

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:
«АВТОМАТИЗОВАНА СИСТЕМА «РОЗУМНА ФЕРМА» НА ОСНОВІ
ТЕХНОЛОГІЙ ІНТЕРНЕТУ РЕЧЕЙ»

на здобуття освітнього ступеня бакалавр
за спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис) Світлана ЮХИМОВИЧ
(ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти гр. ІСД-41
Світлана ЮХИМОВИЧ
(ім'я, ПРІЗВИЩЕ)
Керівник: Ольга ПОЛОНЕВИЧ
к.т.н., доцент (ім'я, ПРІЗВИЩЕ)
Рецензент: _____
(ім'я, ПРІЗВИЩЕ)

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

_____ Каміла СТОРЧАК

“_____” _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Юхимович Світлани Валеріївни

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Автоматизована система «розумна ферма» на основі технологій інтернету речей

керівник кваліфікаційної роботи Ольга ПОЛОНЕВИЧ к.т.н., доцент

(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20” лютого 2026р. № 51

2. Строк подання кваліфікаційної роботи “01” червня 2026р.

3. Вихідні дані кваліфікаційної роботи:

1. Інтернет речей.
2. Розумна ферма.
3. Програмно-апаратні засоби.
4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз технологій, апаратних засобів та програмного забезпечення для побудови системи «розумна ферма» на основі IoT.

2. Дослідження особливостей налаштування та програмна реалізація апаратно-програмного комплексу «Розумна ферма».

3. Практична реалізація автоматизованої IoT-системи «Розумна ферма» з веборієнтованим керуванням

5. Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання “20” лютого 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	24.02.26-03.03.26	
2.	Аналіз технологій, апаратних засобів та програмного забезпечення для побудови системи «розумна ферма» на основі IoT.	04.03.26-19.03.26	
3.	Дослідження особливостей налаштування та програмна реалізація апаратно-програмного комплексу «Розумна ферма».	19.03.26-31.03.26	
4.	Практична реалізація автоматизованої IoT-системи «Розумна ферма» з веборієнтованим керуванням	01.04.26-25.04.26	
5.	Оцінка результатів роботи, розробка висновків.	26.04.26-17.05.26	
6.	Формування демонстраційних матеріалів та доповіді	19.05.26-25.05.26	
7.	Оформлення роботи	26.05.26-30.05.26	

Здобувач вищої освіти _____

(підпис)

Світлана ЮХИМОВИЧ

(ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи _____

(підпис)

Ольга ПОЛОНЕВИЧ

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступня бакалавр: 64 стор., 78 рис., 2 табл., 30 джерела.

Мета роботи - розробка автоматизованої системи «Розумна ферма» на основі IoT-технологій для моніторингу та автоматичного керування параметрами фермерського середовища.

Об'єкт дослідження – процеси автоматизації та моніторингу параметрів фермерського середовища.

Предмет дослідження – програмно-апаратні засоби та методи побудови IoT-системи «Розумна ферма».

Короткий зміст роботи. У першому розділі здійснено аналіз сучасних IoT-технологій, бездротових протоколів зв'язку та апаратних компонентів для побудови системи «Розумна ферма». Досліджено архітектуру системи та можливості середовища Arduino IDE для розробки програмного забезпечення.

У другому розділі виконано налаштування та програмну реалізацію апаратно-програмного комплексу «Розумна ферма» на базі ESP32. Реалізовано алгоритми автоматичного керування освітленням, мікрокліматом і системою зрошення, а також проведено тестування роботи системи.

У практичній частині роботи було реалізовано систему «Розумна ферма» з функціями автоматичного поливу, контролю мікроклімату та вебкерування через Wi-Fi. Розроблено вебінтерфейс для дистанційного моніторингу й керування компонентами системи в режимі реального часу.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ, ПРОГРАМА, ПЛАТФОРМА, ДАТЧИК, СЕНСОР, КОНФІГУРАЦІЯ, СИТЕМА, ФЕРМА, ПРОГРАМА, МОНІТОРИНГ, КОНТРОЛЬ.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ТЕХНОЛОГІЙ, АПАРАТНИХ ЗАСОБІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПОБУДОВИ СИСТЕМИ «РОЗУМНА ФЕРМА» НА ОСНОВІ ІОТ.....	10
1.1 Аналіз застосування технологій Інтернету речей у розумному фермерстві.....	10
1.2 Аналіз протоколів передачі даних для IoT-систем розумної ферми.....	13
1.3 Огляд апаратних компонентів.....	15
1.4 Інтегроване середовище розробки Arduino.....	27
2 НАЛАШТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ АПАРАТНО- ПРОГРАМНОГО КОМПЛЕКСУ «РОЗУМНА ФЕРМА».....	30
2.1 Система автоматичного керування освітленням.....	30
2.2 Система контролю температури.....	34
2.3 Система моніторингу вологості ґрунту.....	40
2.4 Система автоматичного зрошення.....	44
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ ІОТ-СИСТЕМИ «РОЗУМНА ФЕРМА» З ВЕБОРІЄНТОВАНИМ КЕРУВАННЯМ.....	50
3.1 Реалізація системи автоматичного поливу на основі IoT-технологій.....	50
3.2 Налаштування системи автоматизованого контролю мікроклімату.....	54
3.3 Налаштування та програмна реалізація вебкерування.....	58
3.4. Програмна реалізація інтерфейсу користувача та функцій керування...	64
ВИСНОВКИ.....	67
ПЕРЕЛІК ПОСИЛАНЬ.....	70
ДОДАТОК А ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СИСТЕМИ МОНІТОРИНГУ ТА КЕРУВАННЯ «РОЗУМНОЮ ФЕРМОЮ».....	73
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	78

ВСТУП

Актуальність теми. Сучасний розвиток інформаційних технологій, цифровізації та концепції Інтернету речей (IoT) суттєво впливає на автоматизацію різних сфер діяльності людини, зокрема аграрного сектору. Використання інтелектуальних систем керування, сенсорних мереж та бездротових технологій зв'язку дозволяє забезпечити постійний моніторинг параметрів навколишнього середовища, автоматизувати виробничі процеси та підвищити ефективність використання ресурсів. У сучасних умовах розвиток систем розумного фермерства є одним із перспективних напрямів впровадження IoT-технологій у сільському господарстві.

Зростання населення, зміна кліматичних умов, дефіцит водних ресурсів та необхідність підвищення продуктивності аграрного виробництва зумовлюють потребу у впровадженні автоматизованих систем моніторингу й керування фермерським середовищем. Традиційні методи контролю процесів поливу, освітлення та підтримання мікроклімату потребують значних витрат часу й ресурсів, а також не завжди забезпечують оперативне реагування на зміну умов навколишнього середовища. Саме тому використання технологій Інтернету речей дозволяє реалізувати ефективні системи автоматичного контролю та дистанційного керування в режимі реального часу.

Мета роботи - розробка автоматизованої системи «Розумна ферма» на основі IoT-технологій для моніторингу та автоматичного керування параметрами фермерського середовища.

Для досягнення мети, у бакалаврській роботі успішно виконано наступні завдання:

1. Аналіз технологій, апаратних засобів та програмного забезпечення для побудови системи «розумна ферма» на основі IoT.

2. Дослідження особливостей налаштування та програмна реалізація апаратно- програмного комплексу «Розумна ферма»

3. Практична реалізація автоматизованої IoT-системи «Розумна ферма» з веборієнтованим керуванням

Об'єкт дослідження – процеси автоматизації та моніторингу параметрів фермерського середовища.

Предмет дослідження – програмно-апаратні засоби та методи побудови IoT-системи «Розумна ферма».

Методи дослідження. Під час виконання поставлених задач у роботі було застосовано методи програмно-апаратного проєктування, моделювання систем, програмування мікроконтролерів, методи тестування IoT-систем та методи експериментальних досліджень.

Наукова новизна одержаних результатів. Наукова новизна роботи полягає у вдосконаленні підхід до автоматизованого контролю фермерського середовища шляхом інтеграції сенсорних модулів, бездротових технологій передачі даних та веборієнтованих засобів моніторингу в єдину IoT-систему.

Практична значущість одержаних результатів. Практична значущість полягає у створенні працездатної автоматизованої системи «Розумна ферма», яка може використовуватись для моніторингу та автоматичного керування процесами поливу, освітлення і мікроклімату, забезпечуючи підвищення ефективності використання ресурсів та спрощення контролю фермерського середовища.

1 АНАЛІЗ ТЕХНОЛОГІЙ, АПАРАТНИХ ЗАСОБІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПОБУДОВИ СИСТЕМИ «РОЗУМНА ФЕРМА» НА ОСНОВІ ІОТ

1.1 Аналіз застосування технологій Інтернету речей у розумному фермерстві

Зростання чисельності населення у світі зумовлює необхідність впровадження сучасних технологій та інноваційних підходів у сферу сільського господарства для забезпечення продовольчих потреб суспільства. Одним із перспективних напрямів розвитку аграрної галузі стало використання технологій Інтернету речей. Після активного розвитку комп'ютерних систем, мережі Інтернет і мобільних комунікацій саме ІоТ став наступним етапом цифрової трансформації, що суттєво впливає на розвиток світової економіки та сучасних інформаційних технологій.

Інтернет речей являє собою сукупність взаємопов'язаних пристроїв, сенсорів, програмного забезпечення, машин та користувачів, які об'єднані через мережеві платформи для передачі даних, обміну інформацією та взаємодії між фізичним і цифровим середовищем [1]. Основу ІоТ формують інтелектуальні системи керування, автономні мережі та технології оброблення інформації.

На сьогодні технології Інтернету речей використовуються у багатьох сферах діяльності, серед яких точне землеробство, логістика, моніторинг навколишнього середовища, транспортні системи, медицина, розумні будівлі та системи громадської безпеки. У сільському господарстві ІоТ забезпечує автоматизацію процесів спостереження та керування, що стало можливим завдяки розвитку сенсорних, комунікаційних і мережевих технологій.

Використання Інтернету речей у аграрному секторі дозволяє здійснювати збір даних у режимі реального часу та автоматично контролювати основні

параметри середовища. Сучасні IoT-рішення об'єднують сільськогосподарську техніку, сенсори та виконавчі механізми в єдину систему, що підвищує ефективність виконання таких процесів, як посів, полив, внесення добрив, обприскування, моніторинг стану рослин і збирання врожаю.

Постійний розвиток Інтернет-технологій сприяє вдосконаленню систем моніторингу аграрної продукції та автоматизації виробничих процесів. Застосування IoT у сільському господарстві забезпечує підвищення якості продукції, покращення контролю за умовами вирощування рослин і збільшення точності роботи систем спостереження та керування у розумному фермерстві.

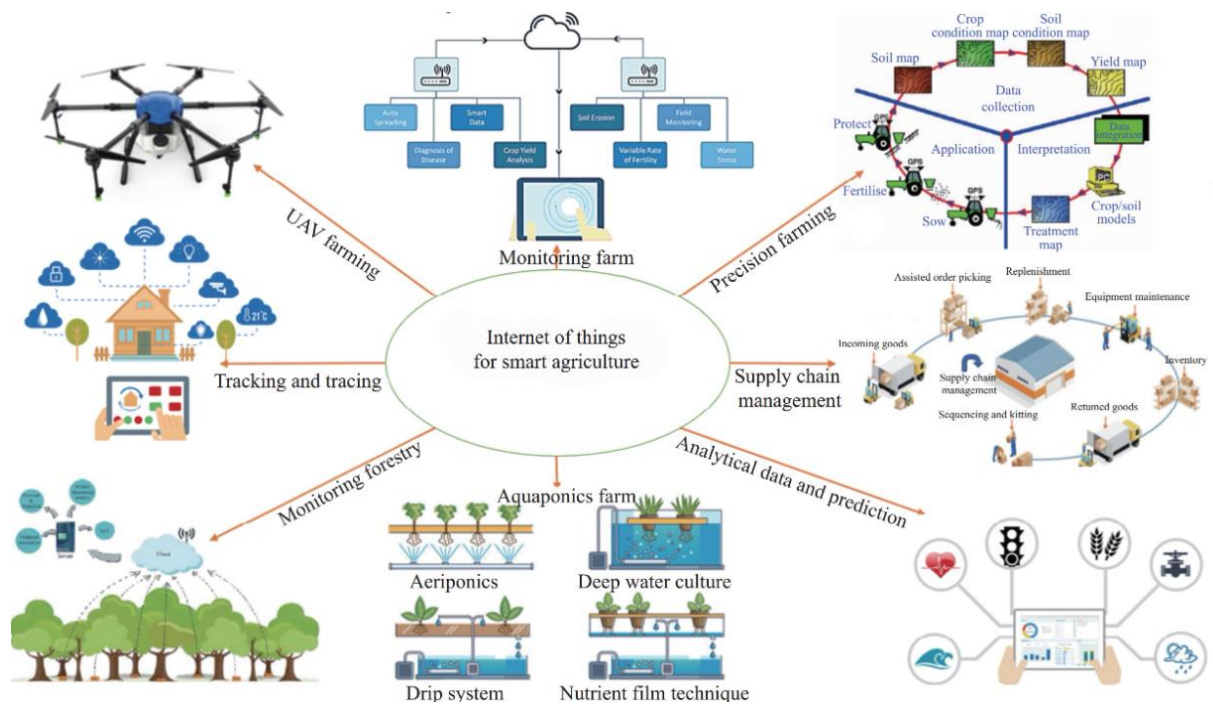


Рис.1.1 Застосування Інтернету речей у різних сферах для розумного сільського господарства

Для отримання даних із сенсорних пристроїв у системах розумного фермерства широко застосовуються бездротові сенсорні мережі (WSN). Для ефективного аналізу, зберігання та оброблення інформації технології Інтернету речей інтегруються з хмарними сервісами, що дозволяє автоматизувати процес прийняття рішень [2]. Важливу роль у функціонуванні сучасного розумного

фермерства відіграють інформаційно-комунікаційні технології, наземні сенсори, системи автоматизованого керування, роботизовані комплекси та автономні транспортні засоби.

Для стабільної роботи таких систем необхідне використання високошвидкісного доступу до Інтернету, сучасних мобільних мереж та супутникових технологій. У дослідженні [3] описано практичне застосування IoT-системи в режимі реального часу для виявлення захворювань листя рису та бананів за допомогою супутникових знімків і сенсорів, встановлених на фермерських угіддях. Отримані дані оброблялися сервером, після чого результати аналізу передавалися фермерам для оперативного прийняття рішень.

Відповідно до даних Продовольчої та сільськогосподарської організації ООН, щорічні втрати врожаю через шкідників, хвороби та недостатній контроль стану посівів можуть становити від 20 % до 40 %. Використання інтелектуальних систем та сенсорних технологій дозволяє здійснювати моніторинг погодних умов, контролювати стан ґрунту та визначати оптимальну кількість добрив, необхідних для ефективного росту рослин. Недостатнє або неправильне внесення добрив негативно впливає на родючість ґрунтів та продуктивність сільськогосподарського виробництва.

Основні напрями використання технологій Інтернету речей у сільському господарстві наведено на рис. 1.2.

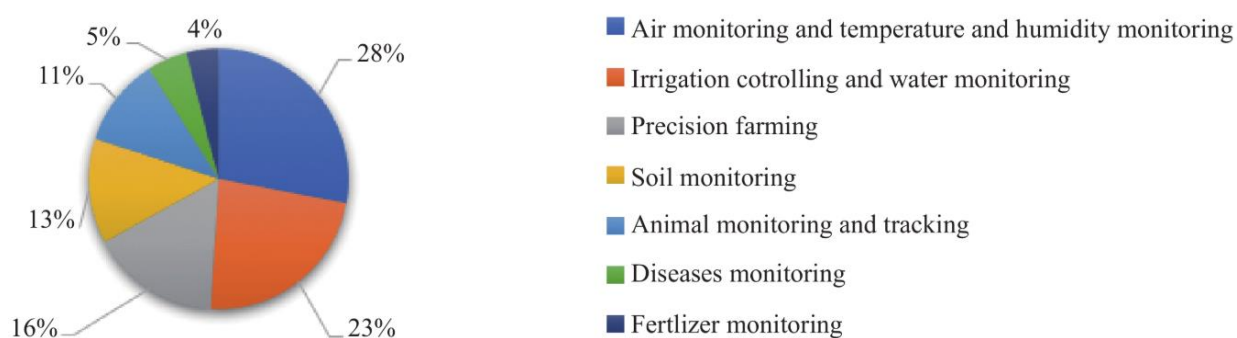


Рис.1.2. Галузеве застосування Інтернету речей для розумного сільського господарства

1.2 Аналіз протоколів передачі даних для IoT-систем розумної ферми

Розвиток технологій бездротового зв'язку, сенсорних систем та дистанційного зондування суттєво спростив реалізацію концепції точного землеробства. Використання бездротових мереж забезпечує передачу даних до центрів оброблення інформації, що сприяє підвищенню ефективності сільськогосподарського виробництва та покращенню якості продукції. Ефективне використання обмежених ресурсів аграрної галузі значною мірою залежить від стабільного зв'язку, який забезпечується сучасними бездротовими технологіями [3].

Більшість сільськогосподарських угідь розташовані у віддалених районах із недостатнім або нестабільним доступом до мережі Інтернет. Для вирішення цієї проблеми застосовуються бездротові сенсорні мережі (WSN), які дозволяють організувати передачу даних від польових сенсорів до вузлів із надійним підключенням до хмарних сервісів [2]. Такі мережі об'єднують сенсорні вузли, комунікаційні модулі та відкриті протоколи обміну даними.

Основу WSN складають компактні енергоефективні пристрої, призначені для збору та передавання інформації з мінімальним енергоспоживанням. У системах розумного фермерства широко використовуються недорогі сенсорні модулі з обмеженими обчислювальними ресурсами, що дозволяє реалізувати масштабовані системи моніторингу параметрів середовища. Важливим етапом розгортання сенсорної мережі є визначення оптимальної кількості вузлів та їх розташування. Для цього використовуються методи локалізації, зокрема визначення часу прибуття сигналу, різниці часу прибуття та індикації рівня прийнятого сигналу RSSI [4].

Однією з ключових задач при побудові бездротових систем є забезпечення достатньої дальності передачі даних та зменшення енергоспоживання сенсорних вузлів. Вибір технології зв'язку залежить від частотного діапазону, структури мережі, топології та площі покриття. У системах розумного сільського

господарства найбільш поширеними є протоколи ближнього та далекого радіуса дії з низьким енергоспоживанням, основні типи яких наведено на рис. 1.3.

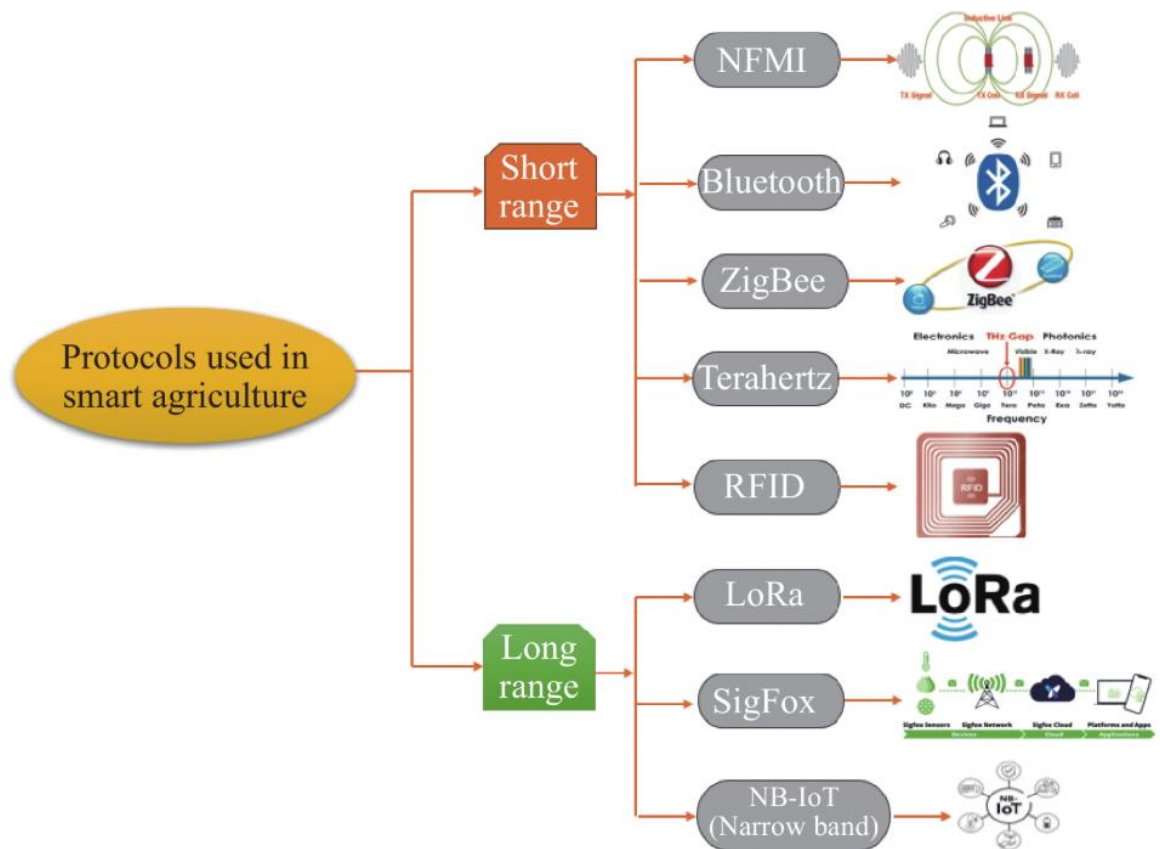


Рис.1.3. Протоколи, що використовуються для Інтернету речей у розумному сільському господарстві

Функціонування систем розумного землеробства неможливе без використання сучасних технологій бездротового зв'язку. У сільському господарстві активно застосовуються такі технології, як SigFox, Bluetooth, ZigBee, LoRa, NB-IoT та Wi-Fi, які забезпечують передавання даних між сенсорами, контролерами та хмарними сервісами [5].

Основними напрямками використання бездротових технологій у аграрній сфері є автоматизоване зрошення, моніторинг стану ґрунту, виявлення шкідників, контроль мікроклімату в теплицях, а також системи захисту рослин. Завдяки впровадженню таких технологій забезпечується автоматизація процесів освітлення та поливу, контроль рівня поживних речовин, зменшення ризику виникнення

лісових пожеж і підтримання необхідного рівня кисню у рибницьких господарствах.

Використання бездротових мереж у системах розумного фермерства сприяє підвищенню ефективності управління аграрними процесами, зниженню витрат ресурсів та покращенню якості сільськогосподарської продукції.

Таблиця 1.1.

Протоколи зв'язку

Технологія	Wi-Fi	ZigBee	Z-Wave	LoRaWAN
Стандарт зв'язку	IEEE 802.11	IEEE 802.15.4	ITU-T G.9959	IEEE 802.15.4
Робоча частота	2.4 ГГц	2.4 ГГц	868.4 МГц	868 МГц
Радіус дії	50–100 м	75–100 м	до 100 м	до 15 км
Енергоспоживання	Високе	Низьке	Низьке	Дуже низьке
Швидкість даних	до 600 Мбіт/с	до 250 Кбіт/с	до 100 Кбіт/с	до 50 Кбіт/с
Топологія мережі	«Зірка»	Mesh	Mesh	«Зірка»
Проникна здатність	Низька	Середня	Середня	Висока
Макс. вузлів	~250	65 000+	232	тисячі

1.3 Огляд апаратних компонентів автоматизованої системи «розумна ферма»

Основу апаратної частини системи становить мережа сенсорних пристроїв, призначених для безперервного збору інформації про параметри навколишнього середовища та її подальшого перетворення у цифровий формат. Взаємодія датчиків із мікроконтролером забезпечує оперативний моніторинг стану системи та створює основу для автоматизованого прийняття керуючих рішень.

Сільськогосподарські системи Інтернету речей включають декілька основних складових, серед яких сенсорні пристрої, мережеве підключення, бездротові технології зв'язку, а також засоби збору, передавання та оброблення

даних. Ефективність функціонування IoT-систем значною мірою залежить від правильно обраної технології бездротового зв'язку, яка визначається частотним діапазоном, дальністю передавання сигналу та умовами використання [6].

У системах розумного фермерства застосовується широкий спектр датчиків, вибір яких залежить від поставлених функціональних задач і особливостей аграрного процесу. Найпоширеніші типи сенсорів, що використовуються у сільськогосподарських IoT-системах, наведено на рис. 1.4.

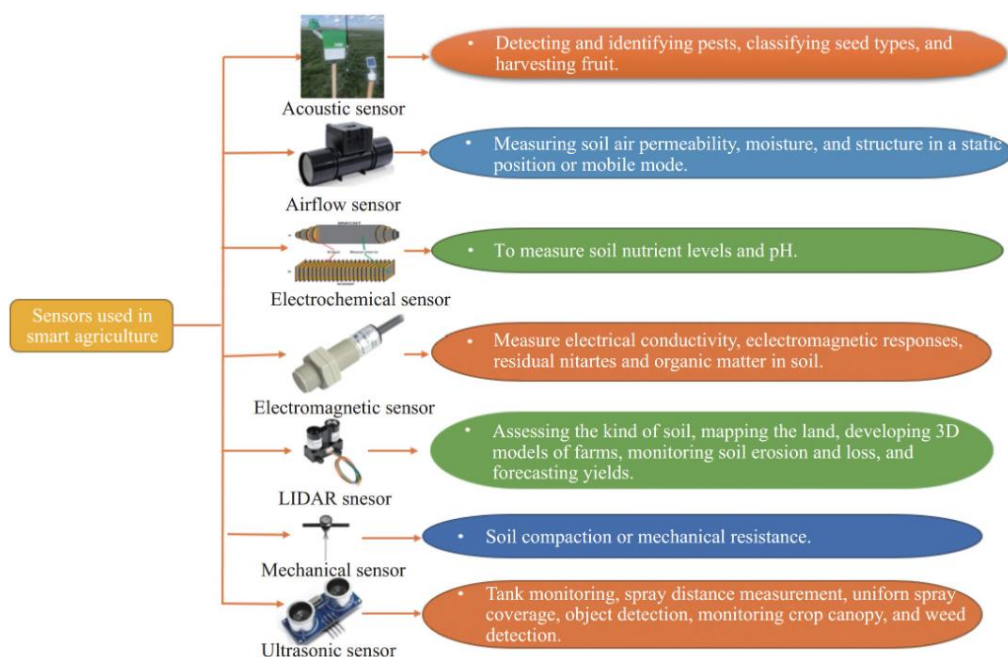


Рис.1.4. датчики та їх застосування в розумному сільському господарстві

Досліджувана система має багаторівневу архітектуру, яка забезпечує повний цикл роботи — від збору фізичних параметрів до їх оброблення, аналізу та відображення користувачу. В основі системи лежить взаємодія апаратних і програмних компонентів, де нижній рівень виконує функції збору телеметричних даних, а верхній забезпечує моніторинг і керування через користувацький інтерфейс. Використання мікроконтролера з підтримкою бездротових технологій дозволило об'єднати всі елементи в єдину IoT-мережу, що забезпечує автоматизований контроль та двосторонній обмін даними між користувачем і виконавчими пристроями в режимі реального часу [7].

Для реалізації верхнього рівня системи використано стандартні вебтехнології. Мова HTML застосовується для побудови структури вебінтерфейсу та розміщення елементів керування, а CSS забезпечує візуальне оформлення та адаптацію сторінок для різних типів пристроїв. Обмін даними між вебінтерфейсом і мікроконтролером реалізовано за допомогою технології AJAX, що дозволяє виконувати асинхронне оновлення інформації без перезавантаження сторінки. Це сприяє зменшенню мережевого навантаження та підвищує швидкодію системи моніторингу. У результаті було створено зручну та функціональну панель керування для користувача.

Середній рівень системи реалізовано на базі плати розробки з модулем ESP32-WROOM-32, який підтримує бездротові стандарти Wi-Fi та Bluetooth. Плата оснащена інтерфейсами UART, I2C та SPI, а також вбудованим датчиком Холла, що дозволяє використовувати її як центральний обчислювальний вузол системи. Мікроконтролер забезпечує оброблення сигналів від сенсорів, передачу даних та керування виконавчими механізмами автоматизованої системи [8].

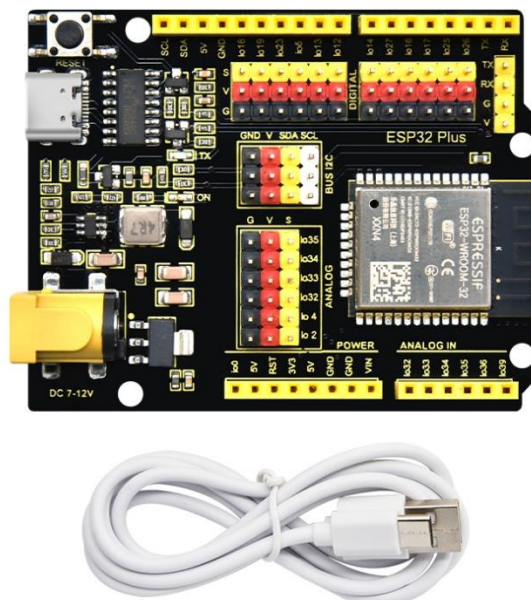


Рис.1.5. Приклад контролера ESP32 PLUS

Компоненти нижнього рівня системи. Основною функцією сенсорної

підсистеми є збір інформації про параметри навколишнього середовища. У системах Internet of Things (IoT) датчики виконують роль базових елементів, які здійснюють перетворення фізичних величин у цифрові сигнали з подальшою передачею даних до мережі безпосередньо або через проміжні модулі оброблення. Оскільки кожна IoT-система має власні функціональні особливості, вибір сенсорних компонентів визначається вимогами конкретного прикладного застосування. Характеристики та особливості використаних технічних засобів нижнього рівня наведено нижче.

Для контролю параметрів навколишнього середовища використовується цифровий модуль, який поєднує ємнісний датчик вологості та термістор для вимірювання температури. Отримані значення передаються через цифровий вивід на мікроконтролер, де надалі використовуються для візуалізації даних або роботи алгоритмів автоматичного керування [9].

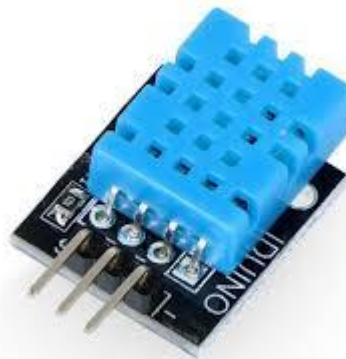


Рис.1.6. Приклад датчика DHT11

Пристрій призначений для безконтактного визначення відстані до об'єктів. Принцип його роботи ґрунтується на випромінюванні ультразвукових хвиль та вимірюванні часу їх повернення після відбиття від перешкоди. На основі отриманих даних система здатна в режимі реального часу визначати наближення або розташування об'єктів [10].



Рис.1.7. Приклад ультразвукового датчика

Аналоговий сенсорний елемент, електричний опір якого змінюється залежно від рівня освітленості поверхні. Модуль перетворює інтенсивність світла на електричний сигнал, що дає змогу автоматично керувати освітлювальними пристроями та іншими компонентами системи [11].



Рис.1.8. Приклад фоторезистора

Модуль призначений для виявлення наявності води або вологи на контактній поверхні сенсора. У разі потрапляння рідини змінюється електропровідність датчика, що перетворюється в аналоговий сигнал та використовується для фіксації опадів або виявлення протікання [12].



Рис.1.9. Приклад датчика пари

Пристрій призначений для визначення рівня рідини за допомогою системи паралельних провідників. Датчик забезпечує моніторинг кількості рідини в резервуарах та передає отримані значення на мікроконтролер для подальшого оброблення [13].

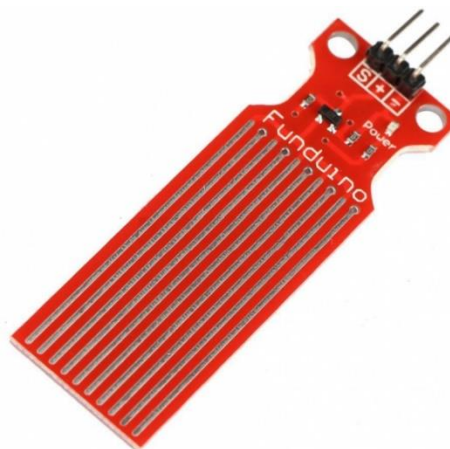


Рис.1.10. Приклад датчика рівня води

Інфрачервоний сенсорний модуль, призначений для виявлення змін теплового випромінювання в зоні контролю. Датчик використовується для фіксації руху людей або тварин та формує логічний сигнал у момент спрацювання [14].

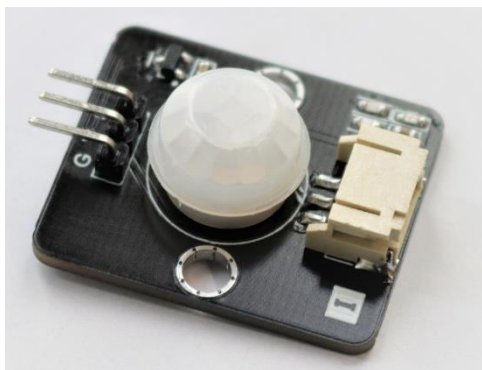


Рис.1.11. Приклад PIR-датчика руху

Виконавчий механізм, який забезпечує точне позиціонування вала на заданий кут повороту. Керування сервоприводом здійснюється за допомогою ШІМ-сигналу, що дозволяє використовувати його для автоматичного переміщення елементів конструкції, зокрема відкривання та закривання засувки [15].



Рис.1.12. Приклад сервопривіда

Електромагнітний комутаційний пристрій, який дозволяє мікроконтролеру з низьковольтним керуванням здійснювати вмикання та вимикання потужних електричних приладів, зокрема водяних насосів, вентиляторів або освітлювальних систем, що працюють від зовнішнього джерела живлення [16].



Рис.1.13. Приклад релейного модуля

Рідкокристалічний дисплейний модуль, призначений для відображення текстової та числової інформації. Використання інтерфейсу I2C дозволяє здійснювати передачу даних лише через два контакти мікроконтролера, забезпечуючи зручне відображення показників сенсорів та системних повідомлень [17].



Рис.1.14. Приклад LCD-дисплей 1602 (I2C)

Електронний звуковий компонент, який генерує аудіосигнал під дією зовнішнього імпульсу заданої частоти. На відміну від активних зумерів, пасивний зумер дозволяє формувати різні тональності та мелодії шляхом зміни частоти керуючого сигналу [18].

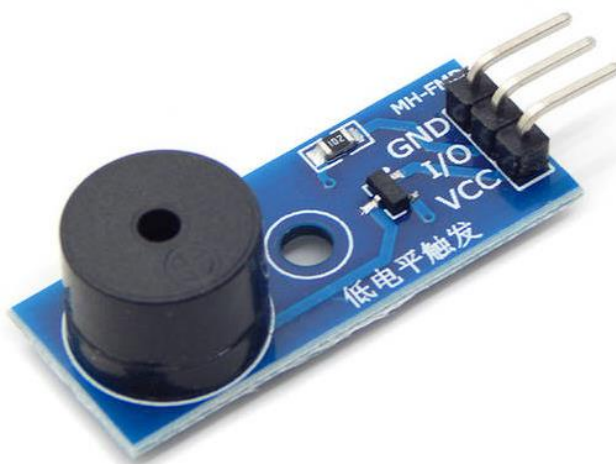


Рис.1.15. Приклад зумера

На відміну від резистивних аналогів, даний модуль визначає рівень вологості за допомогою ємнісного принципу вимірювання, що забезпечує підвищену стійкість до корозії та збільшує термін експлуатації сенсора. Датчик перетворює вміст води в ґрунті на аналоговий сигнал, який використовується для точного контролю стану субстрату [19].

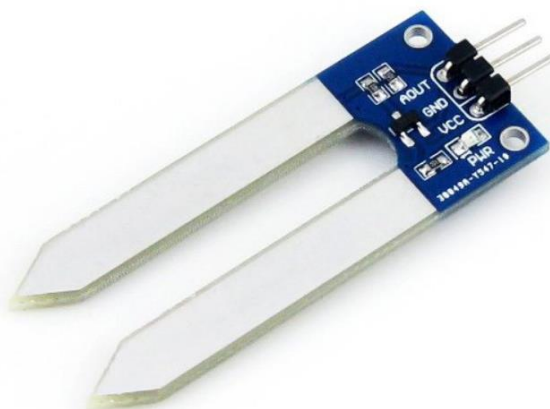


Рис.1.16. Приклад ємнісного датчика вологості ґрунту

Простий елемент введення, який забезпечує замикання електричного кола під час натискання. Кнопка використовується для ручного керування окремими функціями системи, зокрема перемикання режимів роботи або активації виконавчих механізмів [20].



Рис.1.17. Приклад модуля кнопки

Пристрій, призначений для перетворення енергії світлового випромінювання в електричну енергію. У структурі системи сонячна панель може використовуватися як альтернативне джерело живлення або як сенсор для визначення інтенсивності сонячного освітлення.

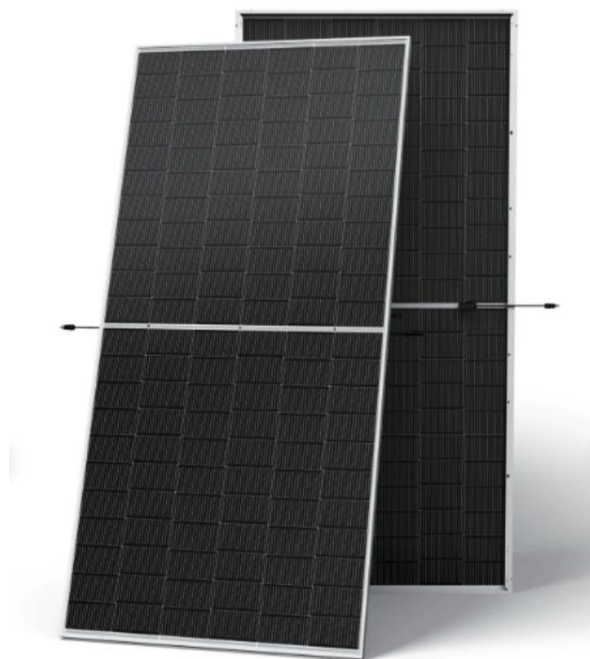


Рис.1.18. Приклад модуля сонячної панелі

Модуль призначений для встановлення елементів живлення, зокрема акумуляторів типу 18650, що забезпечує автономне функціонування системи без постійного підключення до USB-порту або зовнішньої електромережі.



Рис.1.19. Приклад акумуляторного відсіка

Виконавчий пристрій постійного струму, призначений для перекачування рідини між резервуарами. Для керування насосом зазвичай використовується релейний модуль, оскільки споживаний струм перевищує допустимі можливості безпосереднього підключення до мікроконтролера.



Рис.1.20. Приклад водяної помпи

Невеликий електродвигун постійного струму з лопатями, призначений для створення повітряного потоку, рис. 1.21. Використовується для регулювання температури та забезпечення циркуляції повітря у закритому середовищі [21].

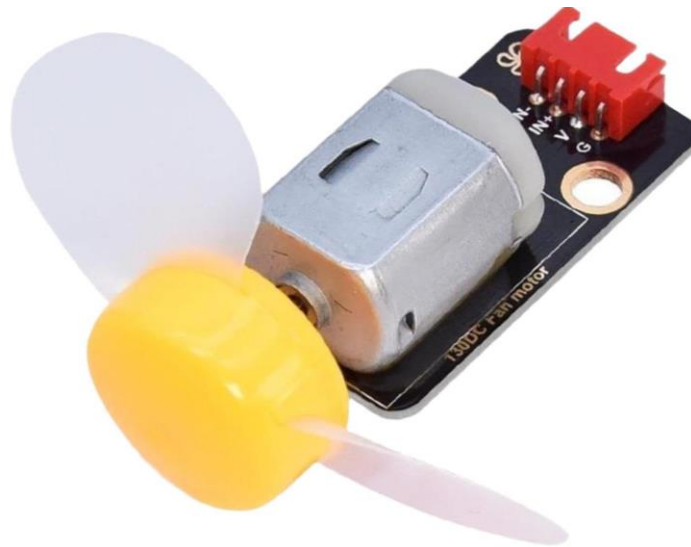


Рис.1.21. Приклад вентилятора

Світлодіод є напівпровідниковим джерелом світла, яке випромінює світло під час подачі електричної напруги. У системі він використовується як засіб візуальної індикації стану або елемент освітлення. Модуль підключається до цифрового виводу мікроконтролера, що забезпечує програмне керування режимами роботи світлодіода, зокрема його вмиканням, вимиканням, створенням ефекту мерехтіння та регулюванням яскравості за допомогою ШІМ [22].



Рис.1.22. Приклад LED-модуля

1.4 Інтегроване середовище розробки Arduino

Для розробки та налагодження програмного забезпечення системи було обрано середовище Arduino IDE, яке є одним із найбільш поширених інструментів для програмування мікроконтролерних платформ. Вибір саме цього середовища зумовлений його простотою використання, підтримкою великої кількості апаратних платформ, зокрема ESP32, а також наявністю значної кількості готових бібліотек для роботи з датчиками, модулями зв'язку та периферійними пристроями. Крім того, Arduino IDE забезпечує швидке завантаження програмного коду, зручне тестування та налагодження роботи пристроїв, що є важливим під час створення IoT-систем та прототипування електронних рішень.

Arduino IDE являє собою інтегроване середовище розробки, призначене для створення, компіляції та запису програмного коду на мікроконтролерні плати сімейства Arduino та сумісні пристрої. Дане середовище об'єднує текстовий редактор, компілятор і засоби взаємодії з мікроконтролером, що дозволяє виконувати повний цикл розробки програм без використання додаткових програмних засобів [23].

Однією з ключових особливостей Arduino IDE є орієнтація на простоту та доступність. Інтерфейс середовища має мінімалістичну структуру, завдяки чому користувач може зосередитися безпосередньо на написанні програмного коду. Для початку роботи достатньо обрати тип плати, визначити COM-порт та завантажити створену програму в пам'ять мікроконтролера. Програмування здійснюється мовою, що базується на C/C++, однак має спрощену структуру, засновану на використанні двох основних функцій — `setup()` та `loop()`. Середовище автоматично формує необхідну структуру проєкту, підключає службові файли та виконує налаштування компілятора, що значно спрощує процес розробки.

Після компіляції вихідний код перетворюється у виконуваний файл, який передається до мікроконтролера через послідовний інтерфейс, зазвичай за

допомогою USB-з'єднання та вбудованого завантажувача. Це дозволяє швидко тестувати роботу системи та вносити необхідні зміни до програмного забезпечення.

Arduino IDE підтримує широкий спектр мікроконтролерних платформ, серед яких Arduino, ESP32, ESP8266, AVR та ARM-сумісні пристрої. Налаштування типу плати здійснюється через меню Tools → Board, після чого середовище автоматично застосовує необхідні параметри компіляції відповідно до обраної апаратної архітектури.

Для розширення функціональних можливостей програм у середовищі передбачена підтримка бібліотек, які спрощують роботу з датчиками, дисплеями, мережевими модулями та іншими компонентами. Встановлення бібліотек може виконуватися як вручну, так і через вбудований менеджер пакетів, що автоматизує процес завантаження та оновлення необхідних модулів. Також середовище підтримує роботу з ядрами платформ, які визначають особливості компіляції та взаємодії з різними типами мікроконтролерів. Це забезпечує універсальність Arduino IDE та дозволяє використовувати його для створення широкого спектра вбудованих та IoT-систем.

Приклад зображення інтерфейсу середовища розробки Arduino IDE під час створення та компіляції програмного коду представлено на рис.1.23.



Рис.1.23. Інтерфейс середовища розробки Arduino IDE

На рис.1.24 представлено екосистему Arduino IDE та взаємозв'язки середовища розробки з програмними платформами, інструментами, спільнотами користувачів і технологічними партнерами. Схема демонструє підтримку різних операційних систем, використання мов програмування та додаткових засобів розробки, а також інтеграцію з бібліотеками, системами тестування й безперервної інтеграції. Окремо відображено відкритий характер платформи Arduino, її активну спільноту розробників і сумісність із популярними апаратними рішеннями [23].

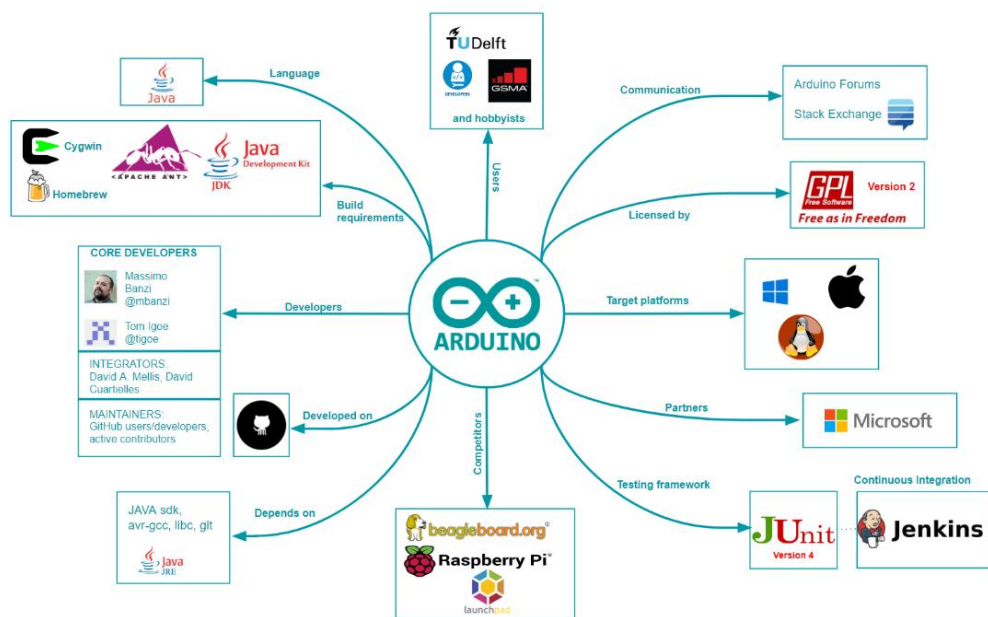


Рис.1.24. Контекстний вигляд середовища розробки Arduino IDE, що описує зв'язки з його оточенням

2 НАЛАШТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ «РОЗУМНА ФЕРМА»

У даному розділі розглянуто процес розробки та налаштування апаратно-програмного комплексу «Розумна ферма» на основі технологій Інтернету речей. Основну увагу приділено конфігурації апаратних компонентів, програмуванню мікроконтролера ESP32 та реалізації систем автоматизованого моніторингу й керування параметрами фермерського середовища.

У процесі розробки було виконано підключення та налаштування датчиків температури, вологості ґрунту, рівня води, освітлення та руху. Також реалізовано системи автоматичного поливу, контролю температури, освітлення та сигналізації. Для забезпечення дистанційного доступу до системи використано технології бездротового зв'язку Wi-Fi, вебінтерфейс керування та мобільний застосунок.

Розроблений апаратно-програмний комплекс забезпечує можливість централізованого контролю та автоматизації основних процесів у системі «розумної ферми», що дозволяє підвищити ефективність моніторингу стану середовища та оптимізувати керування обладнанням.

2.1 Система автоматичного керування освітленням на основі фоторезистора

Система вимірювання освітленості на основі фоторезистора. Для налаштування та перевірки роботи фоторезистора у середовищі Arduino IDE було створено програмний код Photocell-sensor, який забезпечує зчитування значень освітленості з датчика та передачу отриманих даних до мікроконтролера ESP32. Даний програмний код використовується для аналізу зміни рівня освітлення навколишнього середовища в режимі реального часу, даний код представлено на рис. 2.1

```

#define PhotocecllPin 34 //Define the photoresistor pin

void setup() {
  //Initialize the serial port
  Serial.begin(9600);
  //Set the pin to input mode
  pinMode(PhotocecllPin, INPUT);
}

void loop() {
  //Read the value of photoresistor
  int ReadValue = analogRead(PhotocecllPin);
  //Print the value. NOTE: ESP32 board is 12-bit ADC, whose detection value range is within 0~4095.
  Serial.print("Photocecll value: ");
  Serial.println(ReadValue);
  delay(500);
}

```

Рис.2.1 Програмний код Photocell-sensor

У середовищі Arduino IDE було обрано плату ESP32 Dev Module та відповідний COM-порт, після чого виконано компіляцію і завантаження програмного коду до мікроконтролера ESP32 для подальшого тестування роботи датчика освітлення, рис.2.2.



Рис. 2.2 Налаштування плати ESP32 та COM-порту в Arduino IDE

Для перевірки роботи фоторезистора відкриваємо вікно Serial Monitor у середовищі Arduino IDE, де в режимі реального часу відображаються значення, отримані з датчика освітлення.

У процесі тестування встановлено, що зі збільшенням інтенсивності освітлення значення, яке зчитується фоторезистором та відображається у моніторі порту, також збільшується. Це підтверджує коректну роботу датчика та можливість його використання для контролю рівня освітленості в системі «розумної ферми».

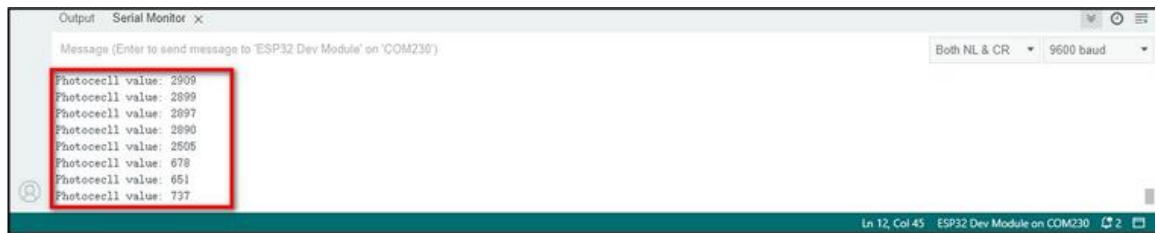


Рис. 2.3 Результат зчитування даних із фоторезистора

Фоторезисторний модуль використовується для визначення рівня освітленості шляхом перетворення світлового впливу на електричний сигнал. Робота датчика базується на властивості змінювати електричний опір залежно від кількості світла, що потрапляє на його поверхню.

При збільшенні інтенсивності освітлення опір фоторезистора поступово зменшується, унаслідок чого змінюється величина вихідної напруги. Отримане значення передається на мікроконтролер ESP32 для подальшої обробки та аналізу. Такий принцип роботи дозволяє використовувати фоторезистор у системах автоматичного керування освітленням і моніторингу умов навколишнього середовища.

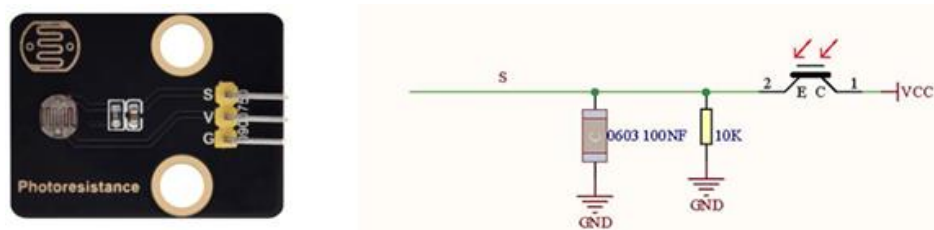


Рис.2.4 Зовнішній вигляд та схема підключення модуля фоторезистора

Система керування освітленням. Для налаштування системи автоматичного керування освітленням у середовищі Arduino IDE було відкрито програмний код Light-Control-System. Даний код призначений для зчитування значень із фоторезистора та керування світлодіодним модулем залежно від рівня освітленості навколишнього середовища, рис.2.5.

```

#define PhotocecllPin 34 //Define the photoresistor pin
#define LED           27 //Define LED pin

void setup() {
  //Initialize serial port
  Serial.begin(9600);
  //Set the photoresistor pin to input mode
  pinMode(PhotocecllPin,INPUT);
  //Set the LED pin to output mode
  pinMode(LED,OUTPUT);
}

void loop() {
  //Read the value of the photoresistor
  int ReadValue = analogRead(PhotocecllPin);

```

Рис.2.5 Програмний код системи автоматичного керування освітленням

Наступним кроком було обрано плату ESP32 Dev Module та відповідний COM-порт, після чого виконано компіляцію й завантаження програмного коду на плату ESP32. Це дозволило перейти до перевірки роботи системи автоматичного керування освітленням, рис.2.6.



Рис.2.6 Вибір плати та COM-порту в середовищі

У процесі тестування системи автоматичного керування освітленням було перевірено взаємодію фоторезистора та світлодіодного модуля. Під час роботи система постійно зчитувала значення рівня освітленості та передавала їх до мікроконтролера ESP32 для подальшої обробки.

Встановлено, що при перевищенні значення освітленості понад 800, що відповідає умовам достатнього денного освітлення, світлодіод автоматично вимикався. У випадку зменшення значення нижче 800 система вмикала світлодіод, забезпечуючи автоматичне керування освітленням залежно від умов навколишнього середовища.

Такий підхід дозволяє реалізувати енергоефективний режим роботи системи освітлення в складі комплексу «розумна ферма».

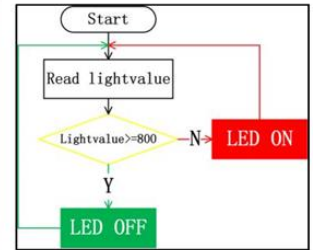
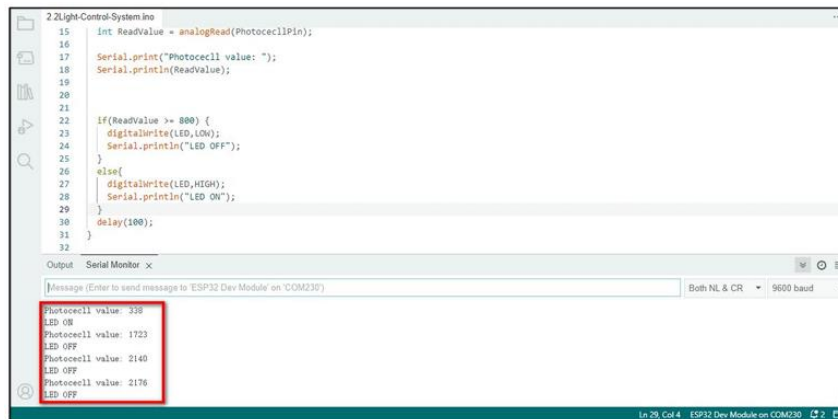


Рис.2.7 Результат роботи системи автоматичного керування освітленням на основі фоторезистора

2.2 Система контролю температури

Датчик температури та вологості DHT11. Датчик температури та вологості DHT11 формуює цифрові сигнали на виході. У його роботі використовуються принципи зчитування та перетворення аналогових сигналів, а також технології вимірювання температури й вологості, що забезпечує високу надійність та довготривалу стабільність роботи.

Крім того, датчик містить високоточний резистивний сенсор вологості та терморезистивний датчик температури, які підключені до високопродуктивного 8-бітного мікроконтролера, рис.2.8.

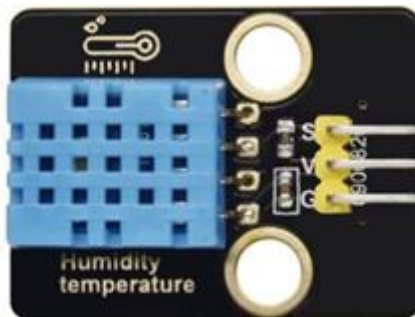


Рис.2.8 Модуль датчика температури та вологості DHT11

Наступним кроком у середовищі Arduino IDE було розроблено програмний код Intelligent-Feeding-System для налаштування та тестування системи контролю температури та вологості на основі датчика DHT11, рис. 2.9.

```
#include <dht11.h>
#define DHT11PIN 17

dht11 DHT11;

void setup()
{
  Serial.begin(9600);
  Serial.println("DHT11 TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT11LIB_VERSION);
  Serial.println();
}

void loop()
{
  Serial.println("\n");
```

Рис. 2.9 Програмний код Intelligent-Feeding-System

Далі обрано плату ESP32 Dev Module та відповідний COM-порт, після чого здійснено компіляцію і завантаження програмного коду до мікроконтролера ESP32 для подальшого налаштування та перевірки роботи системи, рис.2.10.



Рис.2.10 Вибір COM-порту для програмування ESP32

Для перевірки роботи датчика було відкрито вікно Serial Monitor та встановлено швидкість передачі даних 9600 бод. У процесі роботи системи в моніторі порту в режимі реального часу відображались поточні значення температури та вологості повітря, які надходили з датчика.

Отримані результати підтвердили коректне зчитування та передачу даних до мікроконтролера ESP32, що свідчить про стабільну роботу системи контролю температури та вологості в складі комплексу «розумна ферма».

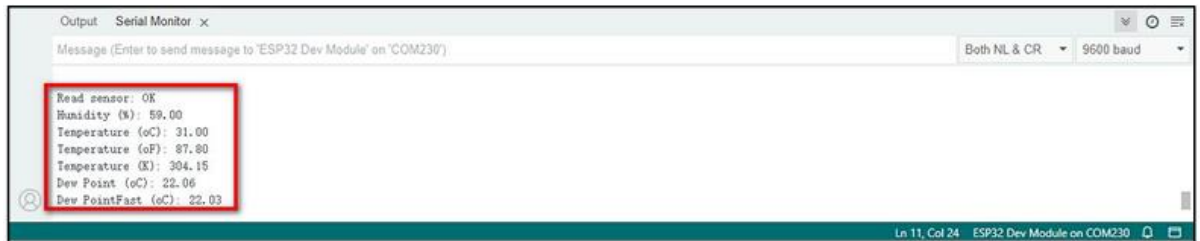
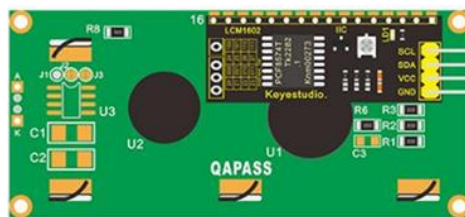


Рис.2.11 Відображення показників температури та вологості в Serial Monitor

Модуль LCD1602 являє собою символічний рідкокристалічний дисплей, оснащений стандартним 14-контактним інтерфейсом без підсвічування або 16-контактним інтерфейсом із підсвічуванням. Використання спеціального драйвера ІС із підтримкою інтерфейсу I2C дає змогу значно скоротити кількість виводів мікроконтролера, необхідних для підключення та керування дисплеєм.



Specifications

Power supply range: 5V
 Working current: <130mA
 Recommended ambient temperature: -10°C ~ 50°C
 Product size: 80mm * 36mm
 IIC address: 0x27

Рис.2.12 Зовнішній вигляд та технічні характеристики модуля LCD1602 з інтерфейсом I2C

Наступним етапом було створено та відкрито програмний код LCD1602 для налаштування й тестування роботи LCD-дисплея.

```
#include <LiquidCrystal_I2C.h>

//Initialize LCD 1602, 0x27 is I2C address
LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  //Initialize LCD
  lcd.init();
  // Turn the (optional) backlight off/on
  lcd.backlight();
  //lcd.noBacklight();

  //Set the position of cursor
  lcd.setCursor(0, 0);
  //LCD prints
  lcd.print("HELLO WORLD 0");
  lcd.setCursor(0, 1);
```

Рис. 2.13 Програмний код LCD1602

Було обрано плату ESP32 Dev Module та відповідний COM-порт, після чого програмний код завантажено до мікроконтролера.



Рис.2.14 Вибір плати ESP32 Dev Module та COM-порту

У процесі роботи модуля LCD1602 було активовано підсвічування дисплея, після чого на екрані відобразилися повідомлення «HELLO WORLD 0» та «HELLO WORLD 1», рис.2.15.

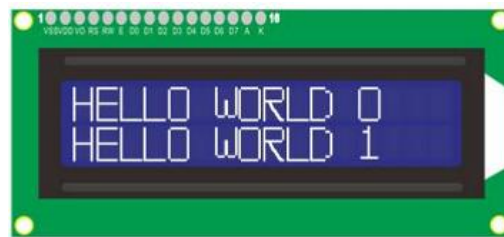


Рис.2.15 Результат роботи LCD1602 із відображенням текстових повідомлень

Двигун та вентилятор. Двигун типу 130 підтримує можливість регулювання швидкості обертання за допомогою PWM-сигналу. Для забезпечення керування роботою двигуна необхідно підключити два керувальні контакти.

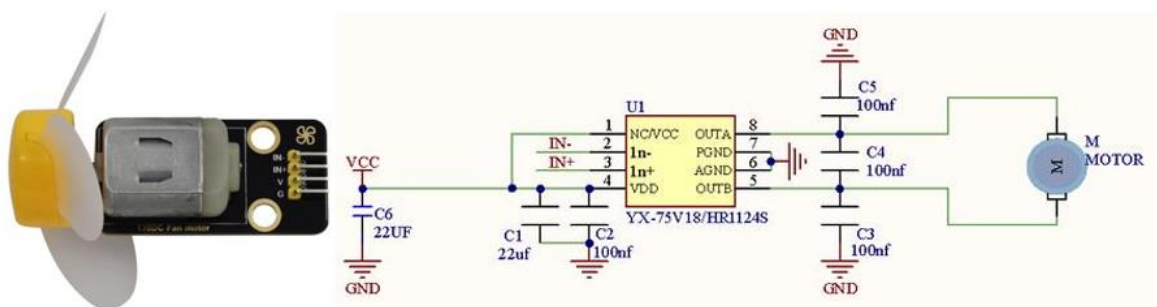


Рис.2.16 Зовнішній вигляд двигуна з вентилятором зі схемою його підключення

Для налаштування та тестування роботи двигуна у середовищі Arduino IDE було створено програмний код Motor, рис.2.17.

```
#define MotorPin1 19//(IN+)
#define MotorPin2 18//(IN-)

void setup() {
  pinMode(MotorPin1,OUTPUT);
  pinMode(MotorPin2,OUTPUT);

  // Set PWM output to adjust the speed of motor
  ledcSetup(1, 1200, 8); // Set frequency of LEDC Channel 1 to 1200, PWM resolution to 8, so duty cycle
  ledcAttachPin(MotorPin1, 1); // Bound LEDC Channel 1 to the specified left motor pin gpio19 to output
  ledcSetup(3, 1200, 8); // Set frequency of LEDC Channel 3 to 1200, PWM resolution to 8, so duty cycle
  ledcAttachPin(MotorPin2, 3); // Bound LEDC Channel 3 to the specified left motor pin gpio18 to output
}

void loop() {
  //Turn left
  ledcWrite(1, 70);
}
```

Рис.2.17 Програмний код Motor

Далі в середовищі було обрано плату ESP32 Dev Module та відповідний COM-порт, після чого програмний код завантажено до мікроконтролера, рис.2.18.



Рис.2.18 Вибір плати ESP32 Dev Module та COM-порту для завантаження програмного коду

У процесі роботи системи двигун типу 130 почергово обертався ліворуч і праворуч із інтервалом 2 секунди.

Оскільки вентилятор належить до потужних електронних компонентів, для його стабільного живлення рекомендовано використовувати батареї.

Система контролю температури.

Для налаштування та тестування системи контролю температури в середовищі Arduino IDE було розроблено та відкрито програмний код Temperature-Control-System, за допомогою якого реалізовано зчитування температурних

показників, обробку отриманих даних і керування роботою системи. Програмний код наведено на рис. 2.19.

```
#include <LiquidCrystal_I2C.h>
#include <dht11.h>

#define DHT11PIN 17
#define MotorPin1 19//(IN+)
#define MotorPin2 18//(IN-)

dht11 DHT11;

LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  lcd.init();
  lcd.backlight();

  pinMode(MotorPin1,OUTPUT);
  pinMode(MotorPin2,OUTPUT);

  //Set PWM output to adjust the speed of motor
  ledcSetup(1, 1200, 8);//Set frequency of LEDC Channel 1 to 1200, PWM resolution to 8, so duty cycle =
  ledcAttachPin(MotorPin1, 1); //Bound LEDC Channel 1 to the specified left motor pin gpio19 to output
```

Рис.2.19 Програмний код Temperature-Control-System

У середовищі Arduino IDE було обрано плату ESP32 Dev Module та відповідний COM-порт, після чого виконано завантаження програмного коду до мікроконтролера.



Рис.2.20 Вибір плати ESP32 Dev Module та COM-порту

У процесі роботи системи при досягненні температури 29 °C вентилятор автоматично активувався для відведення надлишкового тепла. У разі зниження температури нижче встановленого порогового значення вентилятор вимикався. Такий алгоритм роботи забезпечує імітацію процесу автоматичного охолодження та сприяє підвищенню енергоефективності фермерської системи.

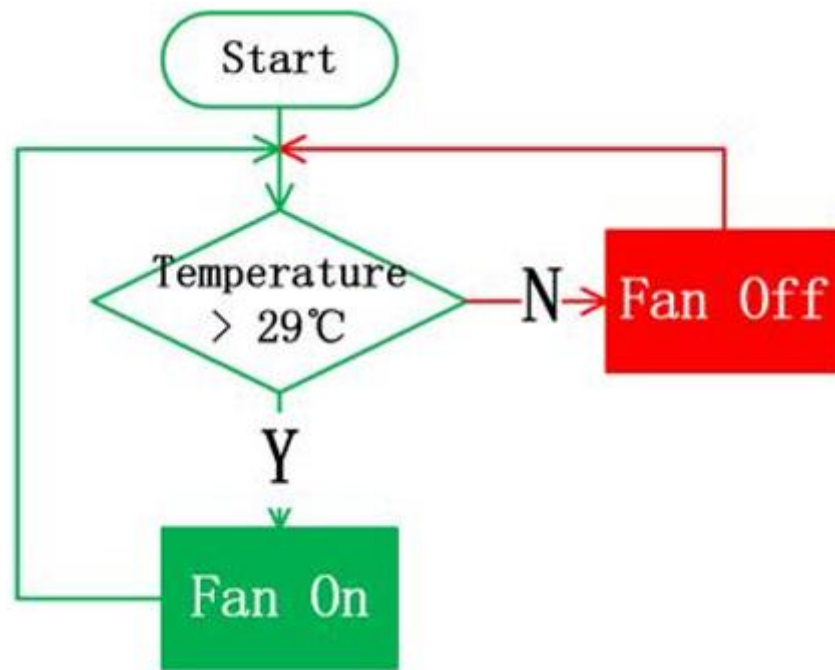


Рис. 2.21 Алгоритм автоматичного керування вентилятором залежно від температури

2.3 Система моніторингу вологості ґрунту

Датчик вологості ґрунту. Датчик вологості ґрунту призначений для визначення рівня вологи в ґрунті шляхом вимірювання його електропровідності. Принцип роботи датчика базується на зміні електричного опору між двома металевими щупами залежно від кількості вологи в ґрунті. Отримані дані можуть використовуватись для автоматичного керування системами поливу та моніторингу стану ґрунту в системах розумного фермерства.



Рис.2.22 Зовнішній вигляд датчика вологості ґрунту

Для налаштування та перевірки роботи датчика вологості ґрунту було створено та реалізовано програмний код Soil-Humidity-Sensor. Програмний код забезпечує зчитування значень вологості, обробку отриманих даних і передачу результатів для подальшого моніторингу стану ґрунту, рис.2.23.

```
#define SoilHumidityPin 32

void setup() {
  Serial.begin(9600);
  pinMode(SoilHumidityPin,INPUT);
}

void loop() {
  //Define a variab as the value of soil humidity sensor
  int ReadValue = analogRead(SoilHumidityPin);
  Serial.println(ReadValue);
  delay(500);
}
```

Рис.2.23 Програмний код Soil-Humidity-Sensor

Наступним кроком обрано плату ESP32 Dev Module та відповідний COM-порт, після чого виконано завантаження програмного коду до мікроконтролера, рис. 2.24.



Рис.2.24 Відображення значень датчика вологості ґрунту в Serial Monitor

Датчики вологості ґрунту переважно використовуються для вимірювання вмісту води в ґрунті, моніторингу рівня вологості, автоматизації зрошення сільськогосподарських культур та захисту лісових насаджень.

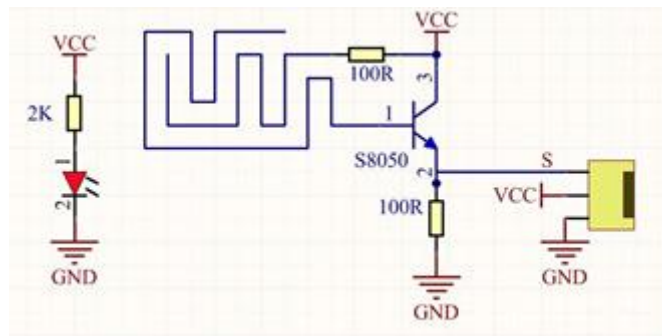


Рис.2.25 Електрична схема модуля вологості ґрунту

Система моніторингу вологості ґрунту.

Для відображення поточного рівня вологості ґрунту в режимі реального часу в системі використовується дисплей LCD1602, на якому відображаються значення, отримані з датчика вологості. У разі зниження рівня вологості нижче встановленого порогового значення система автоматично активує буюер, який подає звуковий сигнал про необхідність поливу ґрунту.

Для налаштування та перевірки роботи системи моніторингу вологості ґрунту було створено програмний код Soil-Humidity-Testing-System. Даний код забезпечує зчитування даних із датчика вологості ґрунту, їх обробку мікроконтролером ESP32 та відображення результатів на LCD-дисплеї, рис.2.26.

```
#include <LiquidCrystal_I2C.h>

#define BuzzerPin 16
#define SoilHumidityPin 32

LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {

    pinMode(BuzzerPin,OUTPUT);
    pinMode(SoilHumidityPin,INPUT);

    lcd.init();
    lcd.backlight();
    lcd.clear();
}
```

Рис.2.26 Програмний код Soil-Humidity-Testing-System

Далі обрано плату ESP32 Dev Module та відповідний COM-порт, після чого виконано завантаження програмного коду до мікроконтролера, рис.2.27.



Рис.2.27 Плату ESP32 Dev Module та відповідний COM-порт

У процесі тестування системи моніторингу вологості ґрунту на дисплеї LCD1602 у режимі реального часу відображались поточні значення вологості, отримані з датчика. Мікроконтролер ESP32 постійно аналізував показники сенсора та виконував контроль встановленого порогового значення.

Встановлено, що при зменшенні значення вологості ґрунту нижче 200 система автоматично активувала буюер, який подавав звуковий сигнал для сповіщення про необхідність поливу. Такий підхід дозволяє своєчасно контролювати стан ґрунту та забезпечує автоматизований моніторинг у системі «розумна ферма».

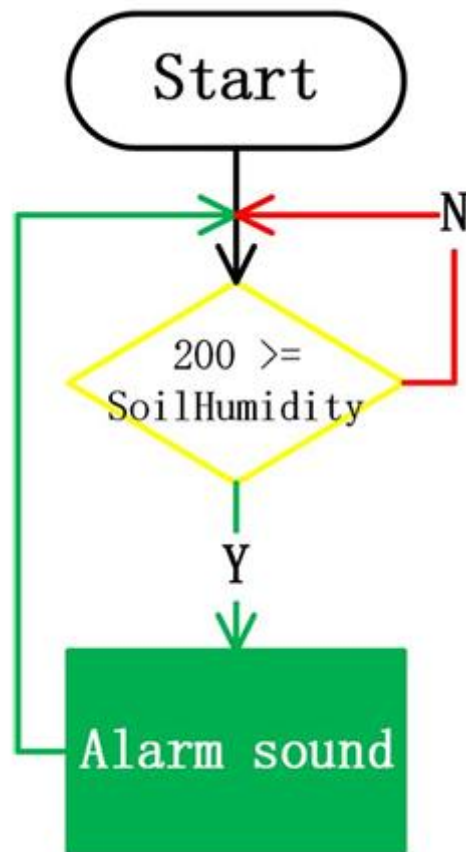


Рис.2.28 Блок-схема алгоритму роботи системи моніторингу вологості ґрунту

2.4 Система автоматичного зрошення

Система подачі води. У даній системі плата ESP32 використовується для керування процесом увімкнення та вимкнення водяного насоса за допомогою релейного модуля. Водяний насос забезпечує подачу та перекачування води в системі автоматичного зрошення й використовується для підтримання необхідного рівня вологості ґрунту.

Для комутації живлення насоса застосовується релейний модуль, який виконує функцію електронного перемикача та дозволяє керувати навантаженнями з більшими значеннями струму й напруги за допомогою сигналів від мікроконтролера ESP32.

Релейний модуль.

Релейний модуль широко використовується в системах автоматизації для керування потужними електричними пристроями, зокрема електродвигунами, насосами, освітлювальними приладами та іншими виконавчими механізмами. Використання реле дозволяє ізолювати низьковольтну частину мікроконтролера від силового кола навантаження, забезпечуючи безпечну та стабільну роботу системи, рис.2.29.



Рис.2.29 Зовнішній вигляд релейного модуля

Основні контакти релейного модуля.

Релейний модуль містить три основні контакти для підключення зовнішнього електричного кола:

- NO (Normally Open) — нормально розімкнений контакт;
- COM (Common Contact) — загальний контакт;
- NC (Normally Closed) — нормально замкнений контакт.

Керування процесом замикання та розмикання контактів релейного модуля виконується мікроконтролером ESP32, що забезпечує автоматизоване керування роботою водяного насоса в системі подачі води, сама ж схема представлена на рис.2.30.

Water Pump:



Parameters:

- Power voltage: 5V
- Static current: 2mA
- Maximum contact voltage: 250VAC/30VDC
- Maximum current: 10A

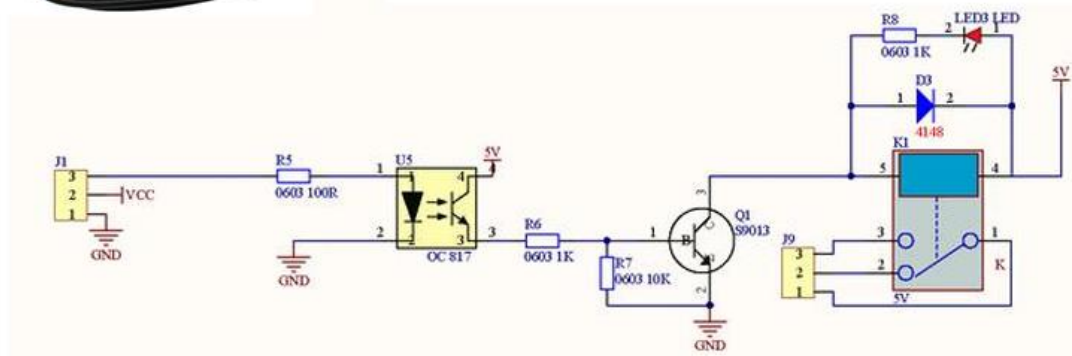


Рис. 2.30 Водяний насос та схема його підключення через релейний модуль

Для налаштування та тестування роботи системи подачі води, наступним кроком було створено програмний код Water-Pump.

```
#define RelayPin 25

void setup() {
  Serial.begin(9600);
  pinMode(RelayPin, OUTPUT);
}

void loop() {
  //Serial.read() receives one byte once. For example, when input "aaa", it receives one "a" at a time
  if(Serial.available() > 0) {
    if (Serial.read() == 'a') //When the input value equals to "a", irrigation begins.
    {
      digitalWrite(RelayPin, HIGH);
      delay(400); //irrigation delay
      digitalWrite(RelayPin, LOW);
      delay(700);
    }
  }
}
```

Рис.2.31 Програмний код Water-Pump

Потім обрано плату ESP32 Dev Module та відповідний COM-порт, після чого виконано завантаження програмного коду до мікроконтролера.



Рис.2.32 Плат ESP32 Dev Module та відповідний COM-порт

Для перевірки роботи водяного насоса через монітор порту надсилався символ «а», після чого мікроконтролер ESP32 передавав сигнал на релейний модуль для активації насоса.

У результаті подачі команди насос одноразово виконував подачу води, що підтвердило коректну роботу системи керування та взаємодію між мікроконтролером, реле і виконавчим механізмом.

Система автоматичного зрошення.

У даній системі до плати ESP32 підключено два датчики, показники яких використовуються для автоматизованого керування процесом поливу. Програмний код забезпечує безперервне зчитування значень вологості ґрунту та рівня води, після чого отримані дані аналізуються мікроконтролером для прийняття рішення щодо активації системи зрошення.

У випадку зниження вологості ґрунту нижче встановленого порогового значення релейний модуль автоматично вмикає водяний насос, який забезпечує подачу води для поливу рослин. Якщо датчик рівня води фіксує недостатню кількість води, система блокує роботу насоса для запобігання його роботі без води, а буюер подає звуковий сигнал попередження. Для налаштування та перевірки роботи системи автоматичного зрошення у середовищі Arduino IDE було створено програмний код Auto-irrigation, який реалізує алгоритм автоматичного контролю вологості ґрунту та керування процесом подачі води.

```
#include <LiquidCrystal_I2C.h>

#define BuzzerPin 16
#define SoilHumidityPin 32
#define WaterLevelPin 33
#define RelayPin 25
#define ButtonPin 5 //Define a button pin
int value = 0;      //Set an initial button value

LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
    //Set the pins mode
    pinMode(BuzzerPin,OUTPUT);
    pinMode(SoilHumidityPin,INPUT);
```

Рис.2.33 Програмний код Auto-irrigation

Далі було обрано плату ESP32 Dev Module та відповідний COM-порт, після чого виконано завантаження програмного коду до мікроконтролера.



Рис.2.34 Плат ESP32 Dev Module та відповідний COM-порт

У процесі роботи системи дисплей LCD1602 у режимі реального часу відображав поточні значення вологості ґрунту та рівня води, отримані з відповідних датчиків. Це забезпечує постійний моніторинг основних параметрів системи автоматичного зрошення та дозволяло контролювати стан ґрунту й рівень води під час роботи комплексу «розумна ферма».



Рис. 2.35 Відображення значень вологості ґрунту та рівня води на дисплеї LCD1602

Якщо зафіксоване значення вологості ґрунту ставало меншим за 500, буюер подавав звуковий сигнал, повідомляючи про пересушений ґрунт. Якщо значення рівня води перевищувало 1000, система автоматично запускала процес зрошення.

У випадку, коли значення рівня води ставало меншим за 500, система подачі води припиняла роботу, а буюер подавав сигнал про недостатню кількість води. Для вимкнення звукового сигналу необхідно натиснути кнопку.

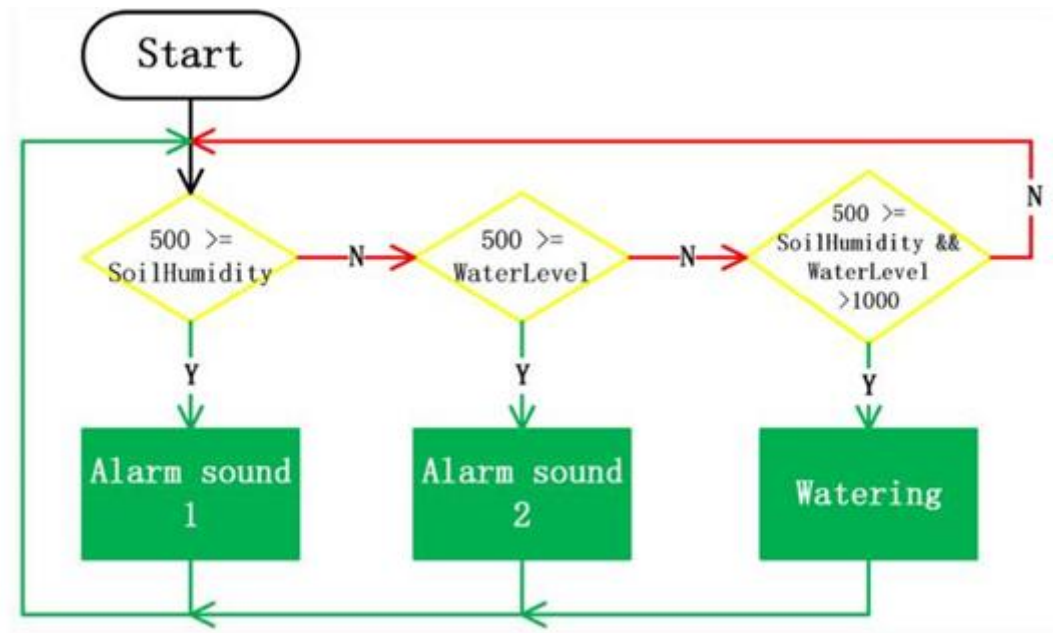


Рис. 2.36 Блок-схема контролю вологості ґрунту та рівня води

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ ІОТ-СИСТЕМИ «РОЗУМНА ФЕРМА» З ВЕБОРІЄНТОВАНИМ КЕРУВАННЯМ

3.1 Реалізація системи автоматичного поливу на основі ІоТ-технологій

Розумний полив є перспективним рішенням для управління водними ресурсами в точному землеробстві. Ця система підвищує ефективність використання води та врожайність, інтегруючи дані про ґрунт та клімат у режимі реального часу, сприяючи стійкості сільськогосподарських практик через зростаючі проблеми дефіциту води та наслідки зміни клімату. Завдяки поєднанню сенсорних мереж та концепції Інтернету речей, традиційне сільське господарство трансформується у цифрову галузь, де управління ресурсами базується на об'єктивних даних, а не на суб'єктивних оцінках.

Неправильні методи зрошення можуть призвести до надмірного використання води, зростання витрат та зниження якості та врожайності врожаю. Тому оптимізація способу поливу з використанням певних прогнозів на основі даних має вирішальне значення як для збереження води, так і для максимізації сільськогосподарського виробництва. Ці параметри відіграють важливу роль у підтримці фермерів у веденні сільського господарства, прийнятті правильних рішень щодо того, скільки часу потрібно зрошувати ґрунт, удобрювати, боротися зі шкідниками та приймати багато інших важливих дій та рішень.

Головною метою дослідження є оцінка ефективності системи моніторингу та управління зрошенням на основі Інтернету речей, а також технології збору даних для сільськогосподарського застосування. Основними результатами даного дослідження є:

1. Створення цілісного комплексу на базі технологій Інтернету речей, що забезпечує стабільний зв'язок між датчиками та виконавчими пристроями в режимі реального часу.

2. Використання мікроконтролера ESP32 як центрального вузла збору даних, що забезпечує стабільну роботу при мінімальному енергоспоживанні.

3. Розробка коду з відкритим доступом для автоматичного збору даних та керування обладнанням.

4. Створення сторінки керування, яка автоматично оновлює показники датчиків на екрані користувача без необхідності перезавантажувати сторінку.

5. Визначення та впровадження точних порогів вологості (30%) та освітлення, що дозволяє економити воду та підтримувати ідеальний мікроклімат.

Налаштування та програмна реалізація розумного поливу.

Технічна архітектура розробленої системи реалізована за модульним принципом, у якому центральним елементом керування виступає мікроконтролер ESP32. Вибір даної платформи обумовлений її високою продуктивністю, підтримкою бездротових інтерфейсів Wi-Fi та Bluetooth, а також можливістю роботи з аналоговими й цифровими сигналами. Налаштування апаратної частини системи розпочиналося з конфігурації портів введення-виведення для забезпечення коректної взаємодії контролера з сенсорами та виконавчими пристроями.

Для контролю параметрів навколишнього середовища до ESP32 було підключено комплекс сенсорних модулів. Одним із основних елементів системи став датчик вологості ґрунту, для якого проведено попереднє калібрування з визначенням мінімальних та максимальних значень вологості. Це дозволило реалізувати програмне перетворення аналогового сигналу у відсоткове значення вологості ґрунту та забезпечити точність автоматичного керування системою поливу. Додатково в систему інтегровано цифровий датчик температури та вологості повітря для моніторингу параметрів мікроклімату в зоні вирощування рослин.

Під час проєктування особливу увагу приділено підключенню виконавчих механізмів, зокрема водяного насоса та вентилятора. Оскільки мікроконтролер працює з низьковольтними сигналами, безпосереднє підключення потужних пристроїв є неможливим. Для узгодження силових і логічних кіл було використано

релейний модуль, який забезпечує електричну ізоляцію та надійне керування навантаженням без ризику пошкодження контролера під час пускових струмів.

Завершальним етапом стало налаштування системи живлення та засобів локального відображення інформації. Живлення системи реалізовано із розділенням логічної та силової частин, що підвищує стабільність роботи комплексу. Для візуалізації поточних параметрів використано рідкокристалічний дисплей, підключений через інтерфейс I2C. Реалізоване поєднання апаратних компонентів дозволило створити стабільну систему «Розумного поливу», здатну працювати в режимі реального часу та підтримувати необхідні умови для росту рослин.

На рис. 3.1 наведено схему підключення компонентів системи «Розумного поливу».

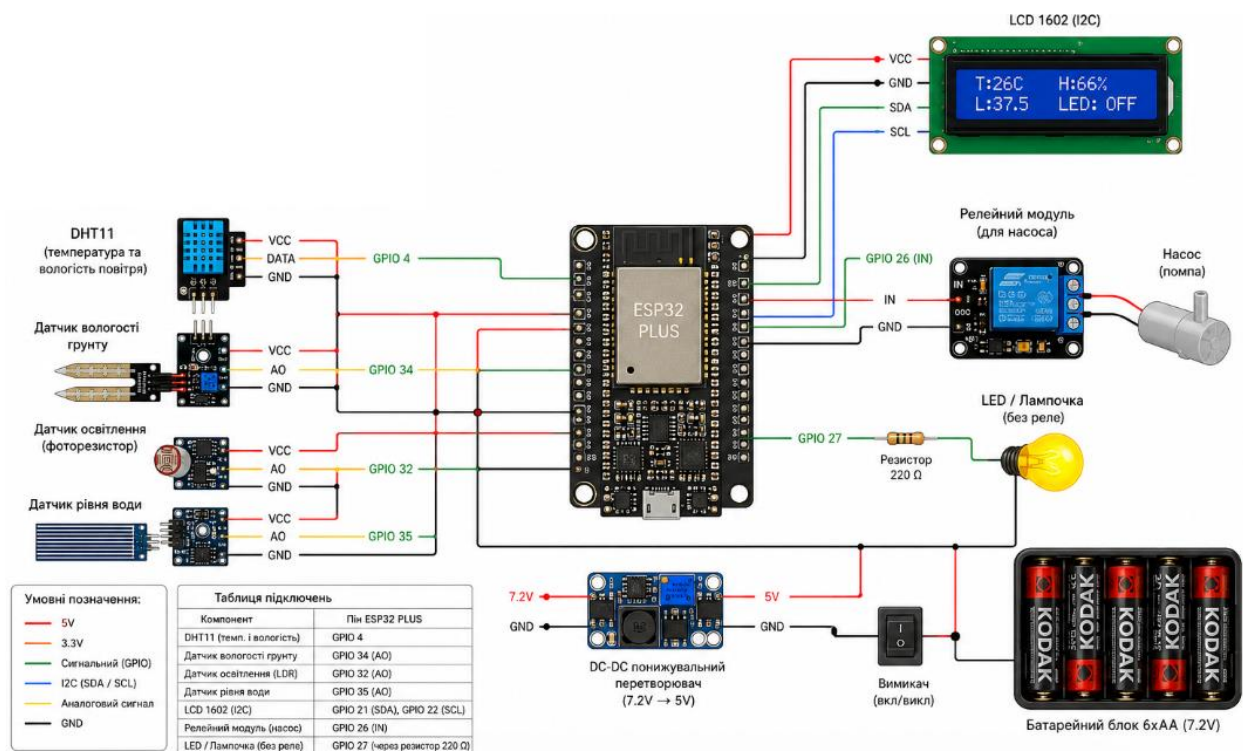


Рис.3.1. Схема підключення компонентів системи «Розумного поливу»

Сам проект автоматичного поливу рослин на базі ESP32 PLUS, представлено на рис.3.2.

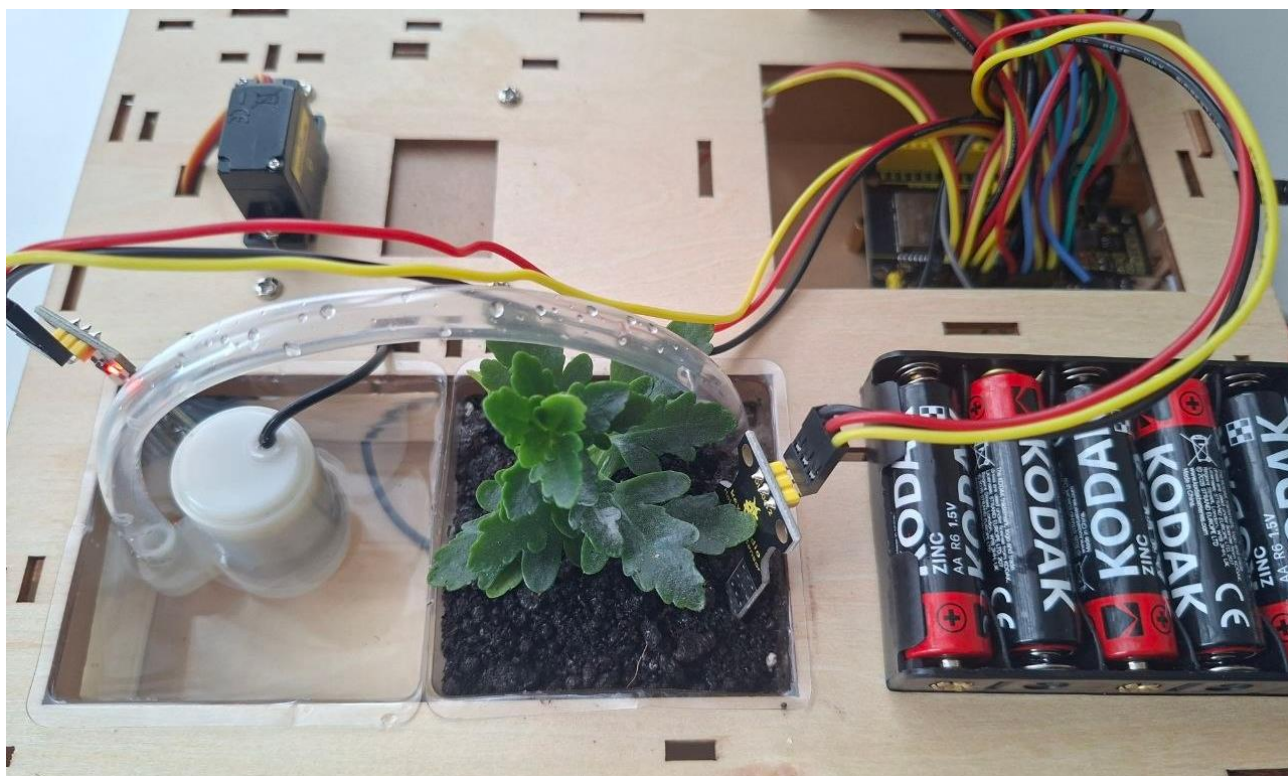


Рис.3.2. Підсистема автоматичного поливу рослин на базі ESP32 PLUS

Програмний алгоритм автоматизованого поливу реалізований за принципом безперервного моніторингу рівня вологості ґрунту в режимі реального часу. На початковому етапі система зчитує аналоговий сигнал із датчика вологості, після чого отримані значення програмно перетворюються у відсоткову шкалу, де 0 % відповідає сухому ґрунту, а 100 % — максимальному рівню зволоження.

Основою логіки керування є порівняння поточного значення вологості із заданим порогом у 30 %. Якщо показник стає нижчим за встановлене значення, мікроконтролер формує керуючий сигнал на релейний модуль для активації водяного насоса. Для підвищення надійності роботи системи в алгоритм інтегровано додаткову перевірку ($\text{soilPercent} > 2$), яка унеможливорює запуск насоса у випадку несправності або відключення датчика, запобігаючи аварійним ситуаціям та надмірному зрошенню.

Подача води здійснюється короткими інтервалами тривалістю 2 секунди, після чого виконується програмна затримка для рівномірного розподілу вологи в ґрунті та стабілізації показників сенсора. У процесі роботи всі дані автоматично

передаються на вебінтерфейс за допомогою протоколу SSE, що забезпечує можливість дистанційного моніторингу параметрів системи та контролю використання водних ресурсів у режимі реального часу.

```
int soilRaw = analogRead(SOIL_PIN);
int soilPercent = constrain(map(soilRaw, 4095, 2100, 0, 100), 0, 100);
int waterMl = constrain(map(analogRead(WATER_PIN), 0, 4095, 0, 500), 0, 500);
if (soilPercent < 30 && soilPercent > 2) {
    digitalWrite(RELAY_PIN, HIGH);
    delay(2000);
    digitalWrite(RELAY_PIN, LOW);
}
```

Рис.3.3. Фрагмент програмного коду керування системою автоматичного поливу на базі ESP32

3.2 Налаштування системи автоматизованого контролю мікроклімату

Підсистема клімат-контролю призначена для підтримання оптимальних показників температури та вологості повітря, що є важливими умовами для нормального розвитку рослин, ефективного фотосинтезу та запобігання виникненню грибкових захворювань. Робота модуля ґрунтується на використанні цифрового датчика температури та вологості, який забезпечує точне вимірювання параметрів навколишнього середовища в режимі реального часу. Отримані дані передаються до мікроконтролера для подальшого аналізу та порівняння із заданими агротехнічними нормами.

Основним засобом регулювання мікроклімату є система примусової вентиляції. У разі перевищення встановленого температурного порогу мікроконтролер формує сигнал на релейний модуль, який активує вентилятор для забезпечення циркуляції повітря та охолодження зони вирощування рослин. Це дозволяє запобігти перегріванню у закритих приміщеннях або під дією інтенсивного сонячного випромінювання.

Одночасно система виконує постійний моніторинг вологості повітря, що

забезпечує можливість своєчасного реагування на зміну параметрів мікроклімату та підтримання сприятливих умов для росту рослин.

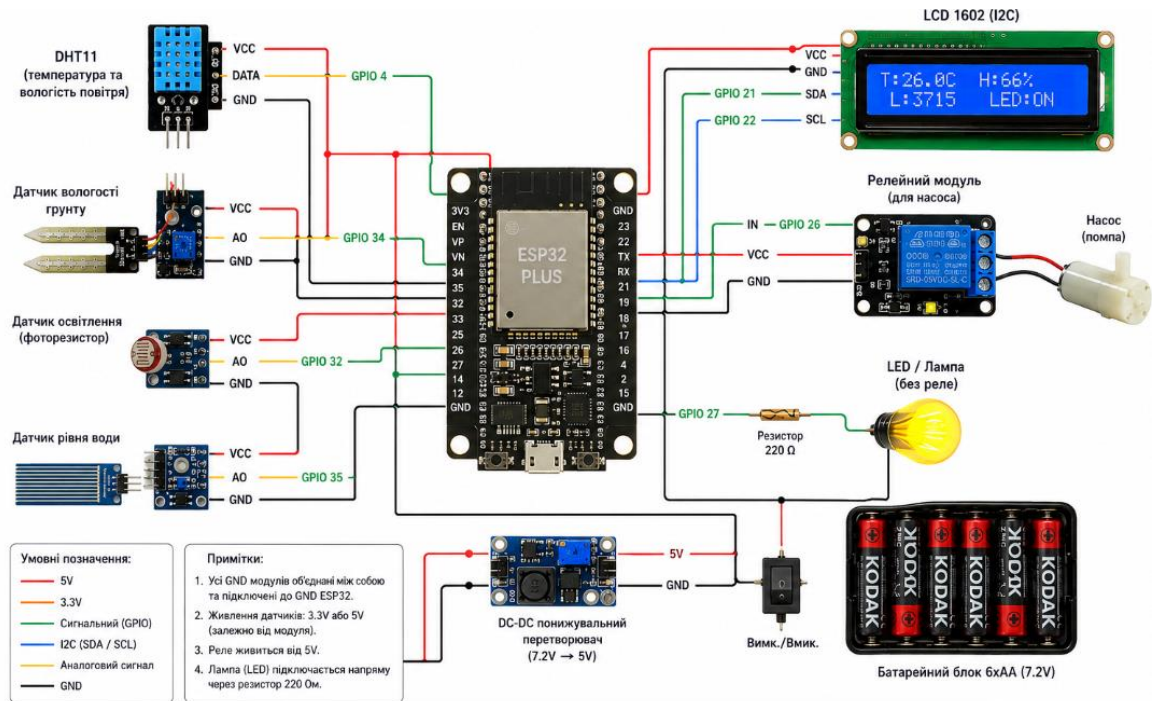


Рис.3.4. Схема підключення компонентів системи «Мікроклімату»

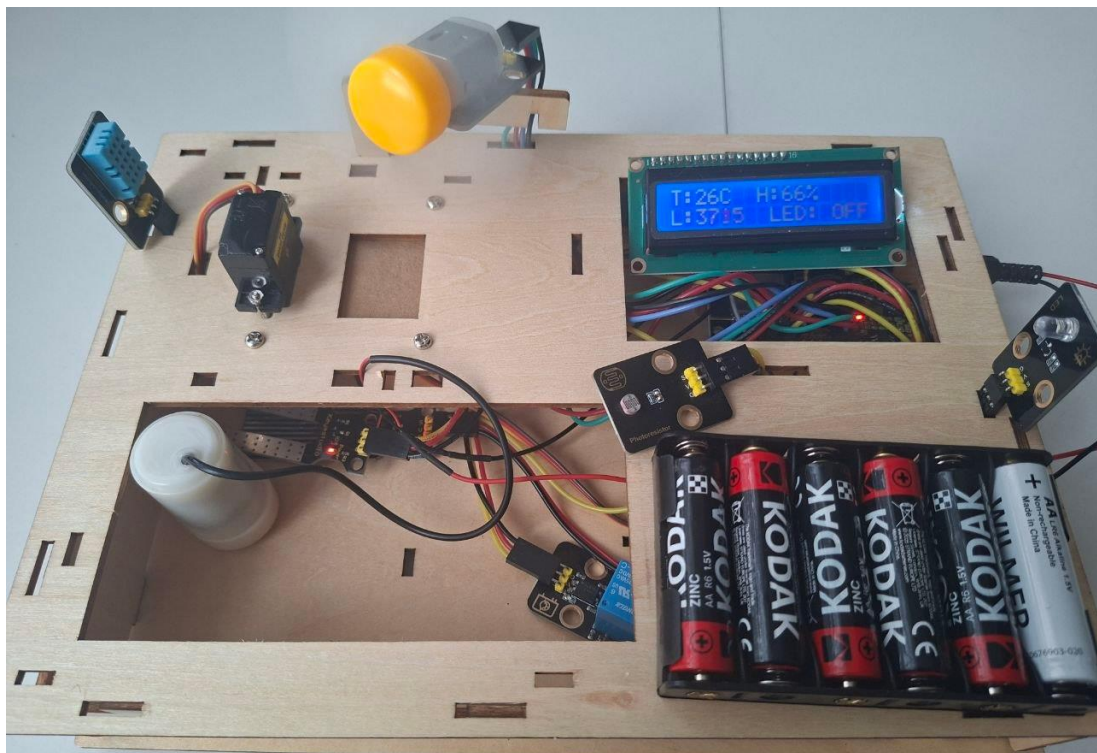


Рис.3.5. Макет системи мікроклімату

Важливою особливістю програмної реалізації є інтеграція параметрів мікроклімату у вебінтерфейс системи. Користувач має можливість у режимі реального часу переглядати показники температури та вологості, а також дистанційно керувати роботою вентиляційної системи в ручному режимі за необхідності. Поєднання автоматичного керування та можливості ручного втручання забезпечує гнучкість функціонування системи клімат-контролю та дозволяє підтримувати оптимальні умови в межах «Розумної ферми».

Програмний алгоритм керування мікрокліматом реалізовано з урахуванням природних циклів розвитку рослин. Для забезпечення оптимальних умов росту система автоматично змінює температурні пороги залежно від часу доби. У денний період вентиляція активується при перевищенні температури 28–30 °С, що сприяє ефективному охолодженню середовища під час активного фотосинтезу. У нічний час порогове значення автоматично знижується до 20–22 °С, забезпечуючи природне зменшення температури для підтримання нормальних біологічних процесів рослин.

Такий підхід дозволяє моделювати природні умови вирощування та запобігати надмірному виснаженню рослин через підвищені нічні температури. Перемикання між режимами роботи виконується автоматично, а поточні параметри та графіки змін відображаються у вебінтерфейсі для зручного моніторингу та контролю системи.

```
DHT11.read(DHT11PIN);
int tempC = DHT11.temperature;
int tempLimit = (lightRaw < 800) ? 22 : 28;
if (tempC >= tempLimit) fanSpeed = 255;
else if (tempC >= (tempLimit - 2) && fanSpeed > 0 && fanSpeed < 255) fanSpeed = 130;
analogWrite(FAN_PIN1, fanSpeed);
digitalWrite(FAN_PIN2, LOW);
```

Рис.3.6. Фрагмент коду керування кліматом

Важливим елементом системи забезпечення оптимальних умов для росту рослин є модуль автоматичного доосвітлення. Його функціонування реалізовано на

основі фоторезистора (LDR-сенсора), який визначає рівень природної освітленості в зоні вирощування. Отримані значення аналізуються програмним алгоритмом та порівнюються із заданим пороговим рівнем освітлення. У випадку недостатньої кількості світла мікроконтролер через релейний модуль автоматично активує LED-панель.

Реалізований підхід дозволяє компенсувати нестачу природного освітлення у похмурий період або автоматично збільшувати тривалість світлового дня, що є необхідною умовою для стабільного проходження процесів фотосинтезу та нормального розвитку рослин.

Окрім автоматичного режиму роботи, інформація про стан системи освітлення відображається на локальному LCD-дисплеї та вебпанелі, що забезпечує користувачеві можливість постійного візуального контролю та ручного керування освітленням у разі потреби. На рис. 3.7 наведено фрагмент програмного коду реалізації системи автоматизованого доосвітлення.

```
if (manualLedActive) {  
    if (millis() - ledTimer >= 5000) {  
        digitalWrite(LED_PIN, LOW);  
        manualLedActive = false;  
    }  
} else {  
  
    if (lightRaw < 2000) digitalWrite(LED_PIN, HIGH);  
    else digitalWrite(LED_PIN, LOW);  
}
```

Рис.3.7. Фрагмент програмного коду керування освітленням

3.3 Налаштування та програмна реалізація вебкерування розумною фермою

Підключення плати ESP32 до мережі. Мікроконтролер ESP32 оснащений вбудованими модулями бездротового зв'язку Wi-Fi 2,4 ГГц та Bluetooth 4.2, що забезпечує можливість підключення пристрою до локальної мережі та організації обміну даними між компонентами системи «розумної ферми». Завдяки підтримці бездротових технологій плата може передавати інформацію на мобільні пристрої, комп'ютери або вебсервіси в режимі реального часу.

Для налаштування мережевого підключення необхідно використовувати такі компоненти:

- мережу Wi-Fi стандарту 2,4 ГГц, яка може бути створена маршрутизатором або мобільною точкою доступу;
- параметри підключення до мережі, зокрема назву Wi-Fi мережі (SSID) та пароль доступу;
- смартфон, планшет або персональний комп'ютер, підключений до тієї ж локальної мережі для подальшого керування та моніторингу системи.

У середовищі розробки Arduino IDE для роботи з бездротовим модулем використовується бібліотека <WiFi.h>, яка забезпечує налаштування параметрів мережі, встановлення з'єднання та контроль стану підключення ESP32 до Wi-Fi мережі.

Модуль ESP32 підтримує декілька режимів роботи мережевого інтерфейсу:

- режим станції (STA) — плата функціонує як Wi-Fi клієнт та підключається до існуючої точки доступу;
- режим точки доступу (AP або Soft-AP) — ESP32 створює власну бездротову мережу, до якої можуть підключатися інші пристрої;
- режим AP-STA — комбінований режим, у якому мікроконтролер одночасно працює як точка доступу та клієнт іншої мережі.

Також підтримуються сучасні механізми захисту бездротових мереж, зокрема стандарти WPA, WPA2 та WEP, що забезпечує безпечний обмін даними між компонентами системи.

ESP32 має можливість виконувати пошук доступних Wi-Fi мереж шляхом активного або пасивного сканування ефіру. Додатково підтримується режим моніторингу мережевого трафіку (promiscuous mode), який дозволяє аналізувати пакети стандарту IEEE 802.11 Wi-Fi.

Для реалізації дистанційного керування елементами системи «розумної ферми» було розроблено веб-інтерфейс на основі мікроконтролера ESP32. Використання вбудованого Wi-Fi модуля дозволяє створити локальний вебсервер, через який здійснюється моніторинг показників датчиків та керування виконавчими пристроями у режимі реального часу.

У даному прикладі реалізовано вебкеровану систему, що забезпечує відображення поточних параметрів роботи ферми та дистанційне керування окремими модулями за допомогою браузера на комп'ютері або мобільному пристрої. Загальний вигляд алгоритму вебкерування розумною фермою представлено на рис. 3.8.

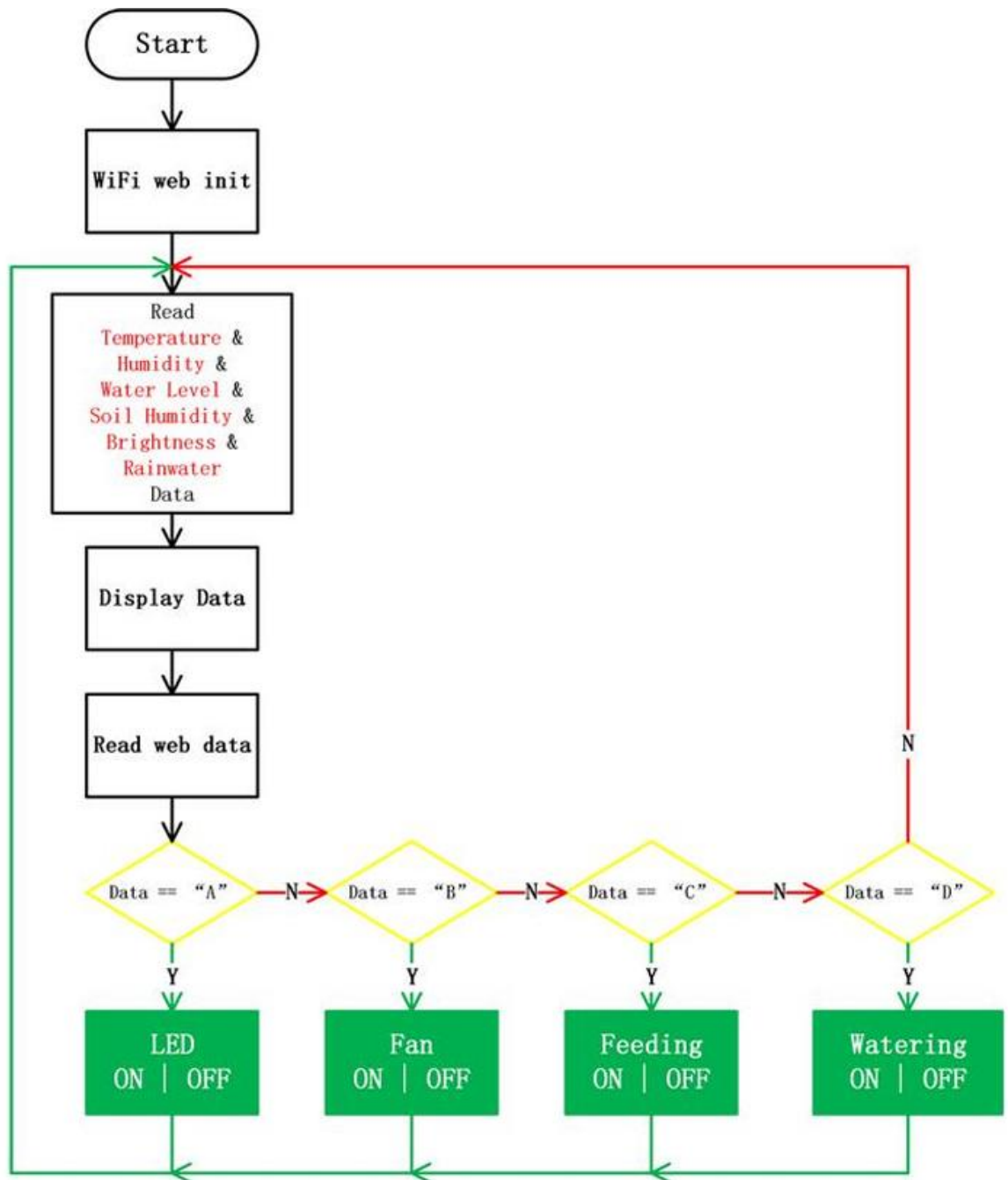


Рис 3.8. Алгоритм роботи вебкеруваної системи розумної ферми

Для реалізації підключення плати до мережі створено програмний код Connect-the-ESP32-to-the-Network, який використовується для налаштування та перевірки роботи мережевого з'єднання системи.

```
#include <WiFi.h>

const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

void setup() {
  Serial.begin(9600);
  //Initialize Wifi
  WiFi.begin(ssid, password);
  //Scan for wifi. If connection fails, stay in connecting, and execute "while" loop
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  //Connected. Print the IP address
  Serial.println("Connected to WiFi");
  Serial.println(WiFi.localIP());
}

void loop() {
}
```

Рис.3.9 програмний код Connect-the-ESP32-to-the-Network

У програмному коді виконується заміна параметру `your_SSID` на назву використовуваної Wi-Fi мережі, а значення `your_PASSWORD` - на пароль доступу до цієї мережі. Це забезпечує коректне підключення плати ESP32 до локальної бездротової мережі для подальшого обміну даними та вебкерування системою.

```
#include <WiFi.h>
const char* ssid = "Cudy-37EC";
const char* password = "93128384";
```

Рис.3.10 Заміна паролю

Далі необхідно обрати тип плати ESP32 Dev Module та вказати відповідний COM-порт, до якого підключено мікроконтролер. Після цього виконується компіляція та завантаження програмного коду у пам'ять плати ESP32 для подальшої перевірки працездатності системи.



Рис.3.11 Вибір типу плати ESP32 Dev Module та відповідний COM-порт

В результаті, після завантаження програмного коду мікроконтролер ESP32 автоматично підключається до вказаної мережі Wi-Fi. Після успішного встановлення з'єднання у вікні Serial Monitor відображається призначена IP-адреса пристрою, яка використовується для подальшого вебкерування та доступу до системи через локальну мережу, рис. 3.12

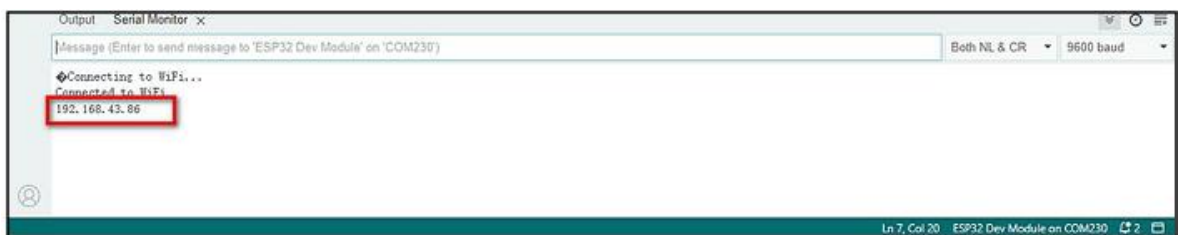


Рис.3.12 IP-адреса ESP32 у Serial Monitor

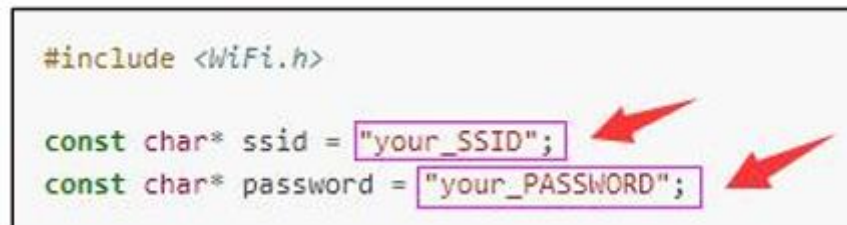
Налаштування веб-сайту. Після успішного підключення до мережі Wi-Fi бібліотека вебсервера ESP32 забезпечує можливість створення та відображення вебсторінок у локальній мережі. У програмному коді реалізовано веб-інтерфейс, який після відкриття у браузері відображається

Для налаштування та перевірки роботи вебсторінки було розроблено і реалізовано програмний код, частину коду зображено на рис.3.13.

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html><head>
<meta charset="utf-8"><meta name="viewport" content="width=device-width, initial-scale=1">
<title>Smart Farm Pro</title>
<style>
:root { --bg: #e0e5ec; --shadow: #a3b1c6; --light: #ffffff; --text: #444; --accent: #81c784; }
body { font-family: sans-serif; background: var(--bg); color: var(--text); display: flex; flex-direction: column; align-items: center; padding: 20px; }
.card-list { width: 100%; max-width: 400px; display: flex; flex-direction: column; gap: 12px; }
.row { background: var(--bg); padding: 15px; border-radius: 20px; box-shadow: 7px 7px 15px var(--shadow), -7px -7px 15px var(--light);
  display: flex; align-items: center; justify-content: space-between; }
.val { font-weight: bold; color: #58a6ff; }
.btn-grid { display: grid; grid-template-columns: 1fr 1fr 1fr; gap: 15px; width: 100%; max-width: 400px; margin-top: 20px; }
.btn { background: var(--bg); border: none; padding: 18px 5px; border-radius: 20px; box-shadow: 5px 5px 10px var(--shadow), -5px -5px 10px var(--light);
  cursor: pointer; font-size: 0.8em; font-weight: bold; }
.active-btn { color: #81c784; box-shadow: inset 4px 4px 8px var(--shadow); }
</style>
</head><body>
<h3>Smart Farm Pro</h3>
<div class="card-list">
  <div class="row"><span>🌡 Temperature:</span><span class="val"><span id="tc">0</span>°C / <span id="tf">0</span>°F</div>
  <div class="row"><span>💧 Air Humidity:</span><span class="val" id="h">0</span>%</div>
  <div class="row"><span>🌱 Soil Moisture:</span><span class="val" id="s">0</span>%</div>
  <div class="row"><span>💡 Light Level:</span><span class="val" id="lr">0</span> (<span id="phase">...</span>)</div>
  <div class="row"><span>💧 Water Tank:</span><span class="val" id="w">0</span> ml</div>
</div>
<div class="btn-grid">
  <button class="btn" id="lbtn" onclick="cmd('led')">💡 LED 5s</button>
  <button class="btn" onclick="cmd('pump')">🚰 PUMP</button>
  <button class="btn" id="fbtn" onclick="cmd('fan')">🌀 FAN</button>
</div>
)
```

Рис.3.13 Програмний код веб-сайту

На наступному етапі, у програмному коді необхідно вказати назву Wi-Fi мережі у параметрі `your_SSID`, а також пароль доступу до мережі у параметрі `your_PASSWORD`. Після внесення відповідних змін виконується завантаження програмного коду на плату ESP32 для подальшого запуску вебсервера, рис.3.14.



```
#include <WiFi.h>

const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";
```

Рис.3.14 Налаштування параметрів Wi-Fi мережі у програмному коді ESP32

Для доступу до вебсторінки ESP32 необхідно ввести отриману IP-адресу пристрою в адресний рядок браузера на комп'ютері або смартфоні, підключеному до тієї ж Wi-Fi мережі. Після цього у браузері відобразиться вебсторінка, сформована мікроконтролером ESP32, рис.3.15.

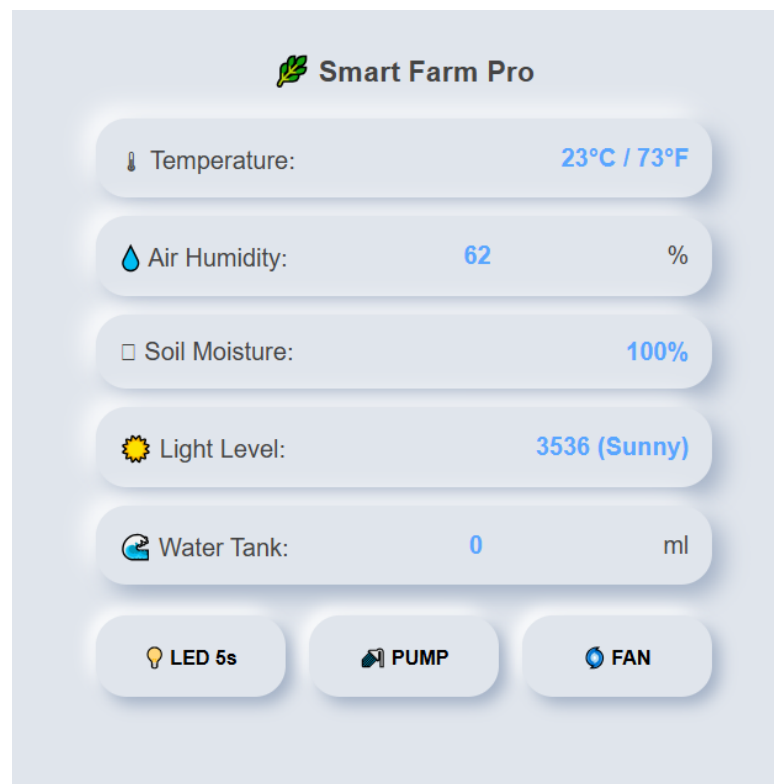


Рис.3.15 Відображення вебсторінки ESP32 у браузері

Для коректної роботи системи необхідно використовувати мережу Wi-Fi стандарту 2,4 ГГц, оскільки модуль ESP32 не підтримує підключення до мереж 5 ГГц. Також комп'ютер або смартфон, з якого виконується доступ до IP-адреси пристрою, повинен бути підключений до тієї ж локальної Wi-Fi мережі, що й плата ESP32, рис.3.16.

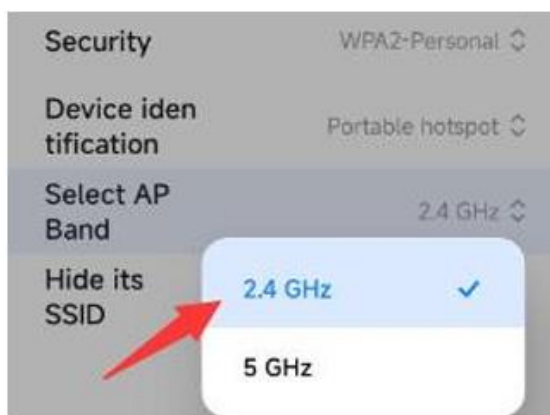


Рис.3.16 Вибір мережуї Wi-Fi

3.4 Програмна реалізація інтерфейсу користувача та функцій керування

Для забезпечення ефективної взаємодії між користувачем та автоматизованою системою «Розумна ферма» було розроблено інтерактивну веб-панель керування. Інтерфейс побудований на принципах інтуїтивності та оперативності, що дозволяє здійснювати моніторинг стану рослин та втручатися в роботу виконавчих механізмів у реальному часі.

Веб-інтерфейс забезпечує дистанційний моніторинг параметрів системи «розумної ферми» та керування виконавчими пристроями через бездротову мережу Wi-Fi Основні функції веб-інтерфейсу є:

- забезпечення моніторингу поточного значення температури у системі «розумної ферми» в режимі реального часу;
- реалізація відображення поточного рівня вологості повітря для контролю мікрокліматичних параметрів фермерського середовища;

- забезпечення відображення значення вологості ґрунту в режимі реального часу, що дозволяє контролювати ефективність системи зрошення та стан ґрунту;
- реалізація відображення рівня освітленості ферми для оцінювання умов освітлення у робочій зоні системи;
- забезпечення контролю поточного рівня води у резервуарі системи «розумної ферми» в режимі реального часу.

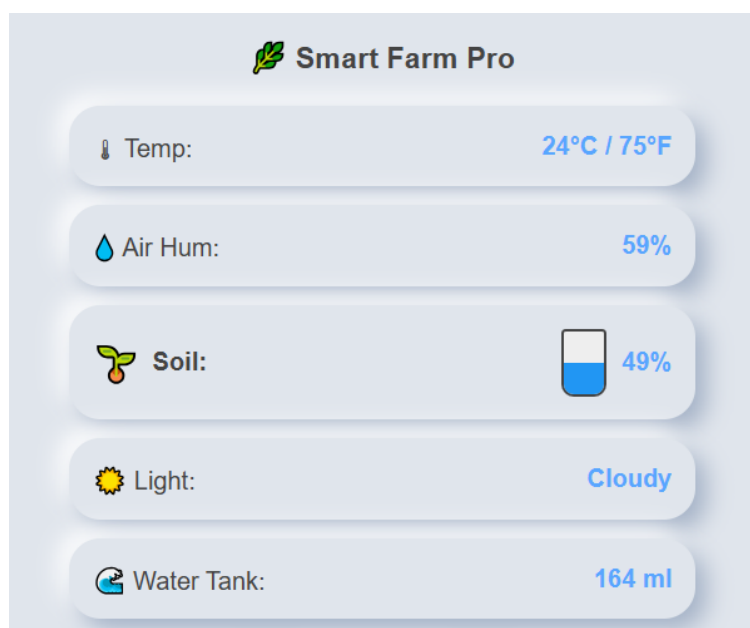


Рис.3.17 Головна сторінка застосунку

Основним інструментом ручного впливу на систему є блок функціональних кнопок, кожна з яких відповідає за конкретний вузол автоматизації:

1. Кнопка керування системою зрошення: Дана функція реалізує можливість примусової активації водяної помпи. Це необхідно у випадках, коли користувач візуально фіксує потребу в додатковому зволоженні або проводить технічну перевірку справності гідравлічної системи. При натисканні на кнопку мікроконтролер ігнорує поточні показники датчика вологості ґрунту та вмикає реле на заданий програмний інтервал.

2. Кнопка активації вентиляції: Призначена для ручного керування мікрокліматом. Користувач може ініціювати провітрювання боксу або зниження

температури, навіть якщо автоматичний поріг спрацювання ще не був досягнутий. Це дозволяє гнучко регулювати циркуляцію повітря залежно від зовнішніх чинників.

3. Кнопка керування освітленням: Дозволяє вмикати та вимикати додаткове LED-освітлення. Функція корисна для візуального огляду рослин у нічний час або для примусового подовження світлового дня в похмуру погоду, що безпосередньо впливає на інтенсивність процесів фотосинтезу.

Програмна логіка взаємодії побудована на використанні HTTP-запитів. Кожне натискання кнопки на веб-сторінці генерує запит до асинхронного сервера на базі мікроконтролера ESP32, який миттєво змінює стан відповідного цифрового піна. Для покращення користувацького досвіду в інтерфейсі реалізовано динамічну індикацію: колір кнопки або текстовий статус поруч із нею змінюється залежно від того, чи активний виконавчий механізм у даний момент. Такий підхід забезпечує надійний зворотний зв'язок та мінімізує ризик помилкових дій з боку користувача.

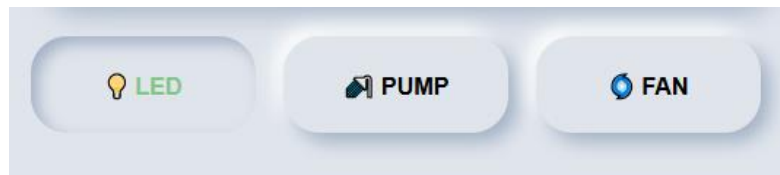


Рис.3.18. Меню керування виконавчими модулями системи

ВИСНОВКИ

У першому розділі проведено аналіз сучасних підходів до побудови систем розумного фермерства на основі технологій Інтернету речей. Розглянуто основні сфери застосування IoT у сільському господарстві, принципи функціонування бездротових сенсорних мереж, а також особливості передачі даних у системах моніторингу та автоматизації аграрних процесів. Виконано порівняльний аналіз протоколів бездротового зв'язку, зокрема Wi-Fi, ZigBee, Z-Wave та LoRaWAN, що дозволило визначити їх переваги та обмеження при використанні в системах розумного фермерства.

Проаналізовано апаратні компоненти автоматизованої системи, зокрема мікроконтролер ESP32, сенсори, виконавчі механізми та допоміжні модулі, що забезпечують збір, обробку та передачу даних у режимі реального часу. Окрему увагу приділено структурі програмно-апаратної архітектури системи та взаємодії її окремих рівнів. Крім того, розглянуто можливості інтегрованого середовища розробки Arduino IDE, яке було обрано для створення та налагодження програмного забезпечення системи завдяки простоті використання, підтримці ESP32 та широкому набору бібліотек для роботи з IoT-компонентами.

Проведений аналіз підтвердив доцільність використання сучасних IoT-технологій для побудови автоматизованої системи «Розумна ферма», що забезпечує ефективний моніторинг параметрів середовища та автоматизацію процесів керування.

Виконуючи друге поставлене завдання, виконано налаштування та програмну реалізацію апаратно-програмного комплексу «Розумна ферма» на основі мікроконтролера ESP32 і технологій Інтернету речей. У процесі роботи було реалізовано підключення, програмування та тестування основних сенсорних і виконавчих компонентів системи, зокрема датчиків освітленості, температури,

вологості ґрунту, рівня води, а також релейних модулів, вентилятора, LCD-дисплея та водяного насоса.

Розроблено програмні алгоритми для автоматичного керування освітленням, контролю температури, моніторингу вологості ґрунту та системи автоматичного зрошення. Проведене тестування підтвердило коректність роботи сенсорів, стабільність передачі даних та ефективність взаємодії між апаратними й програмними компонентами системи. Реалізовані алгоритми забезпечують автоматичне реагування на зміну параметрів середовища, що дозволяє підвищити рівень автономності системи.

Окрему увагу приділено використанню середовища Arduino IDE для створення та налагодження програмного забезпечення. Використання ESP32 забезпечило підтримку бездротового зв'язку Wi-Fi та можливість інтеграції системи з вебінтерфейсами й мобільними застосунками для дистанційного моніторингу та керування.

Отримані результати підтвердили працездатність розробленого апаратно-програмного комплексу та доцільність використання IoT-технологій для автоматизації процесів контролю мікроклімату, освітлення та зрошення у системах розумного фермерства.

У третьому розділі виконано практичну реалізацію автоматизованої системи «Розумна ферма». У процесі роботи було реалізовано підсистеми автоматичного поливу, контролю мікроклімату, автоматизованого освітлення та дистанційного вебкерування виконавчими механізмами системи.

Розроблено програмні алгоритми для моніторингу вологості ґрунту, температури, вологості повітря, рівня освітленості та контролю рівня води. На основі отриманих даних система автоматично приймає рішення щодо активації поливу, вентиляції та освітлення, що дозволяє підтримувати оптимальні умови для росту рослин і забезпечує раціональне використання ресурсів.

У межах практичної реалізації було створено вебінтерфейс для дистанційного моніторингу та керування системою через мережу Wi-Fi. Вебпанель

забезпечує відображення поточних параметрів середовища в режимі реального часу, а також дозволяє користувачеві вручну керувати поливом, вентиляцією та освітленням за допомогою браузера на комп'ютері або мобільному пристрої.

Проведене тестування підтвердило стабільність роботи розробленого апаратно-програмного комплексу, коректність функціонування сенсорів, виконавчих механізмів і вебсистеми керування. Отримані результати свідчать про ефективність використання IoT-технологій для автоматизації процесів моніторингу та керування у сучасних системах розумного фермерства.

ПЕРЕЛІК ПОСИЛАНЬ

1. Pathmudi V. R., Khatri N., Kumar S., Abdul-Qawy A. S. H., Vyas A. K. A systematic review of IoT technologies and their constituents for smart and sustainable agriculture applications. *Scientific African*. 2023.
2. Singh S. P., Dhiman G., Juneja S., Viriyasitavat W., Singal G., Kumar N., Johri P. A new QoS optimization in IoT-smart agriculture using rapid adaption based nature-inspired approach. *IEEE Internet of Things Journal*. 2023.
3. Shaikh F. K., Karim S., Zeadally S., Nebhen J. Recent trends in Internet-of-Things-enabled sensor technologies for smart agriculture. *IEEE Internet of Things Journal*. 2022. Vol. 9, no. 23. P.83–98.
4. Chataut R., Phoummalayvane A., Akl R. Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and Industry 4.0. *Sensors*. 2023. Vol. 23, no. 16. P.194.
5. Morchid A., El Alami R., Raezah A. A., Sabbar Y. Applications of Internet of Things (IoT) and sensors technology to increase food security and agricultural sustainability: Benefits and challenges. *Ain Shams Engineering Journal*. 2024.
6. Rudrakar S., Pughani P. IoT based agriculture (Ag-IoT): A detailed study on architecture, security and forensics. *Information Processing in Agriculture*. 2023.
7. Ayaz M., Ammad-Uddin M., Sharif Z., Mansour A., Aggoune E. M. Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk. *IEEE Access*. 2019. Vol. 7.
8. Khanna A., Kaur S. Evolution of Internet of Things (IoT) and its significant impact in the field of precision agriculture. *Computers and Electronics in Agriculture*. 2019. Vol. 157. P. 218–231.
9. Kour V. P., Arora S. Recent developments of the Internet of Things in agriculture: A survey. *IEEE Access*. 2020. Vol. 8. P. 12–57.

10. Raju K. L., Vijayaraghavan V. IoT technologies in agricultural environment: A survey. *Wireless Personal Communications*. 2020. Vol. 113, no. 4. P. 24–46.
11. Ojha T., Misra S., Raghuwanshi N. S. Internet of Things for agricultural applications: The state of the art. *IEEE Internet of Things Journal*. 2021. Vol. 8, no. 14. P. 73–97.
12. Sinha B. B., Dhanalaksmi R. Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Generation Computer Systems*. 2022. Vol. 126, no. 4. P. 169–184.
13. Prakash C., Singh L. P., Gupta A., Lohan S. K. Advancements in smart farming: A comprehensive review of IoT, wireless communication, sensors, and hardware for agricultural automation. *Sensors and Actuators A: Physical*. 2023.
14. Pawar A., Deosarkar S. B. IoT-based smart agriculture: An exhaustive study. *Wireless Networks*. 2023. Vol. 29, no. 6. P.57–70.
15. Quy V. K., Hau N. V., Anh D. V., Quy M. N., Ban N. T., Lanza S., Randazzo G., Muzirafuti A. IoT-enabled smart agriculture: Architecture, applications, and challenges. *Applied Sciences*. 2022. P. 396.
16. Tao W., Zhao L., Wang G., Liang R. Review of the Internet of Things communication technologies in smart agriculture and challenges. *Computers and Electronics in Agriculture*. 2021.
17. Avşar E., Mowla M. N. Wireless communication protocols in smart agriculture: A review on applications, challenges and future trends. *Ad Hoc Networks*. 2022.
18. Senoo E. E. K., Akansah E., Mendonça I., Aritsugi M. Monitoring and control framework for IoT, implemented for smart agriculture. *Sensors*. 2023.
19. Xu J., Gu B., Tian G. Review of agricultural IoT technology. *Artificial Intelligence in Agriculture*. 2022. Vol. 6. P. 10–22.
20. Verdouw C., Sundmaeker H., Tekinerdogan B., Conzon D., Montanaro T. Architecture framework of IoT-based food and farm systems: A multiple case study. *Computers and Electronics in Agriculture*. 2019.

21. Parnian A., Mahbod M., Mitra C. K., Beyrami H., Prasad M. N. V. Internet of Things (IoT) in soil management for achieving smart agriculture. *Agroecological Approaches for Sustainable Soil Management*. 2023. P. 57–86.
22. Roussaki I., Doolin K., Skarmeta A., Routis G., Lopez-Morales J. A., Claffey E., Mora M., Martinez J. A. Building an interoperable space for smart agriculture. *Digital Communications and Networks*. 2023. Vol. 9, no. 1. P. 183–193.
23. Chamara N., Islam M. D., Bai G., Shi Y., Ge Y. Ag-IoT for crop and environment monitoring: Past, present, and future. *Agricultural Systems*. 2022.
24. Abbasi R., Martinez P., Ahmad R. The digitization of agricultural industry—A systematic literature review on agriculture 4.0. *Smart Agricultural Technology*. 2022.
25. Singh G., Kalra N., Yadav N., Sharma A., Saini M. Smart agriculture: A review. *Siberian Journal of Life Sciences and Agriculture*. 2022. Vol. 14, no. 6. P. 423–454.
26. Atalla S., Tarapiah S., Gawanmeh A., Daradkeh M., Mukhtar H., Himeur Y., Mansoor W., Hashim K. F. B., Daadoo M. IoT-enabled precision agriculture: Developing an ecosystem for optimized crop management. *Information*. 2023. Vol. 14, no. 4. P. 205.
27. Benyezza H., Bouhedda M., Kara R., Rebouh S. Smart platform based on IoT and WSN for monitoring and control of a greenhouse in the context of precision agriculture. *Internet of Things*. 2023.
28. Farooq M. S., Riaz S., Helou M. A., Khan F. S., Abid A., Alvi A. Internet of Things in greenhouse agriculture: A survey on enabling technologies, applications, and protocols. *IEEE Access*. 2022. Vol. 10. P. 53–97.
29. Diya V., Nandan P., Dhote R. R. IoT-based precision agriculture: A review. *Proceedings of Emerging Trends in Technology and Intelligent Systems (ETTIS)*. 2022. P. 73–86.
30. Patroo L., Thacooree K., Mungur A. A smart precision irrigation and monitoring system. *Proceedings of International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering*. Springer, 2018. P. 105–114.

ДОДАТОК А
РЕАЛІЗАЦІЯ ВЕБ-СИСТЕМИ
МОНІТОРИНГУ ТА КЕРУВАННЯ «РОЗУМНОЮ ФЕРМОЮ»

```
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <dht11.h>
#include <LiquidCrystal_I2C.h>

#include <WiFi.h>
const char* ssid = "Cudy-37EC";
const char* password = "93128384";

#define DHT11PIN 17
#define SOIL_PIN 32
#define WATER_PIN 35
#define LIGHT_PIN 34
#define LED_PIN 27
#define FAN_PIN1 19
#define FAN_PIN2 18
#define RELAY_PIN 25

dht11 DHT11;
LiquidCrystal_I2C lcd(0x27, 16, 2);
AsyncWebServer server(80);

int fanSpeed = 0;
unsigned long lastDisplayUpdate = 0;
int currentSlide = 0;

// Змінні для таймера лампочки
unsigned long ledTimer = 0;
bool manualLedActive = false;

String getPhase(int light) {
    if (light < 800) return "Night";
    if (light < 2000) return "Cloudy";
    return "Sunny";
}
```

```

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html><head>
  <meta charset="utf-8"><meta name="viewport" content="width=device-
width, initial-scale=1">
  <title>Smart Farm Pro</title>
  <style>
    :root { --bg: #e0e5ec; --shadow: #a3b1c6; --light: #ffffff; --text: #444; --
accent: #81c784; }
    body { font-family: sans-serif; background: var(--bg); color: var(--text);
display: flex; flex-direction: column; align-items: center; padding: 20px; }
    .card-list { width: 100%; max-width: 400px; display: flex; flex-direction:
column; gap: 12px; }
    .row { background: var(--bg); padding: 15px; border-radius: 20px; box-
shadow: 7px 7px 15px var(--shadow), -7px -7px 15px var(--light);
    display: flex; align-items: center; justify-content: space-between; }
    .val { font-weight: bold; color: #58a6ff; }
    .btn-grid { display: grid; grid-template-columns: 1fr 1fr 1fr; gap: 15px; width:
100%; max-width: 400px; margin-top: 20px; }
    .btn { background: var(--bg); border: none; padding: 18px 5px; border-radius:
20px; box-shadow: 5px 5px 10px var(--shadow), -5px -5px 10px var(--light);
    cursor: pointer; font-size: 0.8em; font-weight: bold; }
    .active-btn { color: #81c784; box-shadow: inset 4px 4px 8px var(--shadow); }
  </style>
</head><body>
  <h3>  Smart Farm Pro</h3>
  <div class="card-list">
    <div class="row"><span>  Temperature:</span><span
class="val"><span id="tc">0</span>°C / <span
id="tf">0</span>°F</span></div>
    <div class="row"><span>  Air Humidity:</span><span class="val"
id="h">0</span>%</div>
    <div class="row"><span>  Soil Moisture:</span><span
class="val"><span id="s">0</span>%</span></div>
    <div class="row"><span>  Light Level:</span><span
class="val"><span id="lr">0</span> (<span
id="phase">...</span>)</span></div>
    <div class="row"><span>  Water Tank:</span><span class="val"
id="w">0</span> ml</div>
  </div>
  <div class="btn-grid">
    <button class="btn" id="lbtn" onclick="cmd('led')">  LED 5s</button>
    <button class="btn" onclick="cmd('pump')">  PUMP</button>

```

```

<button class="btn" id="fbtn" onclick="cmd('fan')">  FAN</button>
</div>
<script>
function cmd(a) { fetch(`/cmd?act=${a}`); }
setInterval(() => {
  fetch('/data').then(r => r.json()).then(d => {
    document.getElementById('tc').innerText = d.tc;
    document.getElementById('tf').innerText = d.tf;
    document.getElementById('h').innerText = d.h;
    document.getElementById('s').innerText = d.s;
    document.getElementById('lr').innerText = d.lr;
    document.getElementById('phase').innerText = d.phase;
    document.getElementById('w').innerText = d.w;
    document.getElementById('lbtn').classList.toggle('active-btn', d.ledStatus
== 1);
    document.getElementById('fbtn').classList.toggle('active-btn', d.f > 0);
  });
}, 1000);
</script>
</body></html>
)rawliteral";

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT); pinMode(RELAY_PIN, OUTPUT);
  pinMode(FAN_PIN1, OUTPUT); pinMode(FAN_PIN2, OUTPUT);

  lcd.init(); lcd.backlight();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) { delay(500); }

  server.on("/", HTTP_GET, [](AsyncWebServerRequest *f){ f->send(200,
"text/html", index_html); });

  server.on("/data", HTTP_GET, [](AsyncWebServerRequest *f){
    int lightRaw = analogRead(LIGHT_PIN);
    int soilRaw = analogRead(SOIL_PIN);
    int soilP = constrain(map(soilRaw, 4095, 2100, 0, 100), 0, 100);
    int water = constrain(map(analogRead(WATER_PIN), 0, 4095, 0, 500), 0,
500);
    int tempC = DHT11.temperature;
    int tempF = (tempC * 1.8) + 32;

    String json = "{\"tc\":\"" + String(tempC) +

```

```

        ",|\"tf|\": \" + String(tempF) +
        \",|\"h|\": \" + String(DHT11.humidity) +
        \",|\"s|\": \" + String(soilP) +
        \",|\"lr|\": \" + String(lightRaw) +
        \",|\"w|\": \" + String(water) +
        \",|\"phase|\": \" + getPhase(lightRaw) +
        \",|\"ledStatus|\": \" + String(digitalRead(LED_PIN)) +
        \",|\"f|\": \" + String(fanSpeed) + \"}";
    f->send(200, "application/json", json);
});

server.on("/cmd", HTTP_GET, [](AsyncWebServerRequest *f){
    String act = f->getParam("act")->value();

    if(act == "led") {
        digitalWrite(LED_PIN, HIGH);
        ledTimer = millis();
        manualLedActive = true;
    }

    if(act == "fan") { fanSpeed = (fanSpeed > 0) ? 0 : 255; }
    if(act == "pump") { digitalWrite(RELAY_PIN, HIGH); delay(2000); }
    digitalWrite(RELAY_PIN, LOW); }
    f->send(200, "text/plain", "OK");
});

server.begin();
}

void loop() {
    DHT11.read(DHT11PIN);
    int lightRaw = analogRead(LIGHT_PIN);
    int soilRaw = analogRead(SOIL_PIN);
    int soilPercent = constrain(map(soilRaw, 4095, 2100, 0, 100), 0, 100);
    int waterMl = constrain(map(analogRead(WATER_PIN), 0, 4095, 0, 500), 0,
500);
    int tempC = DHT11.temperature;

    if (manualLedActive) {
        if (millis() - ledTimer >= 5000) {
            digitalWrite(LED_PIN, LOW);
            manualLedActive = false;
        }
    } else {

```

```
    if (lightRaw < 2000) digitalWrite(LED_PIN, HIGH);  
    else digitalWrite(LED_PIN, LOW);  
}
```

```
int tempLimit = (lightRaw < 800) ? 22 : 28;  
if (tempC >= tempLimit) fanSpeed = 255;  
else if (tempC >= (tempLimit - 2) && fanSpeed > 0 && fanSpeed < 255)  
fanSpeed = 130;  
analogWrite(FAN_PIN1, fanSpeed);  
digitalWrite(FAN_PIN2, LOW);
```

```
if (soilPercent < 30 && soilPercent > 2) {  
    digitalWrite(RELAY_PIN, HIGH);  
    delay(2000);  
    digitalWrite(RELAY_PIN, LOW);  
}
```

```
// 4. LCD СЛАЙДЫ  
if (millis() - lastDisplayUpdate > 10000) { lastDisplayUpdate = millis();  
currentSlide = (currentSlide + 1) % 3; lcd.clear(); }  
if (currentSlide == 0) {  
    lcd.setCursor(0, 0); lcd.print("IP: "); lcd.print(WiFi.localIP().toString());  
} else if (currentSlide == 1) {  
    lcd.setCursor(0, 0); lcd.print("T:"); lcd.print(tempC); lcd.print("C H:");  
lcd.print(DHT11.humidity); lcd.print("%");  
    lcd.setCursor(0, 1); lcd.print("Soil:"); lcd.print(soilPercent); lcd.print("%");  
} else {  
    lcd.setCursor(0, 0); lcd.print("Light:"); lcd.print(lightRaw);  
    lcd.setCursor(0, 1); lcd.print("W:"); lcd.print(waterMl); lcd.print("ml L:");  
lcd.print(getPhase(lightRaw).substring(0,1));  
}  
  
    delay(200);  
}
```

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«АВТОМАТИЗОВАНА СИСТЕМА «РОЗУМНА ФЕРМА» НА
ОСНОВІ ТЕХНОЛОГІЙ ІНТЕРНЕТУ РЕЧЕЙ»

Виконала: здобувачка вищої освіти гр. ІСД-41

Світлана ЮХИМОВИЧ

Керівник: к.т.н., доцент

Ольга ПОЛОНЕВИЧ

Київ 2026

МЕТА ТА ЗАВДАННЯ

2

Мета роботи - розробка автоматизованої системи «Розумна ферма» на основі IoT-технологій для моніторингу та автоматичного керування параметрами фермерського середовища.

Об'єкт дослідження – процеси автоматизації та моніторингу параметрів фермерського середовища.

Предмет дослідження – програмно-апаратні засоби та методи побудови IoT-системи «Розумна ферма».

Завдання бакалаврської роботи:

1. Аналіз технологій, апаратних засобів та програмного забезпечення для побудови системи «розумна ферма» на основі IoT.
2. Дослідження особливостей налаштування та програмна реалізація апаратно- програмного комплексу «Розумна ферма»
3. Практична реалізація автоматизованої IoT-системи «Розумна ферма» з веборієнтованим керуванням

АНАЛІЗ ТЕХНОЛОГІЙ ІОТ ТА ПРОТОКОЛІВ ПЕРЕДАЧІ ДАНИХ У СИСТЕМАХ РОЗУМНОГО ФЕРМЕРСТВА

3

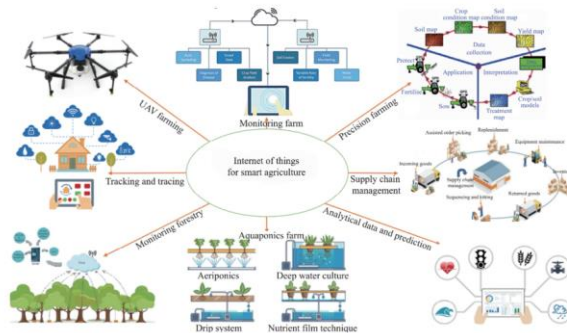


Рис.1. Застосування Інтернету речей у різних сферах для розумного сільського господарства

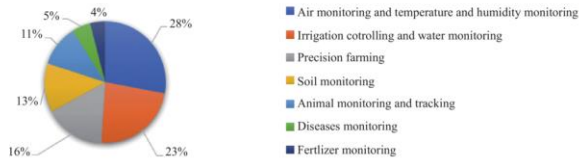


Рис.2. Галузеве застосування Інтернету речей для розумного сільського господарства

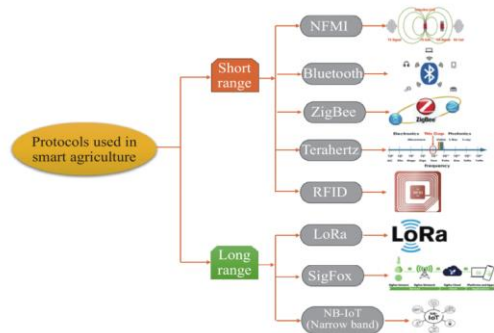


Рис.3. Протоколи Інтернету речей у розумному сільському господарстві

Табл.1.Протоколи зв'язку

Технологія	Wi-Fi	ZigBee	Z-Wave	LoRaWAN
Стандарт зв'язку	IEEE 802.11	IEEE 802.15.4	ITU-T G.9959	IEEE 802.15.4
Робоча частота	2.4 ГГц	2.4 ГГц	868.4 МГц	868 МГц
Радіус дії	50–100 м	75–100 м	до 100 м	до 15 км
Енергоспоживання	Високе	Низьке	Низьке	Дуже низьке
Швидкість даних	до 600 Мбіт/с	до 250 Кбіт/с	до 100 Кбіт/с	до 50 Кбіт/с
Топологія мережі	«Зірка»	Mesh	Mesh	«Зірка»
Проникна здатність	Низька	Середня	Середня	Висока
Макс. вузлів	~250	65 000+	232	тисячі

ОГЛЯД АПАРАТНИХ КОМПОНЕНТІВ АВТОМАТИЗОВАНОЇ СИСТЕМИ «РОЗУМНА ФЕРМА»

4

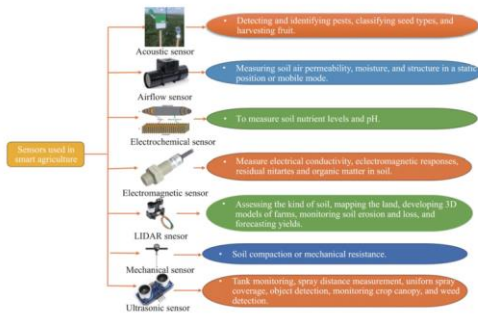


Рис.4. Датчики та їх застосування в розумному сільському господарстві



Рис.5. Приклад контролера ESP32 PLUS



Рис.6. Релейний модуль

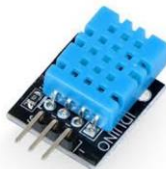


Рис.7. Датчик DHT11



Рис.8. Ультразвуковий датчик



Рис.9. Датчик рівня води



Рис.10. Датчика пари



Рис.11. Вентилятор

СЕРЕДОВИЩЕ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ARDUINO IDE

5



Рис.12. Інтерфейс середовища розробки Arduino IDE

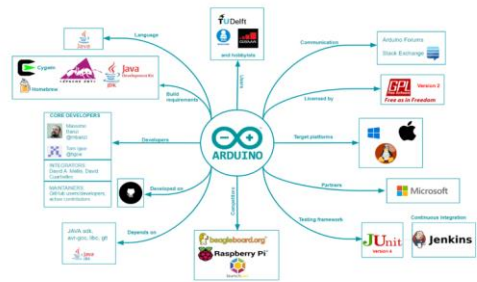
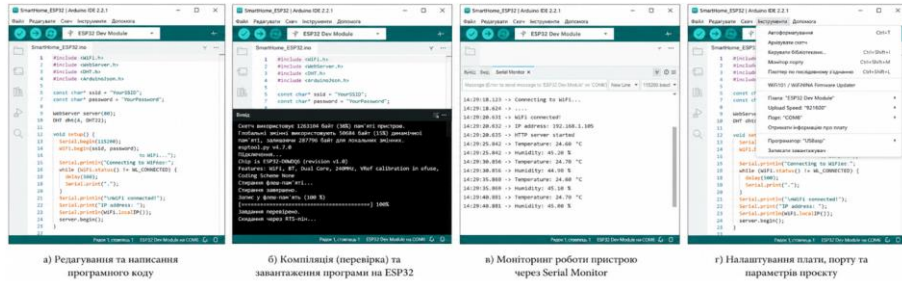


Рис.13 Контекстний вигляд середовища розробки Arduino IDE, що описує зв'язки з його оточенням



а) Редагування та написання програмного коду

б) Компіляція (перевірка) та завантаження програми на ESP32

в) Моніторинг роботи пристрою через Serial Monitor

г) Налаштування плати, порту та параметри проєкту

Рис.14 Основні етапи розробки та налагодження програмного забезпечення в середовищі Arduino IDE

НАЛАШТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ «РОЗУМНА ФЕРМА»

6

Water Pump:



Parameters:

- Power voltage: 5V
- Static current: 2mA
- Maximum contact voltage: 250VAC/30VDC
- Maximum current: 10A

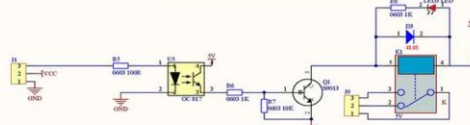


Рис. 15. Водяний насос та схема його підключення через релейний модуль



Рис.16. Програмний код Water-Pump

Рис.17. Програмний код Auto-irrigation



Рис.18. Плат ESP32 Dev Module та відповідний COM-порт



Рис.19. Плат ESP32 Dev Module та відповідний COM-порт



Рис. 20. Відображення значень вологості ґрунту та рівня води на дисплеї LCD1602

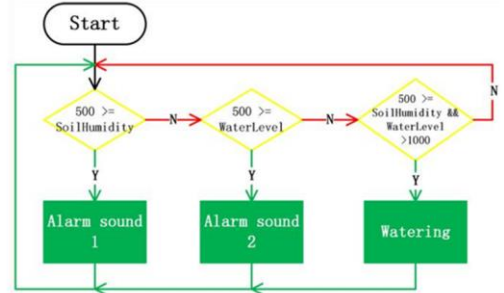


Рис. 21. Блок-схема контролю вологості ґрунту та рівня води

РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЧНОГО ПОЛИВУ НА ОСНОВІ ІОТ-ТЕХНОЛОГІЙ

7

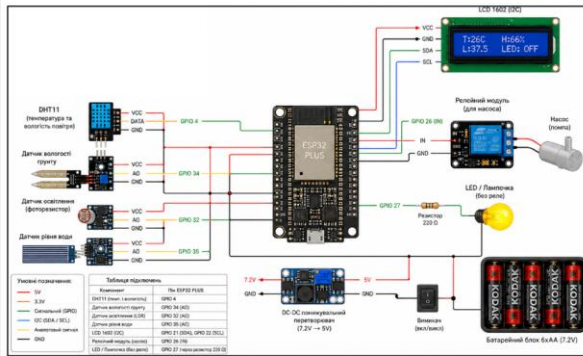


Рис.22. Схема підключення компонентів системи «Розумного поливу»

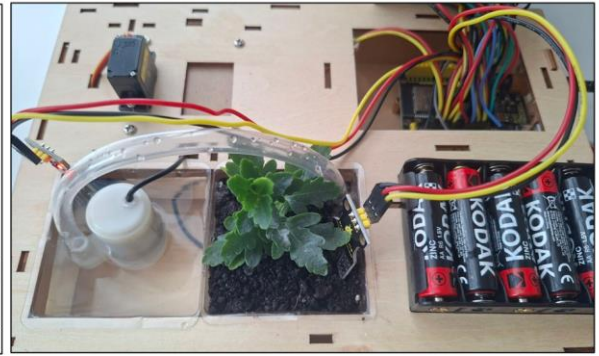


Рис.23. Підсистема автоматичного поливу рослин на базі ESP32 PLUS

```
int soilRaw = analogRead(SOIL_PIN);
int soilPercent = constrain(map(soilRaw, 4095, 2100, 0, 100), 0, 100);
int waterMl = constrain(map(analogRead(WATER_PIN), 0, 4095, 0, 500), 0, 500);
if (soilPercent < 30 && soilPercent > 2) {
    digitalWrite(RELAY_PIN, HIGH);
    delay(2000);
    digitalWrite(RELAY_PIN, LOW);
}
```

Рис.24. Фрагмент програмного коду керування системою автоматичного поливу на базі ESP32

РОЗРОБКА ВЕБІНТЕРФЕЙСУ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ФУНКЦІЙ ДИСТАНЦІЙНОГО КЕРУВАННЯ СИСТЕМОЮ «РОЗУМНА ФЕРМА»

8

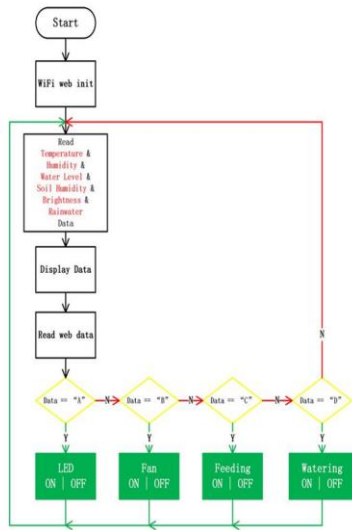


Рис. 25. Алгоритм роботи вебкеруваної системи розумної ферми

```
#include <WiFi.h>
const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

void setup() {
    Serial.begin(9600);
    //Initialize WiFi
    WiFi.begin(ssid, password);
    //Scan for WiFi. If connection fails, stay in connecting, and execute "while" loop
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
    //Connected. Print the IP address
    Serial.println("Connected to WiFi");
    Serial.println(WiFi.localIP());
}

void loop() {
}
```

Рис.26. програмний код для налаштування та перевірки роботи мережного з'єднання системи

```
const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

void setup() {
    Serial.begin(9600);
    //Initialize WiFi
    WiFi.begin(ssid, password);
    //Scan for WiFi. If connection fails, stay in connecting, and execute "while" loop
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
    //Connected. Print the IP address
    Serial.println("Connected to WiFi");
    Serial.println(WiFi.localIP());
}

void loop() {
    //Read data from sensors
    //Read Temperature & Humidity
    float temperature = readTemperature();
    float humidity = readHumidity();
    //Read Water Level
    int waterLevel = readWaterLevel();
    //Read Soil Humidity
    float soilHumidity = readSoilHumidity();
    //Read Brightness
    float brightness = readBrightness();
    //Read Rainwater
    float rainwater = readRainwater();
    //Display data
    displayData(temperature, humidity, waterLevel, soilHumidity, brightness, rainwater);
    //Read web data
    String webData = readWebData();
    //Check conditions
    if (webData == "1") {
        digitalWrite(LED_PIN, HIGH);
    } else if (webData == "2") {
        digitalWrite(LED_PIN, LOW);
    } else if (webData == "3") {
        digitalWrite(LED_PIN, HIGH);
    } else if (webData == "4") {
        digitalWrite(LED_PIN, LOW);
    }
}
```

Рис.27 Програмний код веб-сайту

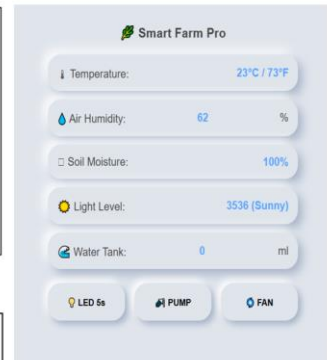


Рис.28 Відображення вебсторінки ESP32 у браузері



Рис.29. Меню керування виконавчими модулями системи

ВИСНОВКИ

9

У результаті виконання дипломної роботи було розроблено автоматизовану систему «Розумна ферма» на базі технологій Інтернету речей. Створена система забезпечує автоматичний моніторинг параметрів навколишнього середовища, керування поливом, вентиляцією та освітленням, а також дистанційний контроль через вебінтерфейс.

У ході дослідження проаналізовано сучасні IoT-технології, обґрунтовано вибір апаратних і програмних засобів та реалізовано апаратно-програмний комплекс на базі мікроконтролера ESP32.

Проведене тестування підтвердило працездатність і стабільність роботи системи, а отримані результати засвідчили ефективність використання технологій Інтернету речей для автоматизації процесів сучасного фермерства.

АПРОБАЦІЯ РЕЗУЛЬТАТІВ

10



1. Юхимович С. В. Сучасний технологічний прогрес і його вплив на апаратні системи побутового та промислового призначення // матеріали III всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» – Київ, 2026



2. Юхимович С. В. Апаратні платформи IoT у цифровій трансформації розумних міст та промисловості // матеріали VII Науково-технічної конференція «Сучасний стан та перспективи розвитку IoT». – Київ, 2026

ДЯКУЮ ЗА УВАГУ, ДОПОВІДЬ ЗАВЕРШЕНО