

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ**

**Навчально-науковий інститут кібернетики, інформаційних технологій та
інженерії**

"До захисту допущена"

Зав. кафедри комп'ютерних наук та
прикладної математики

« ___ » _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

**Розробка додатку для моніторингу та керування станом мініпарника на
основі мікроконтролера**

Виконав: Гунько Владислав Андрійович

(прізвище, ім'я, по батькові)

група ПЗ-41

(підпис)

Керівник: ст. викладач Зубик. Я. Я.

(науковий ступінь, вчене звання, посада, прізвище, ініціали)

(підпис)

ЗМІСТ

РЕФЕРАТ	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕХНОЛОГІЙ МОНІТОРИНГУ МІНІПАРНИКІВ	8
1.1 Роль мініпарників у сільському господарстві та садівництві	8
1.2 Сучасні технології вирощування у мініпарниках	10
1.3 Вплив температури та вологості ґрунту на розвиток рослин	11
1.4 Проблеми традиційних методів контролю мініпарників	13
1.5 Інтернет речей	14
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ МІНІПАРНИКОМ.....	18
2.1 Реалізація мікроконтролерної системи	18
2.2 Плата Arduino UNO	18
2.3 Онлайн сервіс Tinkercad.....	26
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ДЛЯ СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ МІНІПАРНИКОМ.....	32
3.1 Реалізація серверної частини веб-додатка	32
3.2 Реалізація веб-інтерфейсу для моніторингу даних	33
3.3 Функціональні можливості веб-додатку	35
ВИСНОВКИ	41
ДОДАТКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53

РЕФЕРАТ

Кваліфікаційна роботи: с. 53, 15 малюнків, 5 додатків, 9 джерел

Метою кваліфікаційної роботи є розробка додатку для моніторингу та керування станом мініпарника на основі мікроконтролерної системи.

Об'єкт дослідження – система моніторингу і керування в умовах мініпарника.

Предмет дослідження – розробка інтегрованого програмно-апаратного комплексу для моніторингу та керування параметрами середовища (температура, вологість ґрунту) в мініпарниках на базі мікроконтролерної технології.

Методи дослідження – методи програмування мікроконтролерів Arduino, UNO R3 для реалізації програмного забезпечення, технології веб-розробки HTML5, CSS3, JavaScript, Python для створення клієнтського і серверного компонентів веб-додатку.

Ця робота є актуальною з урахуванням зростаючої потреби у стабільному і якісному виробництві продуктів харчування, що вимагає точного контролю параметрів середовища для забезпечення оптимальних умов вирощування рослин у мініпарниках.

Ключові слова: мініпарник, мікроконтролер, моніторинг, керування, веб-додаток, IoT, температура, вологість.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- Arduino IDE** – інтегроване середовище розробки для програмування мікроконтролерів Arduino.
- Python** – високорівнева мова програмування загального призначення, використовується для розробки серверної частини та обробки даних.
- HTML** – гіпертекстова мова розмітки, використовується для створення структури веб-сторінок.
- CSS** – каскадні таблиці стилів, мова для опису вигляду і форматування веб-документів.
- JavaScript** – мова програмування, що використовується для створення інтерактивних веб-сторінок та взаємодії з користувачем.
- Flask** – мікрофреймворк для Python, використовується для створення веб-додатків та API.
- Chart.js** – JavaScript бібліотека для візуалізації даних за допомогою графіків та діаграм.

ВСТУП

У контексті зростаючої потреби у стабільному та якісному виробництві продуктів харчування, інноваційні підходи в сфері сільського господарства набувають особливого значення. Використання мініпарників дозволяє контролювати умови вирощування рослин незалежно від зовнішніх факторів, що є критично важливим для забезпечення врожайності та якості продукції. Однак, для досягнення максимальної ефективності, такі системи потребують точного моніторингу та автоматизованого управління ключовими параметрами середовища, насамперед температурою та вологістю ґрунту.

Актуальність теми дослідження зумовлена необхідністю оптимізації умов вирощування рослин в мініпарниках, зокрема, підтримання оптимальних рівнів температури та вологості ґрунту. Ці параметри безпосередньо впливають на розвиток кореневої системи, засвоєння поживних речовин та загальний стан рослин. Автоматизація процесів моніторингу та управління цими параметрами не тільки підвищує ефективність вирощування, але й зменшує трудові витрати, мінімізує людські помилки та сприяє економії ресурсів. Крім того, розвиток технологій Інтернету речей (IoT) та веб-технологій відкриває нові можливості для створення гнучких систем управління, доступних через інтернет з будь-якого пристрою.

Існуючі рішення для управління мініпарниками часто мають обмежений функціонал, високу вартість або не надають можливості дистанційного моніторингу та управління. Тому розробка веб-додатку для моніторингу та керування температурою і вологістю ґрунту в мініпарнику на основі мікроконтролера є актуальною науково-практичною задачею, яка має потенціал для впровадження як у малих фермерських господарствах, так і для домашнього використання.

Мета роботи полягає у створенні інтегрованого програмно-апаратного комплексу, що включає веб-додаток та систему на базі мікроконтролера, для

ефективного моніторингу та автоматизованого управління температурою та вологістю ґрунту в мініпарнику.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Проаналізувати існуючі методи та технології контролю температури та вологості ґрунту в мініпарниках, визначити їх переваги та недоліки.
2. Розробити архітектуру системи, що включатиме мікроконтролер, сенсори температури та вологості ґрунту та канали зв'язку.
3. Розробити алгоритми та програмне забезпечення для мікроконтролера, які забезпечать точний збір даних з сенсорів, їх первинну обробку, керування та обмін даними з веб-сервером.
4. Спроекувати та реалізувати серверну частину веб-додатку, яка забезпечить прийом даних від мікроконтролера, їх обробку, зберігання в базі даних та надання через API.
5. Розробити клієнтську частину веб-додатку з адаптивним, інтуїтивно зрозумілим інтерфейсом для візуалізації даних в реальному часі, налаштування параметрів та управління інформацією, яку передає система.
6. Реалізувати систему сповіщень про критичні зміни температури або вологості ґрунту.
7. Провести тестування розробленої системи в реальних умовах, оцінити її ефективність та вплив на розвиток рослин.

У роботі використовуються методи системного аналізу, методи програмування мікроконтролерів (Arduino, UNO R3), технології веб-розробки (HTML5, CSS3, JavaScript, Python), бази даних (SQLite, MySQL), а також підходи до інтеграції апаратного та програмного забезпечення в рамках концепції IoT.

Наукова новизна роботи полягає у розробці інтегрованого підходу до управління мініпарника за допомогою мікроконтролерної системи, яка поєднує

точний контроль ключових параметрів ґрунту та температури з доступністю управління через веб-інтерфейс.

Практична цінність роботи полягає у створенні доступної та простої у використанні системи для моніторингу ключових параметрів мініпарника. Розроблений веб-додаток надає користувачам можливість дистанційно відслідковувати температуру та вологість ґрунту в реальному часі, що є критично важливим для здоров'я рослин. Це дозволяє оперативно реагувати на зміни умов, запобігаючи стресу рослин від перегріву, переохолодження або нестачі вологи. Крім того, аналіз зібраних даних може допомогти оптимізувати ручне управління системами обігріву та поливу, що потенційно зменшить витрати ресурсів та підвищить якість урожаю. Доступність веб-інтерфейсу з будь-якого пристрою з браузером (комп'ютер, планшет, смартфон) забезпечує зручність користування та дозволяє контролювати стан мініпарника навіть за відсутності прямого доступу до нього.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕХНОЛОГІЙ МОНІТОРИНГУ МІНІПАРНИКІВ

1.1 Роль мініпарників у сільському господарстві та садівництві

Мініпарники, також відомі як домашні теплиці, є важливою частиною сучасного сільського господарства та садівництва. Їх використання значно розширюється, оскільки вони забезпечують оптимальні умови для росту рослин незалежно від зовнішніх погодних умов. У цьому розділі ми досліджуємо, як мікротеплиці сприяють розвитку сільського господарства та садівництва, які переваги вони пропонують, і аналізуємо їхню роль у забезпеченні продовольчої безпеки та стійкості.

Переваги мініпарників:

Контрольоване середовище. Мініпарник дозволяє створити і підтримувати ідеальні умови для росту рослин. Це включає контроль температури, вологості, освітлення та вентиляції. Таким чином, рослини можуть рости в стабільному середовищі, допомагаючи покращити врожайність і якість продукції.

Захист від несприятливих погодних умов. Мініпарники забезпечують захист рослин від несприятливих погодних умов, таких як заморозки, сильні вітри, зливи та снігопади. Це особливо важливо в регіонах з суворим кліматом, де вирощування рослин у відкритому ґрунті може бути ускладненим або навіть неможливим.

Ефективне використання ресурсів. В мініпарниках можливо більш ефективно використовувати ресурси, такі як вода і добрива. Системи автоматизованого поливу і дозування добрив дозволяють точно контролювати їх кількість, що сприяє економії ресурсів та зменшенню негативного впливу на навколишнє середовище.

Екологічні переваги. Мініпарники також мають позитивний вплив на навколишнє середовище. Вони сприяють зменшенню використання пестицидів та гербіцидів, оскільки контрольоване середовище дозволяє краще управляти шкідниками і захворюваннями. Крім того, ефективні системи поливу допомагають зменшити витрати води, що є важливим у регіонах з обмеженими водними ресурсами.

Розширення асортименту культур. Завдяки контролю над мікрокліматом, мініпарники дозволяють вирощувати культури, які зазвичай не ростуть в даному кліматичному регіоні. Це відкриває нові можливості для фермерів та садівників, дозволяючи їм експериментувати з новими видами рослин і розширювати асортимент продукції.

Вплив на продовольчу безпеку. Мініпарники відіграють важливу роль у забезпеченні продовольчої безпеки, особливо в умовах змін клімату та зростаючого населення. Вони сприяють стабільному виробництву харчових продуктів, незалежно від сезонних змін та екстремальних погодних умов. Це дозволяє підтримувати постійне постачання свіжих овочів та фруктів протягом усього року, що є критично важливим для задоволення потреб населення в здоровому харчуванні.

Економічні аспекти. Застосування мініпарників може бути економічно вигідним для малих і середніх фермерських господарств. Хоча початкові інвестиції у будівництво мініпарника можуть бути значними, довгострокові переваги включають збільшення врожайності, покращення якості продукції та зменшення ризиків, пов'язаних з кліматичними умовами. Крім того, мініпарники можуть сприяти розвитку місцевих економік, створюючи нові робочі місця і підтримуючи фермерські громади.

Мініпарники відіграють ключову роль у сучасному сільському господарстві та садівництві, забезпечуючи контрольоване середовище для вирощування рослин, захист від несприятливих погодних умов та ефективне

використання ресурсів. Вони сприяють забезпеченню продовольчої безпеки, економічному розвитку та зменшенню негативного впливу на навколишнє середовище. Усе це робить мініпарники важливим елементом сталого сільського господарства майбутнього.

1.2 Сучасні технології вирощування у мініпарниках

Сучасні технології вирощування у мініпарниках представляють собою велику революцію в аграрному секторі, що сприяє покращенню ефективності та продуктивності рослинного виробництва. Один з ключових аспектів цих технологій - це гідропоніка та аеропоніка, що дозволяють вирощувати рослини без використання ґрунту. Гідропоніка базується на системах зволоження та живлення рослин спеціальними розчинами поживних речовин у воді, а аеропоніка використовує спреювання поживних розчинів на кореневу систему рослин у вигляді туману. Ці методи дозволяють забезпечити рослини всіма необхідними речовинами без негативного впливу ґрунту.

Додатково, використання штучного світла стає все більш популярним у мініпарниках. Сучасні LED-лампи, фітолампи та програмовані системи освітлення забезпечують рослини необхідним спектром світла для фотосинтезу та розвитку. Це особливо важливо у зимовий період або для культур, які потребують додаткового освітлення для оптимального росту.

Ще одним ключовим аспектом сучасних технологій є використання сенсорів та автоматизація процесів управління. Сенсори, що вимірюють температуру, вологість, рН-рівень та інші параметри, дозволяють точно контролювати умови у мініпарнику. Автоматизовані системи управління на основі зібраних даних надають можливість автоматично регулювати полив, освітлення та інші параметри в залежності від потреб рослин, що дозволяє забезпечити їм оптимальні умови для росту та розвитку.

Не менш важливою є роль інноваційних матеріалів, таких як біофарб та гідрогельні полімери, у вирощуванні рослин у мініпарниках. Вони дозволяють

зберігати вологу та живильні речовини у ґрунті, що є особливо важливим у регіонах з недостатньою вологою або для збереження ресурсів у вирощуванні.

Узагальнюючи, сучасні технології вирощування у мініпарниках дозволяють забезпечити рослини оптимальними умовами для росту та розвитку, зменшуючи споживання ресурсів і покращуючи ефективність сільського господарства.

1.3 Вплив температури та вологості ґрунту на розвиток рослин

Температура і вологість ґрунту є ключовими факторами, що впливають на розвиток і ріст рослин. Ці параметри визначають швидкість фізіологічних процесів у рослинах, таких як поглинання води, мінералів, фотосинтез та дихання. Оптимальні умови температури та вологості ґрунту покращують врожайність і якість продукції.

Вплив температури ґрунту

Фізіологічні процеси в рослинах. Температура ґрунту впливає на швидкість росту коренів та їх здатність поглинати воду та мінеральні речовини. При оптимальних температурах активність ферментів, що беруть участь у метаболічних процесах, є найвищою. Наприклад, для більшості овочевих культур оптимальна температура ґрунту становить від 18 до 24 °С.

Затримка росту при низьких температурах. При низьких температурах (менше 10 °С) активність коренів значно знижується, що призводить до уповільнення росту рослин і зниження врожайності. В таких умовах рослини можуть страждати від дефіциту поживних речовин і води, оскільки корені не можуть ефективно їх поглинати.

Негативний вплив високих температур. Занадто висока температура ґрунту (понад 30 °С) також може мати негативний вплив на рослини. Вона призводить до підвищення випаровування води з ґрунту і збільшення ризику

зневоднення рослин. Крім того, висока температура може пошкодити корені і знизити їхню функціональність.

Вплив вологості ґрунту

Забезпечення водою. Волога ґрунту є основним джерелом води для рослин. Вода необхідна для транспортування поживних речовин, фотосинтезу, підтримки тургору клітин та інших життєво важливих процесів. Оптимальний рівень вологості ґрунту забезпечує найкращі умови для росту і розвитку рослин.

Наслідки дефіциту вологи. Недостатня вологість ґрунту призводить до зневоднення рослин, що може спричинити в'янення, зниження швидкості фотосинтезу і, в кінцевому результаті, зниження врожайності. Особливо чутливі до дефіциту вологи молоді рослини і сіянці, які мають менш розвинену кореневу систему.

Наслідки надмірної вологості. Занадто висока вологість ґрунту також може бути шкідливою для рослин. Це може призвести до гниття коренів, поширення грибкових інфекцій та зниження доступності кисню для кореневої системи. Погана аерація ґрунту при надмірній вологості спричиняє анаеробні умови, що негативно впливає на ріст коренів і загальний стан рослин.

Оптимальні умови для розвитку рослин

Температурні умови. Для більшості культур оптимальна температура ґрунту коливається в межах 18-24 °С. Наприклад, для томатів та огірків оптимальна температура становить близько 20-25 °С, тоді як для капусти та шпинату вона може бути нижчою, близько 15-20 °С.

Вологість ґрунту. Оптимальна вологість ґрунту залежить від типу культури і стадії її розвитку. Для більшості овочевих культур вологість ґрунту повинна бути на рівні 60-70% від повної польової вологості. Це забезпечує достатню кількість води для фізіологічних процесів і запобігає надмірному випаровуванню.

Мініпарники дозволяють ефективно контролювати температуру та вологість ґрунту, що забезпечує оптимальні умови для вирощування рослин. Використання автоматизованих систем моніторингу і керування дозволяє підтримувати стабільні параметри, що позитивно впливає на ріст і розвиток культур. Це включає використання датчиків вологості та температури, автоматизованих систем поливу і вентиляції.

1.4 Проблеми традиційних методів контролю мініпарників

Традиційні методи контролю мініпарників, особливо ручний контроль, часто стикаються з певними обмеженнями та недоліками, які можуть вплинути на ефективність вирощування рослин та врожайності. Ручний контроль, наприклад, може бути неточним через людський фактор, що призводить до несправедливих вимірювань та неадекватних рішень. Затримки в реагуванні на зміни у середовищі мініпарника також можуть виникати через ручні втручання, особливо коли потрібно контролювати багато параметрів одночасно або коли мініпарник розташований віддалено від місця проживання.

Крім того, ручний контроль вразливий до людських помилок, втоми та забудькування, що може стати причиною недостатнього або неправильного контролю параметрів і, в результаті, вплинути на здоров'я рослин та якість врожаю.

Щодо простих автоматичних систем без моніторингу, вони також мають свої обмеження. Такі системи можуть бути менш точними у порівнянні зі складнішими системами, які мають можливість постійного моніторингу та аналізу даних. Це може вплинути на точність регулювання параметрів і створити неоптимальні умови для росту рослин.

Також важливо враховувати, що прості автоматичні системи можуть бути обмеженими у своїх можливостях реагування на зміни в середовищі, особливо на раптові зміни температури чи вологості. Відсутність аналітичних можливостей у

таких системах також ускладнює процес управління мініпарником та підтримки оптимальних умов для росту культур.

Отже, для забезпечення ефективного та точного контролю у мініпарниках важливо використовувати сучасні автоматизовані системи, які мають можливість постійного моніторингу, аналізу даних та автоматичного регулювання параметрів в залежності від умов у середовищі та потреб рослин.

1.5 Інтернет речей

Інтернет речей (IoT) у сучасному агробізнесі

Інтернет Речей (IoT) - це концепція, що революціонує підхід до управління та моніторингу у сільському господарстві. Вона передбачає підключення фізичних об'єктів, таких як сенсори, пристрої вимірювання, механізми та інші, до Інтернету для збору, обробки та обміну даними в реальному часі без необхідності прямого втручання людини [1].

Одним із основних застосувань IoT у сільському господарстві є регулювання росту та розвитку рослин. За допомогою датчиків, розміщених у землі, ви можете отримати інформацію про вологість, температуру та склад поживних речовин. Це сприяє створенню необхідних умов для оптимального росту та розвитку рослин, а також своєчасному виявленню проблем зі здоров'ям рослин та реагування на них.

Окрім моніторингу умов росту рослин, IoT можна використовувати для спостереження за станом обладнання, пов'язаного з сільськогосподарським бізнесом. За допомогою датчиків, вбудованих в обладнання, можна розпізнати потенційні проблеми та необхідний ремонт. Це сприяє продовженню безперебійної роботи обладнання та скорочує вартість і час його регулярного обслуговування.

Вплив Інтернету речей на сільське господарство: переваги та виклики

Однією з головних переваг використання IoT у сільському господарстві є підвищення ефективності та продуктивності. За допомогою технологій IoT фермерам легше регулювати умови росту та розвитку рослин, стан тварин та техніки. Це сприяє збільшенню врожайності та продуктивності з гектара землі та зниженню витрат.

Ще одна суттєва перевага використання IoT у сільському господарстві полягає в мінімізації ризиків. За допомогою технологій IoT можна виявити проблеми з ростом рослин, станом тварин і обладнання до пошкодження, що дозволяє швидко реагувати та мінімізувати ризик пошкодження та втрати.

Використання технологій IoT також може сприяти підвищенню прибутковості агробізнесу. За допомогою IoT можна підвищити продуктивність, знизити витрати та підвищити якість, що призводить до більшого прибутку на одиницю продукції.

Однією з найбільших перешкод для використання Інтернету речей у сільському господарстві є висока вартість технологій та обладнання. Це може бути особливо складним для малих і середніх агробізнесів, які не мають фінансових можливостей придбати дороге обладнання.

Ще однією перешкодою для використання IoT у сільському господарстві є складність впровадження та підтримки технології. учасники агробізнесу повинні мати глибоке розуміння того, як використовувати IoT, а також технічні навички для усунення несправностей і обслуговування датчиків та інших пристроїв.

Іншою проблемою, пов'язаною з використанням IoT у сільському господарстві, є безпека даних. Збір і передача інформації в Інтернеті може бути вразливою до кібератак або крадіжки даних, що може становити загрозу для конфіденційності інформації щодо виробництва та передачі даних.

Впровадження технологій Інтернету речей у сільському господарстві може сприяти підвищенню продуктивності, забезпечити якість продукції та збільшити дохід. Однак існують обмеження та проблеми, пов'язані з впровадженням і підтримкою Інтернету речей, а також із забезпеченням безпеки даних.

Щоб максимально ефективно використовувати IoT у сільському господарстві, необхідно розглянути ці проблеми та перешкоди та розробити плани, щоб зменшити витрати та забезпечити безпеку даних. Це дозволить підвищити конкурентоспроможність агробізнесу та задовольнити потреби ринку.

Крім того, необхідно враховувати особливості кожного виробничого процесу та типу продукту, щоб визначити, які конкретні рішення для IoT принесуть користь корпорації.

Наприклад, використання Інтернету речей у сільському господарстві може полегшити визначення часу зрошення, знизити вартість ресурсів і підвищити врожайність. У тваринництві IoT може полегшити моніторинг здоров'я та поведінки тварин, це дозволить швидко виявляти захворювання та зменшити вартість лікування. Також, вкрай важливо переконатися, що персонал пройшов відповідну підготовку та використовувалися методи керування даними для забезпечення безпеки та захисту від кібератак.

Загалом IoT може слугувати потужною платформою для розвитку агробізнесу та виробництва екологічно чистих продуктів харчування. Однак для того, щоб впровадження IoT було успішним, важливо враховувати проблеми та перешкоди, пов'язані з технологією, і розробляти плани та стратегії.

Застосування інтернету речей (IoT) у моніторингу та керуванні мініпарниками

Інтернет речей (IoT) перетворює традиційні мініпарники в інтелектуальні системи, що забезпечують автоматизований моніторинг та керування умовами росту рослин. IoT впроваджує розумні сенсори, мережеві пристрої та аналітичні

системи для збору, обробки та аналізу даних, що дозволяє оптимізувати процеси вирощування та підвищувати врожайність.

Сенсори та датчики IoT. Перевага IoT полягає у використанні різноманітних сенсорів та датчиків для вимірювання параметрів росту рослин. Сенсори температури, вологості ґрунту, рівня освітлення, рН-рівня та інших параметрів дозволяють отримувати точні дані про умови в мініпарнику в реальному часі.

Аналітика даних та штучний інтелект. Зібрані дані від сенсорів передаються до аналітичних систем, де вони обробляються та аналізуються за допомогою методів штучного інтелекту. Аналіз даних дозволяє виявляти закономірності та патерни у розвитку рослин, що дозволяє підбирати оптимальні стратегії управління мініпарником.

Автоматизоване управління. На основі аналізу даних та патернів росту рослин, системи IoT автоматично регулюють умови в мініпарнику. Наприклад, система може автоматично контролювати полив, освітлення, вентиляцію та інші параметри, щоб забезпечити оптимальні умови для росту та розвитку культур.

Збір даних та моніторинг віддалено. Однією з переваг IoT є можливість віддаленого моніторингу та керування. Фермер чи господар може в реальному часі отримувати дані про стан мініпарника та вносити необхідні корективи в роботу системи з будь-якої точки, де є доступ до Інтернету.

Безпека та надійність. При використанні IoT у мініпарниках, важливу роль відіграє питання безпеки та надійності систем. Застосування сучасних технологій шифрування даних, захист від кібератак та створення резервних копій даних є ключовими аспектами для забезпечення безпеки та стабільності роботи систем IoT у мініпарника.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ МІНІПАРНИКОМ

2.1 Реалізація мікроконтролерної системи

Для реалізації мікроконтролерної системи для збору даних та керування станом мініпарника необхідний сам мікроконтролер, датчик температури та вологості, а також LCD-екран для відображення отриманих даних. Найкраще використовувати платформу Arduino, оскільки ця плата є легко програмованою та доступною за ціною. До неї необхідно підключити екран та датчики, написати програмний код, і система буде готова для встановлення у мініпарник.

Зазвичай, існуючі мініпарники представлені як невеликі конструкції або агротканини для жителів сільської місцевості. У них рідко передбачена наявність системи збору та керування параметрами середовища. Це означає, що фермери та садівники змушені використовувати ручні методи контролю, які можуть бути неточними та неефективними.

Використання мікроконтролерної системи на основі Arduino дозволяє автоматизувати процеси моніторингу та керування, забезпечуючи точне і своєчасне зчитування даних про стан мініпарника. Це дозволяє оперативно реагувати на зміни умов, що в свою чергу сприяє покращенню росту рослин та збільшенню врожайності. Автоматизовані системи можуть зменшити людський фактор, що знижує ризик помилок і підвищує загальну ефективність вирощування.

2.2 Плата Arduino UNO

Arduino Uno [2] — доступна як плата з відкритим кодом, яка використовує мікроконтролер ATmega328p у платі. Плата Arduino UNO використовується в електронних проектах і схемотехніці. Він містить 14 контактів вводу/виводу (6 з яких є аналоговими входами), 1 роз'єм живлення, 1 роз'єм USB, 1 кнопку скидання, 1 роз'єм ICSP та інші компоненти. Усі перераховані компоненти підключаються до плати Arduino UNO, щоб вона працювала та могла

використовуватися в проектах. Плату можна заряджати через порт USB або плату можна заряджати безпосередньо від джерела постійного струму.



Рис. 2.1 Плата Arduino UNO R3

Плата Arduino UNO в основному використовується початківцями, які можуть використовувати її в електронних проектах і програмувати на цій платі. Дошка містить регулярні інновації та виправлення помилок у дизайні дошки, що робить дошку придатною для використання в проектах. Ця дошка вважається стандартною дошкою, яку найчастіше використовують новачки у своїх проектах. Плата Arduino UNO в основному призначена для заміни інших продуктів Arduino з наступних причин.

Оскільки плату можна легко підключити до іншої комп'ютерної системи через USB-порт. Порт USB, підключений до плати, служить двом цілям. Його можна використовувати для живлення демонстраційної плати або як послідовний пристрій для підключення демонстраційної плати до комп'ютерної системи.

Виводи плати можуть працювати від постійної напруги 5 В, а цифрові та аналогові виводи використовуються для регулювання напруги на платі.

Завдяки простому дизайну плати її можуть використовувати кілька користувачів, і спільнота підтримує плату Arduino UNO.

Плата Arduino UNO має ряд апаратних компонентів і здатність інтерфейсу з цими пристроями. Пристрій включає Bluetooth, Інтернет, керування двигуном тощо.

Головне, що відрізняє плату Arduino UNO від інших плат Arduino, це ціновий фактор. Ціна цієї плати найнижча в порівнянні з іншими продуктами Arduino. Це головна причина, чому новачки віддають перевагу цій платі перед іншими платами.

Особливості плати Arduino UNO

Функції плати Arduino UNO:

1. Плата підтримує інтерфейс USB, що дозволяє платі працювати як послідовний пристрій і забезпечує можливість підключення плати до інших інтерфейсів. Інтерфейс USB також використовується для живлення плати розробки.

2. Плата Arduino UNO містить чіп, який підключається безпосередньо до порту USB і діє як віртуальний послідовний порт для комп'ютерної системи. Це робить зв'язок дуже плавним і допомагає дощці підключатися до різних типів комп'ютерних систем.

3. Мікроконтролер, який використовується в платі Arduino UNO ATmega328, легко доступний і простий у використанні. Якщо мікросхема контролера пошкоджена або несправна, її можна зняти та замінити. Інші плати Arduino не пропонують такої гнучкості.

4. Ця панель розробки надається як інструмент з відкритим кодом, який має свої переваги, оскільки велика кількість користувачів використовує цю плату розробки та допомагає у вирішенні інших проблем, пов'язаних із цією платою розробки. Це робить етапи налагодження, пов'язані з проектом, легкими.

5. Штири, які використовуються на платі, діють як генератори частоти близько 16 МГц, що корисно для більшості програм. Це не змінює швидкість мікроконтролера

6. На платі також є функція регулювання напруги, яка допомагає регулювати джерело живлення на платі. Плату можна живити безпосередньо без використання зовнішнього джерела живлення, для цього можна використовувати порт USB. В якості зовнішнього джерела живлення для плати можна використовувати блок живлення 12 В.

7. Виводи Arduino UNO містять 14 виводів цифрового типу та 6 виводів аналогового типу. Штифти забезпечують функціональність плати для підключення обладнання до плати. За допомогою них ви можете покращити функціональність свого комп'ютера.

8. Інструмент також має роз'єм ICSP, який допомагає обійти USB-порт і напряму підключити Arduino та діяти як послідовний пристрій.

9. Плата має загалом 32 КБ флеш-пам'яті для зберігання даних.

10. Плата також має вбудований світлодіод для полегшення процесу налагодження та допомагає знайти помилки в коді.

11. Плата розробки також має кнопку скидання, щоб допомогти перезапустити програму за допомогою плати розробки.

Основними характеристиками Arduino UNO є:

- Мікроконтролер ATmega328
- Робоча напруга 5 В
- Напруга живлення (рекомендована) 7-12 В
- Напруга живлення (гранична) 6-20 В
- Цифрові входи/виходи 14 (з них 6 можуть використовуватися як ШИМ-виходи)
- Аналогові входи 6
- Максимальний струм одного виводу 40 мА
- Flash-пам'ять 32 КБ (ATmega328), з яких 0.5КБ використовуються завантажувачем.
- SCRAM 2 КБ (ATmega328)
- EEPROM 1 КБ (ATmega328)
- Тактова частота 16 МГц

РК-дисплей (рідкокристалічний дисплей) — це плоский дисплей, який використовує рідкокристалічний дисплей як основну форму роботи. Світлодіоди мають широкий і різноманітний спектр споживчого та комерційного використання, оскільки вони зазвичай зустрічаються в смартфонах, телевізорах, комп'ютерних моніторах і панелях приладів.

РК-дисплеї були величезним кроком вперед у технологіях, які вони замінили, включаючи світлодіоди (світлодіоди) і плазмові дисплеї. РК-дисплеї дозволяють дисплеям бути набагато тоншими, ніж технологія електронно-променевої трубки (CRT). РК-дисплеї споживають набагато менше енергії, ніж

світлодіодні та газові дисплеї, оскільки вони блокують світло, а не випромінюють його. Світлодіод випромінює світло, а рідкі кристали в РК-дисплеї створюють зображення за допомогою підсвічування.

Оскільки РК-дисплеї замінили старі технології відображення, РК-дисплеї почали замінювати нові технології відображення, такі як OLED.

Принципи роботи дисплея LCD

РК-дисплеї [3] складаються з мільйонів пікселів. Якість дисплея зазвичай стосується кількості пікселів; наприклад, монітор 4К складається з 3840 x 2160 або 4096 x 2160 пікселів. Піксель складається з трьох субпікселів: червоного, синього та зеленого, які часто називають RGB. Різні кольори можуть бути створені, коли субпікселі всередині пікселя змінюють колірні комбінації. Оскільки всі пікселі на моніторі працюють разом, монітор може відтворювати мільйони різних кольорів. Коли пікселі швидко вмикаються і вимикаються, показується зображення.



Рис. 2.2 Символьний дисплей LCD (синій)

Кожен тип монітора керує пікселями по-різному; CRT, LED, LCD і новіші монітори мають різні опорні пікселі. Простіше кажучи, РК-дисплеї мають підсвічування та використовують рідкі кристали для обертання поляризованого світла для електронного ввімкнення та вимикання пікселів. Поляризаційні скляні фільтри розміщені спереду та позаду всіх пікселів, при цьому передній фільтр розміщено під кутом 90 градусів. Між двома фільтрами знаходиться рідкий кристал, який можна вмикати та вимикати електронним способом.

РК-дисплеї виготовляються з використанням пасивної або активної матриці. РК-дисплеї з активною матрицею також відомі як дисплеї на тонкоплівкових транзисторах (TFT). РК-дисплеї з пасивною матрицею мають сітку провідників з пікселями, розташованими на кожному перетині сітки. Електричний струм надсилається через два провідники в сітці для керування світлом від будь-якого пікселя. Активна матриця має транзистор на стику кожного пікселя, що потребує меншого струму для керування яскравістю пікселя. У результаті струм на дисплеї з активною матрицею можна вмикати та вимикати частіше, що призводить до скорочення часу оновлення екрана.

Деякі РК-дисплеї з пасивною матрицею мають подвійне сканування, тобто вони двічі сканують сітку за допомогою струму, тоді як одноразове сканування в оригінальній технології займає стільки ж часу. Однак активна матриця все ще є кращою технологією з двох.

Різниця між OLED і QLED

Зараз рідкокристалічні дисплеї випереджають інші технології відображення, але вони ще не зовсім у минулому. РК-дисплеї поступово замінюють OLED або органічні світлодіоди.

У OLED-дисплеях використовується одна скляна або пластикова панель, тоді як у РК-дисплеях використовуються дві скляні або пластикові панелі. Оскільки OLED-дисплеї не потребують підсвічування, як РК-дисплеї, OLED-пристрої, такі як телевізори, зазвичай тонші та мають глибший чорний колір, оскільки кожен піксель OLED підсвічується окремо. Якщо РК-монітор здебільшого чорний, але потрібно підсвітити лише невелику частину, уся задня панель усе ще освітлюється, що спричиняє витік світла на передню частину монітора. OLED-екрани уникають цього та мають кращий контраст, кращі кути огляду та менше енергоспоживання. Завдяки пластиковим панелям OLED-дисплеї можна згинати і складати і продовжувати працювати. Це можна побачити в таких смартфонах, як суперечливий Galaxy Fold або iPhone X, які загинають нижню частину дисплея так, що стрічковий кабель дисплея може простягатися в телефон, усуваючи потребу в нижній рамці.

Однак OLED-дисплеї, як правило, дорожчі і можуть страждати від вигорання екрана, як плазмові дисплеї.

QLED означає квантовий світлодіод і світлодіод із квантовими точками. Дисплеї QLED були розроблені компанією Samsung і їх можна знайти в новіших телевізорах. QLED працює найбільш подібно до LCD і все ще може вважатися

типом LCD. РК-дисплеї QLED додають шар плівки з квантовими точками, що значно покращує колір і яскравість порівняно з іншими РК-дисплеями. Плівки квантових точок складаються з малих кристалічних напівпровідникових частинок. Кристалічними напівпровідниковими частинками можна керувати за їх кольором.

Коли ви вибираєте між дисплеями QLED і OLED, QLED яскравіший і не вигоряє. Однак дисплеї OLED все ще мають кращу контрастність і глибший чорний колір, ніж дисплеї QLED.

Аналоговий датчик температури TMP36

На прикладі датчику температури виберемо один з наявних в мережі котрий підходить під Arduino.

TMP36 [4] - простий у засвоєнні, дешевий і досить точний інтегральний датчик температури з аналоговим виходом.



Рис. 2.3 Аналоговий датчик температури TMP36

Принцип роботи температурного датчика TMP36 простий: вихідна напруга на датчику пропорційна до вимірюваної температури, вираженої в градусах Цельсія. Всі датчики відкалібровуються за шкалою Цельсія при виготовленні.

TMP36 має похибку вимірювання ± 1 °C при температурі + 25 °C і близько ± 2 °C в діапазоні від -40 °C до + 125 °C. Для визначення температури використовується технологія твердотільної електроніки.

Датчик може працювати з одно- і двополярною напругою від 2,7 В до 5,5 В. Власний рівень розігріву вкрай низький за рахунок того, що струм становить не більше 50 мкА. У режимі сну датчик споживає не більше 0,5 мкА.

Датчик сумісний з будь-якими мікроконтролерами Arduino.

Датчик вологості ґрунту

Простий датчик [5], що дозволяє вимірювати вологість ґрунту або різних сипучих матеріалів. Датчик має два виходи, один з них - аналоговий та видає аналогове значення пропорційно рівню вологості контактної площадки, другий - дискретний та видає логічну "1" при досягненні певного рівня вологості, що налаштовується потенціометром. Завдяки цьому датчик вологості ґрунту легко підключити до мікроконтролерних плат типу Arduino, AVR, STM і т.д.



Рис. 2.4 Датчик вологості ґрунту

Датчик має антикорозійне покриття, що збільшує його довговічність. Може використовуватися в міні-теплицях, системах автоматичного поливу і т.д.

2.3 Онлайн сервіс Tinkercad

Tinkercad [6] — це безкоштовна програма для 3D-моделювання, відома своєю простотою використання. Він на 100% базується на Інтернеті, і ним може користуватися кожен, хто має підключення до Інтернету.



Рис. 2.5 Сервіс Tinkercad

Діти, педагоги та любителі використовують його, щоб створити все, що тільки можна уявити. Дизайни Tinkercad можна оживити за допомогою 3D-друку, лазерного різання або будівельних блоків.

Багато шкіл використовують його для навчання проєктів 3D дизайну, електроніки та блоків візуального коду.

3D-дизайн у Tinkercad

Tinkercad був запущений у 2011 році, щоб зробити 3D-дизайн доступним для широкого загалу. Проєктування в Tinkercad — чудовий спосіб вивчити основи моделювання твердого тіла.

Суцільне моделювання — це практика побудови об'єктів у їх оригінальній формі. Користувачі можуть перетягувати готові фігури в робочу область для створення об'єктів.

Комбінуючи наявні форми, користувачі можуть робити свої унікальні дизайни.

Електроніка у Tinkercad

Унікальні конструкції також використовують переваги вбудованих можливостей Tinkercad для використання руху та світла.

Користувачі можуть почати з стартової схеми або зробити власну, користуючися готовими збірками проводів. Віртуальні схеми — це хороший спосіб навчитися та потренуватися перед тим, як використовувати їх у реальному житті.

Блоки вбудованого коду або спеціальний C++ дозволяють запускати компоненти схеми. Користувач може коригувати та додатково визначати їх до

досягнення

бажаного

результату.

Блоки візуального коду у Tinkercad

Перетягніть разом готові блоки коду, щоб створити власні форми. Блоки коду дозволяють створювати 3D-проекти за допомогою візуального програмування.

Зміна порядку блоків коду може покращити деталі форми. Наприклад, у циліндричному блоці, вирізаному з коробчатого блоку, будуть пробиті отвори.

Щоб уникнути витрат часу, Tinkercad містить кілька прикладів блоків коду, які включено в проект. Усі попередньо розщеплені зразки знаходяться в розділі «Початковий».

Створення власних блоків коду передбачає зміну блоку коду, а потім натискання «Виконати», щоб спостерігати за змінами в просторі.

Створення системи зі збору даних та керування в Tinkercad

Після створення особистого облікового запису необхідно натиснути кнопку «Створити» (Create), після чого бачимо розділ із трьома пунктами. Виберіть опцію «Схеми» (Circuits) та вирушайте до робочого простору. У верхньому правому куті сторінки є розділ «Компоненти» (Components), де потрібно перемістити необхідні компоненти в робочу область.

Перелік компонентів:

1. Arduino UNO R3
2. Мала макетна плата
3. ЖК-екран 16x2
4. Потенціометр
5. Резистор
6. Датчик температури TMP36
7. Датчик вологості ґрунту

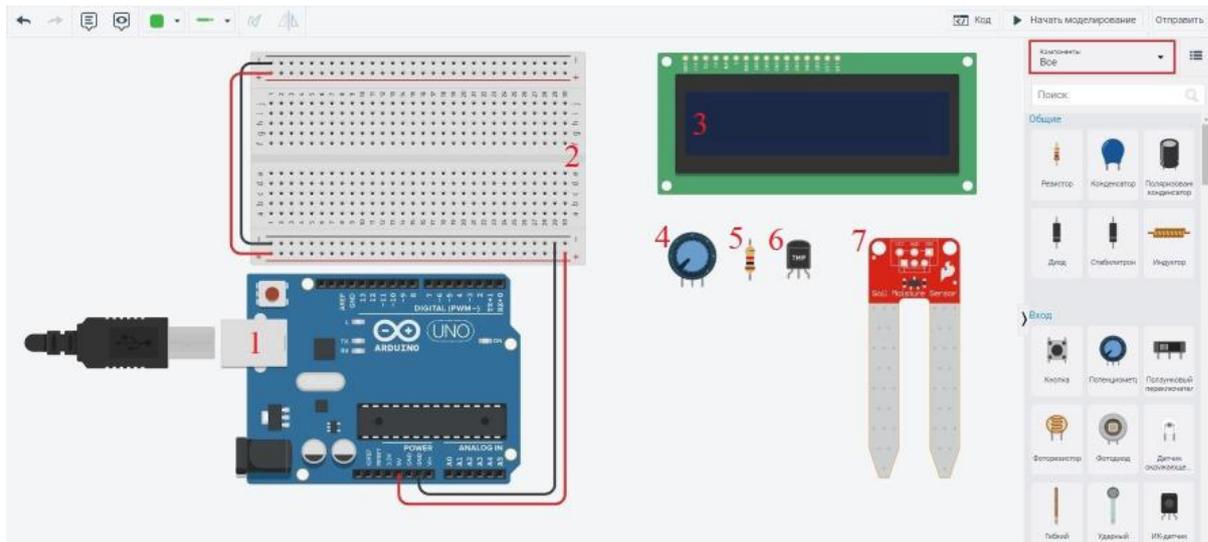


Рис. 2.6 Робочий простір з компонентами для схеми

Після вибору необхідних компонентів їх необхідно підключити до Arduino, потягнувши їх кінці до відповідних контактів Arduino на платі.

ЖК-екран:

- підключаємо негативну сторону РК-екрана до негативної сторони Arduino UNO R3, це створить спільне заземлення для електричного кола.
- підключаємо RS до контакту 12, E до контакту 11, а DB4, DB5, DB6 і DB7 до контактів 5, 4, 3, 2 на Arduino UNO R3, це полегшить передачу даних і керування РК-екраном.
- підключаємо VCC до негативної сторони маленької макетної плати, яка буде жити РК-екран.
- підключаємо V0 до f15 на маленькій макетній платі, це дозволить нам змінити контрастність РК-екрану.
- підключаємо мінусову сторону макетної плати до заземленого проводу, це активує РК-екран і режим читання або запису.
- підключаємо світлодіод до позитивного проводу на макетній платі та пінні j26, за допомогою якої можна керувати освітленням РК-екрану.

Потенціометр:

- З'єднання між j14, j15 і j16 полегшить використання потенціометра для зміни опору або напруги ланцюга.

Резистор:

- підключаємо j26 до негативної сторони маленької макетної плати, це дозволить нам використовувати з'єднання резистора з іншими компонентами або регулювати струм.

Датчик температури TMP36:

- підключаємо датчик температури до конкретних розеток e25, e26, e27, що дозволить нам використовувати його для підключення до аналогових вхідних розеток Arduino.

Датчик вологості ґрунту:

- підключаємо VCC до кабелю 5 В на Arduino UNO R3, який буде живити датчик вологості.
- підключаємо GND до контакту GND, який створить спільну землю для датчика та Arduino.
- підключаємо вхідний сигнал датчика вологості до виходу SIG, це дозволить нам зчитувати та вимірювати вологість з датчика.

Процедура написання коду для мікроконтролерної системи передбачає спочатку написання логіки пристрою, потім зчитування даних з датчиків і відображення їх на РК-екрані. Код реалізовано на мові програмування Arduino, що дозволяє використовувати апаратні компоненти з наданою бібліотекою. Після написання коду він компілюється та підключається до плати Arduino через порт USB.

Бібліотеки використовуються в Arduino IDE, щоб полегшити просте об'єднання кількох датчиків і компонентів. Наприклад, РК-дисплей 16x2 реалізовано за допомогою бібліотеки LiquidCrystal, ця бібліотека забезпечує прості засоби відображення інформації.

Переконавшись, що все підключено правильно, натисніть кнопку «Код» (Code) у верхньому правому куті, щоб відкрити текстовий редактор для написання програмного коду. Код програми наведено в додатку А. Після написання коду натисніть кнопку «Почати моделювання», щоб перевірити роботу проекту.

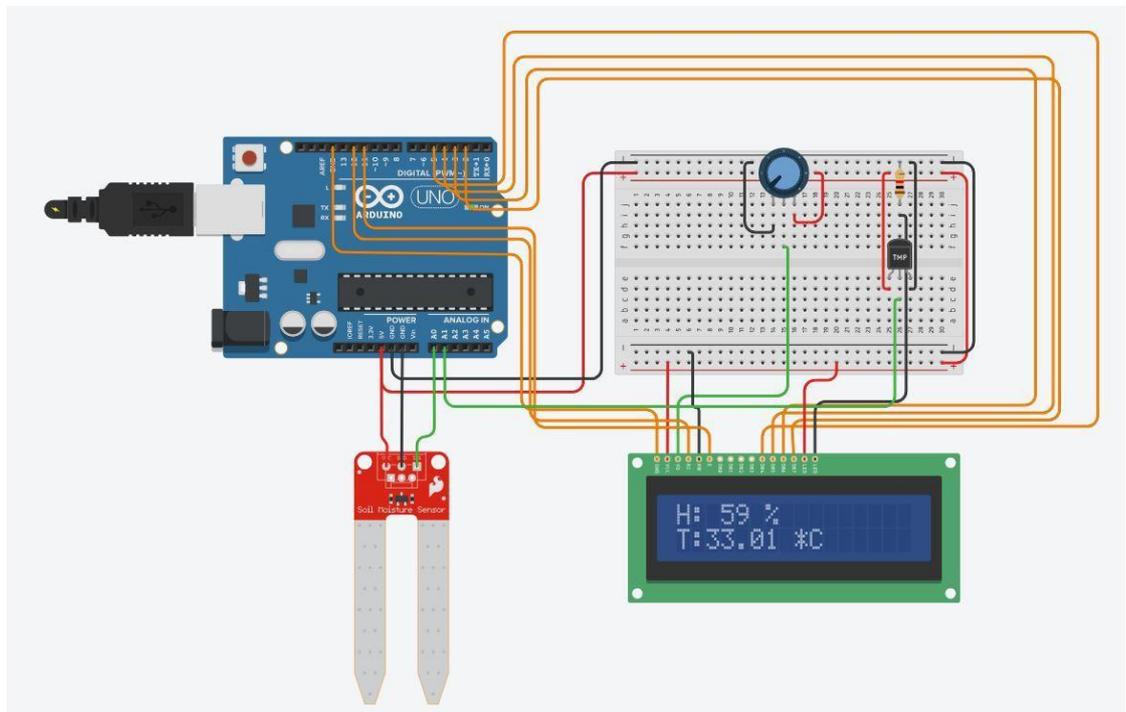


Рис. 2.7 Реалізація схеми системи

Після успішного моделювання в Tinkercad ми вирішили реалізувати систему в реальному житті. Зберіть компоненти відповідно до розробленої схеми та завантажте програмний код на плату Arduino через Arduino IDE.

Arduino IDE (Integrated Development Environment) є офіційним середовищем розробки для програмування плат Arduino. Він забезпечує зручний інтерфейс для написання, компіляції та завантаження коду на мікроконтролери. IDE підтримує мови програмування на основі C/C++ і містить бібліотеки для роботи з різними датчиками та модулями. Це середовище забезпечує швидке та ефективне впровадження системи моніторингу міні-теплиці.

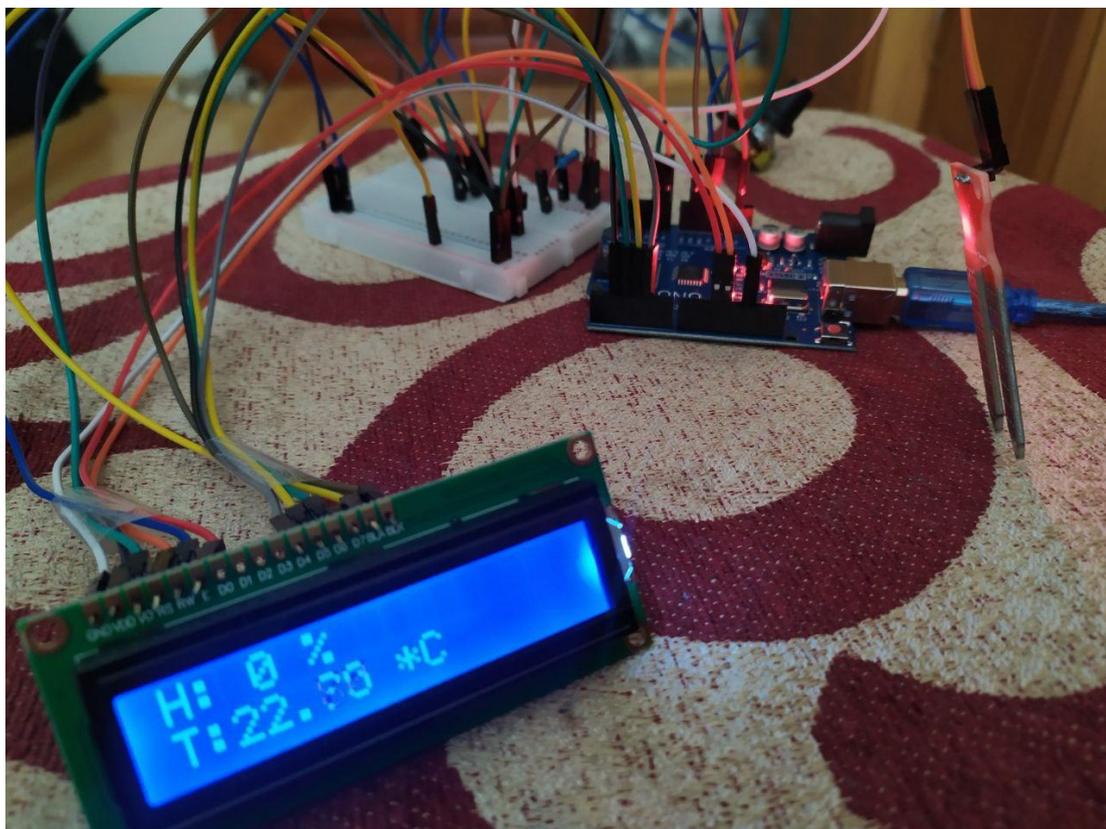


Рис. 2.8 Системи моніторингу та управління мініпарником

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ДЛЯ СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ МІНІПАРНИКОМ

3.1 Реалізація серверної частини веб-додатка

Після успішної реалізації апаратної частини системи моніторингу та управління мініпарником, необхідно створити для неї веб-додаток. Веб-додаток дозволить зручно відслідковувати дані з датчиків та керувати системою в режимі реального часу. Для цього потрібно створити серверну частину, яка буде обробляти запити від клієнта, взаємодіяти з мікроконтролером Arduino, зберігати дані та передавати їх у клієнтську частину.

Серверна частина веб-додатку написана на мові програмування Python із використанням мікрофреймворка Flask [7]. Flask є потужним інструментом для створення веб-додатків, забезпечуючи необхідний набір функцій для обробки запитів, маршрутизації та рендерингу шаблонів. Для забезпечення зв'язку в реальному часі між клієнтом та сервером використовується бібліотека Flask-SocketIO, яка додає підтримку WebSockets.



Рис. 3.1 Мікрофреймворк Flask

Для отримання даних з мікроконтролера Arduino використовується бібліотека PySerial. Вона дозволяє встановити серійне з'єднання з Arduino, зчитувати дані з датчиків температури та вологості, а також передавати їх у серверну частину для подальшої обробки. Після встановлення з'єднання з Arduino, дані зчитуються у режимі реального часу та перевіряються на валідність.

Структура серверної частини

Серверна частина складається з декількох основних компонентів:

1. Ініціалізація веб-сервера:

- a. Створюється об'єкт Flask для запуску веб-сервера.
 - b. Ініціалізується Flask-SocketIO [8] для забезпечення зв'язку в реальному часі.
2. Маршрутизація:
 - a. Визначається маршрут для головної сторінки, яка відображає інтерфейс користувача.
 3. Підключення до Arduino:
 - a. Встановлюється серійне з'єднання з Arduino на визначеному порту (наприклад, COM3) зі швидкістю 9600 бодів.
 4. Обробка та зберігання даних:
 - a. Дані зчитуються з Arduino, перевіряються на валідність за допомогою регулярних виразів та зберігаються у базу даних SQLite.
 - b. Створюється функція для збереження даних у базу даних, яка відкриває з'єднання з базою, виконує SQL-запит для вставки даних та закриває з'єднання.
 5. Передача даних у реальному часі:
 - a. Дані, отримані з Arduino, передаються у клієнтську частину через WebSockets. Це дозволяє відобразити актуальні дані на веб-сторінці без необхідності постійного оновлення сторінки.
 6. Запуск веб-сервера:
 - a. Якщо скрипт запускається безпосередньо, запускається веб-сервер.

Код серверної частини наведений в Додатку Б. У цьому коді сервер отримує дані з мікроконтролера Arduino через серійний порт, перевіряє їх на валідність, парсить, зберігає у базу даних SQLite та передає на клієнтську частину в режимі реального часу за допомогою Flask-SocketIO.

Ця реалізація дозволяє створити ефективну та функціональну систему для моніторингу та управління мініпарником, яка може бути легко розширена та налаштована для різних потреб.

3.2 Реалізація веб-інтерфейсу для моніторингу даних

Для забезпечення ефективного та зручного моніторингу даних з мініпарника було розроблено сучасний веб-інтерфейс, який поєднує в собі базові технології веб-розробки та використовує сучасні бібліотеки для візуалізації даних.

Використані технології:

HTML/CSS/JavaScript

- HTML використовується для створення структури сторінки.
- CSS відповідає за стилізацію і зовнішній вигляд інтерфейсу.
- JavaScript використовується для динамічної взаємодії з користувачем та оновлення даних без перезавантаження сторінки.

Chart.js [9]

- Для візуалізації даних у вигляді графіків використовується Chart.js. Ця бібліотека дозволяє побудувати різноманітні типи графіків, такі як лінійні графіки для відображення змін температури і вологості з часом.

Під час розробки веб-інтерфейсу було звернуто увагу на кілька ключових аспектів:

- Маршрутизація і відображення даних
 - Для створення основної сторінки моніторингу (/) був використаний шаблон `index.html`, який включає в себе інтерактивний інтерфейс для відображення даних з мініпарника.
- Взаємодія з Arduino
 - На серверній стороні було встановлено серійне з'єднання з Arduino для зчитування даних про температуру і вологість.
 - Отримані дані перевіряються на валідність і, при успішній перевірці, зберігаються в базі даних SQLite для подальшого використання.
- Оновлення даних у реальному часі
 - За допомогою SocketIO дані, отримані з Arduino, передаються на клієнтську частину веб-інтерфейсу, що дозволяє миттєво оновлювати інформацію на сторінці без необхідності її перезавантаження.
- Візуалізація даних
 - Використання Chart.js дозволяє автоматично оновлювати графік з новими даними про температуру і вологість, що допомагає користувачам спостерігати за змінами у показниках з часом.

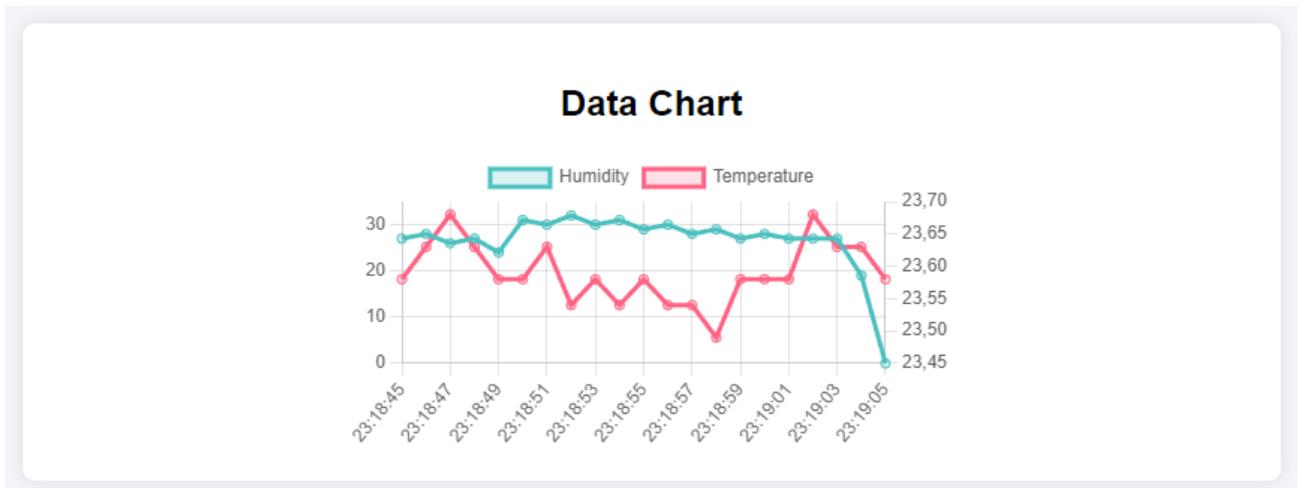


Рис. 3.2 Приклад графіка у режимі реального часу

- Інтерактивні елементи управління
 - В інтерфейсі додано кнопки для старту та зупинки збору даних з Arduino, що забезпечує користувачам контроль над процесом збору і відображення даних.

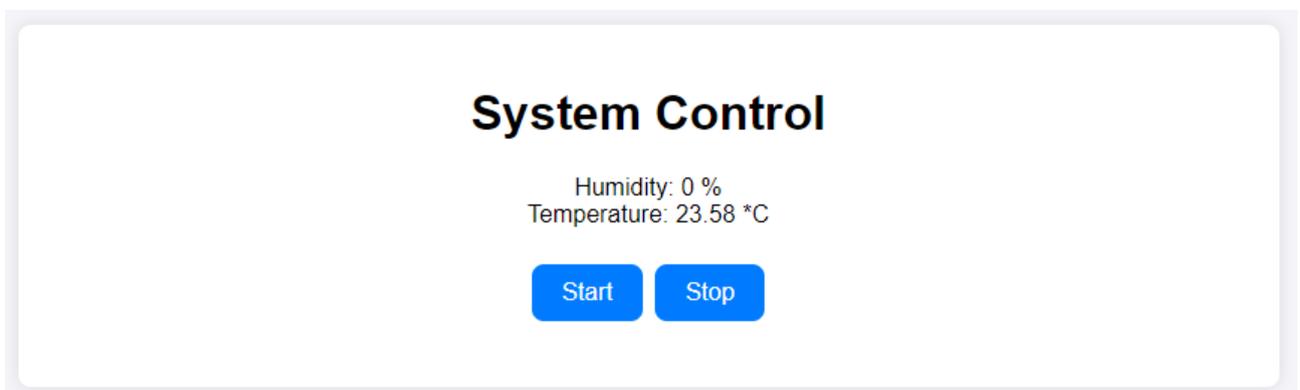


Рис. 3.3 Приклад контролю системи кнопками Start і Stop

Розроблений веб-інтерфейс для моніторингу даних з мініпарника є не лише функціональним, але й зручним для використання. Використання сучасних технологій HTML, CSS, JavaScript разом з бібліотекою Chart.js дозволило створити інтерактивний інтерфейс, який забезпечує швидку відповідь на дані в режимі реального часу. Цей веб-інтерфейс може бути легко розширений і адаптований для різних потреб користувачів, що робить його ідеальним інструментом для систем моніторингу та управління в різних областях застосування.

3.3 Функціональні можливості веб-додатку

Надалі розписуємо ключові функціональні можливості додатку, які включають в себе зручну навігацію за допомогою структурованого меню,

адаптивність для роботи на різних типах екранів та систему сповіщень, що повідомляє користувача про важливі стани системи.

Додаток "System Control" пропонує користувачеві зручну навігацію за допомогою чітко структурованого меню:

- **Home**
 - Головна сторінка, де відображається загальна інформація про стан системи моніторингу, зокрема поточні показники вологості та температури.

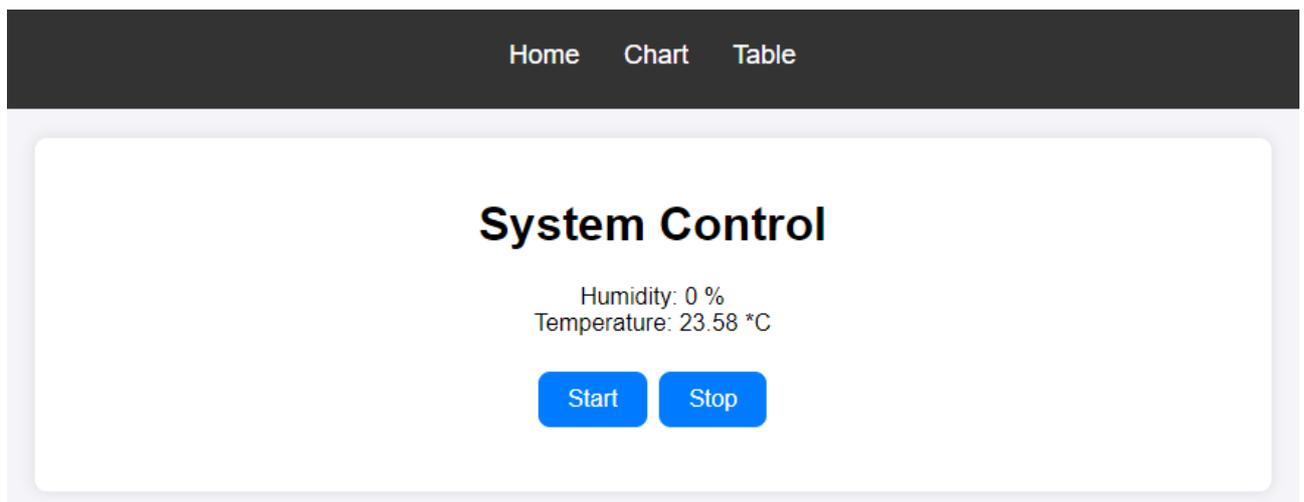


Рис. 3.4 Приклад меню додатка

```
<div id="chart-section">
  <h1>System Control</h1>
  <div id="humidity">Humidity: -- %</div>
  <div id="temperature">Temperature: -- *C</div>
  <div class="button-container">
    <button id="startButton" class="styled-button">Start</button>
    <button id="stopButton" class="styled-button">Stop</button>
  </div>
```

Chart

- Секція з графічним представленням даних, що дозволяє візуалізувати зміни вологості та температури в часі.

```
<div id="chart-section">
```

```

<h2>Data Chart</h2>

<canvas id="dataChart" width="400" height="200"></canvas>

</div>

```

Table

- Таблиця останніх даних, де користувач може переглядати деталізовані вимірювання вологості та температури.

The screenshot shows a web interface titled "Recent Data". At the top, there are three blue buttons: "Show Table", "Clear Table", and "Close Table". Below the buttons is a table with three columns: "Timestamp", "Humidity (%)", and "Temperature (°C)". The table contains three rows of data.

Timestamp	Humidity (%)	Temperature (°C)
23:18:42	30.50	23.92
23:18:52	28.00	23.62
23:19:02	28.60	23.57

Рис. 3.5 Приклад роботи таблиці зі збору даних

```

<div id="table-section">

  <h2>Recent Data</h2>

  <div class="button-container">

    <button id="showTableButton" class="styled-button">Show
    Table</button>

    <button id="clearTableButton" class="styled-button">Clear
    Table</button>

    <button id="closeTableButton" class="styled-button">Close
    Table</button>

  </div>

  <table id="dataTable">

    <thead>

      <tr>

        <th>Timestamp</th>

        <th>Humidity (%)</th>

```

```

        <th>Temperature (*C)</th>

    </tr>

</thead>

<tbody></tbody>

</table>

```

Додаток повністю адаптований для роботи на мобільних пристроях, що забезпечує зручне користування на будь-якому типі екрану.

- **Гнучка структура сторінки:** Елементи інтерфейсу масштабуються і перестраюються для оптимального використання на маленьких екранах.

```

@media screen and (max-width: 600px) {

    nav ul {

        flex-direction: column;

        align-items: center;

    }

    nav ul li {

        margin: 10px 0;

    }

    #data-container, #chart-section, #table-section {

        margin: 10px;

        padding: 10px;

    }

    .styled-button {

        padding: 8px 16px;

        font-size: 14px;

    }

}

```

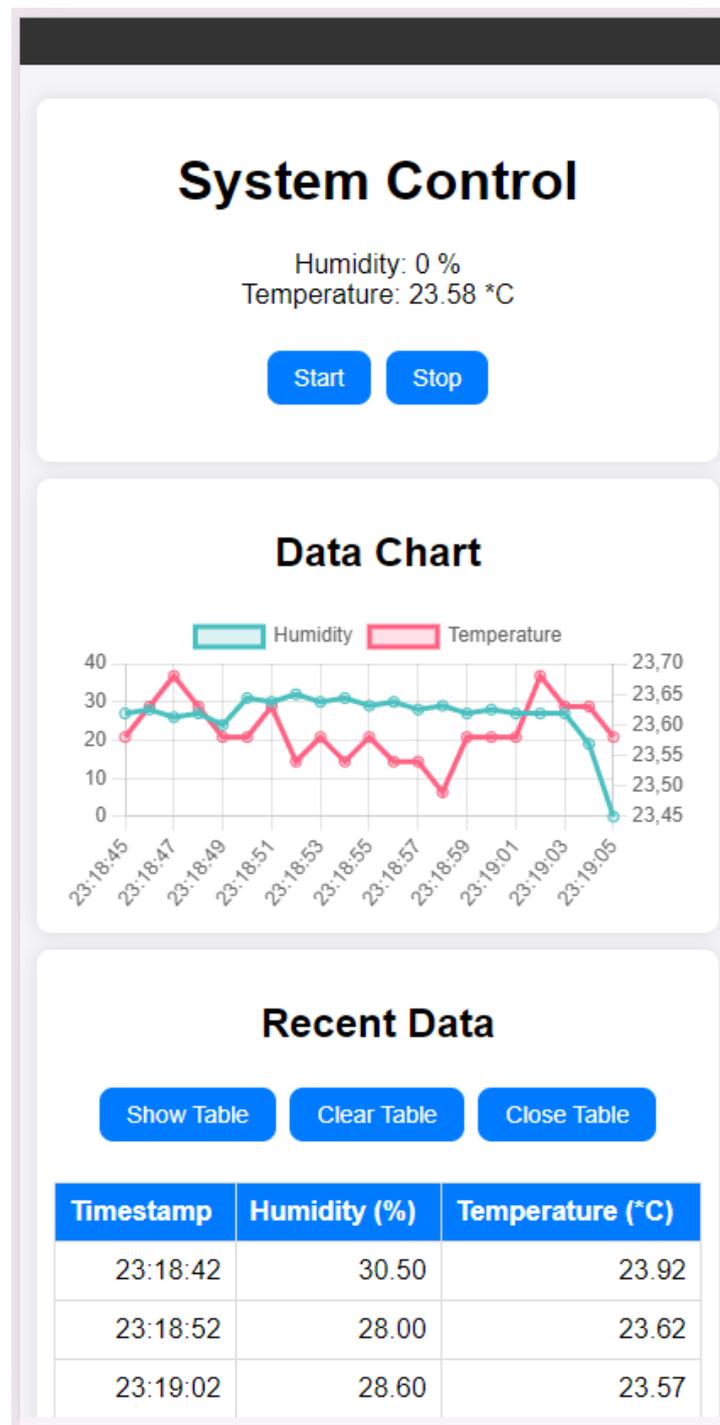


Рис. 3.6 Вигляд веб-додатку у телефонному вікні

Додаток має систему сповіщень, яка інформує користувача про важливі події та стани системи.

- **Відображення сповіщень:** Сповіщення демонструються у вигляді анімованих блоків, які автоматично зникають після відображення.

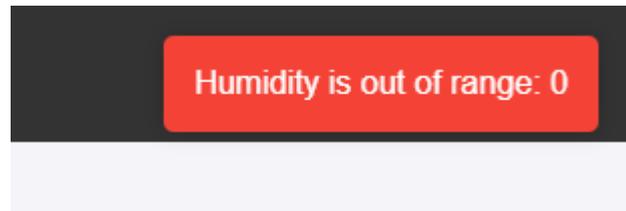


Рис. 3.7 Сповіщення про низьку вологість

```
// Функція для показу сповіщення  
  
function showNotification(type, value) {  
  
    var notification = document.createElement('div');  
    notification.className = 'notification';  
    notification.id = type.toLowerCase() + 'Notification';  
    notification.innerHTML = `${type} is out of range: ${value}`;  
    document.body.appendChild(notification);  
  
}
```

Ці функціональні можливості роблять додаток "System Control" хорошим інструментом для моніторингу та управління системою мініпарника, забезпечуючи зручність у використанні на будь-яких пристроях.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи були розроблені та імплементовані інноваційні технології для моніторингу та керування параметрами середовища в мініпарниках. Використання мікроконтролерів у поєднанні з веб-технологіями дозволило створити інтегровану систему, що забезпечує точний збір даних про температуру та вологість ґрунту, їхнє зберігання та подальше відображення у реальному часі через зручний веб-інтерфейс.

Розроблений веб-додаток не лише спрощує моніторинг та управління умовами в мініпарниках, а й значно підвищує їхню ефективність. Користувач може легко відстежувати та реагувати на зміни параметрів в реальному часі з будь-якого пристрою, що підключений до Інтернету, що робить цю систему дуже зручною та універсальною для використання у сільському господарстві.

Аналіз даних, зібраних системою, може служити основою для оптимізації процесів обігріву та поливу, що забезпечує економію ресурсів та покращення якості вирощування рослин. Такий підхід відкриває нові можливості для розвитку сільського господарства, зокрема у підтримці стабільного та якісного виробництва продуктів харчування, що є вкрай актуальним у сучасних умовах.

Отже, розробка і впровадження інтегрованого програмно-апаратного комплексу для мініпарників є кроком у напрямку оптимізації сільськогосподарських технологій та підвищення їхньої продуктивності, що має потенціал для широкого впровадження у сільському господарстві та домашньому використанні.

ДОДАТКИ

Додаток А

Код мікроконтролерної системи Arduino

```
#include <LiquidCrystal.h>

// Підключення бібліотеки LiquidCrystal

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Ініціалізація об'єкту LiquidCrystal з вказаними пінами для з'єднання з LCD-
екраном

float temperature;

int humidity;

int very_moist = 0;

int tmp = A1;

int grunt = A0;

// Оголошення змінних для температури, вологості та пінів

const int MAX_DATA_POINTS = 10;

// Задання константи для максимальної кількості збережених даних

float temperatureData[MAX_DATA_POINTS];

int humidityData[MAX_DATA_POINTS];

int dataCount = 0;

// Оголошення масивів для збереження даних та змінних для лічильника даних

const int MIN_HUMIDITY_THRESHOLD = 40;

const int MAX_HUMIDITY_THRESHOLD = 80;
```

```
// Задання порогових значень для вологості

const float MIN_TEMPERATURE_THRESHOLD = 10.0;

const float MAX_TEMPERATURE_THRESHOLD = 35.0;

// Задання порогових значень для температури

const int RELAY_PIN = 6;

// Ополошення піна для керування реле

const int SMOOTHING_WINDOW_SIZE = 10; // Розмір вікна згладжування

float temperatureSum = 0.0;

float temperatureBuffer[SMOOTHING_WINDOW_SIZE];

int temperatureIndex = 0;

bool bufferFilled = false;

void setup() {

    lcd.begin(16, 2);

    pinMode(tmp, INPUT);

    pinMode(RELAY_PIN, OUTPUT);

    // Ініціалізація LCD-екрану та пінів як входи та виходи

    Serial.begin(9600);

    // Ініціалізація послідовного порту з швидкістю 9600 бод

    // Ініціалізація буфера згладжування

    for (int i = 0; i < SMOOTHING_WINDOW_SIZE; i++) {
```

```

    temperatureBuffer[i] = 0.0;

}

}

void loop() {

    lcd.clear();

    // Очищення екрану LCD

    // Зчитування температури

    float rawTemperature = analogRead(tmp) * (5.0 / 1024.0); // Правильний
коєфіцієнт для перетворення аналогового значення у напругу

    rawTemperature = (rawTemperature - 0.5) * 100.0;           // Конвертація
напруги у температуру

    // Оновлення буфера згладжування

    temperatureSum -= temperatureBuffer[temperatureIndex];

    temperatureBuffer[temperatureIndex] = rawTemperature;

    temperatureSum += rawTemperature;

    temperatureIndex++;

    if (temperatureIndex >= SMOOTHING_WINDOW_SIZE) {

        temperatureIndex = 0;

        bufferFilled = true;

    }

    // Обчислення згладженого значення температури

    if (bufferFilled) {

        temperature = temperatureSum / SMOOTHING_WINDOW_SIZE;

    } else {

        temperature = temperatureSum / temperatureIndex;

```

```
}  
  
// Зчитування вологості  
humidity = analogRead(grunt);  
humidity = map(humidity, very_moist, 876, 0, 100);  
  
// Збереження вимірених значень в масиви  
temperatureData[dataCount] = temperature;  
humidityData[dataCount] = humidity;  
  
dataCount++;  
  
if (dataCount >= MAX_DATA_POINTS) {  
    dataCount = 0;  
}  
  
// Виведення значення вологості на перший рядок LCD  
lcd.setCursor(0, 0);  
lcd.print("H: ");  
lcd.print(humidity);  
lcd.print(" %");  
  
// Виведення значення температури на другий рядок LCD  
lcd.setCursor(0, 1);  
lcd.print("T:");  
lcd.print(temperature);  
lcd.print(" *C");  
  
// Виведення значень вологості та температури на послідовний порт  
Serial.print("Humidity: ");  
Serial.print(humidity);  
Serial.print(" %\t");
```

```

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(" *C");

// Затримка 1 секунда

delay(1000);

}

```

Додаток Б

Фрагмент коду серверної частини

```

from flask import Flask, render_template
from flask_socketio import SocketIO
import serial
import time
import re
import threading
import sqlite3
from datetime import datetime

app = Flask(__name__)
socketio = SocketIO(app)

@app.route('/')
def index():
    return render_template('index.html')

arduino_port = 'COM3'
baud_rate = 9600
ser = serial.Serial(arduino_port, baud_rate)
time.sleep(2)
def is_valid_data(data):

```

```

humidity = re.search(r'Humidity:\s*(\d+)\s*%', data).group(1)

    temperature = re.search(r'Temperature:\s*(-?\d+\.\d+)\s*\s*C',
data).group(1)

    return humidity, temperature

data_collection_active = False

def save_to_db(humidity, temperature):
    conn = sqlite3.connect('arduino_data.db')
    c = conn.cursor()
    c.execute('''
        INSERT INTO sensor_data (humidity, temperature) VALUES (?, ?)
    ''', (humidity, temperature))
    conn.commit()
    conn.close()

def collect_data():
    while data_collection_active:
        time.sleep(0.1)
        if ser.in_waiting > 0:
            data = ser.readline().decode('utf-8').rstrip()
            if is_valid_data(data):
                humidity, temperature = parse_data(data)
                save_to_db(humidity, temperature)
                timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
                socketio.emit('data_update', {'timestamp': timestamp,
'humidity': humidity, 'temperature': temperature})
            else:
                print('Invalid data:', data)
                continue

```

```

@socketio.on('connect')

def test_disconnect():
    print('Client connected')

@socketio.on('disconnect')

def test_disconnect():
    print('Client disconnected')

@socketio.on('start_data')

def start_data():
    global data_collection_active
    data_collection_active = True
    threading.Thread(target=collect_data).start()

@socketio.on('stop_data')

def stop_data():
    global data_collection_active
    data_collection_active = False

if __name__ == '__main__':
    socketio.run(app)

```

Додаток В

Функція оновлення графіку

```

// Функція для оновлення графіку

var ctx = document.getElementById('dataChart').getContext('2d');
var dataChart = new Chart(ctx, {
    type: 'line',
    data: {
        labels: [],
        datasets: [

```

```
{
  label: 'Humidity',
  borderColor: 'rgba(75, 192, 192, 1)',
  backgroundColor: 'rgba(75, 192, 192, 0.2)',
  data: [],
  yAxisID: 'y'
},
{
  label: 'Temperature',
  borderColor: 'rgba(255, 99, 132, 1)',
  backgroundColor: 'rgba(255, 99, 132, 0.2)',
  data: [],
  yAxisID: 'y1'
}
]
},
options: {
  scales: {
    y: {
      type: 'linear',
      position: 'left',
    },
    y1: {
      type: 'linear',
      position: 'right',
      grid: {
        drawOnChartArea: false,
      },
    },
  },
},
},
```

```

    }
  });

function updateChart(humidity, temperature) {
  const time = new Date().toLocaleTimeString();
  if (dataChart.data.labels.length > 20) {
    dataChart.data.labels.shift();
    dataChart.data.datasets[0].data.shift();
    dataChart.data.datasets[1].data.shift();
  }
  dataChart.data.labels.push(time);
  dataChart.data.datasets[0].data.push(humidity);
  dataChart.data.datasets[1].data.push(temperature);
  dataChart.update();
}

```

Додаток Г

Функція оновлення сповіщень

```

// Функція для оновлення сповіщень
function updateNotifications(humidity, temperature) {
  // Сповіщення для вологості
  if (humidity < humidityThresholdLow || humidity >
humidityThresholdHigh) {
    if (!humidityNotificationVisible) {
      showNotification('Humidity', humidity);
      humidityNotificationVisible = true;
    } else {
      updateNotification('humidityNotification', 'Humidity',
humidity);
    }
  }
}

```

```

    } else {
        if (humidityNotificationVisible) {
            removeNotification('humidityNotification');
            humidityNotificationVisible = false;
        }
    }

    // Сповіщення для температури
    if (temperature < temperatureThresholdLow || temperature >
temperatureThresholdHigh) {
        if (!temperatureNotificationVisible) {
            showNotification('Temperature', temperature);
            temperatureNotificationVisible = true;
        } else {
            updateNotification('temperatureNotification', 'Temperature',
temperature);
        }
    } else {
        if (temperatureNotificationVisible) {
            removeNotification('temperatureNotification');
            temperatureNotificationVisible = false;
        }
    }
}}

```

Додаток Д

Функція оновлення таблиці

```

// Функція для оновлення таблиці
function updateTable(timestamp, humidity, temperature) {
    const table =
document.getElementById('dataTable').getElementsByTagName('tbody')[0];

```

```
const newRow = table.insertRow();  
const timestampCell = newRow.insertCell(0);  
const humidityCell = newRow.insertCell(1);  
const temperatureCell = newRow.insertCell(2);  
timestampCell.innerHTML = timestamp;  
humidityCell.innerHTML = humidity;  
temperatureCell.innerHTML = temperature;  
}
```


СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Використання технологій IoT в агробізнесі. URL: <https://agronomy.com.ua/statti/1625-vykorystannia-tekhnologii-internetu-rechei-v-ahrobiznesi.html> (дата звернення 04.06.2024)
2. Arduino UNO R3. URL: <https://doc.arduino.ua/ru/hardware/Uno> (дата звернення 21.05.2024)
3. Символьний екран LCD. URL: <https://uamper.com/LCD-екран-синій> (дата звернення 21.05.2024)
4. Аналоговий датчик температури та вологості TMP36. URL: <https://uamper.com/TMP36> (дата звернення 21.05.2024)
5. Датчик вологості ґрунту. URL: <https://uamper.com/Датчик-вологості-ґрунту> (дата звернення 21.05.2024)
6. Tinkercad. URL: <https://www.tinkercad.com/> (дата звернення 25.05.2024)
7. Фреймвок Flask. URL: <https://uk.wikipedia.org/wiki/Flask> (дата звернення 27.05.2024)
8. Документація Flask-SocketIO. URL: <https://flask-socketio.readthedocs.io/en/latest/> (дата звернення 27.05.2024)
9. Документація Charts.js. URL: <https://www.chartjs.org/docs/latest/> (дата звернення 28.05.2024)