Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Комплексна бакалаврська дипломна робота на тему:

«ІОТ-СИСТЕМА МОНІТОРИНГУ ПОКАЗНИКІВ СТАНУ ПРИРОДНИХ ВОДОЙМ. ПІДСИСТЕМА ЗБОРУ ДАНИХ»

Виконав: студент 4 курсу, групи 2ІСТ-20б спеціальності 126 – «Інформаційні системи та технології»
_____ Богдан КОНОТОП
Керівник: к.т.н., доц. каф. САІТ
_____ Дмитро ПРОЦЕНКО
«31» 05 2024 р.
Рецензент: к.т.н., доц. каф. КН
_____ Володимир ОЗЕРАНСЬКИЙ
«10» 06 2024 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«03» 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
д.т.н., проф. Віталій МОКІН
"05" 03 2024 року

ЗАВДАННЯ

НА КОМПЛЕКСНУ БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ

СТУДЕНТУ

Конотопу Богдану Петровичу

1. Тема роботи: «IoT-система моніторингу показників стану природних водойм. Підсистема збору даних»
керівник роботи: Проценко Д. П., к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «11» 03 2024 року №80
2. Термін подання студентом роботи 31.05
3. Вихідні дані до роботи:
Дані зібрані за допомогою: станція Sigfox, модуль зв'язку SFM10R1P, сенсори температури DS18B20, ph DFRobot, каламутності SEN0189;
4. Зміст текстової частини:
1) Загальна характеристика об'єкту дослідження. Аналіз архітектурних рішень інформаційних систем моніторингу стану природних водойм;
2) Аналіз елементів IoT-системи моніторингу показників стану природних водойм;
3) Розробка технології збору даних та їх первинної обробки з використанням хмарних сервісів. Результати експериментальних досліджень.
4) Розширення функціоналу IoT-системи.
5. Перелік ілюстративного матеріалу:
1) Класифікація параметрів моніторингу води;
2) Мікропроцесорна плата Arduino Uno;
3) Цифровий сенсор температури DS18B20;
4) Сенсор pH від компанії DF Robot;
5) Модуль каламутності SEN0189;
6) Модуль Sigfox та антена;

7) Зібраний пристрій;
8) Схема моделі пристрою в середовищі Proteus;
9) Проведення моніторингу води на території озера прилеглого до ВНТУ;
10) Система моніторингу якості води у роботі;
11) Структурна схема запропонованого пристрою;
12) Блок-схема алгоритму роботи запропонованої програми.

6. Консультанти розділів роботи:

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

7. Дата видачі завдання 11.03

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	11.03	31.03	вик
2	Порівняльний аналіз IoT-систем моніторингу показників стану природних водойм	01.04	15.04	вик
3	Розроблення IoT-системи показників стану природних водойм	16.04	30.04	вик
4	Проведення експериментального дослідження розробленої системи	01.05	15.05	вик
5	Оформлення матеріалів до захисту БДР	16.05	31.05	вик

Студент _____ Богдан КОНОТОП

Керівник роботи _____ Дмитро ПРОЦЕНКО

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 77 сторінок формату А4, на яких є 43 рисунка, список використаних джерел містить 24 найменувань.

Метою роботи є дослідження та розширення функціоналу інтернет речей системи моніторингу показників стану природних водойм, а саме розроблення підсистеми збору даних.

В роботі наведено дослідження підсистеми збору даних для IoT-системи моніторингу стану природних водойм. Робота націлена на вдосконалення через інтеграцію нових сенсорів і розробку програмного забезпечення для обробки і передачі даних

Ключові слова: IoT-система, моніторинг водойм, водні ресурси, вимірювання якості води, Sigfox Cloud, ThingSpeak, інтеграція сенсора, забруднення води, програмування Arduino.

ABSTRACT

The bachelor's thesis consists of 77 A4 pages, which contain 43 figures, the list of sources used contains 24 titles.

The objective of this work is to explore and expand the functionality of the Internet of Things system for monitoring the status of natural water bodies, namely the development of a data collection subsystem.

The work presents research on the data collection subsystem for an IoT system monitoring the status of natural water bodies. The work is aimed at improvement through the integration of new sensors and the development of software for data processing and transmission.

Keywords: IoT system, water body monitoring, water resources, water quality measurement, Sigfox Cloud, ThingSpeak, sensor integration, water pollution, Arduino programming.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасному світі індустріалізація невпинно зростає, внаслідок чого фабрики та заводи викидають значні обсяги шкідливих відходів, що призводить до забруднення природних водойм. Враховуючи критичне значення чистої води для всіх живих істот, потреба у точному моніторингу стану водних ресурсів стає особливо актуальною. Саме тому обрана тема бакалаврської дипломної роботи, спрямована на вдосконалення методів вимірювання та моніторингу забруднення води через розробку IoT-системи. Вона дозволяє не лише виявляти негативні зміни в екології водойм, але й вчасно реагувати на потенційні екологічні загрози, що сприяє ефективному управлінню водними ресурсами та захисту біорізноманіття.

Мета і задачі дослідження. Метою роботи є підвищення інформативності та ефективності моніторингу показників стану природних водойм за рахунок впровадження IoT-системи, яка реалізується з використанням сучасних технологій.

Для досягнення поставленої мети необхідно розв’язати такі задачі:

– провести аналіз існуючих IoT-систем моніторингу стану природних вод та визначити ключові параметри;

– проаналізувати елементи вибраної IoT-системи;

– здійснити експериментальні для розробленого на кафедрі САІТ пристрою моніторингу для перевірки працездатності системи;

– використати хмарні сервіси для збору та первинної обробки даних, зокрема ThingSpeak, та провести аналіз отриманих результатів;

– розробити методику інтеграції додаткових сенсорів для підвищення ефективності збору даних та моніторингу води;

– вдосконалити функціональні можливості IoT-системи шляхом розробки нових архітектурних рішень та програмного коду.

Об’єктом дослідження є процеси зміни показників стану природніх водойм, а саме: річки, озера, протоки, ставки, джерела і т.д.

Предметом дослідження є методи та технології збору, обробки та передачі даних про якість води в IoT-системі моніторингу природних водойм.

Публікації. За результатами даної роботи була зроблена доповідь на тему «IoT система моніторингу поверхневих вод» на VI Міжнародній науковій студентській конференції Молодіжної наукової ліги (2024) з публікацією тез [1].

Робота виконувалась на замовлення науково-дослідної лабораторії автоматизованих систем управління в енергетиці та транспорті (НДЛ АСУЕТ) кафедри САІТ Вінницького національного технічного університету.

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕННЯ. АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ ІНФОРМАЦІЙНИХ СИСТЕМ МОНІТОРИНГУ СТАНУ ПРИРОДНИХ ВОДОЙМ

1.1 Опис об'єкта досліджень

Об'єктом дослідження є стан річок, озер, ставків та інших природних водойм, тому що заводи та інші промислові організації скидають стічні води, що містять різні хімічні сполуки, які можуть погіршити якість води та вплинути на екологію водойм. Тому дослідження та розробка методів вимірювання та моніторингу забруднення води є важливими для захисту навколишнього середовища та забезпечення доступу до чистої води для всіх організмів.

Стан водних ресурсів в Україні є складним через природні та конфліктні фактори, які призводять до забруднення водойм. Поточний конфлікт пошкодив промислову, сільськогосподарську та житлову інфраструктуру, що спричинило значне забруднення, особливо поблизу районів бойових дій, таких як річка Сіверський Донець. Управління якістю води в Україні стикається з системними проблемами, які ускладнюють узгодження з директивами ЄС. Попри зусилля, спрямовані на покращення практик управління водними ресурсами до 2030 року, ефективна реалізація залишається під питанням через поточну геополітичну та економічну ситуацію [2, 3].

Поточний стан водних ресурсів в Україні є складним через вплив конфліктів та системні проблеми в управлінні, що призводить до значного забруднення водойм та ускладнює узгодження з директивами ЄС, попри наявні зусилля з покращення.

У кваліфікаційній роботі «Оцінка впливу твердих побутових відходів на стан поверхневих вод у Львівській області та обґрунтування природоохоронних заходів» [5], що присвячена аналізу впливу твердих побутових відходів на екологічний стан поверхневих вод у Львівській області, основним джерелом забруднення є фільтраційні води з полігонів зберігання відходів, які містять токсичні речовини, такі як сполуки миш'яку, кадмію, хрому, свинцю, ртуті та

нікелю. Ці забруднюючі речовини проникають в ґрунт, потрапляють у підземні водоносні горизонти та врешті-решт у водні об'єкти. Проведений моніторинг виявив, що санітарний стан води не відповідає нормативам через різні фактори, включаючи забруднення ґрунтів і атмосфери, техногенне навантаження та неефективну роботу каналізаційно-очисних споруд.

Важливо розуміти, що управління якістю водних ресурсів є важливим не тільки у побуті, але й постає елементом національної екологічної політики й організовується на різних рівнях влади. На національному рівні існують спеціальні механізми та органи влади, що забезпечують управління якістю води. Ці механізми включають системи моніторингу та аналізу, нормативно-правові акти та закони, що встановлюють стандарти і вимоги до якості води та визначають відповідальність за забруднення водних ресурсів. Державний контроль над водними ресурсами захищає і зберігає водні екосистеми для майбутніх поколінь та забезпечує безпечне і здорове довкілля для суспільства.

Державний моніторинг вод здійснюється з метою: забезпечення збирання, обробки, зберігання, узагальнення й аналізу інформації про стан водних об’єктів, прогнозування його змін та розроблення науково обґрунтованих рекомендацій для прийняття рішень у галузі використання, охорони вод та відтворення водних ресурсів [2].

Управління якістю водних ресурсів включає в себе розміщення станцій моніторингу в стратегічно важливих локаціях, зокрема в місцях впливу промислових та міських відходів на річки, озера та інші водойми.

Дані, зібрані станціями, регулярно передаються в центральні лабораторії для детальнішого аналізу. Лабораторний аналіз дозволяє точно ідентифікувати токсичні елементи та інші забруднювачі на молекулярному рівні, що є ключовим для прийняття рішень щодо заходів у водоохоронній діяльності. Ці дані моніторять якість води, дозволяючи вчасно реагувати на загрози забруднення, такі як нещодавні викиди промислових відходів або неочікуване зростання концентрацій шкідливих речовин.

Моніторинг якості води є надзвичайно важливим через зростання промислової діяльності, яка спричиняє значне забруднення водних ресурсів

шкідливими відходами. В контексті сучасних екологічних викликів, вода як найважливіший ресурс вимагає ретельного нагляду і контролю, що може бути забезпечено за допомогою вдосконалених технологій моніторингу. Це дозволяє не тільки виявляти і відстежувати зміни в якості води, але й реагувати на потенційні загрози для здоров'я людей і стану екосистем. Враховуючи все це, моніторинг води стає не лише технічною необхідністю, але й ключовим елементом національної екологічної політики, направленої на забезпечення сталого використання та охорону водних ресурсів на різних рівнях управління.

1.2 Аналіз аналогічних методів та інформаційних технологій

При розробці нових або вдосконаленні існуючих продуктів важливо не тільки визначити їх функціональні характеристики, але й ретельно проаналізувати технічні аспекти.

На сьогоднішній час на ринку є безліч різних продуктів для відслідковування та аналізу якості води, тому при розробці свого прототипу важливо визначити і ретельно проаналізувати технічні і функціональні аспекти аналогів.

Буї від новозеландської компанії «Adroit» [6] успішно працюють для моніторингу якості води у прісноводних або солоно водних регіонах. В них встановлені сенсори, зонди та інші вимірювальні пристрої. Буї використовують сонячну енергію, що дозволяє проводити тривалий моніторинг даних. Компанія «Adroit» використовує комбінацію буїв та захисних систем для запобігання обростанню або утворенню нальоту на сенсорах і покриттях, що дозволяє досягти полегшення в обслуговуванні підводних пристроїв.

Приклад використання буїв від «Adroit» показано на рисунку 1.1.

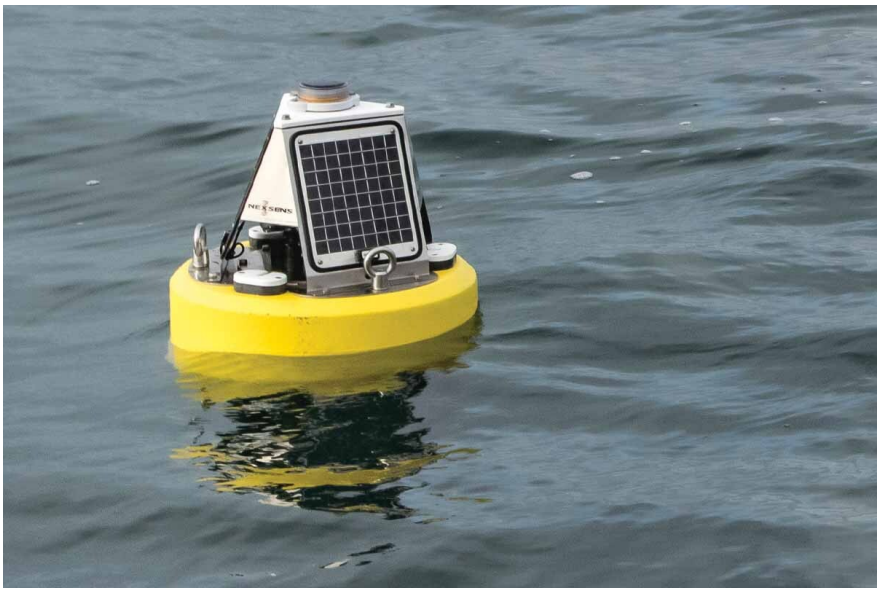

Рисунок 1.1 - Приклад використання буїв від «Adroit»

Також існує аналог від іспанської компанії «Libelium» - Smart Water Xtreme [7] (рис. 1.2), який являє собою невелику плату для управління до якої підключаються потрібні датчики.

Рисунок 1.2 - Libelium Smart Water Extreme

Smart Water розроблено для виявлення найбільш важливих параметрів якості води, таких як розчинений кисень, окисно-відновний потенціал(ОВП), pH, електропровідність і температура. Усі зонди сенсорів мають вбудований температурний сенсор для покращення вимірювань. Цей процес називається «інтегрованою температурною компенсацією». Робочі зонди сенсорів виготовлені з ПВХ та нержавіючої сталі, що дає стійкість до тиску 5-6 бар. Для кожного сенсора доступні різні монтажні аксесуари для розміщення у трубах або баках. Також доступні захисні аксесуари для дикої природи. Сенсори можуть бути розміщені у буях для моніторингу морів і озер, забезпечуючи інформацію в реальному часі. Система забезпечує сумісність з більшістю відповідних бездротових технологій і підтримує протоколи зв'язку LPWAN та стільникового зв'язку для ізольованих місць. Вона сумісна з будь-якою хмарною платформою, забезпечуючи легке встановлення, сонячне живлення та тривалий термін служби батареї.

Фізичні характеристики Smart Water включають мінімальні витрати на технічне обслуговування з корпусами IP65 та сенсорами IP68. Корпуси та сенсори є водонепроникними та витримують до 5 бар (~50 метрів). Крім того, система є легкою у розгортанні, що робить її універсальним і надійним рішенням для моніторингу якості води.

Аналогом також є зонд EXO1 [8] (рис. 1.3) від Yellow Springs Instrument (YSI) компанії. Він має 4 порти для сенсорів, що дає змогу вимірювати тільки потрібні дані.

Рисунок 1.3 - Багато параметричний зонд EXO1

Зонд оснащений сенсорами для вимірювання електропровідності, температури, розчиненого кисню, флуоресценції розчинених органічних речовин, іон-селективними електродами(амонію, хлориду і нітрату), ультрафіолетового нітрату, pH, окисно-відновного потенціалу (ОВП), родаміну (флуоресцентних барвників), загальних водоростей (хлорофіл + фікоціанін і фікоеритрин), а також каламутності.

EXO також може вимірювати абсолютний тиск, аміак, глибину, місцевий відсоток розчиненого кисню, тиск, фотосинтетично-активне випромінювання, опір, загальну кількість розчинених твердих речовин, загальну кількість зважених твердих речовин та щільність води. Сенсори EXO проходять тестування у різних суворих польових і лабораторних умовах, щоб забезпечити високу точність та швидкість відгуку.

Особливості дизайну сенсора включають з'єднувачі, стійкі до корозії, ізольовані компоненти для запобігання коротким замиканням, міцну конструкцію, яка зварена для додаткової міцності та оснащена подвійними ущільнювачами, що забезпечує захист від витоків води чи інших рідин, а також пластиковий корпус високої міцності і титанові деталі.

Пристрій для вимірювання показників води є високоточним, що забезпечує високий рівень довіри до даних, отриманих під час моніторингу якості води.

У таблиці 1.1 наведені основні параметри та відповідні показники точності вимірювань.

Таблиця 1.1 - Сенсор, діапазон і точність для кожного параметра якості води [9]

Параметр	Сенсор	Діапазон	Точність
Температура	Терморезистор	-5-50°C	± 0.02°C
Питома провідність	Терморезистор	0-100,000 μСм/см	± 1%

Розчинений кисень	Оптика, тривалість люмінесценції	0-50 мг/л	0-200%: ±1%; 200-500%: ±5%; 0-20 mg/L: ±0.1 mg/L; 20-50 mg/L: ±0.5 mg/L.
pH	Комбінований електрод із скла	0-14	±0.2 pH з -5 до +50°C
Загальна кількість водоростей (Хлорофіл)	Оптика, флуоресценція	0-400 мкг/л	Не надано

Зонд EXO1 може використовуватись у різних погодних умовах для різних цілей. Він добре підходить для моніторингу водних екосистем, оцінки впливу стічних вод, академічних та наукових досліджень, моніторингу підземних вод, агровиробництва та зрошення.

На рисунку 1.4 показано приклад використання зонду EXO1.

Рисунок 1.4 - Приклад використання зонду EXO1

Також аналогом є система розроблена в дослідженні «Design and Implementation of Water Quality Monitoring System (Temperature, pH, TDS) in Aquaculture Using IoT at Low Cost» на «6th International Conference of Food, Agriculture, and Natural Resource (IC-FANRES 2021)» [10].

Мета цього дослідження є розробка та впровадження системи моніторингу якості воду в аквакультурі, яка буде використовуватися у малих та середніх підприємствах. Автором розроблена недорога IoT система, яка може моніторити параметри аквапонічної системи, такі як температура, pH і електропровідність, через платформу Android. Апаратна частина використовує кілька інтегрованих сенсорів: сенсор температури води типу DS18B20, pH сенсор: SEN0161, сенсор електропровідності: SEN0244. Також використовується WIFI плата NodeMCU ESP8266 для передачі даних на смартфон за допомогою інтернет-з'єднання, що дозволяє моніторити їх за допомогою додатку Blynk.

Зовнішній вигляд пристрою зображений на рисунку 1.5. Також в пристрій вбудований LCD дисплей для відображення даних в режимі реального часу (рис. 1.6).

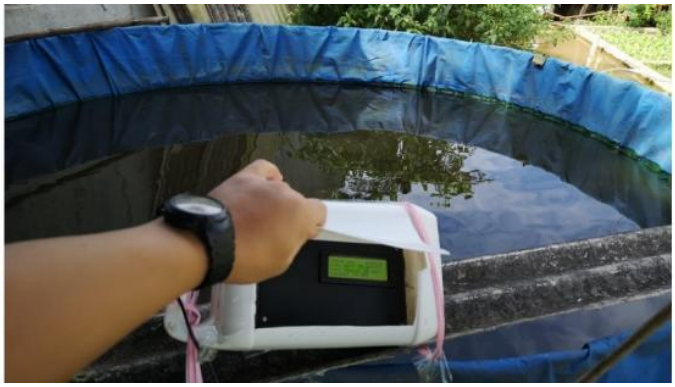

Рисунок 1.5 – Зовнішній вигляд пристрою

Рисунок 1.6 – LCD дисплей пристрою

1.3 Висновки

На основі проведеного аналізу існуючих систем моніторингу води та можливостей для їх покращення, можна зробити висновок, що потрібно приділити увагу розширенню спектру вимірюваних параметрів, зокрема, доцільно включити вимірювання токсичних речовин, органічних сполук і металів. Окрім того, важливою є інтеграція передових сенсорів і розширення апаратної платформи, що включає підтримку додаткових модулів для забезпечення точності.

2 АНАЛІЗ ЕЛЕМЕНТІВ IOT СИСТЕМИ МОНІТОРИНГУ ПОКАЗНИКІВ СТАНУ ПРИРОДНИХ ВОДОЙМ

2.1 Розробка концепцій IoT-системи моніторингу стану природних вод

Для IoT-системи розроблена діаграма (рис. 2.1), що візуалізує класифікацію основних параметрів потрібних для моніторингу стану води.

Рисунок 2.1 – Класифікація параметрів моніторингу води

У роботі за основу взятий пристрій для моніторингу параметрів стану води, таких як ph, температура та каламутність, описаний у статті під назвою «IOT-система вимірювання параметрів стану вод на базі Sigfox» [11], був розроблений науковцями Вінницького національного технічного університету.

Метою цієї системи є моніторинг рівня забруднення та стану водних ресурсів у реальному часі. Пристрій використовує сенсори для вимірювання температури, pH та каламутност. Дані передаються через мережу Sigfox до платформи ThingSpeak, де вони обробляються.

Пристрій включає мікроконтролер Arduino Uno та сенсори DS18B20 (температура), PH DFRobot (pH), та Sen0189 (каламутність). Підключений до джерела живлення, пристрій фіксує показники води та передає їх через мережу

Sigfox до платформи ThingSpeak. Дані обробляються і відображаються у вигляді графіків, що дозволяє здійснювати аналіз і приймати рішення щодо стану водних ресурсів.

Розробка та випробування прототипу показали, що система може успішно використовуватися для моніторингу стану вод у природних умовах, що підтверджує її ефективність і доцільність подальшого вдосконалення, тому було прийняте рішення розробити структурну схему існуючої системи(рис. 2.2).

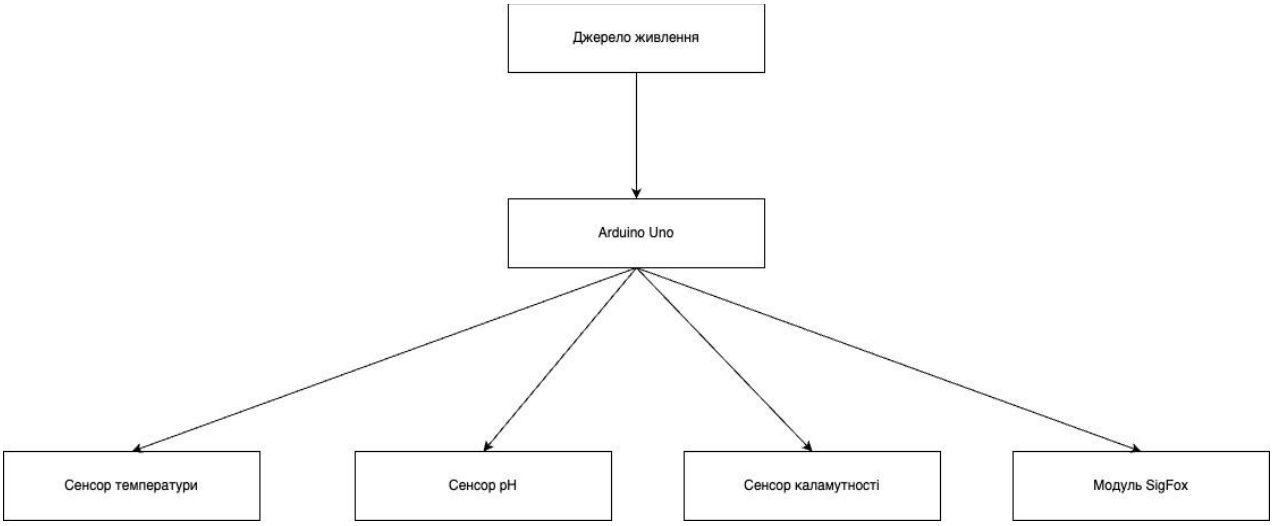

Рисунок 2.2 - Структурна схема пристрою

Алгоритми є основою для розробки програмного забезпечення будь-якої складності, включаючи IoT-системи для моніторингу стану поверхневих вод. Вони визначають послідовність дій, яку має виконувати система для досягнення бажаного результату. У випадку моніторингу поверхневих вод - алгоритми допомагають забезпечити точність та надійність вимірювань, обробки даних та їх передачі.

Після аналізу та опису існуючих аналогічних пристроїв, необхідно розробити детальний алгоритм роботи створюваної IoT-системи.

Алгоритм, представлений на рисунку 2.3, описує основні етапи роботи системи, починаючи з ініціалізації до передачі даних та їх обробки.

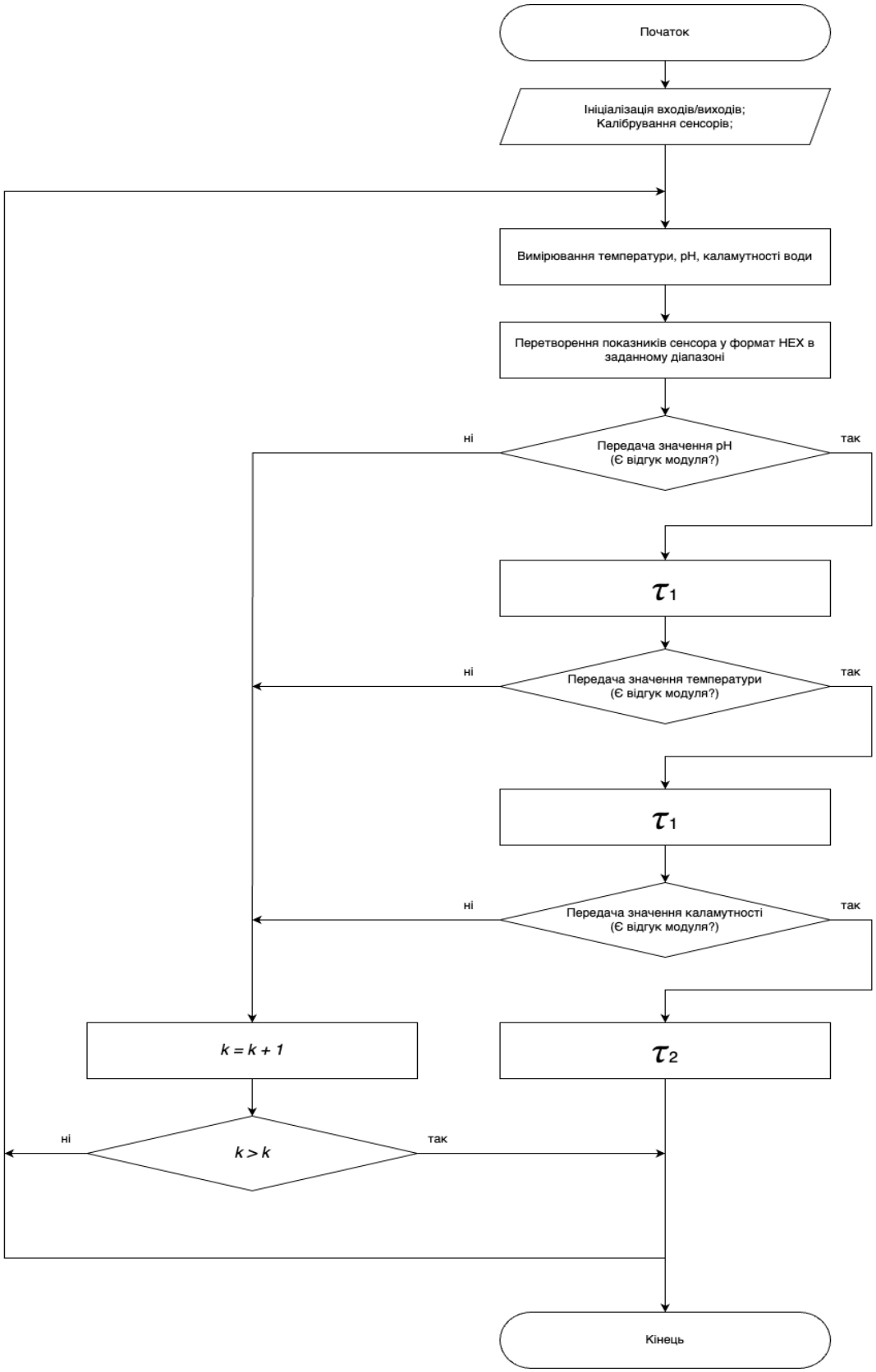

Рисунок 2.3 - Блок-схема алгоритму IoT-системи

Початок алгоритму полягає в ініціалізації процесу. На цьому етапі відбувається налаштування всіх необхідних параметрів системи, включаючи ініціалізацію входів і виходів, а також калібрування сенсорів.

Далі, сенсори здійснюють вимірювання трьох ключових параметрів стану води: температури, рівня pH та каламутності. Ці параметри є критично важливими для оцінки якості води. Після зчитування значень, отримані з сенсорів дані перетворюються у формат HEX. Це необхідно для забезпечення коректної передачі та обробки даних у подальших етапах роботи системи.

Після перетворення даних алгоритм перевіряє наявність зворотного зв'язку від модуля після передачі кожного параметра. Якщо відгук від модуля отриманий, значення pH передається далі, якщо ні – система переходить до повторного вимірювання. Для стабілізації процесу передачі даних передбачені затримки, позначені як $\tau 1$ і $\tau 2$.

Після передачі значень pH, температури та каламутності система збільшує лічильник ітерацій на одиницю. Якщо лічильник досягає заданого значення, алгоритм завершує роботу. В іншому випадку, процес повторюється знову, починаючи з вимірювання параметрів води.

В кінці алгоритму, пристрій повертається в свій стан перед вимірюванням температури, pH та каламутності що дозволяє працювати пристрою безперервно і весь час отримувати нові дані, які відправляють сенсори.

Розроблений алгоритм є основою для створення надійної та ефективної IoT-системи для моніторингу стану поверхневих вод. Він забезпечує безперервний збір та обробку даних, що дозволяє своєчасно реагувати на будь-які зміни у якості води. Такий підхід дозволяє підвищити ефективність екологічного моніторингу.

2.2 Аналіз технічних характеристик сенсорів та модулів для побудови IoT-системи моніторингу стану природних вод

Мозком цієї системи є Arduino Uno (рис. 2.4). Він дає змогу керувати всіма підключеними сенсорами та модулями, обробляти сигнали з цих сенсорів і передавати зібрані дані на інші модулі або комп'ютери.

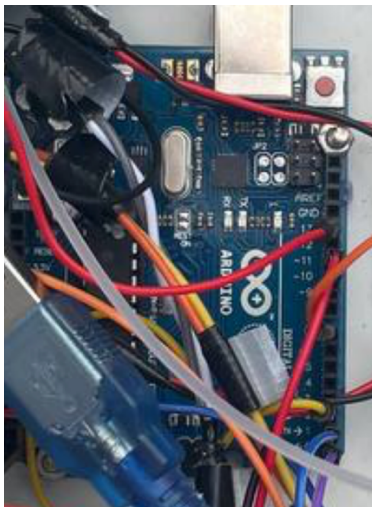

Рисунок 2.4 – Мікропроцесорна плата Arduino Uno

Arduino Uno добре підходить для цього завдання через велику кількість цифрових і аналогових входів і виходів, які дозволяють підключати всі необхідні сернсори. Мікроконтролер ATmega328p, що використовується в Arduino Uno, забезпечує достатню обчислювальну потужність для обробки сигналів у режимі реального часу.

Плата для мікропроцесора ATmega328P має керамічний резонатор на 16 МГц (CSTCE16M0V53-R0), 6 аналогових входів, 14 цифрових входів/виходів (з яких 6 можна використовувати як ШИМ-виходи), порт USB, роз'єм живлення, заголовок ICSP і кнопку скидання. Він постачається з усім необхідним для підтримки мікроконтролера; щоб використовувати його, просто підключити USB кабель, адаптер змінного/постійного струму або батарею для живлення [12].

Крім того, Arduino Uno відрізняється простотою використання і наявністю великої кількості бібліотек і прикладів коду, що значно спрощує процес розробки навіть для новачків. Він підтримує програмування через USB, що зручно для налагодження та оновлення програмного забезпечення. Завдяки своїй популярності, Arduino Uno має велику спільноту розробників, готових поділитися своїм досвідом і вирішити різні технічні проблеми.

Іншим компонентом даної системи є цифровий сенсор температури DS18B20 (рис. 2.5). Він дуже добре підходить для цієї задачі завдяки своїй здатності використовувати 1-Wire шину для комунікації, що дозволяє підключати кілька сенсорів до одного порту мікроконтролера. Це значно

полегшує встановлення та використання сенсорів у системах з багатьма сенсорами.

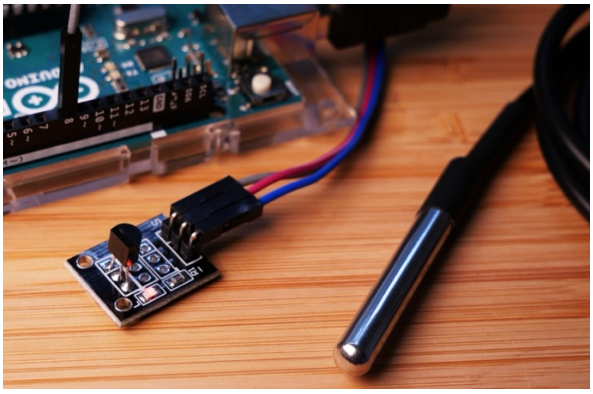

Рисунок 2.5 - Цифровий сенсор температури DS18B20

Сенсор може забезпечувати 9-, 10-, 11- та 12-бітову роздільну здатність для вимірювання температури в градусах Цельсія, що відповідає прирощенням 0,5 °C, 0,25 °C, 0,125 °C та 0,0625 °C відповідно. Цю роздільну здатність можна вибрати, записуючи відповідні дані в конфігураційний регістр сенсора (за замовчуванням встановлено 12-бітову роздільну здатність). На одиницю коду перетвореного значення температури доводиться 0,0625 °C. Абсолютна похибка перетворення менше 0,5 °C в діапазоні контрольованих температур від -10 °C до +85 °C [13].

Цей сенсор має невеликий розмір і високу точність, що робить його ідеальним вибором для різних проектів контролю температури. DS18B20 може працювати на великих відстанях до 100 метрів, що дозволяє встановлювати його у важкодоступних місцях при збереженні високої якості вимірювань.

Окрім того він легко інтегрується з Arduino завдяки наявності готових бібліотек, що значно полегшує процес програмування та налагодження. Цей сенсор також підтримує різні режими живлення, що дозволяє оптимізувати енергоспоживання системи.

PH DFRobot - це спеціалізований сенсор для вимірювання рівня pH рідин, який використовується в різних середовищах, таких як: акваріуми, гідропоніка, системи водопостачання та у моніторингу екології.

Цей сенсор потрібен для точного визначення кислотності або лужності води. Він складається з pH-зонда та електронного модуля, який перетворює аналоговий сигнал з зонда в цифровий формат, що легко читається мікроконтролером.

PH DFRobot дуже добре підходить для задач моніторингу якості води завдяки своїй точності та стабільності вимірювань. Він забезпечує надійні результати навіть при тривалому використанні. Цей сенсор легко інтегрується з Arduino, що робить процес його підключення та програмування простим і зручним. Існують готові бібліотеки для роботи з цим сенсором, що дозволяє швидко почати збирати дані про pH води DF Robot пропонує сенсор pH, який використовується для вимірювання рівня pH розчину. pH є мірою кислотності або лужності розчину і є важливим параметром у багатьох застосуваннях, включаючи моніторинг якості води, акваріуми, гідропоніку(вирощення рослин без ґрунту) та багато іншого.

Сенсор pH від компанії DFRobot (рис. 2.6) зазвичай використовує електрод для вимірювання рівня pH розчину. Електрод генерує напругу, пропорційну рівню pH, яка потім може бути зчитана мікроконтролером, таким як Arduino [14].

Рисунок 2.6 - Сенсор pH від компанії DFRobot

Основні характеристики сенсора [15]:

– клас сенсора: Лабораторний;
– діапазон виявлення: 0~14;
– діапазон температур: 5~60°C
– нульова точка: 7±0.5;
– час відгуку: <2 хвилин;
– внутрішній опір: <250MΩ.

Сенсор є дуже надійним та довговічним, що робить його ідеальним вибором для постійного моніторингу стану води. Він забезпечує стабільні та точні результати, що дозволяє ефективно контролювати якість водних ресурсів і вчасно реагувати на зміни в екологічному стані води.

Одним із важливих сенсорів - є сенсор SEN0189 (рис. 2.7). Це високоефективний інструмент для вимірювання каламутності води, що є основним параметром для оцінки якості водних ресурсів. Цей сенсор використовує оптичний метод визначення каламутності, базуючись на розсіюванні світла інфрачервоними діодами, що дозволяє точно фіксувати наявність та концентрацію твердих частинок у воді.

Рисунок 2.7 – Модуль каламутності SEN0189

SEN0189 має вбудований оптичний сенсор, який аналізує рівень розсіяного світла, коли воно проникає крізь воду. Збільшення каламутності води призводить до інтенсивного розсіювання світла завдяки частинкам, що знаходяться у воді. Сформований таким чином аналоговий сигнал трансформується в цифровий формат, що дозволяє його легко зчитувати та аналізувати за допомогою мікроконтролера.

У журналі Journal of Physics: Conference Series статті «Characterization of turbidity water sensor SEN0189 on the changes of total suspended solids in the water» проводилось дослідження для характеристики здатності датчика виявляти каламутність води. Метод полягав у тестуванні датчика за допомогою осадових ґрунтів, відфільтрованих до діаметра <60 мкм, доданих у басейн з 1 літром води. Результати показали, що зі збільшенням концентрації осаду у воді вихідна напруга датчика зменшувалася. Датчик має чутливість -0.0008, вихідну напругу 3.9994 вольт при 0 мг/л, робочу напругу 5V і може лінійно виявляти каламутність води у діапазоні від 1.873 мг/л до 1011.93 мг/л [16].

У цю систему також інтегровано модуль Sigfox (рис. 2.8), який забезпечує надійну і ефективну передачу даних з сенсорів на великі відстані. Цей модуль використовує малопотужний зв’язок для передачі невеликих обсягів даних через мережу Sigfox, що дозволяє мінімізувати витрати енергії та продовжити термін служби батарей у віддалених пристроях моніторингу.

Рисунок 2.8 – Модуль Sigfox та антена

Протокол SigFox дуже простий. Він не вимагає рукостискання (сигналізація для встановлення зв’язку, тобто процедура представлення або взаємної ідентифікації партнерів по зв’язку при встановленні зв’язку) і передає пакети лише 12 байт (плюс додаткові дані, такі як ідентифікатор радіо та час). Як уже згадувалося, передача здійснюється в дуже вузькій смузі частот, використовуючи модуляцію D-BPSK (Differential Binary Phase Shift Keying) із швидкістю 100 або 600 біт / с. Так, саме – 100 біт / с при шести секундних циклах передачі. Однак цей низькошвидкісний та вузько частотний спектр економить заряд акумулятора та забезпечує широкий діапазон покриття [17].

Sigfox приймає низку команд для управління пристроями, передавання даних і налаштування:

– AT$SS: Відправка повідомлення через мережу Sigfox. Формат команди включає корисне навантаження повідомлення, і команда ініціює передавання;

- AT$SF: Схожа на AT$SS, ця команда відправляє кадр через мережу Sigfox. Вона включає параметри для конфігурації зворотного зв'язку, що дозволяє пристрою отримувати дані назад із мережі;
- AT$RC: Перевірка радіоконфігурації пристрою. Використовується для перевірки регіональних налаштувань та забезпечення відповідності місцевим регламентам;
- AT$CB: Очищення буфера очікуваних повідомлень. Це корисно для керування потоком даних і забезпечення, що старі або непотрібні повідомлення не передаються;
- AT$P: Переведення пристрою в режим сну для збереження енергії. Ця команда є важливою для пристроїв, що працюють від батарей, і потребують максимального збільшення терміну служби;
- AT$I=10: Отримання ідентифікатора пристрою, який є унікальним для кожного модуля Sigfox. Цей ідентифікатор використовується для ідентифікації та реєстрації;
- AT$I=11: Отримання PAC (Porting Authorization Code) пристрою, який використовується для реєстрації пристрою на платформі Sigfox;
- AT$IF: Перевірка версії прошивки пристрою. Це корисно для обслуговування і забезпечення актуальності пристрою з останніми покращеннями програмного забезпечення та виправленнями.

Додатково, інтеграція модуля Sigfox дозволяє системі використовувати різні режими передачі даних для оптимізації споживання та забезпечення тривалої роботи без підзарядки.

Таким чином, інтеграція модуля Sigfox у систему IoT для моніторингу стану поверхневих вод відкриває нові можливості для покращення ефективності моніторингу та управління даними, забезпечуючи стабільність та надійність передачі інформації у критичних для екології проектах.

У рамках розробки системи моніторингу якості води, представлений макет (рис. 2.9), розроблений у програмі Proteus, яка використовується для створення

схематичних зображень та симуляції поведінки електронних схем перед фізичним виготовленням у який входять описані пристрої та сенсори.

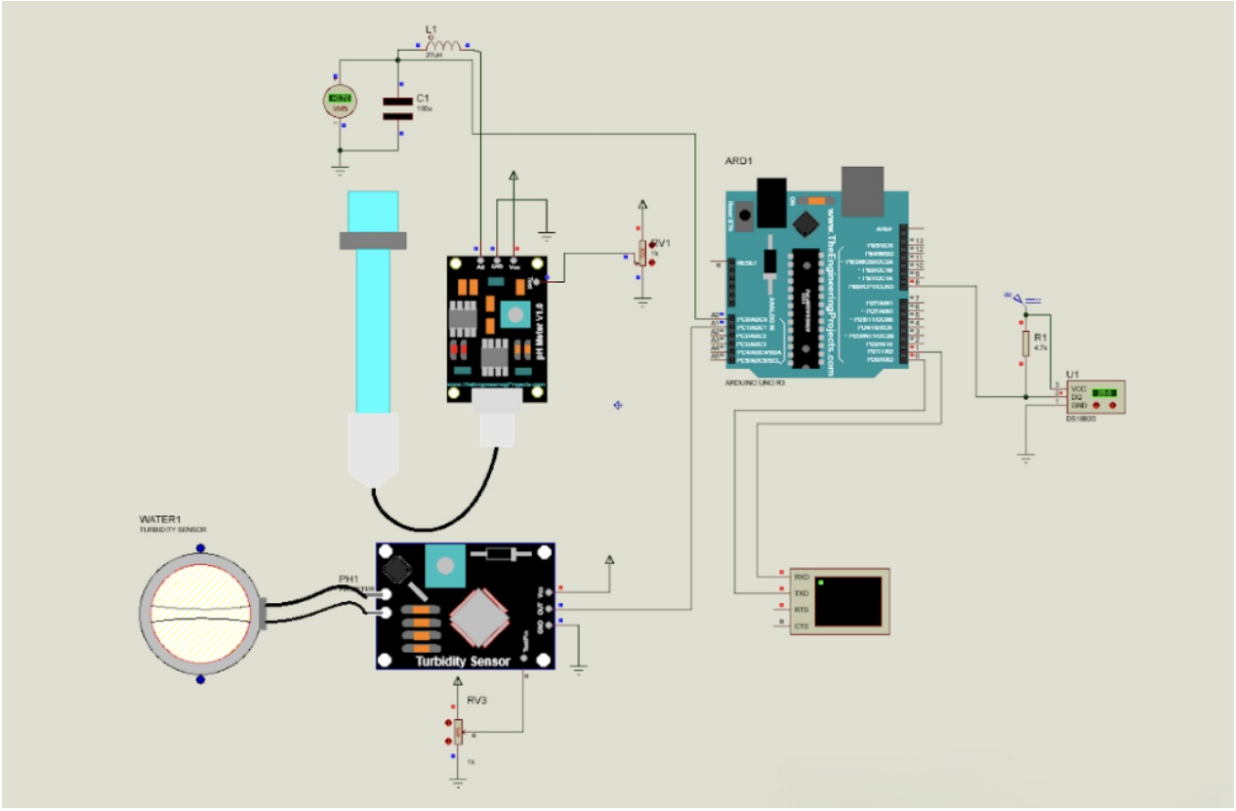

Рисунок 2.9 - Схематичне зображення пристроїв

2.3 Компоновка апаратного забезпечення

У ході бакалаврської роботи розроблений пристрій, що представлений на рисунку 2.10. У ході його виготовлення були використані різноманітні інструменти та матеріали.

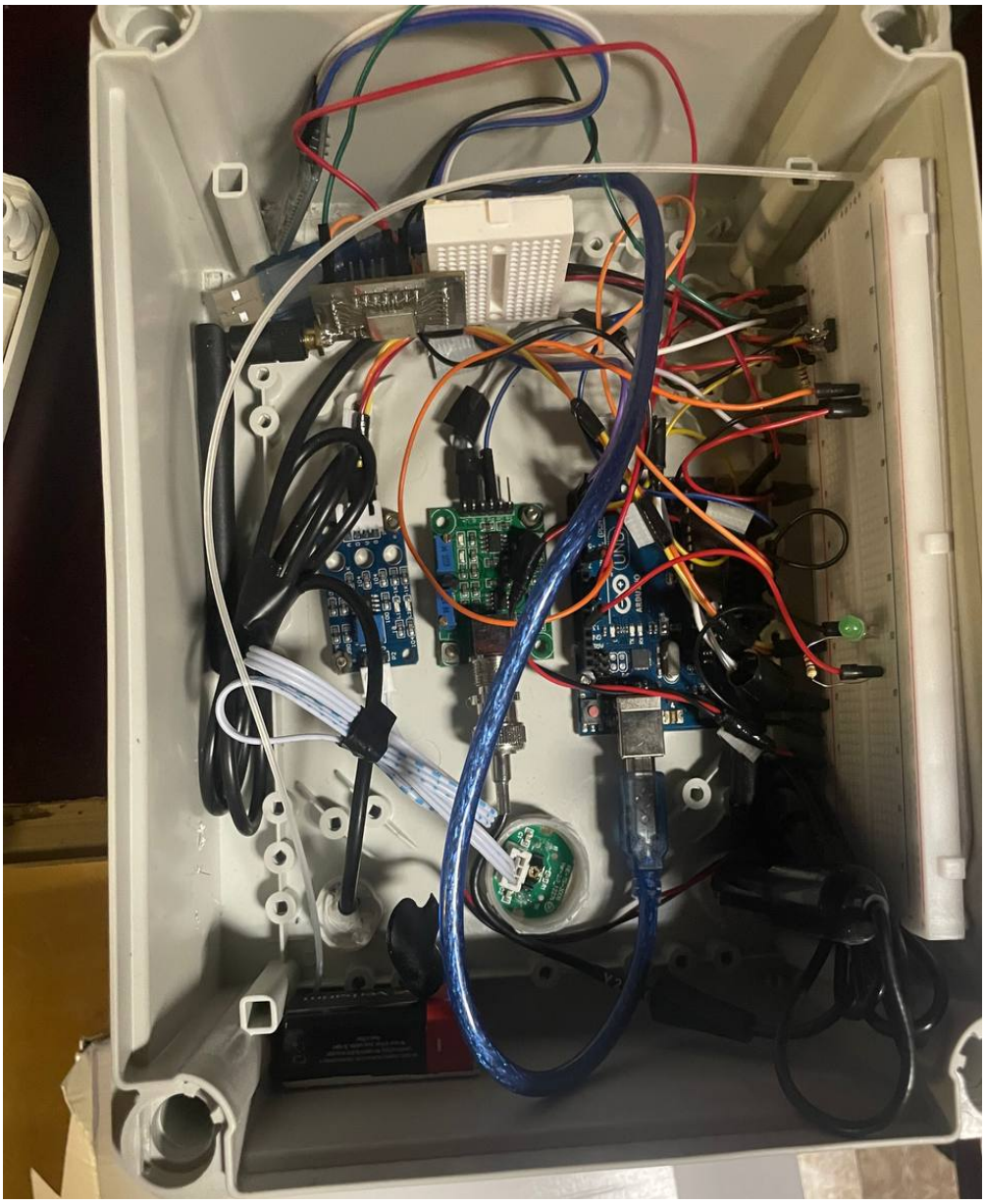

Рисунок 2.10 - Зібраний пристрій

Зокрема, паяльник застосовувався для з'єднання електронних компонентів із платами та сенсорами.

Клейовий пістолет використовувався для фіксації компонентів всередині корпусу, щоб забезпечити додаткову міцність з'єднань та уникнути їх випадкового відривання під час експлуатації пристрою на водних середовищах.

Для ефективного закріплення деталей до стінок корпусу використовується двосторонній скотч. Він дозволяє швидко та легко закріпити потрібні компоненти системи без необхідності використання додаткових інструментів, таких як шурупи, для яких потрібно також мати викрутку або шурупокрут.

Використання скотчу також ізолює електричний прибор від стінки, що допомагає уникнути пошкоджень чутливих компонентів від тряски чи вібрації.

Всередині корпусу було встановлено кілька монтажних платформ для забезпечення надійного кріплення електронних компонентів, що мінімізує ризик пошкодження під час транспортування чи експлуатації.

2.4 Характеристика розробленої програми та вибір її оптимальних налаштувань

Для управління сенсорами та загального контролю системи моніторингу якості води на базі Arduino було розроблено програмний код використовуючи мову програмування C++, що ідеально підходить для взаємодії з апаратним забезпеченням мікроконтролерів. Завдяки бібліотекам він дозволяє ефективно управляти сенсорами пристрою.

Arduino використовує апаратне забезпечення, відоме як розробницька плата Arduino. Програмне забезпечення для розробки коду відоме як Arduino IDE (інтегроване середовище розробки). Вона побудована на основі 8-бітних мікроконтролерів Atmel AVR, які виробляються Atmel, або 32-бітних мікроконтролерів Atmel ARM, які можна легко програмувати використовуючи мови C або C++ в Arduino IDE.

Плата Arduino також може використовуватися для завантаження нового коду на плату Arduino за допомогою USB-кабелю. Arduino IDE забезпечує спрощену інтегровану платформу, яка може працювати майже на всіх персональних комп'ютерах, і користувачі можуть писати програми для Arduino, використовуючи мови програмування C або C++ [18].

IDE спрощує всі етапи розробки від написання до відлагодження коду. Середовище забезпечує користувачів усіма необхідними інструментами, включаючи редактор коду з підсвічуванням синтаксису та вікно для виводу повідомлень про помилки, а також функціонал для прямої компіляції та завантаження програм на підключені до комп'ютера плати Arduino.

У процесі роботи з Arduino IDE, розробники спочатку створюють програмний код, використовуючи вбудовані функції та бібліотеки. Після написання коду вони компілюють його в машинний код, зрозумілий для мікроконтролера, за допомогою компілятора, інтегрованого в IDE. Завершальним етапом є завантаження програми на плату через USB-з'єднання, після чого розробники можуть спостерігати за її виконанням і, при необхідності, проводити відлагодження. Цей підхід значно спрощує управління всіма аспектами роботи системи моніторингу, підвищуючи стабільність, точність вимірювань та загальну надійність системи.

Arduino IDE доступне у двох форматах: як десктопна програма, яка завантажується на персональний комп'ютер так і онлайн веб-версія, яка працює у браузері. Кожна з цих версій пропонує унікальні інструменти та можливості для розробників, відповідаючи на різноманітні потреби у сфері програмування мікроконтролерів.

Десктопна версія Arduino IDE встановлюється безпосередньо на комп'ютер. Це забезпечує розробникам повний контроль над середовищем програмування та дозволяє працювати офлайн, що є надзвичайно зручно у випадках обмеженого або нестабільного інтернет-з'єднання. Програма включає розширені налаштування, що можна адаптувати під конкретні проекти, а також вбудований менеджер бібліотек для легкого управління та встановлення додаткових бібліотек. Важливим інструментом є також серійний монітор, який дозволяє спостерігати за даними, що передаються між Arduino та ПК, що надзвичайно корисно для відлагодження програм.

Десктопна версія програми Arduino IDE зображена на рисунку 2.11.

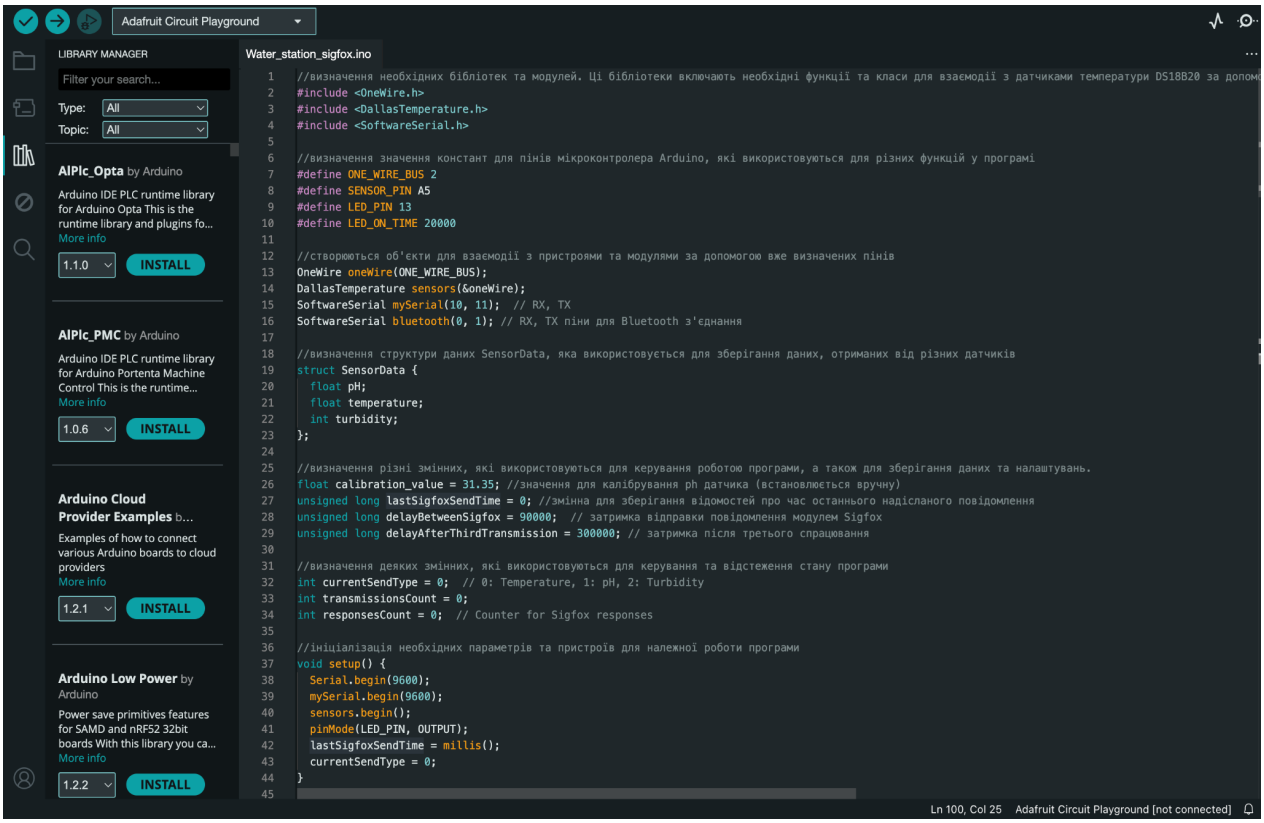

Рисунок 2.11 - Десктопна версія програми Arduino IDE

Онлайн Веб-версія Arduino IDE, з іншого боку, забезпечує глобальний доступ до проектів з будь-якого пристрою, який має підключення до інтернету. Всі проекти зберігаються в хмарі, що забезпечує їх безпеку і доступність. Онлайн-версія також спрощує співпрацю, дозволяючи користувачам легко ділитися кодом і спільно працювати над проектами. Оновлення програмного забезпечення відбуваються автоматично, гарантуючи, що користувачі завжди мають доступ до останніх функцій та виправлень безпеки. Ще однією перевагою є сумісність з різними операційними системами, що усуває потребу у встановленні додаткового програмного забезпечення.

Онлайн Веб-версія програми Arduino IDE зображена на рисунку 2.12.

Рисунок 2.12 - Онлайн Веб-версія Arduino IDE

2.5 Програмний код мікропроцесорного пристрою

У блок-схемі алгоритму, з початку здійснюється процедура ініціалізації, яка передбачає необхідні налаштування входів/виходів мікроконтролера параметрів та пристроїв для належної роботи програми, що реалізується у вигляді методу «setup» (рис. 2.13). Цей метод виконує критичні налаштування, які гарантують, що система здатна збирати та обробляти дані від сенсорів.

```
void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  sensors.begin();
  pinMode(LED_PIN, OUTPUT);
  lastSigfoxSendTime = millis();
  currentSendType = 0;
}
```

Рисунок 2.13 - Метод «setup» для ініціалізації основних параметрів

Метод «measurePH» (рис. 2.14) зчитує дані, після чого сортує їх, обраховує середнє значення, після чого розраховує pH.

```
void measurePH(SensorData& data) {
  int buffer_arr[10], temp;
  unsigned long int avgval;

  for (int i = 0; i < 10; i++) {
    buffer_arr[i] = analogRead(A0);
    delay(30);
  }

  for (int i = 0; i < 9; i++) {
    for (int j = i + 1; j < 10; j++) {
      if (buffer_arr[i] > buffer_arr[j]) {
        temp = buffer_arr[i];
        buffer_arr[i] = buffer_arr[j];
        buffer_arr[j] = temp;
      }
    }
  }

  avgval = 0;
  for (int i = 2; i < 8; i++) avgval += buffer_arr[i];

  float volt = (float)avgval * 5.0 / 1024 / 6;
  float pH_act = -5.70 * volt + calibration_value;
  pH_act = 400 + (round(pH_act * 10.0) / 10.0) * 10;

  data.pH = pH_act;
}
```

Рисунок 2.14 - Метод «measurePH»

Метод «measureTemperature» (рис. 2.15) робить запит температури, після чого отримане значення температури округлює для точності і зберігає ці дані в структурі.

```
void measureTurbidity(SensorData& data) {
  int sensorValue = analogRead(SENSOR_PIN);
  int turbidity = max(0, map(sensorValue, 0, 640, 100, 0));
  int turbidity_s = 600 + turbidity;

  data.turbidity = turbidity_s;
}
```

Рисунок 2.15 - Метод «measureTemperature»

Метод «measureTurbidity» (рис. 2.16) зчитує значення з аналогового входу, призначеного для сенсора каламутності, після чого за допомогою функції «map» змінює вхідні значення в заданий діапазон, після чого результат коригується для отримання кінцевого показника.

```
void measureTurbidity(SensorData& data) {
  int sensorValue = analogRead(SENSOR_PIN);
  int turbidity = max(0, map(sensorValue, 0, 640, 100, 0));
  int turbidity_s = 600 + turbidity;

  data.turbidity = turbidity_s;
}
```

Рисунок 2.16 - Метод «measureTurbidity»

Після чого отриманні дані відправляються через модуль SigFox у відповідних минуло-оброблених форматах. Програма також включає в себе логіку для управління затримками, включаючи спеціальні умови після кожної третьою відправки для додаткової затримки.

Весь код знаходиться в основному вічному циклі, що дає можливість не перезавантажувати Arduino кожен раз після отримання даних.

2.6 Висновки

У рамках бакалаврської роботи проведено аналіз компонентів IoT-системи моніторингу стану природних водойм. Було розроблено діаграму, яка візуалізує класифікацію основних параметрів моніторингу води, а також створено структурну схему пристрою та блок-схему алгоритму. Проведено детальний аналіз технічних характеристик сенсорів і модулів, розроблено макет із схемами пристроїв, розглянуто компонування апаратного забезпечення. Також проаналізовано характеристики розробленої програми для Arduino та середовище розробки IDE.

3 РОЗРОБКА ТЕХНОЛОГІЇ ЗБОРУ ДАНИХ ТА ЇХ ПЕРВИННОЇ ОБРОБКИ З ВИКОРИСТАННЯМ ХМАРНИХ СЕРВІСІВ. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

3.1 Аналіз хмарного сервісу для обробки даних ThingSpeak

Результати експериментальних досліджень розробленої системи моніторингу якості води збираються та аналізуються за допомогою веб-сервісу ThingSpeak. Він є хмарним API Інтернет речей (IoT) платформою, яка дозволяє збирати, зберігати, аналізувати та візуалізувати дані в реальному часі. Ця платформа використовується для підтримки зв'язку між сенсорами та додатками для керування даними через інтернет.

Отже ThingSpeak надає функціональні можливості для обробки і аналізу даних за допомогою вбудованих функцій, таких як математичні операції та обробка даних у реальному часі. Користувачі можуть налаштувати математичні вирази для обчислення статистичних показників або для генерації нових датасетів на основі отриманих даних.

Також ThingSpeak дозволяє візуалізувати дані через діаграми та графіки, що забезпечує користувачам зрозумілий візуальний засіб перегляду та аналізу змін у параметрах води. Користувачі можуть налаштувати діаграми для відображення трендів у реальному часі, що є особливо корисним для спостереження за динамікою змін водних ресурсів.

Коли вимірювальні параметри виходять за порогові значення, ThingSpeak можна налаштувати так, щоб він автоматично відправляє сповіщення. Завдяки цьому користувач може швидко відреагувати на різкі зміни або потенційні проблеми в якості води.

Таким чином, ThingSpeak діє як місток між зібраними сенсорами даними і кінцевими користувачами, надаючи засоби для їх збору, обробки, аналізу, візуалізації та моніторингу в одній інтегрованій платформі. Це забезпечує глибокий і доступний аналіз стану водних ресурсів, що є критично важливим для екологічних досліджень та управління.

ThingSpeak - це хмарна платформа з відкритим вихідним кодом, яка підключає сенсори або інші пристрої до системи через інтернет-з'єднання з підтримкою HTTP і надсилає дані з локальної мережі в хмару. Вона відповідає за оновлення кожного журналу даних, які користувачі можуть надсилати та отримувати з сенсорів, програм для визначення стану та відстеження місцезнаходження. Щоб скористатися цією функцією, користувач повинен спочатку створити обліковий запис з кількома каналами для моніторингу різних параметрів системи або віддаленого пристрою. Користувачі та адміністратори можуть переглядати дані у вигляді графіків у цій хмарі. Дані про споживання енергії передаються на маршрутизатор і стають загальнодоступними в Інтернеті завдяки моніторингу через Інтернет [19].

Сам сайт ThingSpeak (рис. 3.1) має легкий та інтуїтивний інтерфейс, у якому легко розібратись з навігацією й управлінням своїми проектами навіть не досвідченому користувачеві.

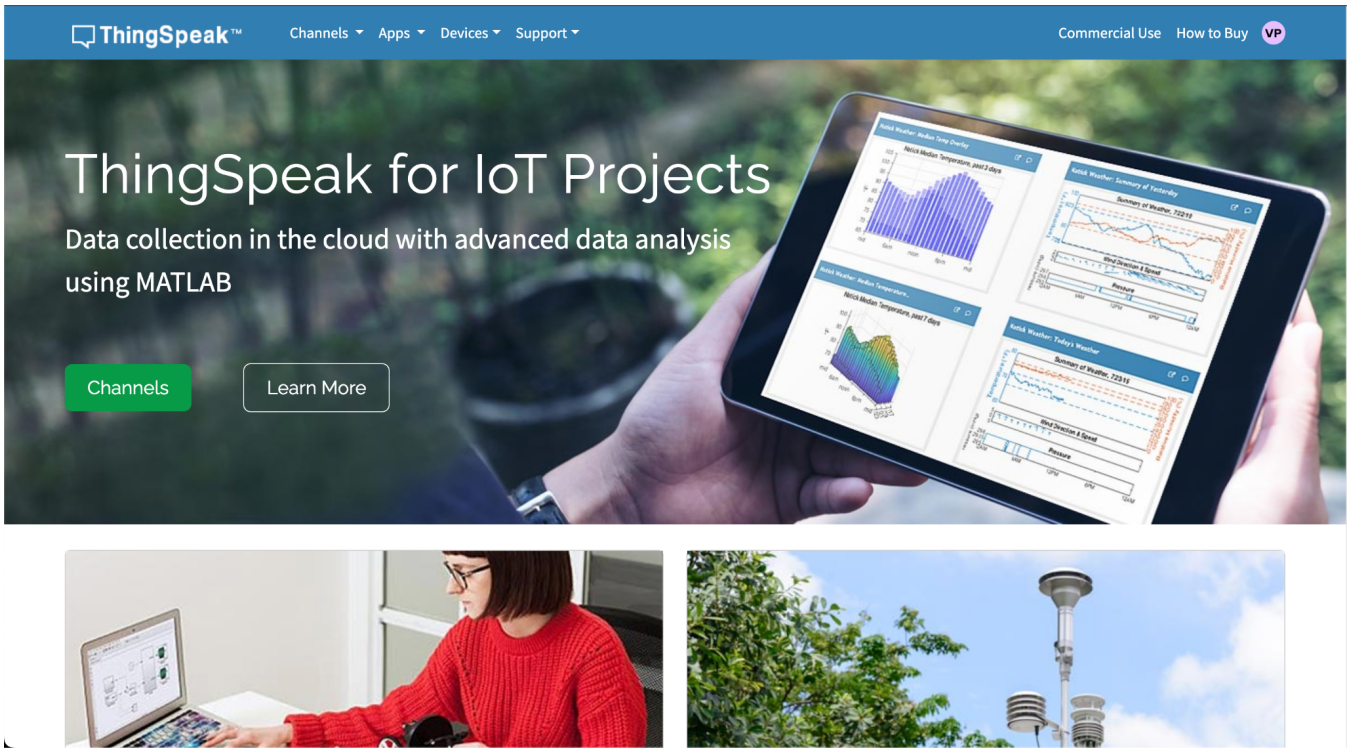

Рисунок 3.1 - Офіційний сайт ThingSpeak

3.2 Експериментальне дослідження IoT-системи моніторингу природніх вод. Експорт і обробка даних з ThingSpeak

Дослідження води у дипломній роботі, було проведено у мікрорайоні Вишенька, який розташований у місті Вінниця. Об'єктом дослідження стало озеро, прилегле до Вінницького національного технічного університету (ВНТУ). Його місцезнаходження зображено на рисунку 3.2. Це місце було обрано через його зручність у тестуванні прилада через його відсутність течії та його доступність, що дає змогу легко використовувати для розміщення приладів.

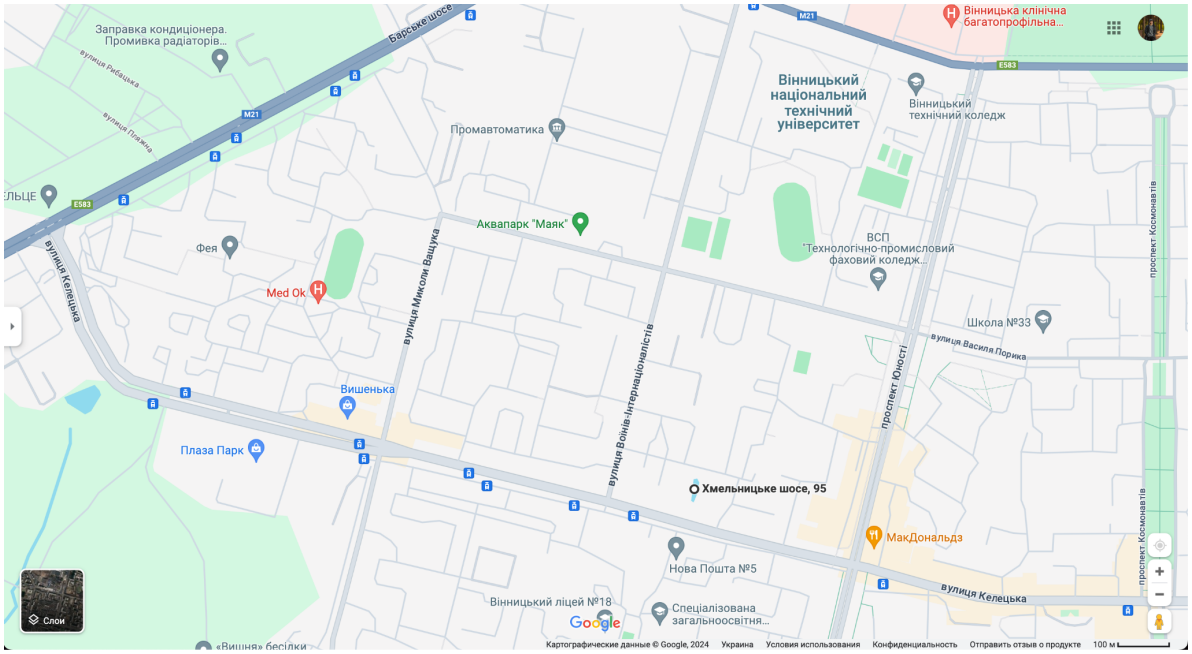

Рисунок 3.2 - Місце де проводилось дослідження

Після успішного збору даних(рис. 3.3, 3.4) з системи моніторингу якості води, що використовує платформу ThingSpeak для агрегації та візуалізації даних, наступним кроком є їх експорт (рис. 3.5) для подальшого аналізу.

Рисунок 3.3 – Проведення моніторингу води на території озера прилеглого до ВНТУ (Зліва направо: **Конотоп Б. П**, Гаврилов Г. А., Панасюк В. Р., Проценко Д. П.)

Рисунок 3.4 - Система моніторингу якості води у роботі

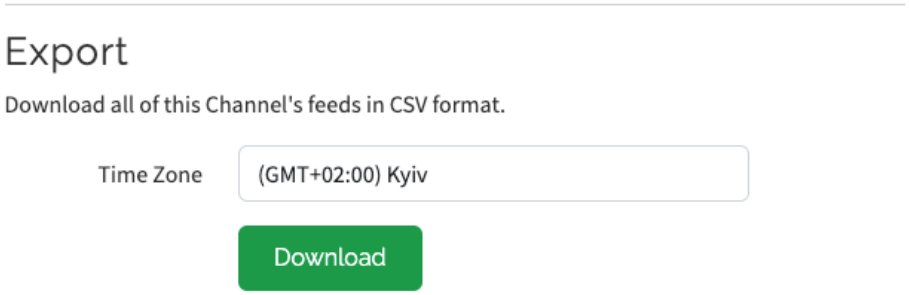

Рисунок 3.5 - Експорт даних в CSV

Отримані дані через ThingSpeak, ми плануємо експортувати у форматі CSV і отримуємо таку таблицю як на рисунку 3.6.

created_at	entry_id	field1
2024-05-22T05:27:58+00:00	1	00cd
2024-05-22T16:28:51+00:00	2	00b9
2024-05-22T16:29:45+00:00	3	00aa
2024-05-22T16:31:16+00:00	4	1E+02
2024-05-22T16:32:47+00:00	5	258
2024-05-22T16:38:16+00:00	6	00bb
2024-05-22T16:39:48+00:00	7	1E+01
2024-05-22T16:41:19+00:00	8	258
2024-05-22T16:46:50+00:00	9	00ba
2024-05-22T16:48:19+00:00	10	01ca
2024-05-22T16:49:51+00:00	11	258

Рисунок 3.6 - Отриманні дані у форматі CSV

Після експорту даних з ThingSpeak у форматі CSV, наступним кроком є перетворення даних з шістнадцяткового (hex) формату в десятковий. Для цих цілей був використаний онлайн Веб-сервіс «rapidtables» [20] (рис. 3.7).

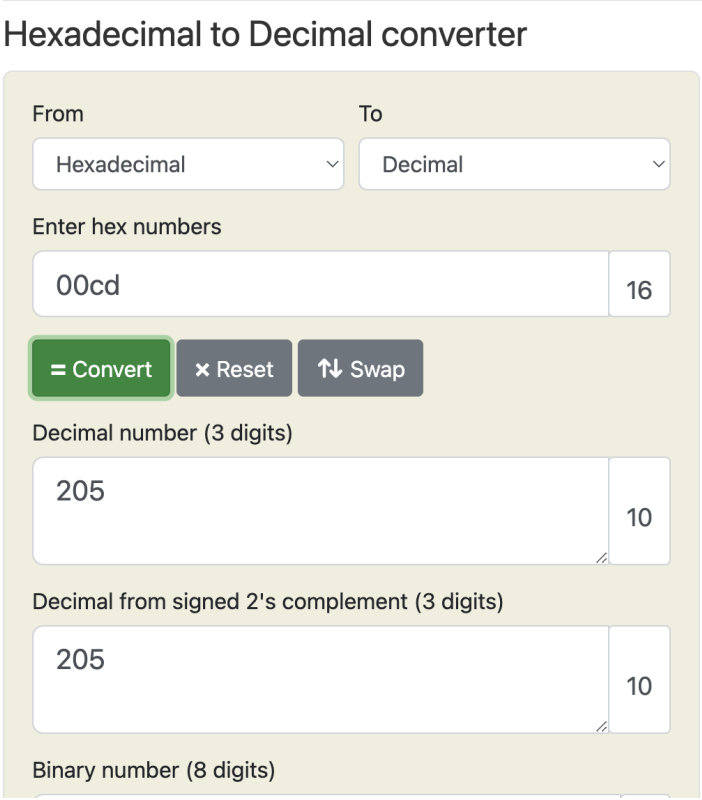

Рисунок 3.7 - Rapidtables приклад переводу з формату HEX

На рисунку 3.7 ми отримали значення після процесу конвертації шістнадцяткових значень у десяткові. Дивлячись на розроблену схему обрахування (рис. 3.8), було отримано значення температури, яке після переводу получилось 205, що є температурою.

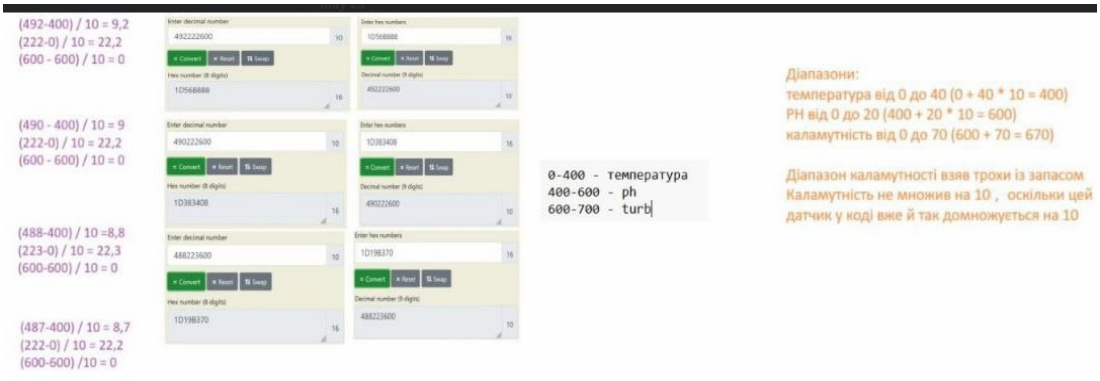

Рисунок 3.8 - Процес обробки даних після конвертації

Далі йде процес конвертації, у даному випадку температури, що вираховується з урахуванням мінімального значення діапазону та

максимального обмеження. (Hex−0)/10. Після обробки наша температура виходить 20.5 °C.

Іншим значенням було отримано HEX 01DF, що після конвертації стало 479. Це означає, що це число було отримано з сенсору pH і після обробки формулою отримуємо 7.9 pH.

ThingSpeak пропонує вбудований MATLAB Analysis і MATLAB Visualizations для аналізу та візуалізації даних, що зібрані в каналі. Щоб використовувати ці інструменти, спершу потрібно створити новий MATLAB Analysis або Visualization скрипт з вкладки 'Apps' у ThingSpeak. Ці скрипти дозволяють виконувати обчислення, створювати графіки, обробляти дані та багато іншого.

Використовуючи MATLAB код у ThingSpeak (рис. 3.9), можна трансформувати зібрані дані для подальшого аналізу. Наприклад, можна очистити дані від викидів, згладити часові ряди для кращого виявлення трендів або виконати статистичний аналіз. Можна також реалізувати більш складні математичні моделі та алгоритми для передбачення або класифікації станів.

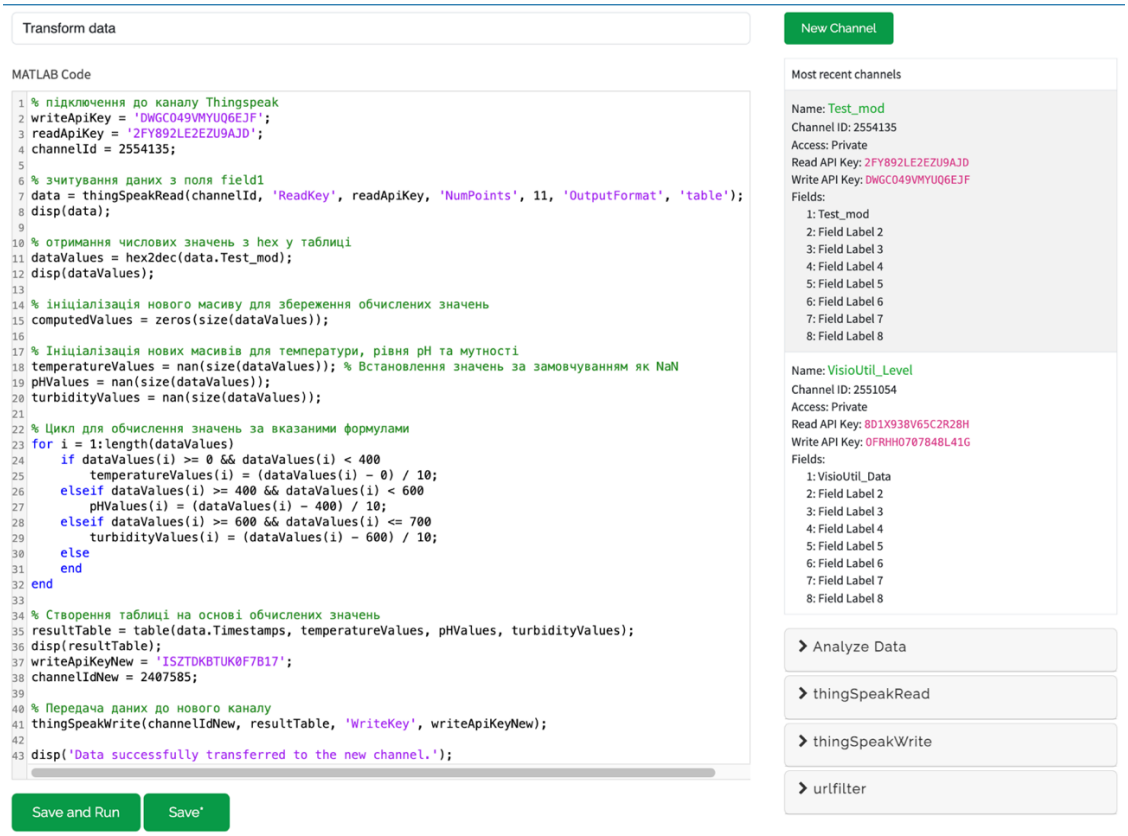

Рисунок 3.9 – MATLAB код у ThingSpeak

Для використання коду на іншому проекті або каналі основними елементами, які потребують заміни, є API ключі та ідентифікатор каналу. Потрібно замінити:

1. writeApiKey – це ключ API, що дозволяє записувати дані до каналу ThingSpeak;
2. readApiKey – це ключ API для читання даних з каналу ThingSpeak;
3. channelId – ідентифікатор каналу, з якого читаються або в який записуються дані;
4. Змінити параметр змінної data у функції hex2dec на назву каналу.

Після обробки та аналізу даних результати можна візуалізувати прямо у ThingSpeak за допомогою MATLAB Visualizations. Це може включати створення графіків, карт тепла, гістограм та інших візуалізацій, які допомагають краще зрозуміти зібрані дані. Візуалізації можна налаштовувати та публікувати для ширшої аудиторії, забезпечуючи легкий доступ до результатів аналізу.

3.3 Аналіз результатів експериментального дослідження

Після завершення етапу збору та обробки даних за допомогою веб-сервісу ThingSpeak, відбулася їх конвертація з вихідного формату у більш використовуваний для подальшого аналізу. Результатом цієї конвертації стало отримання виміряних значень pH, температури та каламутності води, які можна побачити на таблицях 3.1-3.3.

Таблиця 3.1 - Виміряні значення pH

Дата та час вимірювання	pH
2024-05-22 16:31:16	8.1
2024-05-22 16:39:48	8.2
2024-05-22 16:48:19	7.9

Таблиця 3.2 - Виміряні значення температури

Дата та час вимірювання	Температура (°C)
2024-05-22 16:28:51	18.5
2024-05-22 16:38:16	18.7
2024-05-22 16:46:50	18.6

Таблиця 3.3 - Виміряні значення каламутності

Дата та час вимірювання	Каламутність (NTU)
2024-05-22 16:32:47	59
2024-05-22 16:41:19	63
2024-05-22 16:49:51	62

На основі даних, отриманих під час дослідження озера, можна зробити висновок, що параметри води мали наступні характеристики:

1. Значення pH води варіювались від 7.9 до 8.2, що свідчить про слаболужне середовище. Значення змінювались незначно, що може вказувати на стабільний хімічний баланс у воді.
2. Температура води коливалась від 17°C до 20.5°C протягом дня, що є типовим для весняного чи ранньо літнього періоду. Загалом температура залишалася в межах нормального діапазону для цього часу року.
3. Вимірювання каламутності показали значення від 59 до 63 NTU, що може вказувати на присутність дрібних часток або забруднень у воді.

3.4 Висновки

В результаті дослідження та експериментування за допомогою хмарного сервісу ThingSpeak, було виконано збір та первинну обробку даних про якість води з природних водойм. Результати показують ефективність системи моніторингу для відстеження водної якості в реальному часі, що може слугувати надійним інструментом для екологічного моніторингу.

Для подальшого вдосконалення та ефективного використання системи моніторингу якості води можливе покращення можна здійснити, додаючи до системи новий сенсор.

4 РОЗШИРЕННЯ ФУНКЦІОНАЛУ IOT-СИСТЕМИ

4.1 Аналіз сенсорів для Arduino Uno

У рамках бакалаврської роботи одним з досліджуваних сенсорів є сенсор розчиненого кисню від компанії DFRobot (рис. 4.1).

Рисунок 4.1 - Сенсор розчиненого кисню від DFRobot

Цей сенсор призначений для вимірювання кількості кисню, який знаходиться у розчиненому стані в воді. Розчинений кисень є ключовим параметром, який впливає на біологічні процеси в водних екосистемах і важливий для підтримання життя водних організмів, зокрема риб та мікроорганізмів. Моніторинг рівня розчиненого кисню дозволяє оцінювати екологічний стан водойм, виявляти зони з низьким вмістом кисню, що може свідчити про забруднення або евтрофікацію, та вживати заходів для відновлення якості води та здоров'я водних екосистем.

Сенсор розчиненого кисню виконаний у вигляді гальванічної проби, що має діапазон виявлення від 0 до 20 мг/л. Він може працювати в температурному діапазоні від 0 до 40 градусів Цельсія. Час реакції сенсора складає до 90 секунд для досягнення 98% повної відповіді при температурі 25 градусів Цельсія. Діапазон тиску, на який розрахований сенсор, становить від 0 до 50 PSI. Термін служби електрода становить один рік при нормальному використанні.

Щодо обслуговування, капелюшок мембрани потрібно змінювати кожні 1-2 місяці, якщо використовується в мулових водах, та кожні 4-5 місяців для чистої води. Рідину для заповнення необхідно замінювати щомісяця. Довжина кабелю складає 2 метри, а з'єднувач проби – це BNC.

Крім того, до комплекту входить плата перетворювача сигналів, яка забезпечує напругу живлення від 3.3 до 5.5 вольт і виводить сигнал від 0 до 3.0 вольт. Коннектор кабелю також є BNC, а з'єднувач сигналу – це аналоговий інтерфейс Gravity (PH2.0-3P). Розміри плати складають 42 мм на 32 мм [21].

Одним з ключових параметрів, який важливо вимірювати, є солоність води. Для цього можна використовувати сенсор солоності від компанії Vernier — Vernier Salinity Sensor. Цей сенсор призначений для точного вимірювання концентрації розчинених солей у воді, що є критично важливим для аналізу водних екосистем, аквакультури та водопостачання.

Сенсор Vernier Salinity Sensor дозволяє отримати кількісну оцінку солоності, що може використовуватись для моніторингу змін у солоності водойм внаслідок природних флуктуацій чи антропогенного впливу. Результати вимірювань солоності можуть слугувати важливим індикатором стану води, впливаючи на рішення щодо управління водними ресурсами та охорони навколишнього середовища.

Сенсор Vernier Salinity Sensor зображений на рисунку 4.2.

Рисунок 4.2 - Vernier Salinity Sensor

Сенсор від Vernier має наступні властивості [22]:

– Діапазон сенсору солоності: від 0 до 50 ppt (від 0 до 50,000 ppm);

– Точність за заводським калібруванням: ±3% від повного діапазону вимірювань;

– Точність з індивідуальним калібруванням: ±1% від повного діапазону вимірювань;

– Час реакції: 90% від повного діапазону вимірювань за 10 секунд;

– Компенсація температури: автоматична від 5 до 35°C;

– Температурний діапазон використання: від 0 до 80°C;

– Стала клітини: 10 см^-1;

– Опис: тип занурення, корпус з епоксидної смоли, паралельні платинові електроди;

– Розміри: зовнішній діаметр 12 мм, довжина 150 мм.

Також важливим параметром моніторингу води є електропровідність.

Електропровідність є критичним параметром, що вказує на концентрацію розчинених іонів у воді, що у свою чергу, може служити індикатором різноманітних змін у хімічному складі водойм. Додавання цього сенсора

дозволяє не тільки підвищити точність вимірювань, але й забезпечити більшу інформативність даних для аналізу та вжиття відповідних заходів у водних екосистемах. Це нововведення сприятиме кращому розумінню стану водних ресурсів та ефективнішому управлінню ними, відповідаючи вимогам сучасної екологічної політики та потребам спільноти.

Електропровідність води – це кількісна характеристика, яка визначається наявністю заряджених частинок – позитивних та негативних іонів. До останніх відносяться хімічні елементи, що входять до складу наступних органічних та неорганічних сполук: луги, солі лужноземельних та інших металів, насамперед хлориди та сульфіди (сульфати), карбонати [22].

Сенсор DFRobot Gravity: аналоговий вимірювач електропровідності (рис. 4.3) призначений для вимірювання електропровідності водних розчинів і оцінки якості води.

Рисунок 4.3 – Сенсор DFRobot Gravity: аналоговий вимірювач електропровідності

В офіційній специфікації DFRobot [24] вказано, що сенсор має такі параметри:

1) Плата перетворення сигналу (трансмітер):

- Напруга живлення: 3.0~5.0V;
- Вихідна напруга: 0~3.4V;
- Роз'єм для проби: BNC;
- Роз'єм сигналу: PH2.0-3Pin;
- Точність вимірювань: ±5% від повного діапазону;
- Розмір плати: 42 мм * 32 мм / 1.65 дюйма * 1.26 дюйма.

2) Сенсор електропровідності**:**

- Тип проби: проба лабораторного рівня;
- Стала клітини: 1.0;
- Підтримуваний діапазон виявлення: 0~20 мс/см;
- Рекомендований діапазон виявлення: 1~15 мс/см;
- Температурний діапазон: 0~40°C;
- Термін служби: більше 0.5 років (залежить від частоти використання);
- Довжина кабелю: 100 см.

4.2 Розробка нових архітектурних рішень для вдосконалення пристрою

Для вдосконалення моніторингу стану води у роботі пропонується інтегрувати в існуючу IoT-систему додатковий сенсор DFRobot Gravity, що здатний вимірювати електропровідність води.

Структура пристрою для моніторингу якості води була оновлена і до вже існуючої конфігурації було включено новий сенсор електропровідності (рис. 4.4).

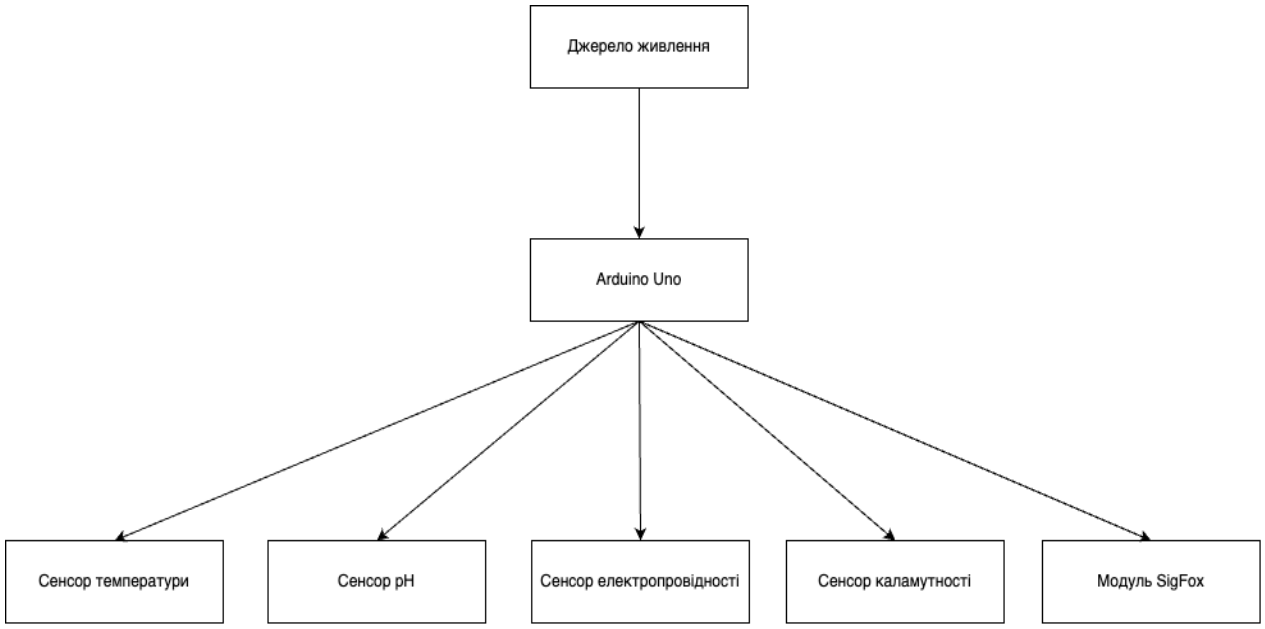

Рисунок 4.4 - Структурна схема запропонованого пристрою

Блок-схема алгоритму для пристрою моніторингу якості води також оновлена для інтеграції нового сенсора електропровідності (рис. 4.5), що дозволяє системі проводити більш глибокий аналіз водних ресурсів. Нова схема включає додаткові етапи обробки даних, які забезпечують зчитування, обробку і аналіз показників електропровідності разом з іншими ключовими параметрами води, такими як температура, pH та каламутність. Оновлення схеми вимагає врахування нових даних від сенсора електропровідності, їх аналізу на мікроконтролері Arduino Uno, і передачі цих даних через модуль SigFox для подальшої візуалізації та обробки у хмарних сервісах.

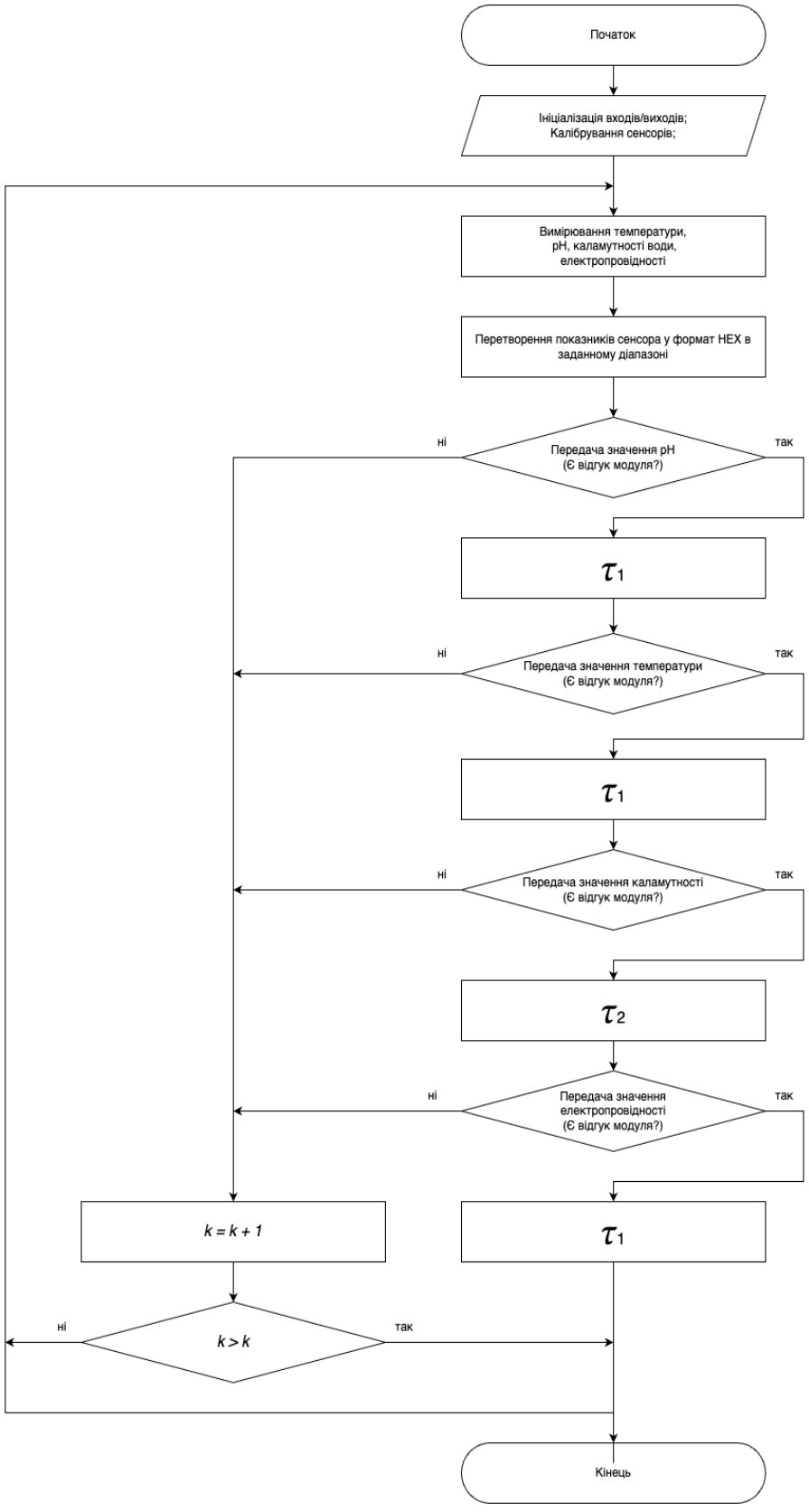

Рисунок 4.5 - Блок схема алгоритму роботи програми

4.3 Розробка програмного коду Arduino Uno

У рамках дослідження сенсора електропровідності від DFRobot, аналізуючи офіційну документацію [23], приклад коду (рис. 4.6), який необхідно інтегрувати у програмне забезпечення для використання сенсора.

```
#include "DFRobot_EC.h"
#include <EEPROM.h>

#define EC_PIN A1
float voltage,ecValue,temperature = 25;
DFRobot_EC ec;

void setup()
{
  Serial.begin(115200);
  ec.begin();
}

void loop()
{
    static unsigned long timepoint = millis();
    if(millis()-timepoint>1000U)  //time interval: 1s
    {
      timepoint = millis();
      voltage = analogRead(EC_PIN)/1024.0*5000;   // read the voltage
      //temperature = readTemperature();          // 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,1);
      Serial.print("^C  EC:");
      Serial.print(ecValue,2);
      Serial.println("ms/cm");
    }
    ec.calibration(voltage,temperature);          // calibration process by Serail CMD
}

float readTemperature()
{
  //add your code here to get the temperature from your temperature sensor
}
```

Рисунок 4.6 - Код для інтеграції

Перш за все, у процесі написання програми для Arduino, важливо підключити необхідні бібліотеки за допомогою директиви #include, які містять функції та класи для роботи з обладнанням або специфічними даними. Далі, використовуючи директиву #define, потрібно визначити константи, що спростять код та зроблять його більш читабельним. Також слід створити змінні, які будуть використовуватися для зберігання та обробки даних у програмі. Після цього, ці компоненти можна інтегрувати у логіку програми для виконання необхідних завдань.

На рисунку 4.7 показано інтеграцію бібліотек, директив та змінних.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SoftwareSerial.h>
#include "DFRobot_EC.h"
#include <EEPROM.h>

#define ONE_WIRE_BUS 2
#define SENSOR_PIN A5
#define LED_PIN 13
#define LED_ON_TIME 20000
#define EC_PIN A1

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
SoftwareSerial mySerial(10, 11);
SoftwareSerial bluetooth(0, 1);
DFRobot_EC ec;

struct SensorData {
  float pH;
  float temperature;
  int turbidity;
  float ecValue;
};
```

Рисунок 4.7 - Інтеграція бібліотек, директив та змінних

Для ініціалізації сенсора електропровідності в методі «setup» програми Arduino, потрібно додати рядок коду, який активує цей сенсор. Це робиться за допомогою методу «begin», який належить до об'єкту сенсора (рис. 4.8).

```
void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  sensors.begin();
  ec.begin();
  pinMode(LED_PIN, OUTPUT);
  lastSigfoxSendTime = millis();
  currentSendType = 0;
}
```

Рисунок 4.8 - Інтеграція ініціалізації сенсора

На рисунку 4.8 додано використання методу «begin» класу «DFRobot_EC». Це гарантує, що сенсор електропровідності готовий до зчитування даних, коли програма перейде до основного циклу виконання.

У програмі для Arduino було реалізовано (рис. 4.9) і запущено (рис. 4.10) метод «measureEC».

```
void measureEC(SensorData& data) {
    int buffer_arr[10], temp;
    unsigned long int avgval;

    for (int i = 0; i < 10; i++) {
        buffer_arr[i] = analogRead(EC_PIN);
        delay(30);
    }

    for (int i = 0; i < 9; i++) {
        for (int j = i + 1; j < 10; j++) {
            if (buffer_arr[i] > buffer_arr[j]) {
                temp = buffer_arr[i];
                buffer_arr[i] = buffer_arr[j];
                buffer_arr[j] = temp;
            }
        }
    }

    avgval = 0;
    for (int i = 2; i < 8; i++)
        avgval += buffer_arr[i];

    float voltage = (float)avgval * 5.0 / 1024 / 6;

    float ecValue = ec.readEC(voltage, data.temperature);
    data.ecValue = ecValue;
}
```

Рисунок 4.9 - Реалізація методу «measureEC»

```
64      SensorData data;
65
66      measurePH(data);
67      measureTemperature(data);
68      measureTurbidity(data);
69      measureEC(data);
70
71      printData(data);
```

Рисунок 4.10 - Запуск методу «measureEC»

Цей метод призначений для зчитування значень електропровідності з відповідного сенсора. «measureEC» викликається в головному циклі програми для отримання актуальних даних про стан електропровідності води.

У методі «measureEC» зроблено ініціалізацію змінних - масиву для зберігання десяти зчитувань з сенсора, тимчасової змінної для сортування та змінної для зберігання середнього значення. Потім у циклі здійснюються зчитування з аналогового піну, кожне з яких супроводжується затримкою для стабілізації зчитувань.

Далі проводиться фільтрація даних за допомогою методу сортування бульбашкою для видалення випадкових шумів, що забезпечує стабільність отриманих даних. Середнє значення обчислюється з серединних шести значень масиву, виключаючи два крайні значення, що дозволяє зменшити вплив аномалій на кінцевий результат.

Результат середнього значення перетворюється в напругу, яка потім використовується для розрахунку електропровідності з компенсацією температури. Виклик методу «ec.readEC» з бібліотеки DFRobot_EC дозволяє отримати точне значення електропровідності, яке записується у структуру даних, готову для подальшої обробки або відображення. Цей метод є важливим інструментом для точного моніторингу якості водних ресурсів.

На рисунку 4.11 додано запуск методу «sendECToSigfox», який знаходиться у циклі відправлення даних модулем Sigfox.

```
void sendToSigfox(const SensorData& data) {
  switch (currentSendType) {
    case 0:
      sendTemperatureToSigfox(data);
      break;
    case 1:
      sendPHToSigfox(data);
      break;
    case 2:
      sendTurbidityToSigfox(data);
      break;
    case 3:
      sendECToSigfox(data);
      break;
  }

  currentSendType = (currentSendType + 1) % 4;
}
```

Рисунок 4.11 - запуск методу «sendECToSigfox»

На рисунку 4.12 представлено метод «sendECToSigfox». Цей метод дозволяє забезпечувати точне та надійне вимірювання електропровідності, конвертуючи отримані аналогові значення у шістнадцятковий формат. Після форматування, дані уніфіковано відправляються через серійний порт за допомогою зв'язку з модулем Sigfox, що дозволяє збирати та аналізувати інформацію про стан води на віддалених платформах моніторингу. Цей метод важливий для забезпечення зв'язку між локальною системою вимірювання та централізованою системою обробки даних, сприяючи покращенню управління водними ресурсами.

```
void sendECToSigfox(const SensorData& data) {
  String hexValue_EC = String((int)data.ecValue, HEX);

  if (hexValue_EC.length() == 1) {
    hexValue_EC = "000" + hexValue_EC;
  } else if (hexValue_EC.length() == 2) {
    hexValue_EC = "00" + hexValue_EC;
  } else if (hexValue_EC.length() == 3) {
    hexValue_EC = "0" + hexValue_EC;
  }

  char message_EC[12];
  sprintf(message_EC, "AT$SF=%s", hexValue_EC.c_str());

  mySerial.println(message_EC);
  Serial.println(message_EC);
}
```

Рисунок 4.12 - метод «sendECToSigfox»

4.4 Висновки

У рамках бакалаврської дипломної роботи виконано значну роботу з розробки та оптимізації системи моніторингу стану поверхневих вод. Серед ключових досягнень проекту – розробка нової структурної схеми запропонованого пристрою, яка включає всі необхідні компоненти та забезпечує ефективну інтеграцію сенсорів та засобів передачі даних. Також була створена нова блок-схема алгоритму, яка детально описує всі кроки обробки та аналізу отриманих від сенсорів даних, забезпечуючи надійність та точність функціонування системи.

Крім того, було розроблено спеціалізований програмний код для Arduino, що дозволяє здійснювати збір даних в реальному часі, їх обробку та відправлення через модуль Sigfox для віддаленого моніторингу. Цей код включає функції для вимірювання різних параметрів води, таких як електропровідність, pH, температура, каламутність, а також механізми для калібрування та налагодження системи.

ВИСНОВКИ

В ході виконання бакалаврської дипломної роботи виконано:

1. Підвищено інформативність та ефективність моніторингу показників стану природних водойм за рахунок впровадження IoT-системи, яка реалізується з використанням сучасних технологій;

2. Охарактеризовано об’єкт дослідження, відзначено на основі аналізу систем моніторингу природних вод основні технічні параметри, які характеризують їх стан. Здійснено аналіз аналогів. Також розглянуто апаратне і програмне забезпечення для систем моніторингу, що використовується для визначення стану природних вод: річок, озер, ставків.

3. На основі проведеного аналізу здійснено вибір оптимального рішення IoT-системи моніторингу для розв’язання поставленої задачі.

4. В роботі запропоновано модернізація розробленого на кафедрі САІТ пристрою моніторингу, що фіксував температуру, pH та каламутність води за рахунок додавання сенсору електропровідності і відповідних методів до програмного забезпечення мікроконтролера.

5. Проведені на озері, що знаходиться на території ВНТУ експериментальні дослідження IoT-системи моніторингу стану природних водойм засвідчили коректність роботи запропонованого пристрою.

6. Визначено задачі, що основним напрямом подальшого вдосконалення системи є інтеграція інших сенсорів, що дозволяють отримати значення інших показників для моніторингу води.

Отже, під час написання бакалаврської дипломної роботи досягнуто основні завдання – удосконалено пристрій автоматичного збору даних, який успішно пройшов тестові експериментальні дослідження на реальному об'єкті перевірена і доведена працездатність та ефективність розробленого програмного рішення, яке за певних умов, перевищує аналоги.

За результатами даної роботи була зроблена доповідь на тему «IoT система моніторингу поверхневих вод» на VI Міжнародній науковій студентській конференції Молодіжної наукової ліги (2024) з публікацією тез [1].

Робота виконувалась на замовлення науково-дослідної лабораторії автоматизованих систем управління в енергетиці та транспорті (НДЛ АСУЕТ) кафедри САІТ Вінницького національного технічного університету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конотоп Б.П., Гаврилов Г.А., Панасюк В.Р., Проценко Д.П. IoT система моніторингу поверхневих вод., Молодіжна наукова ліга, міжнародна наукова студентська конференція «Глобалізація наукових знань: міжнародна співпраця та інтеграція галузей наук (Біла Церква, 2024 рр.)». URL: https://archive.liga.science/index.php/conference-proceedings/issue/view/inter-07.06.2024

2. CEOBS, Ukraine conflict environmental briefing: Water, 2022. URL: https://ceobs.org/ukraine-conflict-environmental-briefing-water/

3. OECD, Developing a Water Policy Outlook for Georgia, the Republic of Moldova and Ukraine, OECD Studies on Water, OECD Publishing, Paris 2021, 118p.

4. Бабій В. М. Оцінка впливу твердих побутових відходів на стан поверхневих вод у Львівській області та обґрунтування природоохоронних заходів. Кваліфікаційна робота. Кафедра екології. Львів-Дубляни, Львівський НУП, 2024 р., 79 с. URL: https://repository.lnup.edu.ua/jspui/handle/123456789/1086

5. Постанова про «Порядок здійснення державного моніторингу вод» (3 пункт порядку), від 24 грудня 2021 року № 1336. URL: https://zakon.rada.gov.ua/laws/show/758-2018-%D0%BF#Text

6. Adroit's data buoy has received technical approval from the Ministry of Primary Industries, 2022. URL: https://adroit.nz/salinity-monitoring-data-buoy-receives-mpi-approval/

7. Smart Water Xtreme Liebelium, 2022. URL: https://market.netmoregroup.com/en/listings/1801852-smart-water-extreme-by-libelium

8. YSI EXO1, 2018. URL: https://www.ysi.com/exo1

9. Micaelina Sarmiento Martinez Assessing the water quality of three bar-built estuaries in the Central Coast of California, Master of Science in Environmental Sciences and Management Projects, the Faculty of California Polytechnic State University, 2021, 70p. URL: https://digitalcommons.calpoly.edu/nres_rpt/14/

10. N. D. Susanti, D. Sagita, I. F. Apriyanto, C. E. W. Anggara, D. A. Darmajana, A. Rahayuningtyas, Design and Implementation of Water Quality Monitoring System (Temperature, pH, TDS) in Aquaculture Using IoT at Low Cost, Indonesian Institutes of Sciences, Centre of Appropriate Technology Development. *6th International Conference of Food, Agriculture, and Natural Resource (IC-FANRES 2021).* Subang, 41213, Indonesia, Advances in Biological Sciences Research, volume 16. DOI:10.2991/absr.k.220101.002

11. Гончаренко Д. В., Мокін В.Б., Проценко Д. П., Горячев Г. В., Варчук І. В., Іот-система вимірювання параметрів стану вод на базі sigfox., ВНТУ 2024. URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/41813/20077.pdf

12. Baballe, M. A., Bello, M. I., & Ayagi, S. H. Obstacle Avoidance Robot using an ultrasonic Sensor with Arduino Uno. Global Journal of Research in Engineering & Computer Sciences, Vol. 3 Issue 5 URL: https://gjrpublication. com/gjrecs,

13. Характеристика цифрового датчика температури DS18B20 фірми « Dallas Semiconductor» та дослідження його точності / О. В. Микитин // Гірничий вісник., 2014. Вип. 98. С. 86-89. URL: http://nbuv.gov.ua/UJRN/girvi_2014_98_23

14. HANTORO, Kusdarnowo. Monitoring the pH and temperature of IoT-based fish farming using Arduino, Jurnal CoreIT, Universitas Bhayangkara Jaya, Bekasi, Indonesia, 2023, 9.1. DOI: 10.24014/coreit.v9i1.18908

15. Специфікація сенсора pH від DFRobot, 2024 URL: https://wiki.dfrobot.com/Gravity__Analog_pH_Sensor_Meter_Kit_V2_SKU_SEN0161-V2

16. W. L. Hakim, L. Hasanah, B. Mulyanti, A. Aminudin, Characterization of turbidity water sensor SEN0189 on the changes of total suspended solids in the water, Physics Department, Faculty of Mathematics and Science Education Universitas Pendidikan Indonesia, Journal of Physics: Conference Series, Volume 1280, Issue 2. DOI: 10.1088/1742-6596/1280/2/022064

17. Беренко Я. О. Особливості функціонування систем з низьким енергоспоживання для Інтернету речей. ,Бакалаврська робота. КПІ ім. Ігоря

Сікорського 2021, URL: https://ela.kpi.ua/server/api/core/bitstreams/2df9d3cc-315c-4bf9-b23e-59505a04ea33/content

18. Alisher S. I., Zafar B. J., Study of arduino microcontroller board, Andijan Machine Building Institute, *Science and Education Scientific Journal*, Volume 3 Issue 3, 2022, 172-179. URL: https://www.openscience.uz/index.php/sciedu

19. Gokul Chandrasekaran, Neelam Sanjeev Kumar, Chokkalingam A, Gowrishankar V. IoT enabled smart solar water heater system using real time ThingSpeak IoT platform. *IET Renewable Power Generation*, 1–13 (2023). DOI: https://doi.org/10.1049/rpg2.12760

20. Online Calculators & Tools, 2024. URL: https://www.rapidtables.com/

21. Специфікація сенсора розчиненого кисню від DFRobot, 2024 URL: https://wiki.dfrobot.com/Gravity__Analog_Dissolved_Oxygen_Sensor_SKU_SEN0237

22. Властивості сенсора солоності від компанії Vernier, 2024 URL: https://www.vernier.com/product/salinity-sensor/

23. Документація DFRobot сенсор електропровідності, 2024 URL: https://www.dfrobot.com/product-1123.html

24. Церуш В. В. Екологічне обґрунтування якості води Дніпровського водосховища в межах міста Дніпро, Магістерська дипломна робота, ДДАУ 2024. URL: https://dspace.dsau.dp.ua/bitstream/123456789/9510/1/1.pdf

Додаток А

(обов’язковий)

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

“____”_____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську дипломну роботу

ІОТ-СИСТЕМА МОНІТОРИНГУ ПОКАЗНИКІВ СТАНУ ПРИРОДНИХ ВОДОЙМ. ПІДСИСТЕМА ЗБОРУ ДАНИХ

08-34.БДР.006.00.000 ТЗ

Керівник: к.т.н., доц.

_____ Дмитро ПРОЦЕНКО

«__» _____ 2024 р.

Розробив студент гр. 2ІСТ-20б

_____ Богдан КОНОТОП

«__» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2024р., та індивідуальне завдання на БДР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2024р.

2. Джерела розробки:

1) Гончаренко Д. В., Мокін В.Б., Проценко Д. П., Горячев Г. В., Варчук І. В., Іот-система вимірювання параметрів стану вод на базі sigfox., ВНТУ 2024. URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/41813/20077.pdf

2) Офіційна документація DFRobot сенсор електропровідності, 2024. URL: https://www.dfrobot.com/product-1123.html

3. Мета і призначення роботи.

Метою роботи є підвищення інформативності та ефективності моніторингу показників стану природних водойм за рахунок впровадження IoT-системи, яка реалізується з використанням сучасних технологій.

4. Вихідні дані для проведення робіт:

1) Станція Sigfox, модуль зв'язку SFM10R1P, дані зібрані сенсорами температури DS18B20,ph DFRobot, каламутності SEN0189;

2) Місце проведення дослідження – водойма на території ВНТУ.

5. Методи дослідження:

Використання IoT-системи моніторингу показників стану природних вод

6. Етапи роботи і терміни їх виконання:

a) Аналіз предметної області _____ – _____

b) Порівняльний аналіз IoT-систем моніторингу показників стану природних водойм _____ – _____

c) Розроблення IoT-системи показників стану природних водойм _____ – _____

d) Проведення експериментального дослідження розробленої системи _____ – _____

e) Оформлення матеріалів до захисту БДР _____ – _____

7. Очікувані результати та порядок реалізації

Розробка IoT-системи для моніторингу стану природних водойм, здатної точно вимірювати ключові параметри води та передавати дані для подальшого аналізу.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)».

9. Порядок приймання роботи

Публічний захист «___» _____ 2024 р.

Початок розробки «___» _____ 2024 р.

Граничні терміни виконання БДР «___» _____ 2024 р.

Розробив студент групи 2ІСТ-20б _____ Богдан КОНОТОП

Додаток Б

(обов'язковий)

Протокол перевірки бакалаврської дипломної роботи на наявність текстових запозичень

Назва роботи: «IoT-система моніторингу показників стану природних водойм. Підсистема збору даних»
Тип роботи: бакалаврська дипломна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 95% Схожість 5%

Аналіз звіту подібності (відмітити потрібне)

● Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
○ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
○ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Богдан КОНОТОП

Керівник роботи _____ (підпис) Дмитро ПРОЦЕНКО

Додаток В

(довідниковий)

Фрагмент лістингу програми

Код файлу додатку Arduino:

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SoftwareSerial.h>
#include "DFRobot_EC.h"
#include <EEPROM.h>

#define ONE_WIRE_BUS 2
#define SENSOR_PIN A5
#define LED_PIN 13
#define LED_ON_TIME 20000
#define EC_PIN A1

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
SoftwareSerial mySerial(10, 11);
SoftwareSerial bluetooth(0, 1);
DFRobot_EC ec;

struct SensorData {
  float pH;
  float temperature;
  int turbidity;
  float ecValue;
};

float calibration_value = 31.35;
unsigned long lastSigfoxSendTime = 0;
unsigned long delayBetweenSigfox = 90000;
unsigned long delayAfterThirdTransmission = 300000;

int currentSendType = 0;
int transmissionsCount = 0;
int responsesCount = 0;

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  sensors.begin();
  ec.begin();
  pinMode(LED_PIN, OUTPUT);
  lastSigfoxSendTime = millis();
  currentSendType = 0;
}

void loop() {
```

```
 if (Serial.available()) {
  char incomingChar = Serial.read();
  if (incomingChar == '1') {
   sendTemperatureIfResumeCharZero();
  }
  else if (incomingChar == '2' || incomingChar == '4') {
   delayAfterThirdTransmission = 90000;
  }
  else if (incomingChar == '3') {
   delayAfterThirdTransmission = 300000;
  }
 }

 SensorData data;

 measurePH(data);
 measureTemperature(data);
 measureTurbidity(data);
 measureEC(data);

 printData(data);

 delay(1000);
 while (mySerial.available()) {
  char responseChar = mySerial.read();
  Serial.print(responseChar);
  delay(10);
  if (responseChar == '\n') {
   responsesCount++;
   digitalWrite(LED_PIN, HIGH);
   delay(LED_ON_TIME);
   digitalWrite(LED_PIN, LOW);
   if (responsesCount % 3 == 0) {
    // After every third response, introduce a 5-minute delay
    delay(delayAfterThirdTransmission);
   }
  }
 }
 delay(1000);

 if (transmissionsCount == 0) {
  sendToSigfox(data);
  lastSigfoxSendTime = millis();
  transmissionsCount++;
 } else {
  unsigned long currentTime = millis();
  if (currentTime - lastSigfoxSendTime > delayBetweenSigfox) {
   sendToSigfox(data);
   lastSigfoxSendTime = currentTime;
   transmissionsCount++;
  }
 }
}

void measurePH(SensorData& data) {
```

```
  int buffer_arr[10], temp;
  unsigned long int avgval;

  for (int i = 0; i < 10; i++) {
   buffer_arr[i] = analogRead(A0);
   delay(30);
  }

  for (int i = 0; i < 9; i++) {
   for (int j = i + 1; j < 10; j++) {
    if (buffer_arr[i] > buffer_arr[j]) {
     temp = buffer_arr[i];
     buffer_arr[i] = buffer_arr[j];
     buffer_arr[j] = temp;
    }
   }
  }

  avgval = 0;
  for (int i = 2; i < 8; i++)
   avgval += buffer_arr[i];

  float volt = (float)avgval * 5.0 / 1024 / 6;
  float pH_act = -5.70 * volt + calibration_value;
  pH_act = 400 + (round(pH_act * 10.0) / 10.0) * 10;

  data.pH = pH_act;
}

void measureTemperature(SensorData& data) {
  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);
  temperature = 0 + (round(temperature * 10.0) / 10.0) * 10;

  data.temperature = temperature;
}

void measureTurbidity(SensorData& data) {
  int sensorValue = analogRead(SENSOR_PIN);
  int turbidity = max(0, map(sensorValue, 0, 640, 100, 0));
  int turbidity_s = 600 + turbidity;

  data.turbidity = turbidity_s;

void measureEC(SensorData& data) {
    int buffer_arr[10], temp;
    unsigned long int avgval;

    for (int i = 0; i < 10; i++) {
        buffer_arr[i] = analogRead(EC_PIN);
        delay(30);
    }

    for (int i = 0; i < 9; i++) {
        for (int j = i + 1; j < 10; j++) {
```

```
        if (buffer_arr[i] > buffer_arr[j]) {
          temp = buffer_arr[i];
          buffer_arr[i] = buffer_arr[j];
          buffer_arr[j] = temp;
        }
      }
    }

    avgval = 0;
    for (int i = 2; i < 8; i++)
      avgval += buffer_arr[i];

    float voltage = (float)avgval * 5.0 / 1024 / 6;

    float ecValue = ec.readEC(voltage, data.temperature);
    data.ecValue = ecValue;
}

void printData(const SensorData& data) {
  // Implementation for printing data (unchanged)
}

void sendToSigfox(const SensorData& data) {
  switch (currentSendType) {
    case 0:
      sendTemperatureToSigfox(data);
      break;
    case 1:
      sendPHToSigfox(data);
      break;
    case 2:
      sendTurbidityToSigfox(data);
      break;
    case 3:
      sendECToSigfox(data);
      break;
  }

  currentSendType = (currentSendType + 1) % 4;
}

void sendTemperatureToSigfox(const SensorData& data) {
  String hexValue_temp = String((int)data.temperature, HEX);
  if (hexValue_temp.length() == 1) {
  hexValue_temp = "000" + hexValue_temp;
} else if (hexValue_temp.length() == 2) {
  hexValue_temp = "00" + hexValue_temp;
} else if (hexValue_temp.length() == 3) {
  hexValue_temp = "0" + hexValue_temp;
}
  char message_temp[12];
  sprintf(message_temp, "AT$SF=%s", hexValue_temp.c_str());

  mySerial.println(message_temp);
  Serial.println(message_temp);
```

```
}

void sendTemperatureIfResumeCharZero() {
  SensorData data;
  measureTemperature(data);

  String hexValue_temp = String((int)data.temperature, HEX);
  if (hexValue_temp.length() == 1) {
    hexValue_temp = "000" + hexValue_temp;
  } else if (hexValue_temp.length() == 2) {
    hexValue_temp = "00" + hexValue_temp;
  } else if (hexValue_temp.length() == 3) {
    hexValue_temp = "0" + hexValue_temp;
  }

  char message_temp[12];
  sprintf(message_temp, "AT$SF=%s", hexValue_temp.c_str());

  mySerial.println(message_temp);
  Serial.println(message_temp);
}

void sendPHToSigfox(const SensorData& data) {
  String hexValue_pH = String((int)data.pH, HEX);
  if (hexValue_pH.length() == 3) {
    hexValue_pH = "0" + hexValue_pH;
  }
  char message_pH[12];
  sprintf(message_pH, "AT$SF=%s", hexValue_pH.c_str());

  mySerial.println(message_pH);
  Serial.println(message_pH);
}

void sendTurbidityToSigfox(const SensorData& data) {
  String hexValue_turbidity = String(data.turbidity, HEX);
  if (hexValue_turbidity.length() == 3) {
    hexValue_turbidity = "0" + hexValue_turbidity;
  }
  char message_turbidity[12];
  sprintf(message_turbidity, "AT$SF=%s", hexValue_turbidity.c_str());

  mySerial.println(message_turbidity);
  Serial.println(message_turbidity);
}

void sendECToSigfox(const SensorData& data) {
  String hexValue_EC = String((int)data.ecValue, HEX);

  if (hexValue_EC.length() == 1) {
    hexValue_EC = "000" + hexValue_EC;
  } else if (hexValue_EC.length() == 2) {
    hexValue_EC = "00" + hexValue_EC;
  } else if (hexValue_EC.length() == 3) {
    hexValue_EC = "0" + hexValue_EC;
```

```
  }

  char message_EC[12];
  sprintf(message_EC, "AT$SF=%s", hexValue_EC.c_str());

  mySerial.println(message_EC);
  Serial.println(message_EC);
}
```

Додаток Г

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІОТ-СИСТЕМА МОНІТОРИНГУ ПОКАЗНИКІВ СТАНУ ПРИРОДНИХ ВОДОЙМ. ПІДСИСТЕМА ЗБОРУ ДАНИХ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

Рисунок Г.1 – Класифікація параметрів моніторингу води

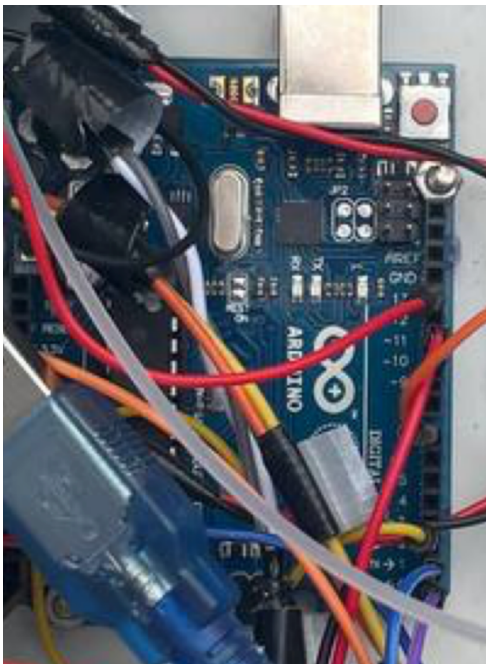

Рисунок Г.2 – Мікропроцесорна плата Arduino Uno

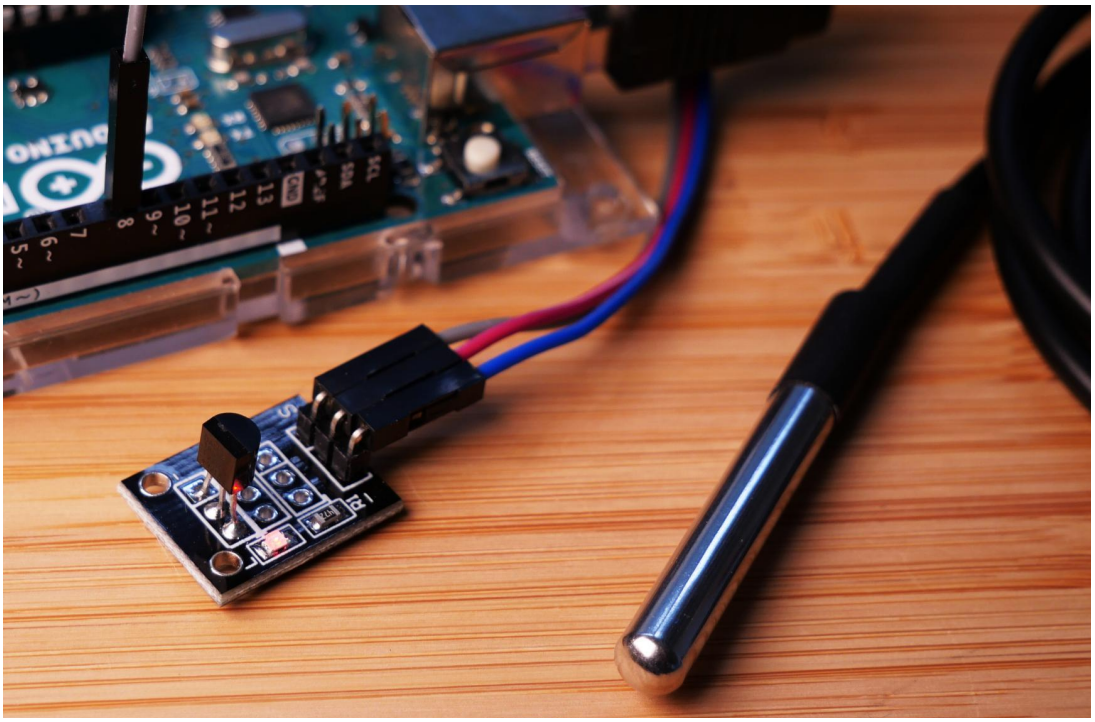

Рисунок Г.3 - Цифровий сенсор температури DS18B20

Рисунок Г.4 - Сенсор pH від компанії DF Robot

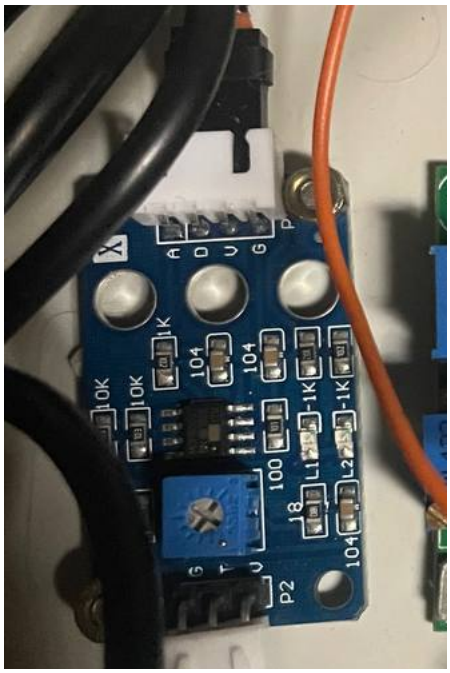

Рисунок Г.5 – Модуль каламутності SEN0189

Рисунок Г.6 – Модуль Sigfox та антена

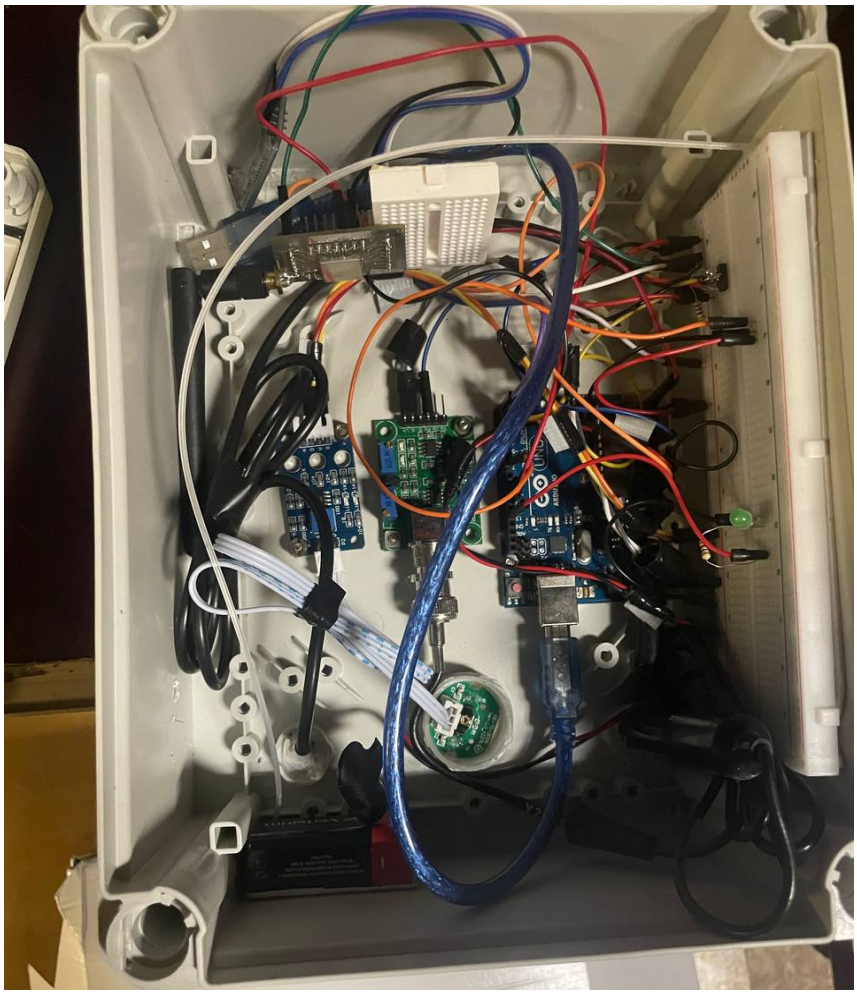

Рисунок Г.7 - Зібраний пристрій

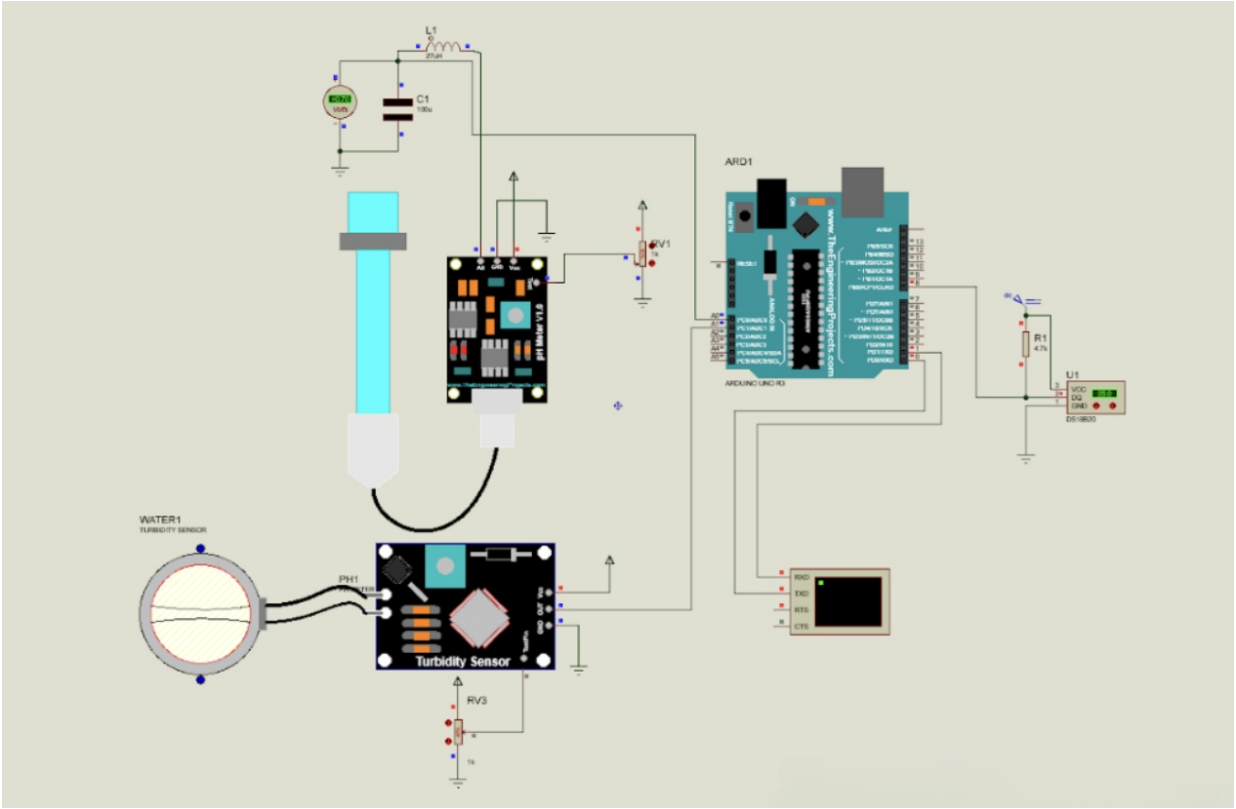

Рисунок Г.8 – Схема моделі пристрою в середовищі Proteus

Рисунок Г.9 – Проведення моніторингу води на території озера прилеглого до ВНТУ

Рисунок Г.10 - Система моніторингу якості води у роботі

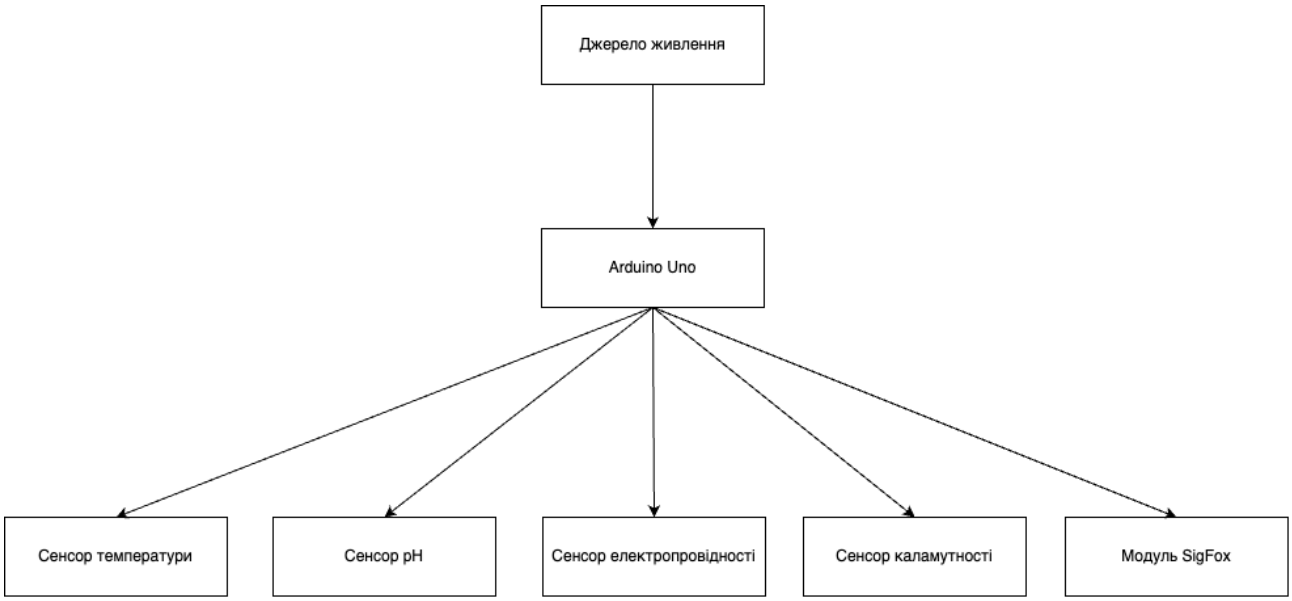

Рисунок Г.11 – Структурна схема запропонованого пристрою

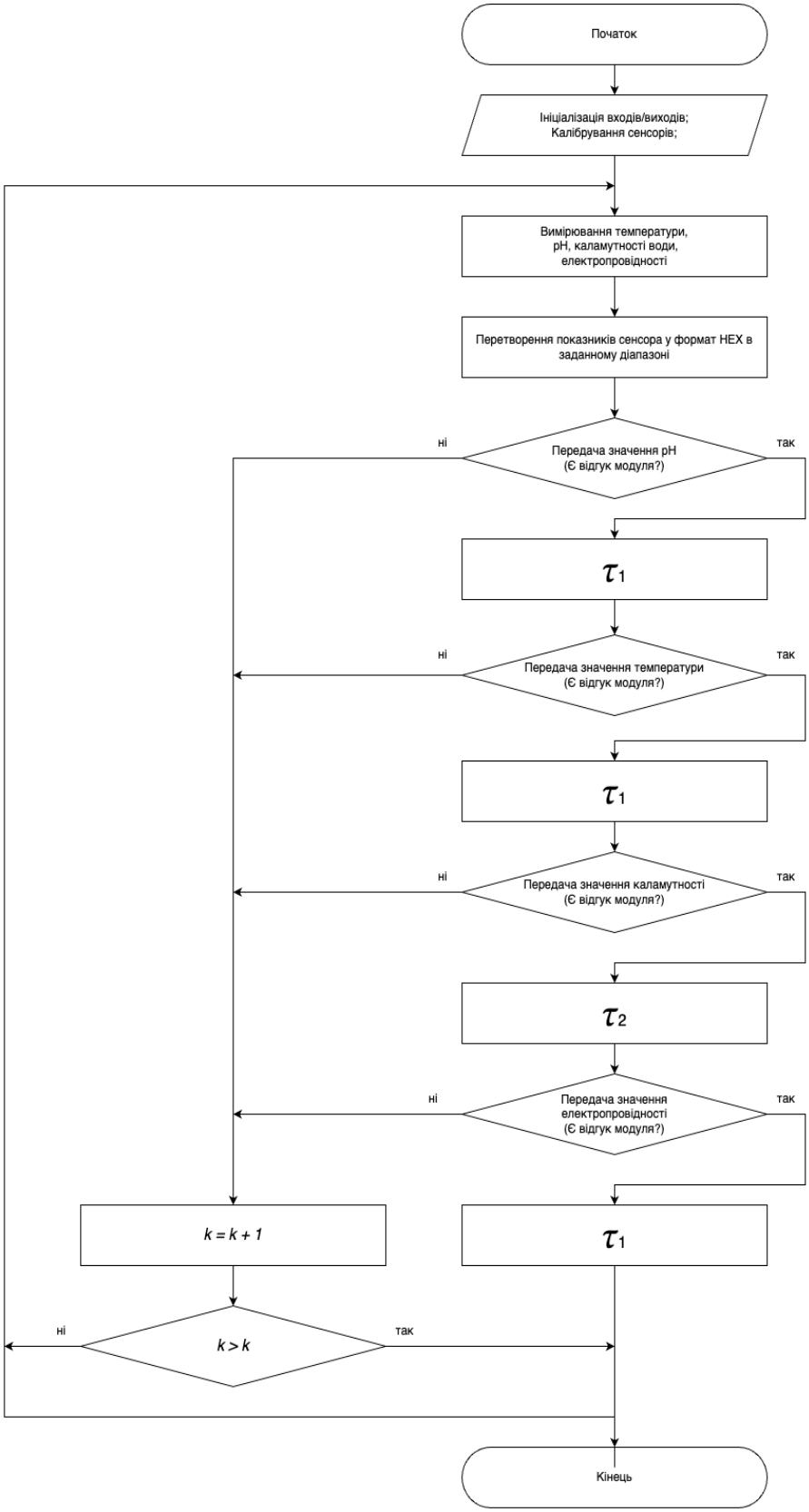

Рисунок Г.12 - Блок-схема алгоритму роботи запропонованої програми