

МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА
на тему:

РОЗРОБКА СИСТЕМИ КЕРУВАННЯ СОНЯЧНОЮ БАТАРЕЄЮ НА
БАЗІ RASBERRY PI

Здобувача вищої освіти першого (бакалаврського) рівня,
4 курсу, групи КБ-19-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за
галузями)»


(підпис)

Артеменко Максим Олексійович
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Редич Олександр
Володимирович, к.е.н. доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Керівник освітньо-
професійної програми


(підпис)

Федотова - Півень Ірина
Миколаївна, к.т.н. доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Допущено до захисту за рішенням кафедри
протокол від «26» травня 2023 № 13

Завідувач кафедри


(підпис)

Філоненко Михайло
Миколайович, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене
звання)

Регистрація № 5

05.06.2023
(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту
(до захисту, на доопрацювання)

29.05.2023
(дата)



(підпис)

Результат захисту

5 (відмінно)
(оцінка)

14.06.2023
(дата)



(підпис)

Голова комісії:



(підпис)

В.М. Франчук

Члени комісії:



(підпис)

М.М. Філоненко



(підпис)

О.В. Редич



(підпис)

Є.В. Котух

БЛАНК ОЦІНЮВАННЯ ВИПУСКНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	29		14.06.2023 Франчук В.М.  Філоненко М.М. 
2. Дотримання вимог щодо оформлення	від 0 до 10	9		Редич О.В. 
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	5		Котух С.В.  Суховерхівка Л.П. 
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	47		

Оцінювання кваліфікаційної роботи науковим керівником - 46 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підписи)
93	A	5 (відмінно)	14.06.2023	Франчук В.М.  Філоненко М.М.  Редич О.В.  Котух С.В.  Суховерхівка Л.П. 

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	3 (задовільно)
60-69	D	
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційні управляючі системи та технології (за галузями)»

Затверджую
Завідувач кафедри
 М.М. Філоненко
« 20 » 11 2022 р.

Завдання на кваліфікаційну роботу (проект)
здобувачу вищої освіти

Артеменку Максиму Олексійовичу

1. Тема кваліфікаційної роботи:

«Розробка системи керування сонячною батареєю на базі Raspberry Pi»

Затверджена наказом ректора від «15» листопада 2022 № 1205 (зі змінами наказ від «04» квітня 2023 № 361)

2. **Строк здачі закінченої роботи:** «20» травня 2023

3. **Мета роботи:** розробити систему керування сонячною батареєю на базі Raspberry Pi

Об'єкт дослідження: системи керування сонячними батареями

Предмет дослідження: системи управління на базі Raspberry Pi та симулятор Packet Tracer.

Орієнтовна база дослідження: підприємства та установи з інфраструктурою інтернету речей

4. **Вихідні дані до роботи:** існуючі реалізації систем керування сонячними батареями

5. **Перелік графічного матеріалу:** рисунки та графіки, формули, що пояснюють суть застосованих методів та засобів. Структурні схеми та таблиці, графіки результатів моделювання.

6. **Зміст випускної кваліфікаційної (дипломної) роботи (перелік питань за кожним розділом) :**

Розділ 1. Огляд існуючих рішень систем керування сонячними батареями

1.1. Огляд технології сонячних батарей;

1.2. Огляд апаратних реалізацій системи керування сонячною батареєю ;

1.3. Обґрунтування вибору технічної моделі.

Розділ 2. Огляд технологій моделювання систем

2.1. Огляд програмного забезпечення управління контролерами;

2.2. Огляд бібліотек Python для програмування управління СБ;

2.3. Розробка аналітичної та технічної моделі УСБ

Розділ 3. Розробка моделі у середовищі Packet Tracer

3.1. Моделювання системи ;

3.2. Розробка алгоритму та програмування компонентів УСБ;

7. Календарний план виконання роботи

№ п/п	Назва етапів випускної кваліфікаційної (дипломної) роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури ДР	До 15.11.2022 р.	« 18 » 11 2022
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	26. 03. 2023 р.	« 01 » 04 2023
3	II розділ (аналіз експериментальної частини про і методики досліджень)	20. 04.2023 р.	« 25 » 04 2023
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	15. 05.2023 р.	« 16 » 05 2023
5	Висновки	18.05.2023 р.	« 18 » 05 2023
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	20.05.2023 р.	« 25 » 05 2023
7	Отримання відгуку керівника	03.06.2023 р.	« 03 » 06 2023
8	Отримання зовнішньої рецензії	05.06.2023р.	« 05 » 06 2023

Дата видачі завдання « 20 » листопада 2022 р.

Керівник дипломної роботи


(підпис) О.В. Редич

Керівник освітньо-професійної програми


(підпис) І.М. Федотова-Пі

Завдання до виконання отримав
здобувач вищої освіти


(підпис) М.О. Артеменко

АНОТАЦІЯ

«Розробка системи керування сонячною батареєю на базі Raspberry Pi»

Метою випускної кваліфікаційної (дипломної) роботи є створення інформаційної системи реального часу для моніторингу параметрів роботи домашньої сонячної електростанції, що надає можливість покращити комунікацію між апаратним забезпеченням та користувачами і підвищити ефективність людських ресурсів.

Завдання, що були вирішені при виконанні роботи є вибір методу, який дозволить підвищити надійність та універсальність системи.

Об'єктом дослідження є основні параметри методів оцінки деградації сонячних панелей для віддаленої передачі даних.

Інформаційною базою дослідження є існуючі методи оцінки та діагностики сонячних панелей.

За результатами дослідження демонструють перш за все важливість оцінки системи в загальному, а також те, наскільки правильно підібрано склад системи, та чи раціонально застосовані певні методи й підходи до проектування та дослідження системи.

Дипломна робота містить 59 сторінок, 12 рисунків, 1 таблиця і список використаних джерел з 41 найменувань.

Ключові слова: інтеграцію сонячних панелей з Raspberry Pi, створення автоматизованого контролера, який оптимально використовує сонячну енергію, та забезпечення моніторингу та віддаленого управління системою. Ця розробка передбачає використання датчиків для вимірювання рівня сонячної інсоляції, температури та заряду батареї, а також розробку алгоритмів управління, які забезпечують оптимальну ефективність роботи системи.

Рік виконання дипломної роботи: 2022-2023

Рік захисту роботи: 2023.

ABSTRACT

Development of a solar battery control system based on Raspberry Pi

The purpose of the final qualification (diploma) work is to create a real-time information system for monitoring the parameters of a home solar power plant, which makes it possible to improve communication between hardware and users and increase the efficiency of human resources.

The tasks that were solved in the course of the work are to choose a method that will increase the reliability and versatility of the system.

The object of study is the main parameters of methods for assessing the degradation of solar panels for remote data transmission.

The information base of the study is the existing methods for assessing and diagnosing solar panels.

The results of the study demonstrate, first of all, the importance of evaluating the system in general, as well as how well the system composition is selected, and whether certain methods and approaches to system design and research are rationally applied.

The thesis contains 59 pages, 12 figures, 1 table and a list of 41 references.

Key words: integration of solar panels with Raspberry Pi, creation of an automated controller that optimally utilizes solar energy, and provision of monitoring and remote control of the system. This development involves the use of sensors to measure solar insolation, temperature, and battery charge, as well as the development of control algorithms that ensure optimal system efficiency.

Year of thesis completion: 2022-2023

Year of thesis defense: 2023.

Зміст

ВСТУП.....	3
1.1 Огляд технології сонячних батарей.....	5
1.2 Огляд апаратних реалізацій системи керування сонячною батареєю..	8
1.3. Обґрунтування вибору технічної моделі.....	18
РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ МОДЕЛЮВАННЯ СИСТЕМ.....	21
2.1 Огляд програмного забезпечення управління контролерами.....	21
2.2 Огляд бібліотек Python для програмування управління СБ.....	27
2.3 Розробка аналітичної та технічної моделі УСБ.....	38
РОЗДІЛ 3. РОЗРОБКА МОДЕЛІ У СЕРЕДОВИЩІ RASCKET TRACER.....	44
3.1 Моделювання системи.....	44
3.2 Розробка алгоритму та програмування компонентів УСБ.....	49
ВИСНОВОК.....	53
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	55

ВСТУП

Зважаючи на постійний ріст населення і його споживання енергоносіїв, спостерігається сучасна тенденція до зростання електрифікації в пропорційній мірі. У зв'язку з цим, в майбутньому може відбутися значне збільшення темпів виробництва та споживання електричної енергії. Сонячні панелі є одним з перспективних та безпечних джерел виробництва, які не мають негативного впливу на навколишнє середовище, і отримують все більшу актуальність з кожним днем.

Основною метою дослідження є розробка програмно-апаратної системи, яка здатна ефективно керувати збереженням та використанням енергії, отриманої з сонячних батареї. Така система має на меті оптимізувати використання сонячної енергії, забезпечуючи постійний доступ до електричної енергії та зменшуючи залежність від зовнішнього електромережі.

«Основний функціонал системи може включати зчитування стану заряду батареї, керування зарядкою та розрядкою, а також моніторинг енергопотреб. Використовуючи вбудований аналого-цифровий перетворювач (ADC) на Raspberry Pi, система може зчитувати дані про рівень заряду сонячної батареї та визначати оптимальні параметри для зарядки та розрядки»[1].

Стану проблеми управління сонячними батареями, який сучасній літературі та наукових джерелах вже проведений ряд досліджень, спрямованих на управління сонячними батареями. Існують різні методи та системи управління, які мають свої переваги та недоліки.

Наприклад, використання мікроконтролерів Raspberry Pi та симулятора Packet Tracer вже знайшли застосування в цій галузі. Проте, досить незначна кількість досліджень спрямована на розробку комплексної системи керування сонячною батареєю, що використовує можливості Raspberry Pi та інтегрується з симулятором Packet Tracer.

Об'єктом дослідження є система керування сонячною батареєю на базі

Raspberry Pi.

Предмет дослідження полягає у розробці алгоритмів та програмного забезпечення для оптимального управління сонячними панелями, забезпечення точного вирівнювання панелей з сонцем та максимального отримання сонячної енергії на базі Raspberry Pi та використання симулятора Packet Tracer для моделювання та випробування системи.

Отримані результати демонструють успішну реалізацію системи управління сонячною батареєю та підтверджують її ефективність у використанні сонячної енергії. Дослідження відкривають перспективи для подальшого вдосконалення системи, оптимізації алгоритмів та розширення їх функціональності.

Методи дослідження для досягнення поставленої мети ми використовуємо різні методи дослідження, включаючи загальнонаукові підходи, такі як індукція, дедукція, системний аналіз та експериментальні дослідження.

Ми моделюємо та випробовуємо розроблену систему на базі Raspberry Pi та симулятора Packet Tracer, а також аналізуємо результати для досягнення оптимального управління сонячною батареєю.

База досліджень в нашому дослідженні ми використовуємо існуючі приклади управління сонячними батареями як базу для аналізу та порівняння. Це дозволяє нам оцінити ефективність розробленої системи управління на базі Raspberry Pi та симулятора Packet Tracer в порівнянні з існуючими рішеннями.

Практична значимість отриманих результатів нашого дослідження матимуть практичну значимість, оскільки розроблена система управління сонячною батареєю на базі Raspberry Pi та симулятора Packet Tracer дозволить ефективно використовувати сонячну енергію та підвищити продуктивність та тривалість роботи батареї.

Мій особистий внесок у дослідження полягає у розробці алгоритмів управління, створенні програмного забезпечення, проведенні експериментів та аналізі отриманих результатів.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМ КЕРУВАННЯ СОНЯЧНИМИ БАТАРЕЯМИ

1.1 Огляд технології сонячних батарей

Упродовж багаторічної історії розвитку фотовольтаїки сонячної енергетики було розроблено низку різноманітних видів сонячних батарей, які охоплюють три покоління технологій: кремнієві - перше покоління, тонкоплівкові - друге покоління і органічні- третє покоління. Типи сонячних елементів СЕ другого і третього поколінь різняться в залежності від ефективності, вартості, довготривалості, підходів щодо отримання матеріалів та їхньої модифікації. Основну увагу приділяють матеріалам, що поглинають світло: від неорганічних моно і полікристалічних напівпровідників до органічних, полімерних і гібридних сполук. Лише завдяки сукупності всіх необхідних компонентів сонячної батареї можливо досягти необхідної ефективності перетворення енергії світла в електричну.

«Останнім часом на ринку сонячних батарей з'явилися клітини з перовських метал-галогенідів, які мають високу ефективність і можуть бути вироблені за допомогою простих методів. Клітини з перовських метал-галогенідів мають потенціал стати найбільш ефективним типом сонячних батарей. Ці клітини складаються з органічних та неорганічних матеріалів, таких як металеві галогеніди та етилендіамонієві катіони. Клітини з перовських метал-галогенідів мають високу ефективність перетворення сонячної енергії у порівнянні з іншими типами сонячних батарей, такими як кристалічний кремній та тонкі плівки»[6].

Нагадаємо, управління сонячними батареями має проблеми, з якими можна стикнутися:

- 1) Недостатній збір сонячної енергії: Якщо сонячні батареї не збирають достатню кількість енергії, можуть бути такі причини:
- 2) Забруднення панелей: Регулярно очищайте панелі від пилу, бруду або снігу. Неправильне розташування: Переконайтеся, що панелі зорієнтовані на південь і немає перешкод, що блокують сонячне

випромінювання.

3) Пошкоджені або застарілі панелі: Перевірте, чи немає пошкоджень або зношених елементів і, якщо необхідно, замініть їх.

4) Проблеми зі зберіганням енергії: Якщо сонячна батарея не зберігає або втрачає енергію, можливі наступні причини:

— Пошкодження акумуляторів: Перевірте стан акумуляторів і замініть їх, якщо вони пошкоджені або втратили свою ефективність.

— Неправильні налаштування контролера заряду: Переконайтеся, що контролер заряду налаштований правильно для оптимального заряду та розряду акумуляторів.

— Енерговтрати в системі: Перевірте всі електричні з'єднання на предмет витоків або проблем зі з'єднанням, що можуть призводити до втрати енергії.

— Управління зарядом і розрядом: Контролюйте процеси заряду та розряду батарей, щоб підтримувати їх в оптимальному стані.

— Батареї з надмірним розрядом: Установіть належні параметри заряду та розряду в контролері заряду, щоб уникнути надмірного розряду батарей. Розряджайте батареї тільки до рекомендованого рівня, щоб забезпечити їх тривалий термін служби.

5) Недостатня чи перевищена потужність: Переконайтеся, що сонячні панелі та батареї мають достатню потужність для вимог вашої системи. Якщо потужність недостатня, вам може знадобитися додаткове обладнання, а якщо потужність перевищена, можна розглянути додаткові батареї або розширення системи.

«Окрім матеріалів, що використовуються для створення фотовольтаїчних клітин, існують різні технології виготовлення сонячних батарей, такі як монохроматичні та мультикристалльні. Монохроматичні сонячні батареї мають один кристалічний блок, тоді як мультикристалльні

сонячні батареї складаються з багатьох менших кристалів. Мультикристалльні сонячні батареї менш ефективні, але дешевші у виробництві. Іншою важливою характеристикою сонячних батарей є ефективність» [5].

«Підвищення ефективності фотоелектричних панелей досягається, зокрема, за рахунок поєднання різних технологій та комбінування матеріалів. Одним з перспективних на сьогодні виглядає тандемна структура комірки c-Si/Perovskite, яку впроваджує компанія Oxford PV, що перевищила останній світовий рекорд ефективності сонячних панелей — 29,52% наприкінці 2020р. (потенціал ефективності складає близько 35 %). На даний час Oxford PV створює свою першу комерційну виробничу установку потужністю 125 МВт для тандемних комірок c-Si/перовскіт на виробничих потужностях у Німеччині (орієнтовно перші продажі відбудуться у 2022р.)»[4].

Нарешті, потрібно звернути увагу на те, що сонячні батареї можуть мати різні розміри та форми, що дозволяє їх використовувати у різних застосуваннях. Наприклад, маленькі сонячні батареї можуть бути використані для живлення пристроїв з низьким споживанням електроенергії, таких як калькулятори або світлодіодні ліхтари, тоді як великі сонячні батареї можуть бути встановлені на дахах будівель для генерації електроенергії для житлових будинків або комерційних приміщень.

Крім того, важливо пам'ятати, що використання сонячних батарей має багато переваг. Сонячні батареї є екологічно чистим джерелом енергії, вони не викидають шкідливих викидів в атмосферу, як традиційні джерела енергії на базі вугілля або газу. Вони також не потребують забруднюючих матеріалів для виробництва, які використовуються в інших типах батарей, таких як свинцево-кислотні батареї.

Вимірювання та моделювання сонячного випромінювання є ключовим етапом проектування та ефективної експлуатації сонячної електростанції. Це необхідно для визначення рівня виробництва електроенергії та оптимального розміщення сонячних панелей. Нижче розглянемо детальніше процес вимірювання та моделювання сонячного випромінювання.

Вимірювання сонячного випромінювання проводять за допомогою пірометрів, пірографів, піроелектричних приладів тощо. Ці інструменти дозволяють вимірювати рівень сонячної радіації, що падає на поверхню Землі в різних точках та різний час доби. Результати вимірювання збираються в спеціальні бази даних, які використовуються для створення моделей сонячного випромінювання.

«Моделювання сонячного випромінювання зазвичай використовує математичні моделі, які враховують різні фактори, такі як місце розташування сонячної електростанції, нахил поверхні землі, орієнтація сонячних панелей та кліматичні умови. Такі моделі можуть бути використані для прогнозування виробництва електроенергії на різних часових інтервалах, від годин до декількох років. Окрім того, існують спеціальні програмні засоби, які дозволяють моделювати сонячне випромінювання. Наприклад, програма PVsyst дозволяє розраховувати виробництво електроенергії для різних конфігурацій сонячних електростанцій»[2].

Ці програмні засоби для моделювання сонячного випромінювання дозволяють розраховувати ефективність сонячних електростанцій для різних конфігурацій та умов, таких як нахил та орієнтація сонячних панелей, місце розташування, погода та час доби. При розрахунках програма враховує різні фактори, такі як втрати енергії, які виникають під час конвертації енергії від сонячних панелей до електричного струму, та втрати через температуру. Крім того, програма дозволяє враховувати взаємодію між сонячними панелями та враховувати відбивання світла від поверхні Землі.

Отримані результати дозволяють розрахувати, скільки електроенергії можна виробити з сонячних панелей, які встановлені на певній території, та скільки це буде коштувати. Це дозволяє зробити ефективний вибір конфігурації сонячних електростанцій, що допоможе зменшити витрати на виробництво електроенергії та збільшити її виробництво з допомогою сонячних панелей.

1.2 Огляд апаратних реалізацій системи керування сонячною батареєю

Система керування сонячною батареєю включає в себе контролер заряду, інвертор і батарею. Контролер заряду регулює потік електричного струму між сонячними панелями і батареєю, щоб запобігти перевантаженню і перерозряду батареї. Інвертор перетворює збережену електричну енергію у змінний струм, щоб його можна було використовувати у домашній електричній мережі.

У залежності від типу сонячної батареї, можуть використовуватися різні типи контролерів заряду. Наприклад, для свинцево-кислотних батарей використовуються прості контролери заряду, які регулюють напругу і струм заряду. Для літій-іонних батарей можуть використовуватися більш складні контролери заряду, які регулюють температуру, струм і напругу заряду.

Щодо апаратної реалізації системи керування сонячною батареєю, то вона може бути реалізована як на основі мікроконтролера, так і на основі програмованого логічного контролера (ПЛК). Для більш складних систем можуть використовуватися ПЛК, що мають більш високу продуктивність і забезпечують більш точне керування системою.

Загалом, апаратна реалізація системи керування сонячною батареєю залежить від типу батареї і вимог щодо функціональності системи. Якщо вам потрібні більш конкретні деталі щодо реалізації системи, то вам, можливо,

буде корисно звернутися до документації від виробника обладнання або до фахівців з енергетичного обладнання.

Також важливим аспектом апаратної реалізації системи керування сонячною батареєю є захист системи від перенапруги, перевантаження та короткого замикання. Для цього можуть використовуватися різні захисні пристрої, такі як захисні вимикачі, пристрої автоматичного відключення, перевантажувальні захисні діоди та інші.

Крім того, система керування сонячною батареєю може бути підключена до мережі з різними типами зв'язку, такими як RS-485, Modbus, CAN і т. д., щоб забезпечити зв'язок з моніторинговою системою або з контрольно-управляючим центром.

У сучасних системах керування сонячною батареєю також можуть бути реалізовані функції підтримки схем заряду з ефективним використанням електроенергії, оптимізації заряду з урахуванням різних часових періодів та інших функцій, що дозволяють забезпечити ефективне використання сонячної енергії та збереження енергетичних ресурсів.

Апаратна реалізація системи керування сонячною батареєю залежить від вимог до функціональності системи, типу сонячної батареї та виробника обладнання. Для отримання більш конкретної інформації щодо реалізації системи керування сонячною батареєю вам слід звернутися до фахівців з енергетичного обладнання або виробника обладнання.

Використання динамічних систем стеження за сонцем має кілька переваг:

- Збільшення ефективності: Динамічні системи стеження за сонцем можуть підтримувати сонячні панелі направлені в напрямку сонця на протязі всього дня, що збільшує ефективність виробництва електроенергії.
- Зменшення втрат енергії: Динамічні системи стеження за сонцем дозволяють уникнути втрат енергії, які можуть відбуватися в результаті невірного направлення панелей.
- Зменшення потреби в панелях: Використання динамічних систем стеження за сонцем може дозволити зменшити кількість сонячних панелей, які потрібні для виробництва електроенергії.
- Адаптація до змінних умов: Динамічні системи стеження за сонцем можуть змінювати своє положення в залежності від зміни умов, таких як погода або пора дня.
- Використання земної поверхні: Динамічні системи стеження за сонцем дозволяють використовувати земну поверхню для розміщення сонячних панелей, що збільшує їх ефективність.
- Використання у різних географічних регіонах: Динамічні

системи стеження за сонцем можуть бути використані у різних географічних регіонах, що дозволяє ефективніше використовувати сонячну енергію.

Використання динамічних систем стеження за сонцем є ефективним способом забезпечення оптимального виробництва електроенергії з сонячних панелей, зменшення втрат енергії та збільшення загальної ефективності сонячних енергетичних систем. Однак, використання динамічних систем стеження за сонцем потребує більшої складності в обладнанні та програмуванні системи керування, а також більшої кількості датчиків та приводів для руху сонячних панелей. Також, відповідна система стеження може вимагати більшої кількості енергії для руху панелей в порівнянні з фіксованими системами, що може зменшити загальну ефективність системи.

«Щоб підтвердити ефективність використання трекерів, для прикладу, розглянемо дані про виробіток електроенергії фотоелектричних станцій (ФЕС) з різним типом кріплення протягом року. Дані взяті для 10kW ФЕС, з урахуванням втрат в 24,3%, розташовану під кутом 34° до землі, на території Києва» [9].

Таблиця 1 – Кількість kWh вироблених кожного місяця для ФЕС з різним кріпленням

Місяць	Статична ФЕС	Одноосьова ФЕС	Двоосьова ФЕС
Січень	262	297	306
Лютий	417	494	501
Березень	847	1080	1090
Квітень	1010	1350	1370
Травень	1210	1720	1780
Червень	1160	1620	1670
Липень	1180	1620	1680
Серпень	1090	1480	1520
Вересень	817	1050	1080
Жовтень	591	721	753
Листопад	250	281	297
Грудень	195	218	243
Всього за рік	9029	1193	12290

«Аналізуючи дані з таблиці 1 та рис 1.1 очевидно, що ФЕС з використанням трекерів з осями обертання відчутно ефективніше за статично встановлену. В даному випадку виробіток електроенергії збільшився на

36,1% за рік у порівнянні з ФЕС без використання трекара» [19].

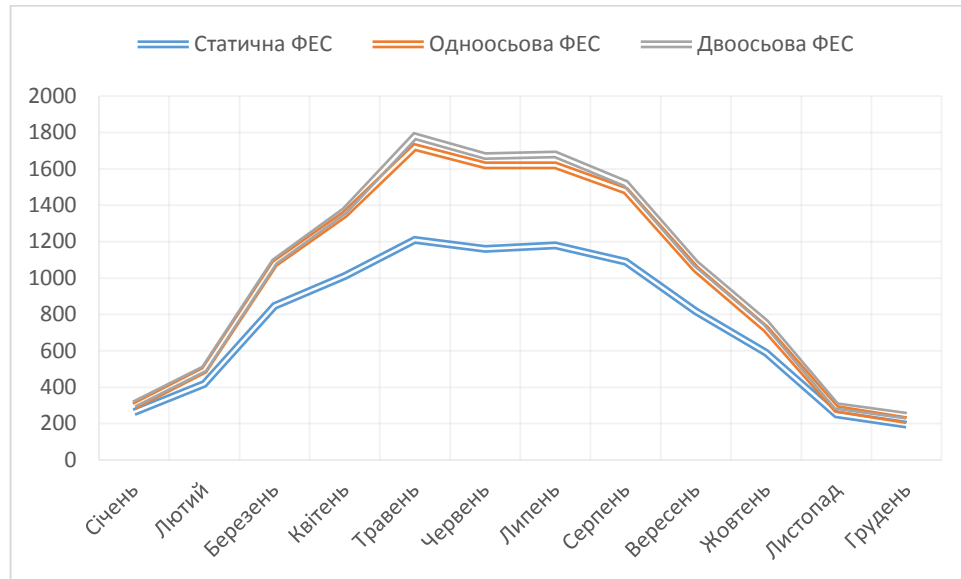


Рис 1.1 Графік вироблення ФЕС із різним кріпленням

Загалом, динамічні системи стеження за сонцем є ефективним способом для забезпечення оптимального виробництва електроенергії з сонячних панелей, особливо в умовах змінної погоди та розташування в різних географічних регіонах. Однак, їх використання може бути складнішим та вимагати більшої кількості ресурсів порівняно з фіксованими системами.

«Сонячний трекара - це пристрій, який встановлюється на сонячну панель і дозволяє їй повертатися відповідно до руху сонця, забезпечуючи максимальний збір сонячної енергії. Сонячний трекара складається з трьох основних компонентів: кріплення, приводу та електронної системи керування.

Кріплення є основою сонячного трекара і відповідає за установку сонячної панелі. Це може бути різні типи кріплення, залежно від типу сонячної панелі та її розміру. Привод відповідає за рух сонячної панелі відповідно до руху сонця. Зазвичай використовуються двигуни або сервоприводи, що керують рухом панелі»[11].

Електронна система керування відповідає за контроль за рухом сонячного трекара. Вона включає в себе датчики, які визначають положення сонця та кути нахилу панелі, а також програмне забезпечення, що керує приводом.

Для ефективної роботи сонячного трекера необхідно точно визначити положення сонця в реальному часі, використовуючи різні датчики, такі як компаси, сенсори нахилу та позиціонування. Для керування приводом використовуються мікроконтролери, які отримують дані з датчиків та розраховують необхідні кути нахилу та повороту панелі для максимального збору сонячної енергії.

Залежно від типу сонячного трекера, можуть використовуватися різні датчики та приводи. Наприклад, одні з сучасних трекерів використовують GPS-датчики та лазерні приводи для точного визначення положення сонця та його руху. Інші трекери можуть використовувати астрономічні алгоритми та сенсори нахилу для визначення положення сонця.

«Є два основних типи сонячних трекерів - одноосні та двоосні. Одноосні трекери можуть рухатися лише в одній площині, тобто вгору і вниз, і вони підходять для тих регіонів, де сонце рухається від сходу до заходу. Двоосні трекери можуть рухатися в двох площинах - вгору і вниз, а також з боку на бік, і вони підходять для регіонів, де сонце рухається вздовж орбіти, змінюючи свій кут до горизонту протягом дня»[27].

Одним з головних переваг сонячних трекерів є їх ефективність. Сонячні панелі, що встановлені на трекерах, можуть збирати від 30% до 50% більше сонячної енергії порівняно з панелями, що зафіксовані на статичних кріпленнях. Крім того, сонячні трекери можуть забезпечити стабільну продуктивність панелей протягом усього дня, навіть в умовах зміни погодних умов.

Іншою перевагою сонячних трекерів є їх здатність пристосовуватися до різних кутів зони сонячного випромінювання. Вони можуть бути встановлені на різних широтах та орієнтовані під різні кути, що дозволяє їм збирати більше енергії навіть в умовах зміни кута сонячної радіації.

Хоча встановлення сонячного трекера може бути витратним заходом, його ефективність та здатність до збору більшої кількості сонячної енергії може допомогти зменшити загальні витрати на енергію в майбутньому. Крім

того, використання сонячних трекерів може бути особливо корисним для великих сонячних ферм, де велика кількість панелей може бути встановлена на масштабному майданчику.

«Однак, на жаль, сонячні трекери також мають свої недоліки. Оскільки вони мають більше рухомих частин, ніж статичні сонячні панелі, вони потребують більше енергії для їхнього руху та підтримки. Це може зменшити їх ефективність та збільшити загальні витрати на енергію. Крім того, їхній рух може вимагати більшої технічної підтримки та обслуговування, що може збільшити загальні витрати на їх використання»[31].

При виборі між статичними та динамічними системами сонячного стеження, слід враховувати кілька факторів, таких як розмір інсталяції, розташування та ефективність панелей, погодні умови, доступність електроенергії та технічної підтримки.

Існує кілька способів керування сонячним трекером. Ось кілька з них:

— Пошук максимуму потужності (MPPT) - цей метод використовується для максимального отримання енергії від сонячних панелей. Він вимірює напругу та струм, що генерується сонячними панелями та налаштовує кут нахилу трекера для досягнення максимальної потужності.

— Інерційний навігаційний система - цей метод використовує гіроскопи та акселерометри для визначення положення трекера. Наприклад, якщо трекер знаходиться в одному положенні, а сонце рухається, то трекер може бути налаштований на відповідний кут для стеження за сонцем.

— Керування з використанням зірок - цей метод використовує зірки для визначення положення сонця. Він використовує два датчики зірок, щоб визначити точне положення сонця та налаштувати кут нахилу трекера для його стеження.

— Керування за допомогою GPS - цей метод використовує GPS-дані для визначення точного положення трекера та часу. Він налаштовує кут нахилу трекера відповідно до географічного положення та часу дня для

стеження за сонцем.

— Ручне керування - цей метод використовується в основному для малих сонячних трекерів, де керування може бути здійснене вручну. Власник може змінювати кут нахилу трекера, відповідно до поточного положення сонця.

— Керування за допомогою штучного інтелекту (AI) - цей метод використовується для максимальної ефективності сонячних трекерів. Він використовує дані про метеорологію, час дня, географічні координати та інші фактори для розрахунку оптимального кута нахилу трекера для стеження за сонцем.

«Продуктивність сонячних панелей залежить не тільки від сонячного випромінювання чи стану і температури навколишнього середовища, але і від кута падіння сонячної радіації на поверхню панелі. Отже, основне завдання трекера – встановити кут нахилу поверхні панелей, орієнтуючи їх точно на Сонце. Протягом дня Сонце рухається по небу зі сходу на захід, тому кут між сонцем і фіксованою поверхнею панелей постійно змінюється. Таким чином, концентрація потужності на фіксованому модулі сонячної батареї менше, ніж у падаючого сонячного світла» [16].

Останнім часом багато робіт було зроблено для того, щоб максимізувати витяг потужності з сонячної панелі. Сонячна система відстеження пропонує практичне технологічне рішення підвищити енергоефективність, що створюють панелі. Широкі дослідження і експерименти показали, що підтримання постійного прямого кута впливу Сонця на поверхню панелей, завдяки системі стеження виробляють до 40% більше потужності ніж фіксовані панелі.

«Як правило, сонячний трекер є пристроєм, що використовується для вирівнювання фотоелементів панелі до сонця. У цьому контексті існують три типи система стеження за сонцем, а саме пасивний трекер, активний трекер і трекер з відкритим циклом. Пасивний трекер використовує два пневмоупори

зі стисненою газовою рідиною, що розміщується відповідно на схід і на захід від відстеження. Однак цей газовий трекер рідко ставить панель безпосередньо на Сонце через непередбачувану атмосферну температура» [4].

Тим не менш, як просування сенсорної технології, датчик освітленості використовується в активній системі відстеження сонця і його розміщують в різних місцях трекера, щоб визначити найкраще положення сонця, завдяки якому електромеханічні засоби управління керують нахилом панелей. Тим не менш, ефект хмари і тіні є основним недоліком системи. Це спричиняє втрату ефективності системи, адже витрачається потужність на привід нахилу панелей в різні положення, під час пошуку максимально оптимального їх розміщення. Крім того, трекер відкритої петлі також зазвичай використовується для відстеження шляху сонця, як показано на рис 1.2.

«Трекер використовує заздалегідь визначену астрономічну базу даних для визначення позиції сонця для будь-якого заданого часу і координати розташування за допомогою мікроконтролера. Крім того, трекер з відкритою петлею оснований на висоті і азимутальній позиції Сонця, отже, ця система не залежить від температури навколишнього середовища і від погоди. Дані обставини дають великі переваги, такі, що вона не виключає небажаного розсіювання потужності і складності. І навпаки, трекер з відкритою петлею також має недоліки, які могли б зменшити ефективність виходу, як приклад, невідповідність трекера під час налаштування через людську помилку» [14].

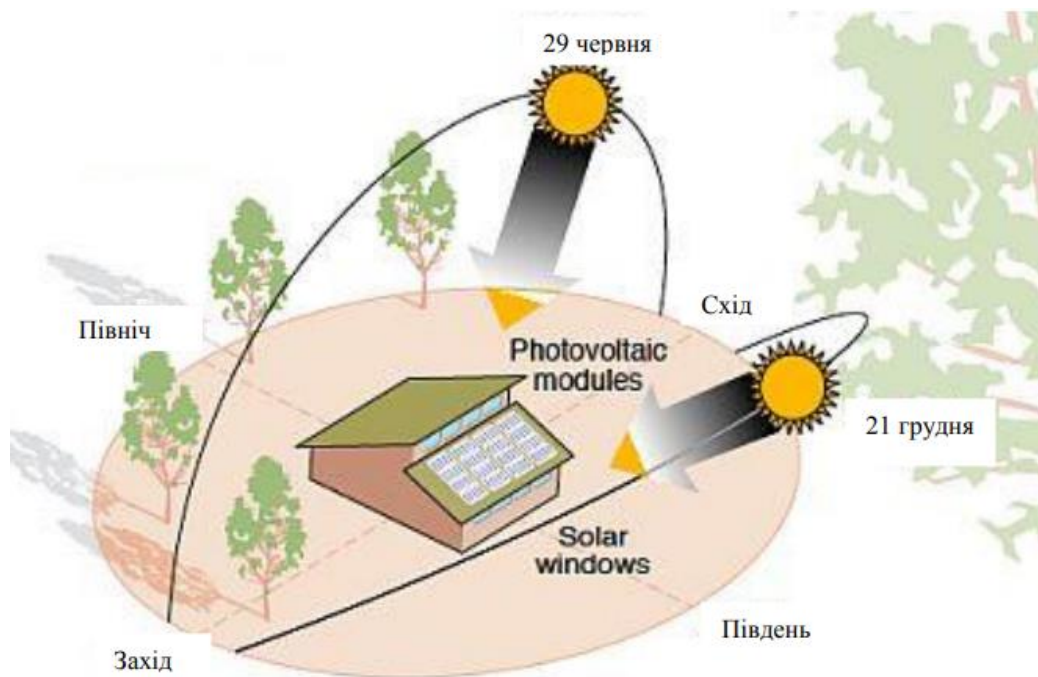


Рис 1.2 Зображення руху Сонця зимою та літом

Крім того, перед розгортанням сонячного трекера база даних сонячної траєкторії встановлюється вручну постачальником або підрядником. Як ми всі знаємо, кожна локація має свій унікальний сонячний шлях, внаслідок чого вона передбачає додаткову втручання людини, що збільшить вартість налаштування або установки. Також буде змінюватися конструкція механічної системи. Як правило, деякі можуть використовувати простий одноосьовий трекер, який лише на 20%, має кращий енергетичний коефіцієнт, ніж статичне розташування панелей.

«Теоретично, також існує система позиціонування сонячних панелей без застосування електромеханічні засоби управління та датчиків світла. Вона базується на зміні кута нахилу панелей завдяки ручній праці. Як правило, ці системи змінюють кут вздовж однієї осі (північ - південь) і виконують це залежності від пори року та безпосередньо висоти сонця над горизонтом. Проте треба враховувати, що якщо ви маєте у власному керування досить потужну СЕС, то для зміни положення панелей вручну потрібно буде затратити дуже багато часу та сил» [7].

Хоча для відносно невеликих СЕС, приріст енергії в порівнянні зі статичним розташуванням панелей буде досить значний і може досягати до

10%. Керування трекером реалізовується за різними концепціями, основні з них:

— Робота системи позиціонування сонячних панелей базується на фотоприймачах. Вони встановлюються на різних сторонах трекеру, та в залежності від положення Сонця отримують дані про їх освітленість. Далі вони передають керуючі сигнали на електромеханічні засоби управління до тих пір, поки дані на фотоприймачах не стануть однакові. Основним недоліком такої системи є некоректна робота у хмарну погоду та при забруднених фотодетектора, а перевагою – доступна ціна.

— Управління електромеханічні засоби управління за допомогою програми, яка відповідно до інтервалів часу визначає теоретичне місце розташування Сонця. Сонце протягом року рухається по небу із сходу на захід та його положення описується двома координатами: Азимут і Зенітний кут (рис 1.3). Азимут – це кут між розташуванням Сонцем та півдня. Зенітний кут – кут між лінією, що сполучає Сонце та предмет, на який воно світить та горизонтальною площиною.

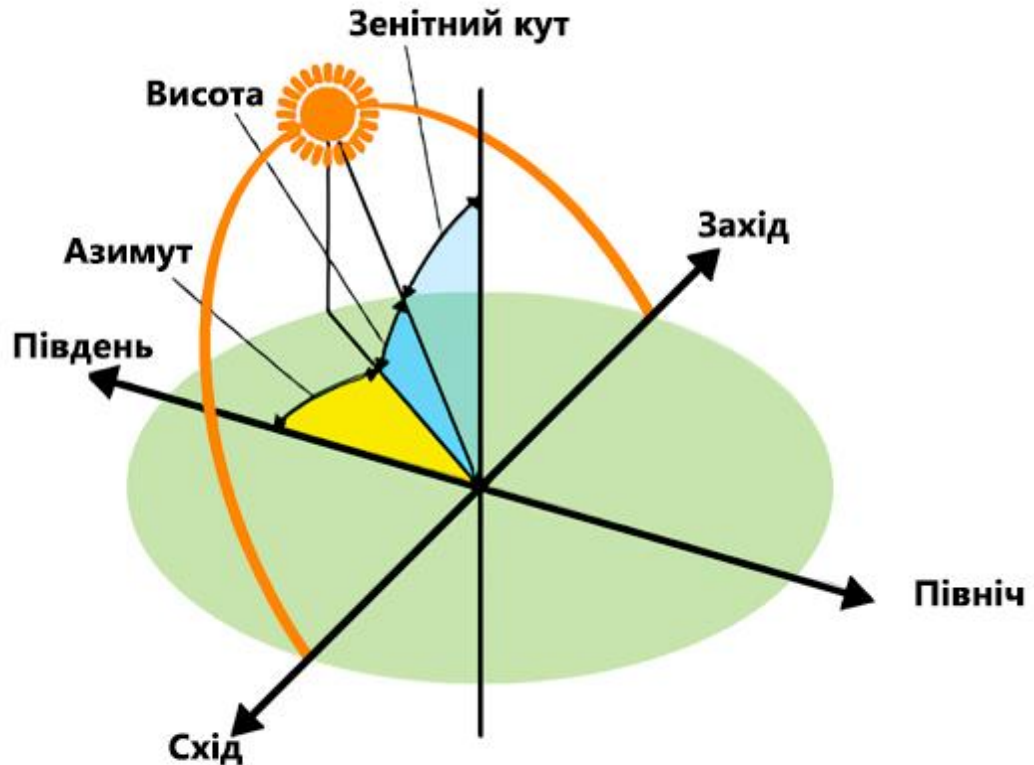


Рис 1.3 Розташування Сонця за Азимутом та Зенітним кутом

«За внутрішнім годинником, який вмонтовано у пристрій, програма розраховує значення Азимута і Зенітного кута та надає блоку керування. Також враховується місцезнаходження трекера. Далі виконується перерахунок координат для зміни положення сонячних панелей 3. Керування положенням трекера за допомогою Азимута і Зенітного кута. Основна ідея роботи системи базується на тому, що треба правильно компенсувати лише два зміщення Землі: □ добове зміщення, внаслідок обертання Землі довкола своєї осі; □ річне зміщення, внаслідок обертання Землі навколо Сонця» [38].

1.3. Обґрунтування вибору технічної моделі

Перш за все, потрібно визначити, для яких цілей вам потрібна сонячна панель та які вимоги вона повинна задовольняти. Наприклад, якщо вам потрібна панель для забезпечення електроенергією великої будівлі, то можливо варто обрати модель з великою потужністю та високою ефективністю.

Далі, важливо враховувати місцезнаходження та кліматичні умови, де

будуть встановлені панелі. Наприклад, якщо ваші панелі будуть встановлені в регіоні з високими температурами, ви можете вибрати модель з високою термічною стабільністю.

Також, важливо враховувати простір, доступний для встановлення панелей, та їх вагу. Наприклад, якщо у вас обмежений простір для встановлення панелей, можливо варто обрати модель з меншою площею та меншою вагою.

Крім того, важливо звернути увагу на виробника та якість матеріалів. Вибір виробника з доброю репутацією та високою якістю матеріалів допоможе забезпечити більш довговічну та надійну роботу панелей.

«Окрім цього, варто враховувати вартість панелі та клітинну технологію. Різні моделі можуть мати різну ціну, тому важливо зважати на баланс між вартістю та продуктивністю. Крім того, різні моделі можуть мати різну клітинну технологію, яка може впливати на ефективність панелей.

Для вирішення цих завдань можна використовувати різні методи, такі як аналіз технічних характеристик, порівняння вартості та ефективності, а також врахування досвіду та рекомендацій від експертів» [18].

При виборі технічної моделі сонячних панелей, потрібно враховувати кілька важливих факторів:

- Потужність: потужність сонячних панелей вимірюється в ватах (W). Чим більша потужність, тим більша кількість енергії може бути вироблена з кожної панелі. Зазвичай, чим більше потужність, тим дорожча панель.

- Ефективність: ефективність сонячних панелей вимірюється в відсотках і вказує, яка частина сонячної енергії, яка падає на панель, перетворюється в електричну енергію. Чим вища ефективність, тим менше панелей потрібно для вироблення потрібної кількості електричної енергії.

- Розмір та вага: розмір та вага сонячних панелей має значення, особливо, якщо ви маєте обмежений простір для їх розміщення. Деякі моделі

можуть бути меншими та легшими, ніж інші, тому важливо зважати на ці фактори при виборі моделі.

— Тривалість експлуатації та гарантія: більшість сонячних панелей мають гарантію від виробника на десятиліття. Важливо обрати надійного виробника, що ваші панелі будуть працювати тривалий час.

— Вартість: вартість сонячних панелей може значно відрізнятись в залежності від моделі. Важливо зважати на фактори, які ми вже згадували, та визначити оптимальний баланс між вартістю та якістю.

— Клітинна технологія: існує кілька видів клітинних технологій, таких як кристалічні силіконові (монокристалічні та полікристалічні), аморфні та тонкі плівки. Кожен з цих типів має свої переваги та недоліки, тому важливо вибрати технологію, яка найкраще відповідає вашим потребам.

«Одним з важливих факторів є потужність та ефективність панелі. Більш потужна панель може забезпечити більше енергії, але вона може бути більшої ваги та розміру, що може впливати на вартість монтажу та транспортування» [1].

Також, важливо звернути увагу на технологію клітин, оскільки вона може впливати на ефективність панелі та її вартість.

Інший важливий фактор - термічна стабільність. Якщо панелі будуть встановлені в регіоні з високими температурами, важливо звернути увагу на їх термічну стабільність, оскільки панелі можуть втратити ефективність при високих температурах.

Також, важливо звернути увагу на гарантійні умови та репутацію виробника, оскільки ці фактори можуть впливати на тривалість роботи панелей та їхню надійність.

Після аналізу всіх цих факторів, можна вибрати технічну модель сонячної панелі, яка відповідає всім вимогам та місцевим умовам, забезпечує ефективність та продуктивність та має оптимальну вартість.

РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ МОДЕЛЮВАННЯ СИСТЕМ

2.1 Огляд програмного забезпечення управління контролерами

«Контролери для сонячних панелей - це електронні пристрої, які контролюють та регулюють енергетичний потік між сонячними панелями, батареєю та споживачами енергії»[8]. Вони дозволяють оптимізувати виробництво та використання електроенергії, зберігати її в батареї та забезпечувати безперебійну роботу електростанції на базі сонячних панелей (Рис 2.1).



Рис 2.1 Контролер заряду сонячних батарей

«Контролери мають різноманітні функції, які допомагають забезпечити ефективну та стабільну роботу сонячної енергетичної системи. Наприклад, вони контролюють напругу та струм, які генерують сонячні панелі, регулюють енергетичний потік між сонячними панелями, батареєю та споживачами енергії, а також забезпечують захист від перенапруги та короткого замикання» [10].

Існують наступні види контролерів:

- з найпростішим алгоритмом ON/OFF;
- ШІМ – з широтно-імпульсною модуляцією;

— MPPT (Maximum Power Point Tracking) – пристрій, які стежить за точками максимальної потужності;

— гібридний.

Найпростіший пристрій – ON/OFF. Він вимикає заряд, коли струм максимальний, щоб не було перегріву. Мінус – акумулятор ніколи не заряджатиметься на 100%, через що термін експлуатації може суттєво зменшитися.

Контролер заряду сонячної батареї ШІМ знижує струм, коли показник досягає максимуму. Такі найчастіше використовують у регіонах, де сонце найактивніше.

«MPPT контролер заряду для сонячних батарей – найпопулярніший. Його принцип дії такий: перетворення максимальної напруги в ту, яка оптимальна. Завдяки цьому заряд можливий до ста відсотків.

MPPT аналізує показники системи, а на їхній основі визначає, завдяки яким параметрам обладнання досягне найбільшої потужності. Ефективність роботи, порівняно з ШІМ, зростає до 30%»[33].

Гібридні – використовують для сонячних систем, до складу яких входять вітрогенератори та фотоелементи. Основна функція такого – уникнення енергії в надлишку.

Для сонячних електростанцій часто використовуються MPPT-контролери. Вони дозволяють оптимізувати виробництво електроенергії шляхом максимального використання потужності, яку можуть генерувати сонячні панелі. Це досягається завдяки тому, що MPPT-контролери аналізують вихідні параметри сонячних панелей та знаходять точку максимальної потужності, що дозволяє забезпечити максимальну ефективність роботи електростанції на базі сонячних панелей.

«Крім того, контролери можуть мати різні інтерфейси для забезпечення комунікації між сонячними панелями, батареєю та споживачами енергії. Це може бути, наприклад, RS-485, Modbus, CAN, Ethernet, Wi-Fi або Bluetooth.

Ці інтерфейси дозволяють зчитувати та відображати дані про стан сонячних панелей, батареї та споживачів енергії, а також віддалено керувати режимами роботи системи» [26].

«Тепер, коли ви розумієте, що таке контролер для сонячної панелі, ми можемо перейти до програмного забезпечення, яке використовується для управління цими контролерами. Програмне забезпечення для управління контролерами сонячної енергії може бути розроблено на базі різних платформ, таких як Arduino, Raspberry Pi, або спеціалізованих мікроконтролерів» [23].

Теоретичні аспекти програмного забезпечення для управління контролерами сонячної енергії включають розуміння протоколів зв'язку, які використовуються для зв'язку між контролером та програмним забезпеченням, розуміння алгоритмів зарядки та розрядки батареї, а також розуміння основних функцій контролера, таких як захист від перезаряду та не до заряду, регулювання напруги та струму, тощо.

Щодо практики, найкращий спосіб для ознайомлення з програмним забезпеченням для управління контролерами сонячної енергії - це розробка прототипу. Розробка прототипу дозволяє перевірити та налагодити роботу програмного забезпечення, взаємодію контролера з батареєю та сонячними панелями у реальних умовах.

Першим кроком в розробці прототипу є вибір платформи для програмування контролера. Для початківців найкращим варіантом може бути Arduino, яка є доступною та має велику кількість готових бібліотек та прикладів. Для більш складних завдань можна використовувати Raspberry Pi або спеціалізовані мікроконтролери.

«Далі, необхідно розробити програмне забезпечення для контролера, яке буде відповідати всім вимогам та функціональним можливостям. Основні функції програмного забезпечення повинні бути пов'язані з управлінням зарядкою та розрядкою батареї, а також контролем напруги та струму. Також, необхідно додати функції, що дозволять відстежувати рівень заряду

батареї, стан сонячних панелей та інші параметри, які можуть бути корисні для моніторингу та управління» [17].

«Після розробки програмного забезпечення необхідно виконати його перевірку на роботу здатність. Для цього можна підключити контролер до сонячних панелей та батареї, та перевірити роботу програмного забезпечення у реальних умовах. Важливо перевірити, чи правильно працює захист від перезаряду та не до заряду, чи коректно працює регулювання напруги та струму, та інші функції контролера.

Крім того, можна розглянути можливості використання спеціалізованих програм для моніторингу та управління сонячними панелями та батареєю. Такі програми дозволяють зчитувати рівень заряду батареї та стан сонячних панелей в реальному часі, а також здійснювати управління контролером з використанням комп'ютера або мобільного пристрою»[31].

У разі виникнення проблем з роботою контролера або сонячних панелей, можна використовувати різноманітні діагностичні інструменти, які дозволяють відстежувати стан системи та виявляти можливі проблеми.

Такі інструменти можуть бути вбудованими у програмне забезпечення контролера або ж додатково придбаними та встановлені на комп'ютері або мобільному пристрої.

Важливо також пам'ятати про безпеку при роботі з сонячними панелями та батареєю. Необхідно дотримуватися правил монтажу та експлуатації обладнання, а також використовувати захисні елементи та перевіряти їх роботу регулярно.

Огляд програмного забезпечення управління контролерами для сонячних панелей є важливим етапом у розробці та впровадженні сонячної енергетики. Необхідно враховувати як теоретичні аспекти, так і практичні вимоги, та забезпечувати безпеку при роботі з обладнанням.

«Raspberry Pi - це комп'ютер у форм-факторі одноплатного мікроконтролера, який працює на базі процесора ARM та має вбудований набір засобів для взаємодії з периферійними пристроями. Він має вбудований

Ethernet-порт, Wi-Fi та Bluetooth-модулі, що дозволяє підключати його до мережі Інтернет та комунікувати з іншими пристроями»[37].

«Для реалізації контролера для сонячної енергетичної системи на базі Raspberry Pi можна використовувати спеціалізовані додатки та програми, які дозволяють зчитувати дані з датчиків, обробляти їх та керувати енергетичним потоком. Наприклад, можна використовувати програму SolarPV для зчитування та аналізу даних з сонячних панелей, а також програму Home Assistant для керування режимами роботи системи та відображення даних про стан сонячних панелей, батареї та споживачів енергії»[25].

Для забезпечення зберігання електроенергії можна використовувати літій-іонні акумуляторні батареї, які забезпечують довгий термін зберігання та високу енергоефективність. Також можна використовувати системи автоматичного перемикавання на резервний джерело живлення в разі відмови основного джерела.

У цілому, контролер для сонячної енергетичної системи на базі Raspberry Pi може бути реалізований як програмно-апаратний комплекс, що забезпечує збір та обробку даних про стан сонячних панелей, батареї та споживачів енергії, керування режимами роботи системи, забезпечення зберігання електроенергії та автоматичного перемикавання на резервне джерело живлення в разі відмови основного джерела.

«Для програмування контролера можна використовувати різноманітні мови програмування, такі як Python, C++, Java та інші. Python є однією з найбільш популярних мов програмування для Raspberry Pi, оскільки вона має простий синтаксис, широкий спектр бібліотек та модулів, що дозволяють легко реалізувати програми для зчитування та обробки даних з датчиків, керування електропотокм та інше»[36].

Можна використовувати веб-інтерфейс, який дозволяє віддалено керувати роботою контролера. Веб-інтерфейс можна розробити на основі фреймворку Flask для Python. Він дозволяє реалізувати веб-сервер, який

відповідає на запити користувачів та відображає поточний стан сонячних панелей та батареї. Також можна використовувати для збору та аналізу даних про роботу сонячних панелей та батареї. Для цього можна використовувати датчики, які зчитують температуру, освітленість та інші параметри, які впливають на роботу сонячних панелей. Зібрані дані можна зберігати в базі даних, наприклад, SQLite, а потім аналізувати за допомогою інструментів аналізу даних, таких як Pandas, NumPy, SciPy та інших.

Наприклад, можна використовувати алгоритм Random Forest для прогнозування рівня заряду батареї на основі даних про роботу сонячних панелей та інших параметрів. Для цього потрібно створити модель машинного навчання, яка буде навчатися на основі зібраних даних та дозволить прогнозувати рівень заряду батареї на майбутній період.

«Для збору даних про роботу сонячних панелей та батареї можна використовувати датчики температури, освітленості та інших параметрів, а для їх зберігання та аналізу - базу даних SQLite та інструменти аналізу даних, такі як Pandas та NumPy»[20].

Таким чином, контролер для сонячної енергетичної системи на базі Raspberry Pi може бути ефективним та зручним засобом для забезпечення моніторингу та керування енергопотокom. Завдяки широкому спектру інтерфейсів та можливостей програмування, він може бути налаштований під конкретні потреби та вимоги, а також може бути розширений за потреби користувача. Однак, при розробці контролера на базі необхідно враховувати деякі особливості.

«Наприклад, Raspberry Pi має обмеження щодо споживання електроенергії, тому важливо добирати датчики та інші компоненти з мінімальним споживанням енергії. Також необхідно враховувати температурні умови роботи, оскільки висока температура може негативно вплинути на роботу пристрою» [41].

Окрім того, при підключенні контролера до мережі Інтернет важливо забезпечити безпеку даних та забезпечити захист від зловмисників. Для цього

можна використовувати різноманітні заходи, такі як шифрування даних, налаштування файрвола, авторизацію та інші.

«У цілому, контролер для сонячної енергетичної системи є важливою складовою, що забезпечує моніторинг та керування енергопотокom. Завдяки використанню Raspberry Pi та інших спеціалізованих модулів, можна створити ефективний та зручний засіб керування, що відповідає потребам користувача. Однак, при розробці контролера необхідно враховувати особливості використання та безпеки даних» [27].

Крім того, важливо розглянути питання взаємодії контролера з іншими елементами сонячної енергетичної системи, такими як інвертор, акумулятор, сонячні панелі та інші. Для цього можна використовувати різні комунікаційні протоколи, наприклад, Modbus, CAN або інші.

Крім забезпечення моніторингу та керування енергопотокom, контролер може виконувати інші завдання. Наприклад, він може забезпечувати керування освітленням, системою опалення та охолодження, системою безпеки, які також можуть працювати на сонячній енергії.

Загалом, контролер для сонячної енергетичної системи є важливою складовою, яка забезпечує моніторинг та керування енергопотокom та іншими функціями системи. Розробка контролера може бути здійснена з використанням різних технологій та програмних платформ, залежно від потреб користувача та вимог до системи.

2.2 Огляд бібліотек Python для програмування управління СБ

Використання Raspberry Pi з Python дозволяє створювати гнучкі та ефективні системи керування сонячними панелями, які можуть забезпечувати максимальну продуктивність та ефективність.

«Додатково, Raspberry Pi можна використовувати для збору та аналізу даних про роботу сонячних панелей та їх ефективності. Для цього можна встановити на Raspberry Pi програми, які збирають та аналізують дані про роботу сонячних панелей та їх продуктивність. Це може допомогти забезпечити оптимальну роботу сонячної системи та зменшити витрати на

енергію» [40].

Таким чином, використання бібліотек в Python дозволяє значно спростити розробку програм та забезпечити швидке та ефективне виконання різноманітних завдань.

Кожна бібліотека використовується в залежності від поставленої задачі. Для керування і коректної роботи сонячних панелей, треба використовувати датчики які 1 температуру, світловий потік сонячних променів, напрям сонця і якщо конструкція більш технологічна і розвинута то керування приводами для супроводження панелі за сонцем.

«Одним з варіантів використання Raspberry Pi в сонячних панелях може бути встановлення на нього програм, які дозволяють контролювати напругу та струм, що генерують сонячні панелі. Це може допомогти забезпечити оптимальну роботу сонячних панелей, зменшити ризик перевантаження та збільшити продуктивність системи в цілому» [3].

Для розробки програм для керування сонячними панелями на Python з Raspberry Pi можна використовувати різні бібліотеки наприклад:

- RPi.GPIO - бібліотека дозволяє керувати GPIO-портами на Raspberry Pi, що необхідно для підключення до сонячних панелей та батарей. Вона дозволяє працювати з цифровими та аналоговими GPIO-портами, а також зчитувати та записувати значення вхідних/вихідних портів.

- Flask - бібліотека дозволяє створювати веб-інтерфейси для управління сонячними батареями на Raspberry Pi. Вона дуже проста у використанні та дозволяє створювати сторінки веб-інтерфейсу для

- PySerial - дозволяє здійснювати зв'язок з пристроями через порт UART (Universal Asynchronous Receiver/Transmitter), який використовується для зчитування даних з сонячних панелей та відправлення їх до Raspberry Pi.

- NumPy - містить функції для обробки числових даних, що можуть бути корисними при аналізі даних, отриманих з сонячних панелей.

— TensorFlow - для аналізу та обробки даних, яка може бути корисною для аналізу даних, отриманих з сонячних панелей, і прогнозування їх ефективності. Pandas може зчитувати дані з різних джерел, включаючи CSV-файли та бази даних, і надавати зручний інтерфейс для їх обробки та аналізу. Використання Pandas може допомогти зменшити час, необхідний для обробки та аналізу великої кількості даних, пов'язаних з сонячними панелями.

— PySolar - дозволяє отримувати дані про сонячні панелі з метою оптимізації виробництва енергії. Вона включає в себе функції для обчислення позиції сонця, а також для розрахунку потужності виробництва енергії в залежності від змінних параметрів, таких як погода та змінна позиція сонця.

— MQTT - дозволяє здійснювати обмін повідомленнями між різними компонентами системи, такими як Raspberry Pi та сонячні панелі. Це може бути корисним для розробки розподіленої системи управління сонячними панелями.

— Adafruit CircuitPython - дозволяє легко підключати різні пристрої до Raspberry Pi, такі як сенсори, дисплеї та інше. Вона призначена для спрощення програмування мікроконтролерів і одноплатних комп'ютерів, таких як Raspberry Pi, у вбудованих системах.

Ці бібліотеки можуть бути корисними при розробці програмного забезпечення для управління сонячними батареями на Raspberry Pi. Вони дозволяють легко керувати GPIO-портами, зчитувати дані з датчиків та ADC, комунікувати з пристроями через порти RS-232 та RS-485, створювати веб-інтерфейси для відображення стану та керування батареями, та проводити науково-технічні обчислення та аналіз даних.

Для роботи з "Raspberry Pi" потрібно вивчити кілька бібліотек, включаючи бібліотеку для взаємодії з GPIO (General Purpose Input/Output). GPIO - це інтерфейс вводу/виводу загального призначення, який забезпечує

зв'язок між компонентами комп'ютерної системи, наприклад, мікропроцесором та різними периферійними пристроями. Порт GPIO міститься у внутрішній структурі процесорів, чіпсетів та інших допоміжних чіпів управління, і може обробляти як вхідні, так і вихідні цифрові сигнали.

Вхідний порт може бути використаний для зв'язку ЦП або РСН з сигналами, отриманими від перемикачів або датчиків, а вихідний порт - для формування сигналів управління зовнішніми операціями на основі програм виконуваних процесором, наприклад, для управління світлодіодним підсвічуванням дисплея або для виведення сигналів управління двигуном.

GPIO контакти можуть бути налаштовані як вхідні або вихідні, а також групуватися в порти. Це важливо знати, оскільки без цих знань неможливо працювати з зовнішніми модулями на Raspberry Pi. Не думайте, що програмування цього складне - з відповідними матеріалами може впоратися навіть дитина. Витрачені час та зусилля повернуться вам у вигляді гідного результату.

«Порти GPIO зазвичай мають два режими роботи: вхідний та вихідний. У вхідному режимі порт приймає цифрові сигнали з зовнішніх пристроїв, наприклад, від перемикачів або датчиків. У вихідному режимі порт відправляє цифрові сигнали до зовнішніх пристроїв, наприклад, для управління світлодіодами або двигунами» [20].

GPIO дає можливість програмістам та електронним інженерам створювати різноманітні пристрої та системи з використанням електроніки та програмування. Наприклад, можна підключити до Raspberry Pi сенсорну панель, яка буде передавати дані про натискання кнопок, або створити систему дистанційного керування роботом з використанням GPIO портів.

Таким чином, GPIO є важливим компонентом для створення електронних систем та додатків. Його практичне використання дозволяє створювати різноманітні пристрої з використанням електроніки та програмування, що можуть виконувати різні завдання в залежності від потреб користувача.

Можна використовувати RPi.GPIO для керування сонячними панелями, наприклад, для включення або вимикання живлення.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
# Встановлення піну 18 як вихідного
GPIO.setup(18, GPIO.OUT)
# Включення живлення
GPIO.output(18, GPIO.HIGH)
# Затримка 5 секунд
time.sleep(5)
# Вимкнення живлення
GPIO.output(18, GPIO.LOW)
# Звільнення ресурсів GPIO
GPIO.cleanup()
```

Є інструментом для роботи з послідовним портом, що може використовуватися для комунікації між Raspberry Pi та зовнішніми пристроями, такими як контролери заряду сонячних батарей.

«Наприклад, прилад Raspberry Pi може підключатися до контролера заряду через послідовний порт, використовуючи PySerial, для читання даних про стан заряду сонячних батарей та керування процесом заряду. За допомогою PySerial можна відправляти та отримувати дані від контролера, наприклад, запитувати інформацію про стан заряду батарей, налаштовувати параметри заряду, відстежувати рівень заряду та інші важливі параметри»[36].

Таким чином, PySerial дозволяє Raspberry Pi взаємодіяти з контролером заряду та керувати процесом заряду сонячних батарей, що є важливим елементом в системах живлення з використанням відновлюваних джерел енергії, таких як сонячні батареї.

Створюємо скрипт на мові Python, який встановлює зв'язок з

контролером заряду та виконує команди для керування процесом заряду. Ось приклад коду:

```
import serial

# встановлення параметрів порту
port = '/dev/ttyUSB0' # назва послідовного порту
baud_rate = 9600 # швидкість передачі даних
timeout = 1 # тайм-аут для читання даних

# встановлення з'єднання з портом
ser = serial.Serial(port=port, baudrate=baud_rate, timeout=timeout)

# відправлення команди на зчитування даних про стан заряду батарей
ser.write(b'Read battery status\n')

# отримання відповіді від контролера заряду
response = ser.readline().decode().strip()

# обробка відповіді та виведення інформації
if response == 'Battery is fully charged':
    print('Батарея повністю заряджена')
elif response == 'Battery is charging':
    print('Батарея заряджається')
else:
    print('Помилка отримання статусу батареї')

# закриття з'єднання з портом
ser.close()
```

У цьому прикладі коду Raspberry Pi відправляє команду на зчитування даних про стан заряду батарей до контролера заряду за допомогою функції `write()` з бібліотеки `PySerial`. Після цього Raspberry Pi очікує відповідь від контролера заряду за допомогою функції `readline()`, яка зчитує один рядок даних з послідовного порту. Отриману відповідь Raspberry Pi обробляє та виводить на екран.

Цей приклад коду демонструє, як `PySerial` можна використовувати для зчитування даних про стан заряду батарей та контролювати процес заряду в

системах з використанням сонячних батарей. Залежно від можливостей та протоколу комунікації контролера заряду, код може бути розширений для виконання різних дій, наприклад, налаштування параметрів заряду або отримання додаткової інформації про стан батарей.

Потужний інструмент для наукових обчислень, що дозволяє працювати з масивами даних та виконувати різні математичні операції над ними. NumPy є дуже корисним при розв'язанні задач управління сонячними батареями на приладі Raspberry Pi. Ось деякі способи, які можна застосувати NumPy у цій області:

- «Обробка даних: NumPy дозволяє зручно та ефективно завантажувати, зберігати та обробляти дані. Ви можете використовувати його для читання даних з сенсорів, які вимірюють сонячну радіацію або стан батарей, і виконувати операції над цими даними, наприклад, обчислення середнього значення, стандартного відхилення тощо.

- Математичні обчислення: NumPy має багатий набір функцій для виконання математичних операцій. Ви можете використовувати ці функції для обчислення енергетичного балансу сонячної батареї, розрахунку коефіцієнта ефективності, аналізу змін рівня заряду батареї тощо» [34].

- Обробка сигналів: Управління сонячними батареями часто пов'язане з обробкою сигналів. NumPy має потужні функції для фільтрації сигналів, виявлення піків, витягування характеристик сигналу тощо. Ви можете використовувати ці функції для аналізу сонячних змін, виявлення аномалій або оптимізації роботи батарей у відповідь на зміну сонячної активності.

- Оптимізація: NumPy містить функції для оптимізації обчислень та розв'язання оптимізаційних задач. Ви можете використовувати ці функції для оптимізації роботи сонячної батареї, наприклад, для визначення оптимального режиму заряду та розвантаження батареї на основі вхідних даних про сонячну активність.

Використання бібліотеки NumPy на Raspberry Pi допоможе вам зручно та ефективно працювати з даними та виконувати математичні операції, пов'язані з управлінням сонячними батареями. Нижче наведемо приклад:

```
import numpy as np

# Задання прикладових даних сонячної активності
solar_data = np.array([0.2, 0.3, 0.5, 0.7, 0.8, 0.6, 0.4, 0.3, 0.2, 0.1])

# Обчислення середнього значення сонячної активності
average = np.mean(solar_data)

print("Середнє значення сонячної активності:", average)

# Обчислення максимального значення сонячної активності
maximum = np.max(solar_data)

print("Максимальне значення сонячної активності:", maximum)

# Обчислення індексів, де сонячна активність перевищує певне
# порогове значення
threshold = 0.5

indices = np.where(solar_data > threshold)

print("Індекси, де сонячна активність перевищує", threshold, ":", indices)

# Розрахунок енергетичного балансу за допомогою векторних операцій
energy_input = solar_data * 10 # Припустимо, що кожне значення
# представляє вхідну енергію, помножену на 10

energy_output = np.random.rand(10) # Припустимо, що це вихідна
# енергія

energy_balance = energy_input - energy_output

print("Енергетичний баланс:", energy_balance)
```

У цьому прикладі ми спочатку імпортуємо бібліотеку NumPy як np. Потім ми створюємо масив solar_data, який представляє сонячну активність протягом деякого періоду часу.

Далі ми використовуємо функції NumPy, такі як np.mean і np.max, для обчислення середнього та максимального значення сонячної активності відповідно. Ми також використовуємо функцію np.where для знаходження

індексів, де сонячна активність перевищує певне порогове значення.

«На останньому кроці ми використовуємо векторні операції NumPy для обчислення енергетичного балансу, де кожне значення вхідної енергії множиться на 10, а вихідна енергія генерується випадковим чином»[21]. Результат зберігається у масиві `energy_balance`.

TensorFlow є потужним інструментом для розробки та навчання глибоких нейронних мереж. Її можна успішно використовувати на пристрої Raspberry Pi, пов'язаному з управлінням сонячними батареями, для різних завдань.

Ось кілька способів, які можуть бути застосовані для використання TensorFlow на Raspberry Pi для управління сонячними батареями:

- Прогнозування виробництва енергії: Використовуючи TensorFlow, ви можете навчити нейронну мережу прогнозувати виробництво енергії сонячних батарей на основі різних факторів, таких як погода, сонячна радіація, позиція сонця тощо. Це може допомогти вам визначити оптимальний час для зарядки батареї та планувати використання енергії.

- Оптимізація заряду та розряду: Використовуючи TensorFlow, можна створити модель, яка враховуватиме історичні дані про використання енергії та підлаштовуватиме заряд і розряд батареї відповідно. Це може допомогти зменшити витрати на енергію, оптимізувати використання сонячних батарей та зберегти їх ресурси.

- Виявлення несправностей та діагностика: TensorFlow можна використовувати для навчання моделей, які виявлятимуть несправності або аномалії у роботі сонячних батарей. Наприклад, ви можете навчити модель розпізнавати аномальні показники напруги або струму, що свідчать про проблеми з батареєю. Такі моделі можуть допомогти вчасно виявляти проблеми та забезпечувати безперебійну роботу системи.

- Автоматичне регулювання енергії: TensorFlow можна використовувати для створення моделей, які автоматично регулюють

використання енергії сонячних батарей. Ви можете навчити модель прогнозувати споживання енергії та відповідно керувати зарядом та розрядом батарей. Це може забезпечити ефективне використання енергії та збереження ресурсів батарей.

Окрім цього, TensorFlow також може бути використаний для обробки та аналізу даних сонячної енергії, візуалізації результатів, створення інтерактивних інтерфейсів та багатьох інших завдань, пов'язаних з управлінням сонячними батареями на Raspberry Pi.

Загалом, TensorFlow на Raspberry Pi є потужним інструментом для розробки і реалізації різних алгоритмів та моделей управління сонячними батареями, які можуть допомогти оптимізувати їх використання та збереження енергії.

Зважаючи на обмежену потужність Raspberry Pi, важливо розробляти легкі моделі та оптимізувати їх для виконання на цьому пристрої. Ось приклад коду, який показує, як можна використати TensorFlow на Raspberry Pi для прогнозування виробництва енергії сонячної батареї:

```
import tensorflow as tf
import numpy as np
# Завантаження навченої моделі
model = tf.keras.models.load_model('solar_model.h5')
# Збирання даних для прогнозування
weather_data = [0.2, 0.3, 0.4, 0.5] # Приклад даних про погоду (можна
замінити на реальні дані)
input_data = np.array([weather_data])
# Нормалізація даних
input_data = (input_data - np.mean(input_data)) / np.std(input_data)
# Прогнозування виробництва енергії
predicted_energy = model.predict(input_data)
# Відновлення оригінального масштабу прогнозу
predicted_energy = predicted_energy * std_energy + mean_energy
```

```
# Виведення результату
```

```
print("Прогнозоване виробництво енергії: ", predicted_energy)
```

Цей приклад передбачає, що ви вже навчили модель TensorFlow для прогнозування виробництва енергії на основі даних погоди. Дані погоди, які використовуються для прогнозу, повинні бути підготовані та нормалізовані перед використанням моделі. Також, у прикладі використовуються середнє значення (`mean_energy`) та стандартне відхилення (`std_energy`) енергії для відновлення оригінального масштабу прогнозу.

Це лише один приклад використання TensorFlow на Raspberry Pi для управління сонячними батареями. Ви можете адаптувати цей код та розробити більш складні моделі та функції залежно від вашого конкретного випадку застосування.

В бібліотека PySolar є відкритим джерелом і написана на мові програмування Python. Вона надає можливість визначити параметри сонячної радіації для будь-якої точки на земній поверхні, включаючи інформацію про сонячні діаграми та деклінацію сонця.

Ця бібліотека може бути корисною для розрахунку оптимального положення сонячних панелей для отримання максимальної кількості сонячної енергії. Наприклад, вона може бути використана для визначення оптимального кута нахилу панелів в залежності від широти та довготи місця розташування.

Існує кілька можливих застосувань PySolar на пристроях Raspberry Pi, що пов'язані з управлінням сонячними батареями:

- Планування режиму заряду: Використовуючи дані, отримані з PySolar, можна прогнозувати і оптимізувати режим заряду сонячних батарей. Наприклад, на основі передбачуваної сонячної активності та позиції сонця можна розрахувати оптимальний час та потужність заряду, щоб ефективно використовувати сонячну енергію.

- Відстеження позиції сонця: Використовуючи PySolar, можна

визначити поточну позицію сонця в реальному часі. Це може бути корисно для автоматичного відстеження сонячних панелей або дзеркальних систем з метою максимального збору сонячної енергії.

— Моніторинг сонячної активності: PySolar надає доступ до даних про сонячну радіацію. Ці дані можна використовувати для моніторингу сонячної активності в реальному часі або для аналізу і прогнозування трендів. Наприклад, можна створити систему, яка автоматично реагує на зміну сонячної активності і адаптує роботу сонячних батарей відповідно.

«Врахуйте, що це лише деякі приклади використання бібліотеки PySolar для управління сонячними батареями. Застосування можуть бути більш різноманітними залежно від ваших конкретних потреб та проекту» [32].

Для отримання даних про сонячну активність та визначення оптимального часу заряду сонячних батарей:

```
from datetime import datetime
from pysolar.solar import get_altitude, get_azimuth
# Координати місця розташування
latitude = 52.5200
longitude = 13.4050
# Отримання поточної позиції сонця
date = datetime.now()
altitude = get_altitude(latitude, longitude, date)
azimuth = get_azimuth(latitude, longitude, date)
# Виведення результатів
print("Поточна позиція сонця:")
print("Висота (градуси):", altitude)
print("Азимут (градуси):", azimuth)
# Прогнозування оптимального часу заряду
# Наприклад, якщо ми хочемо заряджати батареї, коли сонце перебуває
вище 30 градусів
```

```
if altitude > 30:
```

```
    print("Оптимальний час для заряду батарей!")
```

```
else:
```

```
    print("Неоптимальний час для заряду батарей.")
```

У цьому прикладі ми використали функції `get_altitude` та `get_azimuth` з бібліотеки `PySolar` для отримання поточної позиції сонця на основі заданих координат та дати. Потім ми вивели отримані дані про висоту та азимут сонця. Далі ми зробили просту перевірку, чи є поточна висота сонця вищою за 30 градусів, щоб визначити оптимальний час для заряду сонячних батарей.

Це лише простий приклад, і ви можете розширити його функціональність залежно від ваших потреб. Наприклад, ви можете використовувати дані про сонячну активність для автоматичного керування зарядом сонячних батарей або створити розширену систему моніторингу.

2.3 Розробка аналітичної та технічної моделі УСБ

«Для позиціонування фотоелектричних елементів в напрямку Сонця необхідно обчислити два основних параметри, а саме кутову висоту Сонця та азимутувий кут» [12]. Вони не можуть бути повторно використані для іншого проміжку часу через нахил Землі і її руху навколо Сонця. Конвертація градусів у радіани. Для математичних обчислень основною одиницею вимірювання плоских кутів вважається радіан, тому для перетворення градусів в радіан слід використовувати наступну формулу:

$$Rad = \pi \cdot v / 180, \quad (2.1)$$

де v – це градусна одиниця, яку потрібно перевести в радіани.

Меридіани місцевого стандартного часу. Меридіан місцевого стандартного часу вважається еталонним меридіаном, який використовується для певного часового поясу, а також нульовими меридіаном, які використовують для середнього часу за Гринвічем

Меридіан місцевого стандартного часу вираховується за формулою:

$$LSTM = 15 \cdot \Delta TUTC \quad (3.2)$$

де $\Delta TUTC$ – це часовий пояс, який зазначений у Всесвітньому

координованому часі (UTC).

Для визначення рівняння часу потрібно скористатися нижче наведеною формулою:

$$EoT = 9.87 \cdot \sin(2 \cdot Brad) - 7.53 \cdot \cos(Brad) - 1.5 \cdot \sin(Brad) \quad (3.3)$$

Де B визначається за формулою:

$$B = 360 \cdot 365 \cdot (DY - 81) \quad (3.4)$$

де DY – це порядковий номер дня з початку року. Корекційний коефіцієнт часу.

«Корекційний коефіцієнт часу враховує зміни місцевого сонячного часу в межах даного часового поясу через зміни довготи в межах часового поясу, а також містить рівняння часу» [22]. Корекційний коефіцієнт часу можна визначити за формулою:

$$B = 4 \cdot (Longitude - LSTM) \cdot EoT \quad (3.5)$$

де $Longitude$ – це координати довготи, рад;

$LSTM$ – це меридіан місцевого стандартного часу;

EoT – це рівняння часу.

Місцевий сонячний час. Час який визначається в місці знаходження спостерігача видимим положенням Сонця на небесній сфері. Для визначення місцевого сонячного часу потрібно скористатися наступною формулою:

$$LST = H + TC \cdot 60 + M \cdot 60 \quad (3.6)$$

де H – поточна година спостерігача;

M – поточна хвилина спостерігача;

TC – корекційний коефіцієнт часу.

Часовий кут перетворює місцевий сонячний час в число радіан на які Сонце рухається по небу. Часовий кут визначається за формулою:

$$HRA = 15 \cdot LST \cdot \pi / 180 \quad (3.7)$$

де LST – це місцевий сонячний час.

«Кут схилення Сонця який являє собою кут між променями сонця і площиною екватора Землі. Даний кут змінюється в залежності від сезону

через нахил Землі навколо своєї осі обертання і обертання Землі навколо Сонця» [24]. Кут схилення сонця можна обчислити за формулою:

$$\delta = 23.45 \cdot \sin (Brad) \cdot \pi 180 \quad (2.2)$$

де B - обчислюється за формулою

Кутова висота Сонця. Висота Сонця над горизонтом не є постійною величиною, вона змінюється протягом дня і залежить від широти конкретного місця і дня року. Щоб визначити кутову висоту Сонця протягом дня потрібно скористатися наступною формулою:

$$\begin{aligned} Elevation = & \arcsin ((\sin(\delta) \cdot \sin(Latitude)) + \\ & + (\cos(\delta) \cdot \cos(Latitude) \cdot \cos (HRA)) \end{aligned} \quad (2.3)$$

де δ – кут схилення Сонця;

$Latitude$ – координати широти, рад;

HRA – це часовий кут.

«Азимутальний кут. Азимутальний кут являється напрямком компаса, з якого надходить сонячне світло (рис. 3). В час сонячного полудня сонце завжди знаходиться прямо на півдні в північній півкулі і прямо на півночі в південній півкулі. У дні рівнодення сонце встає прямо на сході і сідає прямо на заході незалежно від широти, тому азимутальні кути в цей час складають рівно 90° на сході та 270° на заході»[16]. Проте азимутальний кут змінює кут від залежності широти і пори року. Азимутальний кут обчислюється за формулою:

$$\begin{aligned} Azimuth = & \arccos (\cos(Latitude) \cdot \sin(\delta) - \\ & - \cos(\delta) \cdot \sin(Latitude) \cdot \cos(HRA) \cos(Elevation)) \end{aligned} \quad (2.4)$$

де δ – кут схилення сонця;

$Latitude$ – кут координати широти, рад;

HRA – це часовий кут;

$Elevation$ – це часовий кут.

«Часові зміни. Щоб коректно керувати сонячним трекером потрібно визначити часові зміни, а саме тривалість дня, час сходу за заходу Сонця. Ці

дані є необхідні, щоб налаштувати робочий час для коректної роботи сонячного трекера» [28]. Розрахунок сходу та заходу скористаємося наступними формулами:

$$Sunrise = 12 - \arccos(A) \cdot 0.261799 - TC \cdot 60 \quad (2.5)$$

$$Sunrise = 12 + \arccos(A) \cdot 0.261799 - TC \cdot 60 \quad (2.6)$$

де TC – корекційний коефіцієнт часу;

A – визначається за формулою.

$$Sunrise = -1 \cdot \sin(Latitude) \cdot \sin(\delta) \cos(Latitude) \cdot \cos(\delta) \quad (2.7)$$

де - Latitude координати широти, рад;

δ - кут схилення Сонця.

Тривалість дня розрахуємо за формулою:

$$DL = 24 \pi \cdot \arccos(-\tan(Latitude) \cdot \tan(\delta)) \quad (2.8)$$

де Latitude - координати широти, рад;

δ - кут схилення Сонця.

Запропонований метод розрахунку положено в основу алгоритму автоматизованої роботи сонячного трекера зображено на рис 2.2.

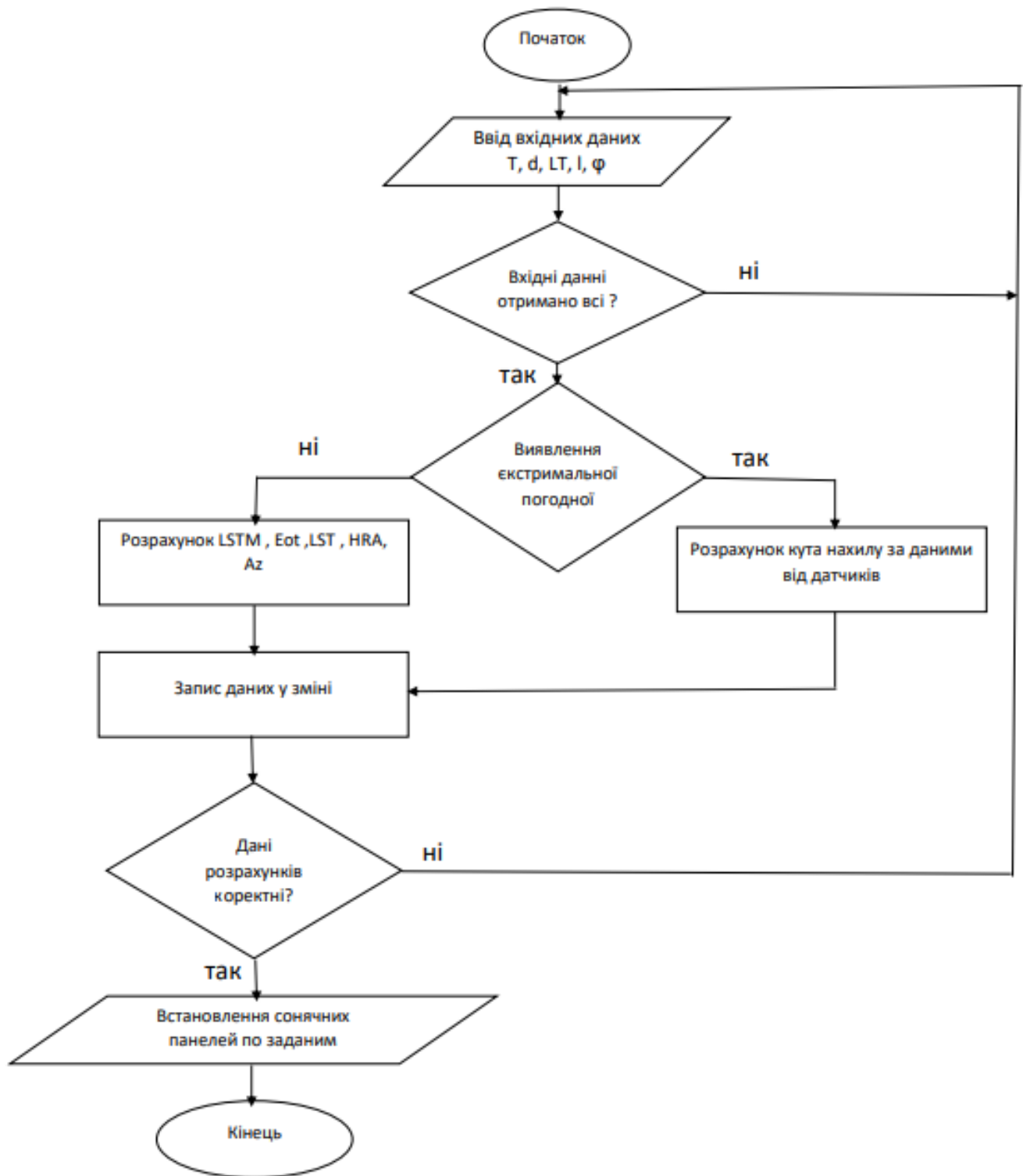


Рис 2.2 Алгоритм автоматизованої роботи сонячного трекера

«Вихідними даними алгоритму є часовий пояс T , кількість днів з початку року d , поточна година LT , координати довготи l та широти φ місця розташування сонячної електростанції. На основі розрахунків кута висоти α та азимуту Az можна точно виявити знаходження Сонця та встановити сонячні панелі під оптимальним кутом до нього» [15].

Представлена блок схема інтелектуального методу керування сонячною панеллю на рис 2.3.

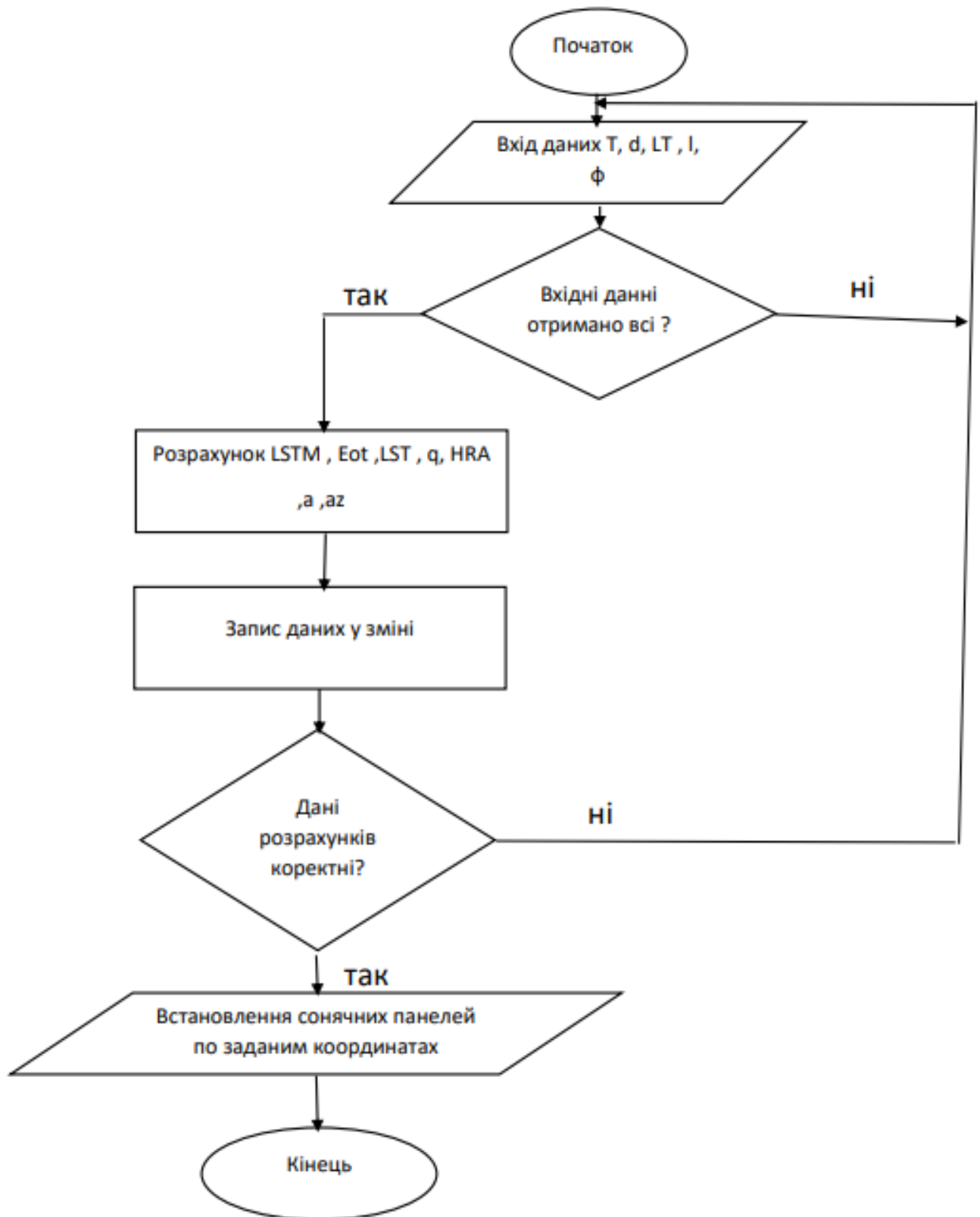


Рис 2.3 Блок схема інтелектуального методу керування сонячною панеллю

РОЗДІЛ 3. РОЗРОБКА МОДЕЛІ У СЕРЕДОВИЩІ PACKET TRACER

3.1 Моделювання системи

У контексті розробки моделі управління сонячними панелями, ми можемо використовувати Packet Tracer для створення віртуальної мережі, що включатиме сонячні панелі, сонячні контролери, схеми електроживлення та інші пристрої, пов'язані з управлінням сонячними системами.

«Packet Tracer дозволяє нам створювати, конфігурувати та тестувати взаємодію між цими компонентами у віртуальному середовищі. Ми можемо встановлювати параметри сонячних панелей, відстежувати енергетичні показники, налаштовувати сонячні контролери та аналізувати їх роботу» [29].

Також ми можемо моделювати різні сценарії управління сонячними системами та перевіряти їх ефективність і надійність у віртуальному середовищі, що дозволяє нам економити час, зусилля та ресурси.

У контексті розробки моделі управління сонячними панелями, ми можемо використовувати для створення віртуальної мережі, що включатиме сонячні панелі, сонячні контролери, схеми електроживлення та інші пристрої, пов'язані з управлінням сонячними системами показано на рис 3.1.

«Загальний підхід до використання Packet Tracer для моделювання управління сонячними панелями дозволяє нам краще розуміти технологію сонячних систем, тестувати різні сценарії та знаходити оптимальні рішення перед їх фізичною реалізацією» [30].

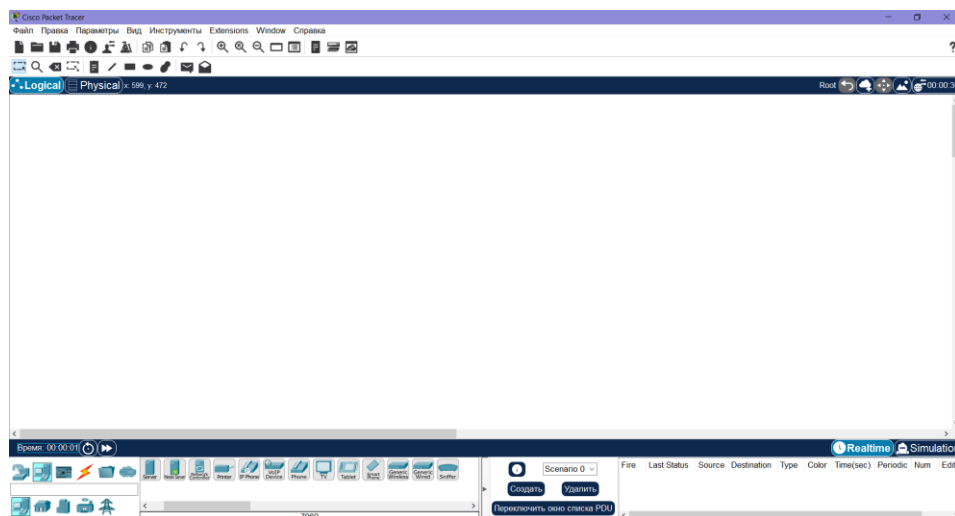


Рис 3.1 Створюємо проект в Packet Tracer

Опишемо компоненти, які будуть використовуватися для моделювання системи у середовищі Packet Tracer для управління сонячними панелями:

1. Smartphone (Смартфон): Смартфон використовується як інтерфейс для зручного управління системою управління сонячними панелями. За допомогою спеціального додатку на смартфоні можна контролювати та налаштовувати режими роботи панелей, переглядати статистику та отримувати сповіщення про стан системи.

2. Tablet (Планшет): Планшет використовується для візуалізації та моніторингу системи управління сонячними панелями. «На планшеті можна відображати графіки вироблення електроенергії, стан заряду батареї, значення сенсорів та іншу корисну інформацію»[39].

3. Data Link Connector (DLC) (Конектор передачі даних): DLC використовується для забезпечення зв'язку між різними компонентами системи управління. «Він передає дані та команди між смартфоном, планшетом, мікроконтролером та іншими компонентами системи»[39].

4. Solar Panel (Сонячна панель): Сонячна панель є головним джерелом вироблення електроенергії у системі. «Вона перетворює сонячну радіацію на електричний струм. Модель сонячної панелі у Packet Tracer відображає її фізичні параметри, такі як потужність та коефіцієнт ефективності»[39].

5. Power Meter (Електромер): Електромер використовується для вимірювання виробленої сонячною панеллю електроенергії. Він надає інформацію про потужність, напругу та струм, що виробляються сонячною панеллю.

6. Battery (Акумулятор): Акумулятор використовується для зберігання надлишкової електроенергії, яку виробляють сонячні панелі. Це дозволяє використовувати зібрану енергію в періоди, коли сонячна радіація недостатня або вночі.

7. Microcontrollers (MCU) (Мікроконтролери): Мікроконтролери використовуються для програмування та контролю режимів роботи системи.

Вони здійснюють обробку даних від сенсорів, приймають команди від інтерфейсу користувача та керують роботою сонячних панелей та інших компонентів.

8. Photo Sensor (Фотоелемент): Фотоелементи використовуються для вимірювання інтенсивності сонячної радіації. Вони передають дані про освітленість до мікроконтролера, що дозволяє системі адаптуватися до зміни умов освітлення.

9. Motor (Двигун): Двигун використовується для автоматичного відстеження сонячної панелі за рухом сонця. Він підключений до мікроконтролера та рухає панель таким чином, щоб максимізувати отримання сонячної радіації.

10. Temperature Sensor (Температурний сенсор): Температурний сенсор вимірює температуру навколишнього середовища або температуру сонячних панелей. Це дозволяє системі враховувати вплив температури на роботу панелей та проводити оптимізацію енергетичного виробництва.

«Моделювання системи у середовищі Packet Tracer дозволяє візуалізувати та симулювати взаємодію між цими компонентами, тестувати різні алгоритми та налаштування, а також аналізувати ефективність та оптимізувати роботу системи управління сонячними панелями» [35]. Нижче надане місце для подальшого включення рисунків, що ілюструють компоненти та моделювання системи (Рис 3.2).

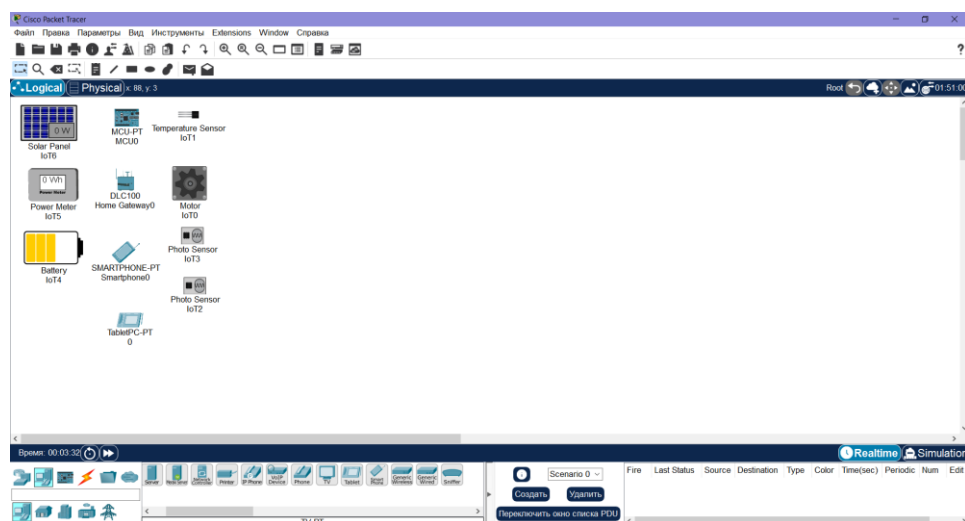


Рис 3.2 Додаємо компоненти Packet Tracer

«Інструмент "IoT Custom Cable" використовується для підключення компонентів у Packet Tracer. Це спеціальний кабель, розроблений специфічно для інтернету речей, який дозволяє з'єднувати різні IoT-пристрої та сенсори з мережею»[39]. За допомогою цього інструменту можна створювати власні кабелі, налаштовувати їх параметри та підключати до відповідних портів компонентів у Packet Tracer. І нам він сильно підходить для нашого проекту, який поданий на Рис 3.3.

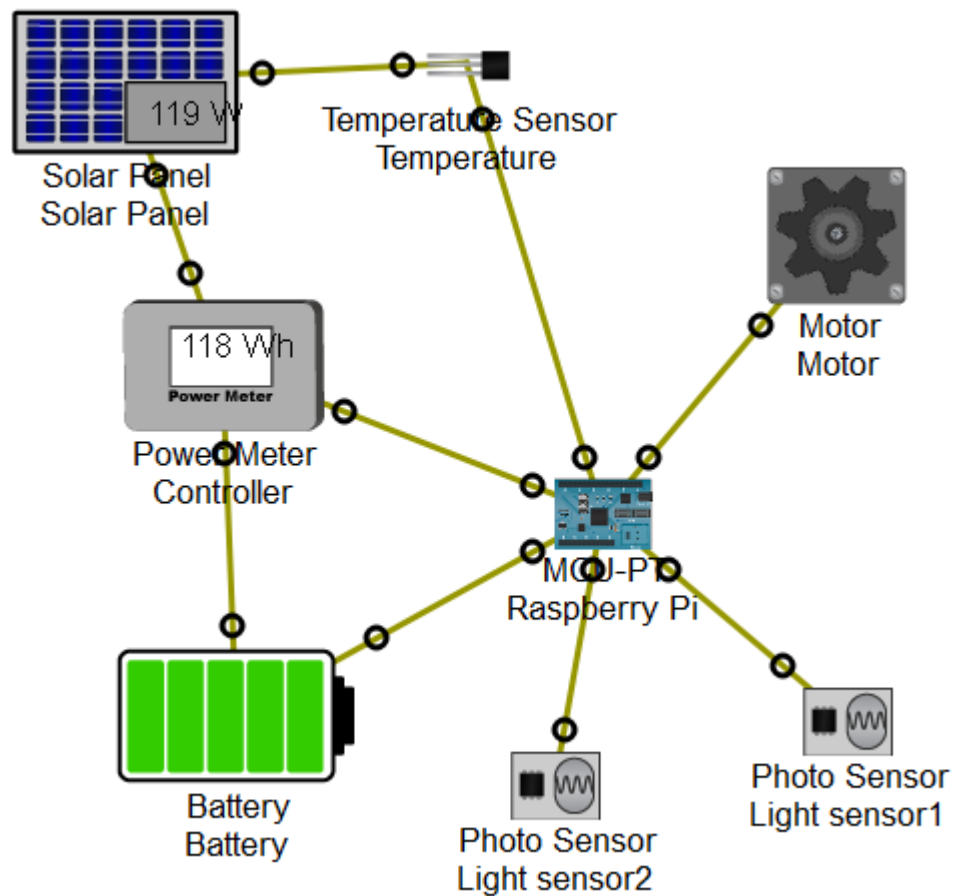


Рис 3.3 Підключення компонентів

З'єднувальний елемент, який дозволяє створювати логічні зв'язки між різними мережевими пристроями, такими як маршрутизатори, комутатори, комп'ютери та інші. За допомогою DLC ви можете налаштовувати з'єднання, встановлювати параметри і передавати дані між компонентами в Packet Tracer.

Щоб підключити до DLC самого потрібно його налаштувати, щоб ніхто не зміг зайти та змінити самі настройки, який показаний на рис 3.4. Після

налаштування підключаємо всі інші компоненти (рис 3.5).

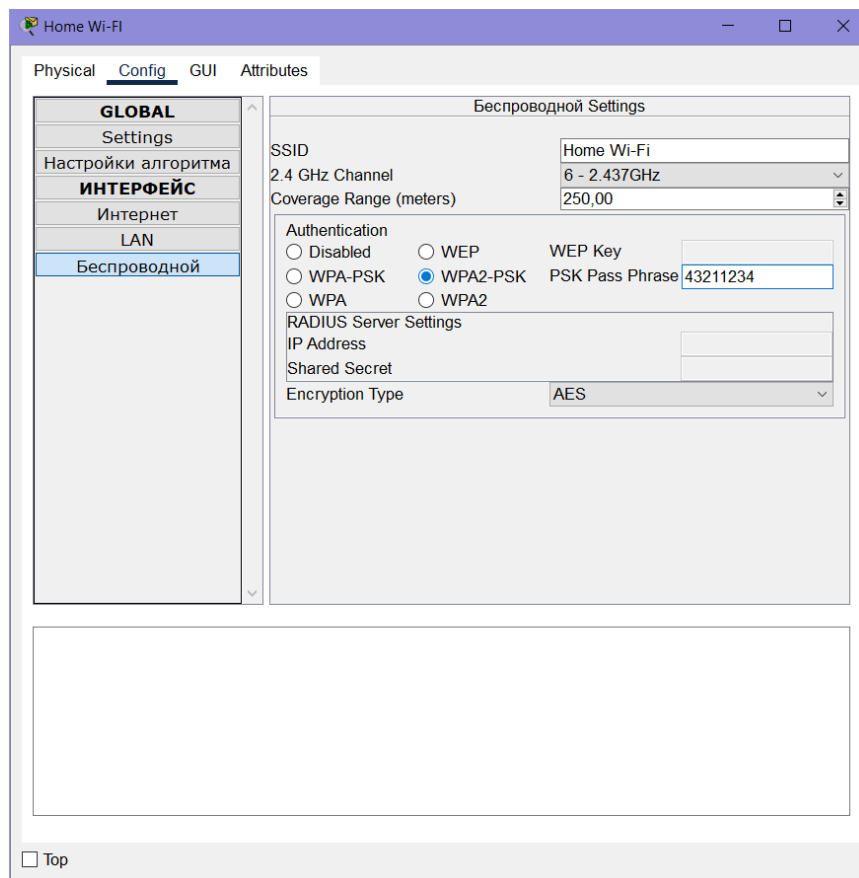


Рис 3.4 Настройка DLC на Home WI-FI

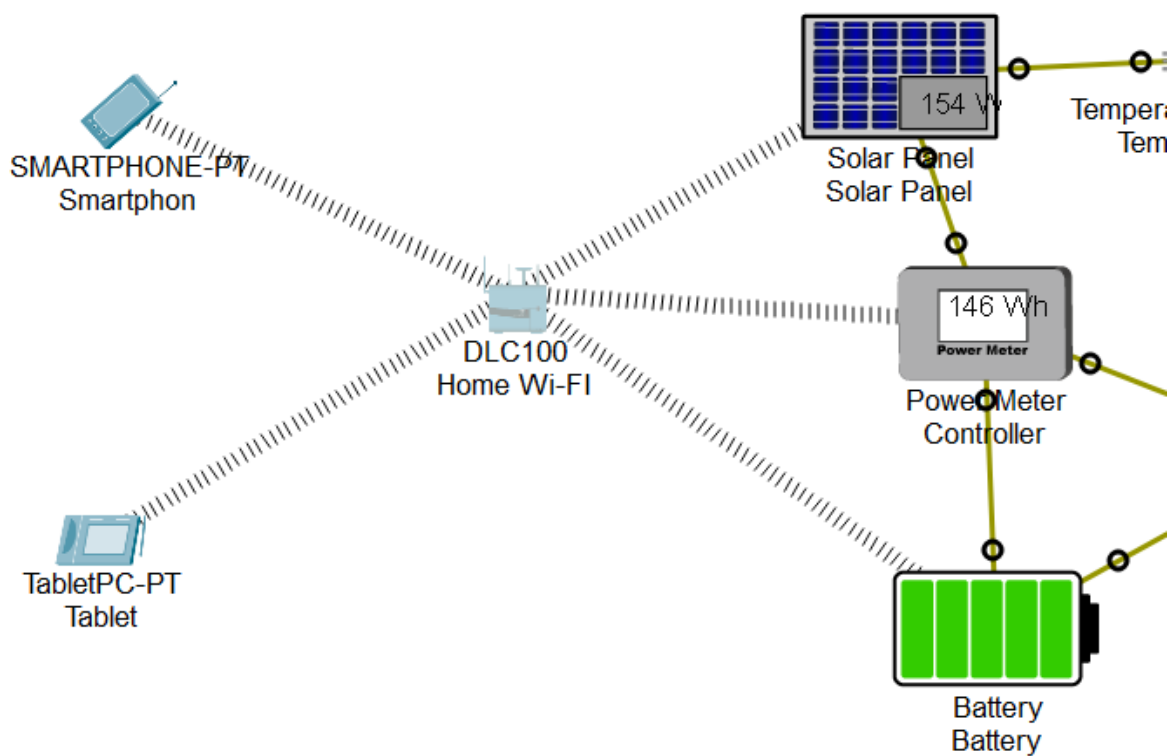


Рис 3.5 Підключення інших компонентів

3.2 Розробка алгоритму та програмування компонентів УСБ

Мікроконтролер (MCU) - це невеликий комп'ютер, побудований на базі системи на кристалі (SoC). Він містить процесорне ядро, пам'ять і програмовані периферійні пристрої вводу/виводу. Мікроконтролери використовуються для вбудованих додатків, які вимагають мало ресурсів комп'ютера. Вони забезпечують керування системою стеження сонячної панелі, зчитування даних з датчиків, обробку даних та відображення інформації на смартфоні або планшеті.

Завдяки підтримці мови програмування Python, можна легко реалізувати складні алгоритми і аналізувати дані, пов'язані з ефективністю сонячних панелей. Можна зчитувати дані з датчиків, які вимірюють сонячне випромінювання, температуру і стан панелей, і використовувати ці дані для оптимального керування сонячними панелями.

За допомогою мови програмування JavaScript можна реалізувати інтерактивний інтерфейс для керування сонячними панелями через веб-додаток або мобільний додаток. Користувачі зможуть зручно моніторити та керувати станом сонячних панелей, встановлювати режими роботи і отримувати повідомлення про проблеми або несправності.

Також, мікроконтролер може бути програмований для автоматичного відстеження сонячного світла, що дозволяє оптимально розташовувати панелі для отримання максимальної сонячної енергії протягом дня. Це дозволяє створити гнучку та ефективну систему управління сонячними панелями, що забезпечує оптимальне використання сонячної енергії.

Система автоматичного керування сонячними панелями використовує код для керування положенням панелей, враховуючи виміри фотодатчиків та інших сенсорів. Код також включає керування пристроями на основі вимірів сенсорів і умовних операторів.

Давайте розглянемо його по кроках:

```
from gpio import *
```

```

bateria = 'A0'          # Змінна, що вказує на вхідний порт для батареї
motor = 'A2'            # Змінна, що вказує на вихідний порт для мотора
photo_sensor1 = 'A3'    # Змінна, що вказує на вхідний порт для
фотодатчика 1
photo_sensor2 = 'A1'    # Змінна, що вказує на вхідний порт для
фотодатчика 2
temp = 'A1'             # Змінна, що вказує на вихідний порт для
температури
def get_angle_difference(photo_sensor1, photo_sensor2):
    # Функція обчислення різниці між кутами, отриманими від
    фотодатчиків
    difference = photo_sensor1 - photo_sensor2

def setup():
    pinMode(bateria, INPUT)    # Налаштування вхідного порту для
    батареї
    pinMode(motor, OUTPUT)     # Налаштування вихідного порту для
    мотора
    pinMode(temp, OUTPUT)      # Налаштування вихідного порту для
    температури
    print("Blinking")
def loop():
    ba = analogRead(bateria) / 4 # Зчитування значення батареї та
    нормалізація
    if ba >= 10:
        digitalWrite(motor, HIGH) # Вмикаємо мотор, якщо значення
        батареї >= 10
        print("Encendido: " + str(ba))
    else:
        digitalWrite(motor, LOW)  # Вимикаємо мотор, якщо значення
        батареї < 10
        print("Apagado: " + str(ba))
        digitalWrite(temp, HIGH)   # Установка високого рівня на
        вихідному порту для температури
        delay(80)
        digitalWrite(temp, LOW)    # Установка низького рівня на
        вихідному порту для температури
        delay(50)
        temperature = analogRead(temp)

```

```

print("Temperature:" + str(temp)) # Отримання значення температури за
допомогою функції getTemperature()
    if temperature >= 80:
        codeEnabled = False          # Встановлення змінної codeEnabled в
False, якщо температура >= 80
    elif temperature <= 50:
        codeEnabled = True           # Встановлення змінної codeEnabled в
True, якщо температура <= 50
    angle_difference = get_angle_difference(photo_sensor1, photo_sensor2)
# Обчислення різниці кутів фотодатчиків
    if (photo_sensor1 < 50 or photo_sensor2 < 50) and angle_difference < 0:
        digitalWrite(motor, IN)      # Вмикаємо мотор, якщо виконуються
певні умови
    elif (photo_sensor1 < 50 or photo_sensor2 < 50) and angle_difference >
0:
        digitalWrite(motor, IN)      # Вимикаємо мотор, якщо виконуються
певні умови
    else:
        digitalWrite(motor, OUT)      # Вимикаємо мотор у всіх інших
випадках

```

Опис функцій та змінних:

Змінні `bateria`, `motor`, `photo_sensor1`, `photo_sensor2` та `temp` вказують на пін-номери (наприклад, A0, A1) для підключення вхідних/вихідних пристроїв до мікроконтролера.

Функція `get_angle_difference` обчислює різницю між кутами, отриманими від двох фотодатчиків.

Функція `getTemperature` відповідає за отримання значення температури. Логіку отримання значення потрібно реалізувати окремо.

Функція `setup` виконує початкові налаштування вхідних/вихідних портів та виводить повідомлення "Blinking".

Функція `loop` містить головний цикл програми, який виконується постійно. У циклі зчитуються значення батареї, вмикається/вимикається мотор, керується температурою, обчислюється різниця кутів фотодатчиків та керується мотором залежно від умов.

Змінні `angle_difference`, `temperature` та `codeEnabled` є глобальними, що

дозволяє їм зберігати значення між ітераціями циклу loop.

На екрані смартфона або планшета ви побачите інформацію про стан системи стеження. Ця інформація включає показники потужності, які виробляються сонячною панеллю, і може бути відображена у ватах (W) або відсотках від максимальної потужності. Крім того, на екрані буде показано рівень заряду акумулятора, який зберігає вироблену сонячною панеллю енергію. Це дає вам інформацію про те, наскільки повністю акумулятор заряджений. Загалом, цей екран дозволяє вам візуально контролювати ефективність роботи системи стеження та її загальний стан рис 3.6.

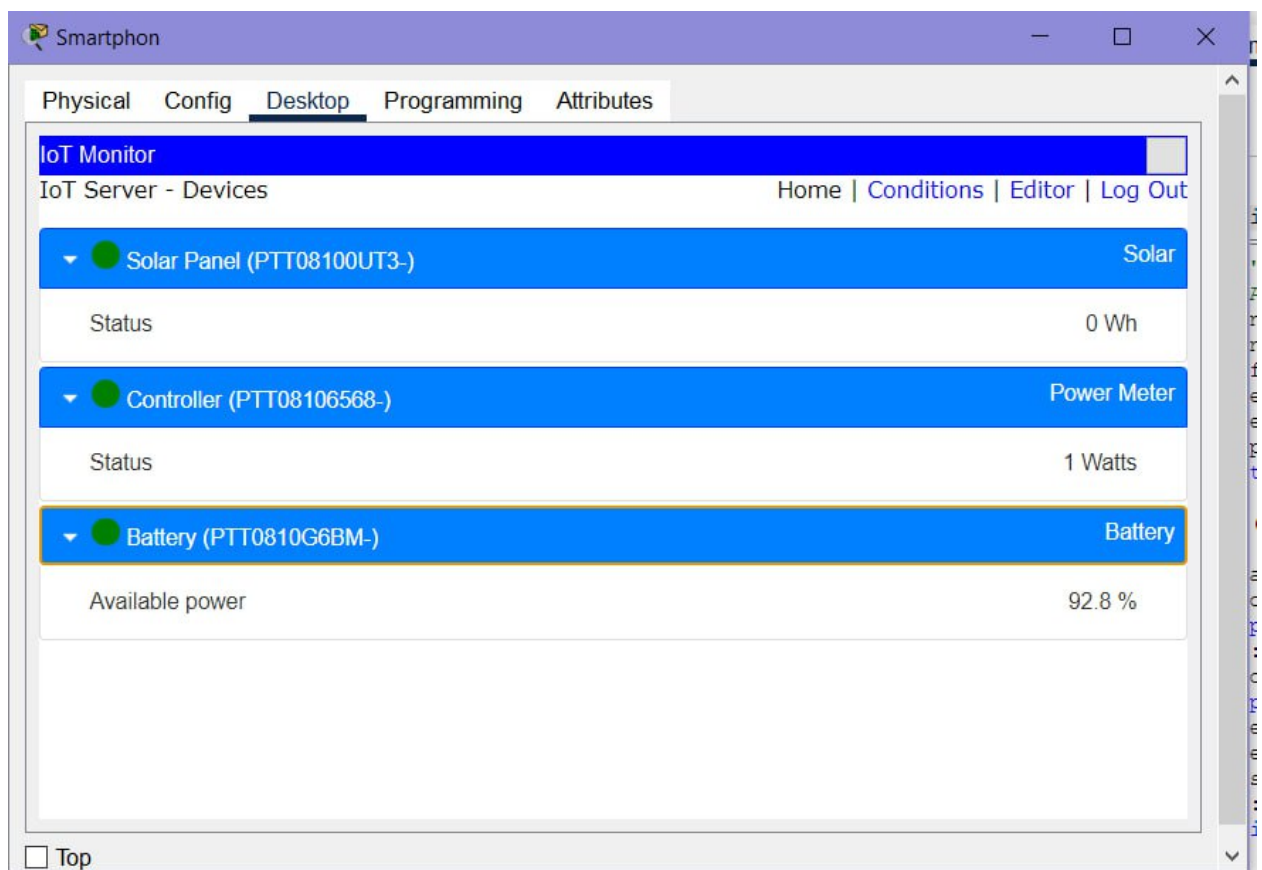


Рис 3.6 Стеження та її загальний стан самої системи

ВИСНОВОК

У першому розділі моєї досвідченої роботи над дипломним проектом була приділена особлива увага детальному вивченню альтернативної енергетики та принципу роботи сонячних електростанцій. Я зосередився на аналізі кількох патентів, що стосуються цієї проблеми, і на їх основі розробив систему автоматичного керування.

Запропонована система дозволяє використовувати отриману енергію для живлення різних постійних навантажень або зберігати її в акумуляторних батареях для майбутнього використання. Крім того, я дослідив роботу системи під час пікового навантаження, коли додаткове живлення надається сумою струмів від сонячної батареї та акумулятора.

Запропоновані результати ефективно зменшують витрати на обслуговування, повністю усувають витрати на паливо і, головне, знижують викиди шкідливих речовин в атмосферу. У другому розділі, після аналізу різних типів та характеристик сонячних електростанцій, я розробив основні схеми автоматизації.

Оскільки продуктивність сонячних панелей прямо залежить від кількості сонячної енергії, яку вони отримують, головна задача полягала в поліпшенні існуючих рішень для позиціонування сонячних панелей відносно сонця. Цю проблему було вирішено за допомогою трекінгової системи, яка дозволяє досягти максимальної генерації електроенергії, коли поверхня сонячних модулів розташовується перпендикулярно потоку сонячного випромінювання.

«Оскільки акумуляторні батареї є одними з найбільш коштовних елементів сонячних електростанцій, вони вимагають особливої уваги. Основними завданнями для підтримки їх нормального функціонування є контроль за їхньою температурою, щоб уникнути перегріву та пошкодження акумуляторів. Розроблена програма симуляції дозволяє встановлювати порогові значення температури, при досягненні яких автоматично припиняється подача струму на акумулятори» [31].

Іншою задачею було забезпечення правильного повороту сонячної панелі, і для її розв'язання я розробив алгоритм автоматизованої роботи, на основі якого була побудована блок-схема. За допомогою цієї блок-схеми було створено відповідний код на мові програмування Python, який найкраще відповідає вимогам.

У третьому розділі я розробив модель, що також працює на мові програмування Python. З використанням Packet Tracer я побудував саму модель і її компоненти, а також розробив систему керування сонячною панеллю. За допомогою фотосенсорів модель вимірює різницю потенціалів під час зміщення сонця, і, відповідно до цих вимірювань, активується двигун, який повертає сонячну панель в оптимальне положення для максимальної ефективності отримання сонячної енергії.

Це підтверджує адекватність запропонованої імітаційної моделі, яка дозволяє більш точно оцінити потужність, що буде виробляти сонячна панель.

Пропонована система не може повністю забезпечити ваш будинок електроенергією, але в разі аварійного відключення основного джерела електроенергії і при наявності повністю заряджених акумуляторів вона здатна забезпечити ваш будинок енергією протягом дня.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Андреев І.В., Барабаш О.В. Інформаційна система реального часу для моніторингу параметрів роботи домашньої сонячної електростанції. Матеріали XIX Міжнародної науково–практичної конференції молодих вчених і студентів «Сучасні проблеми наукового забезпечення енергетики», м. Київ, 20–23 квітня 2021 року. Київ, НТУУ «КПІ імені Ігоря Сікорського», 2021. Том 2. С. 159 – 160.
2. Архів ядра Linux //GPIO Interfaces URL: <https://www.kernel.org/doc/html/latest/driver-api/gpio/intro.html#gpio-interfaces>
3. Вивчайте технології та творіть з Raspberry Pi // SSH (Secure Shell). URL: <https://www.raspberrypi.org/documentation/remote-access/ssh/>
4. В. Ван і С. Лі, "Модель прогнозованого керування двовісним сонячним трекером для сонячної енергетичної системи", Збірник наукових праць сонячної енергетичної системи", Матеріали 31-ї Китайської конференції з управління, 25-27 липня, 2012 року.
5. Гровер Р. Підхід на основі конкуренції для бакалаврської освіти з мехатроніки Education Using the Arduino Platform / International Conference of Teaching. Оцінювання та навчання. - Бангкок : Університет Чулалонгкорн, 2014. - Р. 78-83.
6. Даффі, Д. А., Бекман, В. А., і Блер, Н., "Сонячна інженерія теплових процесів, фотовольтаїки та вітру". (2020). John Wiley & Sons.
7. Деградація сонячних панелей (Potential Induced Degradation) URL: <https://eco-tech.com.ua/ua/a260708-degradatsiya-solnechnyh-panelej.html>
8. Для чого потрібен контролер заряду для сонячних батарей URL: <https://soncedim.com.ua/blog/dlia-chogo-potriben-kontroler-zariadu#>
9. Загальні відомості про системи диспетчерського управління та збору даних – системи SCAD: URL: <http://dssbi.com.ua/sitelab9/blog/hello-world/> – Загол. з титул екрану. – Мова: укр. 18. FLProg - альтернативне середовище програмування Arduino. Опис проекту. [Електронний ресурс]: URL: [http://it-ua.info/news/2014/11/05/flprog-alternativna-seredovische-](http://it-ua.info/news/2014/11/05/flprog-alternativna-seredovische)

programuvannya-arduino-opis-proektu.html#cut/

10. Контролери та їх програмне забезпечення URL: <https://sites.google.com/site/iasunuft/disciplini/ktpapz>

11. Мостефа Гассул, "Розробка автоматичної системи відстеження сонячної енергії для "Design of a Automatic Solar Tracking System to International Journal of New Technology and Advanced Engineering, том 3, випуск 5, травень 2013 р. Віттмер Б., Мермуд А. Інструмент для оптимізації розташування наземних фотоелектричних установок з урахуванням економічних граничних умов // 29th European Європейська конференція та виставка з фотоелектричної сонячної енергетики. - 2014. - Р. 6.

12. Навчальна система Adafruit // Налаштування вашого Pi для I2C. URL: <https://learn.adafruit.com/adafruit-16-channel-servo-driver-with-raspberry-pi/configuring-your-pi-for-i2c>

13. Нур Мохммад, Тарекул Карім, "Проектування та впровадження гібридної автоматичної системи стеження за сонячною енергією", International Journal Of Electrical & Power Engineering, Volume 6, Issue 3-2012

14. Одноосьові трекери підвищують ефективність сонячної станції на 15 - 17% URL: <https://ussolar.systems/ua/news/6-odnoosnye-trekery-povyshayut-effektivnost-solnechnoy-stantsii-na-15-17>

15. Оліфер В. Г., Оліфер Н. О. Комп'ютерні мережі. Принципи, технології, протоколи: Ювілейне видання. - Підручник для вузів, 2020. - С.1008. ISBN:978-5-4461-1426-9.

16. Опис руху сонця. URL: <https://prel.prom.ua/a279654-sonyachnij-treker-scho.html>

17. Основні характеристики сонячних модулів: метод. вказівки / Д.С. Фалеев, 2013. Л.С. Лунін, А.С. Пашенко. Моделювання та дослідження характеристик фотоелектричних перетворювачів на основі GaAs та GaSb / Журнал технічної фізики, 2011, том 81, вип. 9. – С.71-76.

18. Перспективні технології фотоелектричної сонячної енергетики URL: <https://razumkov.org.ua/statti/perspektyvni-tekhnologii-fotoelektrychnoi->

soniachnoi-energetyky

19. Принципи роботи хімічних джерел струму URL: <https://ukrbukva.net/page,6,122328-Principy-raboty-himicheskikh-istochnikovtoka.html> – Загол. з титул екрану. – Мова: укр.

20. Причини втрати продуктивності та основні дефекти сонячних батарей URL: <https://solarsystem.com.ua/ru/prychyny-vtraty-produktyvnosti-sonyachnyh-batarej-osnovni-defekty/>

21. Проект HTTP-сервера Apache // Компіляція та встановлення HTTP-сервера версія 2.4 URL: <https://httpd.apache.org/docs/current/install.html>

22. Рєзцов В.Ф., Матях С.В., Кудреватих О.О. Інтерактивна карта потенціалу сонячної енергії України /Відновлювана енергетика. – 2018. – № 4 (55). – С. 34-42.

23. Система навчання Adafruit // Adafruit 16-канальний серводрайвер з Raspberry Pi з використанням бібліотеки Adafruit. URL: <https://learn.adafruit.com/adafruit-16-channel-servo-driver-with-raspberry-pi/using-the-adafruit-library>

24. Сонячний трекер для збільшення продуктивності сонячних панелей URL: <https://solarsystem.com.ua/zbilshennya-produktyvnosti-sonyachnyh-panelej-sonyachnyj-treker/>

25. Сонячні контролери - які вони бувають і для чого потрібні URL: <https://www.solargarden.com.ua/kontroler-zaryadu-sonyachnyh-batarej-pryznachennya-ta-osoblyvosti-vyboru/?movaUK=UK>

26. Стартовий комплект "Розумний дім" Fibaro Starter Kit (FIB_Start)- назва з екрану // "Технології для життя". - URL: <http://sweetcheap.online>

27. Трекери для сонячних панелей: конструкція, принцип роботи, ефективність URL: <https://www.solargarden.com.ua/trekery-dlya-sonyachnyh-panelej-konstruktsiya-pryntsyp-roboty-efektyvnist/>

28. Хані, С. та Г. Сантосо, "Покращення ємності сонячних елементів за допомогою лінзового концентратора Френеля з сонячною енергією". Lens Concentrator with Solar Tracker Control". Міжнародний журнал наукових

Engineering and Science. (2018). 2(7): p. 41-44.

29. Що таке Cisco Packet Tracer URL:
<http://www.packettracernetwork.com/>

30. Ammach, S. and A. Attia. "Проектування та впровадження автономної енергоефективної системи відстеження сонячної енергії для фотоелектричних електростанцій". у 2019 році Міжнародна Міжнародна конференція з електричних та комп'ютерних технологій та застосувань (ICESTA)(2019).

31. EasySmartBox - готові системи Розумний Дім від виробника // EasySmartBox. - URL: <http://easysmartbox.com>

32. Inter-Integrated Circuit. URL: <https://electrosome.com/i2c/#Disadvantages>

33. MQTT: The Standard for IoT Messaging. - Назва з екрана. – URL: <https://mqtt.org/>

34. NumPy // Посібник з NumPy v1.20 <https://numpy.org/doc/stable/>

35. Packet Tracer 8.2 - конфігурація пристроїв IoT URL:
<https://www.packettracernetwork.com/internet-of-things/pt7-iot-devices-configuration.html>

36. PyPI Індекс пакету Python // Ця бібліотека Python для Raspberry Pi дозволяє легко використовувати складну функціональність датчика INA219 від Texas Instruments для вимірювання напруги, струму та потужності. URL: <https://pypi.org/project/pi-ina219/>

37. Raspberry Pi Spy - неофіційні підручники, посібники та новини про Pi // Використання модуля I2C OLED-дисплея з Raspberry Pi URL: <https://www.raspberrypi-spy.co.uk/2018/04/i2c-oled-display-module-with-raspberry-pi/>

38. S. Mansour. Optimum Design Of On Grid Pv System Using Tracking System / D.W.R.Anis, D.I.M. Hafez // Sci. Technol. Res. – 2015. – №05. – pp. 50–57.

39. Sun, Liangxu, et al. "Порівняння між фізичними пристроями та

симулятором програмного забезпечення для викладання мережевих технологій Cisco". Computer Science & Education (ICCSE), 2013 8th International Conference on IEEE, 2013.

40. T. Twisha. Introducing Dual Axis Solar Tracker with Reflector to Increase Optimal Electricity Generation in Bangladesh // Dev. Renew. Energy Technol. – 2014. – № 3. – pp.1–6.

41. Tech, Learn and Make with Paspberry Pi. URL: <https://www.raspberrypi.org/>

02.06.2023 p.

