

КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів
на базі месенджер-платформ

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ 022075.22.04.26

Виконав здобувач IV курсу, група KI2-22-4



Підпис

Тарас РАДАЙКІН

Ініціали, прізвище

Керівник кандидат техн. наук, доцент

Науковий ступінь, учене звання



Підпис

Олексій ІВАНОВ

Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц.

Науковий ступінь, учене звання



Підпис

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІІС



Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«07» червня 2026 р.

дата

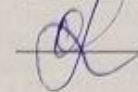
Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ
Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІПС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Радайкіну Тарасу Романовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ

Керівник проекту (роботи) Іванов Олексій Валентинович, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

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

Опис програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ

Проектування системи дистанційного контролю енергозабезпечення

Реалізація та перевірка роботи програмно-технічного засобу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Схема цифрової моделі керування генератором

Електрична принципова схема

Алгоритм роботи програмно-технічного засобу


6. Консультанти розділів кваліфікаційної роботи

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


7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконав
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконав
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконав
4	Робота над розділом 2 – Проектування системи дистанційного контролю енергозабезпечення	01.04.2026	виконав
5	Робота над розділом 3 – Реалізація та перевірка роботи програмно-технічного засобу	29.04.2026	виконав
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконав
7	Попередній захист ВКР	26.05.2026	виконав
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач  Підпис

Тарас РАДАЙКІН
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи  Підпис

Олексій ІВАНОВ
Ім'я, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ».

Автор роботи: Тарас РАДАЙКІН.

Керівник роботи: Олексій ІВАНОВ.


Пояснювальна записка: 85 с., 8 рис., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

АВТОМАТИЧНИЙ ЗАПУСК, ВІДДАЛЕНИЙ КОНТРОЛЬ, ЕНЕРГОЗАБЕЗПЕЧЕННЯ, МІКРОКОНТРОЛЕР, ТЕЛЕГРАМ-БОТ

Кваліфікаційна робота бакалавра присвячена розробці та дослідженню програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ. Актуальність теми зумовлена зростанням потреби в надійному, зручному та оперативному контролі автономних систем живлення, зокрема акумуляторних батарей і генераторів, які використовуються для забезпечення безперервної роботи побутових, офісних та інфокомунікаційних об'єктів в умовах нестабільного електропостачання. Своєчасне отримання інформації про рівень заряду акумуляторної системи, стан генератора та критичні події дає змогу зменшити ризик раптового знеструмлення, підвищити ефективність використання резервного живлення та забезпечити більш зручне керування енергозабезпеченням об'єкта.

Метою роботи є проектування, реалізація та тестування системи для контролю рівня заряду, стану генератора й інформування користувача через Telegram-бота. У межах роботи обґрунтовано вибір апаратної платформи, розроблено структурну та функціональну схеми, програмне забезпечення мікроконтролера і механізми сповіщення про основні події та аварійні ситуації.


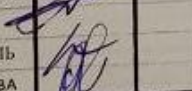
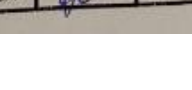

Підпис здобувача

30.05.2026
Дата

ЗМІСТ

ВСТУП.....	4
1 Опис програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ.....	7
1.1 Аналіз предметної області віддаленого контролю енергозабезпечення об'єктів.....	7
1.2 Призначення та функціональні можливості програмно-технічного засобу	11
1.3 Загальна структура програмно-технічного засобу	16
1.4 Аналіз апаратної та програмної частин системи	21
1.5 Висновки до першого розділу	28
2 Проекування системи дистанційного контролю енергозабезпечення.....	30
2.1 Визначення вимог до програмно-технічного засобу.....	30
2.2 Вибір компонентів апаратної частини	35
2.3 Розроблення структурної схеми засобу	40
2.4 Розроблення алгоритму контролю рівня заряду акумуляторної системи	45
2.5 Розроблення алгоритму ручного та автоматичного керування генератором	50
2.6 Розроблення логіки Telegram-бота	55
2.7 Висновки до другого розділу.....	60
3 Реалізація та перевірка роботи програмно-технічного засобу	62
3.1 Підключення апаратних компонентів програмно-технічного засобу ...	62
3.2 Розроблення програмного забезпечення для ESP32.....	65
3.3 Налаштування Telegram-бота та реалізація команд керування	69
3.4 Тестування роботи програмно-технічного засобу.....	73
3.5 Висновки до третього розділу	76

КвРКІ 022075.22.04.08

Зм.	Арк.	№ докум.	Підпис	Дата	Літера	Аркуш	Аркушів
Виконав		Тарас РАДАЙКІН			у	2	88
Перевір.		Олексій ІВАНОВ			ХНУ КІ2-22-4		
Н.контр.		Тетяна КИСІЛЬ					
Затвер.		Ольга ПАВЛОВА					

Висновки.....	79
Перелік джерел посилань	81
Додаток А Схема цифрової моделі керування генератором.....	86
Додаток Б Електрична принципова схема	87
Додаток В Алгоритм роботи програмно-технічного засобу.....	88

ВСТУП

Актуальність дослідження. Стабільне енергозабезпечення є однією з важливих умов безперервної роботи побутових, офісних, виробничих та інформаційно-комунікаційних об'єктів. Порушення основного електроживлення може призводити до зупинки обладнання, втрати даних, зниження безпеки об'єкта та погіршення умов експлуатації технічних систем. Для зменшення таких ризиків широко застосовуються системи резервного енергозабезпечення, до складу яких можуть входити акумуляторні батареї, інвертори, зарядні пристрої та генератори.

Однак наявність резервного джерела живлення сама по собі не гарантує надійності роботи об'єкта. Важливим є своєчасний контроль стану акумуляторної системи, рівня її заряду, факту переходу на резервне живлення, а також стану генератора. Якщо користувач не отримує інформацію про зниження заряду акумуляторів або про необхідність запуску генератора, резервна система може втратити працездатність у критичний момент. Особливо це актуально для об'єктів, які не мають постійного чергового персоналу або перебувають на відстані від власника.

Сучасні мікроконтролерні платформи дають змогу створювати компактні та доступні програмно-технічні засоби для моніторингу технічних параметрів. Зокрема, мікроконтролер ESP32 має вбудований Wi-Fi-модуль, достатню кількість цифрових та аналогових входів/виходів, підтримку поширених середовищ програмування та може використовуватися для побудови простих систем Інтернету речей. Завдяки цьому на його основі можна реалізувати пристрій, який вимірює параметри акумуляторної системи, обробляє отримані дані та передає повідомлення користувачу через мережу Інтернет.

Одним із зручних способів взаємодії користувача з віддаленим пристроєм є використання месенджер-платформ. Застосування Telegram-бота дозволяє не створювати окремий мобільний застосунок або серверну вебпанель, а

					КВРКІ 022075.22.04.26	Арк.
						4
Зм.	Арк.	№ докum.	Підпис	Дата		

використовувати готовий інтерфейс для отримання повідомлень і надсилання команд. Такий підхід зменшує складність реалізації, спрощує доступ користувача до системи та забезпечує збереження історії подій без створення окремої бази даних.

У межах даної кваліфікаційної роботи розглядається програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів на базі мікроконтролера ESP32, Wi-Fi-з'єднання та Telegram-бота. Засіб призначений для контролю рівня заряду акумуляторної системи, повідомлення користувача про зміну стану енергозабезпечення, ручного керування генератором та автоматичного запуску генератора при досягненні критичного рівня заряду. Для формалізації логіки керування також передбачається використання апаратної моделі у середовищі Intel Quartus Prime, яка відображає принцип роботи вузла прийняття рішень щодо запуску генератора.

Актуальність теми полягає у необхідності створення простого, доступного та функціонально достатнього засобу, який дозволяє користувачу віддалено контролювати стан резервного енергозабезпечення без використання складної серверної інфраструктури, GSM-модулів або спеціалізованого програмного забезпечення. Запропонований підхід є доцільним для невеликих об'єктів, де потрібно забезпечити базовий контроль акумуляторної системи та керування генератором з мінімальною кількістю апаратних компонентів.

Метою кваліфікаційної роботи є розроблення програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів на базі мікроконтролера ESP32 та месенджер-платформи Telegram, який забезпечує моніторинг рівня заряду акумуляторної системи, інформування користувача про стан енергозабезпечення та ручне або автоматичне керування запуском генератора.

Об'єктом дослідження є процес віддаленого контролю та керування системою резервного енергозабезпечення об'єкта.

					КвРКІ 022075.22.04.26	Арк. 5
Зм.	Арк.	№ докum.	Підпис	Дата		

Предметом дослідження є методи, алгоритми та програмно-апаратні засоби моніторингу рівня заряду акумуляторів, передавання повідомлень через месенджер-платформу, а також дистанційного й автоматичного керування запуском генератора.

Практичне значення роботи полягає у створенні макета програмно-технічного засобу, який може бути використаний для базового віддаленого контролю резервного енергозабезпечення невеликого об'єкта. Розроблений засіб дозволяє користувачу отримувати повідомлення про зниження заряду акумуляторної системи, переглядати поточний стан системи, запускати або вимикати генератор через Telegram-бота, а також активувати режим автоматичного запуску генератора при досягненні критичного рівня заряду.

У роботі прийнято обмеження, відповідно до яких зв'язок із користувачем реалізується лише через Wi-Fi та Telegram-бота. Використання GSM-модуля, SIM-карти, окремої серверної частини або мобільного застосунку не передбачається. Такий підхід дозволяє зосередитися на розробленні простої та надійної системи з мінімально необхідним складом апаратних і програмних компонентів.

Пояснювальна записка складається зі вступу, трьох розділів, висновків, переліку джерел посилання та додатків. У першому розділі розглядаються призначення, функціональні можливості та загальна структура програмно-технічного засобу. У другому розділі виконується проєктування системи, обґрунтовується вибір компонентів, розробляються структурна схема та алгоритми роботи. У третьому розділі описується програмно-апаратна реалізація засобу, налаштування Telegram-бота та результати тестування основних функцій системи.

Наступним етапом роботи є аналіз предметної області та опис особливостей побудови програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів.

					КВРКІ 022075.22.04.26	Арк. 6
Зм.	Арк.	№ докum.	Підпис	Дата		

1 ОПИС ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ВІДДАЛЕНОГО КОНТРОЛЮ ЕНЕРГОЗАБЕЗПЕЧЕННЯ ОБ'ЄКТІВ НА БАЗІ МЕСЕНДЖЕР-ПЛАТФОРМ

1.1 Аналіз предметної області віддаленого контролю енергозабезпечення об'єктів

Стабільне електроживлення є необхідною умовою функціонування сучасних технічних об'єктів, до складу яких входять побутове обладнання, комп'ютерні системи, мережеві пристрої, системи відеоспостереження, охоронні комплекси, засоби зв'язку та інші електронні пристрої. Навіть короткочасне зникнення основного живлення може призвести до припинення роботи обладнання, втрати інформації, зниження рівня безпеки або порушення технологічних процесів. Тому для підвищення надійності електропостачання використовують системи резервного енергозабезпечення.

Резервне енергозабезпечення невеликих об'єктів зазвичай реалізується на основі акумуляторної системи, інвертора, зарядного пристрою та генератора. Акумуляторна система забезпечує живлення навантаження протягом певного часу після зникнення основної мережі. Генератор використовується як додаткове джерело енергії у випадках тривалого вимкнення електропостачання або зниження рівня заряду акумуляторів до критичного значення. Така структура дозволяє підтримувати працездатність об'єкта навіть за відсутності основного джерела живлення.

Основною проблемою експлуатації таких систем є необхідність постійного контролю їхнього стану. Користувач повинен своєчасно отримувати інформацію про рівень заряду акумуляторів, факт зникнення або відновлення основного живлення, запуск чи зупинку генератора, а також про можливі порушення в роботі резервного джерела енергії. Якщо така інформація відсутня, користувач може не встигнути вчасно запустити генератор або зменшити навантаження на

					КвРКІ 022075.22.04.26	Арк. 7
Зм.	Арк.	№ докum.	Підпис	Дата		

акумуляторну систему. У результаті об'єкт може залишитися без електроживлення у критичний момент.

Для невеликих об'єктів часто недоцільно використовувати складні промислові системи диспетчеризації, оскільки вони потребують значних фінансових витрат, налаштування серверної інфраструктури та професійного обслуговування. У таких умовах доцільним є створення простого програмно-технічного засобу, який виконує базові функції моніторингу та керування, але не потребує складного впровадження. Такий засіб повинен мати мінімальну кількість компонентів, бути доступним за вартістю, простим у налаштуванні та достатнім для контролю основних параметрів системи резервного енергозабезпечення.

Одним із ключових параметрів, який необхідно контролювати, є рівень заряду акумуляторної системи. У практичних пристроях для цього може використовуватися вимірювання напруги акумулятора з подальшим перерахунком у відсоткове значення заряду. Такий підхід не забезпечує високоточного визначення ємності акумулятора, оскільки реальний рівень заряду залежить від типу батареї, навантаження, температури, віку акумулятора та режиму його експлуатації. Проте для простої системи контролю такий метод є достатнім, якщо потрібно визначити не точну ємність, а наближений стан акумуляторної системи та момент досягнення встановлених порогів.

У межах даної роботи доцільно використовувати три основні пороги інформування користувача: 70 %, 50 % і 30 %. Поріг 70 % може розглядатися як попереджувальний рівень, який повідомляє про початок помітного розряду акумуляторної системи. Поріг 50 % свідчить про необхідність підвищеної уваги до стану живлення та можливість підготовки до запуску генератора. Поріг 30 % визначається як критичний рівень, при якому система повинна або повідомити користувача про небезпеку подальшого розряду, або автоматично запустити генератор, якщо відповідний режим активовано.

					КВРКІ 022075.22.04.26	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім контролю заряду, важливою функцією є керування генератором. У найпростішому варіанті генератор повинен мати можливість дистанційного запуску через зовнішній керуючий контакт або вхід запуску. Для подавання керуючого сигналу може використовуватися релейний модуль, який комутує відповідне коло запуску або зупинки генератора. У практичній реалізації необхідно враховувати особливості конкретного генератора, проте для макетної реалізації достатньо використати релейний модуль як виконавчий елемент, що імітує або виконує команду запуску.

Віддалене керування генератором може здійснюватися у двох режимах: ручному та автоматичному. У ручному режимі користувач самостійно надсилає команду на запуск або зупинку генератора. У автоматичному режимі система самостійно приймає рішення про запуск генератора при досягненні критичного рівня заряду акумуляторної системи. Такий підхід дозволяє зменшити залежність від постійної участі користувача та підвищити надійність резервного енергозабезпечення.

Для реалізації віддаленої взаємодії з користувачем доцільно використовувати месенджер-платформу Telegram. Telegram-бот дає змогу організувати простий інтерфейс керування без створення окремого мобільного застосунку або вебсервера. Користувач може отримувати повідомлення про зміну стану системи, переглядати поточні параметри, а також надсилати команди керування безпосередньо з чату. Додатковою перевагою такого підходу є те, що історія повідомлень у чаті може використовуватися як простий журнал подій.

Зв'язок між пристроєм і Telegram-ботом у даній роботі передбачається через Wi-Fi-мережу. Такий варіант є простішим за використання GSM-модуля, не потребує SIM-карти та зменшує вартість апаратної частини. Водночас він має певне обмеження: система може передавати повідомлення лише за умови доступності локальної Wi-Fi-мережі та підключення до Інтернету. Для невеликого об'єкта це обмеження є прийнятним, якщо мережеве обладнання також підключене до резервного живлення.

					КВРКІ 022075.22.04.26	Арк.
						9
Зм.	Арк.	№ докum.	Підпис	Дата		

Центральним елементом програмно-технічного засобу є мікроконтролер ESP32. Він поєднує обчислювальний модуль, вбудований Wi-Fi, цифрові входи/виходи та аналого-цифрові входи, що дозволяє використовувати його для вимірювання напруги акумуляторної системи, оброблення команд Telegram-бота та керування релейним модулем. Завдяки підтримці середовища Arduino IDE та наявності бібліотек для роботи з Wi-Fi і Telegram API, ESP32 є придатною платформою для створення навчального та прикладного програмно-технічного засобу.

Типова структура системи віддаленого контролю енергозабезпечення включає акумуляторну систему, вузол вимірювання напруги, мікроконтролер, виконавчий елемент, генератор та канал зв'язку з користувачем. Вузол вимірювання напруги передає до мікроконтролера сигнал, що відповідає поточному стану акумулятора. Мікроконтролер обробляє це значення, порівнює його з порогами та формує відповідні повідомлення. У разі отримання команди від користувача або спрацювання автоматичного режиму мікроконтролер керує релейним модулем, який відповідає за запуск або вимкнення генератора.

Окремої уваги потребує формування повідомлень користувачу. Повідомлення повинні бути короткими, зрозумілими та інформативними. Наприклад, при досягненні рівня заряду 70 % користувач отримує попередження про зниження заряду, при 50 % — повідомлення про необхідність контролю стану системи, а при 30 % — повідомлення про критичний рівень заряду. Якщо увімкнено режим автозапуску, система повинна також повідомити користувача про автоматичний запуск генератора. У разі вимкнення генератора через несправність або відсутність палива система може повідомити лише сам факт аварійного вимкнення без додаткової діагностики, оскільки в мінімальній реалізації не передбачається використання окремих датчиків палива або технічного стану генератора.

Таким чином, предметна область віддаленого контролю енергозабезпечення об'єктів охоплює задачі моніторингу стану акумуляторної системи, інформування

					КвРКІ 022075.22.04.26	Арк. 10
Зм.	Арк.	№ докum.	Підпис	Дата		

користувача, керування генератором і забезпечення простої взаємодії через мережевий інтерфейс. Для бакалаврської кваліфікаційної роботи доцільним є створення помірно складного засобу на базі ESP32, який реалізує основні функції контролю та керування без зайвого ускладнення апаратної і програмної частин.

Проведений аналіз показує, що розроблення програмно-технічного засобу віддаленого контролю енергозабезпечення є практично доцільним для невеликих об'єктів, де потрібне оперативне інформування користувача та базове керування генератором. Наступним етапом є визначення призначення та функціональних можливостей розроблюваного засобу, що дозволить сформулювати вимоги до його апаратної та програмної реалізації.

1.2 Призначення та функціональні можливості програмно-технічного засобу

Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів призначений для моніторингу стану акумуляторної системи, інформування користувача про зміну параметрів живлення та керування генератором за допомогою месенджер-платформи Telegram. Основне призначення засобу полягає у забезпеченні оперативного зв'язку між користувачем і системою резервного енергозабезпечення без використання складної серверної інфраструктури, GSM-модуля або окремого мобільного застосунку.

Розроблюваний засіб орієнтований на використання на невеликих об'єктах, де необхідно забезпечити базовий контроль резервного живлення. До таких об'єктів можна віднести приватні будинки, невеликі офіси, технічні приміщення, пункти відеоспостереження, мережеві вузли, охоронні системи та інші об'єкти, для яких важливим є збереження працездатності обладнання під час відсутності основного електроживлення. У таких умовах користувач повинен

мати можливість швидко отримати інформацію про стан акумуляторів і за потреби дистанційно запуснути генератор.

Основою програмно-технічного засобу є мікроконтролер ESP32, який виконує функції збору даних, оброблення інформації, взаємодії з мережею Wi-Fi та керування виконавчим елементом. Вимірювання стану акумуляторної системи здійснюється за допомогою вузла контролю напруги, сигнал з якого надходить на аналоговий вхід мікроконтролера. Після оброблення виміряного значення програмне забезпечення визначає орієнтовний рівень заряду акумуляторної системи та порівнює його із заданими пороговими значеннями.

У межах розроблюваної системи передбачено три основні рівні інформування користувача: 70 %, 50 % і 30 % заряду акумуляторної системи. Досягнення рівня 70 % розглядається як початкове попередження про зниження заряду. Рівень 50 % вказує на необхідність контролю стану системи та можливу підготовку до запуску генератора. Рівень 30 % приймається як критичний, при якому система повинна повідомити користувача про небезпеку подальшого розряду, а за умови активованого режиму автоматичного запуску — подати команду на запуск генератора.

Виконавчим елементом системи є релейний модуль або інший комутаційний пристрій, за допомогою якого мікроконтролер може керувати входом запуску генератора. У мінімальній реалізації релейний модуль використовується для подавання сигналу запуску або зупинки генератора. Такий підхід дозволяє спростити апаратну частину та зосередитися на основній логіці роботи системи: контролі заряду, прийманні команд користувача, формуванні повідомлень і керуванні виконавчим елементом.

Взаємодія користувача із системою здійснюється через Telegram-бота. Telegram-бот виконує роль людино-машинного інтерфейсу, через який користувач може отримувати повідомлення про події та надсилати команди керування. Застосування месенджер-платформи дозволяє спростити доступ до системи, оскільки користувачеві не потрібно встановлювати спеціалізоване

програмне забезпечення або відкривати вебінтерфейс. Усі повідомлення та команди зберігаються в чаті, що додатково виконує функцію простого журналу подій.

Функціональні можливості програмно-технічного засобу можна поділити на три групи: функції контролю, функції інформування та функції керування. До функцій контролю належить періодичне вимірювання напруги акумуляторної системи, перетворення виміряного значення в умовний рівень заряду та визначення поточного стану системи. До функцій інформування належить надсилання повідомлень користувачу при досягненні заданих порогів заряду, зміні стану генератора або зміні режиму автозапуску. До функцій керування належить ручний запуск генератора, ручне вимкнення генератора, увімкнення і вимкнення режиму автоматичного запуску.

Важливою особливістю системи є те, що вона не потребує окремої бази даних для зберігання журналу подій. Події, які виникають під час роботи засобу, надсилаються у чат Telegram, де автоматично зберігаються в історії повідомлень. Це спрощує архітектуру системи та зменшує кількість програмних компонентів. Для мінімальної реалізації такого способу зберігання подій достатньо, оскільки основною метою є не глибокий аналіз статистики, а оперативне інформування користувача про стан енергозабезпечення.

До основних подій, які повинні фіксуватися системою, належать: зниження рівня заряду до заданих порогів, запуск генератора, вимкнення генератора, увімкнення або вимкнення режиму автозапуску, а також повідомлення про аварійне вимкнення генератора. У межах даної роботи під аварійним вимкненням генератора розуміється факт припинення його роботи через несправність або відсутність палива без додаткової деталізованої діагностики. Такий підхід відповідає мінімальній реалізації, оскільки не потребує використання окремих датчиків рівня палива, обертів двигуна або додаткових діагностичних модулів.

Командний інтерфейс Telegram-бота повинен бути простим і зрозумілим для користувача. Команди мають відображати основні дії, які користувач може виконати із системою.

Команда `/status` є однією з основних, оскільки дозволяє користувачу швидко отримати загальну інформацію про стан системи. У відповіді на цю команду доцільно виводити рівень заряду акумуляторів, стан генератора, стан режиму автозапуску, наявність основного живлення, якщо така функція передбачена схемою, та останню зафіксовану подію. Команда `/battery` призначена для окремої перевірки рівня заряду акумуляторної системи без виведення додаткових параметрів.

Команди `/generator_on` і `/generator_off` забезпечують ручне керування генератором. При отриманні команди запуску мікроконтролер активує релейний модуль, після чого користувачу надсилається повідомлення про виконання дії. При отриманні команди зупинки система вимикає релейний модуль або подає сигнал зупинки, залежно від прийнятої схеми керування. У макетній реалізації ці команди можуть бути перевірені за допомогою світлодіодної індикації або релейного модуля без підключення до реального генератора.

Команди `/autostart_generator_on` і `/autostart_generator_off` призначені для керування режимом автоматичного запуску генератора. Якщо режим автозапуску активовано, система самостійно запускає генератор після досягнення критичного рівня заряду акумуляторної системи. Якщо режим вимкнено, система лише інформує користувача про критичний заряд, але не виконує запуск генератора без прямої команди користувача.

З погляду користувача робота системи повинна бути максимально простою. Після увімкнення пристрій підключається до Wi-Fi-мережі, ініціалізує взаємодію з Telegram-ботом і переходить у режим періодичного контролю параметрів. Користувач отримує повідомлення лише при виникненні важливих подій або після надсилання команди. Це дозволяє уникнути надмірної кількості повідомлень і зробити використання системи зручним.

Для запобігання дублюванню повідомлень потрібно передбачити логіку фіксації перетину порогів. Наприклад, якщо рівень заряду знизився до 70 %, система повинна надіслати повідомлення один раз, а не повторювати його при кожному циклі вимірювання. Повторне повідомлення може бути сформоване лише після зміни стану або після нового циклу заряду й розряду акумуляторної системи. Аналогічний підхід слід застосувати для порогів 50 % і 30 %.

Функціонування програмно-технічного засобу можна подати як послідовність взаємопов'язаних процесів: вимірювання параметрів, оброблення даних, прийняття рішення, виконання дії та інформування користувача. На етапі вимірювання система отримує значення напруги акумуляторної системи. На етапі оброблення даних це значення перетворюється в умовний відсоток заряду. На етапі прийняття рішення визначається, чи потрібно надсилати повідомлення або запускати генератор. На етапі виконання дії активується або вимикається виконавчий елемент. Після цього користувачу надсилається відповідне повідомлення через Telegram.

До нефункціональних особливостей системи належать простота апаратної реалізації, доступність компонентів, можливість швидкого налаштування, низька вартість і достатня надійність для використання в навчальному або побутовому макеті. Система не призначена для заміни промислових контролерів автоматичного введення резерву або професійних систем диспетчеризації. Її призначення полягає у вирішенні конкретної прикладної задачі — забезпеченні базового віддаленого контролю стану акумуляторної системи та керування генератором через месенджер-платформу.

Таким чином, розроблюваний програмно-технічний засіб має забезпечувати мінімально необхідний набір функцій для віддаленого контролю резервного енергозабезпечення невеликого об'єкта. Його основними перевагами є використання доступної мікроконтролерної платформи ESP32, відсутність потреби в окремому сервері, простий інтерфейс через Telegram-бота та можливість реалізації автоматичного запуску генератора при критичному рівні

заряду акумуляторної системи. Визначені функціональні можливості є основою для подальшого опису загальної структури програмно-технічного засобу та взаємодії його апаратних і програмних компонентів.

1.3 Загальна структура програмно-технічного засобу

Загальна структура програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів визначається його основним призначенням: отримувати інформацію про стан акумуляторної системи, обробляти її за заданими алгоритмами, передавати повідомлення користувачу через Telegram-бота та виконувати керування генератором у ручному або автоматичному режимі. З огляду на це система будується як сукупність взаємопов'язаних апаратних і програмних компонентів, кожен із яких виконує окрему функцію у загальному процесі контролю та керування.

Розроблюваний засіб має модульну структуру. Такий підхід дозволяє розділити систему на окремі функціональні блоки: блок живлення, блок вимірювання параметрів акумуляторної системи, мікроконтролерний блок, блок виконавчого керування генератором, блок мережевої взаємодії та користувацький інтерфейс на основі Telegram-бота. Модульність спрощує проєктування, налагодження та тестування системи, оскільки кожен блок може бути перевірений окремо.

Основним елементом програмно-технічного засобу є мікроконтролер ESP32. Він виконує роль центрального керуючого пристрою, який приймає сигнали від вимірювального вузла, обробляє отримані дані, порівнює їх із заданими пороговими значеннями, формує рішення щодо подальших дій і забезпечує зв'язок із Telegram-ботом через Wi-Fi-мережу. Крім того, ESP32 керує релейним модулем, який використовується для запуску або вимкнення генератора.

До складу апаратної частини входить акумуляторна система, яка є основним об'єктом контролю. Її стан визначається за напругою, що вимірюється за допомогою спеціального вузла вимірювання. У найпростішому варіанті такий вузол може бути реалізований на основі резистивного дільника напруги або готового модуля вимірювання напруги, який забезпечує подання безпечного рівня сигналу на аналоговий вхід ESP32. Оскільки аналогові входи мікроконтролера не розраховані на безпосереднє підключення до напруги акумуляторної системи, вимірювальний вузол є обов'язковим елементом захисту та узгодження рівнів сигналу.

Акумуляторна система у структурі засобу виконує дві ролі. По-перше, вона є джерелом резервного живлення для об'єкта. По-друге, вона є джерелом інформації для системи контролю, оскільки за її напругою визначається орієнтовний рівень заряду. Під час роботи пристрою ESP32 періодично отримує значення напруги, перераховує його в умовний відсоток заряду та визначає, чи досягнуто одного з контрольних порогів.

Вузол вимірювання напруги призначений для узгодження напруги акумуляторної системи з допустимим діапазоном аналогового входу мікроконтролера. Наприклад, якщо використовується акумуляторна система з номінальною напругою 12 В, її напруга не може подаватися безпосередньо на вхід ESP32. Тому застосовується дільник напруги, який зменшує вхідну напругу до безпечного рівня. Після цього мікроконтролер за допомогою аналого-цифрового перетворювача зчитує значення та виконує програмний перерахунок.

Мікроконтролерний блок виконує основні функції оброблення інформації. До його задач належать ініціалізація Wi-Fi-з'єднання, підключення до Telegram Bot API, зчитування значень з аналогового входу, обчислення рівня заряду, оброблення команд користувача, керування релейним модулем і формування повідомлень про події. У програмному забезпеченні ESP32 реалізуються як періодичні операції контролю, так і реакції на команди, що надходять від користувача через Telegram-бота.

Блок мережевої взаємодії побудований на основі вбудованого Wi-Fi-модуля ESP32. Після увімкнення пристрій підключається до заданої Wi-Fi-мережі, отримує доступ до Інтернету та починає обмін даними з Telegram-ботом. Передавання повідомлень і отримання команд здійснюється через Telegram Bot API. Такий підхід дозволяє уникнути використання окремої серверної частини, оскільки роль каналу взаємодії між користувачем і пристроєм виконує сама месенджер-платформа.

Telegram-бот у загальній структурі системи виконує функції користувацького інтерфейсу. Через нього користувач отримує інформацію про стан акумуляторної системи, запуск або зупинку генератора, зміну режиму автозапуску та інші важливі події. Крім того, через Telegram-бота користувач надсилає команди керування. Основними командами є /status, /battery, /start_generator, /stop_generator, /autostart_generator_on, /autostart_generator_off і /help.

Виконавчий блок системи реалізується за допомогою релейного модуля. Його призначення полягає у комутації керуючого кола генератора. У практичній реалізації реле може імітувати натискання кнопки запуску або подавати сигнал на вхід дистанційного керування генератором. У макетному варіанті замість реального генератора може використовуватися світлодіодна індикація або інший простий виконавчий елемент, що відображає стан реле. Це дозволяє перевірити логіку роботи системи без підключення до силового обладнання.

Генератор у структурі програмно-технічного засобу є об'єктом керування. Його запуск може виконуватися вручну за командою користувача або автоматично при досягненні критичного рівня заряду акумуляторної системи. Для безпечної роботи система повинна повідомляти користувача про факт запуску, вимкнення або аварійного припинення роботи генератора. У межах мінімальної реалізації повідомлення про несправність або відсутність палива може формуватися без деталізованої діагностики, тобто як загальна подія про припинення роботи генератора.

Блок живлення забезпечує стабільну напругу для роботи ESP32 та периферійних модулів. Оскільки акумуляторна система може мати напругу, вищу за допустиму для мікроконтролера, необхідно використовувати DC-DC перетворювач або стабілізований блок живлення. Найпростішим рішенням є використання понижувального перетворювача, який формує напругу 5 В для живлення плати ESP32. Стабільність живлення є важливою умовою правильної роботи Wi-Fi-модуля, аналого-цифрового перетворювача та релейного модуля.

Інформаційні потоки в системі можна поділити на вимірювальні, керуючі та повідомлювальні. Вимірювальний потік формується від акумуляторної системи через вузол вимірювання напруги до ESP32. Керуючий потік формується від користувача через Telegram-бота до мікроконтролера, а далі - до релейного модуля і генератора. Повідомлювальний потік формується у зворотному напрямку: від ESP32 через Telegram-бота до користувача. Саме наявність цих трьох потоків забезпечує повний цикл роботи засобу: контроль, прийняття рішення, дія та інформування.

Загальна логіка роботи системи передбачає, що після подавання живлення ESP32 виконує початкову ініціалізацію, підключається до Wi-Fi-мережі та переходить у режим очікування команд і періодичного контролю напруги акумуляторної системи. Якщо користувач надсилає команду, мікроконтролер перевіряє її зміст і виконує відповідну дію. Якщо під час періодичного контролю виявлено перехід через один із порогів заряду, система формує повідомлення. Якщо рівень заряду досяг 30 % і режим автозапуску активований, ESP32 подає сигнал на релейний модуль для запуску генератора.

Для підвищення зрозумілості роботи системи доцільно розглядати її як сукупність двох взаємопов'язаних контурів. Перший контур є контуром моніторингу: акумуляторна система - вузол вимірювання - ESP32 - Telegram-бот - користувач. Другий контур є контуром керування: користувач або алгоритм автозапуску - ESP32 - релейний модуль - генератор. Завдяки такому поділу

можна окремо аналізувати функції контролю та функції керування, що спрощує подальше проектування.

Особливістю запропонованої структури є відсутність окремого сервера, бази даних або мобільного застосунку. Це зменшує складність системи та робить її придатною для навчальної і макетної реалізації. Історія подій зберігається у чаті Telegram, а всі основні операції виконуються безпосередньо мікроконтролером ESP32. Така архітектура є достатньою для поставленої задачі, оскільки система не потребує складної аналітики, а має забезпечити оперативне інформування та базове керування генератором.

Разом із тим структура системи має певні обмеження. Її працездатність залежить від наявності Wi-Fi-мережі та доступу до Інтернету. У разі втрати з'єднання пристрій може продовжувати виконувати локальну логіку контролю та автозапуску, але не зможе надсилати повідомлення користувачу або приймати команди Telegram-бота. Тому для практичного використання бажано, щоб мережеве обладнання також було підключене до резервного живлення.

Таким чином, загальна структура програмно-технічного засобу поєднує апаратні компоненти контролю й керування з програмною взаємодією через Telegram-бота. Центральним елементом системи є ESP32, який забезпечує оброблення даних, прийняття рішень, мережеву взаємодію та керування виконавчим елементом. Запропонована структура є достатньо простою для практичної реалізації, але водночас забезпечує виконання основних функцій віддаленого контролю енергозабезпечення об'єкта. Наступним етапом є детальніший аналіз апаратної та програмної частин засобу, необхідних для реалізації визначеної структури.

					КВРКІ 022075.22.04.26	Арк. 20
Зм.	Арк.	№ докum.	Підпис	Дата		

1.4 Аналіз апаратної та програмної частин системи

Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів складається з апаратної та програмної частин, які функціонують як єдина система. Апаратна частина забезпечує фізичне вимірювання параметрів акумуляторної системи, живлення електронних компонентів і керування виконавчим елементом. Програмна частина відповідає за оброблення вимірних даних, реалізацію алгоритмів прийняття рішень, взаємодію з Telegram-ботом і формування повідомлень користувачу.

Апаратна частина розроблюваного засобу повинна бути максимально простою, оскільки робота орієнтована на створення помірно складного макета, придатного для практичної перевірки. До її складу входять мікроконтролер ESP32, вузол вимірювання напруги акумуляторної системи, релейний модуль для керування генератором, стабілізоване джерело живлення, акумуляторна система та генератор із можливістю дистанційного запуску. Такий склад є мінімально достатнім для реалізації основних функцій контролю й керування.

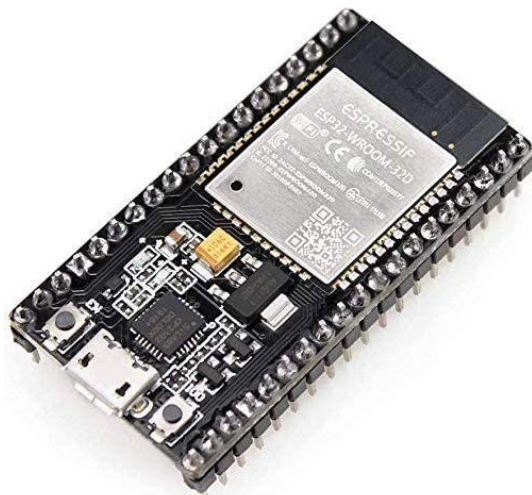


Рисунок 1.1 - Мікроконтролер ESP32 як центральний елемент програмно-технічного засобу [46]

Центральним апаратним компонентом є мікроконтролер ESP32. Його використання є доцільним через наявність вбудованого Wi-Fi-модуля, достатньої кількості цифрових входів/виходів, підтримку аналогових входів і можливість програмування у середовищі Arduino IDE або PlatformIO. У межах даної системи ESP32 виконує зчитування сигналу з вузла вимірювання напруги, обробляє команди Telegram-бота, керує релейним модулем і формує повідомлення про стан системи.

Вузол вимірювання напруги акумуляторної системи призначений для отримання інформації про стан заряду акумуляторів. Оскільки напруга акумуляторної системи може перевищувати допустимий рівень для аналогового входу ESP32, безпосереднє підключення акумулятора до мікроконтролера є неприпустимим. Для узгодження рівнів може використовуватися резистивний дільник напруги або готовий модуль вимірювання напруги. У найпростішому випадку застосовується дільник, який зменшує напругу до безпечного діапазону для аналого-цифрового перетворювача мікроконтролера.

Програмне забезпечення після зчитування значення з аналогового входу виконує перерахунок вимірюваної напруги у відсотковий рівень заряду. Такий спосіб має наближений характер, оскільки фактичний заряд акумулятора залежить не лише від напруги, а й від типу акумулятора, струму навантаження, температури та технічного стану батареї. Проте для задачі базового контролю цього підходу достатньо, оскільки система повинна не виконувати високоточну діагностику акумулятора, а визначати момент досягнення контрольних рівнів 70 %, 50 % і 30 %.

Релейний модуль використовується як виконавчий елемент. Його функція полягає у комутації кола запуску або зупинки генератора. У макетному варіанті релейний модуль може керувати світлодіодом або іншим індикатором, який імітує стан генератора. У практичному застосуванні контакти реле можуть бути підключені до входу дистанційного запуску генератора, якщо така можливість

передбачена його конструкцією. Це дозволяє не втручатися у силову частину генератора, а використовувати лише слабкострумовий керуючий сигнал.

Для живлення ESP32 і периферійних модулів необхідно передбачити стабілізоване джерело живлення. Якщо система живиться від акумуляторної батареї з напругою 12 В, доцільно використати понижувальний DC-DC перетворювач, який формує напругу 5 В для плати ESP32 або 3,3 В для окремих електронних компонентів. Стабільність живлення має важливе значення, оскільки нестабільна напруга може призвести до перезавантаження мікроконтролера, втрати Wi-Fi-з'єднання або некоректної роботи релейного модуля.

Програмна частина системи реалізується у вигляді вбудованого програмного забезпечення для ESP32. Вона повинна забезпечувати початкову ініціалізацію пристрою, підключення до Wi-Fi-мережі, зв'язок із Telegram-ботом, періодичне вимірювання напруги акумуляторної системи, оброблення команд користувача, керування генератором і формування повідомлень про події. Програмне забезпечення виконує роль логічного центру системи, оскільки саме в ньому реалізуються правила прийняття рішень.

Після подавання живлення програма виконує ініціалізацію основних модулів. На цьому етапі налаштовуються виводи мікроконтролера, встановлюється початковий стан релейного модуля, запускається послідовний інтерфейс для налагодження, виконується підключення до Wi-Fi-мережі та ініціалізується взаємодія з Telegram-ботом. Після успішного підключення система переходить у робочий режим.

У робочому режимі програмне забезпечення виконує два основні цикли: цикл контролю параметрів акумуляторної системи та цикл оброблення команд Telegram-бота. У циклі контролю система періодично зчитує значення напруги, обчислює орієнтовний рівень заряду та перевіряє досягнення порогових значень. У циклі оброблення команд програма перевіряє, чи надійшли нові повідомлення від користувача, розпізнає команду та виконує відповідну дію.

Основними програмними модулями доцільно вважати: модуль підключення до Wi-Fi, модуль взаємодії з Telegram-ботом, модуль вимірювання напруги акумуляторної системи, модуль визначення рівня заряду, модуль контролю порогових значень, модуль ручного керування генератором, модуль автоматичного запуску генератора, модуль формування повідомлень користувачу.

Модуль підключення до Wi-Fi забезпечує з'єднання ESP32 з локальною мережею. Для цього у програмі задаються назва Wi-Fi-мережі та пароль доступу. Після встановлення з'єднання мікроконтролер отримує можливість звертатися до Telegram Bot API. У разі втрати з'єднання програмне забезпечення повинно передбачати повторну спробу підключення, оскільки від стабільності мережі залежить можливість надсилання повідомлень користувачу.

Модуль взаємодії з Telegram-ботом забезпечує приймання команд і надсилання відповідей. У програмі використовується токен бота, отриманий під час його створення, а також ідентифікатор чату користувача. Використання ідентифікатора чату дозволяє обмежити доступ до керування системою, щоб команди могли виконуватися лише від дозволеного користувача. Це є важливим елементом базової безпеки, оскільки генератор є виконавчим пристроєм і не повинен керуватися сторонніми особами.

Модуль вимірювання напруги акумуляторної системи зчитує значення з аналогового входу ESP32. Отримане цифрове значення перераховується у фактичну напругу з урахуванням коефіцієнта дільника напруги. Далі напруга перетворюється в умовний відсоток заряду. Для спрощеної реалізації можна задати мінімальну й максимальну напругу акумуляторної системи та виконувати лінійне масштабування між цими значеннями. Наприклад, для 12-вольтової системи можна умовно прийняти, що певне нижнє значення відповідає 0 %, а верхнє - 100 %. Точні значення повинні уточнюватися залежно від типу акумулятора.

					КВРКІ 022075.22.04.26	Арк. 24
Зм.	Арк.	№ докum.	Підпис	Дата		

Модуль контролю порогових значень порівнює розрахований рівень заряду з установленими межами 70 %, 50 % і 30 %. При першому переході через відповідний поріг система формує повідомлення користувачу. Для запобігання багаторазовому надсиланню однакових повідомлень використовуються програмні прапорці, які зберігають факт надсилання попередження. Наприклад, якщо повідомлення про рівень 70 % вже було надіслано, воно не повторюється в кожному наступному циклі вимірювання.

Модуль ручного керування генератором обробляє команди /start_generator і /stop_generator. При отриманні команди запуску програма активує релейний модуль і надсилає користувачу повідомлення про запуск генератора. При отриманні команди зупинки програма вимикає реле або подає сигнал зупинки, після чого повідомляє користувача про виконану дію. У макетній реалізації стан генератора може визначатися за станом релейного виходу.

Модуль автоматичного запуску генератора працює лише тоді, коли користувач активував відповідний режим командою /autostart_generator_on. Якщо автозапуск увімкнено і рівень заряду акумуляторної системи досягає 30 %, програма автоматично активує релейний модуль для запуску генератора. Якщо режим автозапуску вимкнено, система лише надсилає повідомлення про критичний заряд, але не виконує керуючу дію. Такий підхід дає користувачу можливість самостійно обирати рівень автоматизації.

Окреме місце у програмній частині займає формування повідомлень. Повідомлення повинні бути короткими, однозначними та містити тільки необхідну інформацію. Наприклад, повідомлення про заряд 50 % має містити поточний рівень заряду та рекомендацію перевірити стан системи. Повідомлення про критичний рівень 30 % повинно вказувати на потребу запуску генератора або повідомляти про автоматичний запуск, якщо відповідний режим активовано.

Крім основної програмної реалізації на ESP32, для формалізації логіки керування може бути використана спрощена апаратна модель у середовищі Intel Quartus Prime. Така модель не є окремим обов'язковим апаратним компонентом

готової системи, а застосовується як допоміжний засіб проектування. Вона дозволяє подати логіку керування генератором у вигляді цифрового автомата, який реагує на рівень заряду, команди користувача, стан автозапуску та ознаку аварійного вимкнення генератора.

У VHDL-моделі доцільно передбачити вхідні сигнали, які відповідають рівню заряду акумуляторної системи, командам запуску й зупинки генератора, командам увімкнення та вимкнення автозапуску, а також сигналу несправності генератора. Вихідними сигналами такої моделі можуть бути сигнал керування релейним модулем, стан режиму автозапуску та події для формування повідомлень. Це дозволяє перевірити правильність логіки до її програмної реалізації на ESP32.

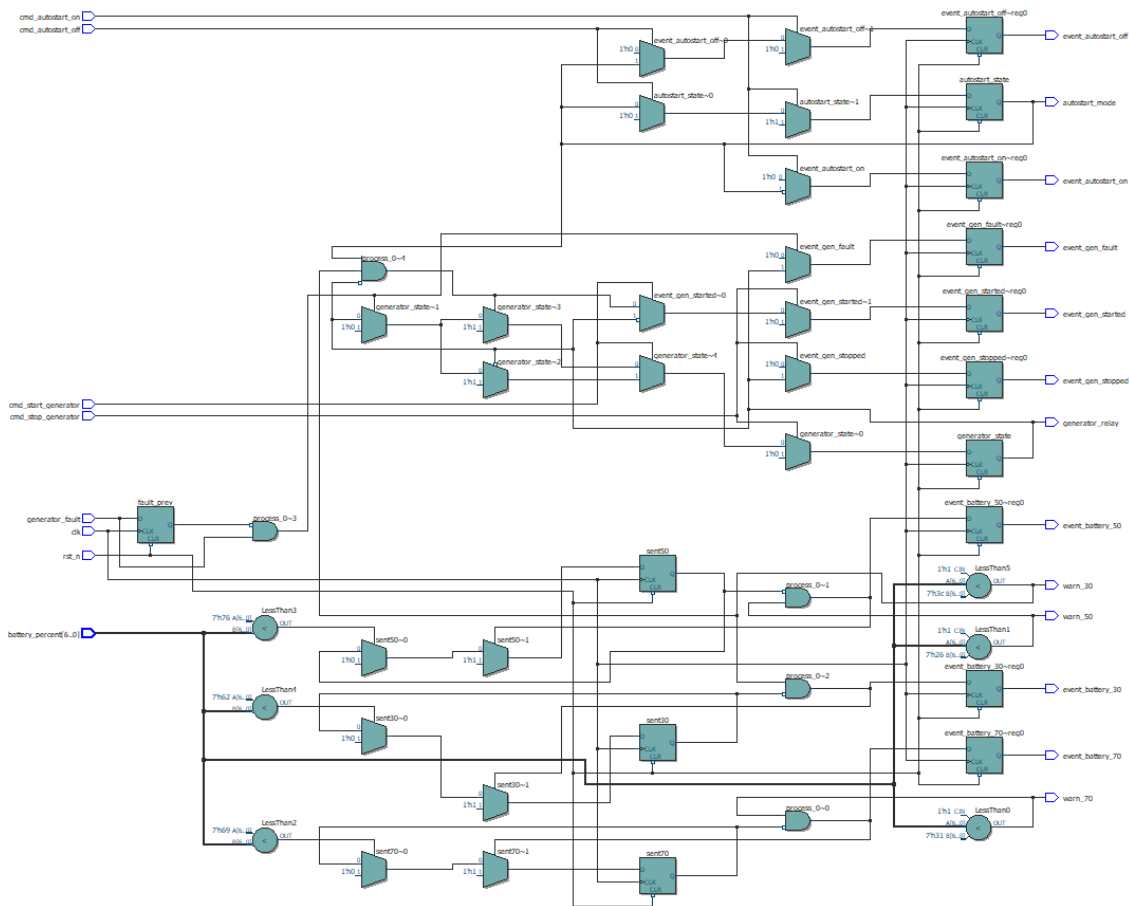


Рисунок 1.2 - Апаратна модель логіки керування генератором у середовищі Quartus

Використання Quartus у межах даної роботи доцільно обмежити саме моделюванням логіки прийняття рішень. Основна практична реалізація залишається на базі ESP32, оскільки саме цей мікроконтролер забезпечує Wi-Fi-зв'язок, роботу з Telegram-ботом і керування реальними зовнішніми модулями. Такий підхід не ускладнює систему, але дозволяє продемонструвати інженерний підхід до формалізації керуючої логіки.

Взаємодія апаратної та програмної частин відбувається через вхідні й вихідні сигнали мікроконтролера. Апаратна частина формує вимірювальні сигнали та виконує фізичні дії, а програмна частина визначає, коли саме ці дії мають бути виконані. Наприклад, вузол вимірювання напруги лише подає аналоговий сигнал на ESP32, але рішення про надсилання повідомлення або запуск генератора приймається програмним алгоритмом. Аналогічно релейний модуль лише виконує комутацію, тоді як умови його ввімкнення визначаються програмою.

Такий розподіл функцій є доцільним для розроблюваного засобу, оскільки дає змогу залишити апаратну частину простою, а основну логіку реалізувати програмно. У разі потреби алгоритм можна змінити без перероблення електричної схеми: наприклад, змінити пороги повідомлень, додати нову команду Telegram-бота або змінити умови автоматичного запуску генератора. Це підвищує гнучкість системи та спрощує її подальше вдосконалення.

Таким чином, апаратна частина програмно-технічного засобу забезпечує фізичний контроль параметрів і виконання керуючих дій, а програмна частина реалізує алгоритми оброблення даних, взаємодію з користувачем і логіку керування генератором. Поєднання ESP32, вузла вимірювання напруги, релейного модуля та Telegram-бота дозволяє створити просту, доступну й функціонально достатню систему віддаленого контролю енергозабезпечення. Проведений аналіз апаратної та програмної частин є основою для формування висновків до першого розділу та переходу до етапу проектування засобу.

					КВРКІ 022075.22.04.26	Арк. 27
Зм.	Арк.	№ докum.	Підпис	Дата		

1.5 Висновки до першого розділу

У першому розділі було розглянуто предметну область віддаленого контролю енергозабезпечення об'єктів та визначено основні передумови створення програмно-технічного засобу на базі мікроконтролера ESP32 і месенджер-платформи Telegram. Встановлено, що для невеликих об'єктів важливим є не лише наявність резервного джерела живлення, а й можливість своєчасного отримання інформації про стан акумуляторної системи, запуск або вимкнення генератора та зміну режимів роботи системи.

Проаналізовано особливості побудови систем резервного енергозабезпечення, до складу яких можуть входити акумуляторна система, генератор, вузол вимірювання напруги, виконавчий елемент і засоби зв'язку з користувачем. Обґрунтовано доцільність використання спрощеної структури, яка не потребує GSM-модуля, SIM-карти, окремої серверної частини або спеціального мобільного застосунку. Зв'язок із користувачем у запропонованому засобі доцільно реалізувати через Wi-Fi-мережу та Telegram-бота.

Визначено призначення програмно-технічного засобу, яке полягає у контролі рівня заряду акумуляторної системи, інформуванні користувача про важливі події та керуванні генератором у ручному або автоматичному режимі. До основних функцій засобу віднесено перегляд поточного стану системи, перевірку рівня заряду акумуляторів, надсилання повідомлень при досягненні порогів 70 %, 50 % і 30 %, ручний запуск і вимкнення генератора, а також увімкнення й вимкнення режиму автоматичного запуску.

Розглянуто загальну структуру програмно-технічного засобу, яка включає акумуляторну систему, вузол вимірювання напруги, мікроконтролер ESP32, Wi-Fi-мережу, Telegram-бота, релейний модуль, генератор і стабілізоване джерело живлення. Центральним елементом системи визначено ESP32, який забезпечує зчитування вимірювальних даних, оброблення команд користувача, формування повідомлень і керування виконавчим елементом.

Проведено аналіз апаратної та програмної частин системи. Апаратна частина забезпечує фізичне вимірювання параметрів і виконання керуючих дій, тоді як програмна частина реалізує логіку контролю, оброблення команд Telegram-бота, формування повідомлень і алгоритм автоматичного запуску генератора. Okремо зазначено можливість використання середовища Intel Quartus Prime для побудови спрощеної апаратної моделі логіки керування, яка дозволяє формалізувати роботу вузла прийняття рішень.

Таким чином, у першому розділі сформовано загальне уявлення про призначення, функціональні можливості та структуру програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів. Отримані результати є основою для подальшого проєктування системи, вибору конкретних апаратних компонентів, розроблення структурної схеми, алгоритмів роботи Telegram-бота та логіки автоматичного запуску генератора.

					КВРКІ 022075.22.04.26	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ СИСТЕМИ ДИСТАНЦІЙНОГО КОНТРОЛЮ ЕНЕРГОЗАБЕЗПЕЧЕННЯ

2.1 Визначення вимог до програмно-технічного засобу

Проєктування програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів доцільно розпочати з визначення вимог до його функціонування. На цьому етапі необхідно встановити, які задачі повинна виконувати система, які апаратні та програмні обмеження мають бути враховані, а також які умови є достатніми для підтвердження працездатності розробленого засобу. Вимоги формуються на основі призначення системи, визначеного у першому розділі, та з урахуванням необхідності збереження помірної складності практичної реалізації.

Розроблюваний засіб призначений для контролю стану резервного енергозабезпечення невеликого об'єкта. Основним об'єктом контролю є акумуляторна система, а основним об'єктом керування – генератор з можливістю дистанційного запуску. Система повинна забезпечувати взаємодію з користувачем через Telegram-бота, що дозволяє отримувати повідомлення про стан енергозабезпечення та надсилати команди керування без використання окремого мобільного застосунку або серверної частини.

До основних функціональних вимог належить контроль рівня заряду акумуляторної системи. Для цього засіб повинен періодично вимірювати напругу акумулятора, перетворювати отримане значення в орієнтовний рівень заряду та порівнювати його із заданими порогоми. У межах роботи прийнято три основні пороги інформування: 70 %, 50 % і 30 %. Досягнення кожного з цих рівнів повинно супроводжуватися повідомленням користувачу через Telegram-бота.

Поріг 70 % використовується як початкове попередження про зниження заряду акумуляторної системи. Поріг 50 % свідчить про необхідність підвищеної уваги до стану резервного живлення. Поріг 30 % приймається як критичний

					КвРКІ 022075.22.04.26	Арк. 30
Зм.	Арк.	№ докum.	Підпис	Дата		

рівень, при якому система повинна повідомити користувача про небезпеку подальшого розряду. Якщо при цьому активовано режим автоматичного запуску, програмно-технічний засіб повинен подати команду на запуск генератора.

Важливою вимогою до алгоритму повідомлень є недопущення їх постійного дублювання. Якщо рівень заряду знизився до певного порога, повідомлення повинно бути надіслане один раз після першого переходу через цей поріг. Повторне повідомлення доцільно формувати лише після зміни стану системи або після нового циклу заряджання й розряджання акумуляторної системи. Це підвищує зручність використання Telegram-бота та запобігає надмірній кількості повідомлень.

Програмно-технічний засіб повинен підтримувати ручне керування генератором. Користувач має мати можливість надіслати команду запуску генератора через Telegram-бота. Після отримання такої команди мікроконтролер ESP32 повинен активувати релейний модуль або інший виконавчий елемент, який відповідає за подавання сигналу запуску. Після виконання команди система повинна надіслати користувачу підтвердження про запуск генератора.

Також повинна бути передбачена можливість ручного вимкнення генератора. При надходженні відповідної команди від користувача ESP32 повинен змінити стан виконавчого елемента та сформував повідомлення про вимкнення генератора. У макетній реалізації ця дія може перевірятися за станом релейного модуля або світлодіодної індикації, що дозволяє протестувати логіку роботи без підключення до реального генератора.

Окремою функціональною вимогою є підтримка режиму автоматичного запуску генератора. Користувач повинен мати можливість увімкнути або вимкнути цей режим через Telegram-бота. Якщо автоматичний запуск активований, система повинна самостійно запустити генератор при досягненні критичного рівня заряду 30 %. Якщо режим автозапуску вимкнений, система повинна лише повідомити користувача про критичний стан акумуляторів, але не виконувати запуск без прямої команди.

Командний інтерфейс Telegram-бота повинен бути простим і достатнім для виконання основних дій. У системі передбачаються такі команди: /status, /battery, /start_generator, /stop_generator, /autostart_generator_on, /autostart_generator_off, /help. Команда /status повинна повертати загальний стан системи, команда /battery - поточний рівень заряду акумуляторів, команди запуску й зупинки - керувати генератором, а команди автозапуску - змінювати режим автоматичного керування.

До повідомлень, які повинна формувати система, належать повідомлення про зниження рівня заряду, запуск генератора, вимкнення генератора, увімкнення або вимкнення автозапуску, а також аварійне припинення роботи генератора. У межах мінімальної реалізації повідомлення про аварійне вимкнення генератора може трактуватися як повідомлення про несправність або відсутність палива без деталізованої діагностики. Це обмеження є прийнятним, оскільки система не передбачає використання окремих датчиків рівня палива, обертів двигуна або технічного стану генератора.

До функціональних вимог також належить збереження історії подій. У даній роботі не передбачається створення окремої бази даних. Історія подій зберігається у вигляді повідомлень у чаті Telegram. Такий підхід спрощує програмну архітектуру та є достатнім для макетної реалізації, оскільки користувач може переглянути попередні повідомлення без додаткових засобів зберігання інформації.

Крім функціональних вимог, необхідно визначити нефункціональні вимоги до системи. Насамперед засіб повинен мати просту апаратну структуру. Кількість компонентів має бути мінімальною, але достатньою для реалізації контролю заряду, зв'язку з користувачем і керування генератором. До обов'язкових компонентів належать ESP32, вузол вимірювання напруги, релейний модуль, DC-DC перетворювач або стабілізоване джерело живлення, акумуляторна система та генератор з можливістю дистанційного запуску.

Система повинна працювати через Wi-Fi-з'єднання. Використання GSM-модуля, SIM-карти, окремого сервера або мобільного застосунку не передбачається. Це обмеження дозволяє зменшити складність апаратної частини та зосередитися на реалізації основної логіки керування. Водночас необхідно враховувати, що працездатність віддаленого інтерфейсу залежить від доступності Wi-Fi-мережі та підключення до Інтернету.

Вимоги до живлення системи передбачають використання стабілізованої напруги для ESP32 та допоміжних модулів. Якщо живлення здійснюється від акумуляторної системи, необхідно застосувати понижувальний DC-DC перетворювач. Напруга живлення повинна бути достатньо стабільною, оскільки нестабільність може спричинити перезавантаження мікроконтролера, втрату Wi-Fi-з'єднання або некоректну роботу виконавчого елемента.

До вимог безпеки належить обмеження доступу до керування системою. Telegram-бот повинен обробляти команди лише від дозволеного користувача або заданого ідентифікатора чату. Це необхідно для запобігання несанкціонованому запуску або вимкненню генератора. Для бакалаврської роботи достатньо реалізувати базову перевірку chat_id, яка дозволить ігнорувати команди від сторонніх користувачів.

Вимоги до надійності системи передбачають коректну роботу основних функцій у разі перезапуску мікроконтролера, короткочасної втрати Wi-Fi-з'єднання або повторного надсилання команд. Після відновлення роботи ESP32 повинен повторно підключитися до Wi-Fi-мережі та продовжити оброблення команд. Для режиму автозапуску доцільно передбачити збереження його стану в енергонезалежній пам'яті або, у спрощеному варіанті, встановлення початкового стану після запуску системи.

Вимоги до точності вимірювання в межах цієї роботи мають прикладний характер. Система не повинна виконувати високоточне вимірювання ємності акумулятора. Її завдання полягає у визначенні наближеного рівня заряду та фіксації досягнення заданих порогів. Тому допустимим є використання

спрощеного методу оцінювання заряду за напругою акумуляторної системи. У тексті роботи необхідно зазначити, що такий метод має обмежену точність, але є достатнім для базового контролю.

Вхідними даними для програмно-технічного засобу є значення напруги акумуляторної системи, команди користувача з Telegram-бота, стан режиму автозапуску та, за наявності відповідного входу, сигнал про стан генератора або основного живлення. Вихідними даними є повідомлення користувачу, стан релейного модуля, ознака роботи генератора та відповіді на команди Telegram-бота.

Для перевірки відповідності системи вимогам необхідно передбачити тестування основних режимів роботи. Зокрема, потрібно перевірити отримання стану системи за командою /status, правильність визначення рівня заряду, формування повідомлень при досягненні порогів 70 %, 50 % і 30 %, ручний запуск і вимкнення генератора, увімкнення та вимкнення автозапуску, а також автоматичний запуск генератора при критичному рівні заряду.

У межах роботи допускається використання макетної реалізації. Замість реального генератора може бути використаний релейний модуль із світлодіодною індикацією або інший простий виконавчий елемент, який підтверджує виконання команди. Зміна рівня заряду акумулятора може імітуватися за допомогою лабораторного джерела живлення або змінного сигналу, що подається на вхід вимірювального вузла. Такий підхід дозволяє безпечно перевірити алгоритми системи без підключення до силового обладнання.

Сформовані вимоги визначають основу для подальшого проектування програмно-технічного засобу. Вони задають необхідний набір функцій, обмеження апаратної реалізації, логіку взаємодії з користувачем і критерії перевірки працездатності. Наступним етапом є вибір апаратних компонентів, які забезпечать реалізацію визначених вимог з мінімальною складністю та достатньою функціональністю.

2.2 Вибір компонентів апаратної частини

Вибір компонентів апаратної частини програмно-технічного засобу виконується з урахуванням вимог, визначених у підрозділі 2.1. Оскільки розроблювана система має бути простою, доступною та придатною для макетної реалізації, до її складу включено лише ті елементи, які необхідні для контролю рівня заряду акумуляторної системи, передавання інформації користувачу через Telegram-бота та керування генератором.

Для практичної реалізації програмно-технічного засобу обрано такий склад апаратних компонентів:

- 1) - мікроконтролерна плата ESP32 DevKit V1 на базі модуля ESP-WROOM-32;
- 2) - резистивний дільник напруги на резисторах $R1 = 100 \text{ кОм}$ і $R2 = 27 \text{ кОм}$;
- 3) - конденсатор 100 нФ для згладжування сигналу на аналоговому вході;
- 4) - захисний стабілітрон BZX55C3V3 на $3,3 \text{ В}$ для обмеження напруги на вході ESP32;
- 5) - одноканальний релейний модуль 5 В на реле SRD-05VDC-SL-C з транзисторним керуванням;
- 6) - понижувальний DC-DC перетворювач LM2596S з вихідною напругою 5 В ;
- 7) - акумуляторна система 12 В ;
- 8) - генератор із входом дистанційного запуску;
- 9) - Wi-Fi-маршрутизатор з доступом до Інтернету.

Центральним компонентом системи є плата ESP32 DevKit V1. Вона виконує функції керуючого модуля, який зчитує сигнал з вузла вимірювання напруги, обробляє команди Telegram-бота, формує повідомлення користувачу та керує релейним модулем. Вибір ESP32 є доцільним, оскільки ця плата має вбудований Wi-Fi-модуль, достатню кількість цифрових входів і виходів,

аналогові входи та підтримується середовищами Arduino IDE і PlatformIO. Завдяки цьому ESP32 дозволяє реалізувати систему без додаткового комунікаційного модуля.

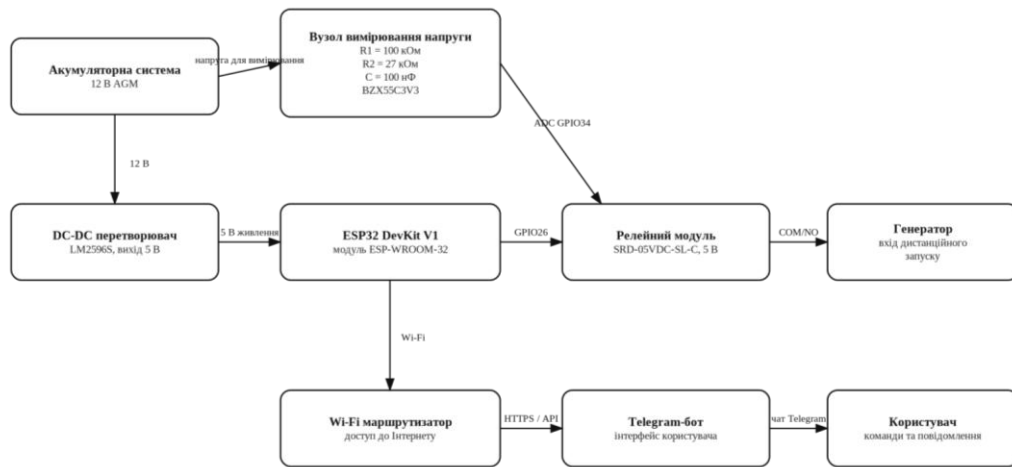


Рисунок 2.1 – Схема вибраних компонентів апаратної частини

Для вимірювання напруги акумуляторної системи використовується резистивний дільник напруги на резисторах 100 кОм і 27 кОм. Така схема зменшує напругу акумулятора до рівня, допустимого для аналогового входу ESP32. При нарузі акумуляторної системи приблизно 14,4 В, що відповідає зарядженому 12-вольтовому акумулятору, на виході дільника формується напруга близько 3,06 В. Це значення не перевищує допустимий рівень для входу мікроконтролера.

До виходу дільника напруги підключається конденсатор 100 нФ, який зменшує вплив короткочасних перешкод і стабілізує вимірювальний сигнал. Для додаткового захисту аналогового входу ESP32 використовується стабілітрон BZX55C3V3 на 3,3 В. Його призначення полягає в обмеженні напруги на вході мікроконтролера у випадку випадкового перевищення допустимого рівня.



Рисунок 2.2 - Зображення стабілітрона BZX55C3V3[46]

Вимірювання рівня заряду акумуляторної системи у цій роботі виконується наближеним способом — за значенням напруги. Програмне забезпечення ESP32 перераховує виміряне значення у відсотковий рівень заряду. Такий метод не є високоточним, однак він достатній для поставленої задачі, оскільки система повинна визначати не фактичну ємність акумулятора, а момент досягнення контрольних рівнів 70 %, 50 % і 30 %.

Для керування генератором обрано одноканальний релейний модуль 5 В на реле SRD-05VDC-SL-C. Релейний модуль виконує функцію виконавчого елемента, який комутує коло дистанційного запуску генератора. Керуючий сигнал на модуль подається з цифрового виходу ESP32 через вхід транзисторного каскаду релейного модуля. Контакти реле використовуються для замикання керуючого входу генератора.

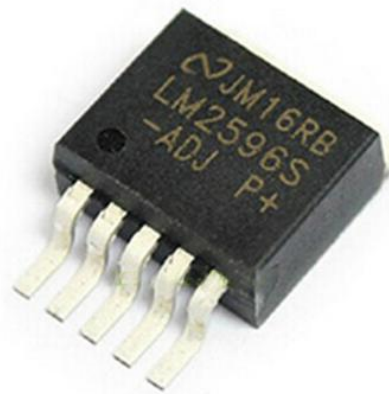


Рисунок 2.4 - Понижувальний DC-DC перетворювач LM2596S[46]

Об'єктом контролю у системі є акумуляторна система 12 В. Для макетної перевірки може використовуватися свинцево-кислотний AGM-акумулятор 12 В. У процесі тестування також допускається застосування лабораторного джерела постійної напруги, яке імітує зміну напруги акумуляторної системи. Це дозволяє безпечно перевірити спрацювання порогів 70 %, 50 % і 30 % без тривалого реального розрядження акумулятора.

Генератор у складі системи розглядається як резервне джерело енергії з входом дистанційного запуску. У практичній схемі контакти релейного модуля підключаються до керуючого входу запуску генератора. У макетній реалізації робота генератора може імітуватися світлодіодним індикатором, підключеним до релейного модуля. Такий варіант дозволяє перевірити логіку запуску, зупинки та автоматичного керування без підключення до силового обладнання.

Зв'язок із користувачем забезпечується через Wi-Fi-мережу. Для цього використовується вбудований Wi-Fi-модуль ESP32. Після підключення до маршрутизатора мікроконтролер отримує доступ до Telegram Bot API, приймає

команди користувача та надсилає повідомлення про стан системи. Додаткові модулі зв'язку в апаратній частині не застосовуються.

Прийнятий склад компонентів дозволяє реалізувати всі основні функції програмно-технічного засобу. ESP32 забезпечує оброблення даних і зв'язок із користувачем, дільник напруги дає змогу контролювати стан акумуляторної системи, релейний модуль виконує керування генератором, а DC-DC перетворювач забезпечує стабільне живлення електронної частини. Такий підхід відповідає вимозі мінімальної апаратної складності та є достатнім для виконання практичної частини кваліфікаційної роботи.

Таким чином, для апаратної реалізації програмно-технічного засобу обрано конкретний набір компонентів: ESP32 DevKit V1, дільник напруги на резисторах 100 кОм і 27 кОм, конденсатор 100 нФ, стабілітрон BZX55C3V3, релейний модуль SRD-05VDC-SL-C, DC-DC перетворювач LM2596S, акумуляторну систему 12 В і генератор із входом дистанційного запуску. Обрані компоненти забезпечують контроль заряду, передавання повідомлень через Telegram-бота та керування генератором. Наступним етапом є розроблення структурної схеми програмно-технічного засобу на основі вибраних апаратних елементів.

2.3 Розроблення структурної схеми засобу

Після вибору апаратних компонентів необхідно визначити їх взаємне розміщення у структурі програмно-технічного засобу та описати напрямки передавання вимірювальних, керуючих і інформаційних сигналів. Структурна схема є основою для подальшого розроблення електричної схеми підключення, алгоритмів роботи та програмного забезпечення мікроконтролера ESP32.

Розроблюваний програмно-технічний засіб має централізовану структуру. Центральним елементом є мікроконтролерна плата ESP32 DevKit V1, до якої підключаються вузол вимірювання напруги акумуляторної системи, релейний

модуль керування генератором і джерело стабілізованого живлення. Передавання даних користувачу здійснюється через Wi-Fi-мережу та Telegram-бота.

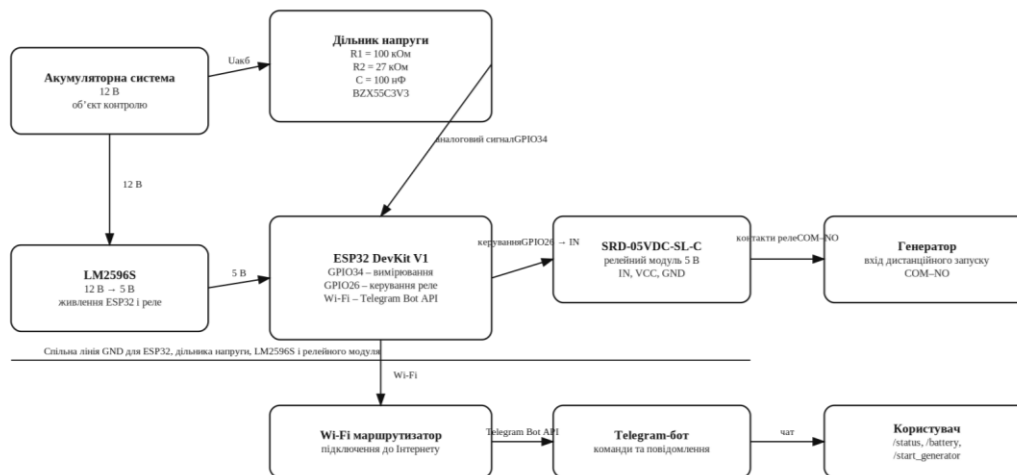


Рисунок 2.5 - Структурна схема програмно-технічного засобу віддаленого контролю енергозабезпечення

У структурній схемі виділено такі основні блоки:

- акумуляторна система 12 В;
- понижувальний DC-DC перетворювач LM2596S;
- вузол вимірювання напруги на резисторах 100 кОм і 27 кОм;
- мікроконтролерна плата ESP32 DevKit V1;
- релейний модуль SRD-05VDC-SL-C;
- генератор із входом дистанційного запуску;
- Wi-Fi-маршрутизатор;
- Telegram-бот;
- користувач.

Акумуляторна система 12 В у запропонованій структурі виконує роль об'єкта контролю та джерела живлення для електронної частини. Її напруга надходить на два функціональні напрямки. Перший напрямок — живлення

понижувального перетворювача LM2596S. Другий напрямок — подавання напруги на резистивний дільник для подальшого вимірювання мікроконтролером.

Блок Понижувальний DC-DC перетворювач LM2596S підключається до акумуляторної системи 12 В і формує стабілізовану напругу 5 В. Вихід 5 В використовується для живлення плати ESP32 DevKit V1 через контакт VIN та для живлення одноканального релейного модуля SRD-05VDC-SL-C. Загальний провід акумуляторної системи, перетворювача, ESP32, дільника напруги та релейного модуля об'єднується в одну спільну лінію GND.

Вузол вимірювання напруги реалізується на резисторах $R1 = 100 \text{ кОм}$ і $R2 = 27 \text{ кОм}$. Резистор $R1$ підключається між плюсовим виводом акумуляторної системи та вимірювальною точкою. Резистор $R2$ підключається між вимірювальною точкою та загальним проводом. З вимірювальної точки сигнал подається на аналоговий вхід ESP32, наприклад GPIO34. Цей вхід використовується лише для зчитування аналогового сигналу, тому він зручний для контролю напруги акумулятора.

На виході дільника напруги встановлюється конденсатор 100 нФ, підключений між вимірювальною точкою та GND. Він зменшує вплив імпульсних перешкод і робить вимірювання стабільнішим. Для захисту входу ESP32 використовується стабілітрон BZX55C3V3 на 3,3 В. Він підключається між аналоговим входом і загальним проводом та обмежує напругу на вході мікроконтролера у випадку перевищення допустимого рівня.

Для акумуляторної системи 12 В напруга повністю зарядженого акумулятора може досягати приблизно 14,4 В. При використанні дільника 100 кОм і 27 кОм на вході ESP32 формується напруга близько 3,06 В. Це значення не перевищує допустимий рівень аналогового входу мікроконтролера. У програмному забезпеченні ESP32 виконується зворотний перерахунок виміряного значення з урахуванням коефіцієнта дільника.

Плата ESP32 DevKit V1 виконує функції керуючого та обчислювального блока. Вона зчитує аналоговий сигнал з GPIO34, перетворює його у значення напруги акумуляторної системи, визначає орієнтовний рівень заряду та порівнює його з порогами 70 %, 50 % і 30 %. Також ESP32 обробляє команди, отримані від Telegram-бота, та формує керуючий сигнал для релейного модуля.

Для керування релейним модулем використовується цифровий вихід ESP32, наприклад GPIO26. Цей вивід підключається до входу IN релейного модуля SRD-05VDC-SL-C. Вивід VCC релейного модуля підключається до стабілізованої лінії 5 В від перетворювача LM2596S, а вивід GND — до спільної землі системи. При надходженні команди запуску генератора ESP32 змінює стан GPIO26, внаслідок чого реле спрацьовує та замикає контакти керуючого кола генератора.

Контакти реле використовуються для підключення до входу дистанційного запуску генератора. У структурній схемі застосовуються контакти COM і NO релейного модуля. У початковому стані контакт NO розімкнений. Після активації реле контакт NO замикається з COM, і на генератор подається сигнал запуску. Такий спосіб дозволяє керувати генератором через слабкострумний керуючий вхід без прямого підключення ESP32 до електричних кіл генератора.

Інформаційна взаємодія з користувачем реалізується через вбудований Wi-Fi-модуль ESP32. Після подавання живлення ESP32 підключається до Wi-Fi-маршрутизатора, отримує доступ до Інтернету та починає взаємодію з Telegram Bot API. Telegram-бот виконує функцію користувацького інтерфейсу, через який користувач отримує повідомлення про стан системи та надсилає команди керування.

У структурній схемі Telegram-бот не є окремим апаратним компонентом, але він є важливою частиною загальної системи. Він забезпечує обмін інформацією між користувачем і мікроконтролером. Команди /status, /battery, /start_generator, /stop_generator, /autostart_generator_on, /autostart_generator_off і /help надходять від користувача до ESP32 через мережу Інтернет. У відповідь

					КвРКІ 022075.22.04.26	Арк. 43
Зм.	Арк.	№ докum.	Підпис	Дата		

ESP32 надсилає повідомлення про рівень заряду, стан генератора, режим автозапуску та важливі події.

Структурна схема передбачає три основні типи сигналів. Перший тип - вимірювальні сигнали. Вони передаються від акумуляторної системи через дільник напруги до аналогового входу ESP32. Другий тип - керуючі сигнали. Вони формуються ESP32 і передаються на релейний модуль, який керує входом запуску генератора. Третій тип - інформаційні сигнали. Вони передаються між ESP32, Telegram-ботом і користувачем через Wi-Fi-мережу.

Робота системи за структурною схемою відбувається у такій послідовності. Після увімкнення живлення акумуляторна система подає напругу на перетворювач LM2596S. Перетворювач формує стабілізовану напругу 5 В для ESP32 і релейного модуля. ESP32 запускається, підключається до Wi-Fi-мережі та переходить у режим опитування Telegram-бота й контролю напруги акумуляторної системи.

Під час контролю ESP32 зчитує сигнал з GPIO34, обчислює напругу акумулятора та визначає рівень заряду. Якщо рівень заряду знижується до 70 %, 50 % або 30 %, система формує повідомлення користувачу. При досягненні критичного рівня 30 % і активованому режимі автозапуску ESP32 подає керуючий сигнал на GPIO26, унаслідок чого релейний модуль замикає керуюче коло запуску генератора.

У ручному режимі користувач надсилає команду `/start_generator` через Telegram-бота. ESP32 отримує команду, перевіряє ідентифікатор чату, активує релейний модуль і надсилає повідомлення про запуск генератора. При надсиланні команди `/stop_generator` ESP32 вимикає керуючий сигнал на релейному модулі та повідомляє користувача про зупинку генератора.

Для безпечної макетної перевірки генератор у структурній схемі може бути представлений його керуючим входом дистанційного запуску. Силові кола генератора не розглядаються та не підключаються до мікроконтролера. Такий

					КВРКІ 022075.22.04.26	Арк. 44
Зм.	Арк.	№ докum.	Підпис	Дата		

підхід дозволяє зосередитися на роботі логіки керування та взаємодії з користувачем, не ускладнюючи систему силовою електротехнічною частиною.

Важливою особливістю структурної схеми є відсутність проміжного сервера. ESP32 безпосередньо взаємодіє з Telegram Bot API через Wi-Fi-мережу. Це спрощує архітектуру засобу, зменшує кількість компонентів і відповідає вимозі мінімальної складності. Історія подій зберігається у чаті Telegram, тому окрема база даних у системі не використовується.

Запропонована структурна схема забезпечує виконання всіх основних функцій програмно-технічного засобу. Вона дозволяє контролювати рівень заряду акумуляторної системи, передавати повідомлення користувачу, приймати команди через Telegram-бота, керувати генератором у ручному режимі та виконувати автоматичний запуск при критичному рівні заряду. При цьому кількість апаратних компонентів залишається мінімальною.

Таким чином, структурна схема програмно-технічного засобу побудована на базі ESP32 DevKit V1, вузла вимірювання напруги акумулятора, релейного модуля SRD-05VDC-SL-C, DC-DC перетворювача LM2596S і Telegram-бота. Визначені зв'язки між компонентами дають змогу перейти до розроблення алгоритму контролю рівня заряду акумуляторної системи, який є основою для формування повідомлень користувачу та запуску генератора в автоматичному режимі.

2.4 Розроблення алгоритму контролю рівня заряду акумуляторної системи

Алгоритм контролю рівня заряду акумуляторної системи є одним з основних елементів програмно-технічного засобу віддаленого контролю енергозабезпечення. Його призначення полягає у періодичному вимірюванні напруги акумуляторної системи, перерахунку отриманого значення в орієнтовний рівень заряду, порівнянні цього рівня із заданими порогами та формуванні повідомлень користувачу через Telegram-бота.

У розроблюваній системі контроль заряду виконується мікроконтролером ESP32 DevKit V1. Напруга акумуляторної системи 12 В надходить на резистивний дільник напруги, побудований на резисторах $R1 = 100 \text{ кОм}$ і $R2 = 27 \text{ кОм}$. Вихід дільника підключається до аналогового входу GPIO34 мікроконтролера. Завдяки такому підключенню напруга акумулятора зменшується до рівня, допустимого для входу ESP32, після чого може бути оброблена аналого-цифровим перетворювачем.

Зчитування напруги виконується програмно з певним часовим інтервалом. У межах макетної реалізації доцільно виконувати вимірювання один раз на 5–10 секунд. Такий інтервал є достатнім, оскільки рівень заряду акумуляторної системи змінюється відносно повільно. Надто часте вимірювання не має практичної необхідності та може призводити до зайвого навантаження на програмний цикл мікроконтролера.

Після зчитування аналогового значення ESP32 виконує його перерахунок у напругу на вході мікроконтролера. Оскільки аналоговий сигнал надходить після дільника напруги, для визначення фактичної напруги акумулятора необхідно врахувати коефіцієнт ділення. Для обраних резисторів $R1 = 100 \text{ кОм}$ і $R2 = 27 \text{ кОм}$ коефіцієнт перерахунку становить приблизно 4,7. Це означає, що фактична напруга акумулятора приблизно у 4,7 раза більша за напругу, виміряну на аналоговому вході ESP32.

Для визначення орієнтовного рівня заряду акумуляторної системи використовується спрощений метод оцінювання за напругою. У програмному забезпеченні задаються мінімальне та максимальне значення напруги, які відповідають умовним рівням 0 % і 100 %. Для 12-вольтової акумуляторної системи у макетній реалізації можна прийняти, що напруга 11,0 В відповідає розрядженому стану, а напруга 12,7 В — повністю зарядженому стану без урахування режиму заряджання. Якщо система працює разом із зарядним пристроєм, верхнє значення може уточнюватися відповідно до характеристик конкретного акумулятора.

Рівень заряду у відсотках визначається шляхом лінійного масштабування між мінімальною та максимальною напругою. Якщо розраховане значення виходить за межі діапазону, програма обмежує його значеннями 0 % або 100 %. Це потрібно для того, щоб користувач не отримував некоректні значення, наприклад від'ємний заряд або заряд понад 100 %.

Слід зазначити, що визначення рівня заряду акумулятора лише за напругою має наближений характер. Реальний стан акумуляторної системи залежить від типу акумулятора, навантаження, температури, віку батареї та режиму її заряджання. Проте для розроблюваного програмно-технічного засобу такий підхід є достатнім, оскільки метою є не точне вимірювання залишкової ємності, а своєчасне виявлення досягнення контрольних порогів заряду.

У системі передбачено три основні пороги контролю: 70 %, 50 % і 30 %. Поріг 70 % використовується як попереджувальний. Його досягнення означає, що акумуляторна система почала помітно розряджатися, тому користувач повинен отримати перше повідомлення про зміну стану. Поріг 50 % вказує на середній рівень розряду та необхідність уважного контролю ситуації. Поріг 30 % вважається критичним, оскільки подальше розряджання акумулятора може призвести до втрати резервного живлення об'єкта.

При досягненні кожного порога ESP32 формує відповідне повідомлення для користувача. Наприклад, при зниженні заряду до 70 % Telegram-бот надсилає повідомлення про початок розряду акумуляторної системи. При досягненні 50 % користувач отримує повідомлення про необхідність контролю стану резервного живлення. При досягненні 30 % формується повідомлення про критичний рівень заряду. Якщо режим автоматичного запуску генератора активований, на цьому етапі додатково виконується команда запуску генератора.

Важливою вимогою до алгоритму є недопущення багаторазового надсилання однакових повідомлень. Якщо рівень заряду знизився нижче 70 %, система повинна надіслати повідомлення лише один раз. У наступних циклах вимірювання це саме повідомлення не повинно повторюватися. Для цього у

					КВРКІ 022075.22.04.26	Арк. 47
Зм.	Арк.	№ докum.	Підпис	Дата		

програмі використовуються логічні прапорці, які фіксують факт надсилання повідомлення для кожного порога. Окремі прапорці створюються для рівнів 70 %, 50 % і 30 %.

Наприклад, якщо змінна `warning70Sent` має значення `false` і рівень заряду став меншим або рівним 70 %, система надсилає повідомлення та змінює значення цієї змінної на `true`. Після цього повідомлення для порога 70 % більше не надсилається, доки стан системи не буде скинутий або заряд акумулятора не підніметься вище заданого рівня відновлення. Аналогічний принцип застосовується для порогів 50 % і 30 %.

Для підвищення стабільності роботи алгоритму доцільно передбачити невеликий гістерезис. Це означає, що скидання прапорця повідомлення виконується не відразу після підняття заряду вище порога, а лише після досягнення трохи вищого значення. Наприклад, повідомлення для порога 70 % може бути дозволено повторно лише після того, як рівень заряду підніметься вище 75 %. Такий підхід запобігає ситуації, коли через незначні коливання вимірної напруги система багаторазово надсилає повідомлення біля одного й того самого порога.

Алгоритм контролю рівня заряду також пов'язаний із режимом автоматичного запуску генератора. Якщо рівень заряду акумуляторної системи знижується до 30 % і змінна режиму автозапуску має активний стан, ESP32 подає керуючий сигнал на релейний модуль. Реле замикає керуюче коло дистанційного запуску генератора. Після цього Telegram-бот надсилає користувачу повідомлення про автоматичний запуск генератора через критичний рівень заряду.

Якщо режим автоматичного запуску вимкнений, при досягненні 30 % система не запускає генератор самостійно. У такому випадку користувач отримує лише повідомлення про критичний рівень заряду та може самостійно прийняти рішення щодо запуску генератора за допомогою команди

/start_generator. Такий підхід забезпечує можливість вибору між ручним і автоматичним керуванням.

Послідовність роботи алгоритму контролю заряду можна подати так:

- зчитування аналогового значення з входу GPIO34;
- перерахунок аналогового значення у напругу на вході ESP32;
- обчислення фактичної напруги акумуляторної системи з урахуванням коефіцієнта дільника;
- визначення орієнтовного рівня заряду у відсотках;
- обмеження значення заряду в межах від 0 % до 100 %;
- перевірка досягнення порогів 70 %, 50 % і 30 %;
- формування повідомлення користувачу при першому досягненні кожного порога;
- перевірка стану режиму автоматичного запуску при досягненні 30 %;
- запуск генератора через релейний модуль, якщо автозапуск активований;
- повернення до наступного циклу вимірювання.

У програмній реалізації алгоритм доцільно винести в окрему функцію, наприклад checkBatteryLevel(). Така функція повинна виконувати зчитування напруги, обчислення рівня заряду, перевірку порогів і формування подій. Це спрощує структуру програмного коду, оскільки основний цикл програми залишається зрозумілим і не перевантажується деталями вимірювання.

Для зменшення похибок вимірювання можна використовувати усереднення кількох послідовних значень аналогового входу. Наприклад, ESP32 може виконати 10 вимірювань, після чого обчислити середнє значення. Це дозволяє зменшити вплив короточасних перешкод і зробити результат стабільнішим. У поєднанні з конденсатором 100 нФ на виході дільника напруги такий підхід забезпечує достатню стабільність вимірювання для макетної системи.

Результатом роботи алгоритму є не лише числове значення рівня заряду, а й набір логічних станів, які використовуються іншими частинами системи. До

					КВРКІ 022075.22.04.26	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

таких станів належать ознаки досягнення порогів 70 %, 50 % і 30 %, стан критичного заряду, дозвіл на автоматичний запуск генератора та факт надсилання відповідних повідомлень. Ці стани використовуються модулем Telegram-бота та модулем керування релейним виходом.

Таким чином, розроблений алгоритм контролю рівня заряду акумуляторної системи забезпечує періодичне вимірювання напруги, наближене визначення заряду, повідомлення користувача про важливі рівні розряду та запуск генератора при критичному рівні заряду за умови активованого режиму автозапуску. Запропонований алгоритм є простим, зрозумілим і достатнім для реалізації у програмному забезпеченні ESP32. Наступним етапом є розроблення алгоритму ручного та автоматичного керування генератором, який безпосередньо використовує результати контролю рівня заряду.

2.5 Розроблення алгоритму ручного та автоматичного керування генератором

Алгоритм керування генератором є складовою частиною програмно-технічного засобу віддаленого контролю енергозабезпечення. Його призначення полягає у виконанні команд користувача, отриманих через Telegram-бота, а також у автоматичному запуску генератора при досягненні критичного рівня заряду акумуляторної системи. У розроблюваній системі генератор розглядається як резервне джерело енергії, що має вхід дистанційного запуску.

Керування генератором здійснюється мікроконтролером ESP32 DevKit V1 через одноканальний релейний модуль SRD-05VDC-SL-C. Керуючий вхід релейного модуля підключається до цифрового виводу GPIO26 мікроконтролера. При зміні стану цього виводу реле замикає або розмикає контакти керуючого кола генератора. У макетній реалізації стан реле може використовуватися для імітації запуску генератора за допомогою світлодіодного індикатора.

					КВРКІ 022075.22.04.26	Арк. 50
Зм.	Арк.	№ докum.	Підпис	Дата		

У системі передбачено два режими керування генератором: ручний і автоматичний. Ручний режим передбачає, що користувач самостійно надсилає команду запуску або зупинки генератора через Telegram-бота. Автоматичний режим передбачає, що система самостійно запускає генератор при досягненні критичного рівня заряду акумуляторної системи, якщо режим автозапуску попередньо активований користувачем.

Ручне керування генератором реалізується за допомогою команд `/generator_on` і `/generator_off`. Після отримання команди `/generator_on` мікроконтролер перевіряє ідентифікатор чату користувача. Якщо команда надійшла від дозволеного користувача, ESP32 активує вихід GPIO26, релейний модуль спрацьовує, а користувач отримує повідомлення про запуск генератора. Якщо команда надійшла від стороннього користувача, вона ігнорується.

Команда `/generator_off` використовується для ручного вимкнення генератора. Після її отримання ESP32 змінює стан виходу GPIO26, релейний модуль повертається у початковий стан, а користувачу надсилається повідомлення про вимкнення генератора. У ручному режимі команди користувача мають найвищий пріоритет, оскільки саме користувач приймає остаточне рішення щодо запуску або зупинки резервного джерела енергії.

Режим автоматичного запуску генератора вмикається командою `/autostart_generator_on`. Після отримання цієї команди програма встановлює внутрішню змінну `autostartEnabled` у активний стан і надсилає користувачу підтвердження. Вимкнення автоматичного запуску виконується командою `/autostart_generator_off`. У цьому випадку змінна `autostartEnabled` переходить у неактивний стан, а система надсилає повідомлення про вимкнення автозапуску.

Автоматичний запуск генератора виконується лише за одночасного виконання двох умов. Перша умова — рівень заряду акумуляторної системи повинен бути меншим або дорівнювати 30 %. Друга умова - режим автоматичного запуску повинен бути активований користувачем. Якщо обидві

					КВРКІ 022075.22.04.26	Арк. 51
Зм.	Арк.	№ докum.	Підпис	Дата		

умови виконуються, ESP32 активує релейний модуль і надсилає повідомлення про автоматичний запуск генератора через критичний рівень заряду.

Якщо рівень заряду досяг 30 %, але режим автозапуску вимкнений, генератор не запускається. У такому випадку система лише повідомляє користувача про критичний заряд акумуляторної системи. Користувач може самостійно прийняти рішення про запуск генератора та надіслати команду /generator_on. Такий підхід дозволяє поєднати автоматизацію з можливістю ручного контролю.

Для уникнення повторного запуску генератора в алгоритмі використовується змінна стану generatorRunning. Вона показує, чи вважається генератор увімкненим. Якщо генератор уже запущений, повторна команда запуску не повинна повторно активувати реле без потреби. У такому випадку Telegram-бот може надіслати користувачу повідомлення, що генератор уже перебуває в увімкненому стані.

Аналогічно, якщо користувач надсилає команду /stop_generator, але генератор уже вимкнений, система не повинна виконувати зайвих дій. У відповідь користувач може отримати повідомлення про те, що генератор уже перебуває у вимкненому стані. Це робить роботу системи зрозумілішою та зменшує ймовірність помилкових дій.

Пріоритети керування в алгоритмі визначаються таким чином:

- команда ручного вимкнення генератора має найвищий пріоритет;
- команда ручного запуску виконується незалежно від рівня заряду акумулятора;
- автоматичний запуск виконується лише при критичному заряді та активованому режимі автозапуску;
- повторний запуск не виконується, якщо генератор уже має стан «увімкнено»;
- повідомлення користувачу формується після кожної зміни стану генератора або режиму автозапуску.

Такий порядок пріоритетів потрібний для запобігання конфліктам між ручним і автоматичним режимами. Наприклад, якщо користувач вручну вимикає генератор, система повинна виконати цю команду незалежно від попереднього автоматичного запуску. У практичній реалізації після ручного вимкнення генератора доцільно також вимикати режим автозапуску або блокувати повторний автоматичний запуск до наступної команди користувача. Це дозволяє уникнути ситуації, коли користувач вимикає генератор, а система одразу запускає його повторно через низький рівень заряду.

У межах даної роботи приймається рішення, що команда `/stop_generator` вимикає генератор, але не змінює стан режиму автозапуску. Якщо користувач бажає повністю заборонити автоматичний запуск, він повинен окремо виконати команду `/autostart_generator_off`. Такий підхід є простим і зрозумілим, оскільки кожна команда виконує лише одну конкретну дію.

Алгоритм керування генератором також повинен формувати повідомлення про аварійне припинення роботи генератора. У мінімальній реалізації ця подія може бути представлена окремим логічним сигналом або програмною умовою. Якщо система фіксує, що генератор вимкнувся не за командою користувача, користувачу надсилається повідомлення про можливу несправність або відсутність палива. Детальна діагностика причини вимкнення у межах цієї роботи не виконується.

Загальна послідовність роботи алгоритму ручного керування генератором є такою. ESP32 перевіряє наявність нових команд Telegram-бота. Якщо отримано команду `/start_generator`, програма перевіряє дозвіл доступу, аналізує поточний стан генератора та активує релейний модуль. Після цього користувачу надсилається повідомлення про запуск. Якщо отримано команду `/stop_generator`, програма вимикає релейний модуль і надсилає повідомлення про зупинку генератора.

Послідовність роботи алгоритму автоматичного керування відрізняється тим, що запуск генератора ініціюється не командою користувача, а результатом

контролю рівня заряду акумуляторної системи. Після кожного циклу вимірювання напруги ESP32 перевіряє, чи досягнуто критичного рівня 30 %. Якщо цей рівень досягнуто, а режим автозапуску активований і генератор ще не запущений, система активує релейний модуль і надсилає повідомлення про автоматичний запуск.

У програмному забезпеченні алгоритм керування генератором доцільно реалізувати у вигляді окремих функцій: `startGenerator()`, `stopGenerator()` і `checkAutostart()`. Функція `startGenerator()` відповідає за активацію реле, зміну стану `generatorRunning` і формування повідомлення про запуск. Функція `stopGenerator()` виконує вимкнення реле, зміну стану генератора та надсилання повідомлення про зупинку. Функція `checkAutostart()` перевіряє умови автоматичного запуску.

Використання окремих функцій спрощує структуру програми та полегшує її тестування. Наприклад, команду `/start_generator` можна перевірити окремо від алгоритму контролю заряду. Так само режим автозапуску можна протестувати шляхом імітації критичного рівня заряду акумуляторної системи. Це дає можливість перевірити правильність роботи кожного режиму без ускладнення основного програмного циклу.

У макетній реалізації запуск генератора підтверджується спрацюванням релейного модуля. Якщо використовується світлодіодний індикатор, його ввімкнення відповідає стану «генератор увімкнено», а вимкнення — стану «генератор вимкнено». Такий спосіб перевірки є безпечним і достатнім для підтвердження працездатності алгоритму керування.

Стан генератора, режим автозапуску та рівень заряду акумуляторної системи повинні відобразитися у відповіді на команду `/status`. Це дозволяє користувачу в будь-який момент перевірити, чи запущений генератор, чи активований автоматичний запуск і який поточний рівень заряду акумуляторів. Команда `/status` є важливою для контролю правильності роботи алгоритму в реальних умовах.

Для підвищення надійності доцільно передбачити коротку затримку після зміни стану релейного модуля. Це дозволяє уникнути швидкого повторного перемикання реле при випадковому повторенні команди або нестабільному стані системи. У макетній реалізації достатньо програмної затримки тривалістю кілька секунд після запуску або зупинки генератора.

Алгоритм ручного та автоматичного керування генератором має бути узгоджений з алгоритмом контролю рівня заряду акумуляторної системи. Саме результат контролю заряду визначає потребу в автоматичному запуску. Водночас користувач через Telegram-бота повинен мати можливість змінювати режим автозапуску та вручну керувати генератором незалежно від поточного рівня заряду.

Таким чином, розроблений алгоритм забезпечує два основні способи керування генератором: ручний запуск і зупинку за командами користувача та автоматичний запуск при критичному рівні заряду акумуляторної системи. Застосування релейного модуля SRD-05VDC-SL-C та мікроконтролера ESP32 DevKit V1 дозволяє реалізувати цей алгоритм апаратно й програмно з мінімальною кількістю компонентів. Наступним етапом є розроблення логіки Telegram-бота, через який користувач взаємодіє з програмно-технічним засобом.

2.6 Розроблення логіки Telegram-бота

Telegram-бот у програмно-технічному засобі віддаленого контролю енергозабезпечення виконує функцію людино-машинного інтерфейсу. Через нього користувач отримує інформацію про стан акумуляторної системи, режим роботи генератора, активність автоматичного запуску та важливі події, що виникають під час роботи системи. Також Telegram-бот забезпечує приймання команд користувача та передавання їх до мікроконтролера ESP32 для подальшого виконання.

Використання Telegram-бота дозволяє не створювати окремий мобільний застосунок або вебінтерфейс. Користувач взаємодіє із системою через звичайний чат Telegram, що спрощує доступ до функцій контролю й керування. Усі повідомлення, які надсилає система, зберігаються в історії чату, тому Telegram також виконує роль простого журналу подій. Це відповідає прийнятому у роботі підходу до мінімальної програмної та апаратної складності.

У розроблюваній системі Telegram-бот не розміщується на окремому сервері. Оброблення команд виконується безпосередньо мікроконтролером ESP32, який через Wi-Fi-мережу звертається до Telegram Bot API. Після підключення до Інтернету ESP32 періодично перевіряє наявність нових повідомлень, аналізує отримані команди та виконує відповідні дії. У разі виникнення подій система самостійно надсилає повідомлення користувачу.

Для створення Telegram-бота використовується стандартний механізм BotFather. Після створення бота користувач отримує унікальний токен доступу, який застосовується у програмному забезпеченні ESP32 для взаємодії з Telegram Bot API. Крім токена, у програмі необхідно вказати ідентифікатор чату користувача. Це дає змогу надсилати повідомлення саме в потрібний чат і обмежити виконання команд лише дозволеним користувачем.

Базовий захист системи реалізується шляхом перевірки chat_id. Якщо команда надходить із чату, ідентифікатор якого не збігається з дозволеним значенням, ESP32 не виконує цю команду. Такий підхід є простим, але достатнім для макетної реалізації, оскільки запобігає випадковому або сторонньому керуванню генератором через Telegram-бота. Особливо це важливо для команд запуску та зупинки генератора.

Логіка Telegram-бота передбачає оброблення таких команд: /status, /generator_on, /generator_off, /autostart_generator_on, /autostart_generator_off і /history. Кожна команда має чітко визначене призначення та виконує одну конкретну дію. Це робить інтерфейс зрозумілим для користувача та спрощує програмну реалізацію.

Команда `/status` призначена для отримання загального стану системи. У відповіді на неї Telegram-бот повинен надсилати інформацію про поточний рівень заряду акумуляторної системи, стан генератора, стан режиму автоматичного запуску та останню важливу подію. Якщо у схемі передбачено контроль основного живлення, у відповіді також може відобразитися його стан. Ця команда є основною для швидкої перевірки працездатності системи.

Команда `/generator_on` призначена для ручного запуску генератора. Після її отримання ESP32 перевіряє ідентифікатор чату, поточний стан генератора та активує релейний модуль, якщо генератор ще не запущений. Після виконання дії бот надсилає користувачу повідомлення про запуск генератора. Якщо генератор уже перебуває у стані «увімкнено», користувач отримує повідомлення про те, що повторний запуск не потрібний.

Команда `/generator_off` використовується для ручного вимкнення генератора. Після її отримання ESP32 вимикає релейний модуль і змінює внутрішній стан генератора на «вимкнено». Після цього Telegram-бот надсилає повідомлення про зупинку генератора. Якщо генератор уже вимкнений, система не виконує зайвого перемикавання реле та повідомляє користувача про поточний стан.

Команди `/autostart_generator_on` і `/autostart_generator_off` призначені для керування режимом автоматичного запуску генератора. Перша команда активує режим автозапуску, за якого генератор запускається автоматично при досягненні критичного рівня заряду 30 %. Друга команда вимикає цей режим, після чого система лише повідомляє користувача про критичний заряд, але не запускає генератор без окремої команди.

Команда `/history` призначена для виведення історії подій у більш зручному та стислому форматі ніж як це представлено повідомленнями у боті.

Логіку оброблення команд можна подати як послідовність етапів. Спочатку ESP32 перевіряє наявність нових повідомлень у Telegram. Якщо повідомлення отримано, програма визначає його відправника та порівнює

chat_id з дозволеним значенням. Якщо доступ дозволено, текст повідомлення порівнюється з переліком підтримуваних команд. Після розпізнавання команди викликається відповідна функція програми. Якщо команда невідома, користувачу надсилається відповідне повідомлення.

У програмному забезпеченні ESP32 доцільно реалізувати окремі функції для кожної команди. Наприклад, команда /status може викликати функцію sendStatus(), команда /generator_on - функцію startGenerator(), команда /generator_off - функцію stopGenerator(). Така структура робить програмний код зрозумілішим і полегшує подальше тестування.

Крім відповідей на команди, Telegram-бот повинен надсилати автоматичні повідомлення про події. До таких подій належать досягнення порогів заряду 70 %, 50 % і 30 %, запуск генератора, вимкнення генератора, увімкнення або вимкнення режиму автозапуску, а також аварійне припинення роботи генератора. Автоматичні повідомлення формуються без прямої команди користувача, якщо відповідна подія виникла під час роботи системи.

Повідомлення повинні бути короткими та однозначними. Наприклад, при досягненні рівня заряду 70 % система може надіслати повідомлення: «Рівень заряду акумуляторної системи знизився до 70 %». При досягненні 30 % повідомлення повинно мати більш критичний характер: «Критичний рівень заряду акумуляторів - 30 %». Якщо при цьому увімкнено автозапуск, бот додатково повідомляє: «Виконано автоматичний запуск генератора».

Для уникнення надмірної кількості повідомлень логіка Telegram-бота повинна бути узгоджена з алгоритмом контролю рівня заряду. Повідомлення про порогові 70 %, 50 % і 30 % надсилаються лише один раз після першого досягнення відповідного рівня. Якщо рівень заряду залишається нижчим за поріг, повторні повідомлення не надсилаються в кожному циклі вимірювання. Це реалізується за допомогою програмних прапорців, описаних у підрозділі 2.4.

Окремо потрібно врахувати реакцію системи на втрату Wi-Fi-з'єднання. Якщо підключення до мережі відсутнє, ESP32 не може отримувати команди та

надсилати повідомлення через Telegram-бота. При цьому локальна логіка контролю заряду й автоматичного запуску може продовжувати роботу. Після відновлення Wi-Fi-з'єднання система повинна повторно підключитися до Telegram Bot API і продовжити оброблення команд. У практичній реалізації це забезпечується періодичною перевіркою стану Wi-Fi та повторним підключенням у разі розриву.

У відповіді на команду /status доцільно використовувати уніфікований формат повідомлення. Наприклад, повідомлення може містити такі рядки: рівень заряду акумуляторів, напруга акумуляторної системи, стан генератора, стан автозапуску та остання подія. Завдяки цьому користувач отримує повну інформацію в одному повідомленні, а формат відповіді залишається однаковим у різних режимах роботи.

З погляду програмної структури Telegram-бот є частиною вбудованого програмного забезпечення ESP32. Умовно його логіку можна поділити на три блоки: блок приймання команд, блок виконання команд і блок формування повідомлень. Блок приймання команд відповідає за отримання повідомлень від Telegram Bot API. Блок виконання команд викликає відповідні функції системи. Блок формування повідомлень створює текстові відповіді та надсилає їх користувачу.

Перевагою такого підходу є простота реалізації та відсутність потреби в проміжному сервері. Недоліком є залежність від стабільності Wi-Fi-мережі та доступу до Інтернету. Оскільки система призначена для невеликого об'єкта, це обмеження є прийнятним. Для підвищення практичної надійності Wi-Fi-маршрутизатор доцільно підключати до резервного живлення.

Таким чином, логіка Telegram-бота забезпечує зручну взаємодію користувача з програмно-технічним засобом. Вона охоплює приймання команд, перевірку доступу за chat_id, виконання керуючих дій, надсилання відповідей і автоматичне інформування про події. Розроблена логіка є достатньою для реалізації основних функцій системи: контролю заряду, перегляду поточного

стану, ручного керування генератором та налаштування автоматичного запуску. Наступним етапом є формування висновків до другого розділу, у яких узагальнюються прийняті проєктні рішення.

2.7 Висновки до другого розділу

У другому розділі було виконано проєктування програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів. На основі призначення системи та результатів аналізу, проведеного у першому розділі, визначено основні функціональні й нефункціональні вимоги до розроблюваного засобу. Встановлено, що система повинна забезпечувати контроль рівня заряду акумуляторної системи, передавання повідомлень користувачу через Telegram-бота, ручне керування генератором і автоматичний запуск генератора при критичному рівні заряду.

Для реалізації апаратної частини було обрано конкретний набір компонентів: мікроконтролерну плату ESP32 DevKit V1, резистивний дільник напруги на резисторах 100 кОм і 27 кОм, конденсатор 100 нФ, стабілітрон BZX55C3V3, релейний модуль SRD-05VDC-SL-C, понижувальний DC-DC перетворювач LM2596S, акумуляторну систему 12 В і генератор із входом дистанційного запуску. Такий склад є мінімально достатнім для виконання поставлених задач і не потребує використання GSM-модуля, окремого сервера або спеціального мобільного застосунку.

У розділі розроблено структурну схему програмно-технічного засобу, яка відображає взаємодію між акумуляторною системою, вузлом вимірювання напруги, мікроконтролером ESP32, релейним модулем, генератором, Wi-Fi-мережею та Telegram-ботом. Центральним елементом системи визначено ESP32, який виконує зчитування вимірювальних сигналів, оброблення команд користувача, формування повідомлень і керування виконавчим елементом.

					КВРКІ 022075.22.04.26	Арк. 60
Зм.	Арк.	№ докum.	Підпис	Дата		

Окрему увагу приділено розробленню алгоритму контролю рівня заряду акумуляторної системи. Алгоритм передбачає періодичне вимірювання напруги через аналоговий вхід GPIO34, перерахунок отриманого значення у відсотковий рівень заряду та перевірку порогових значень 70 %, 50 % і 30 %. Для запобігання надмірному дублюванню повідомлень передбачено використання програмних прапорців і принципу одноразового сповіщення при досягненні кожного порога.

Також було розроблено алгоритм ручного та автоматичного керування генератором. Ручне керування виконується за командами користувача /generator_on і /generator_off, які надходять через Telegram-бота. Автоматичний запуск генератора виконується при досягненні критичного рівня заряду 30 % за умови, що користувач попередньо активував режим автозапуску командою /autostart_generator_on. Для вимкнення цього режиму передбачено команду /autostart_generator_off.

Розроблено логіку Telegram-бота, який виконує функцію інтерфейсу користувача. Через нього користувач може переглядати поточний стан системи, перевіряти рівень заряду акумуляторів, запускати або вимикати генератор, керувати режимом автозапуску та отримувати автоматичні повідомлення про важливі події.

У результаті проектування було сформовано цілісну модель роботи програмно-технічного засобу. Вона поєднує апаратні елементи контролю й керування з програмною логікою ESP32 та Telegram-бота. Прийняті проєктні рішення забезпечують простоту, доступність і практичну реалізованість системи, а також дозволяють перейти до наступного етапу - програмно-апаратної реалізації, налаштування компонентів і тестування роботи засобу в основних режимах.

3 РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА РОБОТИ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

3.1 Підключення апаратних компонентів програмно-технічного засобу

Практична реалізація програмно-технічного засобу віддаленого контролю енергозабезпечення розпочинається з підключення апаратних компонентів. На цьому етапі формується електрична частина системи, яка забезпечує живлення мікроконтролера, вимірювання напруги акумуляторної системи та керування релейним модулем для запуску генератора.

Для реалізації апаратної частини використано мікроконтролерну плату ESP32 DevKit V1, понижувальний DC-DC перетворювач LM2596S, резистивний дільник напруги на резисторах 100 кОм і 27 кОм, конденсатор 100 нФ, стабілітрон BZX55C3V3, релейний модуль SRD-05VDC-SL-C, акумуляторну систему 12 В та вхід дистанційного запуску генератора. Обраний набір компонентів відповідає вимогам мінімальної складності та забезпечує виконання основних функцій системи.

Центральним елементом апаратної частини є плата ESP32 DevKit V1. Вона виконує зчитування вимірювального сигналу з акумуляторної системи, оброблення отриманих даних, зв'язок із Telegram-ботом через Wi-Fi та керування релейним модулем. Для вимірювання напруги використовується аналоговий вхід GPIO34, а для керування реле — цифровий вихід GPIO26.

Живлення електронної частини здійснюється від акумуляторної системи 12 В через понижувальний перетворювач LM2596S. Вхід IN+ перетворювача підключається до плюсового виводу акумулятора, а IN- — до загального проводу GND. На виході OUT+ формується стабілізована напруга 5 В, яка подається на вхід VIN/5V плати ESP32 та на вивід VCC релейного модуля. Вихід OUT- підключається до спільної лінії GND.

Важливою умовою правильної роботи схеми є наявність спільної землі. До загальної лінії GND підключаються мінусовий вивід акумулятора, вихід OUT-

					КВРКІ 022075.22.04.26	Арк. 62
Зм.	Арк.	№ докum.	Підпис	Дата		

перетворювача LM2596S, контакт GND плати ESP32, контакт GND релейного модуля, нижній вивід дільника напруги, конденсатор і стабілітрон. Це забезпечує коректне вимірювання напруги та правильне спрацювання керуючого сигналу GPIO26.

Вузол вимірювання напруги акумуляторної системи побудовано на резистивному дільнику. Резистор R1 номіналом 100 кОм підключається між плюсовим виводом акумулятора та вимірювальною точкою. Резистор R2 номіналом 27 кОм підключається між вимірювальною точкою та GND. Вимірювальна точка з'єднується з аналоговим входом GPIO34 мікроконтролера ESP32.

Застосування дільника напруги є необхідним, оскільки напруга акумуляторної системи 12 В не може подаватися безпосередньо на аналоговий вхід ESP32. При використанні резисторів 100 кОм і 27 кОм напруга на вході GPIO34 зменшується до безпечного рівня. Це дозволяє мікроконтролеру зчитувати напругу акумулятора через аналого-цифровий перетворювач без ризику пошкодження входу.

Для стабілізації вимірювального сигналу до входу GPIO34 додано конденсатор C1 номіналом 100 нФ, підключений між вимірювальною точкою та GND. Він зменшує вплив короткочасних перешкод і робить покази АЦП стабільнішими. Для захисту аналогового входу використано стабілітрон VZX55C3V3 на 3,3 В. Його катод підключається до вузла ADC_GPIO34, а анод - до GND. У разі перевищення допустимої напруги стабілітрон обмежує її рівень і захищає вхід мікроконтролера.

Керування генератором виконується через релейний модуль SRD-05VDC-SL-C. Вивід VCC релейного модуля підключається до лінії +5 В, вивід GND - до спільної землі, а вхід IN - до цифрового виходу GPIO26 мікроконтролера ESP32. Після подавання керуючого сигналу з GPIO26 реле спрацьовує та змінює стан контактів.

Контакти COM і NO релейного модуля використовуються для підключення до входу дистанційного запуску генератора. Один контакт входу запуску генератора підключається до COM, другий - до NO. У початковому стані ці контакти розімкнені. Після спрацювання реле контакти COM і NO замикаються, що імітує натискання кнопки запуску або подавання сигналу на вхід дистанційного керування генератором.

У реальній системі реле не повинно подавати напругу на генератор, якщо це не передбачено його документацією. У даній роботі релейний модуль розглядається як сухий контакт, який лише замикає два виводи входу дистанційного запуску. Такий спосіб підключення є безпечнішим і не потребує втручання у силову частину генератора.

Після підключення апаратних компонентів необхідно перевірити правильність основних з'єднань. Насамперед перевіряється вихідна напруга перетворювача LM2596S, яка повинна становити 5 В. Далі перевіряється наявність спільної землі між ESP32, релейним модулем, дільником напруги та джерелом живлення. Після цього перевіряється напруга на вході GPIO34, яка не повинна перевищувати допустимий рівень для мікроконтролера.

Окремо перевіряється робота релейного модуля. Для цього на вхід IN реле подається керуючий сигнал з GPIO26. При зміні стану виходу ESP32 реле повинно спрацювати, а контакти COM і NO повинні замикатися. У макетній реалізації це можна визначити за клацанням реле або за світлодіодним індикатором на модулі.

Таким чином, апаратна частина програмно-технічного засобу складається з мінімального набору компонентів, достатнього для реалізації контролю рівня заряду акумуляторної системи та керування генератором. Підключення ESP32, дільника напруги, перетворювача LM2596S і релейного модуля забезпечує фізичну основу для подальшої програмної реалізації. Наступним етапом є розроблення програмного забезпечення для ESP32, яке забезпечує вимірювання

					КВРКІ 022075.22.04.26	Арк. 64
Зм.	Арк.	№ докum.	Підпис	Дата		

напруги, оброблення команд Telegram-бота, формування повідомлень і керування генератором.

3.2 Розроблення програмного забезпечення для ESP32

Після підключення апаратних компонентів наступним етапом є розроблення програмного забезпечення для мікроконтролера ESP32 DevKit V1. Програмна частина забезпечує зчитування напруги акумуляторної системи, обчислення орієнтовного рівня заряду, оброблення команд користувача, керування релейним модулем і надсилання повідомлень через Telegram-бота.

Програмне забезпечення розробляється у середовищі Arduino IDE. Вибір цього середовища зумовлений його простотою, наявністю підтримки плат ESP32, великою кількістю прикладів і доступністю бібліотек для роботи з Wi-Fi та HTTPS-запитами. Для реалізації взаємодії з Telegram-ботом використовуються стандартні можливості ESP32 для підключення до мережі Wi-Fi та виконання запитів до Telegram Bot API.

У програмі використовується кілька основних груп налаштувань. До першої групи належать параметри Wi-Fi-мережі: назва мережі та пароль доступу. До другої групи належать параметри Telegram-бота: токен бота та ідентифікатор дозволеного чату. До третьої групи належать апаратні налаштування: номер аналогового входу для вимірювання напруги та номер цифрового виходу для керування реле.

У розробленій програмі для вимірювання напруги акумуляторної системи використовується аналоговий вхід GPIO34. Цей вхід підключений до виходу резистивного дільника напруги, сформованого на резисторах 100 кОм і 27 кОм. Для керування релейним модулем використовується цифровий вихід GPIO26. Саме через цей вивід ESP32 подає керуючий сигнал на вхід IN релейного модуля.

На початку роботи програма виконує ініціалізацію мікроконтролера. У функції `setup()` налаштовується послідовний порт для налагодження,

встановлюється режим роботи виводу GPIO26, задається початковий вимкнений стан реле, налаштовується аналоговий вхід GPIO34 та виконується підключення до Wi-Fi-мережі. Після успішного підключення система може надсилати повідомлення користувачу через Telegram-бота.

Основний цикл програми реалізовано у функції loop(). У цьому циклі мікроконтролер періодично виконує дві основні задачі: перевіряє рівень заряду акумуляторної системи та перевіряє наявність нових команд від Telegram-бота. Для цього використовуються часові інтервали, що дозволяє уникнути постійного блокування програми та забезпечує стабільну роботу системи.

Вимірювання напруги акумуляторної системи виконується не один раз, а шляхом усереднення кількох показів аналого-цифрового перетворювача. Такий підхід дозволяє зменшити вплив випадкових перешкод і зробити результат вимірювання стабільнішим. Після зчитування значення з GPIO34 програма перераховує його у напругу на аналоговому вході ESP32, а потім множить на коефіцієнт дільника напруги. Таким чином визначається фактична напруга акумуляторної системи.

Для обраного дільника напруги коефіцієнт перерахунку визначається співвідношенням резисторів R1 і R2. Оскільки резистор R1 має номінал 100 кОм, а R2 - 27 кОм, напруга на вході ESP32 є меншою за напругу акумулятора приблизно у 4,7 раза. Це враховується у програмі під час обчислення реальної напруги акумуляторної системи.

Після визначення напруги акумулятора програма виконує її перетворення в орієнтовний рівень заряду у відсотках. Для цього задаються мінімальна та максимальна напруги, які відповідають умовним рівням 0 % і 100 %. У межах цієї роботи прийнято, що напруга 11,0 В відповідає розрядженому стану, а напруга 12,7 В - зарядженому стану акумуляторної системи. Отримане значення обмежується в межах від 0 % до 100 %, щоб уникнути некоректного відображення результатів.

Слід зазначити, що такий спосіб визначення рівня заряду є наближеним. Він не враховує температуру, навантаження, тип акумулятора та його технічний стан. Проте для розроблюваного засобу цього достатньо, оскільки головним завданням є не точна діагностика акумулятора, а визначення моменту досягнення заданих порогових рівнів.

У програмному забезпеченні реалізовано контроль трьох порогів заряду: 70 %, 50 % і 30 %. При зниженні заряду до 70 % користувачу надсилається попереджувальне повідомлення. При досягненні 50 % система повідомляє про необхідність контролю стану резервного живлення. При досягненні 30 % формується повідомлення про критичний рівень заряду. Якщо при цьому увімкнено режим автоматичного запуску, система активує релейний модуль для запуску генератора.

Для запобігання багаторазовому надсиланню однакових повідомлень у програмі використано логічні прапорці. Окремий прапорець фіксує факт надсилання повідомлення для кожного порога. Якщо повідомлення для певного рівня вже було надіслане, воно не повторюється в кожному наступному циклі вимірювання. Повторне надсилання дозволяється лише після того, як рівень заряду знову підніметься вище заданого рівня відновлення.

Керування релейним модулем реалізовано через окремі функції запуску та зупинки генератора. При виконанні команди запуску програма активує вихід GPIO26, внаслідок чого релейний модуль замикає контакти COM і NO. У макетній реалізації це відповідає спрацюванню реле або ввімкненню індикатора. У практичному застосуванні ці контакти підключаються до входу дистанційного запуску генератора.

У програмі передбачено змінну, яка зберігає поточний стан генератора. Якщо генератор уже вважається увімкненим, повторна команда запуску не виконується. Аналогічно, якщо генератор перебуває у вимкненому стані, команда зупинки не спричиняє зайвого перемикання реле. Це дозволяє уникнути некоректних або повторних дій.

Режим автоматичного запуску генератора реалізовано за допомогою окремої логічної змінної. Користувач може змінювати її стан через Telegram-бота. Якщо автозапуск увімкнено і заряд акумуляторної системи знижується до 30 %, програма викликає функцію запуску генератора. Якщо автозапуск вимкнено, система лише надсилає повідомлення про критичний рівень заряду, але не активує релейний модуль.

Взаємодія з Telegram-ботом реалізується через періодичну перевірку нових повідомлень. ESP32 надсилає запит до Telegram Bot API, отримує нові повідомлення, аналізує текст команди та виконує відповідну дію. Для захисту від стороннього доступу програма перевіряє ідентифікатор чату. Якщо команда надійшла не від дозволеного користувача, вона ігнорується.

У програмі передбачено оброблення команд `/status`, `/start_generator`, `/stop_generator`, `/autostart_generator_on`, `/autostart_generator_off` і `/history`. Команда `/status` повертає загальний стан системи, команди запуску й зупинки керують реле, а команди автозапуску змінюють режим автоматичного керування. Команда `/history` надає користувачу історію подій.

Окрему увагу приділено повідомленням користувачу. Система надсилає повідомлення при запуску, при досягненні порогів заряду, при ручному запуску або зупинці генератора, а також при зміні режиму автозапуску. У мінімальній реалізації також передбачено повідомлення про можливу несправність або відсутність палива, якщо генератор було вимкнено не за командою користувача. Детальна діагностика причини вимкнення у межах цієї роботи не реалізується.

Для підвищення зручності програмний код поділено на окремі функціональні блоки. Окремі функції відповідають за підключення до Wi-Fi, надсилання повідомлень, зчитування напруги, обчислення рівня заряду, перевірку порогів, запуск генератора, зупинку генератора та оброблення команд Telegram-бота. Така структура полегшує налагодження програми та дозволяє окремо перевіряти кожну функцію.

Програмне забезпечення також враховує можливу втрату Wi-Fi-з'єднання. Якщо підключення до мережі відсутнє, система періодично намагається повторно підключитися. При цьому локальні функції, такі як зчитування напруги та перевірка умов автозапуску, можуть продовжувати виконуватися. Після відновлення Wi-Fi-з'єднання ESP32 знову отримує можливість приймати команди та надсилати повідомлення користувачу.

Таким чином, розроблене програмне забезпечення для ESP32 забезпечує виконання всіх основних функцій програмно-технічного засобу: вимірювання напруги акумуляторної системи, визначення рівня заряду, контроль порогів 70 %, 50 % і 30 %, керування релейним модулем, реалізацію автоматичного запуску генератора та обмін повідомленнями з користувачем через Telegram-бота. Наступним етапом є опис налаштування Telegram-бота та реалізації команд керування.

3.3 Налаштування Telegram-бота та реалізація команд керування

Для взаємодії користувача з програмно-технічним засобом використовується Telegram-бот. Він виконує роль інтерфейсу керування та інформування, через який користувач може отримувати поточний стан системи, перевіряти рівень заряду акумуляторної системи, запускати або вимикати генератор, а також керувати режимом автоматичного запуску.

Використання Telegram-бота дозволяє не розробляти окремий мобільний застосунок або вебінтерфейс. Усі дії виконуються через звичайний чат Telegram, що спрощує практичне використання системи. Крім того, історія повідомлень у чаті може виконувати роль простого журналу подій, у якому зберігаються повідомлення про зміну рівня заряду, запуск або вимкнення генератора та зміну режиму автозапуску.

Створення Telegram-бота виконується за допомогою службового бота BotFather. Після створення нового бота користувач отримує унікальний токен

					КвРКІ 022075.22.04.26	Арк. 69
Зм.	Арк.	№ докum.	Підпис	Дата		

доступу. Цей токен використовується у програмному забезпеченні ESP32 для надсилання запитів до Telegram Bot API. У тексті кваліфікаційної роботи повний токен не наводиться з міркувань безпеки.

Крім токена бота, у програмі використовується ідентифікатор дозволеного чату `chat_id`. Він потрібний для того, щоб ESP32 надсилав повідомлення саме потрібному користувачу та виконував команди лише з дозволеного чату. Такий підхід забезпечує базовий рівень захисту від стороннього доступу. Якщо команда надходить від іншого користувача або з невідомого чату, програма її ігнорує.

У програмному коді параметри Telegram-бота задаються у вигляді констант. До них належать токен бота та ідентифікатор дозволеного чату. Разом із параметрами Wi-Fi вони формують початкові налаштування системи, які необхідно вказати перед завантаженням програми у мікроконтролер ESP32.

Після запуску ESP32 підключається до Wi-Fi-мережі та отримує доступ до Інтернету. Далі мікроконтролер може надсилати повідомлення користувачу та періодично перевіряти наявність нових команд у Telegram. У розробленій системі не використовується окремий сервер, тому вся логіка оброблення команд виконується безпосередньо на ESP32.

Взаємодія з Telegram-ботом реалізується за принципом періодичного опитування. ESP32 через певний інтервал часу звертається до Telegram Bot API та перевіряє, чи з'явилися нові повідомлення. Якщо повідомлення отримано, програма зчитує його текст, визначає `chat_id` відправника та перевіряє, чи дозволено виконувати команду. Після цього команда порівнюється з переліком підтримуваних команд.

У системі реалізовано такі команди Telegram-бота:

- `/status` — отримання поточного стану системи;
- `/generator_on` - віддалений запуск генератора;
- `/generator_off` - віддалений вимкнення генератора;
- `/autostart_generator_on` - увімкнення автоматичного запуску;

- /autostart_generator_off - вимкнення автоматичного запуску;
- /history - перегляд історії подій.

Команда /status призначена для отримання загальної інформації про стан програмно-технічного засобу. У відповідь на цю команду користувач отримує повідомлення, у якому зазначено рівень заряду акумуляторної системи, напругу акумулятора, стан генератора, стан режиму автозапуску та останню зафіксовану подію. Ця команда є основною для швидкої перевірки роботи системи.

Команда /generator_on забезпечує ручний запуск генератора. Після її отримання ESP32 перевіряє, чи дозволено виконання команди, та аналізує поточний стан генератора. Якщо генератор ще не запущений, мікроконтролер подає сигнал на релейний модуль через GPIO26. Після спрацювання реле користувачу надсилається повідомлення про запуск генератора.

Команда /generator_off використовується для ручного вимкнення генератора. При її виконанні ESP32 змінює стан керуючого виходу GPIO26, релейний модуль повертається у початковий стан, а користувач отримує повідомлення про вимкнення генератора. Якщо генератор уже вимкнений, система не виконує повторного перемикавання та повідомляє користувача про поточний стан.

Команди /autostart_generator_on і /autostart_generator_off призначені для керування режимом автоматичного запуску генератора. Після виконання команди /autostart_generator_on у програмі активується відповідна логічна змінна. Якщо після цього рівень заряду акумуляторної системи знизиться до 30 %, система автоматично подасть сигнал на запуск генератора. Команда /autostart_generator_off вимикає цей режим, після чого при критичному заряді система лише надсилає попередження, але не запускає генератор самостійно.

Команда /history призначена для перегляду останніх подій, зафіксованих програмно-технічним засобом під час роботи. До таких подій належать повідомлення про досягнення рівнів заряду 70 %, 50 % і 30 %, запуск або вимкнення генератора, зміна режиму автозапуску, а також повідомлення про

можливу несправність або відсутність палива. У мінімальній реалізації історія подій може зберігатися у вигляді короткого списку останніх записів у пам'яті ESP32, а також дублюватися в чаті Telegram через автоматичні повідомлення. Після отримання команди /history мікроконтролер надсилає користувачу перелік останніх подій у текстовому вигляді. Це дозволяє швидко переглянути, які зміни стану системи відбувалися раніше, без пошуку окремих повідомлень у чаті.

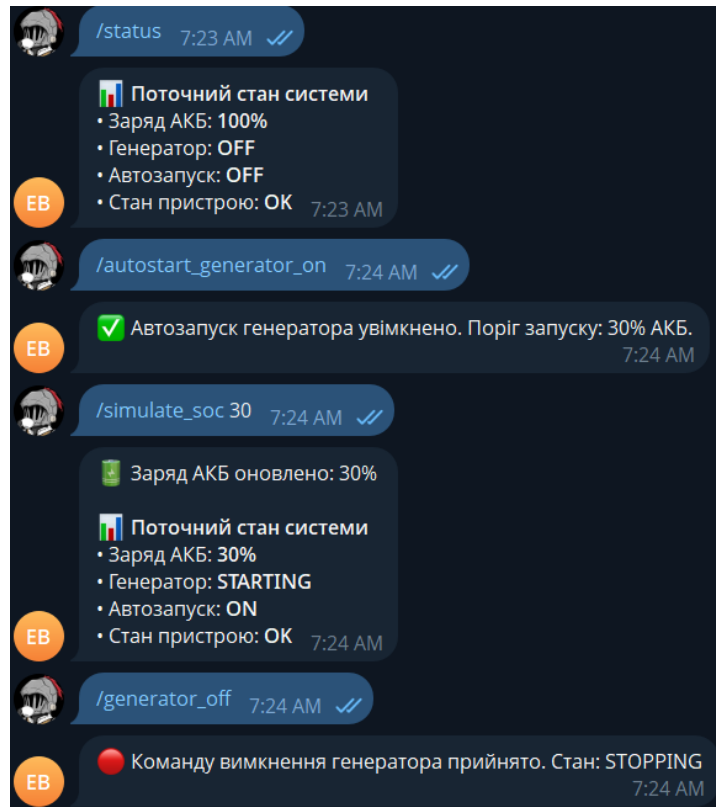


Рисунок 3.1 - Приклад взаємодії користувача з Telegram-ботом

Крім відповідей на команди користувача, Telegram-бот надсилає автоматичні повідомлення про важливі події. До таких подій належать досягнення рівнів заряду 70 %, 50 % і 30 %, запуск генератора, вимкнення генератора, увімкнення або вимкнення режиму автозапуску, а також повідомлення про можливу несправність або відсутність палива. Такі повідомлення формуються програмою без додаткового запиту користувача.

Для запобігання надмірній кількості повідомлень у програмі використано логічні прапорці. Наприклад, після надсилання повідомлення про досягнення рівня заряду 70 % система не повторює його в кожному наступному циклі вимірювання. Повторне повідомлення може бути дозволене лише після того, як заряд акумуляторної системи знову підніметься вище заданого рівня відновлення.

У разі втрати Wi-Fi-з'єднання ESP32 тимчасово не може отримувати команди від Telegram-бота та надсилати повідомлення користувачу. Проте локальна логіка контролю рівня заряду та автоматичного запуску генератора може продовжувати працювати. Після відновлення з'єднання мікроконтролер знову підключається до мережі та продовжує обмін повідомленнями з Telegram.

Таким чином, Telegram-бот забезпечує зручну взаємодію користувача з програмно-технічним засобом. Через нього реалізовано перегляд поточного стану системи, перевірку рівня заряду, ручне керування генератором, увімкнення та вимкнення режиму автозапуску, а також отримання автоматичних повідомлень про важливі події. Застосування перевірки chat_id дозволяє обмежити доступ до керування системою лише дозволеним користувачем, що підвищує безпечність роботи засобу.

3.4 Тестування роботи програмно-технічного засобу

Після розроблення апаратної моделі, програмного забезпечення для ESP32 та логіки Telegram-бота необхідно виконати перевірку працездатності програмно-технічного засобу. Метою тестування є підтвердження того, що розроблена система виконує основні функції: контролює рівень заряду акумуляторної системи, формує повідомлення користувачу, обробляє команди Telegram-бота та реалізує ручне й автоматичне керування генератором.

У межах даної роботи фізична апаратна частина програмно-технічного засобу представлена у вигляді моделі. Це означає, що практична перевірка

виконується не на повністю зібраному реальному пристрої, а на основі розроблених схем, програмної реалізації та імітації основних режимів роботи. Такий підхід дозволяє перевірити логіку функціонування системи без підключення до силового обладнання та реального генератора.

Апаратна частина засобу була подана у вигляді схеми підключення компонентів і моделі електричних вузлів. У моделі враховано акумуляторну систему 12 В, резистивний дільник напруги на резисторах 100 кОм і 27 кОм, конденсатор 100 нФ, стабілітрон BZX55C3V3, понижувальний перетворювач LM2596S, мікроконтролер ESP32 DevKit V1 та релейний модуль SRD-05VDC-SL-C. У моделі генератор подано як вхід дистанційного запуску, який замикається контактами реле.

Оскільки реальний генератор у межах роботи не підключався, його запуск і вимкнення перевіряються за станом керуючого сигналу GPIO26 та умовним спрацюванням релейного модуля. Такий спосіб тестування є безпечним, оскільки не потребує роботи з силовими колами генератора. У практичній реалізації цей самий сигнал може бути використаний для замикання входу дистанційного запуску генератора через контакти COM і NO релейного модуля.

Перевірка електричної частини передбачає аналіз роботи вузла вимірювання напруги акумуляторної системи. Основним завданням є підтвердження того, що напруга 12-вольтової акумуляторної системи після резистивного дільника зменшується до рівня, допустимого для аналогового входу ESP32. При використанні резисторів 100 кОм і 27 кОм напруга на вході GPIO34 є приблизно у 4,7 раза меншою за напругу акумулятора. Це дозволяє безпечно подавати вимірювальний сигнал на мікроконтролер.

У середовищі моделювання перевіряється, що при зміні напруги акумуляторної системи відповідно змінюється напруга у вузлі ADC_GPIO34. Це підтверджує правильність роботи дільника напруги. Додатково враховується наявність конденсатора 100 нФ, який виконує функцію фільтрації короткочасних

перешкод, та стабілітрона на 3,3 В, який обмежує напругу на вході мікроконтролера.

Програмне тестування виконується шляхом перевірки основних режимів роботи програмного забезпечення ESP32. На першому етапі перевіряється запуск програми, ініціалізація виводів GPIO34 і GPIO26, налаштування параметрів Wi-Fi та підключення до Telegram-бота. Після запуску система повинна перейти в робочий режим і бути готовою до приймання команд користувача.

Наступним етапом є перевірка вимірювання напруги акумуляторної системи. У моделі або в програмній імітації задаються різні значення напруги, які відповідають різним рівням заряду. Програма повинна правильно перераховувати виміряне значення у відсотковий рівень заряду та відобразити його у відповідях Telegram-бота.

Окремо перевіряється робота порогових повідомлень. При зниженні рівня заряду до 70 % система повинна надіслати користувачу попереджувальне повідомлення. При досягненні 50 % має бути сформоване повідомлення про необхідність контролю стану резервного живлення. При досягненні 30 % система повинна повідомити про критичний рівень заряду. Якщо режим автозапуску увімкнено, при цьому додатково має виконуватися запуск генератора.

Тестування команд Telegram-бота виконувалося шляхом послідовного надсилання основних команд і перевірки реакції системи. Команди /status і /battery використовувалися для перевірки виведення поточного стану, рівня заряду та напруги акумуляторної системи. Команда /history дозволяла переглянути останні події, зокрема повідомлення про пороги 70 %, 50 % і 30 %, запуск або вимкнення генератора та зміну режиму автозапуску. Команди /start_generator і /stop_generator перевіряли ручне керування генератором, що в моделі підтверджувалося зміною стану GPIO26 і умовним спрацюванням релейного модуля. Команди /autostart_generator_on і /autostart_generator_off використовувалися для перевірки режиму автоматичного запуску. Команда /help підтвердила коректність виведення списку доступних функцій. За результатами

					КВРКІ 022075.22.04.26	Арк. 75
Зм.	Арк.	№ докum.	Підпис	Дата		

перевірки встановлено, що Telegram-бот забезпечує виконання основних команд керування та інформування користувача.

У процесі тестування також враховано обмеження розробленої системи. Оскільки апаратна частина представлена як модель, результати перевірки підтверджують працездатність розробленої логіки та правильність прийнятих схемних рішень, але не замінюють повноцінних випробувань на реальному генераторі. Для практичного впровадження необхідно додатково перевірити відповідність входу дистанційного запуску конкретної моделі генератора, допустимі параметри релейного модуля та умови електробезпеки.

У результаті тестування встановлено, що розроблена модель програмно-технічного засобу виконує поставлені функції. Система забезпечує контроль напруги акумуляторної системи, перерахунок її в орієнтовний рівень заряду, надсилання повідомлень користувачу, оброблення команд Telegram-бота, ручне керування генератором і автоматичний запуск при критичному рівні заряду. Це підтверджує працездатність запропонованого рішення на рівні моделі та програмної реалізації.

3.5 Висновки до третього розділу

У третьому розділі було розглянуто програмно-апаратну реалізацію та тестування програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів. Апаратну частину системи представлено у вигляді моделі, яка відображає основні вузли підключення: мікроконтролер ESP32 DevKit V1, понижувальний перетворювач LM2596S, резистивний дільник напруги, елементи захисту аналогового входу, релейний модуль та вхід дистанційного запуску генератора.

Було описано підключення основних компонентів системи. Для вимірювання напруги акумуляторної системи використано вхід GPIO34, до якого через дільник напруги підключається акумуляторна система 12 В. Для керування

релейним модулем використано вихід GPIO26. Понижувальний перетворювач LM2596S забезпечує формування стабілізованої напруги 5 В для живлення ESP32 та релейного модуля. У моделі генератор подано як вхід дистанційного запуску, що замикається контактами реле.

Розроблено програмне забезпечення для ESP32, яке забезпечує зчитування напруги акумуляторної системи, перерахунок її в орієнтовний рівень заряду, контроль порогових значень 70 %, 50 % і 30 %, формування повідомлень користувачу та керування генератором. Для запобігання дублюванню повідомлень у програмі використано логічні прапорці, які фіксують факт надсилання попередження для кожного порога.

У розділі також описано налаштування Telegram-бота та реалізацію основних команд керування. Через бот користувач може отримати поточний стан системи, перевірити заряд акумуляторної системи, переглянути історію подій, запустити або вимкнути генератор, а також увімкнути чи вимкнути режим автоматичного запуску. Для базового захисту передбачено перевірку ідентифікатора чату користувача.

Тестування роботи засобу виконувалося на рівні моделі та програмної логіки. Було перевірено реакцію системи на команди Telegram-бота, формування повідомлень при досягненні порогів заряду, роботу ручного керування генератором і режим автоматичного запуску при критичному рівні заряду 30 %. У моделі виконання команд запуску та зупинки підтверджувалося зміною стану керуючого виходу GPIO26 і умовним спрацюванням релейного модуля.

За результатами тестування встановлено, що розроблена модель програмно-технічного засобу виконує основні функції, визначені у попередніх розділах. Система забезпечує контроль стану акумуляторної системи, інформування користувача через Telegram-бота, ручне керування генератором і автоматичний запуск при критичному рівні заряду. Отримані результати підтверджують працездатність запропонованого рішення на рівні апаратної моделі та програмної реалізації.

Разом із тим слід зазначити, що фізична апаратна частина не була реалізована у вигляді повністю зібраного пристрою. Тому для практичного впровадження системи необхідно додатково виконати перевірку на реальному обладнанні, врахувати параметри конкретного генератора, умови електробезпеки та особливості підключення входу дистанційного запуску. Незважаючи на це, розроблена модель підтверджує правильність прийнятих технічних рішень і може бути використана як основа для подальшого створення фізичного зразка пристрою.

					<u>КВРКІ 022075.22.04.26</u>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було розроблено модель програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів на базі мікроконтролера ESP32 та месенджер-платформи Telegram. Запропоноване рішення призначене для контролю рівня заряду акумуляторної системи, інформування користувача про важливі події та керування генератором у ручному або автоматичному режимі.

У першому розділі проведено аналіз предметної області віддаленого контролю енергозабезпечення об'єктів. Розглянуто призначення систем резервного живлення, особливості контролю акумуляторної системи та доцільність використання месенджер-платформи як інтерфейсу користувача. Визначено основні функціональні можливості засобу, серед яких контроль рівня заряду, надсилання повідомлень при досягненні порогів 70 %, 50 % і 30 %, ручний запуск і вимкнення генератора, а також увімкнення та вимкнення режиму автозапуску.

У другому розділі проведено проектування програмно-технічного засобу. Визначено вимоги до системи, обрано основні компоненти апаратної частини та розроблено структурну схему засобу. Для реалізації моделі обрано мікроконтролер ESP32 DevKit V1, понижувальний перетворювач LM2596S, резистивний дільник напруги на резисторах 100 кОм і 27 кОм, конденсатор 100 нФ, стабілітрон BZX55C3V3 та релейний модуль SRD-05VDC-SL-C. Також розроблено алгоритми контролю рівня заряду акумуляторної системи, ручного й автоматичного керування генератором та логіку роботи Telegram-бота.

У третьому розділі виконано програмно-апаратну реалізацію та тестування розробленого засобу на рівні моделі. Описано підключення апаратних компонентів, розроблено програмне забезпечення для ESP32, налаштовано логіку взаємодії з Telegram-ботом і реалізовано основні команди керування: /status, /battery, /history, /start_generator, /stop_generator,

					КВРКІ 022075.22.04.26	Арк. 79
Зм.	Арк.	№ докум.	Підпис	Дата		

/autostart_generator_on, /autostart_generator_off та /help. Фізична апаратна частина у межах роботи представлена як модель, що відображає принципи підключення та роботи основних вузлів системи.

Під час тестування перевірено реакцію системи на команди Telegram-бота, формування повідомлень при досягненні заданих порогів заряду, роботу ручного керування генератором і режим автоматичного запуску при критичному рівні заряду 30 %. У моделі виконання команд запуску та зупинки генератора підтверджувалося зміною стану керуючого виходу GPIO26 і умовним спрацюванням релейного модуля.

У результаті виконання роботи підтверджено працездатність запропонованого рішення на рівні апаратної моделі та програмної реалізації. Розроблена система забезпечує віддалене інформування користувача про стан акумуляторної системи, підтримує основні команди керування через Telegram-бота та реалізує автоматичний запуск генератора при критичному рівні заряду. Запропонована модель може бути використана як основа для подальшого створення фізичного зразка пристрою та його перевірки на реальному обладнанні.

					КВРКІ 022075.22.04.26	Арк. 80
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Espressif Systems. ESP-IDF Programming Guide for ESP32. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html> (дата звернення: 16.06.2026).
2. Espressif Systems. Get Started with ESP32. ESP-IDF Programming Guide. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html> (дата звернення: 16.06.2026).
3. Espressif Systems. ESP32 Series Datasheet. URL: https://documentation.espressif.com/esp32_datasheet_en.pdf (дата звернення: 16.06.2026).
4. Espressif Systems. ESP32-WROOM-32 Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (дата звернення: 16.06.2026).
5. Espressif Systems. ESP32-DevKitC V4 Getting Started Guide. URL: <https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/hw-reference/esp32/get-started-devkitc.html> (дата звернення: 16.06.2026).
6. Espressif Systems. Wi-Fi API Reference. ESP-IDF Programming Guide. URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_wifi.html (дата звернення: 16.06.2026).
7. Espressif Systems. Analog to Digital Converter (ADC). ESP-IDF Programming Guide. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/adc/index.html> (дата звернення: 16.06.2026).
8. Espressif Systems. ADC Calibration Driver. ESP-IDF Programming Guide. URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/adc/adc_calibration.html (дата звернення: 16.06.2026).
9. Espressif Systems. Arduino-ESP32 Documentation. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/> (дата звернення: 16.06.2026).

					КВРКІ 022075.22.04.26	Арк. 81
Зм.	Арк.	№ док.ум.	Підпис	Дата		

10. Espressif Systems. Installing Arduino-ESP32. Arduino-ESP32 Documentation. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html> (дата звернення: 16.06.2026).

11. Espressif Systems. ADC. Arduino-ESP32 Documentation. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/adc.html> (дата звернення: 16.06.2026).

12. Espressif Systems. Preferences. Arduino-ESP32 Documentation. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/preferences.html> (дата звернення: 16.06.2026).

13. Arduino. Installing a Board Package in the IDE 2. 2024. URL: <https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-board-manager> (дата звернення: 16.06.2026).

14. Arduino. analogRead(). Arduino Documentation. 2025. URL: <https://docs.arduino.cc/language-reference/en/functions/analog-io/analogRead/> (дата звернення: 16.06.2026).

15. ArduinoJson. ArduinoJson 7 Documentation. URL: <https://arduinojson.org/v7/> (дата звернення: 16.06.2026).

16. ArduinoJson. ArduinoJson 7 API Reference. URL: <https://arduinojson.org/v7/api/> (дата звернення: 16.06.2026).

17. Telegram. Telegram Bot API. URL: <https://core.telegram.org/bots/api> (дата звернення: 16.06.2026).

18. Telegram. From BotFather to “Hello World”. Telegram Bots Tutorial. URL: <https://core.telegram.org/bots/tutorial> (дата звернення: 16.06.2026).

19. IETF. RFC 8259. The JavaScript Object Notation (JSON) Data Interchange Format. 2017. URL: <https://datatracker.ietf.org/doc/html/rfc8259> (дата звернення: 16.06.2026).

20. IETF. RFC 8446. The Transport Layer Security (TLS) Protocol Version 1.3. 2018. URL: <https://datatracker.ietf.org/doc/html/rfc8446> (дата звернення: 16.06.2026).

					КВРКІ 022075.22.04.26	Арк. 82
Зм.	Арк.	№ докум.	Підпис	Дата		

21. IETF. RFC 9110. HTTP Semantics. 2022. URL: <https://datatracker.ietf.org/doc/rfc9110/> (дата звернення: 16.06.2026).
22. Texas Instruments. LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3-A Step-Down Voltage Regulator. Datasheet. URL: <https://www.ti.com/lit/ds/symlink/lm2596.pdf> (дата звернення: 16.06.2026).
23. onsemi. LM2596: 3.0 A, Step-Down Switching Regulator. Datasheet. 2022. URL: <https://www.onsemi.com/download/data-sheet/pdf/lm2596-d.pdf> (дата звернення: 16.06.2026).
24. JLCPCB. SRD-05VDC-SL-C. Ningbo Songle Relay. URL: https://jlcpcb.com/partdetail/Ningbo_SongleRelay-SRD_05VDC_SLC/C35449 (дата звернення: 16.06.2026).
25. HandsOn Technology. SRD-05VDC-SL-C Relay Datasheet. URL: <https://www.handsontec.com/dataspecs/relay/SRD-05VDC-SL-C.pdf> (дата звернення: 16.06.2026).
26. Vishay Intertechnology. BZX55-Series Small Signal Zener Diodes. Datasheet. URL: <https://www.vishay.com/docs/85604/bzx55.pdf> (дата звернення: 16.06.2026).
27. Analog Devices. LTspice Simulator. URL: <https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html> (дата звернення: 16.06.2026).
28. Analog Devices. Getting Started with LTspice. EngineerZone. 2025. URL: <https://ez.analog.com/design-tools-and-calculators/ltspice/a/faqs-docs/c/getting-started-with-ltspice> (дата звернення: 16.06.2026).
29. Altera. Quartus Prime Design Software. URL: <https://www.altera.com/products/development-tools/quartus> (дата звернення: 16.06.2026).
30. Intel. Intel Quartus Prime Standard Edition User Guide: Third-party Simulation. 2024. URL: <https://docs.altera.com/r/docs/683080/current> (дата звернення: 16.06.2026).

31. Altera. Quartus Prime Pro and Standard Software User Guides. URL: <https://www.altera.com/design/guidance/software/quartus-user-guides> (дата звернення: 16.06.2026).

32. EasyEDA. EasyEDA Pro Documentation: Net Label. 2024. URL: <https://prodocs.easyeda.com/en/schematic/place-net-label/> (дата звернення: 16.06.2026).

33. EasyEDA. EasyEDA Pro Documentation: Import KiCad. 2024. URL: <https://prodocs.easyeda.com/en/import-export/import-kicad/> (дата звернення: 16.06.2026).

34. EasyEDA. EasyEDA Pro Documentation: Convert Schematic to PCB. URL: <https://prodocs.easyeda.com/en/schematic/design-update-convert-schematic-to-pcb/> (дата звернення: 16.06.2026).

35. OWASP Foundation. OWASP Internet of Things Project. URL: <https://owasp.org/www-project-internet-of-things/> (дата звернення: 16.06.2026).

36. OWASP Foundation. OWASP IoT Security Testing Guide. 2024. URL: <https://owasp.org/www-project-iot-security-testing-guide/> (дата звернення: 16.06.2026).

37. NIST. The NIST Cybersecurity Framework (CSF) 2.0. 2024. URL: <https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf> (дата звернення: 16.06.2026).

38. KiCad. EasyEDA Import Format Documentation. 2026. URL: <https://devdocs.kicad.org/en/import-formats/easyeda/index.html> (дата звернення: 16.06.2026).

39. ESPBoards. DOIT ESP32 DEVKIT V1 Development Board Details and Pinout. URL: <https://www.espboards.dev/esp32/esp32doit-devkit-v1/> (дата звернення: 16.06.2026).

40. GitHub. Espressif Systems. Arduino Core for the ESP32 Family of SoCs. URL: <https://github.com/espressif/arduino-esp32> (дата звернення: 16.06.2026).

					КВРКІ 022075.22.04.26	Арк. 84
Зм.	Арк.	№ докum.	Підпис	Дата		

41. Espressif Systems. Wi-Fi API. Arduino-ESP32 Documentation. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html> (дата звернення: 16.06.2026).

42. Espressif Systems. GPIO. Arduino-ESP32 Documentation. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/gpio.html> (дата звернення: 16.06.2026).

43. Espressif Systems. Libraries. Arduino-ESP32 Documentation. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/libraries.html> (дата звернення: 16.06.2026).

44. Espressif Systems. Migration from 2.x to 3.0. Arduino-ESP32 Documentation. URL: https://docs.espressif.com/projects/arduino-esp32/en/latest/migration_guides/2.x_to_3.0.html (дата звернення: 16.06.2026).

45. Espressif Systems. ESP HTTP Client. ESP-IDF Programming Guide. URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/protocols/esp_http_client.html (дата звернення: 16.06.2026).

46. РКС Компоненти – РАДІОМАГ. Каталог електронних компонентів. URL: <https://www.rcscomponents.kiev.ua/> (дата звернення: 16.06.2026).

47. ArduinoJson. deserializeJson(). ArduinoJson 7 API Reference. URL: <https://arduinojson.org/v7/api/json/deserializejson/> (дата звернення: 16.06.2026).

48. ArduinoJson. serializeJson(). ArduinoJson 7 API Reference. URL: <https://arduinojson.org/v7/api/json/serializejson/> (дата звернення: 16.06.2026).

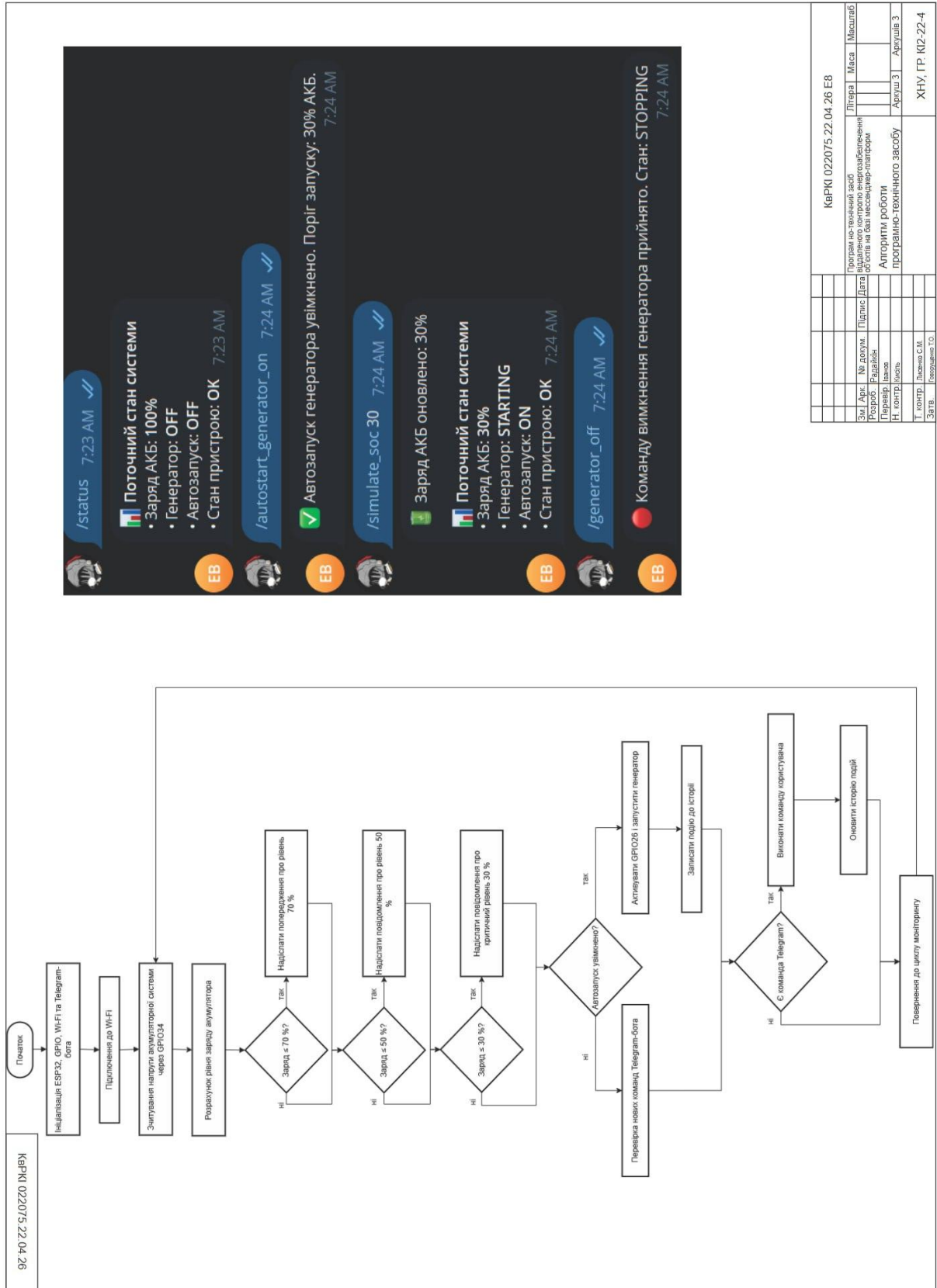
49. GitHub. Espressif Systems. Arduino-ESP32 Releases. URL: <https://github.com/espressif/arduino-esp32/releases> (дата звернення: 16.06.2026).

50. EasyEDA. EasyEDA Pro Documentation: Import Changes from Schematic. URL: <https://prodocs.easyeda.com/en/pcb/design-import-changes-from-schematic/> (дата звернення: 16.06.2026).

					КВРКІ 022075.22.04.26	Арк. 85
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК В (обов'язковий)

Алгоритм роботи програмно-технічного засобу



КВРКІ 022075.22.04.26 EB									
Зм	Арх	№ доумк	Підпис	Дата	Літера	Маса	Масштаб		
								Програмно-технічний засіб	
								об'єктів на сайті месенджер-платформ	
								Алгоритм роботи	
								програмно-технічного засобу	
								Архив 3 Архив 3	
								ХНУ, ГР: КІ2-22-4	
								Затв.	
								Повне С.М.	
								Говоримо ІО.	

Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Тарас Радайкін

ІІІІ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-4

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

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

1 травня 2026 року



Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 7%

ID: 275717 Назва: БКР Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ Додано в БД: 2026-06-17 Автора: Тарас РАДАЙКІН Керівники: Олексій ІВАНОВ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	138196	1119	3010 (2%)	33 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Тарас РАДАЙКІН

Співавтор:

Назва: Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ

Експерт: Олексій ІВАНОВ

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 1.53%

Коефіцієнт подібності 2: 0.46%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 2

Дата створення звіту: 2026-06-17 11:14:07.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-06-18

Дата

Доцент Андрій Нічепорук

експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ

Автор Тарас РАДАЙКІН

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент Олексій ІВАНОВ

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

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

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 1,53%; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

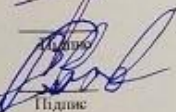
01.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Підпис


Підпис


Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК
Ім'я, ПРІЗВИЩЕ

Олексій ІВАНОВ
Ім'я, ПРІЗВИЩЕ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Радайкін Тарас Романович

Тема: Програмно-технічний засіб віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 85

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення програмно-технічного засобу віддаленого контролю енергозабезпечення об'єктів на базі месенджер-платформ. У роботі запропоновано систему моніторингу стану акумуляторної системи та керування резервним генератором із використанням мікроконтролерної платформи ESP32, Wi-Fi-з'єднання та Telegram-бота. Прийняті технічні рішення передбачають інформування користувача про рівень заряду, автоматичний запуск генератора при досягненні критичного порогу, ручне керування генератором, ведення історії подій та повідомлення про аварійне вимкнення або відсутність палива.

2. Робота повністю відповідає поставленому дипломному завданню, оскільки в ній розглянуто апаратну та програмну частини системи, обґрунтовано вибір компонентів, розроблено алгоритм роботи засобу віддаленого контролю та наведено результати реалізації основних функцій керування й оповіщення користувача.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено аналіз предметної області, обґрунтовано актуальність віддаленого контролю енергозабезпечення об'єктів через месенджер-платформи та сформульовано постановку задачі. У другому розділі виконано проєктування програмно-технічного засобу, обґрунтовано вибір ESP32, датчиків, релейних елементів і засобів зв'язку, а також визначено логіку контролю заряду акумуляторної системи й керування генератором через Telegram-бота. У третьому

розділі описано реалізацію апаратної та програмної частин системи, основні команди бота, сценарії роботи, журнал подій і результати перевірки працездатності. У роботі використано сучасні підходи до побудови IoT-систем, мікроконтролерного керування та віддаленого моніторингу.

4. Позитивні сторони роботи: Робота має високу практичну цінність, оскільки запропонований засіб може бути використаний для підвищення надійності резервного енергозабезпечення житлових будинків, невеликих офісів та інших об'єктів. До позитивних сторін слід віднести актуальність теми, простоту апаратної реалізації, використання доступних компонентів, зручність взаємодії користувача через Telegram-бота, а також можливість подальшого розширення системи для роботи з сонячними панелями, інверторами та акумуляторними батареями більшої ємності.

5. До недоліків роботи можна віднести те, що експериментальна перевірка системи виконана в обмеженій кількості режимів, а питання тривалого автономного тестування та роботи за нестабільного Wi-Fi-з'єднання розглянуто не в повному обсязі.

6. Пояснювальна записка оформлена коректно, згідно з чинними вимогами до оформлення кваліфікаційних робіт. Графічні матеріали, структурні схеми, електрична принципова схема та додатки загалом відповідають змісту роботи й відображають основні технічні рішення.

7. Відгук про роботу в цілому: Кваліфікаційна робота виконана на належному науково-технічному рівні. У роботі продемонстровано вміння автора аналізувати предметну область, обирати технічні засоби, розробляти алгоритми керування, створювати програмну частину системи та обґрунтовувати практичне застосування запропонованого рішення.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно /65/E

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Сячонь Михайло Васильович Ph.D, ст. викладач, к. е. н.
Київського

"18" 06 2026 р.

ШШ (підпис)