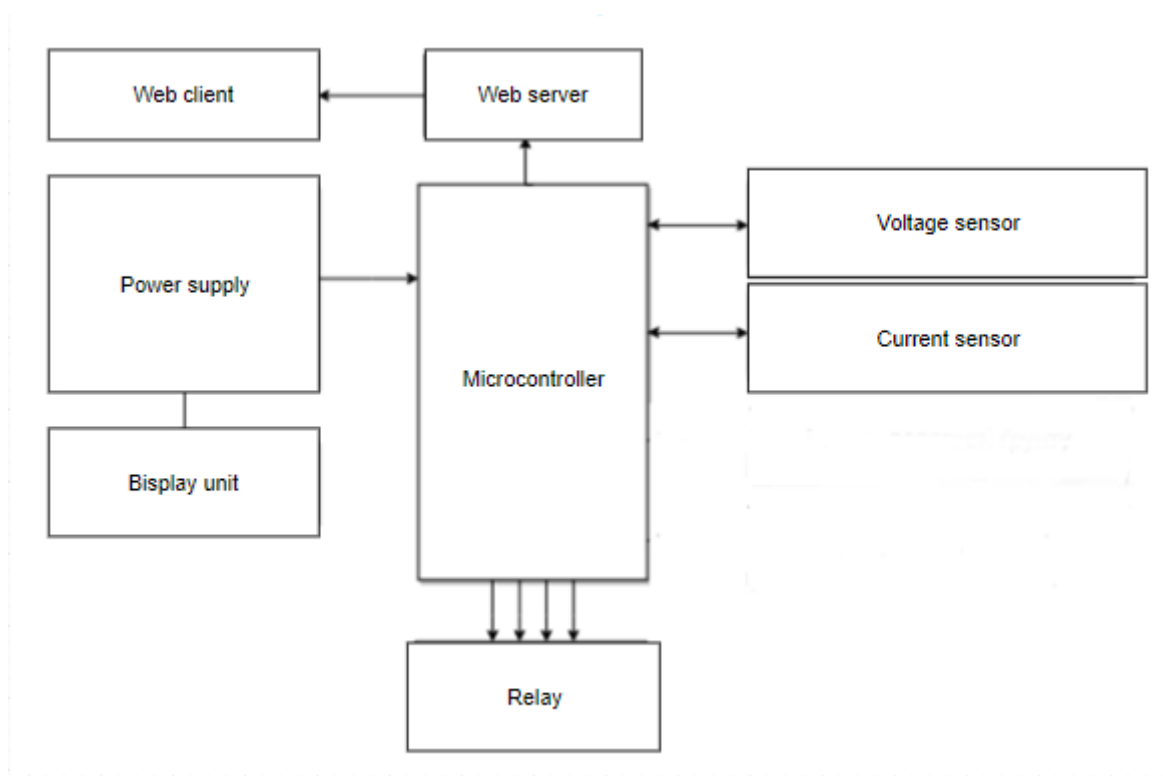


Figure 2. — Block diagram of the proposed system

There are different sensors are used in the block diagram they have similar operation but has different working principle. These sensors are individually connected to the esp32. we are using the 4-channel relay for connect the different devices. In this project we are using the open source cloud storage is connected to MIT app inventor. The sensor blocks record the values of like temperature, gas, PIR values and water level. The cloud storage things speak it will shows the response and values should be display on the MIT app inventor (dashboard). If there is a raise of the values of temperature detects these values and gives it to ESP32. This block continuously updates values to the cloud and then retrieved to operator dashboard. If the recorded value exceeds the predefined threshold level, then the operator dashboard is notified with an alert.

### Hardware Description

ESP32 board will serve as a microcontroller. It is a less-cost, little power system on a chip microcontroller with included Wi-Fi and dual mode Bluetooth. The ESP32 is the heart of the project. It is a microcontroller board used to connect all the sensors. The board is programmed with the source code in order to perform the operations of the project. The source code is stored in the on-chip memory available on the ESP32. This block can be considered as an interface between the programmer and the user. So, it is considered as the heart of the project. The ESP32 operating voltage range is 2.2 to 3.6V. Under normal operation the ESP32 thing will power the chip at 3.3V. The pin description of ESP32 is shown in fig.3.

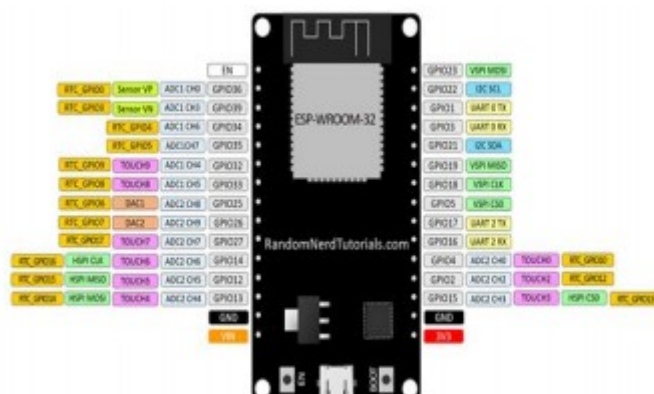


Figure 3. — Pin description of ESP32

Relay Module: In this project relay is an important component and which is employed to control high voltage home appliances. We employed 5V four channel relay to control four appliances. This is having four controlling inputs, Vcc, GND and four outputs. The outputs of relay are connected to four appliances and these are controlled by using control inputs which are connected to four digital output pins of ESP 32.

YHDC current transformer is suitability for use with the energy monitor system. The Yhdc current transformer is manufactured by Beijing YaoHuadechang Electronic Co., Ltd and is widely available from many stockists as Non-invasive AC current sensor (100A max), Model SCT-013-000. It has no internal burden resistor, but a transient voltage suppressor limits the output voltage in the event of accidental disconnection from the burden. It is capable of developing sufficient voltage to fully drive a 5 V input. For test currents up to 100 A, the CT primary consists of from 1 to 20 turns of insulated 16/0.2mm wire. The majority of tests were made at 5 A – thus the primary current seen by the CT could be adjusted in steps of 5 A by enclosing a variable number of turns inside the core. For saturation tests up to 250 A, the primary consists of 50 passes of enamelled copper wire, the current being adjusted in this case.

Since, the sensor data are constructed data type as MQTT messages that will be published to a self-hosted MQTT broker. Therefore, we wrote js scripts to subscribe to that MQTT topics and then stores all messages in time series data in the server database.

MQTT is a publish/subscribe protocol, which is very simple and lightweight messaging, designed for constrained devices and low-bandwidth, unreliable networks. It is a good solution for our design since it provides an easy communication between the server and many IoT nodes. The central server is so called a broker, and sensor nodes can subscribe to the topic and the topics are created automatically. It can also publish the data to topics of any kind of data. The broker then distributes the data to any node that has subscribed to that topic. The publishing can be done at 3 quality of service levels (QoS). In our setup, we use the Eclipse Mosquitto software that run as a broker on our server.

ДОДАТОК Б  
Лістинг програми

## Модуль для обчислення параметрів мережі

```

#define BLYNK_PRINT Serial
#define BLYNK_MAX_READBYTES 512
/*****/
#include "settings.h"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
// #include <SimpleTimer.h>
#ifdef OTA_UPDATES
#include <ArduinoOTA.h>
#endif
/*****/
BlynkTimer timer;
Adafruit_INA219 ina219;
float shuntvoltage = 0.00;
float busvoltage = 0.00;
float current_mA = 0.00, current_mA_Max;
float loadvoltage = 0.00, loadvoltageMax;
float energy = 0.00, energyPrice = 0.000, energyCost, energyPrevious, energyDifference;
float power = 0.00, powerMax, powerAvg;
int sendTimer, pollingTimer, priceTimer, graphTimer, autoRange, countdownResetCon,
countdownResetClock, counter2, secret, stopwatchTimer;
long stopwatch;
int splitTimer1, splitTimer2, splitTimer3, splitTimer4, splitTimer5;
int sendTimer1, sendTimer2, sendTimer3, sendTimer4, sendTimer5;
int loadvoltage_AVG_cycle = 0, current_AVG_cycle = 0, power_AVG_cycle = 0;
float loadvoltage_AVG[AVG_DEPTH_VOLTAGE + 1], current_AVG[AVG_DEPTH_CURRENT + 1],
power_AVG[AVG_DEPTH_POWER + 1];
float loadvoltage_AVG_total, current_AVG_total, power_AVG_total;
/*****/
void getINA219values() {

// get the INA219 values and throw some basic math at them
shuntvoltage = ina219.getShuntVoltage_mV();

```

```

busvoltage = ina219.getBusVoltage_V();
current_mA = ina219.getCurrent_mA();
loadvoltage = busvoltage + (shuntvoltage / 1000); // V
power = (current_mA / 1000) * loadvoltage * 1000; // mW
energy = energy + (power / 1000 / 1000);

// nothing connected? set all to 0, otherwise they float around 0.
if (loadvoltage < 1.1 && loadvoltage > 1 && current_mA < 2 && power < 2) {
  loadvoltage = 0;
  current_mA = 0;
  power = 0;
  energy = 0;
}

// gather voltage averages
loadvoltage_AVG[loadvoltage_AVG_cycle] = loadvoltage;
loadvoltage_AVG_cycle++;
if (loadvoltage_AVG_cycle == AVG_DEPTH_VOLTAGE) loadvoltage_AVG_cycle = 0;

// gather current averages
current_AVG[current_AVG_cycle] = current_mA;
current_AVG_cycle++;
if (current_AVG_cycle == AVG_DEPTH_CURRENT) current_AVG_cycle = 0;

// gather power averages
power_AVG[power_AVG_cycle] = power;
power_AVG_cycle++;
if (power_AVG_cycle == AVG_DEPTH_POWER) power_AVG_cycle = 0;
}

// this function is for updating the REAL TIME values and is on a timer
void sendINA219valuesREAL() {
  // VOLTAGE
  Blynk.virtualWrite(vPIN_VOLTAGE_REAL, String(loadvoltage, 4) + String(" V"));
  // POWER
  if (power > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_POWER_REAL, String((power / 1000), 3) + String(" W"));
  } else {
    Blynk.virtualWrite(vPIN_POWER_REAL, String(power, 3) + String(" mW"));
  }
}

```

```

// CURRENT (REAL TIME)
if (current_mA > 1000 && autoRange == 1) {
  Blynk.virtualWrite(vPIN_CURRENT_REAL, String((current_mA / 1000), 3) + String(" A"));
} else {
  Blynk.virtualWrite(vPIN_CURRENT_REAL, String(current_mA, 3) + String(" mA"));
}
}

// this function is for updaing the AVERGE values and is on a timer
void sendINA219valuesAVG() {
  // VOLTAGE
  for (int i = 0; i < (AVG_DEPTH_VOLTAGE - 1); i++) loadvoltage_AVG_total += loadvoltage_AVG[i];
  loadvoltage_AVG_total = loadvoltage_AVG_total / AVG_DEPTH_VOLTAGE;
  Blynk.virtualWrite(vPIN_VOLTAGE_AVG, String(loadvoltage_AVG_total, 3) + String(" V"));

  // CURRENT
  for (int i = 0; i < (AVG_DEPTH_CURRENT - 1); i++) current_AVG_total += current_AVG[i];
  current_AVG_total = current_AVG_total / AVG_DEPTH_CURRENT;
  if (current_AVG_total > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_CURRENT_AVG, String((current_AVG_total / 1000), 2) + String(" A"));
  } else {
    Blynk.virtualWrite(vPIN_CURRENT_AVG, String(current_AVG_total, 2) + String(" mA"));
  }

  // POWER
  for (int i = 0; i < (AVG_DEPTH_POWER - 1); i++) power_AVG_total += power_AVG[i];
  power_AVG_total = power_AVG_total / AVG_DEPTH_POWER;
  if (power_AVG_total > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_POWER_AVG, String((power_AVG_total / 1000), 2) + String(" W"));
  } else {
    Blynk.virtualWrite(vPIN_POWER_AVG, String(power_AVG_total, 2) + String(" mW"));
  }
}

// this function is for updaing the MAX values and is on a timer
void sendINA219valuesMAX() {
  // LOAD VOLTAGE (HIGH)
  if (loadvoltage > loadvoltageMax) {
    loadvoltageMax = loadvoltage;
    Blynk.virtualWrite(vPIN_VOLTAGE_PEAK, String(loadvoltageMax, 3) + String(" V"));
  }
}

```

```

}
// LOAD CURRENT (HIGH)
if(current_mA > current_mA_Max) {
  current_mA_Max = current_mA;
  if(current_mA_Max > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_CURRENT_PEAK, String((current_mA_Max / 1000), 2) + String(" A"));
  } else {
    Blynk.virtualWrite(vPIN_CURRENT_PEAK, String(current_mA_Max, 2) + String(" mA"));
  }
}
// LOAD POWER (HIGH)
if(power > powerMax) {
  powerMax = power;
  if(powerMax > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_POWER_PEAK, String((powerMax / 1000), 2) + String(" W"));
  } else {
    Blynk.virtualWrite(vPIN_POWER_PEAK, String(powerMax, 2) + String(" mW"));
  }
}
}

// this function is for updaing the ENERGY values and is on a timer
void sendINA219valuesENERGY() {
  energyDifference = energy - energyPrevious;
  // ENERGY CONSUMPTION
  if(energy > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_ENERGY_USED, String((energy / 1000), 5) + String(" kWh"));
  } else {
    Blynk.virtualWrite(vPIN_ENERGY_USED, String(energy, 5) + String(" mWh"));
  }
  energyPrevious = energy;
  // ENERGY COST
  energyCost = energyCost + ((energyPrice / 1000 / 100) * energyDifference);
  if(energyCost > 9.999) {
    Blynk.virtualWrite(vPIN_ENERGY_COST, String((energyCost), 7));
  } else {
    Blynk.virtualWrite(vPIN_ENERGY_COST, String((energyCost), 8));
  }
}
}

```

```

// HOLD BUTTON
BLYNK_WRITE(vPIN_BUTTON_HOLD) {
  if (param.asInt()) {
    timer.disable(sendTimer1);
    timer.disable(sendTimer2);
    timer.disable(sendTimer3);
    timer.disable(sendTimer4);
  } else {
    timer.enable(sendTimer1);
    timer.enable(sendTimer2);
    timer.enable(sendTimer3);
    timer.enable(sendTimer4);
  }
}

// this function only runs when in HOLD mode and select AUTO-RANGE. just shows latest values on
demand
void updateINA219eXtraValues() {
  Blynk.virtualWrite(vPIN_VOLTAGE_PEAK, String(loadvoltageMax, 3) + String(" V"));
  if (current_AVG_total > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_CURRENT_PEAK, String((current_AVG_total / 1000), 2) + String(" A"));
  } else {
    Blynk.virtualWrite(vPIN_CURRENT_PEAK, String(current_AVG_total, 2) + String(" mA"));
  }
  if (powerMax > 1000 && autoRange == 1) {
    Blynk.virtualWrite(vPIN_POWER_PEAK, String((powerMax / 1000), 2) + String(" W"));
  } else {
    Blynk.virtualWrite(vPIN_POWER_PEAK, String(powerMax, 2) + String(" mW"));
  }
}

// AUTO RANGE BUTTON
BLYNK_WRITE(vPIN_BUTTON_AUTORANGE) {
  autoRange = param.asInt();
  updateINA219eXtraValues();
}

// RESET AVERAGES (short) & RESET STOPWATCH (long)
BLYNK_WRITE(vPIN_BUTTON_RESET_AVG) {
  if (param.asInt()) {

```

```

Blynk.virtualWrite(vPIN_VOLTAGE_AVG, "--- V");
Blynk.virtualWrite(vPIN_CURRENT_AVG, "--- mA");
Blynk.virtualWrite(vPIN_POWER_AVG, "--- mW");
for (int i = 0; i < (AVG_DEPTH_VOLTAGE - 1); i++) loadvoltage_AVG[i] = loadvoltage;
for (int i = 0; i < (AVG_DEPTH_CURRENT - 1); i++) current_AVG[i] = current_mA;
for (int i = 0; i < (AVG_DEPTH_POWER - 1); i++) power_AVG[i] = power;
delay(50);
updateINA219eXtraValues();
countdownResetCon = timer.setTimeout(1000, []() {
  Blynk.virtualWrite(vPIN_ENERGY_USED, "0.00000 mWh");
  Blynk.virtualWrite(vPIN_ENERGY_COST, "0.000000");
  energy = 0;
  energyCost = 0;
  energyPrevious = 0;
});
} else {
  timer.disable(countdownResetCon);
}
}

// RESET PEAKS (short) & RESET CONSUMTION (long)
BLYNK_WRITE(vPIN_BUTTON_RESET_MAX) {
  if (param.asInt()) {
    Blynk.virtualWrite(vPIN_VOLTAGE_PEAK, "--- V");
    Blynk.virtualWrite(vPIN_CURRENT_PEAK, "--- mA");
    Blynk.virtualWrite(vPIN_POWER_PEAK, "--- mW");
    loadvoltageMax = loadvoltage;
    current_mA_Max = current_mA;
    powerMax = power;
    delay(50);
    updateINA219eXtraValues();
    countdownResetClock = timer.setTimeout(1000, []() {
      Blynk.virtualWrite(vPIN_ENERGY_TIME, "--:--:--");
      stopwatch = 0;
    });
  } else {
    timer.disable(countdownResetClock);
  }
}
}

```

```

// the stopwatch counter which is run on a timer
void stopwatchCounter() {
  stopwatch++;
  long days = 0, hours = 0, mins = 0, secs = 0;
  String secs_o = ":", mins_o = ":", hours_o = ":";
  secs = stopwatch; //convert milliseconds to seconds
  mins = secs / 60; //convert seconds to minutes
  hours = mins / 60; //convert minutes to hours
  days = hours / 24; //convert hours to days
  secs = secs - (mins * 60); //subtract the converted seconds to minutes in order to display 59 secs max
  mins = mins - (hours * 60); //subtract the converted minutes to hours in order to display 59 minutes max
  hours = hours - (days * 24); //subtract the converted hours to days in order to display 23 hours max
  if (secs < 10) secs_o = ":0";
  if (mins < 10) mins_o = ":0";
  if (hours < 10) hours_o = ":0";
  Blynk.virtualWrite(vPIN_ENERGY_TIME, days + hours_o + hours + mins_o + mins + secs_o + secs);
}

#ifdef FIXED_ENERGY_PRICE
BLYNK_WRITE(vPIN_ENERGY_API) {
  energyPrice = param.asFloat();
  Blynk.virtualWrite(vPIN_ENERGY_PRICE, String(energyPrice, 4) + String('c') );
}
#endif

/*****

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  // CONNECT TO BLYNK
#ifdef USE_LOCAL_SERVER
  Blynk.begin(AUTH, WIFI_SSID, WIFI_PASS, SERVER);
#else
  Blynk.begin(AUTH, WIFI_SSID, WIFI_PASS);
#endif
  while (Blynk.connect() == false) {}
#ifdef OTA_UPDATES
  // SETUP OVER THE AIR UPDATES
  ArduinoOTA.setHostname(OTA_HOSTNAME);

```

```

    ArduinoOTA.begin();
#endif

// START INA219
ina219.begin();

// TIMERS
pollingTimer = timer.setInterval(1000, getINA219values);
graphTimer = timer.setInterval(2000, []() {
    Blynk.virtualWrite(vPIN_GRAPH, current_mA);
});
stopwatchTimer = timer.setInterval(1000, stopwatchCounter);

// setup split-task timers so we dont overload ESP
// with too many virtualWrites per second
timer.setTimeout(200, []() {
    sendTimer1 = timer.setInterval(1000, sendINA219valuesREAL);
});
timer.setTimeout(400, []() {
    sendTimer2 = timer.setInterval(1000, sendINA219valuesAVG);
});
timer.setTimeout(600, []() {
    sendTimer3 = timer.setInterval(1000, sendINA219valuesMAX);
});
timer.setTimeout(800, []() {
    sendTimer4 = timer.setInterval(2000, sendINA219valuesENERGY);
});

// start in auto-range mode & sync widget to hardware state
autoRange = 1;
Blynk.virtualWrite(vPIN_BUTTON_AUTORANGE, 1);

// Check for fixed energy price and update global 'energyPrice'
#ifdef FIXED_ENERGY_PRICE
// else set fixed price with configured price
energyPrice = FIXED_ENERGY_PRICE;
Blynk.virtualWrite(vPIN_ENERGY_PRICE, String(FIXED_ENERGY_PRICE, 4) + String('c') );
#else
// No fixed price set, so pull from local API
Blynk.virtualWrite(vPIN_ENERGY_API, ENERGY_API);
priceTimer = timer.setInterval(20000, []() {

```

```

    Blynk.virtualWrite(vPIN_ENERGY_API, ENERGY_API);
  });
#endif
}
/*****/
void loop() {
  // the loop... dont touch or add to this!
  Blynk.run();
#ifdef OTA_UPDATES
  ArduinoOTA.handle();
#endif
  timer.run();
}

```

### Функція підключення до Wi-Fi

```

void connectToWiFi() {
  lcd.clear();
  lcd.setCursor(3, 0);
  lcd.print("WiFi... ");

  WiFi.mode(WIFI_STA);
  WiFi.setHostname("esp32-energy-monitor");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  // Only try 15 times to connect to the WiFi
  int retries = 0;
  while (WiFi.status() != WL_CONNECTED && retries < 15){
    delay(500);
    Serial.print(".");
    retries++;
  }

  // If we still couldn't connect to the WiFi, go to deep sleep for a
  // minute and try again.
  if(WiFi.status() != WL_CONNECTED){
    esp_sleep_enable_timer_wakeup(1 * 60L * 1000000L);
    esp_deep_sleep_start();
  }

  // If we get here, print the IP address on the LCD
  printIPAddress();
}

```

```
}
```

## Модуль для відправки повідомлень до брокера

```
#include <stdio.h>
#include <stdint.h>
#include <stddef.h>
#include <string.h>
#include "esp_wifi.h"
#include "esp_system.h"
#include "nvs_flash.h"

#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/semphr.h"
#include "freertos/queue.h"
#include "freertos/event_groups.h"

#include "lwip/sockets.h"
#include "lwip/dns.h"
#include "lwip/netdb.h"

#include "esp_log.h"
#include "mqtt_client.h"

static const char *TAG = "MQTTS_SAMPLE";

static EventGroupHandle_t wifi_event_group;
const static int CONNECTED_BIT = BIT0;

static void wifi_event_handler(void* arg, esp_event_base_t event_base,
                               int32_t event_id, void* event_data)
{
    switch (event_id) {
        case WIFI_EVENT_STA_START:
            esp_wifi_connect();
            break;
        case WIFI_EVENT_STA_DISCONNECTED:
```

```

    esp_wifi_connect();
    xEventGroupClearBits(wifi_event_group, CONNECTED_BIT);
    break;
default:
    break;
}
return;
}

```

```

static void ip_event_handler(void* arg, esp_event_base_t event_base,
                             int32_t event_id, void* event_data)
{
    switch (event_id) {
        case IP_EVENT_STA_GOT_IP:
            xEventGroupSetBits(wifi_event_group, CONNECTED_BIT);

            break;
        default:
            break;
    }
    return;
}

```

```

static void wifi_init(void)
{
    tcpip_adapter_init();
    wifi_event_group = xEventGroupCreate();
    ESP_ERROR_CHECK(esp_event_loop_create_default());
    ESP_ERROR_CHECK(esp_event_handler_register(WIFI_EVENT, ESP_EVENT_ANY_ID,
&wifi_event_handler, NULL));
    ESP_ERROR_CHECK(esp_event_handler_register(IP_EVENT, IP_EVENT_STA_GOT_IP,
&ip_event_handler, NULL));
}

```

```

wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
ESP_ERROR_CHECK(esp_wifi_init(&cfg));
ESP_ERROR_CHECK(esp_wifi_set_storage(WIFI_STORAGE_RAM));
wifi_config_t wifi_config = {
    .sta = {
        .ssid = CONFIG_WIFI_SSID,
        .password = CONFIG_WIFI_PASSWORD,
    }
}

```

```

    },
};
ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_STA));
ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config));
ESP_LOGI(TAG, "start the WIFI SSID:[%s] password:[%s]", CONFIG_WIFI_SSID, "*****");
ESP_ERROR_CHECK(esp_wifi_start());
ESP_LOGI(TAG, "Waiting for wifi");
xEventGroupWaitBits(wifi_event_group, CONNECTED_BIT, false, true, portMAX_DELAY);
}

static esp_err_t mqtt_event_handler(esp_mqtt_event_handle_t event)
{
    esp_mqtt_client_handle_t client = event->client;
    int msg_id;
    // your_context_t *context = event->context;
    switch (event->event_id) {
        case MQTT_EVENT_CONNECTED:
            ESP_LOGI(TAG, "MQTT_EVENT_CONNECTED");
            msg_id = esp_mqtt_client_subscribe(client, CONFIG_EMITTER_CHANNEL_KEY"/topic/", 0);
            ESP_LOGI(TAG, "sent subscribe successful, msg_id=%d", msg_id);

            break;
        case MQTT_EVENT_DISCONNECTED:
            ESP_LOGI(TAG, "MQTT_EVENT_DISCONNECTED");
            break;

        case MQTT_EVENT_SUBSCRIBED:
            ESP_LOGI(TAG, "MQTT_EVENT_SUBSCRIBED, msg_id=%d", event->msg_id);
            msg_id = esp_mqtt_client_publish(client, CONFIG_EMITTER_CHANNEL_KEY"/topic/", "data",
0, 0, 0);

            ESP_LOGI(TAG, "sent publish successful, msg_id=%d", msg_id);
            break;
        case MQTT_EVENT_UNSUBSCRIBED:
            ESP_LOGI(TAG, "MQTT_EVENT_UNSUBSCRIBED, msg_id=%d", event->msg_id);
            break;
        case MQTT_EVENT_PUBLISHED:
            ESP_LOGI(TAG, "MQTT_EVENT_PUBLISHED, msg_id=%d", event->msg_id);
            break;
        case MQTT_EVENT_DATA:
            ESP_LOGI(TAG, "MQTT_EVENT_DATA");

```

```

    printf("TOPIC=%.*s\r\n", event->topic_len, event->topic);
    printf("DATA=%.*s\r\n", event->data_len, event->data);
    break;
case MQTT_EVENT_ERROR:
    ESP_LOGI(TAG, "MQTT_EVENT_ERROR");
    break;
default:
    break;
}
return ESP_OK;
}

static void mqtt_app_start(void)
{
    const esp_mqtt_client_config_t mqtt_cfg = {
        .uri = "mqtts://api.emitter.io:443", // for mqtt over ssl
        // .uri = "mqtt://api.emitter.io:8080", //for mqtt over tcp
        // .uri = "ws://api.emitter.io:8080", //for mqtt over websocket
        // .uri = "wss://api.emitter.io:443", //for mqtt over websocket secure
        .event_handle = mqtt_event_handler,
    };

    ESP_LOGI(TAG, "[APP] Free memory: %d bytes", esp_get_free_heap_size());
    esp_mqtt_client_handle_t client = esp_mqtt_client_init(&mqtt_cfg);
    esp_mqtt_client_start(client);
}

void app_main()
{
    ESP_LOGI(TAG, "[APP] Startup..");
    ESP_LOGI(TAG, "[APP] Free memory: %d bytes", esp_get_free_heap_size());
    ESP_LOGI(TAG, "[APP] IDF version: %s", esp_get_idf_version());

    esp_log_level_set("*", ESP_LOG_INFO);
    esp_log_level_set("MQTT_CLIENT", ESP_LOG_VERBOSE);
    esp_log_level_set("TRANSPORT_TCP", ESP_LOG_VERBOSE);
    esp_log_level_set("TRANSPORT_SSL", ESP_LOG_VERBOSE);
    esp_log_level_set("TRANSPORT", ESP_LOG_VERBOSE);
    esp_log_level_set("OUTBOX", ESP_LOG_VERBOSE);
}

```

```
nvs_flash_init();  
wifi_init();  
mqtt_app_start();  
  
}
```