Хмельницький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

SCADA система для малої генерації енергії
Назва теми

КвРАКІТ. 2020036.01.02.ПЗ

Галузь знань 15 «Автоматизація та приладобудування»
Шифр, назва

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»
Шифр, назва

Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»
Назва

Виконав:
студент 3 курсу, група АКІТ$_c$-20-1 _____ Владислав ДЕРУНЕЦЬ
Підпис Ім'я, ПРІЗВИЩЕ

Керівник:
д-р техн. наук, проф. _____ Валерій МАРТИНЮК
Підпис, дата Ім'я, ПРІЗВИЩЕ

Нормоконтролер _____ Людмила КОРЕЦЬКА
Підпис, дата Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
зав. кафедри автоматизації,
комп'ютерно-інтегрованих
технологій та робототехніки _____ Валерій МАРТИНЮК
Підпис, дата Ім'я, ПРІЗВИЩЕ

«19» червня 2023 р.

Хмельницький 2023

Хмельницький національний університет

Факультет *інформаційних технологій*
Кафедра *автоматизації, комп'ютерно-інтегрованих технологій та робототехніки*
Освітній рівень *перший (бакалаврський)*
Галузь знань *15 – Автоматизація та приладобудування*
Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітня-професійна програма *Автоматизація та комп'ютерно-інтегровані технології*

ЗАТВЕРДЖУЮ
Зав. кафедрою АКІТтаР
В. Мартинюк
«01» 02 2023р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дерунець Владислав Васильович

1 Тема роботи: *SCADA система для малої генерації енергі*
керівник роботи Мартинюк В.В., д.т.н, професор
Затверджено наказом по університету від «01» березня 2023р. №5.
2 Строк подання студентом роботи на кафедру:03.06.2023р.
3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування
4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Огляд літературних джерел та патентних даних. Основна частина. Розробка програмного забезпечення SCADA системи малої генерації енергії для сонячні електростанції.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень).
1. Схема електрична принципова плати розширення для аналогових давачів.
2. Схема електрична принципова плати розширення для цифрових давачів, яка містить 16 цифрових входів.
3. Схема електрична принципова плати годинника реального часу та ModBus інтерфейсу.

Завдання отримав _____

Науковий керівник _____

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Федула М.В., к.т.н., доцент каф. АКІТтаР		
Нормоконтроль	Корецька Л.О., к.т.н., доцент каф. АКІТтаР		

7. Дата видачі завдань «__01_»_____02_____2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів (розділів) дипломної роботи	Строк виконання етапів дипломної роботи	Примітка
1	Вступ	15.02.2023р.	Виконано
2	Огляд літературних джерел та патентних даних	15.03.2023р.	Виконано
3	Основна частина	10.04.2023р.	Виконано
4	Розробка програмного забезпечення SCADA системи малої генерації енергії для сонячні електростанції	10.05.2023р.	Виконано
5	Висновки	15.05.2023р.	Виконано
6	Оформлення пояснювальної записки до КРБ	25.05.2023р.	Виконано
7	Оформлення презентаційних матеріалів	1.06.2023р.	Виконано

Студент _____ Підпис В.В. Дерунець Ініціали, прізвище

Керівник роботи _____ Підпис В.В. Мартинюк Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «SCADA система для малої генерації енергії».

Автор роботи: Дерунець Владислав Васильович.

Керівник роботи: Мартинюк Валерій Володимирович

Пояснювальна записка: 65 с., 18 рис., 1 дод., 40 джерел.

Графічна частина: 3 креслення.

SCADA СИСТЕМА, МАЛА ГЕНЕРАЦІЯ ЕНЕРГІЇ, МОНІТОРИНГ ТА КОНТРОЛЬ, СИСТЕМИ НАКОПИЧЕННЯ ЕНЕРГІЇ.

Мета кваліфікаційної роботи – розробка SCADA системи для малої генерації енергії шляхом використання технології Інтернету речей. У роботі наведено основні технічні характеристики SCADA систем для малої генерації енергії, проаналізовано існуючі типи SCADA систем для малої генерації енергії та встановлено їх основні переваги та недоліки. Розроблено принцип роботи SCADA системи для малої генерації енергії. У SCADA системи малої генерації енергії для сонячної електростанції реалізовано API пристрою на основі MQTT, що підтримується протоколом зв'язку MQTT.

Підпис студента

19.06.2023

Дата

ЗМІСТ

Вступ 3
1. Огляд літературних джерел та патентних матеріалів 6
1.1. Особливості систем малої генерації електроенергії 6
1.2. Принципи побудови SCADA систем 9
1.3. Особливості SCADA систем з відкритим кодом 16
1.4. Висновки до першого розділу 18
2. Основна частина 20
2.1. Структурна схема системи малої генерації енергії 20
2.2. Розробка SCADA системи малої генерації енергії для сонячні електростанції 22
2.3. Висновки до другого розділу 32
3. Розробка програмного забезпечення SCADA системи малої генерації енергії для сонячні електростанції 34
3.1. Використання технології Інтернету речей (IoT) для SCADA системи малої генерації енергії для сонячні електростанції 34
3.2. Використання телеметричного транспортного протоколу повідомлень черги для SCADA системи малої генерації енергії для сонячні електростанції 37
3.3. Висновки до третього розділу 55
Висновки 56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 57
Додатки 63

ВСТУП

Актуальність теми. Зі збільшенням кількості невеликих систем відновлюваної генерації для забезпечення належної роботи та обслуговування необхідні низькі витрати на моніторинг і контроль. Малі системи відновлюваної електроенергії зазвичай розподіляються на великих географічних територіях.

Важливо мати надійне, своєчасне, гнучке, економічно ефективне та складне координоване рішення для моніторингу та контролю невеликих систем відновлюваної генерації.

Система SCADA є ідеальним рішенням для цього завдання. Однак доступні системи SCADA є значною мірою дорогими, а тому економічно невиправданими для невеликих програм. З комерційними системами SCADA також виникають проблеми сумісності та високого енергоспоживання, а також потреба у використанні дорогих стандартних систем зв'язку, оскільки ці компоненти зазвичай виготовляються різними виробниками та постачальниками.

Актуальність роботи обумовлена розробкою SCADA системи для малої генерації енергії для подолання їх головних недоліків, а саме: використовуючи доступні пакети SCADA з відкритим кодом.

Метою роботи є розробка SCADA системи для малої генерації енергії для моніторингу та контролю в реальному масштабі часу.

Для досягнення цієї мети необхідно розв'язати такі **завдання:**

- розробити структурну схему SCADA системи для малої генерації енергії;

- розробити алгоритм роботи SCADA системи для малої генерації енергії;

- розробити програму роботи SCADA системи для малої генерації енергії.

Об’єктом дослідження є процес моніторингу та контролю малої генерації електроенергії.

Предметом дослідження є SCADA системи для малої генерації електроенергії.

Методи досліджень. При вирішенні поставлених завдань у роботі були використані методи теорії керування, теорії кіл та методи алгоритмізації та програмування.

Практична значимість отриманих результатів:

У роботі розроблена SCADA системи для малої генерації електроенергії та програмне забезпечення її роботи. Використання такої системи дозволяє підвищити ефективність роботи моніторингу та контролю малої генерації електроенергії.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків до кожного розділу, висновків, списку використаних джерел, 2 додатків. Загальний обсяг роботи складає 65 сторінок комп’ютерного тексту, у тому числі: 18 рисунків, список використаних джерел вміщує 40 найменувань.

У вступі обґрунтована актуальність кваліфікаційної роботи, сформульовано мету та задачі кваліфікаційної роботи, відображено її практичне значення.

В першому розділі були розглянуті наступні питання: особливості систем малої генерації електроенергії, принципи побудови SCADA систем, Особливості SCADA систем з відкритим кодом.

У другому розділі проведено аналіз вимог технічного завдання та розроблена структурна схема системи малої генерації енергії.

Розроблено SCADA систему малої генерації енергії для сонячні електростанції.

Використано локальний сервер Thinger.IO, який отримує напругу PV, струм PV, виміряну потужність PV та напругу акумуляторної батареї від підключеного пристрою ESP32 у хмарній консолі та автоматично записує дані на створені інформаційні панелі та сегменти даних у задані швидкості передачі даних.

У третьому розділі розроблено програмне забезпечення SCADA системи малої генерації енергії для сонячні електростанції.

Використано технологію Інтернету речей (IoT) для SCADA системи малої генерації енергії для сонячні електростанції.

Реалізовано API пристрою на основі MQTT, що підтримується протоколом зв’язку MQTT для сонячної електростанції.

1 ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ ТА ПАТЕНТНИХ МАТЕРІАЛІВ

1.1 Особливості систем малої генерації електроенергії

Електрична енергія стала однією з основних потреб людини завдяки широкому використанню в різних сферах життя. У зв'язку прагненням людей до проживання в екологічно чистому середовищі, для виробництва електроенергії широко використовується відновлювана енергія та менше використовується викопне паливо.

Таким чином, більше відновлюваних джерел енергії, таких як сонячні фотоелектричні та вітряні турбіни, постійно вводяться в сучасні енергетичні системи для формування гібридних енергетичних систем і забезпечення електроенергією для людства. Очікується, що ця тенденція збережеться в найближчі десятиліття [1].

Однак збільшення частки відновлюваних джерел енергії може спричинити ряд проблем в енергосистемі. Щоб зменшити ці проблеми, системи зберігання енергії часто включаються в системи відновлюваної генерації. До проблем відновлюваних джерел енергії відносять наступні проблеми.

1. Труднощі в системному регулюванні частоти через коливання потужностей відновлюваних джерел. За домовленістю, регулювання частоти в основному керується здатністю змінювати потужність теплових генераторів, і це означає неефективну роботу, оскільки теплогенератори не працюють на повну потужність, коли використовуються для цілей регулювання частоти [1].

Це пояснюється тим, що самі генеруючі установки з відновлюваних джерел у більшості випадків забезпечують низьку

економічну ефективність, тому з більшим проникненням відновлюваної генерації відбувається подальше зниження ефективності теплових генераторів.

Системи накопичення енергії можуть пом’якшити ці коливання потужності в системах відновлюваної генерації, що призведе до зниження рентабельності теплових генераторів і, таким чином, призведе до підвищення ефективності їх роботи.

По-друге, на відновлювану енергетику впливають погодні умови, а оскільки погодні умови здебільшого непередбачувані, продуктивність є ненадійною. Це ускладнює підключення їх до мережі.

Для вирішення цих проблем доступні рішення, такі як збільшення кількості встановлених відновлюваних джерел енергії (надлишкова потужність) і поширення відновлюваних генераторів на ширшій географічній території.

Це дає можливість скористатися різними погодними умовами від місця до місця та ефектом згладжування, очікуваним від додаткових вітряних та сонячних генераторів.

Впровадження системи накопичення енергії є найбільш економічно ефективним і надійним заходом, враховуючи вартість додаткової відновлюваної генерації та проблеми будівництва нових об’єктів передачі [1].

Крім того, важливо забезпечити правильний баланс між попитом і пропозицією електроенергії, оскільки дисбаланс погіршить стабільність і якість (напруги і частоти) електропостачання. Через потребу в безперервному та гнучкому постачанні в енергосистемі, щоб уникнути цих проблем, генеруючі станції зазвичай оснащені двома основними функціями на додаток до основної функції генерації.

1. Функція для генерування достатньої потужності, коли це необхідно.

2. Функція контролю частоти, яка точно регулює вихідну частоту електроенергії відповідно до постійних коливань попиту, використовуючи додаткову потужність, коли це необхідно.

Однак установки відновлюваної енергетики, такі як сонячна та вітрова, не володіють функцією генерування достатньої потужності, так і функцією регулювання частоти, якщо їх не модифікувати належним чином.

Такі модифікації можуть стосуватися зменшення потужності (негативний запас потужності) або фазового інвертора. Системи накопичення енергії в об'єктах відновлюваної енергетики допомагають компенсувати такі труднощі за допомогою функції генерування достатньої потужності та функції регулювання частоти.

Крім того, у разі збоїв в електромережі, системи накопичення енергії допомагають у безперервному постачанні електроенергії споживачам [1, 2]. З обговорень вище, важливість систем виробництва електроенергії з відновлюваних джерел, а також їх взаємопов'язаних систем зберігання енергії не можна переоцінити.

Таким чином, існує необхідність впровадження безпечних, своєчасних, надійних і складних засобів управління роботою всієї системи генерації. Хоча може знадобитися наявність локальних засобів для реалізації цього управління, не менш важливо мати скоординоване керування як окремою системою, так і з іншими компонентами в мережі, коли потрібні загальномережні програми.

У більшості випадків силові електронні перетворювачі забезпечують підключення систем відновлюваної генерації та накопичення енергії до мережі. Силові електронні системи також

допомагають у синхронізації різних генеруючих систем з комунальною мережею, включаючи перетворювач, який модулює форми струму та напруги до рівня, який можна безпосередньо подавати в мережу або брати з неї.

Іноді перетворювач може бути підключений до трансформатора для забезпечення необхідної напруги перед підключенням до мережі, таким чином підвищуючи ефективність усієї системи [1].

Перетворювачем керує контролер, який визначає задані значення та параметри системи відновлюваної генерації та накопичення енергії, включаючи величину активної та реактивної потужності та стан заряду (SOC) системи зберігання. У багатьох додатках системи зберігання енергії, силові електронні перетворювачі та інші пристрої, підключені до мережі, є від багатьох виробників і постачальників, тому сумісність і сумісність систем з точки зору зв'язку та електричних з'єднань є обов'язковими.

Таким чином, через таку складну природу всієї системи та стратегій контролю місцева робоча сила може бути нерентабельною, отже, існує потреба в надійній системі дистанційного керування всією системою виробництва електроенергії [1, 2].

1.2 Принципи побудови SCADA систем

SCADA - це абревіатура, утворена з перших літер терміна «Диспетчерське управління та збір даних». Це технологія, яка дозволяє користувачеві збирати дані з одного чи кількох віддалених об'єктів та/або надсилати обмежені контрольні інструкції до цих об'єктів.

Основною функцією SCADA є отримання даних від віддалених пристроїв, таких як батареї, клапани, насоси, тощо, і забезпечення

загального дистанційного керування з програмної платформи SCADA Host [3, 4, 6].

SCADA робить непотрібним призначення оператора для перебування або частого відвідування цих віддалених місць, коли об’єкти працюють у звичайному режимі. Вперше розроблена в 1950-х роках, SCADA еволюціонувала з використання в телефонних релейних системах і міні-комп'ютерах [3].

Перші системи «SCADA» використовували збір даних за допомогою панелей лічильників, світильників і реєстраторів стрічкових діаграм. Диспетчерське керування здійснювалося оператором, натискаючи вручну різні ручки управління.

Ці пристрої використовувалися і досі використовуються для диспетчерського контролю та збору даних на заводах, фабриках і об’єктах виробництва електроенергії [3–8]. Нижче наведено огляд поколінь архітектур SCADA [5].

Перше покоління - монолітна SCADA: оригінальна система SCADA була створена в той час, коли мереж не існувало. Вона об’єднувала автономні системи, які практично не підключалися до інших систем.

Монолітна SCADA була реалізований за допомогою двох ідентично обладнаних мейнфреймів, де глобальні мережі (WAN) спілкувалися лише з віддаленими терміналами (RTU).

Система використовувала здебільшого власне програмне забезпечення, а резервування було досягнуто шляхом підключення резервного мейнфрейму до всіх RTU. Монолітну архітектуру SCADA зображено на рисунку 1.1.

Друге покоління - розподілена SCADA. Друге покоління SCADA використовувало переваги технології локальної мережі для розподілу

системних функцій і процесів між кількома системами. Системи були дешевшими та більш мініатюрними, ніж їхні попередники.

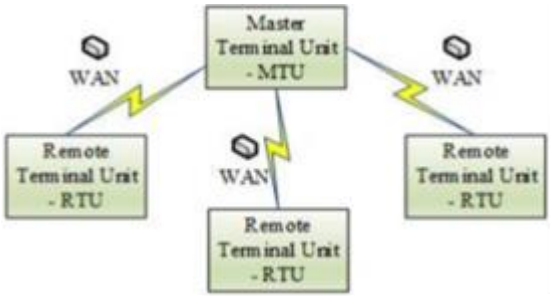


Рисунок 1.1 - Перше покоління - монолітна SCADA [5]

Практично в реальному часі інформація обмінювалася між станціями, кожна з яких мала свої завдання. Розповсюдження збільшило обчислювальну потужність, надійність і резервування системи, але вона не була здатна виходити за межі локального середовища, а використовувані протоколи локальної мережі були здебільшого пропрієтарними. Розподілена архітектура SCADA показана на рисунку 1.2

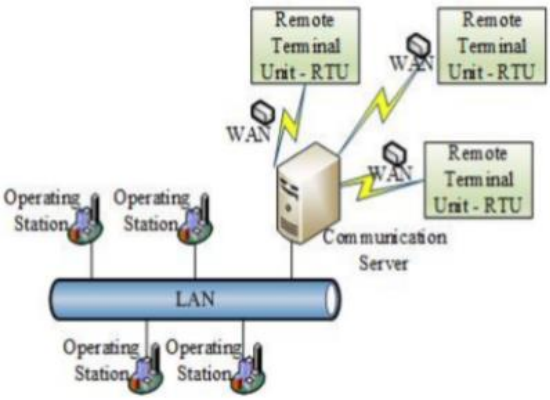


Рисунок 1.2 - Друге покоління - розподілена SCADA [5]

Третє покоління - мережева SCADA: це покоління SCADA передбачало відкриту системну архітектуру з декількома мережевими

системами, які спілкувалися через глобальні мережі, спільно використовували функції головної станції та використовували ПЛК для моніторингу.

З розподіленою функціональністю SCADA в WAN це покоління було дуже схоже на 2-е покоління. Однак, на відміну від 2-го покоління, мережева SCADA могла підключатися до Інтернету та периферійних пристроїв сторонніх виробників за допомогою Інтернет-протоколу (IP).

Ця архітектура SCADA все ще використовується сьогодні. Однак, завдяки технологічному прогресу, системи SCADA тепер розробляються з використанням передового програмного забезпечення, високопродуктивних мікропроцесорів і бездротових, хмарних і Інтернет речей (IoT) технологій, що призвело до четвертого покоління SCADA. Мережева архітектура SCADA показана на рисунку 1.3.

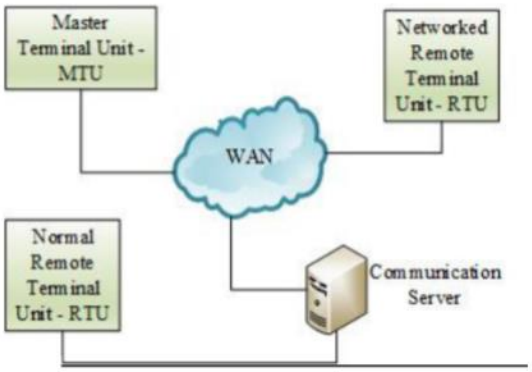


Рисунок 1.3 - Третє покоління - мережева SCADA [5]

Четверте покоління – SCADA базується на основі Інтернету речей (IoT), поєднуючи звичайну SCADA з хмарою, IoT надає системам SCADA альтернативу ПЛК і передбачає використання моделювання даних і складних алгоритмів, що призводить до підвищення доступності даних, гнучкості та доступності, масштабованість і економічна

ефективність. Архітектура SCADA Інтернету речей показана на рисунку 1.4.

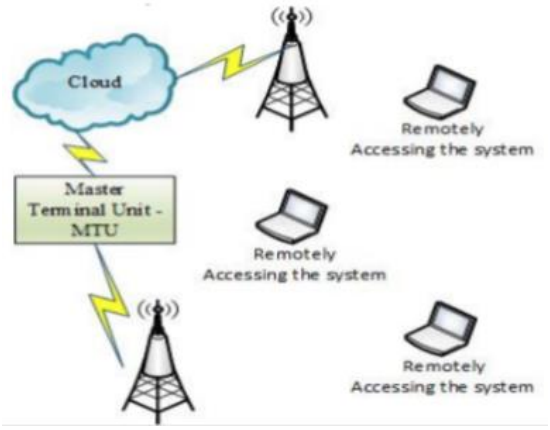


Рисунок 1.4 - Четверте покоління - SCADA на основі Інтернету речей (IoT) [5]

Є чотири основні елементи або рівні системи SCADA, які дозволяють їй виконувати різні функції [3, 4].

1. Польові контрольно-вимірювальні пристрої, як правильно сказав Том де Марко, «ви не можете контролювати те, що не можете виміряти», тобто прилади є ключовим компонентом безпечної та оптимізованої системи контролю.

Ці пристрої включають датчики, виконавчі механізми, передавачі тощо, які безпосередньо підключені до технологічних систем, якими керують, і допомагають вимірювати різні контрольні параметри, такі як струм, напруга, потужність, температура, тиск, стан заряду тощо.

2. Віддалені термінальні пристрої (RTU) - це невеликі комп'ютеризовані пристрої, мікроконтролери, мікропроцесори (такі як програмовані логічні контролери (ПЛК)) тощо, які розгортаються в польових умовах на певних ділянках і місцях, і вони допомагають збирати локально контролювати інформацію з пристроїв польових

приладів, обробляти інформацію та аналізувати їх на головну станцію для взаємодії людини з машиною.

3. Головні термінальні пристрої (MTU)/Хост-платформи SCADA - це більші комп'ютерні консолі або сервери, які служать центральним процесором для системи SCADA.

В основі цього блоку лежить пристрій, який видає всі команди, збирає всі дані, зберігає необхідну інформацію, аналізує іншу інформацію для пов'язаних систем, взаємодіє з різними операторами технологічного обладнання та забезпечує інтерфейс людини і машини (HMI), адміністратору, де можна автоматично контролювати систему та керувати нею відповідно до вхідних даних (датчиків).

4. Комунікаційні мережі - вони допомагають підключатися та передавати дані від польових RTU/PLC до хост-платформи SCADA, яка зазвичай розташована віддалено в польовому офісі або центральному центрі керування за допомогою різних протоколів зв'язку, таких як TCP/IP, Ethernet, Wi-Fi, Fieldbus, Modbus, протокол розподіленої мережі (DNP), Profibus, DirectNet тощо.

Загалом існує два класи апаратного та програмного забезпечення SCADA: комерційні і з відкритим кодом.

Комерційна SCADA - це запатентована система, в якій усі основні компоненти від одного виробника, а стандарти часто є специфічними для цієї системи та розроблені виробником [6]. Компанії розробляють власні апаратні та програмні системи SCADA, які вони продають як готові рішення.

У комерційній SCADA відповідальність за надійність і безпеку системи покладається виключно на виробника, що робить користувача вразливим перед одним виробником/постачальником, оскільки

виробник/постачальник може повільно реагувати на технологічні зміни в підсистемі системи SCADA.

Клієнт також піддається ризику, якщо виробник/постачальник припинить свою діяльність. Такі рішення значною мірою є дорогими, оскільки виробник/постачальник не відчуває такого ж конкурентного тиску, щоб утримувати ціни на низькому рівні після початкового продажу [6].

Існує також проблема гнучкості з уже існуючими пристроями та мережею. Наприклад, у великій системі електропостачання пристрої в системі SCADA повинні обмінюватися даними з усіма іншими пристроями, підключеними до мережі, і якщо такі пристрої від різних постачальників, то SCADA повинна підтримувати протоколи кожного постачальника та впровадження така система збільшує витрати та вимагає більше часу на розробку. Відомі наступні виробники/постачальники SCADA: Siemens, Allen Bradley, General Electric, Emerson, Schneider Electric, Modicon, Mitsubishi, Omron тощо.

SCADA з відкритим вихідним кодом - це система з відкритим кодом, яка дозволяє користувачеві «змішувати та поєднувати» компоненти та вибирати найбільш підходящий з кількох постачальників [6].

Користувач отримує більшу гнучкість, оскільки він не пов’язаний з одним постачальником. Це означає, що в системі з відкритим вихідним кодом жоден постачальник не несе відповідальності за загальну продуктивність системи.

У системі SCADA з відкритим вихідним кодом основні компоненти відповідають певним стандартам, які дозволяють замінювати їх подібними компонентами, виготовленими іншими за тими

ж стандартами. Ці стандарти регулюють взаємозв'язки між основними системними компонентами, такими як прилади та віддалений термінал.

Сьогодні на ринку доступно кілька систем SCADA з відкритим кодом, кожна зі своїми сильними та слабкими сторонами. Пропоновані системи SCADA базуються на архітектурі SCADA Інтернету речей (IoT).

1.3 Особливості SCADA систем з відкритим кодом

Існують різні SCADA системи з відкритим кодом. Прикладами таких SCADA систем є Rapid SCADA, Tango SCADA, PV Browser, Mango, PySCADA, FreeSCADA, IntegraXor тощо.

У [9] представлено інтегровані бездротові датчики та виконавчі мережі (WSAN) та веб-систему Mango SCADA із відкритим кодом для моніторингу критичної інфраструктури за допомогою Інтернет-протоколу (IP) і веб-сервісів, яка надала доступ до Інтернету.

Mango SCADA використовувалася для моніторингу та керування електричною мережею. Існують проблеми, з якими стикаються при розробці типової системи SCADA, включаючи зв'язок у реальному часі, якість обслуговування (QoS), підтримку управління та безпеку.

Основним недоліком використання стандартного програмного забезпечення SCADA, такого як Mango та Rapid SCADA, є те, що вони спілкуються за допомогою кількох протоколів, і важко налаштувати протоколи без спеціального навчання в системі [10]. Ці програмні рішення SCADA з відкритим кодом також не є на 100% безкоштовними.

Так само, як і програмні рішення SCADA з відкритим кодом, сьогодні також доступні серверні рішення SCADA з відкритим кодом. Приклади таких серверів включають системи на основі LabView,

KingView та Simulink. Нижче наведено деякі пов’язані роботи в цій області.

У [11] розглянуто систему SCADA з відкритим кодом для моніторингу та контролю якості електроенергії в інтелектуальній мікромережі з використанням LabView.

Система працювала з дев'ятьма виміряними та дистанційно керованими вузлами, кожен вузол містив кнопку ON/OFF, яка могла використовувати для управління силовим комутаційним елементом універсального вимірювального блоку. Цей блок мережі кожного разу надсилав файл з аналоговими виміряними параметрами, а їх середні значення відображалися в головному вікні системи LabView SCADA для моніторингу.

У [12] розглянуто систему SCADA з відкритим вихідним кодом для моніторингу мікромережі постійного струму Islanded, включаючи вітрові турбіни, фотоелектричні установки та акумуляторні блоки.

Система SCADA виконувала обмін даними між хостом і підлеглими комп’ютерами за допомогою інтелектуальних модулів і різних методів зв’язку. Розроблена система SCADA могла відстежувати параметри в реальному часі, такі як напруга, струм або потужність, і зберігати важливі параметри в базі даних SQL.

У цій системі SCADA головним комп’ютером (сервером) було програмне забезпечення конфігурації KingView, встановлене на ПК з ОС Windows із можливістю зв’язку RS 232C.

Багато інтелектуальних модулів було підключено до цього головного комп’ютера через кабель в одному корпусі (Modbus TCP/IP) і через конвертер RS-232/485 в іншому випадку (Modbus TCP/IP), тоді як дистанційний моніторинг був встановлений за допомогою сервісної платформи GPRS.

У [13] розглянуте масштабоване серверне рішення SCADA з відкритим вихідним кодом разом із економічно ефективним середовищем зв'язку для багатопротокольних інтелектуальних мережевих пристроїв.

У цій системі SCADA сервер OPC був розроблений за допомогою MatLab/Simulink для з'єднання програм на базі Windows і апаратного забезпечення для керування процесами, а також для роботи в якості проміжного програмного забезпечення для встановлення зв'язку в багатопротокольному середовищі.

Крім того, вони використовували недорогий стільниковий (GPRS/3G) протокол зв'язку для передачі даних між центральним сервером OPC SCADA на базі Simulink та пристроями, встановленими на місці (RTU).

Незважаючи на те, що реалізувати LabView/KingView та системи на основі Simulink досить просто, вони дорогі порівняно з іншими недорогими технологіями [10].

1.4 Висновки до першого розділу

1. Існує необхідність впровадження безпечних, своєчасних, надійних і складних засобів управління роботою всієї системи генерації. Хоча може знадобитися наявність локальних засобів для реалізації цього управління, не менш важливо мати скоординоване керування як окремою системою, так і з іншими компонентами в мережі, коли потрібні загальномережні програми.

2. Існує проблема гнучкості з уже існуючими пристроями та мережею. Наприклад, у великій системі електропостачання пристрої в системі SCADA повинні обмінюватися даними з усіма іншими

пристроями, підключеними до мережі, і якщо такі пристрої від різних постачальників, то SCADA повинна підтримувати протоколи кожного постачальника та впровадження така система збільшує витрати та вимагає більше часу на розробку.

3. Основним недоліком використання стандартного програмного забезпечення систем SCADA є те, що вони спілкуються за допомогою кількох протоколів, і важко налаштувати протоколи без спеціального навчання в системі. Ці програмні рішення SCADA з відкритим кодом також не є на 100% безкоштовними.

2 ОСНОВНА ЧАСТИНА

2.1 Структурна схема системи малої генерації енергії

Гібридна система малої генерації енергії складається з різних компонентів. При проектуванні гібридної енергетичної системи малої генерації енергії враховуються такі фактори, як розмір компонентів, конфігурація системи, достатність різноманітних відновлюваних джерел енергії в цьому регіоні, економіка проекту зі зміною навантаження та вартості компонентів.

Структурна схема системи малої генерації енергії зображена на рисунку 2.1.

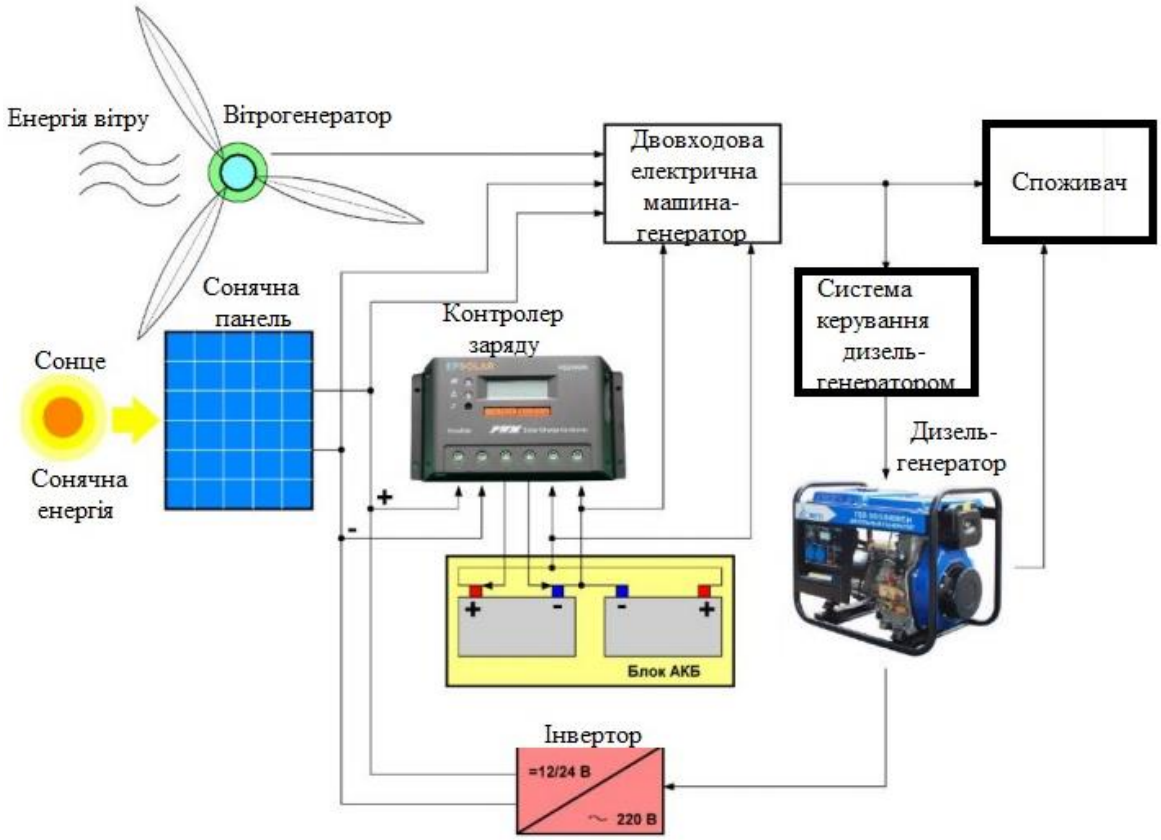


Рисунок 2.1 – Структурна схема системи малої генерації енергії

До складу системи малої генерації енергії входить вітрогенератор, сонячна електростанція, система накопичення енергії та дизельгенератор.

Важливим для система малої генерації енергії є також життєвий цикл системи, чиста поточна вартість система, вартість енергії для кінцевого користувача. Витрати на обслуговування та річні експлуатаційні витрати гібридної системи допоможуть особі, яка приймає рішення, визначити найбільш економічно ефективні рішення гібридної системи для задоволення електричних навантажень, для яких вона розроблена.

Програмне забезпечення Hybrid Optimization of Multiple Energy Resources (HOMER), розроблене Національною лабораторією відновлюваної енергії, моделює мікроенергетичні системи з одним або кількома джерелами живлення.

HOMER допомагає проектувати автономні та підключені до мережі системи найбільш економічно ефективними способами, враховуючи вищевказані фактори. HOMER моделює різні конфігурації, щоб знайти найменш вартісну комбінацію, яка відповідає розглянутим електричним навантаженням.

Можливості оптимізації та аналізу чутливості HOMER допомагають відповісти на важливі питання дизайну, такі як; «Які технології є найбільш економічно ефективними? Якого розміру мають бути компоненти? Що станеться з економікою проекту, якщо зміниться вартість або навантаження? Чи адекватний ресурс відновлюваної енергії?» тощо.

Найбільш поширеними система малої генерації енергії є сонячні електростанції, тому спроектуємо SCADA систему малої генерації енергії для сонячні електростанції.

2.2 Розробка SCADA системи малої генерації енергії для сонячні електростанції

Розробимо недорогу систему SCADA з відкритим кодом, яка базується на архітектурі SCADA Інтернету речей (IoT) і є четвертою та останньою архітектурою SCADA. При з'єднанні апаратних компонентів разом розглянемо дві конфігурації.

У першій конфігурації Raspberry Pi підключається до мережі через кабель Ethernet, щоб користувачі мережі з відповідними авторизаціями могли візуалізувати отримані дані на серверній платформі Thinger.IO IoT.

Крім того, у цій конфігурації, відкривши порт Thinger.IO у мережі, користувачі можуть отримати доступ до збережених даних через Інтернет, використовуючи свій приватний офіс або домашню мережу. У другій конфігурації Raspberry Pi підключено до одного з портів LAN локального маршрутизатора Wi-Fi.

У цьому випадку лише комп'ютери, які підключені до мережі Wi-Fi бездротовим способом або через кабелі Ethernet, можуть отримати доступ до даних на платформі IoT локального сервера Thinger.IO за допомогою IP-адреси Raspberry Pi.

Хоча перша конфігурація є більш гнучкою, друга конфігурація більш безпечна, оскільки ця конфігурація забезпечує створення свого роду промислової мережі, у якій зовнішні користувачі Інтернету, навіть у мережі, відключені.

У будь-якій конфігурації ESP32 Thing може підключатися до сервера Thinger.IO через Wi-Fi за допомогою IP-адреси Raspberry Pi. Дві конфігурації SCADA системи малої генерації енергії для сонячні електростанції показано на рисунку 2.2 і рисунку 2.3 відповідно.

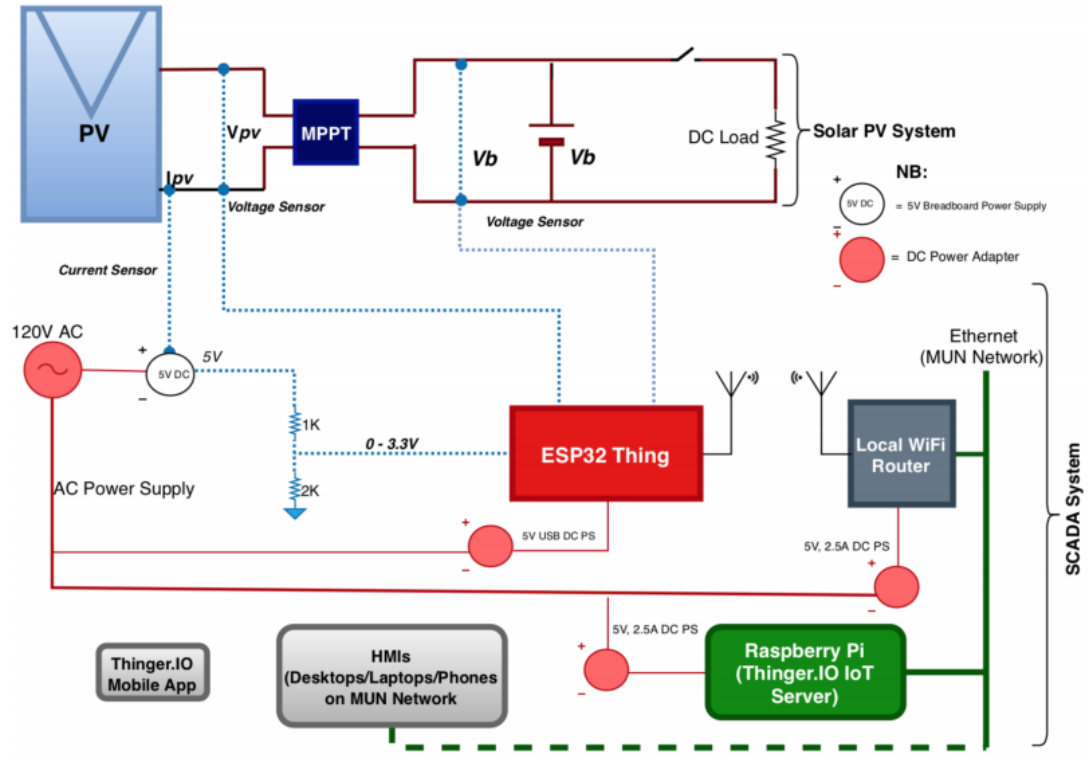


Рисунок 2.2 – Перша конфігурація SCADA системи малої генерації енергії для сонячні електростанції

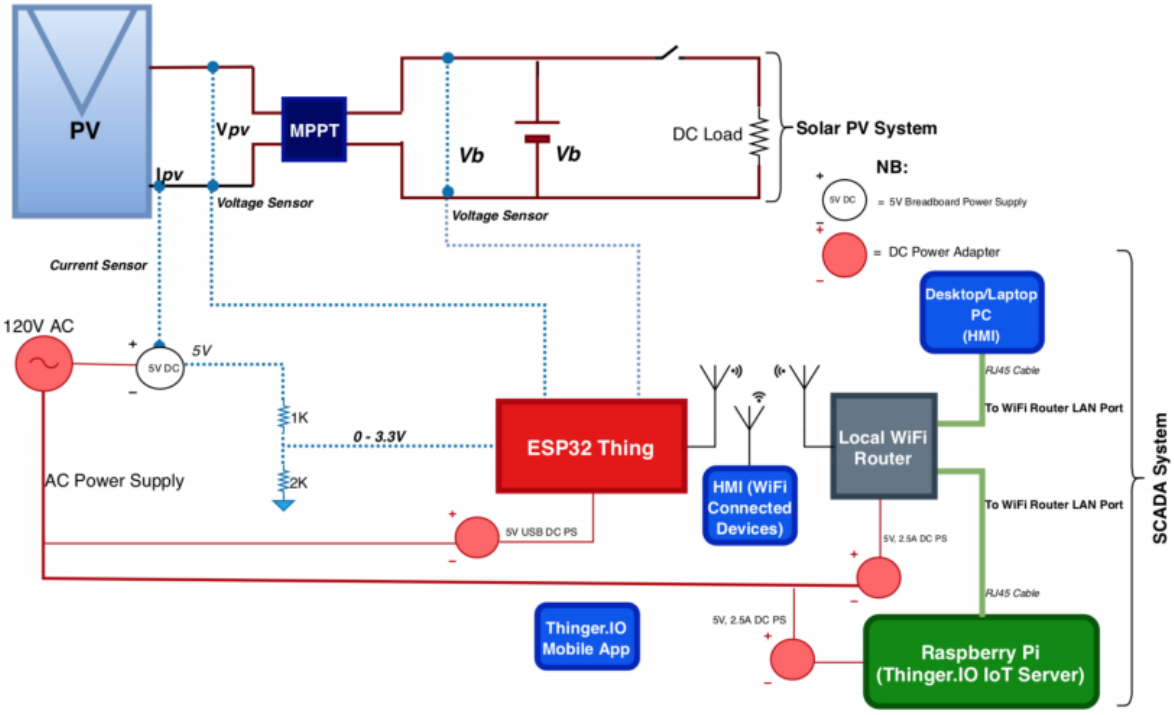


Рисунок 2.3 – Друга конфігурація SCADA системи малої генерації енергії для сонячні електростанції

SCADA системи малої генерації енергії для сонячні електростанції складається з аналогових датчиків напруги та струму для збору даних, мікроконтролера SparkFun ESP32 Thing для отримання, обробки та аналізу даних датчиків, маршрутизатора Wi-Fi для створення локальної мережі Wi-Fi (канал зв'язку) і локального IoT-сервера Thinger.IO з графічним інтерфейсом користувача, створеним для віддаленого моніторингу даних датчиків і диспетчерського контролю.

Датчики є пристроями польового вимірювання, оскільки вони підключені безпосередньо до фотоелектричної системи, якою керують для отримання бажаних даних.

У SCADA системи малої генерації енергії для сонячні електростанції використовуються три аналогових датчика: один датчик струму на ефекті Холла ACS 712 і два модулі електронного датчика напруги МН.

Напруга робочого сигналу (VCC) датчика струму становить 5 В від одного джерела живлення, а напруга датчика напруги становить від 3,3 В до 5 В, тоді як напруга мікроконтролера ESP32 Thing становить від 0 В до 3,3 В. Це означає, що датчик струму не підходить для прямого підключення до контактів аналого-цифрового перетворювача (АЦП) ESP32 Thing.

Тому, щоб переконатися, що датчик відповідає напрузі сигналу 3,3 В ESP32 Thing, виконується деякий зсув рівня за допомогою розташування резисторів, що знижують максимальну напругу 5 В до максимальної напруги 3,3 В мікроконтролера ESP32 Thing, як зображено на рисунку 2.4.

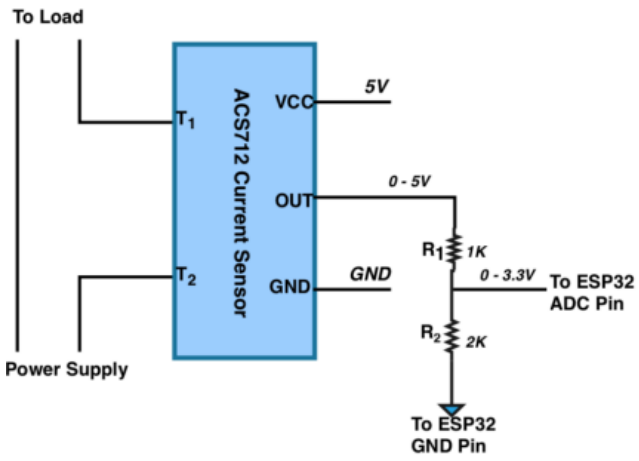


Рисунок 2.4 - Зменшення максимальної напруги 5 В датчик струму на ефекті Холла ACS 712 до максимальної напруги 3,3 В мікроконтролера ESP32 Thing

Використаємо датчик струму на ефекті Холла ACS 712 компанії Allegro MicroSystems, LLC. Це недорога повністю інтегрована мікросхема лінійного датчика струму на основі ефекту Холла з провідником струму з низьким опором.

Він має малошумний тракт аналогового сигналу, працює з одним джерелом живлення 5 В, а також вихідну чутливість від 66 до 185 мВ/А та майже нульовий магнітний гістерезис.

Під час його роботи прикладений струм, що протікає через мідний провідник, створює магнітне поле, яке IC Холла перетворює на пропорційну вихідну напругу. Вихідна напруга пропорційна вимірюваним змінним або постійним струмам.

Цей модуль здатний вимірювати значення струму від 0 А до 30 А. Крім того, у цьому модулі 0 А відповідає 2,5 В, тоді як 30 А відповідає 5 В. Ці параметри, а також параметри контактів АЦП ESP32 Thing враховуються при написанні коду ESP32-Arduino для вимірювання PV струму за допомогою датчика.

Мікроконтролер ESP32 Thing компанії SparkFun Electronics є комплексною платформою розробки. Це сумісний з Wi-Fi мікроконтролер, він підтримує Bluetooth Low-Energy (тобто BLE, BT4.0, Bluetooth Smart) і має майже 30 входів/виходів (I/O).

За словами виробника, її називають «Thing», тому що вона є ідеальною основою для проектів Інтернету речей. Це один із найбільш унікальних недорогих мікроконтролерів з низькою потужністю споживання електричної енергії (приблизно 0,5 Вт), які доступні сьогодні на ринку.

Плату можна живити від джерела живлення USB на 5 В або від одноелементної літій-полімерної (LiPo) батареї, а робочий діапазон напруги сигналу становить від 2,2 В до 3,6 В. На рисунку 2.5 показано зображення плати SparkFun ESP32.

Рисунок 2.5 - Мікроконтролер ESP32 Thing

Мікроконтролер ESP32 Thing програмується за допомогою інтегрованого середовища розробки (IDE) Arduino. Arduino IDE дозволяє писати потрібні програми та завантажувати їх на плату через USB-

кабель. Програми Arduino називаються Sketches, а їхня мова - це лише набір функцій C/C++.

У SCADA системі малої генерації енергії для сонячної електростанції використаємо одноплатний комп’ютер (мікрокомп’ютер) Raspberry Pi 2 моделі B. Raspberry Pi 2 моделі B містить чотирьохядерний процесором BCM2836 (4 процесори в одному чіпі) ARMv7. Це недорога мікросхема із наступними характеристиками.

1. Містить чотирьохядерний процесор ARM Cortex-A7 на 900 МГц.
2. Містить 1 Гб оперативної пам'яті.
3. Містить базовий Ethernet порт.
4. Містить чотири порти USB.
5. Містить сорок контактів GPIO.
6. Містить повний порт HDMI.
7. Містить комбінований аудіороз'єм 3,5 мм і композитне відео.
8. Містить інтерфейс камери (CSI).
9. Містить інтерфейс дисплея (DSI).
10. Містить слот для карти Micro SD.
11. Містить графічне ядро VideoCore IV 3D.

У SCADA системі малої генерації енергії для сонячні електростанції сервер Thinger.IO IoT встановлюється та налаштовується на Raspberry Pi, а Raspberry Pi підключається до інших компонентів у будь-якій із двох описаних раніше конфігурацій.

У SCADA системі малої генерації енергії для сонячної електростанції зв’язок Wi-Fi служить каналом зв’язку між ESP32 Thing (RTU) і Thinger.IO IoT Server (MTU).

Ця локальна мережа Wi-Fi створена за допомогою маршрутизатора D-Link (DI-524 Airplus G). Це високошвидкісний

маршрутизатор зі швидкістю передачі даних 54 Мбіт/с, бездротовим протоколом 802.11b/g і сумісністю зі стандартом IEEE 802.11.

Маршрутизатор має один порт WAN і чотири порти LAN, а його робочий діапазон частот становить 2,4 ГГц. Оскільки ESP32 Thing може реалізувати TCP/IP, повний протокол 802.11b/g/e/i WLAN MAC і Wi-Fi Direct, маршрутизатор налаштовано на налаштування необхідної локальної мережі Wi-Fi для передачі даних датчика з ESP32 Thing на сервер Thinger.IO IoT за допомогою IP-адреси сервера для ідентифікації платформи.

Для контролю доступу до Wi-Fi та безпеки ESP32 Thing підключається до маршрутизатора за допомогою імені мережі Wi-Fi (SSID) і призначеного пароля.

Thinger.IO - це потужна платформа з відкритим кодом для Інтернету речей (IoT). Він підтримує інтерфейс прикладного програмування (API) Representational State Transfer (REST), який дозволяє контролювати та читати смарт-пристрої.

Representational State Transfer (REST) - це архітектура, заснована на веб-стандартах, яка використовує для зв’язку протокол HTTP. Цей протокол забезпечує взаємодію між різними пристроями в Інтернеті, розглядаючи кожен компонент як ресурс, так що до кожного ресурсу можна отримати доступ через загальний інтерфейс, використовуючи загальні методи HTTP, такі як OPTIONS, GET, PUT, POST і DELETE тощо.

Конкретна відповідь кожного ресурсу отримується у форматі JSON/XML. Деякі з важливих функцій протоколу, які роблять можливим підключення та керування пристроями IoT, це автоматичне виявлення API та економія пропускної здатності.

Платформа Thinger.IO IoT підтримує інтеграцію апаратного забезпечення, сумісного з Arduino (будь-яка плата, яку можна запрограмувати за допомогою Arduino IDE, наприклад Arduino + Ethernet, Arduino + Wi-Fi, ESP32/8266/13, NodeMCU, TC CC320 тощо).

Пристрої на базі Linux, такі як Raspberry Pi, Intel Edison або будь-який інший комп'ютер Linux під керуванням Ubuntu або MacOS, а також платформа ARM Mbed і сумісні пристрої.

Платформа IoT має хмарну консоль із чудово розробленим інтерфейсом, де користувач може керувати підключеними пристроями та візуалізувати інформацію про пристрій у хмарі. Щоб запустити проект IoT у Thinger.IO, першим кроком є створення пристроїв, додавши параметри пристрою, які нададуть доступ для підключення пристроїв до облікового запису Thinger.IO.

Будь-який пристрій на платформі має бути зареєстрований, щоб отримати доступ до хмари, і кожен пристрій ідентифікується за його унікальним ідентифікатором та обліковими даними, щоб нескінченну кількість пристроїв можна було додавати до хмарної платформи, при цьому один пристрій не заважає іншому.

Коли користувач створює обліковий запис у хмарі Thinger.IO, маркер доступу отримується з імені користувача та пароля користувача, і цей маркер доступу надає авторизаційний доступ до ресурсів облікового запису підключених пристроїв.

У хмарній консолі є розділ статистики, де користувач може побачити деяку основну інформацію про обліковий запис, таку як кількість підключених пристроїв, кінцевих точок, сегментів даних, статистику споживання пристроєм з точки зору надісланих і отриманих даних, а також карту Google приблизного поточного розташування підключених пристроїв.

У Cloud Console оператор може додавати або видаляти пристрої, створювати інформаційні панелі в реальному часі, отримувати доступ до API пристрою та виконувати інші операції з керування пристроєм і даними. Крім того, ліворуч на екрані статистики є головне меню з усіма функціями платформи, необхідними для створення проектів IoT.

Однією з унікальних особливостей платформи Thinger.IO IoT є те, що вона дозволяє оператору виявляти ресурси, визначені в підключених пристроях. У цьому випадку ресурсом може бути датчик, який показує струм, напругу, температуру, вологість, тиск, або будь-який активний елемент, наприклад світло, реле, двигун тощо.

Після підключення пристрою до облікового запису оператор може отримувати доступ до своїх ресурсів і досліджувати кінцеві точки API REST за допомогою API Explorer, який доступний на інформаційній панелі пристрою, натиснувши маленьку синю кнопку під назвою Device API. По суті, будь-який ресурс пристрою схожий на функцію зворотного виклику, яку можна викликати (на вимогу) через REST API.

На платформі Thinger.IO можна визначити чотири різні типи ресурсів: один для введення (надсилання даних на пристрій), один для виведення (надсилання даних або інформації з пристрою), один для введення/виведення (надсилання та отримання інформації) за один виклик) та ресурс зворотного виклику, який може бути виконаний без надсилання або отримання інформації.

З точки зору API, вхідні та вихідні дані можуть бути будь-яким документом JavaScript Object Notation (JSON).

Thinger.IO має опцію локального сервера, де користувач може придбати (одноразово) образ ISO для установки обладнання або апаратного забезпечення та встановити його на автономній або мережевій машині для належного керування.

Він також має опцію веб-сервера, до якого можна отримати доступ за допомогою URL-адреси, як і будь-якої іншої веб-програми. Як веб-сервер, так і локальний сервер мають можливість IoT, оскільки до даних, що зберігаються в них, можна отримати віддалений доступ за допомогою комп'ютера з підтримкою Інтернету або телефону з підтримкою Інтернету через мобільний додаток Thinger.IO або браузер телефону.

Однак локально встановлений сервер є більш безпечним, оскільки користувач має кращий контроль над сервером і збереженими даними з метою безпеки. У цьому документі використовується варіант локального сервера. У SCADA системі малої генерації енергії для сонячної електростанції використаємо Thinger.IO Raspberry Pi ISO Image, який встановлено та налаштовано на мікрокомп'ютері Raspberry Pi.

Платформа Thinger.IO IoT дозволяє користувачеві створювати всілякі панелі візуалізації та діаграми в режимі реального часу для віддаленого моніторингу, а також підтримує моніторинг і контроль за допомогою електронних листів, HTTP-запитів тощо за допомогою кінцевих точок.

Платформа також має безкоштовні програми для Android та iOS, які можна завантажити з Google Play Store та Apple Store відповідно. Будь-який із цих додатків можна підключити до зареєстрованого пристрою в хмарі, відсканувавши штрих-код QR пристрою. Наприклад, мобільний додаток Thinger.IO на iPhone підключається до пристрою ESP32 на локальному сервері шляхом сканування штрих-коду QR пристрою в хмарі.

Це означає, що на додаток до функцій моніторингу та керування хмарною консоллю, наданих платформою, оператор також може перевіряти стан підключених пристроїв, візуалізувати та оновлювати

вихідні ресурси, редагувати та публікувати вхідні ресурси, а також запускати ресурси будь-де в за допомогою мобільних додатків.

У SCADA системі малої генерації енергії для сонячної електростанції локальний сервер Thinger.IO отримує напругу PV, струм PV, виміряну потужність PV та напругу акумуляторної батареї від підключеного пристрою ESP32 у хмарній консолі та автоматично записує дані на створені інформаційні панелі та сегменти даних у задані швидкості передачі даних.

Це означає, що оператор може віддалено візуалізувати (моніторити) отримані дані на Cloud Console, натиснувши API пристрою, або перевіривши інформаційні панелі та сегменти даних. Оператор також може ініціювати дії наглядового контролю, налаштувавши та використовуючи функції кінцевої точки платформи.

Тут кінцева точка визначається як цільове призначення, яке може бути викликано підключеними пристроями для виконання будь-якої дії, як-от надсилання електронного листа, надсилання SMS, виклик REST API, взаємодія з IFTTT (If This Then That), виклик пристрою з іншого облікового запису або виклик будь-якої іншої кінцевої точки HTTP.

2.3 Висновки до другого розділу

1. SCADA системи малої генерації енергії для сонячні електростанції складається з аналогових датчиків напруги та струму для збору даних, мікроконтролера SparkFun ESP32 Thing для отримання, обробки та аналізу даних датчиків, маршрутизатора Wi-Fi для створення локальної мережі Wi-Fi (канал зв’язку) і локального IoT-сервера Thinger.IO з графічним інтерфейсом користувача, створеним для віддаленого моніторингу даних датчиків і диспетчерського контролю.

2. У SCADA системи малої генерації енергії для сонячні електростанції використовуються три аналогових датчика: один датчик струму на ефекті Холла ACS 712 і два модулі електронного датчика напруги МН.

3. У SCADA системі малої генерації енергії для сонячної електростанції локальний сервер Thinger.IO отримує напругу PV, струм PV, виміряну потужність PV та напругу акумуляторної батареї від підключеного пристрою ESP32 у хмарній консолі та автоматично записує дані на створені інформаційні панелі та сегменти даних у задані швидкості передачі даних.

4. Оператор може віддалено візуалізувати (моніторити) отримані дані на Cloud Console, натиснувши API пристрою, або перевіривши інформаційні панелі та сегменти даних. Оператор також може ініціювати дії наглядового контролю, налаштувавши та використовуючи функції кінцевої точки платформи.

5. Кінцева точка визначається як цільове призначення, яке може бути викликано підключеними пристроями для виконання будь-якої дії, як-от надсилання електронного листа, надсилання SMS, виклик REST API, взаємодія з IFTTT (If This Then That), виклик пристрою з іншого облікового запису або виклик будь-якої іншої кінцевої точки HTTP.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ SCADA СИСТЕМИ МАЛОЇ ГЕНЕРАЦІЇ ЕНЕРГІЇ ДЛЯ СОНЯЧНІ ЕЛЕКТРОСТАНЦІЇ

3.1 Використання технології Інтернету речей (IoT) для SCADA системи малої генерації енергії для сонячні електростанції

Інтернет речей (IoT) - це технологія, яка забезпечує взаємозв'язок фізичних пристроїв, таких як будівлі, транспортні засоби тощо, із вбудованою електронікою, датчиками, програмним забезпеченням і підключенням до мережі, щоб пристрої могли збирати та обмінюватися даними в реальному часі між собою. і оператор через спільну платформу, Інтернет або мережу.

IoT об'єднує механічні (фізичні) об'єкти, обчислювальні пристрої, механічні та цифрові машини, живі істоти, користувацькі інтерфейси та аналітику, які взаємопов'язані через загальну інфраструктуру на основі Інтернету.

Концепція IoT стала можливою завдяки використанню існуючих технологій і концепцій, таких як поширені та повсюдні обчислення, вбудовані пристрої, комунікаційні технології, інтернет-протоколи та додатки, а також сенсорні мережі, які допомагають трансформувати ці фізичні пристрої з їх традиційних форм у розумні прилади.

В останні кілька років концепція IoT використовувалася в будинках, школах, підприємствах і промислових сферах для створення розумних міст, розумних будинків, розумних систем охорони здоров'я,

розумних систем управління енергією, розумного транспорту, розумних виробничих систем, систем промислової автоматизації, розумні системи екстреного реагування.

Незважаючи на те, що технологія IoT існує вже деякий час, експерти в цій галузі не повністю дійшли згоди щодо стандартної архітектури IoT. Різні дослідники пропонували різні архітектури . Однак найпоширенішими архітектурами є трирівнева та п’ятирівнева архітектури.

Трирівнева архітектура складається з рівнів сприйняття, мережі та додатків, тоді як п’ятирівнева архітектура складається з рівнів сприйняття, транспортування, обробки, додатків та бізнес-рівнів. Яку б архітектуру IoT не було реалізовано, в системі на основі IoT необхідний надійний зв’язок, оскільки пристрої IoT зазвичай розосереджені на великих географічних територіях.

Технології IoT використовують чотири рівні загальної моделі TCP/IP. Взаємозв’язок між моделлю TCP/IP і протоколами IoT показано на рисунку 3.1 [14, 15].

SCADA системи малої генерації енергії для сонячні електростанції базується на архітектурі IoT-SCADA, в якій функції IoT включені в звичайну систему SCADA для більш надійного збору даних, віддаленого моніторингу та диспетчерського контролю.

Це означає, що протоколи IoT задіяні в розробці системи SCADA на основі IoT. Дослідження показали, що найбільш критичними альтернативами дизайну для розробки додатків реального часу на основі IoT, таких як SCADA, є протоколи зв’язку, формат кодування повідомлень і веб-платформа або платформа IoT.

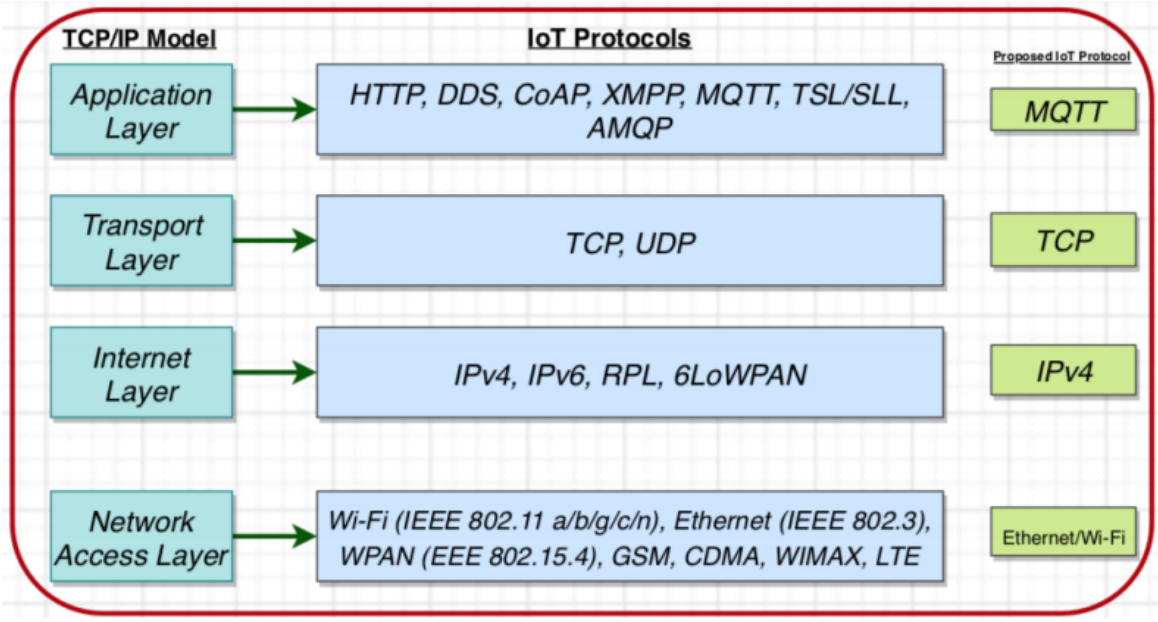


Рисунок 3.1 - Взаємозв'язок між моделлю TCP/IP і протоколами IoT

Тому на кожному з рівнів протоколу IoT, показаних на рисунку 3.1, важливо вибрати правильний протокол для конкретної програми на основі IoT. Наприклад, на рівні додатків потрібно зробити важкий вибір між HTTP, MQTT, CoAP тощо, а також вибрати правильний протокол на рівнях транспорту, Інтернету та доступу до мережі.

У SCADA системи малої генерації енергії для сонячної електростанції протокол передачі даних MQTT реалізовано через бездротове підключення TCP/IP.

ThingsBoard також було обрано як кращу платформу IoT після дослідження альтернатив, і з міркувань безпеки серверна платформа ThingsBoard IoT локально встановлена та розміщена на власній приватній машині (Raspberry Pi) та власній мережі, а не використовувати веб-платформу ThingsBoard, як це зазвичай реалізовано в літературі.

Однією з причин вибору ThingsBoard є те, що він підтримує протокол MQTT через бездротове підключення. Після реалізації серверний API ThingsBoard виконує роль брокера MQTT.

3.2 Використання телеметричного транспортного протоколу повідомлень черги для SCADA системи малої генерації енергії для сонячні електростанції

Message Queuing Telemetry Transport (MQTT) - це легкий міжмашинний протокол передачі даних. MQTT із 2-байтовим фіксованим заголовком особливо підходить для додатків IoT, оскільки він підтримує додатки з обмеженими ресурсами, такими як низька пропускна здатність, низька обчислювальна потужність, низький обсяг пам’яті, що є типовим для домену IoT.

MQTT працює на підключенні TCP/IP і може бути реалізовано в різних мережах, таких як дротові (Ethernet) і бездротові локальні мережі. Спочатку винайдений у 1999 році Енді Стенфорд-Кларком з IBM і Арленом Ніппером з Arcom (тепер Eurotech), MQTT тепер визнаний відкритим стандартом Організацією з удосконалення стандартів структурованої інформації (OASIS).

На рисунку 3.2 показана загальна архітектура протоколу MQTT. По суті, MQTT використовує механізм обміну повідомленнями «Опублікувати-підписатися», і він складається з посередника (сервера) і клієнтів.

Брокера MQTT зазвичай є сервером, що працює в хмарі або в Інтернеті, і відповідає за зберігання опублікованих даних на основі різних тем і випуск повідомлення (даних) законним передплатникам.

Клієнт MQTT - це будь-яке обладнання, програмне забезпечення або комбінація апаратного та програмного забезпечення, яке підключається до Брокера з метою обміну даними. Коли підключений пристрій або Клієнт завантажує дані з Посередника (сервера), процес називається Підпискою, а цей конкретний Клієнт називається Підписником.

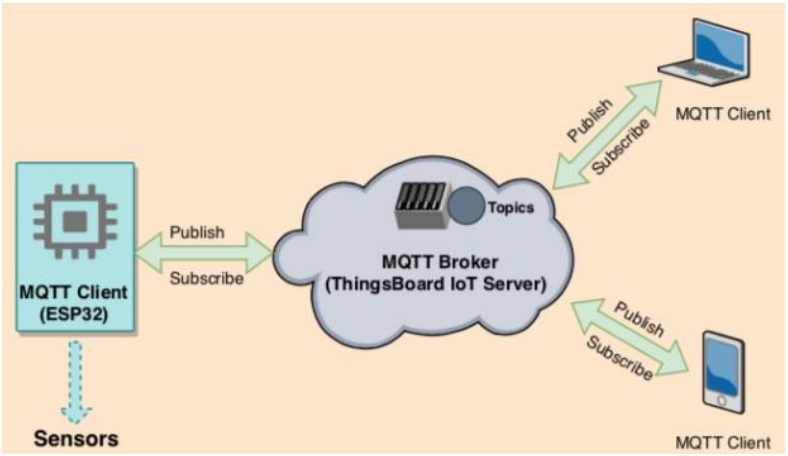


Рисунок 3.2 - Загальна архітектура протоколу MQTT

З іншого боку, коли підключений пристрій (клієнт) надсилає дані брокеру (серверу), клієнт називається видавцем, а процес називається опублікуванням.

Звідси походить термін «Публікація-Підписка», на якому базується протокол MQTT.

Варто зазначити, що брокер може обслуговувати кількох підписників і видавців, залежно від можливостей машини, на якій працює брокер. Клієнт також може діяти як Видавець, так і Підписник.

Коли Клієнт публікує дані, він призначає даним певну тему, розділяючи кожну тему скісною рискою. Для кожної теми є назва теми та пов’язаний із нею рівень теми, а символи підстановки використовуються для відповідності кількох рівнів темі.

Протокол MQTT використовує три рівні якості обслуговування (QoS) для доставки повідомлень: QoS 0, QoS 1 і QoS 2. У QoS 0 одержувач не підтверджує отримання повідомлення, а відправник не передає повідомлення повторно.

На цьому рівні доставка повідомлень базується на гарантії наявності надійного з’єднання між Клієнтом(ами) та Брокером (сервером).

У QoS 1 ідентифікатор пакету призначається повідомленню відправником і одержувач відповідає підтвердженням, якщо він отримує та приймає повідомлення, тоді як непідтверджені повідомлення повторно передаються відправником.

Незважаючи на те, що цей процес може призвести до дублювання повідомлень, ця система реалізована для того, щоб забезпечити фактичне отримання переданого повідомлення брокером MQTT.

Коли використовується QoS 2, між клієнтом MQTT і брокером відбувається обмін кількома повідомленнями, щоб інформація передавалася точно один раз, таким чином запобігаючи втраті та дублюванню даних ціною додаткових накладних витрат.

Отже, існує компроміс між накладними витратами на повідомлення та надійністю. Існує також двоетапний процес підтвердження для публікації повідомлення в цьому QoS.

Залежно від QoS, пов’язаного з повідомленням, контрольні пакети обмінюються перед передачею повідомлення. Пакет керування MQTT складається з фіксованого заголовка, змінного заголовка та корисного навантаження.

Деякі з контрольних пакетів MQTT, якими обмінюються клієнти MQTT і брокер, включають CONNECT, CONNACK, PUBLISH, PUBACK, PUBREC, PUBREL, SUBSCRIBE, SUBACK тощо.

Наприклад, після отримання надісланого повідомлення одержувач надсилає PUBREC, щоб підтвердити отримання пакета і PUBREL передається видавцем повідомлення, щоб вказати випуск опублікованого повідомлення, а потім PUBCOMP передається одержувачем, щоб вказати завершення. видання.

Крім того, MQTT має систему черги, де Брокер буферизує всі повідомлення, якщо Клієнт офлайн, і доставляє їх Клієнту, коли Клієнт знову в мережі. У цьому випадку Клієнт і Брокер зберігають поточний стан сеансу, а Брокер доставляє повідомлення в черзі, коли Клієнт має активний сеанс.

Таким чином, протокол MQTT є надійним і підходить для додатків IoT, де зазвичай є обмежені ресурси, такі як низька пропускна здатність, низький обсяг пам’яті, низьке енергоспоживання тощо.

Ось чому MQTT віддають перевагу іншим протоколам передачі даних, таким як CoAP, HTTP, REST API тощо. MQTT також працює на з'єднанні TCP/IP на відміну від CoAP, який працює на UDP.

Завдяки цим цінним функціям протоколу MQTT його вже використовують численні програми. Наприклад, обмін повідомленнями у Facebook, моніторинг розумного будинку, охорона здоров'я, транспорт, вимірювання енергії, паркування, промислові роботи, виробництво та інтелектуальні системи моніторингу були реалізовані за допомогою протоколів MQTT через різні з'єднання TCP/IP.

У SCADA системи малої генерації енергії для сонячної електростанції реалізовано протокол MQTT для передачі даних PV від клієнта MQTT (мікроконтролер ESP32) до брокера MQTT (сервер ThingsBoard IoT), а персональні комп'ютери та мобільні пристрої можуть підписатися на візуалізацію опублікованих даних на сервері.

Серверний вузол ThingsBoard діє як брокер MQTT і підтримує рівні QoS 0 (не більше одного разу) і 1 (принаймні один раз), а також набір попередньо визначених тем. Цей IoT-сервер ThingsBoard разом із базою даних PostgresSQL побудовано на мікроконтролері Raspberry Pi 2, а клієнтська бібліотека MQTT (Arduino PubSubClient) реалізована за допомогою програмного забезпечення Arduino IDE на платі мікроконтролера ESP32 OLED для збору даних датчиків та їх публікації. на сервер ThingsBoard IoT (MQTT Broker).

До складу SCADA системи малої генерації енергії для сонячної електростанції датчики струму та напруги на ефекті Холла, які служать польовими приладами для отримання даних фотоелектричної системи,

універсальний мікроконтролер ESP32 з OLED-дисплеєм, який є віддаленим терміналом і налаштований як MQTT.

Клієнт для обробки та публікації даних датчиків за допомогою протоколу MQTT, одноплатний комп'ютер Raspberry Pi2, на якому побудовано сервер ThingsBoard IoT, який є головним терміналом і налаштований як брокер MQTT, для взаємодії людини з машиною, зберігання даних, інформаційних панелей, тривоги, публікація даних і підписка та маршрутизатор Wi-Fi для створення підключення TCP/IP Wi-Fi для реалізації протоколу MQTT.

Мікроконтролер ESP32 програмується як клієнт MQTT за допомогою програмного забезпечення Arduino IDE та клієнтської бібліотеки MQTT під назвою PubSubClient.

Плата отримує виміряні в реальному часі PV напругу та значення струму, значення напруги батареї та розраховані значення PV потужності від датчиків, відображає значення як на Arduino IDE Serial Monitor, так і на його OLED-дисплеї, і постійно публікує дані в реальному часі на MQTT Broker, локальну серверну платформу IoT ThingsBoard, використовуючи протокол MQTT через підключення TCP/IP Wi-Fi.

Плата ESP32 представляє віддалений термінальний блок (RTU) у пропонованій системі SCADA на основі IoT, а протокол MQTT, реалізований на платі, допомагає реалізувати основні функції RTU: отримувати та обробляти дані датчиків, а також аналізувати їх для головного термінального блоку - сервера ThingsBoard.

Екран OLED-дисплея плати забезпечує додаткову функцію моніторингу SCADA, надаючи місцевому оператору інтерфейс для перегляду найновіших даних об’єкта, яким керують.

Незважаючи на те, що локальний оператор не може бачити тенденції даних на OLED-дисплеї, як на HMI на головному термінальному блоці (сервер SCADA), можливість бачити найновіші дані може бути критичною у випадку, якщо віддалений оператор SCADA недоступний для здійснення наглядових контрольних дій.

Сервер ThingsBoard IoT, який є брокером MQTT, а також головний термінальний блок для взаємодії людини з машиною та обробки даних встановлено на комп’ютері Raspberry Pi. Це робиться шляхом створення сервера ThingsBoard разом із необхідними сторонніми компонентами та службами в операційній системі Raspberry Pi.

Щоб реалізувати це, ОС Raspbian Buster було встановлено на картку microSD і вставлено в слот для карт micro-SD Raspberry Pi. Відповідно до доступної документації щодо встановлення ThingsBoard і репозиторію сервера ThingsBoard з відкритим кодом на GitHub було запущено декілька кодів встановлення на терміналі Raspberry Pi для завантаження та встановлення сервера ThingsBoard версії 2.4 та його служб на комп’ютері Raspberry Pi.

Базу даних PostgreSQL з відкритим кодом також було встановлено разом із сервером ThingsBoard для зберігання отриманих даних. Після встановлення та конфігурації сервера ThingsBoard і бази даних PostgreSQL на Raspberry Pi, Raspberry Pi було підключено до інших компонентів запропонованої системи SCADA.

Маршрутизатор D-Link (DI-524 Airplus G) використовується для створення підключення до бездротової мережі TCP/IP, через яку реалізовано протокол MQTT для передачі даних від клієнта MQTT (ESP32) до брокера MQTT (сервер ThingsBoard IoT).

Основні характеристики маршрутизатора включають 1 порт WAN і 4 порти LAN, діапазон робочих частот 2,4 ГГц, високу швидкість зі швидкістю передачі даних 54 Мбіт/с, підтримку бездротового протоколу 802.11 b/g і сумісність зі стандартами IEEE 802.11, а також можливості контролю доступу.

Оскільки мікроконтролер ESP32 підтримує стандарти TCP/IP, IEEE 802.11b/g/n Wi-Fi, маршрутизатор налаштовано для встановлення необхідного підключення до локальної мережі Wi-Fi та реалізованого протоколу MQTT на ESP32 (MQTT Client) використовується для публікації отриманих даних датчиків через це бездротове підключення TCP/IP до серверної платформи ThingsBoard IoT (MQTT Broker).

З метою безпеки та контролю доступу підключення до мережі Wi-Fi обмежено за допомогою назви мережі Wi-Fi (SSID) і призначеного пароля, а також інших реалізованих заходів безпеки на маршрутизаторі.

ThingsBoard - це IoT-платформа з відкритим вихідним кодом для збору даних, обробки, візуалізації та керування пристроями. Вона надає готове IoT-хмарне або локальне рішення для створення серверної інфраструктури для різноманітних додатків IoT.

ThingsBoard, створена на платформі Java 8, забезпечує 100-відсоткову підтримку стандартних протоколів IoT для підключення пристроїв, включаючи MQTT, CoAP і HTTP(S), і наразі підтримує три різні варіанти баз даних: SQL, NoSQL і гібридні бази даних.

Платформа ThingsBoard використовує ці бази даних для зберігання об’єктів (таких як пристрої, активи, інформаційні панелі, користувачі, сигнали тривоги, клієнтів тощо) і даних телеметрії (атрибути, покази датчиків часових рядів, статистика, події тощо).

Телеметрії - це часові ряди пар ключ-значення, пов’язаних із певним пристроєм і ThingsBoard зберігає отримані дані телеметрії. Активи є контейнерами для реорганізації отриманих даних і можуть використовуватися для завантаження результатів обробки даних.

Атрибути зазвичай представляють такі функції пристрою, як версія мікропрограми, характеристики апаратного забезпечення тощо, які призначаються зареєстрованим пристроям і активам у формі пар ключ-значення. База даних SQL, така як PostgreSQL, зберігає всі сутності та телеметрію в базі даних SQL, тоді як опція NoSQL, така як Cassandra, зберігає всі сутності та телеметрію в базі даних NoSQL. У варіанті гібридної бази даних зберігаються всі сутності в базі даних SQL, а вся телеметрія зберігається в базі даних NoSQL.

У цьому проекті база даних PostgreSQL встановлена на сервері ThingsBoard для зберігання сутностей і телеметрії.

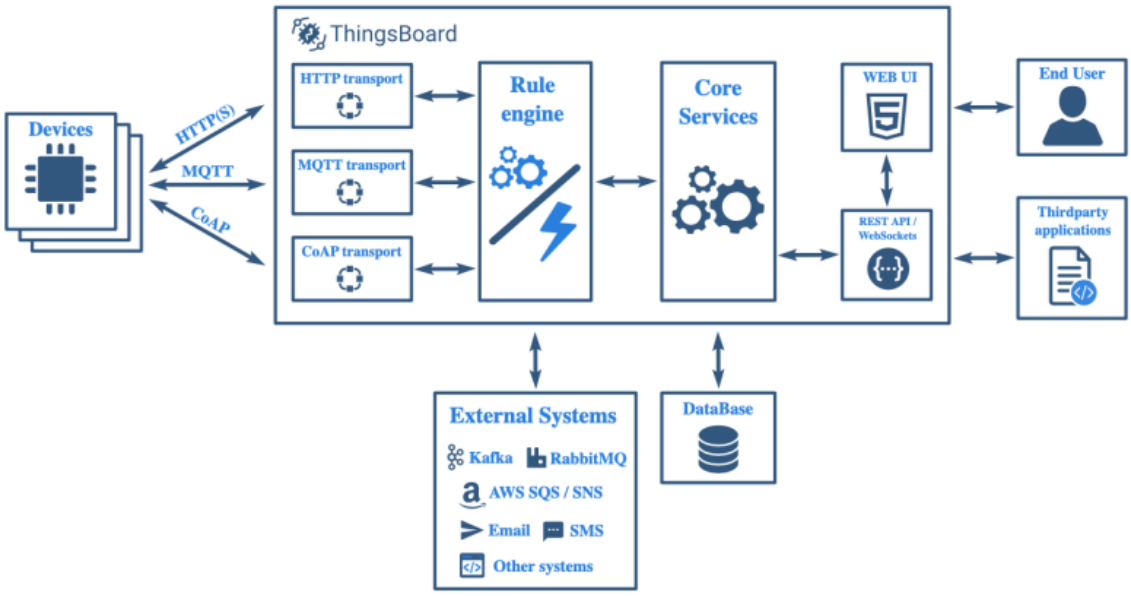
ThingsBoard має дві різні версії: Community Edition, яка є безкоштовною та повністю відкритою, і Professional Edition, яка має розширені функції. Характеристики обох видань та їх основні відмінності показано на рисунку 5.8.

У SCADA системи малої генерації енергії для сонячної електростанції база даних PostgreSQL встановлена на сервері ThingsBoard для зберігання сутностей і телеметрії.

ThingsBoard має дві різні версії: Community Edition, яка є безкоштовною та повністю відкритою, і Professional Edition, яка має розширені функції. Характеристики обох видань та їх основні відмінності показано на рисунку 3.3.

Community Edition є відкритим кодом і доступне безкоштовно як на офіційному веб-сайті ThingsBoard, так і на платформі розробки програмного забезпечення GitHub. Архітектура ThingsBoard розроблена таким чином, щоб бути масштабованою, стійкою до відмов, надійною та ефективною, настроюваною та довговічною. Базова архітектура ThingsBoard показана на рисунку 3.4.

Рисунок 3.3 - Характеристики та основні відмінності Community Edition та Professional Edition



Риунок 3.4 - Базова архітектура ThingsBoard

Ключові характеристики та функції основних компонентів архітектури ThingsBoard наступні.

1. Транспортні компоненти.

До них належать MQTT, HTTP і API на основі CoAP для додатків пристроїв і мікропрограми. Будучи частиною «транспортного рівня» ThingsBoard, кожен із компонентів тут допомагає надсилати дані до механізму правил, а також може використовувати основні служби для надсилання запитів до бази даних для перевірки облікових даних пристрою.

У SCADA системи малої генерації енергії для сонячної електростанції реалізовано API пристрою на основі MQTT, що підтримується протоколом зв’язку MQTT.

MQTT віддають перевагу перед HTTP і CoAP через його унікальні функції, такі як підтримка обмежених ресурсів, таких як низька пропускна здатність, і його можна реалізувати через різні підключення TCP/IP.

Крім того, вузли сервера ThingsBoard діють як посередник MQTT, який підтримує рівні QoS 0 (не більше одного разу) і 1 (принаймні один раз), а також набір попередньо визначених тем, що означає, що зовнішній пристрій можна налаштувати як клієнта MQTT для публікації даних до серверних вузлів. ESP32 запрограмований і налаштований як клієнт MQTT для публікації отриманих даних PV System на вузлах сервера ThingsBoard.

2. Компоненти Rule Engine.

ThingsBoard Rule Engine допомагає обробляти вхідні повідомлення за допомогою визначеної користувачем логіки та потоку. Цей механізм правил, який складається з вузла правил і ланцюга правил, можна легко налаштувати та конфігурувати, і його можна використовувати для обробки складних подій.

Наприклад, системний адміністратор може фільтрувати, збагачувати та перетворювати вхідні повідомлення, надіслані пристроями Інтернету речей і пов’язаними програмами, а також реалізовувати та запускати різні дії, такі як сигнали тривоги, сповіщення про стан підключених пристроїв, сповіщення електронною поштою або інший зв’язок із зовнішніми пристроями за допомогою Rule Engine.

У SCADA системи малої генерації енергії для сонячної електростанції налаштовані механізми правил для загальної реалізації системи (наприклад, передавання даних) і сповіщень про стан сповіщень

про стан фотоелектричної системи, що відстежується (моніторинг напруги акумуляторної батареї), показані на рисунку 3.5 та рисунку 3.6 відповідно.

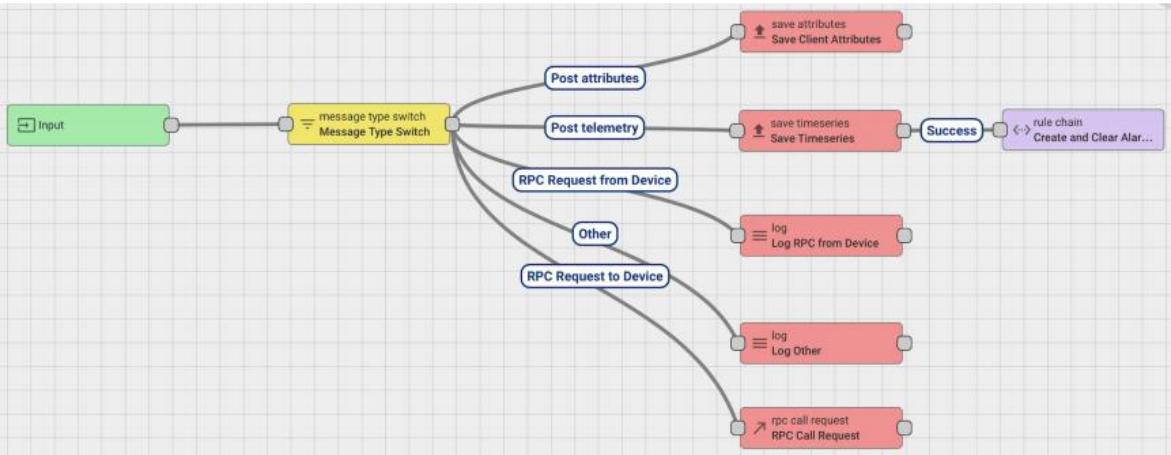


Рисунок 3.5 - Механізм системних правил

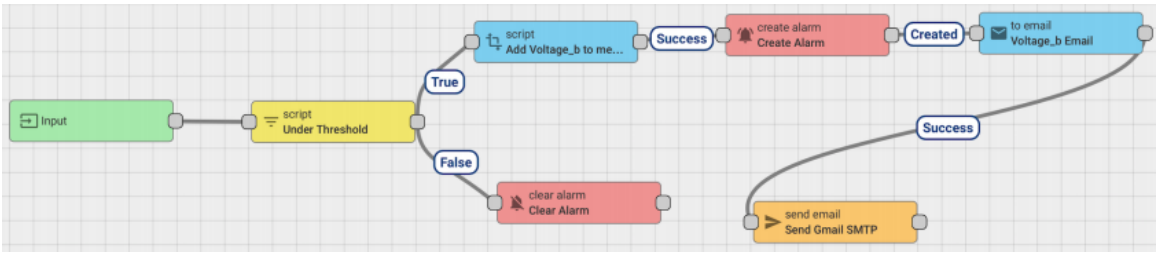


Рисунок 3.6 - Механізм правил тривоги

3. Основні служби.

Основні служби відповідають за обробку викликів REST API, моніторинг станів підключення пристроїв і підписки WebSocket на об’єкти, телеметрію та зміни атрибутів.

4. Зовнішні системи.

За допомогою Rule Engine можна встановити зв’язок між ThingsBoard і зовнішніми системами. Це передбачає надсилання даних до зовнішніх систем, обробку даних і передачу результатів оброблених даних на сервер ThingsBoard для візуалізації.

Можливість таким чином інтегрувати обробку даних ThingsBoard із зовнішніми системами забезпечує велику гнучкість системи, на відміну від більшості платформ Інтернету речей.

Сервер ThingsBoard можна використовувати безпосередньо на платформі Live Demo, інсталювати на локальній приватній машині або розмістити на хмарному сервері, такому як DigitalOcean, Amazon Web Services (AWS), платформа Google Cloud, Microsoft Azure, IMB Cloud тощо.

Безпека в системі SCADA є критичною проблемою, оскільки атаки на SCADA можуть поставити під загрозу важливі дані компанії, що зберігаються в хмарі. Таким чином, у запропонованому рішенні для проектування системи SCADA на основі Інтернету речей реалізовано варіант локального сервера ThingsBoard, розміщеного на власному хості.

Хоча розробники ThingsBoard надали варіанти встановлення та посібники для різних машин і операційних систем, включаючи Windows, Linux (CentOS і Ubuntu), Raspberry Pi, Docker (Linux і MacOS), Docker (Windows), Maven і налаштування кластера, ThingsBoard сервер, який використовується в цій роботі встановлено на машині Raspberry Pi 2 моделі B.

Це пояснюється тим, що Raspberry Pi надійний, портативний і споживає відносно мало енергії порівняно з іншими варіантами. Ця опція Raspberry Pi також абсолютно безкоштовна, оскільки її було

створено з нуля на Raspberry Pi за допомогою посібників зі встановлення на офіційному веб-сайті GitHub і ThingsBoard.

Збереження споживання електроенергії на можливому мінімумі є важливим, оскільки запропонована система SCADA призначена для роботи 24/7 для ефективного моніторингу та диспетчерського контролю.

Встановивши сервер ThingsBoard на комп'ютері Raspberry Pi, підключеному до приватної мережі (мережі MUN і локальної мережі Wi-Fi), можна працювати з публічним Інтернетом і без нього, залежно від обраної конфігурації на основі бажаної безпеки та гнучкість операції.

Наразі ThingsBoard підтримує різні апаратні платформи, включаючи Arduino, ESP32, ESP8266, NodeMCU, Raspberry Pi та LinkIt One, що робить його ідеальним для програм IoT.

Також розробники зараз працюють над підтримкою інших платформ, таких як Intel Edison, C.H.I.P, Samsung Artik, Tessel і Gemalto. Будь-яке вибране обладнання можна підключити до сервера ThingsBoard, і можна реалізувати два різних варіанти автентифікації, які підтримуються ThingsBoard, включаючи маркери доступу та сертифікати X.509.

У SCADA системи малої генерації енергії для сонячної електростанції апаратне забезпечення TTGO ESP32 LoRa32 V1.0 з дисплеєм OLED підключено до встановленого сервера ThingsBoard і реалізовано параметр автентифікації пристрою Access Token.

Це означає, що зареєструвавши пристрій ESP32 на серверній платформі ThingsBoard і призначивши йому маркер доступу, дані підключеного датчика, отримані пристроєм ESP32, можна буде

опублікувати на сервері за допомогою як IP-адреси сервера (IP-адреси Raspberry Pi), так і і маркер доступу для ідентифікації платформи.

На рисунку 3.7 показано інтерфейс встановленого сервера ThingsBoard на машині Raspberry Pi, який показує IP-адресу та численні меню, такі як «Ланцюжки правил», «Клієнти», «Активи», «Пристрої» тощо.

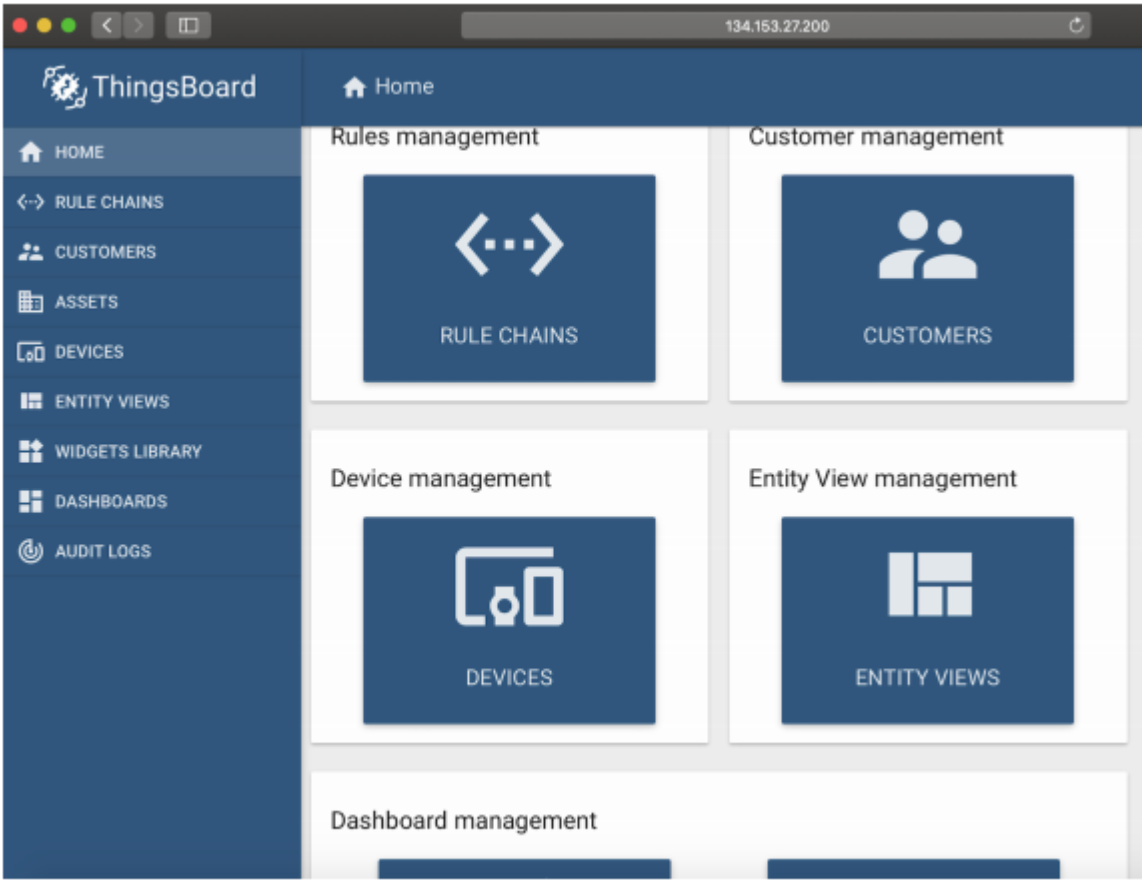


Рисунок 3.7 - інтерфейс встановленого сервера ThingsBoard на машині Raspberry Pi

Цей локально розміщений сервер ThingsBoard на одноплатному комп’ютері Raspberry Pi, який є головним терміналом (MTU) розробленої системи SCADA, виконує роль брокера MQTT, а пристрій ESP32, запрограмований і налаштований як клієнт MQTT, публікує отримані дані датчиків від підключеної фотоелектричної системи до посередника MQTT за допомогою протоколу MQTT через з’єднання бездротової мережі TCP/IP, створене за допомогою маршрутизатора WiFi.

Оскільки ThingsBoard дозволяє створювати настроювані інформаційні панелі в режимі реального часу (HMI) із понад 30 віджетами для візуалізації даних, сигналів тривоги, повідомлень і керування пристроями, красиві інформаційні панелі в реальному часі створюються на локальному сервері ThingsBoard для Візуалізація та керування даними фотоелектричної системи.

Доступ до цих інформаційних панелей (HMI), які показують стани керованої фотоелектричної системи, можна отримати через Інтернет або в автономному режимі, залежно від вибраної розгорнутої конфігурації.

На рисунку 3.8 показано підключений пристрій ESP32. На рисунку 3.9 показані дані датчиків, які надсилаються на вузли сервера ThingsBoard.

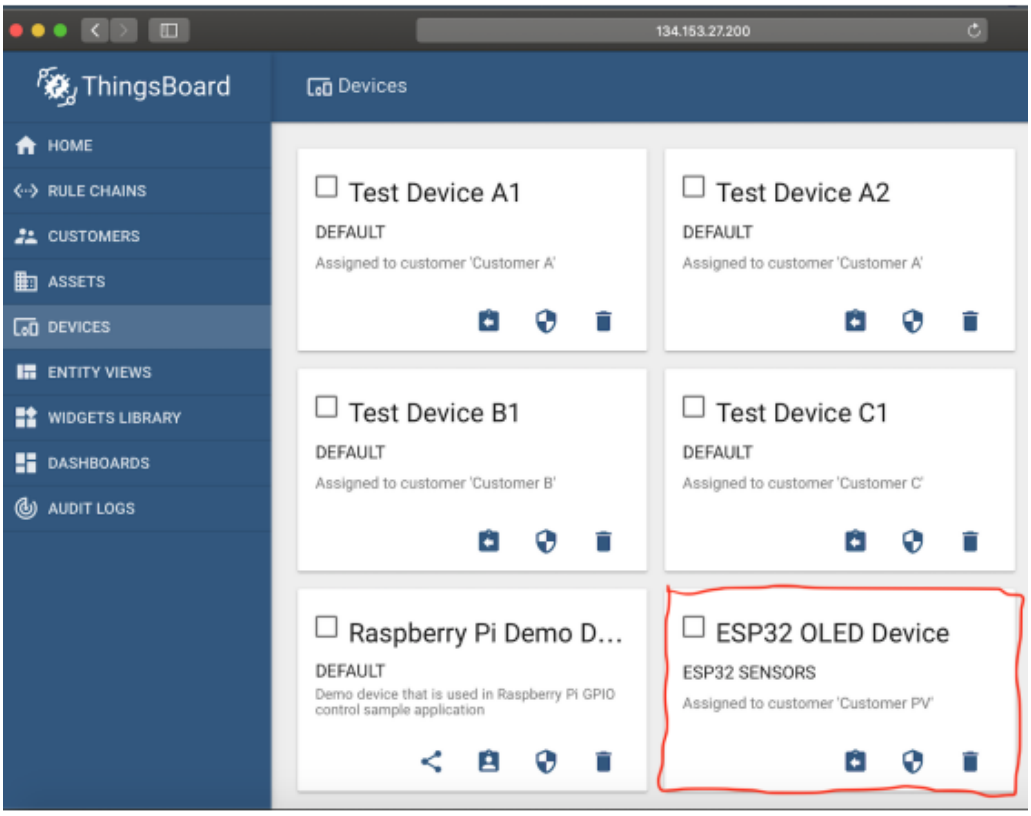


Рисунок 3.8 - Підключений пристрій ESP32

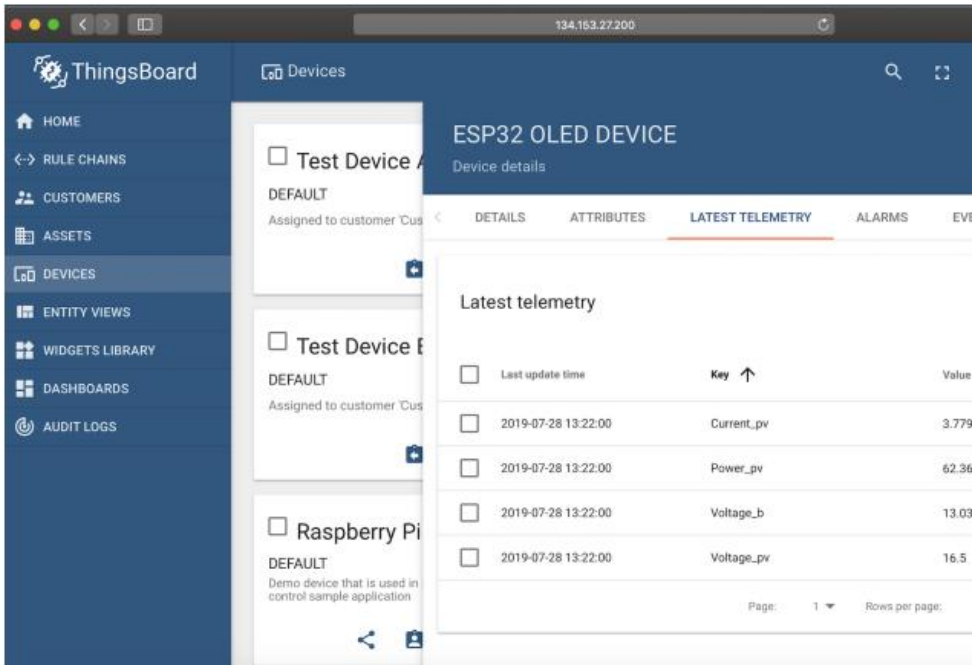


Рисунок 3.9 - Розміщення даних датчиків

3. 3 Висновки до третього розділу

1. SCADA системи малої генерації енергії для сонячні електростанції базується на архітектурі IoT-SCADA, в якій функції IoT включені в звичайну систему SCADA для більш надійного збору даних, віддаленого моніторингу та диспетчерського контролю.

2. У SCADA системи малої генерації енергії для сонячної електростанції реалізовано протокол MQTT для передачі даних PV від клієнта MQTT (мікроконтролер ESP32) до брокера MQTT (сервер ThingsBoard IoT), а персональні комп'ютери та мобільні пристрої можуть підписатися на візуалізацію опублікованих даних на сервері.

3. Сервер ThingsBoard на одноплатному комп'ютері Raspberry Pi, який є головним терміналом (MTU) розробленої системи SCADA, виконує роль брокера MQTT, а пристрій ESP32, запрограмований і налаштований як клієнт MQTT, публікує отримані дані датчиків від підключеної фотоелектричної системи до посередника MQTT за допомогою протоколу MQTT через з'єднання бездротової мережі TCP/IP, створене за допомогою маршрутизатора WiFi.

ВИСНОВКИ

В процесі розробки SCADA система для малої генерації енергії в якості пристрою програмного керування вибрано мікроконтролер SparkFun ESP32 Thing для отримання, обробки та аналізу даних датчиків.

В першому розділі були розглянуті наступні питання: особливості систем малої генерації електроенергії, принципи побудови SCADA систем, Особливості SCADA систем з відкритим кодом.

У другому розділі проведено аналіз вимог технічного завдання та розроблена структурна схема системи малої генерації енергії.

Розроблено SCADA систему малої генерації енергії для сонячні електростанції.

Використано локальний сервер Thinger.IO, який отримує напругу PV, струм PV, виміряну потужність PV та напругу акумуляторної батареї від підключеного пристрою ESP32 у хмарній консолі та автоматично записує дані на створені інформаційні панелі та сегменти даних у задані швидкості передачі даних.

У третьому розділі розроблено програмне забезпечення SCADA системи малої генерації енергії для сонячні електростанції.

Використано технологію Інтернету речей (IoT) для SCADA системи малої генерації енергії для сонячні електростанції.

Реалізовано API пристрою на основі MQTT, що підтримується протоколом зв’язку MQTT для сонячної електростанції.

Локально розміщений сервер ThingsBoard на одноплатному комп’ютері Raspberry Pi, який є головним терміналом (MTU) розробленої системи SCADA, виконує роль брокера MQTT, а пристрій ESP32, запрограмований і налаштований як клієнт MQTT, публікує отримані дані датчиків від підключеної фотоелектричної системи до посередника MQTT за допомогою протоколу MQTT через з’єднання бездротової мережі TCP/IP, створене за допомогою маршрутизатора WiFi.

Оскільки ThingsBoard дозволяє створювати настроювані інформаційні панелі в режимі реального часу (HMI) із понад 30 віджетами для візуалізації даних, сигналів тривоги, повідомлень і керування пристроями, красиві інформаційні панелі в реальному часі створюються на локальному сервері ThingsBoard для Візуалізація та керування даними фотоелектричної системи.

Доступ до цих інформаційних панелей (HMI), які показують стани керованої фотоелектричної системи, можна отримати через Інтернет або в автономному режимі, залежно від вибраної розгорнутої конфігурації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1 L. O. Aghenta and M. T. Iqbal, “Design and Dynamic Modelling of a Hybrid Power System for a House in Nigeria,” International Journal of Photoenergy, vol. 2019, Article ID 6501785, 13 pages, 2019. https://doi.org/10.1155/2019/6501785.

2 IEC White Paper, "Electrical Energy Storage." Internet: https://www.iec.ch/whitepaper/pdf/iecWP-energystorage-LR-en.pdf.

3 J. Lee, S. Lee, H. Cho, K. S. Ham and J. Hong, "Supervisory Control and Data Acquisition for Standalone Hybrid Power Generation Systems," Sustainable Computing: Informatics and Systems, Volume 20, 2018, Pages 141-154, ISSN 2210-5379, https://doi.org/10.1016/j.suscom.2017.11.003. http://www.sciencedirect.com/science/article/pii/S2210537917303062 .

4 K. Stouffer, J. Falco and K. Kent, “Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security—Recommendations of the National Institute of Standards and Technology,” Special Publication 800-82, Initial Public Draft, Sept. 2006.

5 D. Jiao and J. Sun, “Real-Time Visualization of Geo-Sensor Data Based on the Protocol-Coupling Symbol Construction Method,” ISPRS International Journal of Geo-Information, vol. 7, no. 12, p. 460, Nov. 2018.

6 X. Lu, "Supervisory Control and Data Acquisition System Design for CO_2 Enhanced Oil Recovery," Technical Report No. UCB/EECS-2014-123. Master of Engineering Thesis, EECS Department, University of California, Berkeley, CA, USA, 21 May 2014.

7 A. Sajid, H. Abbas and K. Saleem, "Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges,".

IEEE Access, vol. 4, pp. 1375-1384, 2016. doi: 10.1109/ACCESS.2016.2549047.

8 A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,". IEEE Communications Surveys and Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourth quarter 2015. doi: 10.1109/COMST.2015.2444095.

9 A. M. Grilo, J. Chen, M. Díaz, D. Garrido and A. Casaca, "An Integrated WSAN and SCADA System for Monitoring a Critical Infrastructure," in IEEE Transactions on Industrial Informatics, vol. 10, no. 3, pp. 1755-1764, Aug. 2014. doi: 10.1109/TII.2014.2322818.

10 S. L. Jayasinghe, “SCADA System for Remote Control and Monitoring of Grid Connected Inverters”, Master of Engineering Thesis, Memorial University of Newfoundland, Canada, May 2018.

11 M. Regula, A. Otcenasova, M. Roch, R. Bodnar and M. Repak, "SCADA system with power quality monitoring in Smart Grid model," 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, 2016, pp. 1-5. doi: 10.1109/EEEIC.2016.7555577.

12 X. Zhaoxia, G. Zhijun, J. M. Guerrero and F. Hongwei, "SCADA system for islanded DC microgrids," IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, 2017, pp. 2669-2674. doi: 10.1109/IECON.2017.8216449.

13 R. M. J. Rathnayaka and K. Hemapala, "Developing of scalable SCADA in view of acquiring multi-protocol smart grid devices," 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2016, pp. 182-187.

doi: 10.1109/AEEICB.2016.7538269.

14 T. Sultana and K. A. Wahid, "Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things," IEEE Access, vol. 7, pp. 41607-41624, 2019. doi: 10.1109/ACCESS.2019.2907525

15 N. Moustafa, B. Turnbull and K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4815-4830, June 2019. doi: 10.1109/JIOT.2018.2871719

16 Chine, Willard, et al. “Fault detection method for grid-connected photovoltaic plants.” Renewable Energy, vol.66 no. 1, 2014, pp.99–110.

17 Moreno-Garcia, Isabel M. et al. “Real-Time Monitoring System for a Utility-Scale

18 Photovoltaic Power Plant.” Ed. Albert M. K. Cheng. Sensors (Basel, Switzerland) vol. 16, no.6, 2016, pp.1-25.

19 Karoly, Ronay and Christina Dragos Dumitri. “Management of a Power System Based Renewable Energy.” Procedia Technology, vol.12, no.1, 2014, pp. 693 – 697.

20 Manzano, Santiago, et al. “An Overview Of Remote Monitoring PV systems: Acquisition, Storages, Processing And Publication Of Real-Time Data Based On Cloud Computing.” Academia. 2015.

21 Schreck, Tom and Petit Agarwal. “SCADA Versus Web-Based Monitoring.” Ausenco. 2014.

22 Barron-Gafford, Greg A., et al. "The Photovoltaic Heat Island

Effect: Larger solar power plants increase local temperatures." Scientific reports 6 (2016): 35070.

23 Byarak, Gokay and Mehmet Cebeci. “A PV Based Automation System for Fish Farms: An Application Study .” Firat University. 2016.

24 Wang, Yu. “Algorithms for Optimal Energy Management in the Smart Grid.” Auburn University. 2015.

25 Mills-Price, Michael, Michael Rourke, and Darrin Kite. "Adaptive Control

26 Strategies and Communications for Utility Integration of Photovoltaic Solar Sites." proceedings of the 2014 Power and Energy Automation Conference, Spokane, WA. 2014.

27 Kareli, Guatam. “Scada & Monitoring Systems Used In Solar Power Plants In India , What Is Scada For Solar Power Plants?” 2015.

28 Satitpanyapan, Ranon. “SCADA and Monitoring for Solar Energy Plant.” Graphical Achievement Design Achievement Award. 2017.

29 Barnicha, Fatima Ezzahra, Ahmed Khallaayoun and Rachid Lghoul. “Smart Home Energy Management System Monitoring and Control of Appliances Using an Arduino Based Network in the context of a Micro-grid.” Al Akhamethodan University. 2015.

30 Hu Guozhen, Cai tao, Chen Changsong, Duan Shanxu, “Solutions for SCADA systemcommunication reliability in photovoltaic power plants”, Power electronics and motion controlconference,2482-2485, 2009.

31 Review of SCADA system for photovoltaic power plant, April 2018.

32 R. Palma-Behnke, D.Ortiz, L.Reyes, G.Jimenez-Estevez, N .Garrido,

“Social SCADA and demandresponse for sustainable isolated microgrids”, Innovative smart grid technologies (ISGT),2012IEEE PES, 1-1,2012.

33 Vlad-Cristian Georgescu, “SCADA software used in dispatch centre for photovoltaic parks”,Electronics, computers and artificial intelligence (ECAI), 2014 6th international conference on, 1-4,2014.

34 Petru-Claudiu Pasc, Cristian-Dragos Dumitru, “SCADA system for solar MPPT controllermonitoring”,9th international conference interdisciplinarity in engineering, INTER-ENG 2015, 8-9 october 2015

35 XIAO Zhao xia, Guo Zhijun ,Josep M. Guerrero, Fang Hongwei, “SCADA system for islanded DCMicrogrids”, Industrial electronics society, IECON 2017-43rd annual conference of the IEEE,2669-2674, 2017.

36 Ao Dong, Yingying Zhao, Xiwei Liu,Qi Liu, Dahai Kang , “Fault diagnosis and classification inphotovoltaic systems using SCADA data”, Sensing ,diagnostics, prognostics and control (SDPC),2017 international conference on, 117-122,2017.

37 Performance and Evolution of Grid Connected To 5MW Solar Photovoltaic Plant in Shivanasamudra. International Journal of Research in Advent Technology, Vol.3, No.1, January 2015.

38 Alhasnawi, B.N.; Jasim, B.H. SCADA controlled smart home using Raspberry Pi3. In Proceedings of the 2018 InternationalConference on Advance of Sustainable Engineering and Its Ap-plication (ICASEA), Wasit-Kut, Iraq, 14–15 March 2018.

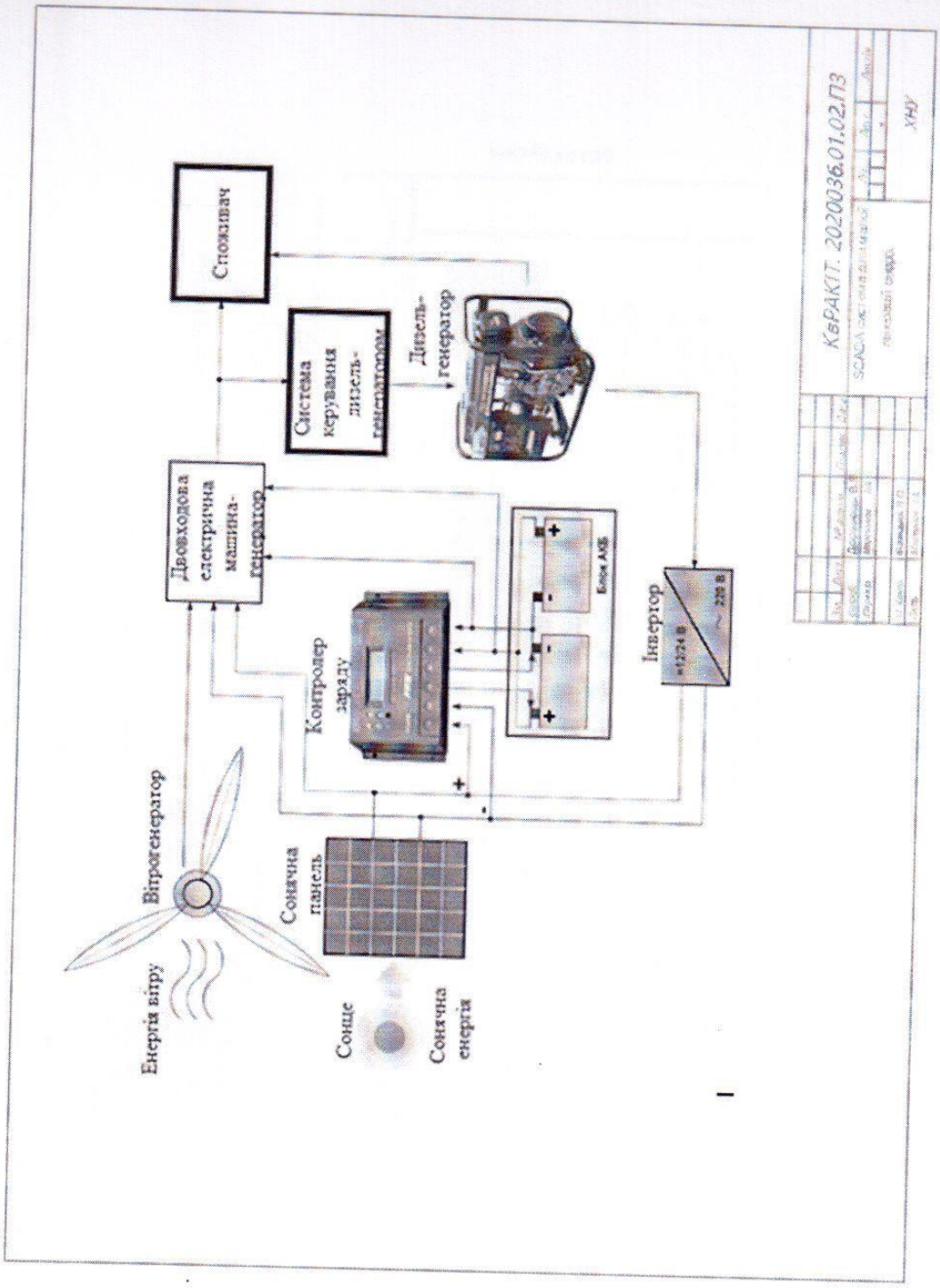
39 Valenciaga F, Puleston PF. Supervisor Control for a Stand-Alone Hybrid Generation System Using Wind and Photovoltaic Energy. IEEE Trans. on Energy Conversion. 2005; 20(2): 398-405.

40 Soetedjo A, Lomi A, Nakhoda YI, Tosadu YP. Combining Web SCADA Software and Matlab-Simulink for Studying Wind-PV-Battery Power Systems. International Journal of Computer Science Issues.2013; 10(2): 51-57.

ДОДАТКИ

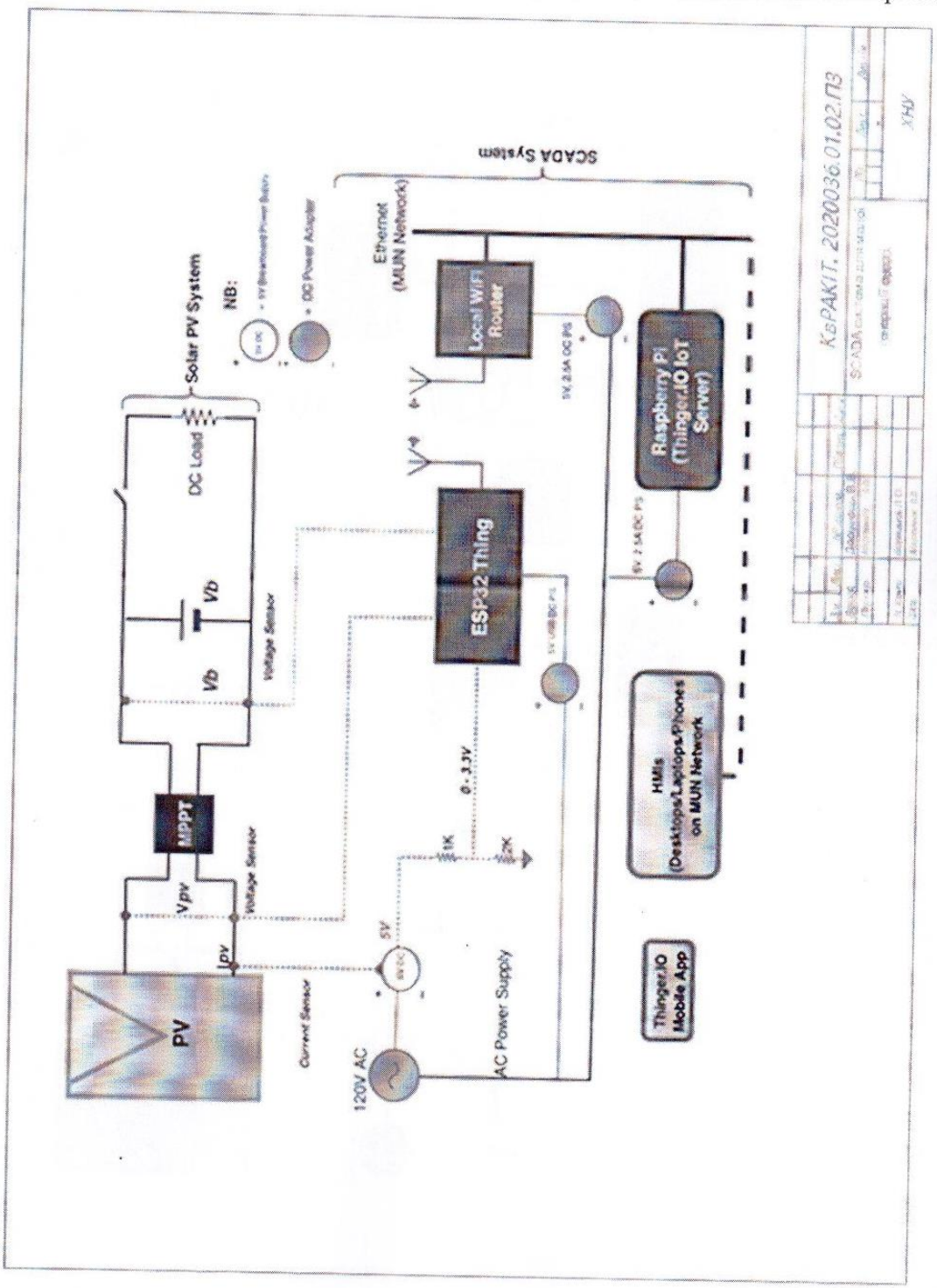
Додаток А

Структурна схема системи малої генерації енергії



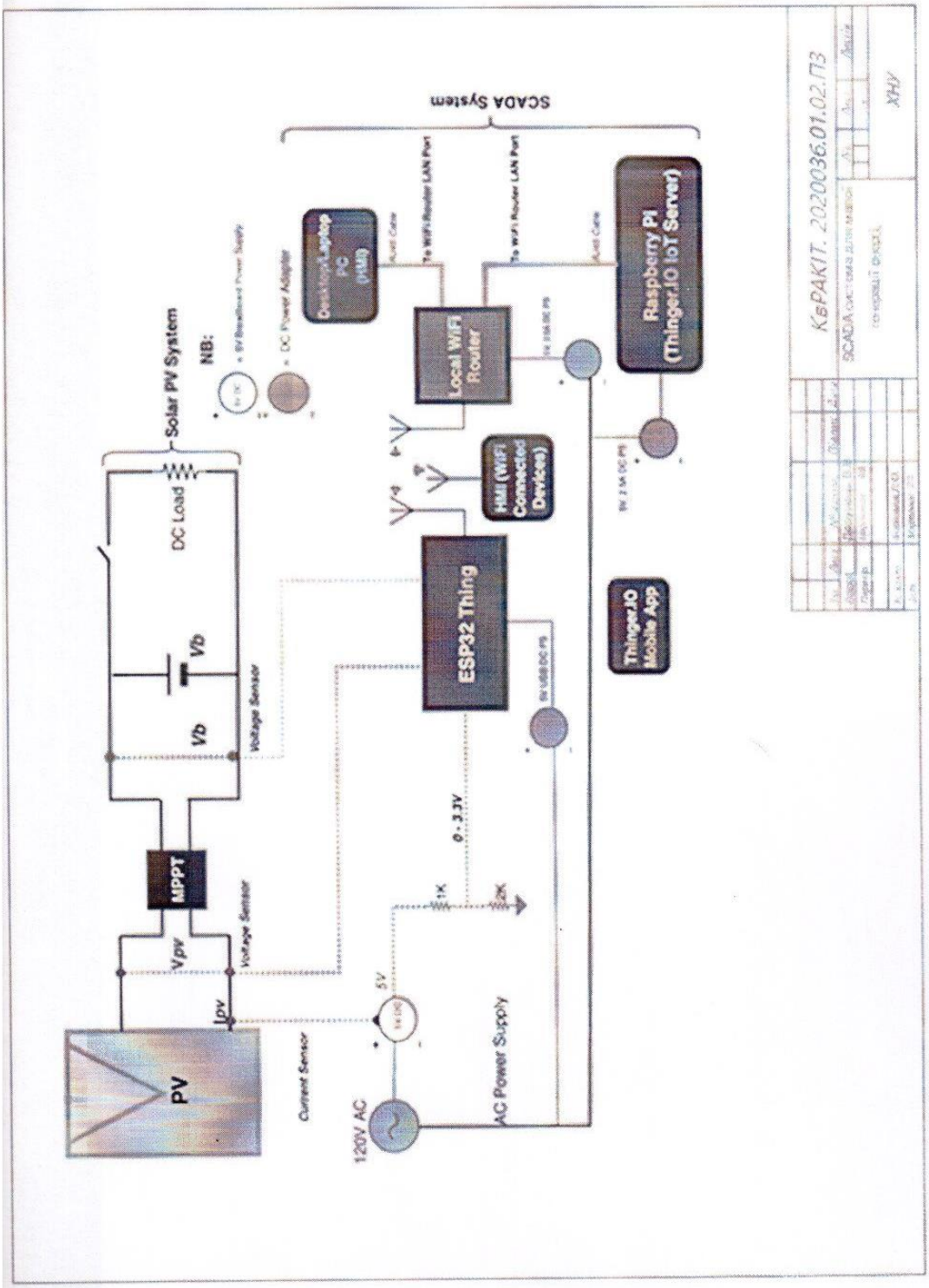
Додаток Б

Перша конфігурація SCADA системи малої генерації енергії для сонячні електростанції



Додаток В

ель 2 SCADA автоматизованої системи керування розподіленою генерацією енергії



UNICHECK
by Turnitin

Ім'я користувача:
Кафедра АКІТіТК

Дата перевірки:
19.06.2023 08:35:06 EEST

Дата звіту:
19.06.2023 08:43:20 EEST

ID перевірки:
1015640271

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005862

Назва документа: **Derunec**

Кількість сторінок: **65** Кількість слів: **11306** Кількість символів: **85838** Розмір файлу: **1.99 MB** ID файлу: **1015286583**

515 слів позначені як "вилучені" та не враховуються у підрахунку слів

3.76%
Схожість

Найбільша схожість: **1.91%** з джерелом з Бібліотеки (ID файлу: **1015268066**)

2.88% Джерела з Інтернету | 104 ..... Сторінка 67

2.44% Джерела з Бібліотеки | 16 ..... Сторінка 68

0.46% Цитат

Цитати | 2 ..... Сторінка 69

Посилання | 1 ..... Сторінка 69

0%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за **8 слів** та **0%**)

Немає вилучених Інтернет-джерел

0% Вилученого тексту з Бібліотеки | 2 ..... Сторінка 69

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 14%**

ID: 116853 Назва: БКР SCADA система для малої генерації енергії Додано в БД: 2023-06-19 Автора: Владислав ДЕРУНЕЦЬ Керівники: Валерій МАРТИНЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	73594	486	2451 (3%)	34 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Дерунець Владислав Васильович

Тема: SCADA система для малої генерації енергії

Спеціальність: 151 «Автоматизація та комп'ютерно-інтегровані технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 65

1. Короткий зміст роботи та прийнятих рішень: розроблено SCADA систему для малої генерації енергії, в SCADA системі малої генерації енергії для сонячної електростанції реалізовано API пристрою на основі MQTT, що підтримується протоколом зв'язку MQTT.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обґрунтована актуальність кваліфікаційної роботи, сформульовано мету та задачі кваліфікаційної роботи, відображено її практичне значення.
В першому розділі були розглянуті наступні питання: особливості систем малої генерації електроенергії, принципи побудови SCADA систем, Особливості SCADA систем з відкритим кодом. У другому розділі проведено аналіз вимог технічного завдання та розроблена структурна схема системи малої генерації енергії. Розроблено SCADA систему малої генерації енергії для сонячні електростанції. Використано локальний сервер Thinger.IO, який отримує напругу PV, струм PV, виміряну потужність PV та напругу акумуляторної батареї від підключеного пристрою ESP32 у хмарній консолі та автоматично записує дані на створені інформаційні панелі та сегменти даних у задані швидкості передачі даних. У третьому розділі розроблено програмне забезпечення SCADA системи малої генерації енергії для сонячні електростанції. Використано технологію Інтернету речей (IoT) для SCADA системи малої генерації енергії для сонячні електростанції.

4. Позитивні сторони роботи: оператор може віддалено візуалізувати (моніторити) отримані дані на Cloud Console, натиснувши API пристрою, або перевіривши інформаційні панелі та сегменти даних.

5. Негативні сторони роботи: у роботі недостатньо уваги приділяється огляду існуючих технічних рішень

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

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

8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: задовільно(3,50/D)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Нейман Віталій Станіславович, к.т.н, доцент кафедри машин і апаратів, електромеханічних та енергетичних систем ХНУ

"19" 06 2023 р. (підпис)

Завідувачу кафедри АКІТтаР
д-ру техн.наук, проф. Мартинюку В.В.

Дерунець В.В.
ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи АКІТс-20-1

ЗАЯВА

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

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

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

06.06.2023
дата

підпис

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ АВТОМАТИЗАЦІЇ, КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: SCADA система для малої генерації енергії
Автор: Дерунець Владислав Васильович
Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітня програма: Освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології»
Науковий керівник: Мартинюк Валерій Володимирович, доктор технічних наук, дпрофесор
Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	**відповідає**
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

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

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальновживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/ схожості, складає 3,76% і адресується до 120 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 19.06.23р.

Завідувач кафедри _____ Валерій МАРТИНЮК

Гарант освітньої програми _____ Юрій ФОРКУН

Керівник кваліфікаційної роботи _____ Валерій МАРТИНЮК