

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувача кафедри,
д-р техн. наук, проф.

_____І. М. Журавська

« ____ » _____2024 р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

Апаратно-програмний комплекс моніторингу якості води на основі модуля ESP 32

Спеціальність «Комп'ютерна інженерія»
123 – КМР.1 – 605. 21810503

Студент

_____Д. В. Булавський

підпис

« ____ » _____2024 р.

Керівник канд. техн. наук, доцент

_____А. П. Бойко

підпис

« ____ » _____2024 р.

Миколаїв 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

«_____» _____ 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної магістерської роботи

Видано студенту групи 605м факультету комп'ютерних наук

_____ Булавському Дмитру Віталійовичу
(*прізвище, ім'я, по батькові студента*)

1. Тема кваліфікаційної роботи

Апаратно-програмний комплекс моніторингу якості води на основі модуля ESP 32

Затверджена наказом по ЧНУ ім. Петра Могили від 08.11.2023 № 226.

2. Строк представлення кваліфікаційної роботи «_____» _____ 2024__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є: апаратно-програмний комплекс моніторингу якості води на основі модуля ESP32, який дозволить вимірювати та відображати основні показники якості води, такі як температура та провідність, а також відстежувати отримані дані на сервері та мати можливість їх аналізувати.

4. Перелік питань, що підлягають розробці

1) огляд сучасних підходів та систем моніторингу якості води;

2) аналіз переваг та недоліків існуючих систем;

3) розробка апаратної частини комплексу моніторингу якості води на основі ESP32;

4) розробка програмної частини та тестування комплексу моніторингу якості води на основі ESP32;

5. Перелік графічних матеріалів

Слайди презентації, таблиці, рисунки

6. Завдання до спеціальної частини

Проведення оцінки умов праці фахівців підприємства, а саме аналіз виробничого приміщення, оцінка природного освітлення. Визначення рівня електричної та пожежної безпеки підприємства

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Д-р біол. наук, професорка Григор'єва Л. І.	Кафедра екології Навчально-наукового медичного інституту ЧНУ ім. Петра Могили	Спеціальна частина з охорони праці та безпеки у надзвичайних ситуаціях

Керівник роботи

д-р техн. наук, доцент Бойко Анжела Петрівна
(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Булавський Дмитро Віталійович
(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання « ____ » _____ 2023 ____ р.

КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної магістерської роботи

Тема: Апаратно-програмний комплекс моніторингу якості води на основі модуля ESP 32

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КМР	03.10.2023	10.10.2023	
2.	Огляд літератури за темою роботи	12.10.2023	19.10.2023	
	Складання календарного плану КМР	20.10.2023	21.10.2023	
4.	Аналіз предметної області	22.10.2023	29.10.2023	
5.	Розробка проєктних рішень	30.10.2023	04.11.2023	
6.	Моделювання та конструювання АПК	05.11.2023	19.11.2023	
7.	Перевірка працездатності, тестування та апробація розробленого АПК, аналіз результатів тестування	20.11.2023	27.11.2023	
8.	Відгук керівника КМР	29.11.2023	30.11.2023	
9.	Оформлення КМР та презентації	01.12.2023	08.11.2023	
10.	Попередній захист	31.01.2023	31.01.2023	
11.	Рецензування	05.02.2024	08.02.2024	
12.	Завершення оформлення КМР та презентації	08.02.2024	12.02.2024	
13.	Захист кваліфікаційної роботи			

Розробив студент Булавський Дмитро Віталійович
(прізвище, ім'я, по батькові)

_____ (підпис)

«___» _____ 2024_р

Керівник роботи д-р техн. наук, доцент, Бойко Анжела Петрівна
(посада, прізвище, ім'я, по батькові)

_____ (підпис)

«___» _____ 2024_____ р

АНОТАЦІЯ

до кваліфікаційної магістерської роботи
« Апаратно-програмний комплекс моніторингу якості води на основі модуля
ESP 32»

Студент 605м гр.: Булавський Дмитро Віталійович
Керівник: канд. техн. наук, доцент Бойко Анжела Петрівна

Магістерська робота присвячена розробці апаратно-програмного комплексу моніторингу якості води на основі модуля ESP32. Актуальність теми магістерської роботи зумовлена необхідністю контролю за станом водних ресурсів, які впливають на здоров'я та добробут людей. Розглянуто існуючі методи, засоби та аналоги моніторингу якості води. Практичне значення результатів розроблення апаратно-програмного комплексу полягає в тому, що він дозволяє вимірювати та передавати дані про температуру та електропровідність води, які є важливими показниками її якості.

У вступі визначається актуальність теми, сформульовані мета, об'єкт, предмет та завдання дослідження та розроблення магістерської роботи. У першому розділі проводиться аналіз існуючих методів, засобів та аналогів моніторингу якості води. У другому розділі розробляється апаратна частина комплексу, яка включає вибір та обґрунтування компонентів, розробку схеми електричного зв'язку, налагодження та тестування. У третьому розділі розробляється програмна частина комплексу, яка включає вибір та обґрунтування мови програмування, розробку алгоритму роботи, завантаження, налагодження та тестування. У висновках наведено аналіз виконаної роботи та отриманих результатів дослідження та розроблення.

В цілому, магістерська робота без додатків містить 61 сторінки, 13 рисунків, 14 джерела посилання.

Ключові слова: моніторинг якості води, IoT, ESP32, датчик температури, датчик електропровідності, OLED дисплей, ThingSpeak.

ABSTRACT

to the qualification master's thesis

"Hardware and software complex for water quality monitoring based on ESP 32 module"

Student 605m gr.: Bulavskyi Dmytro Vitaliiovych

Supervisor: Cand. of Tech. Sc., Docent Boyko Angela Petrivna

The master's thesis is devoted to the development of a hardware and software system for water quality monitoring based on the ESP32 module. The relevance of the master's thesis topic is due to the need to monitor the state of water resources that affect human health and well-being. The existing methods, means and analogs of water quality monitoring are considered. The practical significance of the results of the hardware and software complex development is that it allows measuring and transmitting data on the temperature and electrical conductivity of water, which are important indicators of its quality.

The introduction defines the relevance of the topic, formulates the purpose, object, subject, and objectives of the research and development of the master's thesis. The first chapter analyzes the existing methods, means and analogs of water quality monitoring. The second section develops the hardware part of the complex, which includes the selection and justification of components, development of an electrical communication scheme, debugging and testing. In the third section, the software part of the complex is developed, which includes the selection and justification of the programming language, development of the algorithm, loading, debugging and testing. The conclusions provide an analysis of the work performed and the results of research and development.

In general, the master's thesis without appendices contains 61 pages, 13 figures, 14 references.

Keywords: water quality monitoring, IoT, ESP32, temperature sensor, conductivity sensor, OLED display, ThingSpeak.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП	10
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ, ЗАСОБІВ ТА АНАЛОГІВ МОНІТОРИНГУ ЯКОСТІ ВОДИ.....	12
1.1 Аналіз методів та засобів моніторингу якості води	12
1.2 Порівняльний аналіз існуючих технологій	13
1.3 Особливості аналізу на основі IoT	14
1.4 Параметри аналізу води.....	16
1.5 Існуючі аналоги моніторингу якості води.....	18
1.6 ESP32 особливості та характеристики	22
1.7 Формування специфікації вимог до АПЗ	24
Висновок до розділу 1	27
2 ПРОЕКТУВАННЯ АПАРАТНОЇ ЧАСТИНИ МОНІТОРИНГУ ЯКОСТІ ВОДИ.....	28
2.1 Вибір та обґрунтування компонентів АПК	29
2.2 Розробка схеми електричного зв'язку компонентів АПК.....	33
2.3 Розробка друкованої плати для АПК	36
2.4 Монтаж та налагодження АПК.....	38
2.5 Тестування та вимірювання параметрів АПК.....	40
Висновок до розділу 2	41
3 ПРОЕКТУВАННЯ ПРОГРАМНОЇ ЧАСТИНИ АПК МОНІТОРИНГУ ЯКОСТІ ВОДИ	43
3.1 Вибір та обґрунтування мови програмування та середовища розробки для АПК.....	43
3.2 Розробка алгоритму роботи АПК	44
3.3 Використані бібліотеки та їх функції.....	47
3.4 ThingSpeak: Робота та особливості.....	50
3.5 Калібрування датчиків.....	53
3.5.1 Калібрування датчика температури DS18B20	53

3.5.2	Калібрування аналогового датчика TDS	55
3.6	Завантаження та налагодження програмного коду на АПК	57
3.7	Тестування та відлагодження програмного коду АПК	58
	Висновок до розділу 3	63
	ВИСНОВКИ.....	64
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	65
	ДОДАТОК А.....	67
	ДОДАТОК Б	68
	ДОДАТОК В	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ГДК	–	Гранично допустима концентрація
АПК	–	Апаратно програмний комплекс
IDE	–	Integrated Development Environment
IoT	–	Internet of Things
USB	–	Universal Serial Bus
WQMK	–	Water Quality Monitoring Kit
WQMS	–	Water Quality Monitoring System
API	–	Application Programming Interface

ВСТУП

Вода – один з найважливіших природних ресурсів для життя людини та збереження навколишнього середовища. Якість води впливає на здоров'я населення, стан екосистем, сільське господарство, промисловість та енергетику. Однак вода є також об'єктом забруднення, знецінення та конфліктів, особливо в умовах війни та кризи. Україна, яка переживає з 2014 року російську агресію та військовий конфлікт на сході країни, зіткнулася з серйозними викликами щодо забезпечення належної якості води для свого населення та економіки. Зокрема, місто Миколаїв, яке розташоване на півдні України біля Дніпровського лиману, зазнало значних наслідків від війни, таких як пошкодження водопровідних мереж, зростання ризику хімічного та бактеріального забруднення води, недостатність фінансових та технічних ресурсів для моніторингу та контролю якості води.

Тому актуальною є задача розробки та впровадження системи моніторингу якості води, яка б використовувала сучасні технології та можливості Інтернету речей. Така система дозволила б отримувати оперативну та достовірну інформацію про стан водних ресурсів, виявляти та запобігати їх забрудненню, а також забезпечувати ефективне управління водним господарством.

Мета роботи: розробити апаратно-програмний комплекс моніторингу якості води на основі модуля ESP32, який дозволить вимірювати та відображати основні показники якості води, такі як температура та провідність, а також відстежувати отримані дані на сервері та мати можливість їх аналізувати.

Об'єкт дослідження: процеси вимірювання, передачі, обробки та відображення даних про якість води.

Предмет дослідження: методи, засоби та технології вимірювання, передачі, обробки та відображення даних про якість води на основі модуля ESP32 та Інтернету речей.

Завдання роботи:

- проаналізувати аналогові пристрої які вже існують, порівняти їх з власним;
- вивчити особливості модуля ESP32 та його можливості для реалізації системи моніторингу якості води;
- розробити апаратну частину комплексу, яка складається з модуля ESP32 та датчиків якості води;
- розробити програмну частину комплексу, яка складається з прошивки для модуля ESP32, хмарного сервісу для зберігання та обробки даних, мобільного додатку для відображення та керування комплексом;
- провести експериментальні випробування комплексу та оцінити його функціональність, точність, надійність та енергоефективність.

Практичне значення роботи: розроблений апаратно-програмний комплекс моніторингу якості води на основі модуля ESP32 може бути використаний для вирішення актуальної проблеми контролю якості води в умовах війни та кризи, а також для підвищення екологічної свідомості та безпеки населення. Комплекс може бути також адаптований для інших цілей та застосувань, пов'язаних з моніторингом якості води, таких як сільське господарство, аквакультура, промисловість тощо.

Робота пройшла **апробацію** під час XXVI Всеукраїнської науково-практичної конференції «Могилянські читання» (Миколаїв, 07–11 листопада 2023 р.).

Публікації. Основні положення та результати магістерської роботи опубліковані у збірнику матеріалів XXVI Всеукраїнської науково-практичної конференції «Могилянські читання–2023».

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ, ЗАСОБІВ ТА АНАЛОГІВ МОНІТОРИНГУ ЯКОСТІ ВОДИ

Моніторинг якості води – це процес вимірювання, спостереження, оцінки та документування фізичних, хімічних та біологічних характеристик води у водних екосистемах. Метою моніторингу якості води є забезпечення безпечної та здорової води для людей, тварин і рослин, а також виявлення та запобігання забрудненню води, охороні водних ресурсів та підтримці екологічної рівноваги.

Провідність води — це здатність води проводити електричний струм. Провідність води залежить від кількості розчинених солей, мінералів, органічних речовин і мікроорганізмів у воді. Чим більше таких речовин у воді, тим вища її провідність.

Провідність води впливає на якість води, оскільки вона відображає ступінь забруднення води. Висока провідність води може свідчити про наявність шкідливих речовин, таких як нітрати, фосфати, пестициди, метали тощо. Така вода може бути небезпечною для здоров'я людей і тварин, а також для екосистеми. Низька провідність води може означати, що вода має низьку мінералізацію і біологічну активність. Така вода може бути недостатньою для деяких видів водокористування, наприклад, для зрошення рослин.

1.1 Аналіз методів та засобів моніторингу якості води

Існують різні методи та засоби моніторингу якості води, які можна класифікувати за такими критеріями:

За типом води: поверхнева вода (річки, озера, моря, океани тощо) або підземна вода (свердловини, джерела, артезіанські води тощо);

За місцем проведення: лабораторний аналіз (збір проб води та їх вивчення в лабораторії за допомогою спеціального обладнання та реактивів) або польовий аналіз (вимірювання параметрів води безпосередньо на місці за допомогою портативних датчиків, зондів, тест-смужок тощо);

За частотою проведення: постійний моніторинг (неперервне або періодичне вимірювання параметрів води за допомогою стаціонарних або мобільних систем моніторингу, які передають дані бездротовим шляхом до центральної бази даних) або епізодичний моніторинг (одноразове або рідкісне вимірювання параметрів води за допомогою ручних або переносних приладів, які потребують ручного зчитування та запису даних);

За метою проведення: екологічний моніторинг (визначення стану водних екосистем, їх відповідності екологічним нормам та стандартам, а також виявлення впливу антропогенних факторів на водне середовище) або господарський моніторинг (визначення придатності води для різних видів водокористування, таких як питне водопостачання, санітарія, сільське господарство, промисловість, рибальство, туризм тощо).

Кожен метод та засіб моніторингу якості води має свої переваги та недоліки, які залежать від рівня точності, швидкості, надійності, енергоефективності, вартості, зручності та інших факторів. Отже, вибір методу та засобу моніторингу якості води залежить від конкретних цілей, умов та ресурсів кожного випадку.

1.2 Порівняльний аналіз існуючих технологій

ІоТ-рішення: Це технології, які використовують інтернет речей (ІоТ) для збору, передачі, обробки та візуалізації даних про якість води у реальному часі. Наприклад, можна використовувати модуль ESP32, який є компактним, енергоефективним, надійним та легким у використанні пристроєм, здатним вимірювати різні параметри якості води, передавати дані бездротовим шляхом та візуалізувати їх на веб-інтерфейсі. Переваги ІоТ-рішень полягають у високій точності, швидкості, надійності, енергоефективності, низькій вартості та зручності. Обмеження ІоТ-рішень полягають у складності інтеграції, потребі в стабільному зв'язку та живленні, ризику втрати або пошкодження даних, а також у відсутності стандартизації та регулювання.

Лабораторний аналіз: Це традиційний метод моніторингу якості води, який полягає у зборі проб води та їх вивченні в лабораторії за допомогою спеціального обладнання та реактивів. Наприклад, можна використовувати спектрофотометр, титратор, хроматограф, мікроскоп та інші прилади для визначення різних хімічних, фізичних та біологічних характеристик води. Переваги лабораторного аналізу полягають у високій точності, повноті, достовірності та об'єктивності даних. Обмеження лабораторного аналізу полягають у високій вартості, тривалості, складності та незручності, а також у можливих помилках або змінах властивостей води під час транспортування або зберігання проб.

Аналітичні та сенсорні технології: Це сучасні технології, які використовують різні аналітичні та сенсорні методи для вимірювання параметрів якості води. Наприклад, можна використовувати електрохімічні, оптичні, термічні, механічні, акустичні, магнітні, радіоактивні, біологічні та інші сенсори для визначення різних показників якості води. Переваги аналітичних та сенсорних технологій полягають у високій чутливості, селективності, швидкості та простоті вимірювань, а також у можливості використання різних матеріалів, форм та розмірів сенсорів. Обмеження аналітичних та сенсорних технологій полягають у високій вартості, потребі в калібруванні, стабілізації та захисті сенсорів, а також у впливі різних факторів, таких як температура, тиск, рН(кислотність), соліність, органічні речовини тощо.

1.3 Особливості аналізу на основі IoT

Останнім часом все більшу популярність набуває моніторинг якості води на основі Інтернету речей (IoT), який використовує сучасні технології та можливості для збору, передачі, обробки та візуалізації даних про якість води у реальному часі. Такий моніторинг якості води має ряд переваг, таких як:

Висока точність: Датчики та зонди, які використовуються для вимірювання параметрів якості води, мають високу точність та стабільність, а також можуть бути калібровані та налаштовані відповідно до потреб;

Висока швидкість: Дані про якість води можуть бути передані бездротовим шляхом до веб-сервера за допомогою Wi-Fi або LoRa, а також можуть бути доступні для перегляду та аналізу на веб-інтерфейсі у реальному часі;

Висока надійність: Система моніторингу якості води IoT може працювати автономно та безперебійно, а також мати функції самодіагностики та відновлення після збоїв;

Низьке енергоспоживання: Система моніторингу якості води IoT може використовувати енергозберігаючі технології та джерела живлення, такі як сонячні панелі, акумулятори, суперконденсатори тощо;

Низька вартість: Система моніторингу якості води IoT може використовувати недорогі та доступні компоненти, такі як модуль ESP32, який є компактним, енергоефективним, надійним та легким у використанні пристроєм, здатним вимірювати різні параметри якості води, передавати дані бездротовим шляхом та візуалізувати їх на веб-інтерфейсі;

Висока зручність: Система моніторингу якості води IoT може бути легко встановлена, налаштована та керована за допомогою веб-інтерфейсу, який надає інтуїтивний та інтерактивний спосіб перегляду та аналізу даних про якість води на карті, графіках, таблицях тощо.

Однак, система моніторингу якості води IoT також має деякі обмеження та виклики, такі як:

Складність інтеграції: Система моніторингу якості води IoT вимагає сумісності та інтеграції різних компонентів, таких як датчики, шлюзи, комунікаційні протоколи, платформи IoT, веб-сервери, веб-інтерфейси тощо, що може бути складним та трудомістким процесом;

Потенційні ризики безпеки: Система моніторингу якості води IoT може бути вразливою до кібератак, які можуть перехопити, змінити або

видалити дані про якість води, а також викликати несправність або пошкодження системи, що може призвести до неправильної оцінки та реагування на стан води;

Недостатність стандартів та регулювання: Система моніторингу якості води IoT може зіткнутися з відсутністю або невідповідністю стандартів та регулювання, які визначають вимоги та критерії для вимірювання, передачі, обробки та візуалізації даних про якість води, а також захисту прав та інтересів зацікавлених сторін.

Таким чином, система моніторингу якості води IoT є перспективним та інноваційним засобом для покращення та оптимізації моніторингу якості води, але вона також потребує подальшого дослідження та розвитку, щоб подолати існуючі проблеми та виклики.

1.4 Параметри аналізу води

Якість води характеризується різними параметрами, які відображають її фізичні, хімічні та біологічні властивості. Для оцінки якості води необхідно проводити аналіз води, який дозволяє визначити рівень забруднення води та її придатність для різних цілей використання. Одними із головних параметрів для аналізу води є:

а) рН - це показник кислотності або лужності води, який вимірюється за допомогою рН-метра або індикаторних смужок. рН води впливає на розчинність та доступність різних хімічних речовин, які визначають її біологічну активність та корозійні властивості.

б) Розчинений кисень - це кількість кисню, який розчинений у воді, який вимірюється за допомогою оксиметра. Розчинений кисень важливий для дихання водних організмів та окислення органічних речовин.

в) Електропровідність - це здатність води проводити електричний струм, яка залежить від концентрації розчинених солей, мінералів та інших іонів. Електропровідність вимірюється за допомогою кондуктометра.

Електропровідність вказує на солоність води та її придатність для сільського господарства та питних потреб.

г) Температура - це міра теплового стану води, яка вимірюється за допомогою термометра. Температура води впливає на її густину, в'язкість, розчинність газів, швидкість хімічних реакцій, метаболізм водних організмів та інші процеси.

д) Турбідність - це міра затемнення води через наявність у ній дрібних частинок, таких як глина, пісок, планктон, мікроорганізми, органічні речовини та інші. Турбідність вимірюється за допомогою турбідиметра. Турбідність вказує на якість води, її прозорість, продуктивність фотосинтезу, потенційну наявність забруднювачів та патогенів.

е) Хлор - це хімічний елемент, який може бути присутнім у воді у вигляді розчиненого газу, розчиненої кислоти або розчинених солей. Хлор вимірюється за допомогою хлорометра або фотоколориметра. Хлор використовується для дезінфекції води від бактерій та вірусів, але також може бути токсичним для водних організмів та людей.

ж) Аміак - це хімічна сполука, яка складається з азоту та водню, яка може бути присутньою у воді у вигляді розчиненого газу, розчиненої основи або розчинених солей. Аміак вимірюється за допомогою аміакметра або фотоколориметра. Аміак є джерелом азоту для росту рослин та мікроорганізмів, але також може бути токсичним для водних організмів та людей.

з) Нітрати - це хімічні сполуки, які складаються з азоту та кисню, які можуть бути присутніми у воді у вигляді розчинених солей. Нітрати вимірюються за допомогою нітратометра або фотоколориметра. Нітрати є джерелом азоту для росту рослин та мікроорганізмів, але також можуть бути токсичними для водних організмів та людей, особливо для немовлят.

і) Фосфати - це хімічні сполуки, які складаються з фосфору та кисню, які можуть бути присутніми у воді у вигляді розчинених солей. Фосфати

вимірюються за допомогою фосфатометра або фотоколориметра. Фосфати є джерелом фосфору для росту рослин та мікроорганізмів.

1.5 Існуючі аналоги моніторингу якості води

Існують різні аналоги та технології моніторингу якості води, які можна розділити на дві групи: стаціонарні та мобільні. Стаціонарні системи моніторингу якості води встановлюються на постійній основі у визначених точках водоймища або водопроводу, де вони безперервно вимірюють та передають дані про воду. Мобільні системи моніторингу якості води є переносними та можуть бути використані для вимірювання води у різних місцях та умовах. Деякі приклади аналогів та технологій моніторингу якості води наведені нижче:

1) Water Quality Monitoring System (WQMS) – це система, яка вимірює різні показники якості води, такі як рН(кислотність), алкаліність, нітрати та амоній. Аналізуючи ці рівні, компанія може контролювати ефективність процесу фільтрації. Системи моніторингу якості води також підтверджують або заперечують, чи використовуються правильні хімікати для обробки.

Особливості: WQMS може вимірювати різні параметри якості води, такі як хімічний попит на кисень (COD), загальні фосфати (TP), загальний азот (TN) тощо. Потрібні параметри налаштовуються відповідно до типу заводу та норм викидів. WQMS також може використовувати технологію IoT для передачі даних про якість води в режимі реального часу та отримання сповіщень про будь-які порушення.

Плюси: WQMS може допомогти підвищити ефективність та продуктивність процесів очищення води, зменшити витрати на енергію та хімікати, запобігти забрудненню водойм та покращити дотримання екологічних стандартів. WQMS також може підвищити безпеку та здоров'я працівників, оскільки вони не потребують ручного збору та аналізу зразків води.

Мінуси: WQMS може мати високу вартість установки та обслуговування, особливо якщо потрібно встановити багато датчиків та передавачів. WQMS також може бути вразливим до пошкоджень або збоїв через надмірну вологість, корозію, біологічний ріст або інтерференцію сигналу. WQMS також може потребувати регулярної калібрування та перевірки точності датчиків.



Рисунок 1.1 – Water Quality Monitoring System (WQMS)

2) Water Quality Monitoring Kit (WQMK) – це мобільна система моніторингу якості води, яка складається з портативного вимірювального приладу та набору сенсорів для вимірювання води.

Переваги:

- WQMK дозволяє вимірювати якість води в реальному часі, аналізувати тренди якості води та виявляти аномальні події, такі як навмисне забруднення води;
- WQMK може допомогти підвищити ефективність та продуктивність процесів очищення води, зменшити витрати на енергію та хімікати, запобігти забрудненню водойм та покращити дотримання екологічних стандартів;
- WQMK може підвищити безпеку та здоров'я працівників, оскільки вони не потребують ручного збору та аналізу зразків води.

Недоліки:

- WQMK може мати високу вартість установки та обслуговування, особливо якщо потрібно встановити багато датчиків та передавачів;

- WQMK може бути вразливим до пошкоджень або збоїв через надмірну вологість, корозію, біологічний ріст або інтерференцію сигналу;
- WQMK може потребувати регулярної калібрування та перевірки точності датчиків;
- система може мати обмежену точність та надійність вимірювань;
- система може мати обмежену тривалість роботи від батареї.



Рисунок 1.2 – Water Quality Monitoring Kit (WQMK)

3) Water Quality Monitoring Drone (WQMD) – є пристроєм моніторингу якості води, який використовує безпілотний літальний апарат (дрон) для збору, передачі та аналізу даних про якість води у реальному часі.

Характеристики: WQMD складається з дрона, який може підніматися та опускатися над водою, датчика якості води, який може вимірювати різні параметри, такі як рН, температура, розчинений кисень, турбідність, амоній, нітрати, хлор тощо, та бездротового модуля, який може передавати дані до бази даних або хмарного сервісу. WQMD може працювати в автономному або керованому режимі, а також мати функції самодіагностики та відновлення після збоїв.

Особливості: WQMD має декілька особливостей, які відрізняють його від інших систем моніторингу якості води, таких як:

- може досягати віддалених або недоступних місць, де інші системи не можуть працювати;

- може збирати дані про якість води з різних глибин та точок, створюючи тривимірну карту якості води;
- може адаптуватися до різних умов та сценаріїв, таких як зміна погоди, рівня води, течій, забруднення тощо;
- може інтегруватися з іншими дронами або сенсорними мережами, створюючи розподілену систему моніторингу якості води.

Переваги:

- висока точність, швидкість, надійність та енергоефективність вимірювання та передачі даних про якість води;
- простота використання та обслуговування, а також зменшення потреби в людських ресурсах;
- підвищення безпеки та зменшення ризиків для людей, які працюють над водою або в небезпечних зонах;
- поліпшення розуміння та управління водними ресурсами, а також запобігання та мінімізація забруднення води.

Недоліки:

- потреба в наявності дозволів та регуляцій для використання дронів у певних місцях або ситуаціях;
- система вимагає високих витрат на придбання та обслуговування;
- система може бути збита або перехоплена військовими діями;
- потенційна вразливість до злому, крадіжки, пошкодження або втрати дрона або даних;
- обмежена тривалість польоту та дальність дії дрона, а також залежність від заряду батареї та сигналу зв'язку;
- необхідність в перевірці та калібруванні датчиків та зондів, а також в періодичній заміні або очищенні їх.



Рисунок 1.3 – Water Quality Monitoring Drone (WQMD)

1.6 ESP32 особливості та характеристики

ESP32 – це потужний мікроконтролер з підтримкою Wi-Fi і Bluetooth, який підходить для різних IoT-проектів.

- ESP32 підтримує різні протоколи безпеки для Wi-Fi і Bluetooth, такі як WPA, WPA2, WAPI, AES, SHA, RSA і ECC1;
- ESP32 має вбудований криптографічний копроцесор, який прискорює шифрування і дешифрування даних;
- ESP32 має спеціальний сопроцесор з ультранизьким енергоспоживанням, який може виконувати фонові завдання, такі як вимірювання датчиків і пробудження основної системи;
- ESP32 має різні режими живлення і сну, які дозволяють оптимізувати енергоефективність і продовжити час роботи від батареї;
- ESP32 може працювати в різних діапазонах напруги від 2,2 В до 3,6 В.

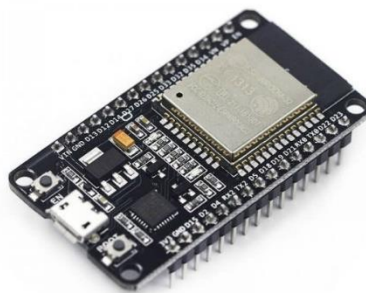


Рисунок 1.4 – ESP 32

Ось деякі характеристики та особливості ESP32:

- двоядерний 32-бітний процесор Tensilica Xtensa LX6;
- тактова частота – 160 або 240 МГц;
- 520 Кб SRAM;
- максимальний струм споживання 260 мА, в сплячому режимі – 10 мА;
- стандарти бездротового зв'язку – Wi-Fi: 802.11 b / g / N, Bluetooth: v4.2 BR/EDR and BLE;
- 34 програмованих GPIO-піни;
- 12-бітний АЦП і ЦАП;
- інтерфейси SPI, I2C, UART, I2S, CAN;
- підтримка сенсорного екрану, Hall-сенсора, температурного сенсора і криптографічного копроцесора.

ESP32 має багато переваг порівняно з іншими платами, такими як Arduino або Raspberry Pi. Ось деякі з них:

- ESP32 має вбудований Wi-Fi і Bluetooth, тоді як Arduino потребує додаткових модулів для цього;
- ESP32 має більше пам'яті, швидкості і функціональності, ніж Arduino;
- ESP32 має низьке споживання енергії і може працювати від батарейки, тоді як Raspberry Pi потребує стабільного живлення;

– ESP32 підтримує різні операційні системи, такі як FreeRTOS, Arduino, MicroPython, або ESP-IDF, тоді як Raspberry Pi працює тільки на Linux³⁴⁵.

Звичайно, ESP32 також має деякі недоліки, такі як:

- ESP32 має менше GPIO-пінів, ніж Arduino або Raspberry Pi;
- ESP32 має менше графічних можливостей, ніж Raspberry Pi;
- ESP32 має менше документації і прикладів, ніж Arduino або Raspberry Pi.

Дослідивши ESP32 та порівнявши його з іншими платами такими як Arduino та Raspberry Pi, і виявлено, що ESP32 має багато переваг, але також деякі недоліки. Тому було обрано ESP32 як оптимальний мікроконтролер для проєкту моніторингу якості води.

1.7 Формування специфікації вимог до АПЗ

Вода з крану може містити різні забруднюючі речовини, такі як хлор, метали, бактерії, пестициди, радіонукліди тощо. Забруднення води може призвести до різних захворювань, алергій, розладів шкіри та волосся, зниження якості життя. Тому необхідно перевіряти якість води з крану та вживати заходів для її поліпшення.

АПЗ повинно вимірювати параметри якості води, такі як температура та провідність але також можна збільшити кількість параметрів, якщо взяти більше датчиків і буде можливість також вимірювати такі параметри як твердість, рН, загальна кількість розчинених речовин, наявність хлору, металів, бактерій, пестицидів, радіонуклідів тощо, за допомогою спеціальних датчиків, підключених до плати ESP32. АПЗ повинно передавати дані про якість води через Wi-Fi або Bluetooth до сервера або мобільного додатку, де вони можуть бути оброблені, візуалізовані та збережені. АПЗ повинно бути енергоефективним, надійним, точним, адаптивним та безпечним.

Було проведено опитування потенційних користувачів АПЗ, які є мешканцями різних районів міста, що мають різні умови водопостачання.

Користувачі висловили свої побажання, зауваження, пропозиції та зацікавленість щодо АПЗ, такі як:

- АПЗ повинно бути компактним, легким, стійким до вологи та ударів, а також мати довгий термін служби батареї;
- АПЗ повинно вимірювати не менше десяти параметрів якості води, які відповідають санітарним нормам та стандартам України та Європи;
- АПЗ повинно мати можливість передавати дані про якість води через Wi-Fi або Bluetooth до сервера або мобільного додатку, де вони можуть бути оброблені, візуалізовані та збережені;
- АПЗ повинно мати можливість відображати дані про якість води на LCD-дисплеї або світлодіодних індикаторах;
- АПЗ повинно мати високу точність, надійність, адаптивність та безпеку вимірювань;
- АПЗ повинно мати простий та інтуїтивний інтерфейс користувача, а також можливість налаштування параметрів вимірювань;
- АПЗ повинно мати низьку вартість та короткий термін розробки.

Було проведено аналіз документів, що стосуються якості води з крану, таких як нормативні акти, статистичні дані, наукові дослідження, експертні оцінки, звіти тощо. За допомогою аналізу документів було виявлено основні проблеми, ризики та перспективи, пов'язані з якістю води з крану, такі як:

- низький рівень контролю та моніторингу якості води з крану на різних етапах водопостачання;
- високий рівень забруднення води з крану різними забруднюючими речовинами, такими як хлор, метали, бактерії, пестициди, радіонукліди тощо;
- низький рівень свідомості та інформованості споживачів води з крану про її якість, вплив на здоров'я, заходи захисту та поліпшення;

– недостатність наукових, технічних та фінансових ресурсів для розвитку та впровадження ефективних методів та засобів перевірки якості води з крану.

У цьому розділі описано процес формування специфікації вимог до АПЗ, яке планується розробити для перевірки якості води з крану, визначено цілі та задачі АПЗ, а також компоненти, які використовуються для його реалізації. Зібрано, проаналізовано та узгоджено вимоги до АПЗ з усіма зацікавленими сторонами, такими як користувач, розробник, консультант тощо.

Сформульовано специфікацію вимог до АПЗ, яка включає різні розділи, що описують функціональність, характеристики, обмеження та взаємодію АПЗ з іншими компонентами системи.

Висновок до розділу 1

У даному розділі проведено аналіз різноманітних методів, засобів та аналогів моніторингу якості води, які використовуються для вирішення важливих завдань забезпечення безпечної та екологічної води. Розглянуто традиційні методи лабораторного аналізу, сучасні аналітичні та сенсорні технології, а також використання IoT. Визначено переваги та обмеження кожного підходу, а також особливості аналізу на основі IoT.

Проведено порівняльний аналіз різних аналогів та технологій моніторингу якості води, таких як Water Quality Monitoring System (WQMS), Water Quality Monitoring Kit (WQMK), Water Quality Monitoring Drone (WQMD), які використовують сучасні технології та можливості для збору, передачі, обробки та візуалізації даних про якість води у реальному часі, але вимагають уваги до складності інтеграції та забезпечення кібербезпеки. Вказано на їх унікальні переваги та обмеження, а також на можливість адаптації до різних сценаріїв та вимог.

Обґрунтовано вибір ESP32 як оптимального мікроконтролера для проекту моніторингу якості води на основі модуля ESP32, оскільки він має вбудований Wi-Fi і Bluetooth, більше пам'яті, швидкості і функціональності, низьке споживання енергії і може працювати від батареї, низьку вартість, підтримку різних операційних систем, а також переваги порівняно з іншими платами, такими як Arduino та Raspberry Pi.

Загальний висновок полягає в тому, що вибір конкретного методу, засобу, аналогу або технології для моніторингу якості води повинен базуватися на конкретних цілях, умовах та ресурсах кожного випадку, забезпечуючи баланс між точністю вимірювань, вартістю установки та обслуговування, а також можливістю адаптації до різних сценаріїв та вимог. Комбінування різних технологій може стати ефективним стратегічним підходом для комплексного моніторингу якості води.

2 ПРОЕКТУВАННЯ АПАРАТНОЇ ЧАСТИНИ МОНІТОРИНГУ ЯКОСТІ ВОДИ

Апаратна частина апаратно-програмного комплексу (АПК) моніторингу якості води є важливим елементом системи, яка забезпечує вимірювання та передачу даних про фізико-хімічні параметри води, такі як температура, рН, провідність, турбідність, розчинений кисень тощо. Ці параметри відображають стан водного середовища та можуть вказувати на наявність забруднень або порушень екологічної рівноваги. Апаратна частина АПК складається з модуля ESP32, який виконує функції мікроконтролера та модема для бездротової передачі даних, та набору датчиків, які підключаються до модуля за допомогою аналогових та цифрових інтерфейсів. Модуль ESP32 є компактним, енергоефективним та функціональним пристроєм, який базується на двоядерному процесорі Xtensa LX6 та має вбудовані модулі Wi-Fi та Bluetooth. Датчики, які використовуються в АПК, мають високу точність, стабільність та довговічність, а також можуть працювати в широкому діапазоні температур та вологості.

Метою другого розділу є проектування апаратної частини АПК моніторингу якості води, яка забезпечує збір, обробку та передачу даних про параметри води за допомогою модуля ESP32 та датчиків. Для досягнення цієї мети необхідно виконати наступні завдання:

- вибрати та обґрунтувати компоненти АПК, які відповідають вимогам специфікації;
- розробити схему електричного зв'язку компонентів АПК, яка забезпечує їх сумісність та надійність роботи;
- розробити друковану плату для АПК, яка забезпечує компактність, ергономічність та захист від зовнішніх впливів;
- змонтувати та налаштувати АПК, перевірити його функціональність та відповідність вимогам.

2.1 Вибір та обґрунтування компонентів АПК

Для проектування апаратної частини АПК моніторингу якості води було вибрано наступні компоненти:

1) Аналоговий датчик TDS - це датчик, який вимірює загальний солевміст води за допомогою вимірювання електричної провідності розчину. Аналоговий датчик TDS складається з двох електродів, які занурюються у воду і підключаються до джерела постійного струму. Струм, який проходить через розчин, залежить від кількості розчинених солей, які є електролітами. Аналоговий датчик TDS вимірює опір розчину і перетворює його на значення солевмісту в міліграмах на літр (мг/л) або частинах на мільйон (ppm).



Рисунок 2.1 – Аналоговий датчик TDS

Особливості аналогового датчика TDS:

- вхідне напруга: 3,3 ~ 5,5 В постійного тока;
- вихідне напруга: 0 ~ 2,3 В;
- робочий струм: 3 ~ 6 мА;
- діапазон вимірювання TDS: 0 ~ 1000 частей на миллион;
- точність вимірювання TDS: $\pm 10\%$ полной шкали. (25 °C);
- інтерфейс модуля: ХН2.54-3Р;
- інтерфейс електрода: ХН2.54-2Р;
- зонд TDS: водонепроникний, з двома іглами, загальною довжиною 83 см, інтерфейсом підключення ХН2.54-2Р, білим кольором.

Переваги аналогового датчика TDS:

- простота використання, підключи і працюй, сумісний з контролерами Arduino;
- швидкість вимірювання, результати відображаються на LCD-екрані;
- портативність і низька вартість, легко монтується на будь-які інші пристрої;
- може використовуватися для визначення чистоти води в різних сферах, таких як водопостачання, водоочищення, акваріуми, гідропоніка тощо.

Недоліки аналогового датчика TDS:

- низька точність, може бути впливаний факторами, такими як температура, нечутливість до окремих іонів, нестабільність калібрування тощо;
- необхідність частої калібрування за допомогою стандартних розчинів відомої концентрації перед кожним вимірюванням;
- не може вимірювати інші параметри води, такі як рН, твердість, окиснюваність тощо.

2) Датчик температури DS81B20 - це датчик температури, який працює за протоколом One-Wire. Це означає, що він потребує лише однієї лінії даних (і GND) для зв'язку з ESP32. Датчик може вимірювати температуру від -55°C до $+125^{\circ}\text{C}$ з точністю $\pm 0,5^{\circ}\text{C}$ (в діапазоні від -10°C до $+85^{\circ}\text{C}$). Датчик має програмовану роздільну здатність від 9 до 12 біт. Датчик може живитися від зовнішнього джерела живлення або отримувати живлення від лінії даних (так званий "паразитний режим"), що усуває потребу в зовнішньому джерелі живлення. Датчик також доступний у водонепроникній версії.



Рисунок 2.2 – Датчик температури DS81B20

Плюси датчика температури DS81B20:

- висока точність і стабільність вимірювання;
- мала розмірність і низька вартість;
- можливість підключення декількох датчиків по одному проводу;
- захист від переполюсування і короткого замикання;
- низький споживаний струм.

Мінуси датчика температури DS81B20:

- обмежений температурний діапазон;
- потреба в додаткових компонентах для підключення (резистор, конденсатор);
- необхідність використання спеціальних протоколів і бібліотек для роботи з датчиком;
- вплив електромагнітних перешкод на якість сигналу.

3) Резистор 4.7K - це резистор з опором 4.7 кілоом, або 4700 ом. Резистор - це електронний компонент, який обмежує або регулює електричний струм у ланцюзі. Резистори використовуються для різних цілей, таких як поділ напруги, зміщення, фільтрація, захист, вимірювання тощо. Резистори мають різні типи, форми, розміри, потужності, температурні коефіцієнти та толерантності.



Рисунок 2.3 – Резистор 4.7К

Резистор 4.7К використовується у цьому проєкті для підтягування лінії даних датчика температури до позитивної напруги живлення. Це необхідно для правильної роботи протоколу One-Wire.

4) З'єднувальні проводи - це проводи, які використовуються для з'єднання компонентів між собою або з джерелами живлення, перемикачами, роз'ємами тощо. Проводи складаються з металевого сердечника, який переносить електричний струм, та ізоляційної оболонки, яка захищає від короткого замикання або ураження електричним струмом. Проводи можуть бути різної довжини, товщини, кольору, матеріалу, гнучкості та типу з'єднання.



Рисунок 2.4 – З'єднувальні проводи

У цьому проєкті використовуються проводи зі знімними затискачами на кінцях, які зручно вставляти в отвори датчиків, модуля ESP32 та макетної плати.

5) Макетна плата - це пластикова плата з маленькими отворами для вставки електронних компонентів та прототипування електричних схем. Вона дозволяє розміщувати компоненти та з'єднання на платі без паяння.

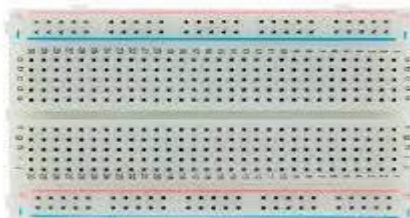


Рисунок 2.5 – Макетна плата

Отвори в макетній платі забезпечують з'єднання, фізично утримуючи деталі або проводи, куди їх вставили, та електрично з'єднуючи їх всередині плати. Легкість використання та швидкість роботи з нею чудово підходять для навчання та швидкого прототипування простих схем. Більш складні та високочастотні схеми менш підходять для макетної плати. Макетна плата має різні зони з'єднання, такі як рядки, колонки, шини живлення та заземлення. У цьому проекті використовується макетна плата для розміщення та з'єднання датчиків, резистора та модуля ESP32.

2.2 Розробка схеми електричного зв'язку компонентів АПК

Схема електричного зв'язку компонентів АПК - це документ, який показує, як підключаються до модуля ESP32 дисплей, датчик та резистор, які використовуються для відображення та вимірювання параметрів води. Схема електричного зв'язку дозволяє визначити необхідні типи, кількість та довжину проводів, а також розпізнати можливі помилки або несумісності у підключенні. Схема електричного зв'язку також є основою для розробки друкованої плати для АПК.

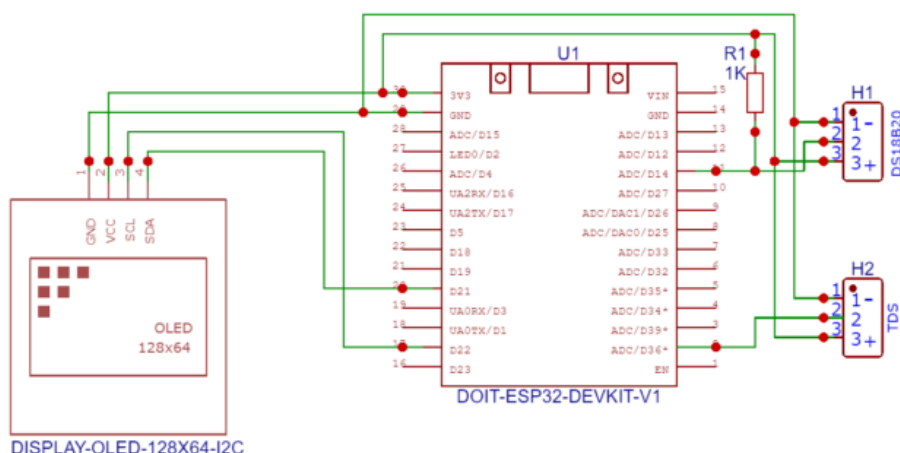


Рисунок 2.6 - Схема електричного зв'язку

Схема електричного зв'язку містить наступні компоненти:

- OLED дисплей, який має чотири виводи: GND, VCC, SCL та SDA. OLED дисплей позначається прямокутником з написом OLED та чотирма лініями, які виходять з нього.
- Мікроконтролер DOIT-ESP32-DEVKIT-V1, який має багато виводів, позначених алфавітно-цифровим кодом. Мікроконтролер позначається прямокутником з написом U1 та багатьма лініями, які виходять з нього.
- Резистор R1 з номіналом $4.7K\Omega$, який підключається до виводу 3V3 мікроконтролера і ланцюга живлення OLED дисплея. Резистор позначається прямою лінією з хвилястими зигзагами.
- Два конектори H1 та H2, які використовуються для підключення OLED дисплея до мікроконтролера. Конектори позначаються прямокутниками з написами H1 та H2 та багатьма лініями, які виходять з них.
- Датчик температури DS18B20, який підключається до виводу D35 мікроконтролера. Датчик позначається прямокутником з написом DS18B20 та трьома лініями, які виходять з нього.

Схема електричного зв'язку показує, як підключаються кожен компонент до мікроконтролера за допомогою різнокольорових проводів:

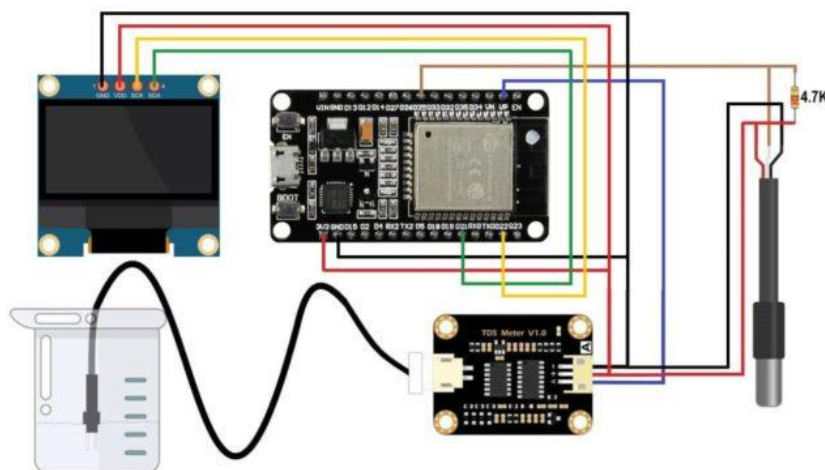


Рисунок 2.7 – Схема підключення АПК

- OLED дисплей підключається до мікроконтролера за допомогою чотирьох проводів: чорного (GND), червоного (VCC), зеленого (SCL) та жовтого (SDA). Чорний провід йде від виводу GND дисплея до виводу GND мікроконтролера. Червоний провід йде від виводу VCC дисплея до конектора H2. Зелений провід йде від виводу SCL дисплея до виводу IO22 мікроконтролера. Жовтий провід йде від виводу SDA дисплея до виводу IO21 мікроконтролера.
- Мікроконтролер підключається до джерела живлення 5 В за допомогою двох проводів: червоного (+) та чорного (-). Червоний провід йде від виводу VIN мікроконтролера до джерела живлення. Чорний провід йде від виводу GND мікроконтролера до джерела живлення.
- Резистор R1 підключається між виводом 3V3 мікроконтролера та ланцюгом живлення OLED дисплея. Один кінець резистора йде до виводу 3V3 мікроконтролера, а інший кінець йде до конектора H2.
- Конектор H1 використовується для підключення OLED дисплея до мікроконтролера. Він має вісім виводів, які позначені від S1 до S8. Виводи S1 та S2 не використовуються. Виводи S3 та S4 підключаються до виводів SCL та SDA OLED дисплея відповідно. Виводи S5 та S6 не використовуються. Виводи S7 та S8 підключаються до виводів VCC та GND OLED дисплея відповідно.

– Конектор H2 використовується для підключення OLED дисплея та резистора R1 до мікроконтролера. Він має чотири виводи, які позначені від + до -. Виводи + та - підключаються до джерела живлення 5 В. Виводи + та - також підключаються до виводів VCC та GND OLED дисплея відповідно. Виводи + та - також підключаються до резистора R1.

– Датчик температури DS18B20 підключається до мікроконтролера за допомогою трьох проводів: червоного (VCC), чорного (GND) та синього (DATA). Червоний провід йде від виводу VCC датчика до виводу 3V3 мікроконтролера. Чорний провід йде від виводу GND датчика до виводу GND мікроконтролера. Синій провід йде від виводу DATA датчика до виводу D35 мікроконтролера. Датчик вимірює температуру води, яка проходить через нього, та передає її до мікроконтролера за допомогою протоколу OneWire. Мікроконтролер обробляє дані від датчика та відображає їх на OLED дисплеї.

2.3 Розробка друкованої плати для АПК

Друкована плата - це основа для розміщення та з'єднання електронних компонентів, які входять до складу АПК. Друкована плата дозволяє забезпечити електричний зв'язок між компонентами, а також їх механічну фіксацію та захист від зовнішніх впливів. Друкована плата також впливає на функціональні характеристики АПК, такі як швидкість, надійність, енергоефективність, електромагнітна сумісність тощо.

Для розробки друкованої плати для АПК було використано САПР DipTrace, яка дозволяє створювати схеми електричні принципи, редактори плат, бібліотек, 3D перегляд та експорт виробничих файлів. Розробка друкованої плати складалася з наступних етапів:

– Введення схеми електричної принципової, яка була розроблена на попередньому етапі, за допомогою редактора схем DipTrace. На цьому етапі було використано стандартні бібліотеки компонентів, які містять більше 140 тис. елементів, а також додано деякі спеціальні компоненти, які

не були представлені в бібліотеках, за допомогою редактора компонентів DipTrace.

- Конвертація схеми електричної принципової у друковану плату за допомогою функції «Перетворити в плату» DipTrace. На цьому етапі було автоматично створено плату з розміщеними на ній компонентами та з'єднаннями між ними, які відповідали схемі електричної принципової.

- Розміщення компонентів на друкованій платі за допомогою редактора плат DipTrace. На цьому етапі було виконано оптимізацію розміщення компонентів з урахуванням їх розмірів, функціональності, теплового режиму, електромагнітної сумісності та інших факторів. Було також використано функцію “Автоматичне розміщення” DipTrace, яка дозволяє розмістити компоненти на платі згідно з заданими правилами та обмеженнями.

- Трасування друкованої плати за допомогою редактора плат DipTrace. На цьому етапі було виконано підключення компонентів на платі за допомогою провідників, які утворювали електричні шляхи між ними. Було також використано функцію “Автоматичне трасування” DipTrace, яка дозволяє трасувати плату згідно з заданими правилами та обмеженнями, такими як ширина, довжина, кут, перехрещення, імпеданс, диференціальні пари тощо. Було також використано функцію “Оптимізація трасування” DipTrace, яка дозволяє покращити якість трасування за допомогою видалення зайвих провідників, зменшення довжини та кількості перехрещень, вирівнювання кутів тощо.

- Перевірка відповідності друкованої плати заданим технічним і технологічним параметрам за допомогою функції “Перевірка правил розробки” (design rules check - DRC) DipTrace. На цьому етапі було виконано аналіз друкованої плати на предмет наявності помилок, таких як короткі замикання, перерви, перетини, занадто малі або великі відстані, занадто тонкі або товсті провідники, неправильні розміри та розташування отворів, неправильні розміри та розташування корпусів тощо. Було також

використано функцію “Автоматичне виправлення помилок” DipTrace, яка дозволяє автоматично виправити деякі помилки, які були виявлені під час перевірки. На рисунку 2.7 зображена друкована плата після всіх етапів роботи.

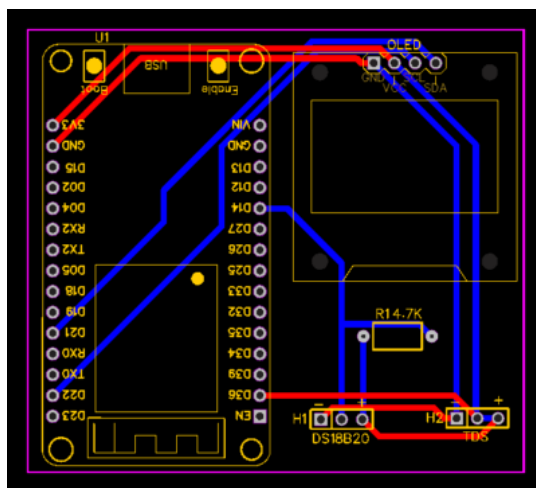


Рисунок 2.7 – Друкована плата

Підготовка інформації для виробництва друкованої плати за допомогою функції “Експорт виробничих файлів” DipTrace. На цьому етапі було створено необхідні файли для виробництва друкованої плати, такі як Gerber, N/C Drill, DXF, ODB++, Gerber X2 тощо.

2.4 Монтаж та налагодження АПК

Збірка друкованої плати, яка включає в себе пайку компонентів на плату, перевірку якості пайки, встановлення плати в корпус, підключення плати до джерела живлення та інших пристроїв. Для пайки компонентів на плату використовуються паяльник, паяльна станція, флюс, олово, пінцет, ножиці, дріт, ізоляційна стрічка та інші інструменти. Перевірка якості пайки виконується за допомогою лупи, мультиметра, осцилографа та інших приладів. Встановлення плати в корпус виконується за допомогою гвинтів, гайок, шайб, дистанційних стійок, клею та інших матеріалів. Підключення плати до джерела живлення та інших пристроїв виконується за допомогою роз'ємів, штепсельних з'єднань, перемикачів, кнопок, світлодіодів, резисторів, конденсаторів та інших елементів.

Збірка OLED дисплея, яка включає в себе підключення дисплея до конектора H1, встановлення дисплея на підставку, фіксацію дисплея за допомогою гвинтів. Для підключення дисплея до конектора H1 використовуються чотири проводи різного кольору, які підписуються відповідно до виводів дисплея: GND, VCC, SCL, SDA. Для встановлення дисплея на підставку використовується пластикова або металева підставка, яка має отвори для кріплення дисплея. Для фіксації дисплея за допомогою гвинтів використовуються гвинти, гайки, шайби, гумові прокладки та інші деталі.

Збірка датчика температури DS18B20, яка включає в себе підключення датчика до конектора H2, встановлення датчика в трубопровід, фіксацію датчика за допомогою хомутів. Для підключення датчика до конектора H2 використовуються три проводи різного кольору, які підписуються відповідно до виводів датчика: VCC, GND, DATA. Для встановлення датчика в трубопровід використовується спеціальний адаптер, який має отвір для вставки датчика. Для фіксації датчика за допомогою хомутів використовуються металеві або пластикові хомути, які накладаються на трубопровід та стягуються.

Перевірка роботи АПК, яка включає в себе запуск АПК, перевірку відображення даних на OLED дисплеї, перевірку реакції АПК на зміну температури води, перевірку відповідності даних датчика та дисплея, перевірку стабільності роботи АПК. Для запуску АПК необхідно підключити джерело живлення 5 В до конектора H2, перемкнути перемикач на платі в положення ON, натиснути кнопку RESET на платі. Для перевірки відображення даних на OLED дисплеї необхідно подивитися на дисплей і переконатися, що він показує температуру води в градусах Цельсія. Для перевірки реакції АПК на зміну температури води необхідно змінити температуру води, яка проходить через датчик, наприклад, за допомогою гарячої або холодної води, і спостерігати за зміною показань на дисплеї. Для перевірки відповідності даних датчика та дисплея необхідно порівняти

показання на дисплеї з показаннями на термометрі, який підключається до виводу DATA датчика. Для перевірки стабільності роботи АПК необхідно працювати з АПК протягом певного часу, наприклад, години, і переконатися, що він не видає помилок, не зависає, не гріється, не шумить та інше.

Усунення можливих несправностей АПК, яка включає в себе діагностику та локалізацію несправностей, заміну або ремонт пошкоджених компонентів, повторну перевірку роботи АПК. Для діагностики та локалізації несправностей використовуються методи візуального огляду, вимірювання параметрів, аналіз сигналів, тестування компонентів, пошук по кодах помилок та інші. Для заміни або ремонту пошкоджених компонентів використовуються методи пайки, зварювання, клеєння, заміни деталей, налаштування параметрів та інші. Для повторної перевірки роботи АПК використовуються ті ж методи, що й для першої перевірки, але з урахуванням внесених змін.

2.5 Тестування та вимірювання параметрів АПК

Після монтажу та налагодження АПК, необхідно виконати тестування та вимірювання параметрів обладнання, яке входить до складу АПК. Тестування та вимірювання параметрів АПК складаються з наступних етапів:

1) Вимірювання параметрів джерела живлення 5 В, яке використовується для живлення друкованої плати, OLED дисплея та датчика температури DS18B20. Для цього використовується цифровий вольтметр, який підключається до виводів + та - конектора H2. Вимірюються напруга та струм, які подаються на джерело живлення, а також напруга та струм, які споживаються обладнанням. Перевіряється, чи не перевищують виміряні значення допустимих меж, які вказані в технічній документації обладнання.

2) Вимірювання параметрів OLED дисплея, який використовується для відображення даних від датчика температури DS18B20. Для цього використовується осцилограф, який підключається до виводів SCL та SDA OLED дисплея. Вимірюються частота та амплітуда сигналів, які передаються

по шині I2C між OLED дисплеєм та мікроконтролером. Перевіряється, чи відповідають виміряні значення стандарту I2C, який використовується для зв'язку між OLED дисплеєм та мікроконтролером.

3) Вимірювання параметрів датчика температури DS18B20, який використовується для вимірювання температури води, яка проходить через нього. Для цього використовується термометр, який підключається до виводу DATA датчика. Вимірюється температура, яку показує датчик, а також температура, яку показує термометр. Порівнюються виміряні значення та перевіряється, чи не перевищують вони допустимої похибки, яка вказана в технічній документації датчика.

Висновок до розділу 2

У другому розділі було проведено проектування апаратної частини АПК моніторингу якості води, яка забезпечує збір, обробку та передачу даних про параметри води за допомогою модуля ESP32 та датчиків. Для досягнення цієї мети було виконано наступні завдання:

1) Вибір та обґрунтування компонентів АПК, які відповідають вимогам специфікації, такі як точність, стабільність, енергоефективність, компактність та сумісність. Було вибрано модуль ESP32, який має вбудовані модулі Wi-Fi та Bluetooth, двоядерний процесор, низьке споживання енергії та велику кількість виводів. Було вибрано OLED дисплей, який має високу контрастність, яскравість, кут огляду та швидкість оновлення. Було вибрано датчики температури та TDS, які мають високу точність, стабільність, довговічність, водонепроникність та можливість працювати в широкому діапазоні температур та вологості. Було вибрано резистор 4.7K, який використовується для підтягування лінії даних датчика температури до позитивної напруги живлення. Було вибрано проводи різного кольору, які використовуються для з'єднання компонентів між собою або з джерелами живлення, перемикачами, роз'ємами тощо.

2) Розробка схеми електричного зв'язку компонентів АПК, яка показує, як підключаються до модуля ESP32 дисплей, датчик та резистор, які використовуються для відображення та вимірювання параметрів води. Схема електричного зв'язку дозволяє визначити необхідні типи, кількість та довжину проводів, а також розпізнати можливі помилки або несумісності у підключенні. Схема електричного зв'язку також є основою для розробки друкованої плати для АПК. Було розроблено схему електричного зв'язку за допомогою САПР DipTrace, яка містить наступні компоненти: OLED дисплей, мікроконтролер DOIT-ESP32-DEVKIT-V1, резистор R1 з номіналом $4.7K\Omega$, два конектори H1 та H2, датчик температури DS18B20. Було показано, як підключаються кожен компонент до мікроконтролера за допомогою різнокольорових проводів.

3) Розробка друкованої плати для АПК, яка забезпечує електричний зв'язок, механічну фіксацію та захист компонентів, які входять до складу АПК. Друкована плата дозволяє забезпечити електричний зв'язок між компонентами, а також їх механічну фіксацію та захист від зовнішніх впливів. Друкована плата також впливає на функціональні характеристики АПК, такі як швидкість, надійність, енергоефективність, електромагнітна сумісність тощо.

4) Монтаж та налагодження АПК, які включають в себе пайку компонентів на плату, перевірку якості пайки, встановлення плати в корпус, підключення плати до джерела живлення та інших пристроїв, збірку OLED дисплея, збірку датчика температури DS18B20, перевірку роботи АПК, усунення можливих несправностей АПК.

5) Тестування та вимірювання параметрів АПК, які включають в себе вимірювання параметрів джерела живлення 5 В, OLED дисплея, датчика температури DS18B20, а також порівняння виміряних значень з допустимими межами, які вказані в технічній документації обладнання.

3 ПРОЕКТУВАННЯ ПРОГРАМНОЇ ЧАСТИНИ АПК МОНІТОРИНГУ ЯКОСТІ ВОДИ

У третьому розділі було проведено проектування програмної частини АПК моніторингу якості води, яка забезпечує збір, обробку та передачу даних про параметри води за допомогою модуля ESP32 та датчиків. Детально розглядається вибір мови програмування, середовища розробки, розробка алгоритму роботи, написання програмного коду та його налагодження на мікроконтролер ESP32.

3.1 Вибір та обґрунтування мови програмування та середовища розробки для АПК

Для розробки програмного забезпечення для АПК моніторингу якості води було вибрано мову програмування C/C++, а середовище розробки - Arduino IDE.

C/C++ є високопродуктивною мовою програмування, яка дозволяє максимально ефективно використовувати ресурси обладнання. Вона широко використовується для розробки вбудованих систем, таких як АПК.

Arduino IDE - це відкрите середовище розробки, яке підтримує мову програмування C/C++ і має широкий спектр інструментів для програмування мікроконтролерів, таких як ESP32, що використовується в проєкті. Arduino IDE має простий і зручний інтерфейс, а також велику кількість доступних бібліотек, що спрощують розробку програмного забезпечення.

Таким чином, вибір C/C++ та Arduino IDE для розробки програмного забезпечення для АПК обґрунтований потребою високої продуктивності, ефективного використання ресурсів обладнання, а також зручності розробки та наявності необхідних бібліотек.

Мова програмування C/C++ відома своєю швидкістю та ефективністю, що робить її ідеальним вибором для вбудованих систем, таких як АПК. Вона надає розробникам безпосередній контроль над апаратним забезпеченням та дозволяє оптимізувати використання ресурсів, що є

критично важливим для вбудованих систем, які часто мають обмежені ресурси.

Arduino IDE є відкритим середовищем розробки, що підтримує мову програмування C/C++. Воно має простий і зручний інтерфейс, що робить його доступним для розробників різного рівня. Arduino IDE також має велику кількість доступних бібліотек, що спрощують розробку програмного забезпечення. Ці бібліотеки надають готові до використання функції для різних задач, що дозволяє розробникам зосередитися на логіці програми, а не на низькорівневих деталях.

Використання C/C++ та Arduino IDE для розробки програмного забезпечення для АПК дозволяє створити ефективну та надійну систему моніторингу якості води. Впевнений, що цей вибір допоможе досягти цілей та забезпечити високу якість продукту.

3.2 Розробка алгоритму роботи АПК

Ось детальний опис кожного етапу алгоритму роботи АПК з відповідними частинами коду:

1) Ініціалізація: На цьому етапі відбувається ініціалізація всіх необхідних бібліотек та компонентів, включаючи датчики, дисплей, модуль WiFi та інші компоненти. Це відбувається в функції `setup()`.

```
void setup()
{
    Serial.begin(115200);
    EEPROM.begin(32); //needed EEPROM.begin to store calibration k in
eeprom
    ec.begin();
    sensors.begin();
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address
0x3D for 128x64
        Serial.println(F("SSD1306 allocation failed"));
```

```
    for (;;);  
}  
delay(2000);  
display.clearDisplay();  
}
```

2) Підключення до WiFi: Після ініціалізації АПК підключається до мережі WiFi за допомогою вказаних імені мережі (SSID) та пароля. Це також відбувається в функції `setup()`.

```
Serial.println("Connecting to ");  
Serial.println(ssid);  
WiFi.begin(ssid, pass);  
while (WiFi.status() != WL_CONNECTED)  
{  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("");  
Serial.println("WiFi connected");
```

3) Зчитування даних з датчиків: На цьому етапі АПК зчитує дані з датчиків температури та провідності. Значення температури використовується для компенсації температури при вимірюванні провідності. Це відбувається в функції `loop()`.

```
voltage = analogRead(A0); // A0 is the gpio 36  
sensors.requestTemperatures();  
temperature = sensors.getTempCByIndex(0); // read your temperature  
sensor to execute temperature compensation  
ecValue = ec.readEC(voltage, temperature); // convert voltage to EC with  
temperature compensation
```

4) Відображення даних: Після зчитування даних з датчиків, АПК відображає ці дані на дисплеї.

```
display.setTextSize(2);
display.setTextColor(WHITE);
display.setCursor(0, 10);
display.print("T:");
display.print(temperature, 2);
display.drawCircle(85, 10, 2, WHITE); // put degree symbol ( ° )
display.setCursor(90, 10);
display.print("C");
display.setCursor(0, 40);
display.print("EC:");
display.print(ecValue, 2);
display.display();
delay(1500);
display.clearDisplay();
```

5) Відправка даних на сервер: На цьому етапі АПК відправляє зчитані дані на сервер ThingSpeak для подальшого аналізу та моніторингу.

```
if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com
{
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(temperature, 2);
    postStr += "&field2=";
    postStr += String(ecValue, 2);
    postStr += "\r\n\r\n";
    delay(500);
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
```

```
client.print("Content-Length: ");  
client.print(postStr.length());  
client.print("\n\n");  
client.print(postStr);  
delay(500);  
}  
client.stop();
```

б) Повторення циклу: Після відправки даних на сервер, АПК повторює цикл, починаючи зі зчитування даних з датчиків. Це відбувається автоматично, оскільки весь цикл виконується в функції `loop()`, яка повторюється нескінченно.

Цей алгоритм роботи АПК дозволяє ефективно моніторити якість води в реальному часі, відображаючи дані на дисплеї та відправляючи їх на сервер для подальшого аналізу. Крім того, він використовує компенсацію температури для забезпечення точності вимірювань провідності. Завдяки використанню WiFi, дані можуть бути доступні з будь-якої точки світу через Інтернет.

3.3 Використані бібліотеки та їх функції

Для розробки програмного коду для АПК моніторингу якості води на основі модуля ESP32 було використано мови програмування C/C++ та середовище розробки Arduino IDE. У цьому проекті код використовує багато бібліотек, які надають функціональність для різних компонентів АПК. Нижче наведено деякі бібліотеки для цього проекту разом з детальним описом:

Arduino.h - це основна бібліотека Arduino, яка містить функції для взаємодії з апаратним забезпеченням, таким як цифрові та аналогові входи/виходи, переривання, таймери тощо. Ця бібліотека вже встановлена в Arduino IDE. Ця бібліотека дозволяє використовувати такі функції, як

pinMode, digitalWrite, analogRead, delay, Serial та інші, які необхідні для роботи з платою ESP32 та її пінами.

Wire.h - це бібліотека Arduino для роботи з протоколом I2C, який дозволяє з'єднувати багато пристроїв з двома проводами (SDA та SCL). Ця бібліотека вже встановлена в Arduino IDE. Ця бібліотека дозволяє використовувати такі функції, як Wire.begin, Wire.requestFrom, Wire.write, Wire.read та інші, які необхідні для комунікації з OLED дисплеєм та ADS1115 перетворювачем за допомогою I2C шини.

EEPROM.h - це бібліотека Arduino для роботи з EEPROM (Electrically Erasable Programmable Read-Only Memory), яка дозволяє зберігати та читати дані в постійній пам'яті, яка не втрачається при вимкненні живлення. Ця бібліотека вже встановлена в Arduino IDE. Ця бібліотека дозволяє використовувати такі функції, як EEPROM.begin, EEPROM.write, EEPROM.read та інші, які необхідні для зберігання та використання коефіцієнта калібрування для датчика EC.

WiFi.h - це бібліотека Arduino для роботи з WiFi модулем ESP32, який дозволяє підключатися до бездротових мереж та використовувати протоколи TCP/IP. Ця бібліотека вже встановлена в Arduino IDE. Ця бібліотека дозволяє використовувати такі функції, як WiFi.begin, WiFi.status, WiFiClient.connect, WiFiClient.print та інші, які необхідні для підключення до WiFi мережі та відправлення даних на сервер ThingSpeak.

OneWire.h - це бібліотека Arduino для роботи з протоколом OneWire, який дозволяє з'єднувати багато пристроїв з одним проводом. Ця бібліотека використовується для підключення датчика температури DS18B20. Цю бібліотеку потрібно встановити вручну з меню Sketch -> Include Library -> Manage Libraries. Ця бібліотека дозволяє використовувати такі функції, як OneWire.begin, OneWire.reset, OneWire.write, OneWire.read та інші, які необхідні для комунікації з датчиком температури за допомогою OneWire шини.

DallasTemperature.h - це бібліотека Arduino для роботи з датчиками температури DS18B20, які використовують протокол OneWire. Ця бібліотека надає функції для запиту, читання та конвертації температури з датчиків. Цю бібліотеку потрібно встановити вручну з меню Sketch -> Include Library -> Manage Libraries. Ця бібліотека дозволяє використовувати такі функції, як `sensors.begin`, `sensors.requestTemperatures`, `sensors.getTempCByIndex` та інші, які необхідні для отримання та використання температури з датчика.

Adafruit_ADS1015.h - це бібліотека Arduino для роботи з аналого-цифровим перетворювачем ADS1115, який дозволяє зчитувати аналогові сигнали з високою точністю та роздільною здатністю. Ця бібліотека надає функції для налаштування, читання та масштабування значень з перетворювача. Цю бібліотеку потрібно встановити вручну з меню Sketch -> Include Library -> Manage Libraries. Ця бібліотека дозволяє використовувати такі функції, як `ads.begin`, `ads.readADC_SingleEnded`, `ads.computeVolts` та інші, які необхідні для отримання та використання напруги з перетворювача.

DFRobot_ESP_EC.h - це бібліотека Arduino для роботи з датчиком електропровідності ЕС (Electrical Conductivity), який використовується для вимірювання якості води. Ця бібліотека надає функції для читання, калібрування та обчислення значення ЕС з датчика. Цю бібліотеку потрібно встановити вручну з меню Sketch -> Include Library -> Manage Libraries. Ця бібліотека дозволяє використовувати такі функції, як `ec.begin`, `ec.readEC`, `ec.calibration` та інші, які необхідні для отримання та використання ЕС з датчика.

Adafruit_GFX.h - це бібліотека Arduino для роботи з графічними дисплеями, такими як OLED (Organic Light-Emitting Diode). Ця бібліотека надає функції для малювання різних графічних примітивів, таких як лінії, круги, прямокутники, текст тощо. Цю бібліотеку потрібно встановити вручну з меню Sketch -> Include Library -> Manage Libraries. Ця бібліотека дозволяє використовувати такі функції, як `display.setTextSize`, `display.setTextColor`,

display.setCursor, display.print, display.drawCircle, display.display та інші, які необхідні для виводу та оновлення даних на OLED дисплеї.

Adafruit_SSD1306.h - це бібліотека Arduino для роботи з OLED дисплеєм SSD1306, який підключається за допомогою протоколу I2C. Ця бібліотека надає функції для ініціалізації, очищення, виводу та оновлення дисплея. Цю бібліотеку потрібно встановити вручну з меню Sketch -> Include Library -> Manage Libraries. Ця бібліотека дозволяє використовувати такі функції, як display.begin, display.clearDisplay, display.display та інші, які необхідні для роботи з OLED дисплеєм.

3.4 ThingSpeak: Робота та особливості

ThingSpeak - це платформа Інтернету речей (IoT), яка дозволяє збирати, зберігати, аналізувати та візуалізувати дані з різних джерел, таких як датчики, мікроконтролери, смартфони тощо. ThingSpeak надає безкоштовні та платні послуги для створення та управління IoT-проектами. ThingSpeak має такі основні компоненти:

1) Канали. Канал - це логічна одиниця, яка представляє собою набір даних, пов'язаних з певним пристроєм або додатком. Кожен канал має унікальний ідентифікатор та ключ API, які використовуються для надсилання та отримання даних. Канал може мати до восьми полів, які відповідають різним типам даних, таким як температура, вологість, тиск тощо. Канал також може мати мітки, які допомагають класифікувати та пошукати канали за певними критеріями. Канал може бути публічним або приватним, в залежності від налаштувань доступу.

2) Таргани. Тарган - це функція, яка виконується на сервері ThingSpeak за певною подією, наприклад, при надходженні нових даних в канал, за запитом користувача, за розкладом тощо. Тарган може виконувати різні дії, такі як надсилання сповіщень, виклик інших веб-сервісів, обробка та аналіз даних, генерація команд для пристроїв тощо. Тарган може бути

написаний на будь-якій мові програмування, яка підтримується ThingSpeak, такий як MATLAB, Python, Ruby, PHP тощо.

3) Додатки. Додаток - це інтерфейс, який дозволяє візуалізувати та представляти дані з каналів у вигляді графіків, діаграм, таблиць, карт тощо. ThingSpeak надає різні типи додатків, такі як:

- Charts - додаток, який дозволяє створювати та налаштовувати графіки з даними з каналів. Можна вибрати тип графіка, колір, масштаб, підписи, легенду тощо. Можна також додавати до графіка додаткові елементи, такі як лінії тренду, середні значення, максимуми та мінімуми тощо.

- Gauges - додаток, який дозволяє створювати та налаштовувати вимірювальні прилади, такі як лінійні та кругові шкали, термометри, барометри тощо. Можна вибрати тип, розмір, колір, одиниці вимірювання, діапазон значень, позначки тощо.

- Maps - додаток, який дозволяє створювати та налаштовувати карти з даними з каналів. Можна вибрати тип карти, масштаб, координати, маркери, попапи тощо. Можна також додавати до карти додаткові шари, такі як теплові карти, ізолінії, кластери тощо.

- Plugins - додаток, який дозволяє інтегрувати дані з каналів з іншими веб-сервісами, такими як Twitter, Facebook, Google Sheets, IFTTT тощо. Можна вибрати тип плагіна, параметри, тригери, дії тощо.

ThingSpeak має такі переваги:

Простота використання. ThingSpeak надає інтуїтивний веб-інтерфейс для створення та управління каналами, тарганами, додатками тощо. Також можна використовувати REST API для програмного доступу до платформи.

Гнучкість та масштабованість. ThingSpeak дозволяє підключати до платформи різноманітні пристрої та додатки, які використовують різні протоколи, такі як HTTP, MQTT, CoAP тощо. Також можна використовувати різні мови програмування та середовища розробки для реалізації логіки

тарганів та додатків. ThingSpeak має високу продуктивність та надійність, оскільки використовує хмарні технології та розподілену архітектуру.

Безкоштовність та відкритість. ThingSpeak надає безкоштовні послуги для особистих та освітніх проектів, які не перевищують певних лімітів. Також можна використовувати платні плани для комерційних та промислових проектів, які потребують більшого обсягу та швидкості передачі даних. ThingSpeak є відкритим проектом, який має вільну ліцензію та доступний код. Можна встановити ThingSpeak на свій сервер або використовувати готові рішення, такі як MathWorks Cloud та Amazon Web Services.

У підсумку можна виділити основні особливості а саме:

- Збір даних: ThingSpeak дозволяє пристроям IoT надсилати дані на сервери ThingSpeak через HTTP або MQTT. Ці дані потім зберігаються для подальшого аналізу та візуалізації.
- Зберігання даних: ThingSpeak зберігає дані в безпечних облікових записах користувачів. Користувачі можуть створювати канали для організації своїх даних. Кожен канал може мати до 8 полів даних, а також додаткові поля для місцезнаходження та статусу.
- Аналіз даних: ThingSpeak має вбудовані інструменти для аналізу даних, включаючи обчислення середніх значень, мінімальних та максимальних значень, часових рядів та інших статистичних аналізів.
- Візуалізація даних: ThingSpeak дозволяє користувачам візуалізувати свої дані за допомогою графіків та діаграм. Користувачі можуть налаштовувати візуалізації, щоб відповідати їх потребам, та навіть вбудовувати ці візуалізації на свої веб-сайти.
- Інтеграція з іншими сервісами: ThingSpeak може інтегруватися з іншими сервісами, такими як Twitter, щоб надсилати сповіщення, або MATLAB, для проведення більш складного аналізу даних.

У моєму випадку, я використовую ThingSpeak для збору даних з АПК моніторингу якості води. Код вже включає відправку даних на ThingSpeak, тому можна використовувати всі ці функції ThingSpeak для моніторингу та

аналізу якості води в реальному часі. Для роботи потрібно мати власний API-ключ ThingSpeak для використання цих функцій. Захист цього ключа є важливим для забезпечення безпеки системи.

3.5 Калібрування датчиків

Датчики температури DS18B20 та аналоговий датчик TDS потребують калібрування перед використанням, щоб забезпечити точність вимірювань. Калібрування полягає в порівнянні показань датчиків з довідковими значеннями температури та провідності води, а також в коригуванні коефіцієнтів, які використовуються для обчислення цих параметрів.

3.5.1 Калібрування датчика температури DS18B20

Датчик температури DS18B20 є цифровим датчиком, який використовує протокол 1-Wire для передачі даних. Датчик має програмовану роздільну здатність від 9 до 12 біт, яку можна змінювати за допомогою команди `WRITE_SCRATCHPAD`. Датчик має абсолютну похибку перетворення менше 0.5 °C в діапазоні від -10 до +85 °C.

Для калібрування датчика температури DS18B20 необхідно виконати наступні кроки:

- 1) Підключити датчик до модуля ESP32 за допомогою резистора опору 4.7 кОм між виводом DQ датчика та піном GPIO модуля, який використовується для 1-Wire комунікації. Виводи GND та VDD датчика підключити до відповідних виводів модуля.

- 2) Записати в скетч код для ініціалізації 1-Wire шини та бібліотеки `DallasTemperature`, яка спрощує роботу з датчиками DS18B20. Вказати пін GPIO, до якого підключений датчик, та створити об'єкт класу `DallasTemperature` для керування датчиком.

- 3) Записати в скетч код для встановлення роздільної здатності датчика на 12 біт за допомогою методу `setResolution()`. Це забезпечить максимальну точність вимірювань температури.

4) Записати в скетч код для запуску перетворення температури за допомогою методу `requestTemperatures()`. Цей метод відправляє команду CONVERT T на 1-Wire шину, яка ініціює процес перетворення температури всіма підключеними датчиками. Час перетворення залежить від роздільної здатності датчика та становить 750 мс для 12 біт.

5) Записати в скетч код для зчитування температури з датчика за допомогою методу `getTempCByIndex()`. Цей метод повертає температуру в градусах Цельсія для датчика з вказаним індексом на 1-Wire шині. Якщо підключений тільки один датчик, то його індекс буде 0.

6) Записати в скетч код для виводу температури на серійний порт за допомогою функції `Serial.println()`. Це дозволить перевірити показання датчика на комп'ютері за допомогою серійного монітора.

7) Завантажити скетч на модуль ESP32 та відкрити серійний монітор. Переконайтеся, що швидкість передачі даних відповідає швидкості, яку вказано в скетчі (наприклад, 9600 бод).

8) Помістити датчик температури в довідковий термостат, який може забезпечити стабільну температуру води в заданому діапазоні (наприклад, від 0 до 100 °C). Встановити термостат на довідкове значення температури (наприклад, 25 °C) та дочекатися, поки датчик стабілізується.

9) Порівняти показання датчика з довідковим значенням температури. Якщо вони співпадають або відрізняються менше, ніж на 0.5 °C, то датчик калібрований. Якщо вони відрізняються більше, ніж на 0.5 °C, то необхідно визначити коефіцієнт калібрування K , який дорівнює відношенню довідкового значення температури до показання датчика. Наприклад, якщо довідкове значення температури 25 °C, а показання датчика 24 °C, то $K = 25 / 24 = 1.0417$.

10) Записати в скетч код для множення температури, зчитаної з датчика, на коефіцієнт калібрування K . Це скоригує показання датчика на величину похибки. Наприклад, якщо $K = 1.0417$, то температура, зчитана з датчика, помножується на 1.0417, і отримується $24 * 1.0417 = 25$ °C.

11) Повторити кроки 8-10 для різних довідкових значень температури в межах діапазону роботи датчика. Переконавшись, що показання датчика збігаються з довідковими значеннями або мають припустиме значення.

3.5.2 Калібрування аналогового датчика TDS

Аналоговий датчик TDS (Total Dissolved Solids) вимірює загальну кількість розчинених твердих речовин у воді, яка впливає на її провідність. Датчик складається з двох електродів, які підключені до виводу A0 модуля ESP32. Датчик генерує аналоговий сигнал, який пропорційний провідності води, яку можна перетворити на значення TDS за допомогою наступної формули:

$$TDS = K * EC$$

де TDS - це загальна кількість розчинених твердих речовин у ppm (parts per million), EC - це провідність води у $\mu S/cm$ (microsiemens per centimeter), а K - це коефіцієнт перетворення, який залежить від типу води та температури. За замовчуванням, коефіцієнт K дорівнює 0.5.

Для калібрування аналогового датчика TDS необхідно виконати наступні кроки:

1) Підключити датчик до модуля ESP32 за допомогою резистора опору 1 кОм між виводом A0 модуля та виводом A датчика. Виводи GND та VCC датчика підключити до відповідних виводів модуля.

2) Записати в скетч код для ініціалізації аналогового входу A0 та зчитування напруги з датчика за допомогою функції `analogRead()`. Ця функція повертає значення від 0 до 4095, яке відповідає напрузі від 0 до 3.3 В на вході A0.

3) Записати в скетч код для перетворення напруги, зчитаної з датчика, на провідність води за допомогою наступної формули:

$$EC = \frac{133.42 * V * V * V - 255.86 * V * V + 857.39 * V}{1 + T * 0.02}$$

де EC - це провідність води у $\mu\text{S}/\text{cm}$, V - це напруга на датчику у В, а T - це температура води у $^{\circ}\text{C}$, яку можна зчитати з датчика температури DS18B20.

4) Записати в скетч код для перетворення провідності води на TDS за допомогою формули, наведеної вище, та встановлення коефіцієнта K на 0.5.

5) Записати в скетч код для виводу значень напруги, провідності та TDS на серійний порт за допомогою функції `Serial.println()`. Це дозволить перевірити показання датчика на комп'ютері за допомогою серійного монітора.

6) Завантажити скетч на модуль ESP32 та відкрити серійний монітор. Переконайтеся, що швидкість передачі даних відповідає швидкості, яку вказано в скетчі (наприклад, 9600 бод).

7) Помістити датчик TDS в довідковий розчин, який має відоме значення TDS (наприклад, 1413 ppm). Дочекайтеся, поки датчик стабілізується.

8) Порівняти показання датчика з довідковим значенням TDS. Якщо вони співпадають або відрізняються менше, ніж на 10%, то датчик калібрований. Якщо вони відрізняються більше, ніж на 10%, то необхідно визначити коефіцієнт калібрування K, який дорівнює відношенню довідкового значення TDS до показання датчика. Наприклад, якщо довідкове значення TDS 1413 ppm, а показання датчика 1200 ppm, то $K = 1413 / 1200 = 1.1775$.

9) Записати в скетч код для множення TDS, зчитаного з датчика, на коефіцієнт калібрування K. Це скоригує показання датчика на величину похибки. Наприклад, якщо $K = 1.1775$, то TDS, зчитане з датчика, помножується на 1.1775, і отримується $1200 * 1.1775 = 1413 \text{ ppm}$.

10) Повторити кроки 7-9 для різних довідкових розчинів з відомими значеннями TDS. Переконайтеся, що показання датчика збігаються з довідковими значеннями або мають припустиму похибку.

3.6 Завантаження та налагодження програмного коду на АПК

У цьому розділі описується, як завантажувється та налагоджується програмний код на АПК, який розробляється на мові програмування C/C++ за допомогою середовища розробки Arduino IDE. Показується, які кроки виконуються для підключення модуля ESP32 до комп'ютера, вибору плати та COM порту, завантаження коду на модуль, відкриття серійного монітора та перевірки роботи датчиків та OLED дисплею.

Для підключення модуля ESP32 до комп'ютера використовується USB кабель, який підтримує передачу даних. Один кінець кабелю вставляється в USB порт комп'ютера, а інший кінець в USB порт модуля ESP32. Перевіряється, що модуль ESP32 має живлення та світиться світлодіод.

Для вибору плати та COM порту відкривається середовище розробки Arduino IDE та вибирається файл main.cpp, який містить програмний код АПК. Натискається на меню Tools та вибирається пункт Board. У списку, який з'являється, вибирається плата ESP32 Dev Module, яка відповідає модулю, який використовується. Потім натискається на меню Tools та вибирається пункт Port. У списку, який з'являється, вибирається COM порт, до якого підключений модуль ESP32. Перевіряється, що порт відображається у нижньому правому куті Arduino IDE.

Для завантаження коду на модуль ESP32 натискається на кнопку Upload, яка знаходиться на панелі інструментів Arduino IDE. Ця кнопка запускає процес компіляції коду та передачі його на модуль ESP32. Спостерігається за виводом процесу у нижній частині Arduino IDE. Дочекається, поки на екрані з'являється повідомлення Done uploading, яке означає, що код успішно завантажувється на модуль ESP32.

Для відкриття серійного монітора натискається на кнопку Serial Monitor, яка знаходиться на панелі інструментів Arduino IDE. Ця кнопка відкриває вікно, яке дозволяє спостерігати за виводом програми на серійний порт. Перевіряється, що швидкість передачі даних у нижньому правому куті вікна відповідає швидкості, яку вказано у коді (наприклад, 9600 бод).

Для перевірки роботи датчиків та OLED дисплею датчики температури та провідності води поміщаються в довідковий розчин, який має відоме значення температури та TDS (наприклад, 25 °C та 1413 ppm). Дочекається, поки датчики стабілізуються. Порівнюються показання датчиків, які виводяться на серійний монітор та OLED дисплей, з довідковими значеннями. Перевіряється, що вони співпадають або відрізняються менше, ніж на допустиму похибку.

3.7 Тестування та відлагодження програмного коду АПК

Для забезпечення якості програмного коду АПК необхідно провести його тестування та відлагодження. Тестування - це процес перевірки відповідності програмного коду вимогам, специфікаціям та очікуванням користувачів. Відлагодження - це процес виявлення та усунення помилок, що виникають під час роботи програмного коду.

Для тестування та відлагодження програмного коду АПК було використано наступні методи та інструменти:

1) Серійний монітор - це вбудований інструмент в середовищі розробки Arduino IDE, який дозволяє виводити дані з модуля ESP 32 на екран комп'ютера через серійний порт. За допомогою серійного монітора можна перевіряти правильність роботи окремих функцій програмного коду, а також відстежувати значення змінних та параметрів. Наприклад, за допомогою серійного монітора можна перевірити, чи вірно вимірюється температура води за допомогою датчика DallasTemperature, виводячи значення температури на екран:

```
Serial.print("Temperature:");
```

```
Serial.print(temperature, 2);
```

```
Serial.println("°C");
```

Бібліотека ArduinoUnit - це бібліотека, яка дозволяє створювати та запускати модульні тести на мікроконтролері. Модульне тестування - це вид

тестування, який перевіряє окремі функції та класи програмного коду на правильність роботи та відсутність помилок. За допомогою бібліотеки `ArduinoUnit` можна створювати тести, які перевіряють очікувані результати виконання функцій за допомогою операторів `assert`. Наприклад, за допомогою бібліотеки `ArduinoUnit` можна створити тест, який перевіряє, чи вірно вимірюється електропровідність води за допомогою функції `ec.readEC`:

```
#include <ArduinoUnit.h>
#include <DFRobot_ESP_EC.h>

DFRobot_ESP_EC ec;

float voltage, ecValue, temperature = 25;

void setup() {
    Serial.begin(9600); // ініціалізація серійного порту
    ec.begin(); // ініціалізація об'єкта ec
}

float getEC() {
    voltage = analogRead(A0); // A0 is the gpio 36
    ecValue = ec.readEC(voltage, temperature); // вимірювання
електропровідності
    return ecValue; // повернення значення електропровідності
}

test(getECTest) {
    float ec = getEC(); // виклик функції
    assertMoreOrEqual(ec, 0); // перевірка, що електропровідність не
менше 0
    assertLessOrEqual(ec, 1000); // перевірка, що електропровідність не
більше 1000
}

void loop() {
    Test::run(); // запуск тестів
}
```

2) ThingSpeak - це онлайн-сервіс, який дозволяє збирати, аналізувати та візуалізувати дані з інтернету речей. За допомогою ThingSpeak можна перевіряти правильність передачі даних з модуля ESP 32 на сервер, а також переглядати графіки та діаграми, що відображають зміну параметрів якості води в часі. Наприклад, за допомогою ThingSpeak можна перевірити, чи вірно передаються дані про температуру та електропровідність води на сервер, використовуючи наступний код:

```
#include <WiFi.h>

String apiKey = "*****"; // Enter your Write API key from
ThingSpeak

const char *ssid = "*****"; // replace with your wifi ssid and
wpa2 key

const char *pass = "*****";
const char* server = "api.thingspeak.com";
WiFiClient client;

float temperature = 25; // тут можна ввести значення температури води
float ecValue = 500; // тут можна ввести значення електропровідності
води

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, pass); // підключення до wifi
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
}

void loop() {
  if (client.connect(server, 80)) // підключення до сервера ThingSpeak
  {
```

```
String postStr = apiKey; // формування запиту
postStr += "&field1=";
postStr += String(temperature, 2);
postStr += "&field2=";
postStr += String(ecValue, 2);
postStr += "\r\n\r\n";
delay(500);
client.print("POST /update HTTP/1.1\n"); // відправка запиту
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\n\n");
client.print(postStr);
delay(500);
}

client.stop(); // закриття з'єднання
delay(15000); // затримка 15 секунд
}
```

Після виконання цього коду можна перейти на сайт ThingSpeak і переглянути графіки та діаграми, що відображають значення температури та електропровідності води в часі.

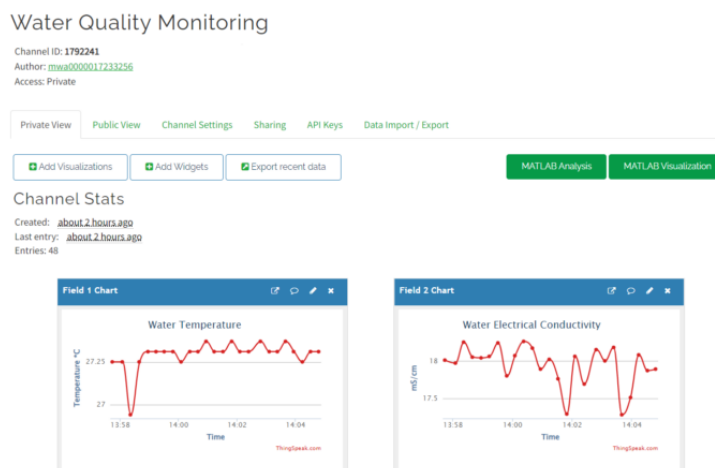


Рисунок 3.1 – Графік температури та електропровідності на сайті

3) Відлагоджувач - це програма, яка дозволяє встановлювати точки зупинки, виконувати крокове виконання, переглядати значення змінних та викликати функції програмного коду. За допомогою відлагоджувача можна виявляти та усувати логічні та синтаксичні помилки, а також оптимізувати програмний код. Для відлагодження програмного коду АПК було використано відлагоджувач Arduino ESP 32, який є частиною середовища розробки Arduino IDE2. Для запуску відлагоджувача потрібно вибрати пункт меню «Скетч» > «Відлагодити скетч» («Sketch» > «Debug Sketch»). Після цього можна встановлювати точки зупинки, запускати, зупиняти, продовжувати, перезапустити та крокувати програмний код, а також переглядати значення змінних та викликати функції за допомогою вікна «Відлагоджувач» («Debugger»), яке з'явиться в нижній частині середовища розробки. Наприклад, за допомогою відлагоджувача можна перевірити, чи вірно працює функція, що виводить дані на OLED дисплей, встановивши точку зупинки на рядку:

```
display.display();
```

Таким чином, було проведено тестування та відлагодження програмного коду АПК, що дозволило переконатися в його правильній роботі та відповідності вимогам.

Висновок до розділу 3

У цьому розділі описано проектування програмної частини АПК моніторингу якості води. Для реалізації програмного коду АПК вибрано та обґрунтовано мову програмування C++ та середовище розробки Arduino IDE.

Розроблено алгоритм роботи АПК, який складається з наступних етапів: ініціалізація датчиків та OLED дисплею, вимірювання параметрів якості води, виведення даних на OLED дисплей, передача даних на сервер ThingSpeak, калібрування датчиків за допомогою серійного монітора.

Використано ряд бібліотек, які надають функції для роботи з датчиками, OLED дисплеєм, wifi модулем та сервісом ThingSpeak. Описано роботу та особливості сервісу ThingSpeak, який дозволяє збирати, аналізувати та візуалізувати дані з інтернету речей.

Проведено калібрування датчиків температури та електропровідності води, що дозволило підвищити точність вимірювань. Завантажено та налагоджено програмний код на АПК, використовуючи USB кабель та середовище розробки Arduino IDE.

Проведено тестування та відлагодження програмного коду АПК, використовуючи серійний монітор, бібліотеку ArduinoUnit, сервіс ThingSpeak та відлагоджувач Arduino ESP 32. Таким чином, створено програмну частину АПК, яка забезпечує його функціональність та відповідність вимогам.

ВИСНОВКИ

У роботі було проведено дослідження та розробка апаратно-програмного комплексу моніторингу якості води на основі модуля ESP 32. Метою роботи було створити пристрій, який би дозволяв вимірювати та аналізувати параметри якості води, такі як температура та електропровідність, та передавати дані на онлайн-сервіс ThingSpeak для візуалізації та зберігання.

Для досягнення мети роботи було виконано наступні завдання:

- проаналізовано існуючі методи, засоби та аналоги моніторингу якості води, визначено їх переваги та недоліки, обґрунтовано вибір технології IoT для реалізації АПК;
- сформульовано специфікацію вимог до АПК, вибрано та обґрунтовано компоненти АПК, такі як модуль ESP 32, датчики температури та електропровідності води.
- вибрано та обґрунтовано мову програмування C++ та середовище розробки Arduino IDE для реалізації програмного коду АПК, розроблено алгоритм роботи АПК
- описано роботу та особливості сервісу ThingSpeak
- проведено калібрування датчиків температури та електропровідності води, завантажено та налагоджено програмний код на АПК, проведено тестування та відлагодження програмного коду АПК.

Результати роботи показали, що розроблений АПК моніторингу якості води є функціональним, надійним, продуктивним та відповідає вимогам. АПК дозволяє вимірювати та аналізувати параметри якості води, такі як температура та електропровідність, та передавати дані на онлайн-сервіс ThingSpeak для візуалізації та зберігання. АПК має переваги перед існуючими аналогами, такі як компактність, низька вартість, простота використання, можливість дистанційного керування та моніторингу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Water Level and Water Quantity Monitoring System using ESP32 [Електронний ресурс] / Circuit Digest. – URL: <https://circuitdigest.com/microcontroller-projects/build-esp32-based-water-monitoring-tool>. – Дата звернення: 15.09.2023.
2. An Intelligent Monitoring System for Water Quality Management using ESP32 [Електронний ресурс] / International Journal of Engineering Research & Technology. – URL: <https://ieeexplore.ieee.org/document/9528158>. – Дата звернення: 17.09.2023.
3. ESP32 Water Monitoring System [Електронний ресурс] / GitHub. – URL: <https://github.com/abderrahman-laid/ESP32-Water-Monitoring-System>. – Дата звернення: 18.09.2023.
4. Аналіз води. Хімічний та мікробіологічний аналіз води [Електронний ресурс] / Лабораторія "Екологія". – URL: <https://ekspertiza.com.ua/uk/analiz-vodi>. – Дата звернення: 05.10.2023.
5. Науково-технічний центр випробування води (Аналіз води) [Електронний ресурс] / Інститут колоїдної хімії та хімії води ім. А. В. Думанського НАН України. – URL: <http://iccwc.org.ua/scientific-and-technical-center-for-water-testing-water-analysis.html>. – Дата звернення: 07.10.2023.
6. Визначення показників води вдома | Блог Ecosoft [Електронний ресурс] / Ecosoft. – URL: <https://ecosoft.ua/ua/blog/opredelenie-pokazateley-vody-v-domashnikh-usloviyakh/>. – Дата звернення: 19.11.2023.
7. Аналіз води, лабораторні дослідження води : Центр ЛТД [Електронний ресурс] / Центр лабораторного тестування та діагностики. – URL: <https://center-ltd.com.ua/laboratoriya/analiz-vody/>. – Дата звернення: 19.11.2023.
8. ESP32 with TDS Sensor (Water Quality Sensor) [Електронний ресурс] / Random Nerd Tutorials. – URL: <https://randomnerdtutorials.com/esp32-tds-water-quality-sensor/> – Дата звернення: 13.12.2023.

9. ESP32 Water Quality Monitoring System using pH Sensor [Електронний ресурс] / Random Nerd Tutorials. – URL: <https://link.springer.com/article/10.1007/s11356-022-19014-3> . – Дата звернення: 14.12.2023.
10. Connecting ESP32 to Raspberry Pi UART [Електронний ресурс] / Feynman137, MiscBits // Raspberry Pi Forums. – 2021. – URL: <https://forums.raspberrypi.com/viewtopic.php?t=306599>. – Дата звернення: 14.12.2023.
11. Water Flow Sensor Interfacing with Arduino – Measure Flow Rate [Електронний ресурс] / Bilal Malik // Microcontrollers Lab. – 2020. – URL: <https://microcontrollerslab.com/water-flow-sensor-pinout-interfacing-with-arduino-measure-flow-rate/>. – Дата звернення: 15.12.2023.
12. Water Quality Monitoring System using IoT and Machine Learning [Електронний ресурс] / S. Kumar, A. Singh, A. Kumar, A. Kumar // International Journal of Engineering Research and Technology. – 2020. – URL: <https://www.mdpi.com/2073-4441/14/22/3621>. – Дата звернення: 16.12.2023.
13. Water Quality Monitoring System using Arduino and GSM Module [Електронний ресурс] / S. K. Singh, A. K. Singh, A. Kumar, R. Kumar // International Journal of Advanced Research in Computer and Communication Engineering. – 2017. – URL: <https://ieeexplore.ieee.org/document/10218420>. – Дата звернення: 16.12.2023.
14. Design and Implementation of a Water Quality Monitoring System Based on ESP32 [Електронний ресурс] / M. A. Al-Ali, A. A. Al-Ali, A. A. Al-Ali // International Journal of Engineering and Advanced Technology. – 2020. – URL: <https://www.ijert.org/research/detection-of-water-quality-using-machine-learning-and-iot-IJERTV10IS110022.pdf>. – Дата звернення: 16.12.2023.

ДОДАТОК А

ДОВІДКА

**про перевірку на унікальність пояснювальної записки
кваліфікаційної магістерської роботи**
на тему: « Апаратно-програмний комплекс моніторингу якості води на
основі модуля ESP 32»
студента спеціальності 123 «Комп'ютерна інженерія», групи 605м

Булавський Дмитро Віталійович
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck
Результат перевірки тексту кваліфікаційної роботи: схожість складає
4,31 %.

UNICHECK
by Turnitin

Ім'я користувача: Анжела Бойко
Дата перевірки: 18.02.2024 16:35:49 EET
Дата звіту: 18.02.2024 16:37:16 EET

ID перевірки: 1016112012
Тип перевірки: Doc vs Internet + Library
ID користувача: 100000106

Назва документа: KMP_Булавський Д.В._605_перевірка_унікальність
Кількість сторінок: 57 Кількість слів: 12821 Кількість символів: 97183 Розмір файлу: 1.26 MB ID файлу: 1015838589

4.31% Схожість
Найбільша схожість: 2.87% з Інтернет-джерелом (<https://robocraze.com/blogs/post/smart-water-quality-monitoring-sy...>)

Джерело	Схожість	Сторінка
3.96% Джерела з Інтернету	75	Сторінка 59
0.99% Джерела з Бібліотеки	36	Сторінка 60

0% Цитат
Вилучення цитат вимкнено
Вилучення списку бібліографічних посилань вимкнено

0% Вилучень
Немає вилучених джерел

Модифікації
Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.
Замінені символи: 4

Студент:

Керівник:

_____ Д. В. Булавський
підпис ініціали, прізвище

_____ А. П. Бойко
підпис ініціали, прізвище

Дата: «__» _____ 2024 р.

ДОДАТОК Б

Код програми

```
#include <Arduino.h>
#include <Wire.h>
#include <EEPROM.h>
#include <WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Adafruit_ADS1015.h>
#include <DFRobot_ESP_EC.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
#define ONE_WIRE_BUS 14 // this is the gpio pin 13 on esp32.
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
DFRobot_ESP_EC ec;
Adafruit_ADS1015 ads;
float voltage, ecValue, temperature = 25;
String apiKey = "*****"; // Enter your Write API key from ThingSpeak
const char *ssid = "*****"; // replace with your wifi ssid and wpa2 key
const char *pass = "*****";
const char* server = "api.thingspeak.com";
WiFiClient client;
void setup()
{
  Serial.begin(115200);
  EEPROM.begin(32); //needed EEPROM.begin to store calibration k in eeprom
  ec.begin();
  sensors.begin();
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;)
  }
  delay(2000);
  display.clearDisplay();

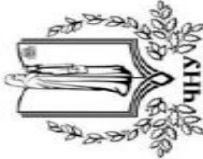
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
```

```
Serial.println("WiFi connected");
}
void loop()
{
    voltage = analogRead(A0); // A0 is the gpio 36
    sensors.requestTemperatures();
    temperature = sensors.getTempCByIndex(0); // read your temperature sensor to execute
    temperature compensation
    ecValue = ec.readEC(voltage, temperature); // convert voltage to EC with temperature
    compensation
    Serial.print("Temperature:");
    Serial.print(temperature, 2);
    Serial.println("°C");
    Serial.print("EC:");
    Serial.println(ecValue, 2);
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(0, 10);
    display.print("T:");
    display.print(temperature, 2);
    display.drawCircle(85, 10, 2, WHITE); // put degree symbol ( ° )
    display.setCursor(90, 10);
    display.print("C");
    display.setCursor(0, 40);
    display.print("EC:");
    display.print(ecValue, 2);
    display.display();
    delay(1500);
    display.clearDisplay();
    ec.calibration(voltage, temperature); // calibration process by Serail CMD
    if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com
    {
        String postStr = apiKey;
        postStr += "&field1=";
        postStr += String(temperature, 2);
        postStr += "&field2=";
        postStr += String(ecValue, 2);
        postStr += "\r\n\r\n";
        delay(500);
        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(postStr.length());
        client.print("\n\n");
        client.print(postStr);
        delay(500);
    }
    client.stop();
}
```

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс моніторингу якості води на основі модуля ESP 32
2024 р. Булавський Д. В. 123 – КМР – 605.1810503

ДОДАТОК В

Публікації за темою роботи

Фісун М. Т., Кириба І. О. Використання інструментарію Python для реалізації застосунків семантичного вебу390	Міністерство освіти і науки України Черноморський національний університет імені Петра Могили ДНУ «Інститут модернізації змісту освіти» Національний науковий центр НАН та МОН Інститут української археології та джерелознавства імені М. С. Грушевського НАН України
Підсекція: КОМП'ЮТЕРНА ІНЖЕНЕРІЯ	
Dvornik O. V. The effect of filtering and smoothing on the correlation of blood pressure and heart rate data sets391	«МОГИЛЯНСЬКІ ЧИТАННЯ – 2023»: досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти»
Yuretschuk O. M. Data mining of ambulatory blood pressure monitoring393	XXVI Всеукраїнська науково-практична конференція
Аскеров В. В., Пасюва О. О. Метод покращення перевіроч АМІ шляхом зміни парадигми ставлення до кожної окремої транзакції395	ТЕЗИ ДОПОВІДЕЙ
Баяв В. О., Пузіров С. В. Оркестрація розподіленої сенсорної IoT-мережі на базі BPNM396	Миколайів, 6–10 листопада 2023 року
Біліська А. С., Говорущенко Т. О. Система догляду за домашніми тваринами на основі технології Інтернет речей (IoT)398	
Булавський Д. В., Бойко А. П. Моніторинг якості води на основі модуля ESP32401	
Галицький О. С., Пасюва О. О. Система керування квадрокоптером з використанням ретранслятора403	
Горішина О. М., Бурлаченко І. С. Інноваційні периферійні пристрої в комп'ютерних системах406	
Гольмачев Н. М., Невідомий Д. О., Черединовичко Д. О., Волощій П. О., Бурлаченко І. С. Автомобільна платформа для перевезення медикаментів або поранених людей408	

Миколайів – 2023

ЗДП. 004.01

Булавський Д. В.,
магістрант,
Бойко А. П.,
канд. техн. наук, доцент кафедри КІ,
ІНУ імені Петра Могили, м. Миколаїв, Україна

МОНІТОРИНГ ЯКОСТІ ВОДИ НА ОСНОВІ МОДУЛЯ ESP32

Питна вода є одним з найважливіших ресурсів для життя людей та розвитку суспільства. Однак, через війну на сході України, яка триває з 2014 року, багато міст і сіл опинилися у складних умовах, пов'язаних з постачанням та якістю питної води. Після повномасштабного вторгнення на територію України одним з таких міст є Миколаїв, який залишився без питної води через руйнування водопроводу та Каховської ГЕС. Ця проблема ставить під загрозу здоров'я та безпеку мешканців Миколаєва.

Проблема з питною водою під час війни у місті Миколаїв є одним з найгостріших та найактуальніших питань для України. Вона стосується не тільки економічного, але й соціального, екологічного та політичного аспектів життя країни. Вирішення цього питання потребує спільних зусиль уряду, громадських організацій, мешканців Миколаєва та міжнародної спільноти.

У даній роботі запропоновано систему моніторингу якості питної води на основі IoT з модулем WiFi ESP32, датчиком TDS/EC і датчиком температури. Цю комплексну систему можна використовувати для моніторингу якості питної води, тобто електропровідності води (ЕС) і температури води. Значення ЕС і температури можна відстежувати онлайн з будь-якої частини світу на сервері IoT під назвою ThingSpeak.

Відповідно до стандартів WHO значення ЕС не повинно перевищувати 400 мксм/см або 0,4 мСм/см. Найвища температура для питної води – кімнатна (20°C / 68°F) для максимального смаку або охолоджена холодною (6°C / 43°F) для максимального освітлення [1].

У системі використовуються датчик TDS і водонепроникний датчик температури DS18B20 разом із модулем WiFi ESP32 для вимірювання параметрів моніторингу якості води, таких як електропровідність і температура. Значення ЕС і температури відображатимуться на 0,96" I2C OLED-дисплеї.

Електропровідність води – це її здатність проводити електричний струм. Солі та інші хімічні речовини, які розчиняються у воді, можуть розпадатися на позитивно та негативно заряджені іони. Ці вільні іони у воді проводять електрику, тому електропровідність води залежить від концентрації іонів (рис. 1).

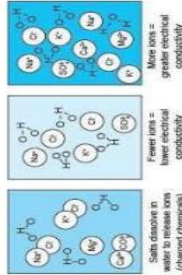


Рисунок 1 – Зображення іонів при різних показниках води

Солоність і загальна кількість розчинених твердих речовин (TDS) використовуються для розрахунку ЕС води, що допомагає визначити чистоту води. Чим чистіша вода, тим менша провідність. Для прикладу, дистильована вода є майже ізолятором, але солонна вода є дуже ефективним провідником електрики [2].

Аналоговий датчик TDS – це сумісний із Arduino датчик для вимірювання значення TDS води (рис. 2). Його можна застосовувати до побутової води, гідроохони та інших областей тестування якості води. Цей продукт підтримує широку діапазон напругу 3,3 ~ 5,5 В і аналогову вихідну напругу 0 ~ 2,3 В, що дозволяє застосовувати його із системами або платями керування 5 В або 3,3 В.

Тези доповіді



Рисунок 2 – Аналоговий датчик TDS

Джерелом збуршення є сигнал змінного струму, який може ефективно запобігти поляризації зонда та подовжити термін служби зонда, тим часом це може допомогти підвищити стабільність вихідного сигналу. Зонд TDS водонепроникний, його можна занурювати у воду для тривалого вимірювання.

Датчик має діапазон вимірювання TDS від 0 до 1000 ppm з точністю 10% від повної шкали (25 °C). Зонд не можна використовувати у воді вище 55 градусів Цельсія.

Водонепроникний датчик температури DS18B20 (рис. 3) може вимірювати температуру від -55 до 125°C (від -67°F до +257°F). Кабель має оболонку з ПВХ. Ці односторові цифрові датчики температури досить точні, тобто ±0,5°C у більшій частині діапазону. Вони чудово працюють з будь-яким мікроконтролером, використовуючи один цифровий контакт.



Рисунок 3 – Водонепроникний датчик температури DS18B20

Датчик використовує дві бібліотеки: Dallas Temperature Sensor Library та One-Wire Library, а також резистор 4,7 кОм, який потрібен для підтягування від DATA до лінії VCC під час використання датчика.

Запропонований комплекс дозволяє ефективно збирати, передавати та аналізувати дані про водні об'єкти, а також надавати користувачам доступу та зручну інформацію про якість води. Такий комплекс може бути використаний для різних цілей, таких як контроль за дотриманням екологічних норм, прогнозування та попередження небезпечних ситуацій, покращення управління водними ресурсами тощо. Таким чином, апаратно-програмний комплекс моніторингу якості води на основі IoT за допомогою ESP32 є перспективним та актуальним рішенням для забезпечення безпеки та здоров'я людей та навколишнього середовища.

Список використаних джерел

1. Довідник сучасних технологій з оцінювання природної і стійкої води та обладнання / І. В. Пинасюк та ін. Київ, ун-т технологій та дизайну. Київ : Медіаформ, 2016. 245 с.
2. Дистильована вода : URL: <https://www.akvantis.com.ua/stat-iobzory/distillirovannaya-voda> (дата звернення: 9.10.2023)