МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА ІНФОРМАТИКИ, ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

Конструювання та програмування IoT

пристрою на базі мікроконтроллеру ESP8266.

Розробка back-end частини та мобільного додатку

Курсова робота з програмування

| Виконавець: | студент 3 курсу 331 групи напряму підготовки 6.040302 «Інформатика» Воробйов Євгеній Андрійович |
| --- | --- |
| Науковий керівник: | кандидат фізико-математичних наук, доцент Валько Наталія Валеріївна |

Херсон — 2018

# ЗМІСТ

[**ЗМІСТ**](#_gyiowv3ruoh) **2**

[**РОЗДІЛ 1. ІНТЕРНЕТ РЕЧЕЙ**](#_48cvypj09fnc) **5**

[1.1 Концепція технології](#_azm2fdlzz9wy) 5

[1.2 Пов'язані технології](#_r6ugm2xyl1ai) 7

[1.3 Проблеми безпеки](#_cnj4olc76yrc) 11

[**РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА ПРОГРАМУВАННЯ   
IOT-ПРИСТРОЮ**](#_d4dxop5ygmz3) **12**

[2.1 Аналіз існуючих проектів](#_v02tvcyvqjv1) 12

[2.2 Вимоги до пристрою](#_e9lmqpl880q3) 13

[2.3 Апаратна реалізація пристрою](#_j1npsokuvyqe) 14

[2.4 Обробка стану приладу](#_75crmsg0bnas) 22

[**РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ТА   
BACK-END ЧАСТИНИ**](#_4ga2nrfh9g2f) **26**

[3.1 Back-end частина](#_644vs134nllz) 26

[3.2 Користувацький додаток](#_ywiokbgny8de) 33

[**ВИСНОВКИ**](#_e6txbjhf6pwb) **37**

[**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**](#_hermnj8ackl6) **38**

**ВСТУП**

З розвитком технологій передачі даних, їх розповсюдженням та ростом швидкості змінюється вага різних чинників у житті людини, зокрема збільшується вплив на соціум мережі Інтернет, яка стає невід'ємною частиною життя в розвинених країнах.

Окремо слід відзначити так званий «Інтернет речей» — мережу фізичних пристроїв, що можуть обмінюватися даними з комп’ютерними мережами за допомогою стандартних протоколів зв’язку. Елементи мережі можуть мати датчики та виконавчі пристрої, що дозволяє аналізувати і деяким чином змінювати оточуючий світ. Окремим прикладом впровадження і розвитку Інтернету речей є система «розумний будинок».

Актуальність дослідження полягає в необхідності подальшого розвитку інтегрованих систем розумного будинку, що забезпечує загальне покращення рівня життя населення.

Об’єкт дослідження — елементи мережі Інтернету речей. Предмет дослідження — розумна розетка з дистанційним керуванням.

Метою роботи є розробка розумної розетки та мобільного додатку як її основного інтерфейсу.

Для реалізації мети поставлено наступні завдання:

1. Проаналізувати характеристики існуючих прототипів розумних розеток, зокрема обсяг їх можливостей.
2. На основі проведеного аналізу розробити вимоги щодо пристрою, його програмного забезпечення, мобільного додатку та back-end частини, що забезпечує їх комунікацію.
3. Відповідно до створених вимог розробити прототип приладу та програмного забезпечення.
4. Сконструювати і запрограмувати прилад, розробити back-end частину та мобільний додаток.
5. Обґрунтувати використані технології при розробці приладу, написанні мобільного додатку та back-end частини.

Очікується, що розроблений продукт буде придатний до масового виробництва та використання в повсякденному житті.

Робота складається з трьох розділів, містить 28 рисунків та три таблиці.

# РОЗДІЛ 1. ІНТЕРНЕТ РЕЧЕЙ

## **1.1 Концепція технології**

### **1.1.1 Мережа фізичних пристроїв**

З розвитком мережі Інтернет вона поступово почала використовуватися у всіх сферах людського життя. Це дозволило значно зменшити необхідність людського втручання в окремі технологічні процеси, зокрема автоматизувавши суттєву частину поставок товарів.

Зі збільшенням обсягів поставок товарів і поштових пересилок між фізичними особами, різко збільшилася потреба у швидкій ідентифікації і відстеженні товарів. Одне із перших згадувань Інтернету речей стосувалось радіочастотної ідентифікації товарів з допомогою RFID-міток (radio frequency identification).

Для об’єднання пристроїв у мережу потрібно декілька технологій — система ідентифікації пристрою, система передачі даних в мережі, а також датчики та виконавчі пристрої для взаємодії з оточуючим світом.

### **1.1.2 Системи ідентифікації пристроїв у мережі**

Надзвичайно широкого поширення набули штрих-коди, QR-коди (рис. 1.1) та інші подібні способи відображення даних, що є зручними для зчитування машиною.

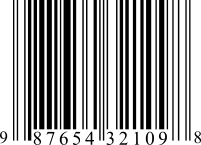
 

Рис. 1.1. Приклад штрих- та QR-коду

Інформація про предмет може зберігатися у RFID-мітках. Через свій малий розмір можуть використовуватися для ідентифікації тварин на фермах.

При використанні систем передачі даних мережі Інтернет, засоби ідентифікації закладені в протоколи передачі даних. Серед них MAC-адреса мереженої карти пристрою та IP-адреса пристрою в мережі.

### **1.1.3 Засоби передачі даних в мережі**

Для бездротової передачі даних особливо важливу роль в побудові Інтернету речей відіграють такі характеристики, як ефективність, відмовостійкість, адаптивність.

Найбільш розповсюдженими в сучасних пристроях є технології Wi-Fi (стандарт IEEE 802.11) та Bluetooth, що базуються на стандарті IEEE 802.15.4.

Wi-Fi — це сімейство локальних бездротових технологій, які використовують 2,4ГГц або 5ГГц радіохвилі. Дозволяють передавати великі обсяги даних по бездротовій мережі між пристроями, проте вимагає багато енергії для роботи, що обмежує використання в компактних автономних пристроях. Логотип Wi-Fi (рис. 1.2) є торговою маркою Wi-Fi Alliance.

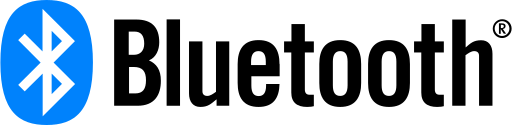


Рис. 1.2. Логотипи Wi-Fi та Bluetooth

Засоби для прийому і передачі даних технологією Wi-Fi вбудовані в більшість смартфонів, планшетів, ноутбуків та подібних мобільних пристроїв. Це дозволить у майбутньому об’єднати всі їх в єдину мережу, за принципом, наприклад, «розумного дому», навіть якщо зараз такого не передбачено виробником.

Bluetooth — це бездротова технологія, яка використовується для передачі даних в персональних мережах. Передає дані по смузі частот від 2,4 до 2,485 ГГц. В останніх версіях протоколу реалізовано можливість наднизького енергоспоживання і збільшений радіус дії до десяти чи до ста метрів (при збільшенні енергоспоживання).

Не зважаючи на те, що технологія Wi-Fi розрахована на передачу даних на невеликій відстані, розробляються концепції всесвітнього покриття доступом до мережі Інтернет з використанням системи орбітальних супутників [3].

## **1.2 Пов'язані технології**

### **1.2.1 Розумний дім**

Одразу після початку використання електричного струму в побутових приладах, почали розроблятися концепції віддаленого керування приладами. Одна з перших стосується патенту Ніколи Тесли на дистанційне керування судами та транспортними засобами в 1898 році [1].

Проте автоматизація будинків стала з’являтися лише в другій половину XX століття після поширення електрифікації. І тільки після появи мікроконтролерів було знято суттєве обмеження можливостей технологій та почався їх розвиток.

У 1995 році розробники технологій Java оголосили однією із цілей технології — «збільшення інтелекту побутових приладів». Програми, написані мовою програмування Java, можуть виконуватися для будь-якої апаратної платформи, для якої розроблене відповідне середовище [2].

Технології розумного дому дозволяють збирати домашню автоматику покомпонентно, обираючи ті функції розумного будинку, які дійсно потрібні користувачеві.

### **1.2.2 Штучний інтелект**

Штучний інтелект вивчає методи розв’язання завдань, які потребують людського розуміння, адже у більшості випадків алгоритм розв’язання завдання невідомий наперед. Робототехніка і штучний інтелект нерідко поєднуються одне з одним; їх об’єднання — створення інтелектуальних роботів можна вважати одним з напрямів штучного інтелекту.

При моделюванні біологічних систем з використанням штучного інтелекту, однією з вирішуваних задач є розпізнавання об’єктів. Безпосередньо з цим пов’язане комп’ютерне бачення, яке сприяє вирішенню серії задач розумного будинку з збільшення комфорту мешканців.

Створення зручнішої форми взаємодії комп’ютера та людини забезпечується розвитком обробки, аналізу та синтезу природної мови.

### **1.2.3 Оточуючий інтелект**

Інтернет речей є одним з прикладів втілення концепції оточуючого інтелекту — середовища, насиченого електронікою, яка реагує на дії людей.

Середовище має адаптуватися та реагувати на присутність людей та речей. Використання інформації має обмежуватися випадками коли це є доцільним, зберігаючи при цьому приватність і безпеку. Реакція має бути у вигляді ненав’язливої розумної допомоги.

Пристрої, що використовуються для створення оточуючого інтелекту мають бути малими, енергоефективними, мати малу вагу і, найголовніше, малу вартість. Вони постійно взаємодіють один з одним, система характеризується надлишковістю та стійкістю до помилок [4].

### **1.2.4 Комп'ютерний зір**

Комп’ютерний зір — технологія створення машин, які можуть проводити виявлення та класифікацію об’єктів. Також може описуватися як розширення або заміна біологічного зору. Ціллю комп’ютерного зору є формування корисних висновків відповідно об’єктів і сцен реального світу на основі аналізу зображень, отриманих з допомогою датчиків [5].

Комп’ютерний зір має широке застосування і може використовуватися як специфічна автономна система, так і входити до складу більших систем, зокрема використовуватися в розумному будинку.

Один з прикладів використання комп'ютерного зору в задачі забезпечення безпеки — ідентифікація людей з допомогою сканеру райдужної оболонки ока або відбитків пальців (рис. 1.3).



Рис 1.3. Райдужна оболонка ока людини та відбиток пальця на папері

### **1.2.5 Вимикач з димером**

Одним з засобів плавної регуляції освітлення або інших характеристик, котрі залежать від напруги, струму або потужності, що має застосування в системі розумного дому, є димер. Плавне ввімкнення освітлення збільшує комфорт мешканців, а його регуляція дозволяє економити електроенергію, уникаючи освітлення на повну потужність тоді, коли це не потрібно.

Зазвичай використовується як проста заміна ручного вимикача (рис. 1.4.а), проте це не обмежує можливості інтеграції аналогічного приладу в систему розумного дому.

### **1.2.6 Розетки з таймером**

Одним з шляхів часткової автоматизації циклічних процесів є використання електричних розеток з механічним або електронним таймером. Після попереднього задання часу ввімкнення та вимкнення шляхом встановлення перемикачів навколо механічного таймера (рис. 1.4.б) або введенням через мініатюрну клавіатуру в електронних розетках, розпочинається цикл роботи, що становить, переважно, одну добу або один тиждень.

Найчастіше використовуються для автоматизації освітлення при вирощуванні рослин (лампи вмикаються в розетку, завдяки якій задається оптимальна довжина світлового дня).

| Related image |  |
| --- | --- |
| а) | б) |
| Рис. 1.4. Приклади застосувань для регуляції електричних приладів | |

## **1.3 Проблеми безпеки**

### **1.3.1 Ботнети та DDoS-атаки**

При недодержанні вимог щодо безпеки, наприклад використання однакових логіну і паролю за замовчуванням, пристрої IoT можуть бути скомпрометованими та використовуватися зловмисниками.

Захоплені зловмисниками прилади, завдяки наявності доступу до мережі Інтернет, можуть використовуватися для генерації небажаного трафіку, наприклад, організовуючи DDoS-атаки (розподілені атаки на відмову в обслуговуванні).

# РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА ПРОГРАМУВАННЯ IOT-ПРИСТРОЮ

## **2.1 Аналіз існуючих проектів**

На підставі проведеного огляду існуючих проектів, визначено ключові характеристики та запропоновано наступну класифікацію розумних розеток:

1. за граничними параметрами електричного струму;
2. за протоколом доступу до мережі;
3. за наявністю таймерів;
4. за наявністю засобів вимірювання характеристик електричного струму;
5. за наявністю захисту;
6. за вартістю;
7. за іншими додатковими можливостями.

Керуючись запропонованою класифікацією, досліджено ознаки присутніх на ринку розумних розеток (табл. 1.1). В останній строчці таблиці наведено інформацію про розроблений прилад.

Таблиця 1.1. Огляд характеристик розумних розеток

| Назва | Граничні параметри струму | Доступ до мережі | Енерго- монітор | Вартість, грн |
| --- | --- | --- | --- | --- |
| Salus SPE600 | 16А | Wi-Fi, 2G/3G | Так | 750 |
| Ні | 680 |
| GS-1 | 260В, 16А | GSM | Ні | 800 |
| Orvibo Wiwo-S20 | 110-250В, 10А | Wi-Fi | Ні | 810 |
| SOKOL-GS4 | 20А | GSM | Так | 1150 |
| TP-LINK HS100 | 220В, 16А | Wi-Fi | Ні | 950 |
| Розроблений прилад | 100-240В, 10А | Wi-Fi | Так | 550\* |

Вартість розрахована з урахуванням роздрібної ціни на придбані компоненти, без урахування супутніх витрат (доставка, витратні матеріали тощо).

Всі розглянуті розетки виконані з використанням стандарту вилок та роз’ємів Shuko.

Крім можливості контролю власне електричного струму, можливо встановлення в корпус розетки інших датчиків, наприклад атмосферних характеристик (температура, вологість та тиск повітря).

Крім дистанційного контролю передбачена можливість прямого керування з допомогою кнопки та зворотній зв’язок одним або кількома кольоровими світлодіодами.

## **2.2 Вимоги до пристрою**

Керуючись поставленими завданнями, на підставі проведеного дослідження, поставлено наступні вимоги:

1. придатність вилки та роз’єму до використання з поширеними в Україні електричними роз’ємами;
2. робота з електричною мережею змінного струму 220В, 50Гц;
3. можливість підключення в розетку більшості побутових пристроїв (максимальна передбачена потужність не менше 2кВт);
4. живлення розетки від струму в мережі;
5. зв’язок з мережею Інтернет через технологію Wi-Fi;
6. можливість вимірювання параметрів електричного струму (напруга, сила струму, потужність);
7. можливість керування пристроєм як дистанційно, так і безпосередньо;
8. індикація режиму роботи розетки.

## **2.3 Апаратна реалізація пристрою**

### **2.3.1 Мікроконтролер ESP8266**

ESP8266 — мікроконтролер з інтерфейсом Wi-Fi. Окрім Wi-Fi, мікроконтролер надає можливістю виконувати програми з зовнішньої флеш-пам'яті з інтерфейсом SPI.

Для розробки проекту обрано плату NodeMCU (рис. 2.1) на базі мікроконтролеру ESP8266. Технічні характеристики плати наведено в табл. 2.1. Плата дозволяє оброблювати, вводити та виводити як дискретні, так і аналогові сигнали.

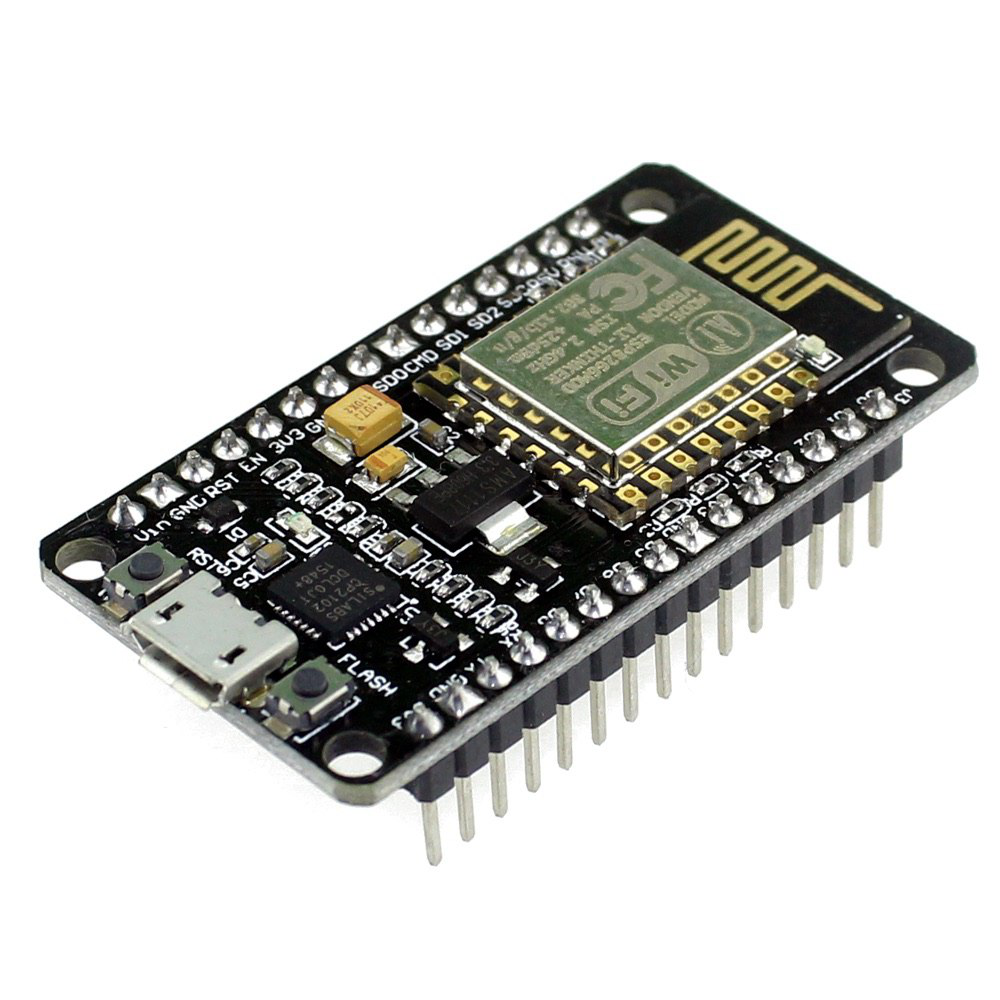


Рис. 2.1. Плата NodeMCU.

Програми можуть бути написаними як мовою програмування Lua, так і на мові, близькій до C++. Крім того, останнє дозволяє використовувати з мінімальними змінами Arduino IDE для розробки програм та їх завантаження до пам’яті плати через USB.

Таблиця 2.1. Технічні характеристики плати NodeMCU

| Стандарти обміну даними | Wi-Fi протокол 802.11 b/g/n |
| --- | --- |
| Режими Wi-Fi | Точка доступу, клієнт |
| Напруга живлення, В | 3,7–20 |
| Робоча напруга, В | 3–3,6 |
| Максимальний струм, мА | 220 |
| Робоча температура, °С | -40–125 |
| Розрядність процесора, біт | 32 |
| Частота процесора, МГц | 80 |
| Час пробудження, мс | 22 |
| Кількість дискретних входів/виходів | 10 |
| Кількість аналогових входів/виходів | 1 |

### **2.3.2 Перетворення напруги та стандарти роз’ємів живлення**

Стандартом CEE 7/4 визначено систему силових вилок та розеток для змінного струму Типу F (рис. 2.2), який називаються «Шу́ко»

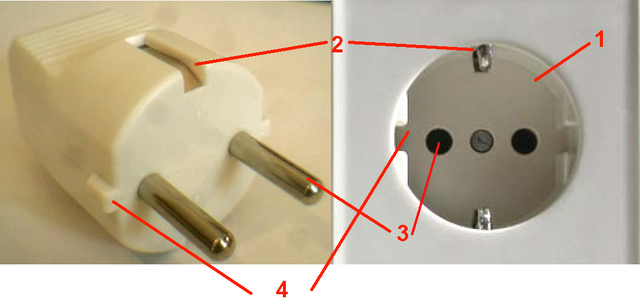
. 

Рис. 2.2. Вилка та роз’єм Shuko

Нерідко в пострадянських країнах їх помилково називають «євророзетками» та «євровилками» (рис. 2.2), можливо через те, що побутові прилади, котрі поставлялися з Німеччини мали таку вилку.



Рис. 2.3. Євровилка

Роз’єми Shuko призначені для використання в мережі 230В та мають номінальне значення сили струму 16А.

Для живлення електронних приладі необхідний постійний струм низької напруги (одиниці вольт). З цією метою використовується імпульсний блок живлення, що перетворює змінний струм у великому діапазоні (100–240В) в постійний напругою 5В.

### **2.3.3 Комутація електричного струму**

Для роз’єднання струму можна використовувати різні застосування, що мають різні характеристики.

В рамках поставлених вимог (змінний струм 230В, 50Гц), сила струму до 10А, обмежений простір всередині корпусу) оптимальним є використання реле.

В пристрій влаштовано модуль електромеханічного реле JQC-3FF-S-Z (рис. 2.4), що дозволяє розмикати струм порівняно меншим сигналом. Для власного живлення реле отримує напругу 3,3В з плати NodeMCU.

З метою забезпечення безпеки, реле підключено у «нормально-розімкнутому» режимі.

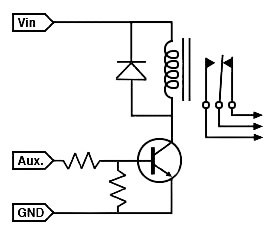


Рис. 2.4. Модуль електромеханічного реле та його спрощена схема

### **2.3.4 Вимірювання параметрів електричного струму**

Основними параметрами електричного струму є його частота, напруга та сила струму.

Частота змінного струму в побутовій електромережі складає 50Гц і, в нормі, відхилення не перевищує 0,5Гц. Сильніші відхилення частоти не допускаються і швидко компенсуються енергетичною мережею країни; тому для проведення розрахунків її можна вважати константною.

Напруга в електричній мережі становить 230В, при цьому відхилення не перевищує ±10%. Для вимірювання напруги між двома точками електричного кола використовуються вольтметри різних конструкцій; зважаючи на необхідність компактних розмірів всіх компонентів, відсутність необхідності у вимірюваннях високої точності а також на високу вимірювану напругу (у порівнянні з одиницями — десятками вольт, що використовуються для живлення більшості електронних компонентів), обрано модуль вольтметра ZMPT101B (рис. 2.5).

До пінів 1, 3–4 подається напруга з плати NodeMCU; до 5–6 — змінний струм, що вимірюватиметься; з піна 2 знімається аналоговий сигнал, що має функціональну залежність від вимірюваної напруги і передається на обробку в плату.

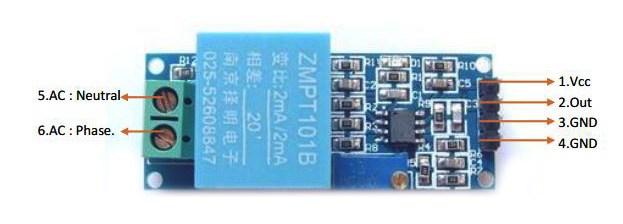


Рис. 2.5. Модуль вольтметра ZMPT101B

Сила струму, що проходить через ділянку струму, вимірюється амперметром. З метою забезпечення безпеки обрано метод вимірювання струму без розриву провідника; при цьому дріт проходить через основний компонент вимірюючої системи — котушку DL-CT1005A з великою кількістю витків тонкого дроту, утворюючи при цьому трансформатор. Змінний струм, що протікає через силовий провідник, наводить у котушці власний індуктивний струм, який після деяких перетворень подається (рис. 2.6) на аналоговий вхід плати і підлягає подальшій обробці.

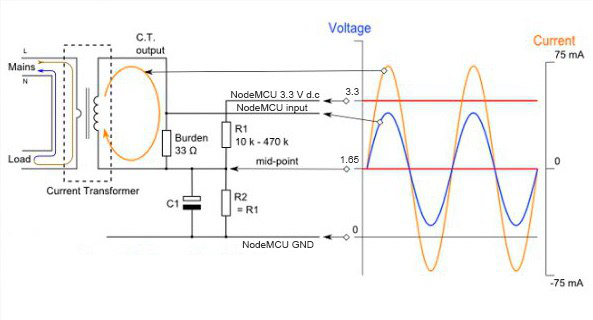


Рис. 2.6. Перетворення струму силового кола в аналоговий сигнал

### **2.3.5 Розширення кількості входів та виходів мікроконтролера**

У зв’язку з наявністю на платі NodeMCU тільки одного аналогового виходу з десятирозрядним АЦП, виникла необхідність у розширенні цього числа зовнішніми засобами. Для вирішення цієї задачі використано аналоговий мультиплексор CD4051BE (рис. 2.7), котрий дозволяє, використовуючи один аналоговий вхід/вихід та чотири дискретні виходи, отримати вісім дискретних входів/виходів та почергово читати/передавати дані.

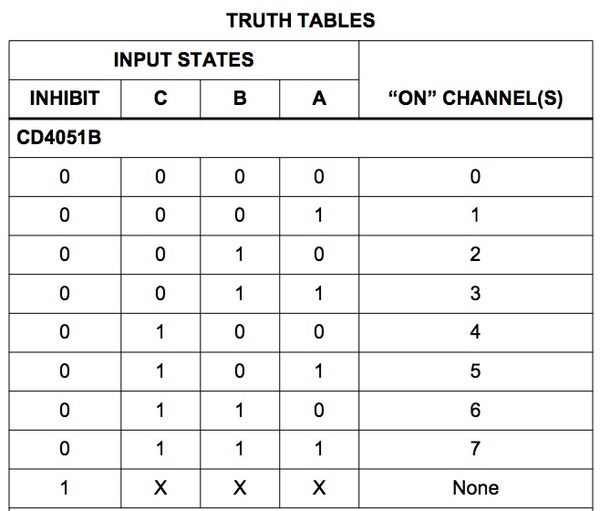
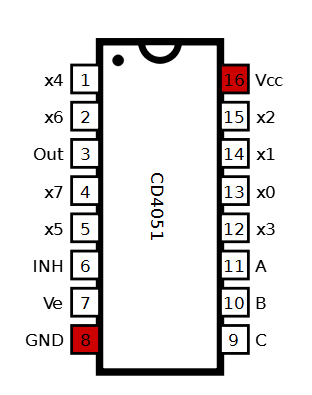
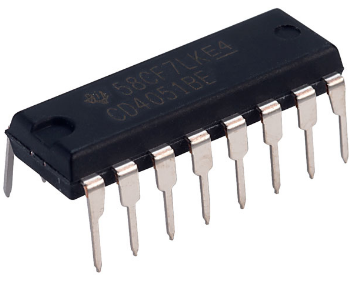


Рис. 2.7. Мультиплексор CD4051BE, його виходи та таблиця залежності його комутації від комбінації вхідних сигналів

Сам мультиплексор, а також частину схеми вимірювання струму компактно розміщено між монтажних ніг котушки DL-CT1005A; на них розведено схему живлення всіх компонентів розетки (рис. 2.8, 6).

### **2.3.6 Користувацький інтерфейс**

Задля керування розеткою без використання мобільного додатку а також швидкого візуального встановлення стану приладу, за «віконцем» на передній стінці корпусу до нього встановлено сенсорну кнопку (рис. 2.9) та RGB-світлодіод. Кнопки реагують на дотик навіть через товщу матеріалу (до декількох міліметрів).

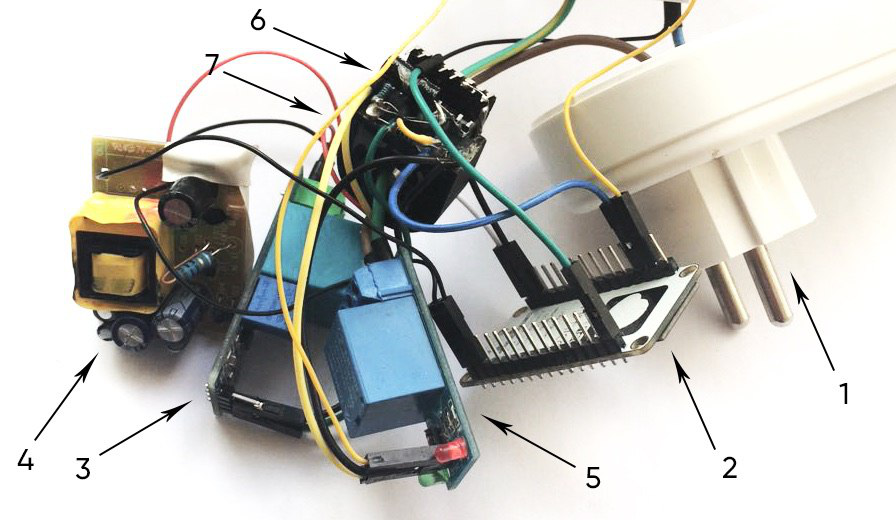


Рис. 2.8. Прилад з відкритим корпусом та його основні компоненти:

1 – корпус з електровилкою та роз’ємом;

2 – плата NodeMCU;

3 – модуль вольтметру ZMPT101B;

4 – адаптер живлення приладу;

5 – модуль електромеханічного реле JQC-3FF-S-Z;

6 – мультиплексор CD4051BE та схема вимірювання струму;

7 – котушка трансформатора DL-CT1005A.

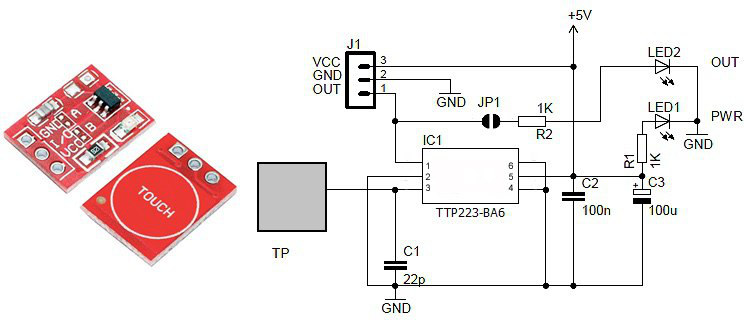


Рис. 2.9. Сенсорна кнопка та її принципова схема

### **2.3.7 Гранично допустимі параметри при використанні приладу та вимоги щодо техніки безпеки**

Температура оточуючого середовища та вологість при роботі приладу обмежуються використаними компонентами; їх перетином є температура -40–+85°С та відносна вологість повітря 40–85%.

Нижній поріг напруги обмежується вхідною напругою, за якої імпульсний перетворювач все ще може видавати струм для живлення компонентів розетки, що складає 100В. Верхній обмежується одночасно верхньою межею вимірювань вольтметра, граничною напругою, яку може комутувати реле та від якої може працювати імпульсний перетворювач, що складає 240В. Максимальний струм, що проходить через розетку, обмежено параметрами реле та налаштуванням схеми вимірювання струму; він складає 10А. Частота струму не має суттєвого впливу на імпульсний блок живлення та компенсується при розрахунку характеристик струму.

Таким чином, найбільша потужність споживача, що може використовуватися з розеткою, складає 2,4кВт.

Зведені характеристики розумної розетки наведено у таблиці 2.2.

Таблиця 2.2. Технічні характеристики розумної розетки

| Стандарти обміну даними | Wi-Fi протокол 802.11 b/g/n |
| --- | --- |
| Напруга мережі, В | 100–240 |
| Максимальна сила струму, А | 10 |
| Частота струму, Гц | 50 |
| Максимальна потужність споживача, Вт | 2400 |
| Робоча температура, °С | -40–+85 |
| Вологість повітря, % | 40–85 |

## **2.4 Обробка стану приладу**

### **2.4.1 Зчитування та обробка виміряних параметрів струму**

В процесі роботи прилад постійно визначає та передає на сервер напругу у мережі та струм, що проходить через нього. Для проведення розрахунків використовується бібліотека з відкритим вихідним кодом EmonLib (Energy monitor library) [6].

Перед початком роботи пристрою створюється об’єкт класу EnergyMonitor, методами voltage та current котрого встановлюються деякі параметри роботи, зокрема піни, до яких підключені відповідні датчики, фазовий зсув та константи калібрування датчиків струму та напруги.

Після проведення розрахунків процедури заносять основні (напруга та сила струму) та деякі додаткові, зокрема потужність, параметри в публічні поля об’єкту, звідки вони використовуються для вирішення поставлених завдань. Так як потужність обраховується як добуток напруги на силу струму, достатньо зберегти будь-які два з цих трьох значень для кожного з вимірювань; третє встановлюватиметься тривіально.

### **2.4.2 Зворотній зв’язок з користувачем**

При отриманні сигналу з сенсорної кнопки в момент натискання користувачем, отриманий сигнал фіксується і відсилається запит на сервер; у відповідь клієнту повертається команда про прийняття нового стану (рис. 2.10).

Одночасно зі зміною стану реле відбувається передача відповідного сигналу на світлодіод.

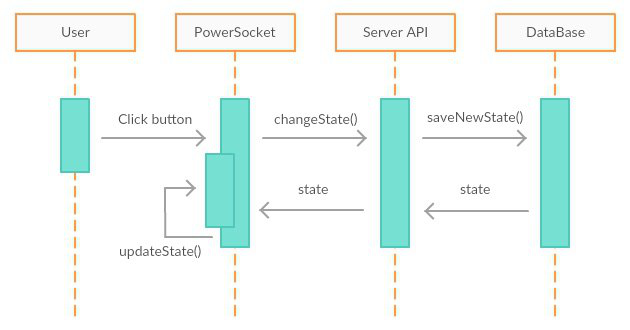


Рис. 2.10. Клієнт-серверна взаємодія при натисканні кнопки

### **2.4.3 Обмін даними з сервером**

В процесі роботи прилад обмінюється з сервером даними через невеликий постійний, проміжок часу, виступаючи при цьому в ролі клієнта. У відповідь на http-запит до сервера з ідентифікаційною інформацією про прилад, його поточний стан та показники датчика, сервером надсилається відповідь з командою про стан, який має прийняти прилад — ввімкнутись чи вимкнутись (рис. 2.11).

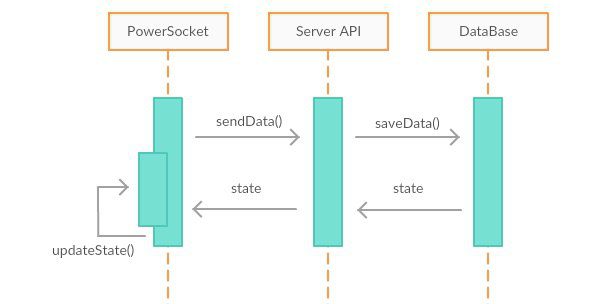


Рис. 2.11. Цикл обміну даних з сервером

### **2.4.4 Перемикання реле**

Як реакція на отриману команду, плата NodeMCU відповідним чином змінює стан електромеханічного реле. Після опрацювання даних, платою подається високий чи низький дискретний сигнальний рівень на командний вхід реле, котре, в свою чергу, розмикає чи замикає силову лінію.

### **2.4.5 Реєстрація приладу у мережі**

При першому ввімкнені, а також при необхідності внесення змін до даних про локальну Wi-Fi мережу (логіну і паролю для авторизацію), пристрій перезавантажується, а ESP8266 переходить в режим точки доступу. При цьому, користувач може під’єднатися до створеної пристроєм мережі з використанням вказаного на приладі паролю і вносити зміни в його налаштування через мобільний додаток (рис. 2.12).

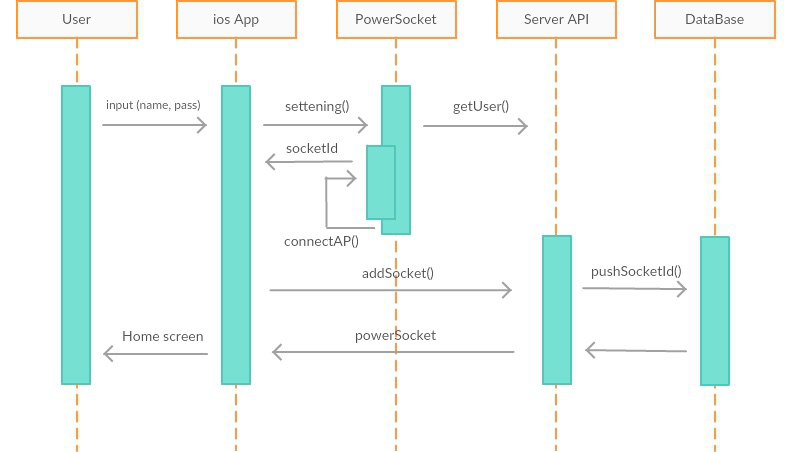


Рис. 2.12. Автентифікація розетки в локальній мережі

Після завершення цього етапу, отримані дані про локальну мережу оброблюються та записуються в EEPROM (енергонезалежну пам’ять) пристрою. Сам пристрій перезавантажується, переключаючись в нормальний режим, та під’єднується до вказаної мережі, входячи в цикл передачі обрахованих даних про струм та прийняття вказаного стану.

Одним із варіантів автентифікації приладу в мережі розглядалося використання раніше згаданого QR-коду (рис. 1.1), проте в реалізованій моделі поведінки це не вносить суттєвого спрощення в процес налаштування приладу.

### **2.4.6 Збереження даних в постійній пам’яті приладу**

Окремі дані розумної розетки, а саме її ідентифікаційна інформація та отримані дані для автентифікації в локальну мережу вносяться до EEPROM ESP8266 (рис. 2.13).

Ідентифікаційні дані вносяться до EEPROM одноразово при виготовленні приладу; одночасно створюється запис про прилад в базі даних.

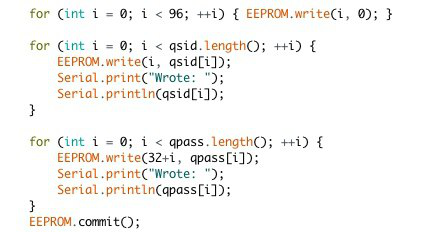


Рис. 2.13. Приклад запису даних в EEPROM

### **2.4.7 Обробка неочікуваних станів приладу**

До нестандартних ситуацій віднесено зникнення вказаної при налаштуванні локальної мережі або неможливість автентифікуватися в ній та вихід виміряних значень електричного струму за встановлені рамки.

Про відсутність підключення до мережі сигналізується встановленим сигнальним світлодіодом. Задача з обробки отриманої інформації про струм та прийняття тих чи інших рішень щодо роботи покладається на back-end частину.

# РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ТА BACK-END ЧАСТИНИ

## **3.1 Back-end частина**

### **3.1.1 Отримання запитів від клієнтів**

Для обміну інформацією між розумною розеткою, користувацьким додатком та back-end частиною використовується протокол передачі гіпертекстових даних HTTP. Передачу даних забезпечує стек транспортних протоколів TCP/IP.

Як основний протокол мережного рівня в стеку протоколів ТСР/IP використовується протокол IP, що створювався саме з метою передачі інформації в розподілених мережах. Перевагою протоколу IP є можливість його ефективної роботи в мережах зі складною топологією. При цьому протокол раціонально використовує пропускну здатність низькошвидкісних ліній зв’язку. В основі протоколу IP закладений дейтаграмний метод, що не гарантує доставку пакета, але «прагне» до цього.

Одним з способів побудови мереживих HTTP-додатків є використання асинхронного подієвого JavaScript–оточення Node. Нижче наведено приклад "hello world", який може одночасно обробляти багато з’єднань (рис. 3.1). 

Рис. 3.1. Приклад коду програми веб-серверу засобами NodeJS

У Node не має функцій, що працюють напряму з I/O, тому процес не блокується ніколи. Як результат, на Node легко розробляти масштабовані системи [7].

Node широко використовує подієву модель, він приймає цикл подій за основу оточення, замість того, щоб використовувати його в якості бібліотеки. В інших системах відбувається блокування виклику для запуску циклу подій.

Те, що Node спроектований без багатопоточності, не означає, що не можливо використовувати можливості кількох ядер у середовищі виконання. Наявна можливість створення дочірних процесів, якими легко керувати з допомогою API.

При розробці використано бібліотеку Express — гнучкий фреймворк для веб-застосунків, побудованих на Node.js, що надає широкий набір функціональності, надаючи множину допоміжних HTTP-методів та проміжних обробників, полегчуючи створення надійних API [8].

Express забезпечує тонкий прошарок базової функціональності для веб-застосунків, що не спотворює звичну та зручну функціональність Node.js.

Маршрутизація визначає: яку саме відповідь застосунок буде видавати клієнту, коли від клієнта йде запит з використанням певного HTTP-методу (GET, POST, і т.д.) та по конкретному URI.

Кожен маршрут може мати одну чи більше функцій-обробників, що виконуються, коли даний маршрут затверджено як співпадаючий.

При отриманні запиту він оброблюватиметься відповідно до визначення маршруту (рис. 3.2), де app є екземпляром express, METHOD є методом HTTP-запиту, PATH є шляхом на сервері, HANDLER є функцією-обробником, що спрацьовує, коли даний маршрут затверджено як співпадаючий.



Рис. 3.2. Структура визначення маршрутів

В прикладах на рис. 3.3 продемонстровано визначення простих маршрутів, що відповідають на GET і POST запити до головної сторінки сайту.



Рис. 3.3. Визначення GET і POST запитів

### **3.1.2 Структура бази даних та збереження обробленої інформації**

NoSQL — бази даних, які забезпечують механізм зберігання та видобування даних відмінний від підходу таблиць-відношень в реляційних базах даних.

Мотиви цього підходу включають: простоту дизайну схеми БД, значно спрощене горизонтальне масштабування на кластери машин, і тонкий контроль над доступністю. Структури даних, що використовуються в NoSQL є відмінними від тих, що використовуються за замовчуванням в реляційних базах, що робить тим самим деякі операції над даними значно швидшими на NoSQL. Точна відповідність використання NoSQL бази даних залежить від проблем, які вирішуються. Іноді структури даних, які використовуються в NoSQL базах можуть розглядатись як більш гнучкі ніж таблиці реляційних моделей [9].

При розробці проекту використано СУБД MongoDB. MongoDB це база даних, що зберігає дані як документи. Зазвичай ці документи мають JSON-подібну структуру. Документ розміщується усередину колекції. Наприклад, об'єкт user (рис. 3.4) може входити до колекції під назвою users.



Рис. 3.4. Зразок об’єкту в нотації MongoDB

Одна із головних особливостей MongoDB - гнучкість структури ії даних. Незважаючи на те, що в першому прикладі об'єкт user мав властивості firstName та lastName, ці властивості можуть бути відсутніми в інших документах user колекції users. Саме це відрізняє MongoDB від баз даних SQL, наприклад, MySQL чи Microsoft SQL Server, в яких для кожного об'єкту, що зберігається в базі даних, необхідна фіксована схема.

У зв’язку з цим використано Mongoose — об'єктно-документний відображувач. Mongoose дозволяє визначати об'єкти зі строго-типізованою схемою, що відповідає документу MongoDB. Наразі Mongoose має вісім типів даних схеми: String, Number, Date, Buffer, Boolean, Array, Mixed, ObjectId (унікальний ідентифікатор об'єкта, первинний ключ).

Для кожного типу даних можна:

1. зазначити значення за налаштуванням;
2. зазначити функцію користувача для перевірки даних;
3. зазначити, що поле необхідно заповнити;
4. зазначити get-функцію, яка дозволяє проводити маніпуляції над даними до їх повернення у вигляді об'єкта;
5. зазначити set-функцію, яка дозволяє проводити маніпуляції над даними перед їх зберіганням до бази даних;
6. визначити індекси для більш швидкого отримання даних.

Окрім цих спільних можливостей для деяких типів даних також можна налаштувати особливості зберігання та отримання даних із бази даних. Зокрема, для типу даних String можна зазначити наступні додаткові опції:

1. конвертація даних до нижнього регістру;
2. конвертація даних до верхнього регістру;
3. обрізання даних перед зберіганням;
4. визначення регулярного виразу, який дозволяє в процесі перевірки даних обмежити дозволені для зберігання варіанти даних;
5. визначення переліку, який дозволяє установити список припустимих рядків.

Для властивостей типу Number і Date можна зазначити мінімально та максимально допустиме значення.

Тип даних ObjectId використовується для визначення посилання на інший документ у базі даних.

Тип даних Array дозволяє вам зберігати JavaScript-подібні масиви. Завдяки цьому типу даних ви можете виконувати над даними типові JavaScript операції над масивами, наприклад, push, pop, shift, slice і т.д.

В базі даних зберігається дві моделі даних (рис. 3.5) — користувачі та розетки. При цьому користувач зберігає інформацію про розетки, до яких він має доступ, а розетка — про вимірювання, зроблені нею за весь час роботи.



Рис. 3.5. Схеми моделей даних, що використовуються проектом

### **3.1.3 Реєстрація та авторизація клієнтів**

Розетки вносяться до бази на етапі виготовлення. Користувачі реєструються в системі після завантаження мобільного додатку через його інтерфейс. Для забезпечення конфіденційності при обміні даними використовується JSON Web Token. Роути, що оброблюють реєстраційні та авторизаційні запити, представлено на рис. 3.7.

JSON Web Token це стандарт токена доступу на основі JSON, стандартизованого в RFC 75[19](https://tools.ietf.org/html/rfc7519). Використовується для верифікації тверджень. JSON Web Token складається з трьох частин: заголовка, вмісту і підпису.

За замовчуванням, заголовок містить тільки тип маркера і алгоритм, що використовується для шифрування. Заголовок кодується в base64.

В корисному навантаженні зберігається будь-яка інформація, яку потрібно перевірити. Кожен ключ в корисному навантаженні відомий як «заява». Наприклад, розглянемо соціальну мережу, в якій приєднатися до спільноти можна тільки за запрошенням. Коли ми хочемо запросити когось у співтовариство, ми відправляємо йому лист із запрошенням. Тут важливо перевірити, що адреса електронної пошти належить людині, яка приймає запрошення, тому ми включимо цю адресу в корисне навантаження, для цього збережемо його в ключі «email» (рис. 3.6).



Рис. 3.6. Приклад корисного навантаження JSONWeb Token

Як і заголовок, корисне навантаження кодується в base64.

Після отримання заголовку і корисного навантаження, можна обчислити підпис. Беруться закодовані в base64: заголовок і корисне навантаження, вони об'єднуються в рядок через точку. Потім цей рядок і секретний ключ надходить на вхід алгоритму шифрування, зазначеного в заголовку ключ «alg»). Ключем може бути будь-який рядок.

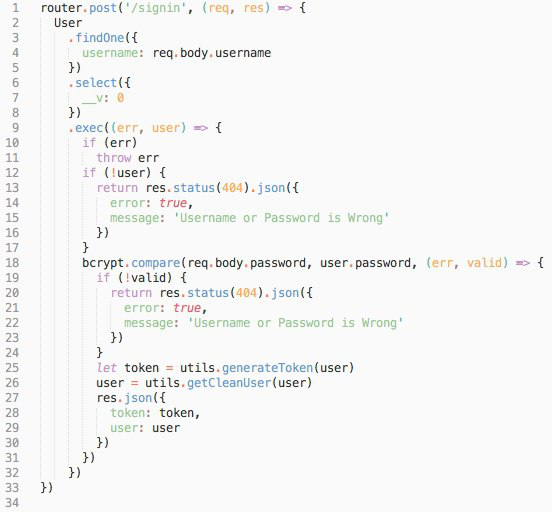


Рис. 3.7. Роути реєстрації та авторизації

### **3.1.4 Оцінка вимог до обладнання**

Back-end частина може виконуватися на будь-якому обладнанні, що має доступ до мережі Інтернет та встановлений стек TCP/IP, має статичну зовнішню IP-адресу, дозволяє приймати запити та передавати відповіді та дозволяє обробляти базу даних MongoDB.

Обрані моделі обробки та зберігання даних дозволяють маштабувати back-end частину як вертикально, так і горизонтально, тобто або збільшувати обчислювальну потужність обладнання, або збільшувати кількість одночасно працюючих розподілених систем.

Крім того, за наявності такої необхідності, можливе статичне розподілення окремих розумних розеток між різними групами серверів, кожен з яких оброблює свій пул розеток.

## **3.2 Користувацький додаток**

### **3.2.1 Вимоги до додатку та до системи виконання**

Мобільний додаток повинен виконуватися на платформі iOS. Необхідна можливість дистанційного налаштування приладу, керування ним та отримання статистичної і поточної інформації про його роботу.

### **3.2.2 Інтерфейс користувача**

При розробці мобільного додатку використано фреймфорк React Native, в основі якого знаходиться бібліотека React, призначена для створення користувацьких інтерфейсів. На відміну від React, призначеного для розробки односторінкових веб-додатків, React Native спрямований на мобільні платформи.

React — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків.

React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC) і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови веб-застосунків.

React підтримує віртуальний DOM, а не покладається виключно на DOM браузера. Це дозволяє бібліотеці визначити, які частини DOM змінилися, порівняно зі збереженою версією віртуального DOM, і таким чином визначити, як найефективніше оновити DOM браузера. Як бібліотека інтерфейсу користувача React часто використовується разом з іншими бібліотеками, такими як Redux, проте у його використанні при розробці проекту не було необхідності.

### **3.2.3 Реєстрація та авторизація в додатку**

Вхід користувача в систему (рис. 3.8) відбувається після завантаження і встановлення мобільного додатку та реєстрації в ньому.

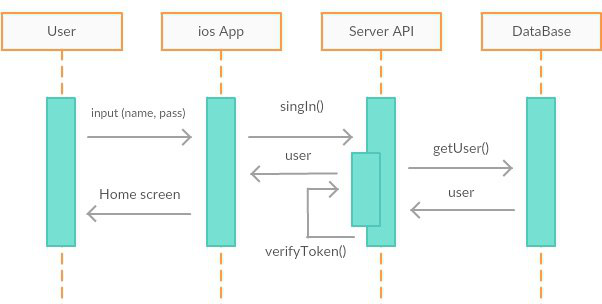


Рис. 3.8. Авторизація користувача в додатку

### **3.2.4 Авторизація приладу в локальній мережі**

Для зв’язку пристрою з сервером необхідне з’єднання з мережею Інтернет. Для надання розетці авторизаційної інформації, вона переводиться в режим точки доступу, після чого користувач під’єднується до створеної нею мережі, входить до мобільного додатку та вводить необхідну інформацію у відповідному вікні (рис. 3.9).

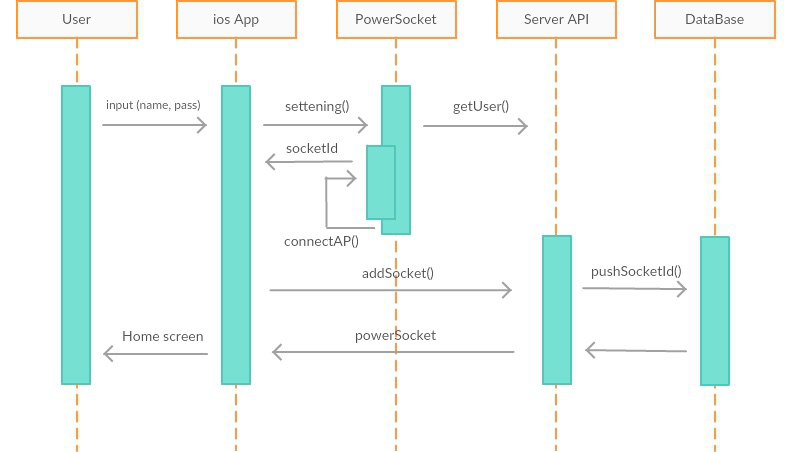


Рис. 3.9. Процес авторизації приладу в локальній мережі

### **3.2.5 Збереження даних на пристрої користувача**

Компонент AsyncStorage надає можливість постійно зберігати додатку пари key-value (ключ-значення) в пам’яті пристрою. В розробленому додатку компонент AsyncStorage використовується для збереження JSON Web Token авторизації. При роботі з даними необхідні наступні операції (рис. 3.10): збереження токена до сховища, його перезапис, видалення та отримання.



Рис. 3.10. Приклад роботи з AsyncStorage

### **3.2.6 Отримання та представлення користувачу статистичної інформації**

Основним наочним засобом представлення змінної інформації є діаграма (рис. 3.11)., що відображає функціональну залежність параметру від часу.

Крім того, діаграма може супроводжуватись статистичними характеристиками (середнє, найбільше і найменше значення, сума тощо), різними стилями оформлення, які користувач обирає в налаштуваннях на власний смак. Окремою функцією в додатку закладено можливість обрахування витрат електроенергії, спожитої через прилад, та її вартості.

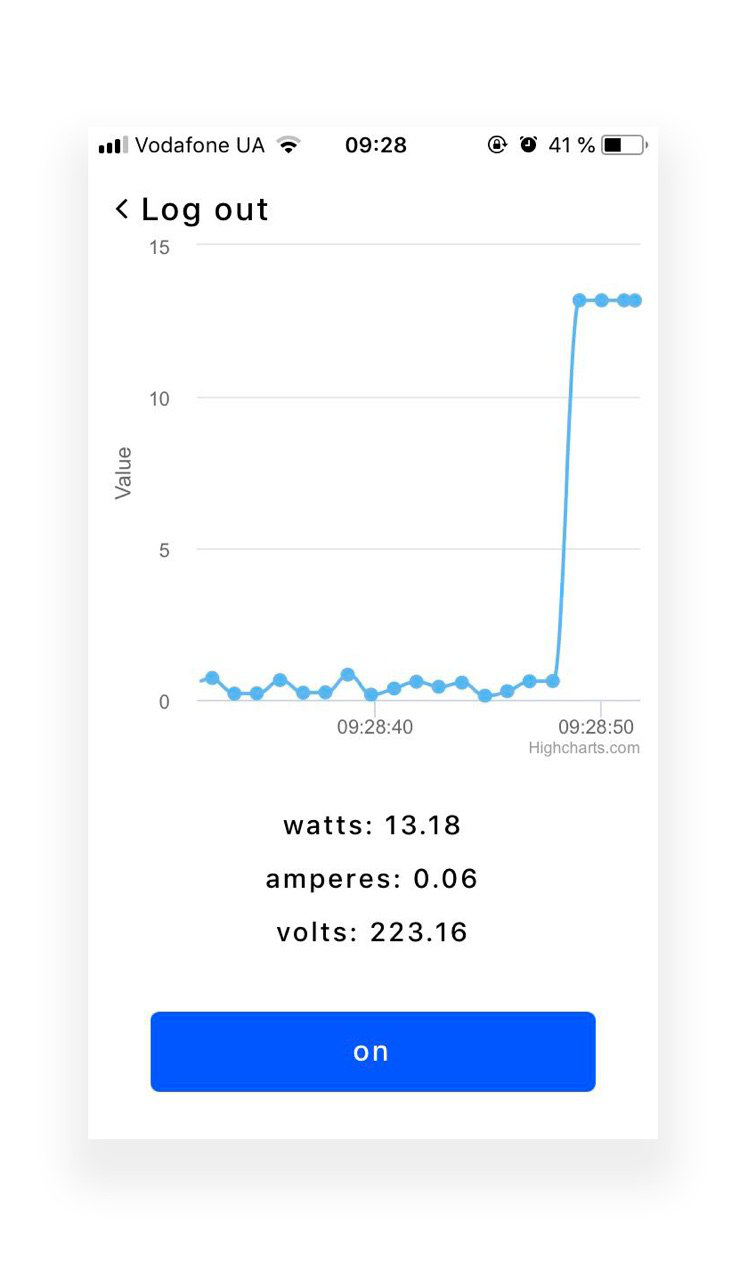


Рис. 3.11. Відображення статистики енергоспоживання

# ВИСНОВКИ

Для виконання поставлених завдань було проведено аналіз поточного розвитку технологій мережі Інтернет та розумного будинку.

1. Проаналізовано характеристики існуючих прототипів розумних розеток, зокрема обсяг їх можливостей. На підставі проведеного огляду, запропоновано класифікацію приладів за основними їх характеристиками, виділено ключові їх відмінності, особливості та наявні в переважній більшості досліджених екземплярів елементи.
2. На основі проведеного аналізу розроблено вимоги щодо пристрою, його програмного забезпечення, мобільного додатку та back-end частини. Керуючись вимогами до пристрою закладено основні компоненти та вимоги до них (розміри, діапазон вимірювань та робочих умов тощо).
3. Перед початком роботи досліджено аналоги обраних технологій та обґрунтовано їх використання при розробці приладу, написанні мобільного додатку та back-end частини.
4. Відповідно до створених вимог розроблено прототип приладу та програмного забезпечення. Після налагодження роботи окремих компонентів, проведено поєднання їх в єдину систему; оптимізовано комутацію силових та сигнальних ліній всередині пристрою.
5. Сконструйовано і запрограмовано прилад, розроблено дизайн мобільного додатку та реалізовано його с використанням сучасних мобільних технологій. Розроблено back-end частину, налагоджено клієнт-серверну взаємодію.

Розроблений пристрій та мобільний додаток для платформи iOS придатні для використання повсякденному житті.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Tesla Nikola Method of and apparatus for controlling mechanism of moving vessels and vehicles//Patent 613809. — United States Patent and Trademark Office, 8 November 1898
2. Жмудь А. В. Аналіз засобів хмарного керування розумним будинком / Жмудь Артем Володимирович – Київ, 2016. – 70 с.
3. Farooq K. Mobile Internet from the Heavens [Електронний ресурс] / Khan Farooq. – 2015. – Режим доступу до ресурсу: <https://arxiv.org/ftp/arxiv/papers/1508/1508.02383.pdf>.
4. Werner Weber, Jan M. Rabaey, Emile Aarts. Introduction // Ambient Intelligence / Werner Weber, Jan M. Rabaey, Emile Aarts. — Springer, 2005.
5. Л. Шапиро, Дж. Стокман. Компьютерное зрение = Computer Vision. — М. : Бином. Лаборатория знаний, 2006. — 752 с.
6. Hudson G. Arduino Energy Monitoring Library [Електронний ресурс] / Glyn Hudson. – 2017. – Режим доступу до ресурсу: <https://github.com/openenergymonitor/EmonLib>.
7. Zeiss M. Node.js v0.12.5 Manual & Documentation / Mirco Zeiss., 2015. – 242 с.
8. Brown T. Web Development with Node and Express, 2nd Edition / Todd Brown. – Sebastopol: O'Reilly Media, 2014. – 325 с.
9. Lith A. Investigating storage solutions for large data [Електронний ресурс] / A. Lith, J. Mattison. – 2010. – Режим доступу до ресурсу: <http://publications.lib.chalmers.se/records/fulltext/123839.pdf>.