

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач відділення
комп'ютерних технологій
Наталія СТЕФУРАК / _____ /
(підпис)
« ____ » _____ 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи
освітньо-професійного ступеня «фаховий молодший бакалавр»
зі спеціальності 123 «Комп'ютерна інженерія»
на тему:
«Система контролю рівня наповненості водяного резервуару
в приватному будинку»

Студент групи КІ-41	Дмитро ЛУШПІНСЬКИЙ	_____ (підпис)
Керівник роботи	Василь ПАВЛЮС	_____ (підпис)
Консультанти: з техніко-економічного обґрунтування	Любов МЕЛЕНЧУК	_____ (підпис)
Нормоконтролер	Ольга СЛЄПЦОВА	_____ (підпис)

Тернопіль – 2024

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділенням
комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

підпис

« ____ » _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»

студенту Лушпінському Дмитру Олеговичу

(прізвище, ім'я та по-батькові студента)

1. Тема роботи: «Система контролю рівня наповненості водяного резервуару в приватному будинку», затверджена наказом по коледжу від «27» листопада 2023р., №234а-н.
2. Термін здачі студентом завершеної роботи: «28» червня 2024 р.
3. Вихідні дані до роботи: актуальні технології та засоби контролю рівня води в резервуарах, наявні рішення на ринку, технічне завдання.
4. Перелік питань, які повинні бути розроблені в роботі:
 - а) основна частина дослідження предметної області, формалізація вимог до системи, проектування структури та алгоритму роботи системи, реалізація та тестування системи.
 - б) техніко-економічне обґрунтування аналіз ринку наявних рішень, дослідження кошторису реалізації системи, підрахунок загальної вартості розробки системи.

5. Перелік графічного матеріалу: структурна схема системи, блок-схема алгоритму роботи системи, схема з'єднання компонентів, електрична принципова схема.

6. Консультанти роботи:

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	Меленчук Л.І. вчена ступінь, звання		
	П.І.П. консультанта		

КАЛЕНДАРНИЙ ПЛАН кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до кваліфікаційної роботи.	23.11.2023	01.12.23
2.	Дослідження предметної області, огляд типових рішень.	02.12.2023	05.02.2024
3.	Дослідження технологій реалізації.	29.01.2024	07.02.2024
4.	Розробка функціональних вимог до системи та робота над її структурою.	08.02.2024	01.03.2024
5.	Проектування системи та підготовка відповідного графічного матеріалу.	03.03.2024	05.04.2024
6.	Вибір, встановлення та налаштування середовища реалізації.	18.03.2024	08.04.2024
7.	Реалізація системи та написання відповідного розділу кваліфікаційної роботи.	09.04.2024	09.05.2024
8.	Доопрацювання апаратної складової.	10.05.2024	16.05.2024
9.	Опрацювання економічного розділу та написання відповідного розділу кваліфікаційної роботи.	11.03.2024	03.05.2024
10.	Тестування системи та усунення недоліків.	17.05.2024	31.05.2024
11.	Робота над оформленням пояснювальної записки.	01.06.2024	18.06.2024
12.	Попередній захист кваліфікаційної роботи, доопрацювання.	18.06.2024	18.06.2024
13.	Підготовка до захисту кваліфікаційної роботи.	19.06.2024	27.06.2024
14.	Захист кваліфікаційної роботи.	28.06.2024	28.06.2024

Дата видачі «27»листопада 2023р. Керівник _____/ Василь ПАВЛЮС

Завдання прийняв до виконання _____/ Дмитро ЛУШПІНСЬКИЙ

Реферат

Кваліфікаційна робота. Система контролю рівня наповненості водяного резервуару в приватному будинку. 50 сторінок, 30 рисунків, 2 додатки, 6 джерел.

Метою даної роботи є розробка та реалізація системи контролю рівня наповненості водяного резервуару в приватному будинку. Система повинна контролювати процес наповнення водяного резервуару в приватному будинку шляхом постійного моніторингу стану рівня води в резервуарі та керування роботою водяного електронасоса. Система повинна не лише ефективно моніторити рівень води, але й практично унеможливити нестачу чи перелив води з резервуару.

Для реалізації проєкту було використано мікроконтролер ESP8266, середовище розробки Arduino IDE для програмування мікроконтролера ESP, бібліотеки для роботи з периферійними пристроями та онлайн сервіс Thinger.io.

У роботі розглянуто процес розробки та реалізації системи контролю рівня наповненості водяного резервуару, представлено її ключові характеристики та переваги, а також надано рекомендації щодо її практичного застосування.

КОНТРОЛЬ РІВНЯ НАПОВНЕНОСТІ ВОДЯНОГО РЕЗЕРВУАРУ,
ДАТЧИКИ РІВНЯ ВОДИ, ARDUINO, ESP8266, THINGER.IO, ІНТЕРНЕТ
РЕЧЕЙ.

Abstract

Graduation Project. System for Monitoring the Water Tank Level in a Private House. 50 pages, 30 figures, 2 appendices, 6 sources.

The aim of this work is to develop and implement a system for monitoring the water tank level in a private house. The system is designed to control the filling process of the water tank in a private house by continuously monitoring the water level in the tank and managing the operation of the water pump. The system should not only effectively monitor the water level but also practically eliminate any shortages or overflows of water from the tank.

To implement the project, the ESP8266 microcontroller, the Arduino IDE development environment for programming the ESP microcontroller, libraries for working with peripheral devices, and the online service Thingier.io were used.

The work discusses the process of developing and implementing the water tank level monitoring system, presents its key features and advantages, and provides recommendations for its practical application.

WATER TANK LEVEL MONITORING, WATER LEVEL SENSORS, ARDUINO, ESP8266, THINGER.IO, INTERNET OF THINGS.

ЗМІСТ

Скорочення та умовні позначки	6
Вступ.....	7
1 Аналіз предметної області та постановка завдання.....	8
1.1 Змістовний аналіз предметної області	8
1.2 Аналіз наявних рішень	9
1.3 Постановка завдання.....	15
2 Проєктування системи	17
2.1 Визначення компонентів системи	17
2.2 Проєктування структури системи	18
2.3 Алгоритм роботи системи	19
3 Реалізація та тестування системи	21
3.1 Вибір компонентів системи	21
3.2 Реалізація принципової електричної схеми та монтаж пристрою	25
3.3 Реалізація програмного коду.....	28
3.4 Підключення та налаштування сервісу thinger.io	30
3.5 Тестування роботи системи	38
4 Техніко-економічне обґрунтування проєкту.....	40
4.1 Аналіз ринку	40
4.2 Розрахунок витрат на реалізацію.....	41
4.3 Обґрунтування доцільності розробки	43
Висновки	45
Перелік джерел посилання	46
Додатки.....	47

					КР.КІ 24.534.07.000 ПЗ						
Зм.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Лушнінський Д.			Система контролю рівня наповненості водяного резервуару в приватному будинку			Лім.	Арк.	Акрушів	
Перевір.		Павлюс В.П								5	50
Рецензент		Посвятовська О.						ГФК. ВКТ. КІ-41			
Н. Контр.		Слепцова О.Я.									
Затверд.		Стефурак Н.А									

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ADC – Analog-to-Digital Converter

GPIO – General-Purpose Input/Output

I2C – Inter-Integrated Circuit

IDE – Integrated Development Environment

IoT – Internet of Things

OLED – Organic Light-Emitting Diode

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver/Transmitter

					<i>КР.КІ 24.534.07.000 ПЗ</i>	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному світі автоматизація та моніторинг різних аспектів повсякденного життя стають все більш невід'ємною частиною нашого існування. Особливо це стосується систем автоматизації сучасних «розумних» будинків, метою яких є забезпечення комфорту та безпеки їх власників.

Дуже часто в заміських приватних будинках відсутнє централізоване водопостачання і використовується власна свердловина з водяним резервуаром. Відповідно, контроль за рівнем наповненості водяного резервуару є доволі актуальним для таких домогосподарств.

Більшість наявних рішень у цій галузі часто не забезпечують повноцінного контролю та безпеки використання, що може призвести до непередбачених ситуацій, таких як перелив або виливання води з резервуару. Крім того, більшість з них є складними у встановленні та експлуатації, а також не завжди надійними у роботі. Часто такі системи потребують значних витрат на обслуговування та не можуть гарантувати стабільний контроль.

Створювана система контролю рівня наповненості водяного резервуару має на меті усунути згадані недоліки та забезпечити додатковий функціонал. Зокрема, завдяки постійному підключенню до Інтернету, система надаватиме можливість відстежувати рівень води в резервуарі в реальному часі та за потреби віддалено керувати ним в ручному режимі. Таким чином, система не лише дозволить ефективно моніторити рівень води, але й практично унеможливить нестачу чи перелив води з резервуару. Такий підхід дозволить уникнути можливих негативних наслідків та забезпечить безпеку використання водних ресурсів.

У цій роботі буде розглянуто процес розробки та реалізації системи контролю рівня наповненості водяного резервуару, представлено її ключові характеристики та переваги, а також надано рекомендації щодо її практичного застосування.

					<i>КР.КІ 24.534.07.000 ПЗ</i>	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Змістовний аналіз предметної області

Наразі використання систем контролю рівня наповненості водяних резервуарів у приватних будинках є досить розповсюдженим, особливо в районах, де відсутнє централізоване водопостачання або коли власники будинків використовують власні джерела води, такі як свердловини.

Більшість таких систем базуються на технології датчиків рівня води в резервуарі, які автоматично включають або вимикають насоси для наповнення резервуару.

Проте, в більшості випадків ці системи не забезпечують можливості віддаленого моніторингу або керування, що може бути незручним для власників будинків у випадках, коли вони перебувають далеко від дому або відсутні на місці.

Деякі сучасні рішення включають в себе «розумні» системи контролю, які можуть бути керовані за допомогою мобільних застосунків. Проте, не зважаючи на розвиток технологій, існують певні проблеми з надійністю та точністю деяких таких систем. Наприклад, деякі датчики можуть виявляти неправильні рівні води через забруднення або дефект, що може призвести до неправильної роботи насосів або навіть до потенційного переливу води.

Також варто відзначити, що вартість та складність встановлення деяких з цих систем можуть бути перешкодою для їхнього широкого застосування. Багато власників будинків шукають більш доступні та простіші у встановленні рішення, які водночас будуть ефективними та надійними.

Системи контролю рівня води зазвичай складаються з датчика рівня води, який може бути різної конструкції та принципів роботи, контролюючого модуля та електронасоса, який використовується для наповнення водяного резервуара. Типова структура даної системи наведена на рисунку 1.1.

					КР.КІ 24.534.07.000 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.1 – Типова структура системи контролю рівня води

Датчики рівня води можуть використовувати різні технології, такі як ультразвукові, плаваючі перемикачі, оптичні датчики або датчики з резистивними або конденсаторними зондами. Вони реагують на зміну рівня води і відправляють сигнали до системи управління.

Електронасос використовується для наповнення резервуара водою, коли рівень води в ньому занадто низький. Коли датчики виявляють, що рівень води опустився нижче певного порогового значення, керуючий модуль активує електронасос для заповнення резервуара водою з джерела чи свердловини.

1.2 Аналіз наявних рішень

Огляд датчиків рівня води

Загалом існує кілька видів датчиків, які можуть використовуватися в системах контролю рівня води в резервуарі: ультразвукові, оптичні, плаваючі перемикачі та з резистивними або конденсаторними зондами [3]. Розглянемо їх детальніше.

Ультразвукові датчики. Ці датчики вимірюють час, який потрібний для того, щоб звуковий сигнал відбився від поверхні рідини і повернувся назад до сенсора (рис. 1.2). Вони можуть працювати не лише в резервуарах з водою, але і з багатьма іншими видами рідин, оскільки не потребують прямого контакту з

рідиною. Однак ці датчики доволі примхливі до умов навколишнього середовища, від яких залежить точність вимірювань.

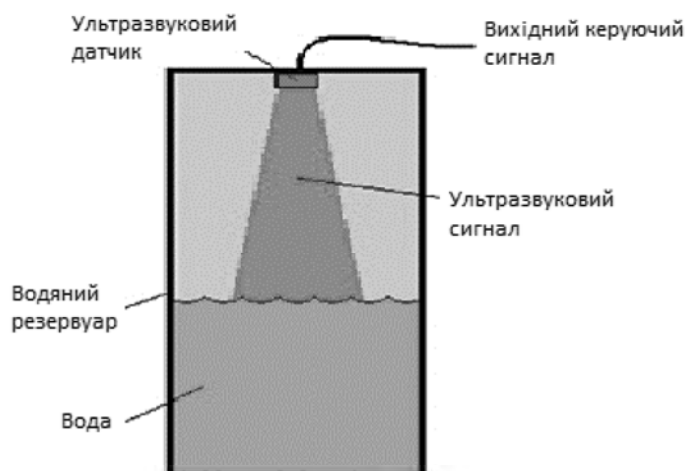


Рисунок 1.2 – Принцип роботи ультразвукового датчика рівня води

Оптичні датчики. Ці датчики використовують світлові сигнали для вимірювання рівня рідини (рис. 1.3). Вони можуть бути чутливими до забруднень або осаду, але як і ультразвукові датчики не вимагають прямого контакту з рідиною.

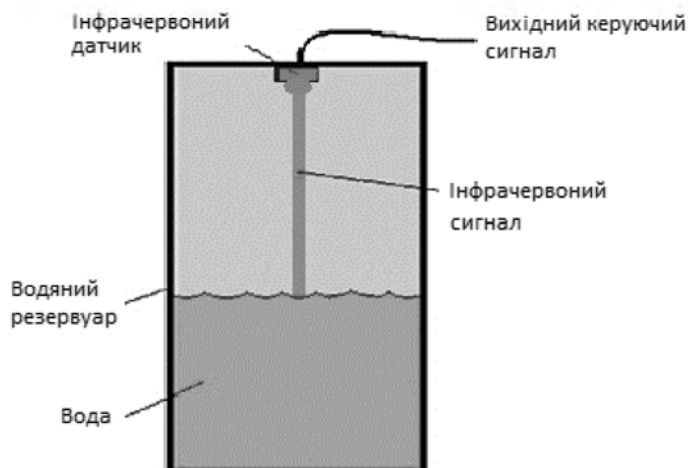


Рисунок 1.3 – Принцип роботи інфрачервоного датчика рівня води

Плаваючі перемикачі. Ці датчики вмикаються або вимикаються при досягненні рівня рідини певного порогового значення (рис. 1.4). Вони прості в установці і використанні, але можуть бути менш точними у порівнянні з іншими типами датчиків.

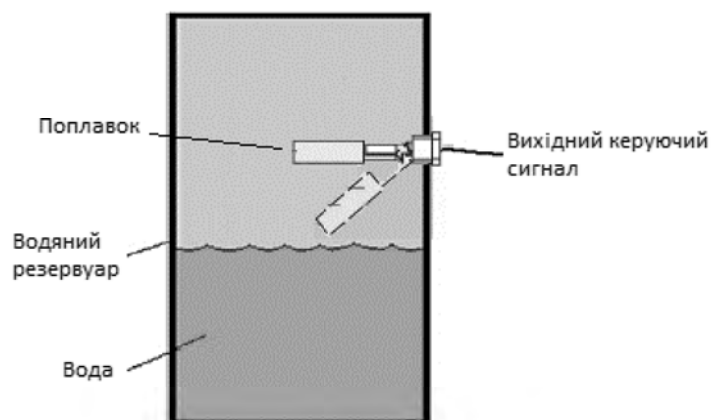


Рисунок 1.4 – Принцип роботи датчика рівня води з плаваючим перемикачем

Датчики з резистивними або конденсаторними зондами. Ці датчики використовують зміни електричного опору або ємності для вимірювання рівня рідини. Як правило, саме ці датчики використовуються в водяних резервуарах, оскільки є досить надійними, простими в експлуатації та недорогими. Принцип роботи такого датчика заснований на зміні опору або ємності між трьома його електродами, один з яких є спільним, а два інші відповідають за мінімальний та максимальний рівні води в резервуарі (рис. 1.5). Кількість таких електродів може бути більшою, якщо потрібно визначати не лише максимальний та мінімальний рівні води, але й будь-які проміжні значення.

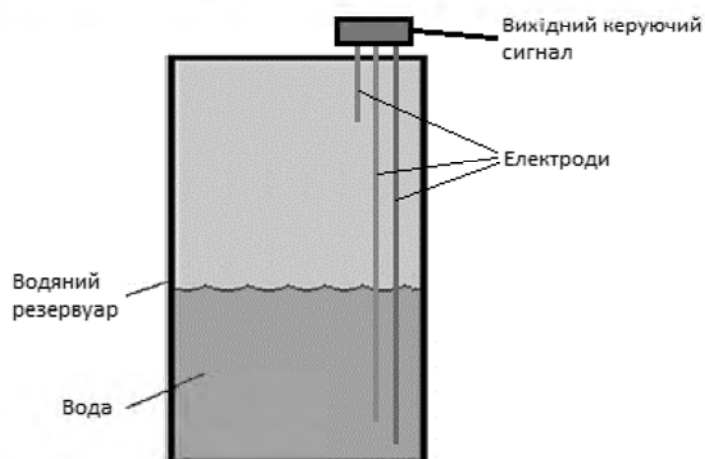


Рисунок 1.5 – Принцип роботи датчика рівня води з резистивним або конденсаторним зондом

Приклади ультразвукового (а), інфрачервоного (б) та поплавкового (в) датчиків наведені на рисунку 1.6. В датчиках з резистивними або

конденсаторними зондами в ролі електродів виступають проводи з алюмінію або нержавіючої сталі.



Рисунок 1.6 – Приклади датчиків рівня води

Огляд систем контролю рівня води

Lovato LVM20. Даний модуль є датчиком рівня рідини, який використовується для вимірювання рівня рідини в резервуарах, басейнах або інших ємностях. Він зазвичай встановлюється всередині резервуара та використовує принцип роботи на основі контактних або безконтактних сенсорів, що визначають рівень рідини.

Lovato LVM20 може мати різні конфігурації (рис. 1.7) та опції, такі як вимірювальний діапазон, матеріал корпусу, спосіб кріплення тощо. Він зазвичай включає в себе електронні компоненти (рис. 1.8) для збору та передачі даних про рівень рідини до системи моніторингу або управління [2].

Основні функції та характеристики модуля Lovato LVM20 включати:

- управління роботою насоса;
- захист насоса від «сухого» ходу;
- автоматичний долив води;
- тривожна сигналізація під час переливу (тільки для 7-зондових контролерів);
- здатність працювати з різними типами рідин, включаючи воду;
- можливість встановлення на різні типи резервуарів або ємностей.



Рисунок 1.7 – Варіанти модулів рівня води Lovato LVM20



Рисунок 1.8 – Датчики рівня води Vagner Pool до модулів Lovato LVM20

Реле контролю рівня води SSR-05. Даний модуль використовується для автоматичного контролю рівня рідини у резервуарах, басейнах, баках чи інших резервуарах. Основна функція цього реле полягає у вмиканні або вимиканні насоса або іншого електричного пристрою, який контролює рівень рідини.



Рисунок 1.9 – Модуль SSR-05 з датчиком рівня води SSE-10

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 24.534.07.000 ПЗ

Арк.

13

Основні характеристики SSR-05 включають:

Датчик рівня води: Реле може бути підключене до датчика рівня води, який знаходиться в резервуарі. Цей датчик реагує на зміни рівня рідини та передає сигнал до реле.

Налаштування порогових значень: Користувач може налаштувати порогові значення рівня води, при досягненні яких реле вмикає або вимикає підключений електричний пристрій.

Індикатори стану: SSR-05 може мати вбудовані світлодіодні індикатори стану, які показують, коли реле активоване або вимкнене.

Захист від перенапруги: Деякі моделі SSR-05 можуть мати вбудований захист від перенапруги, що забезпечує додаткову безпеку підключених електричних пристроїв.

Компактний дизайн: Реле зазвичай має компактні розміри, що дозволяє легко встановлювати його в будь-якому резервуарі або системі контролю рівня води.

Реле контролю рівня води ХН-М203.

Основні характеристики і функції реле контролю рівня води ХН-М203 (рис. 1.10) включають:

Налаштування порогових значень: Користувач може налаштувати порогові значення рівня води, при досягненні яких реле вмикає або вимикає підключений електричний пристрій.

Затримка запуску і вимкнення: ХН-М203 може мати функцію затримки запуску і вимкнення, яка дозволяє уникнути випадкового вимикання або включення підключеного обладнання в разі коливань рівня рідини.

Індикатори стану: Деякі моделі ХН-М203 мають вбудовані світлодіодні індикатори стану, які показують, коли реле активоване або вимкнене.

Компактний дизайн: Це реле має зазвичай компактні розміри, що дозволяє легко встановлювати його в будь-якому резервуарі або системі контролю рівня води.

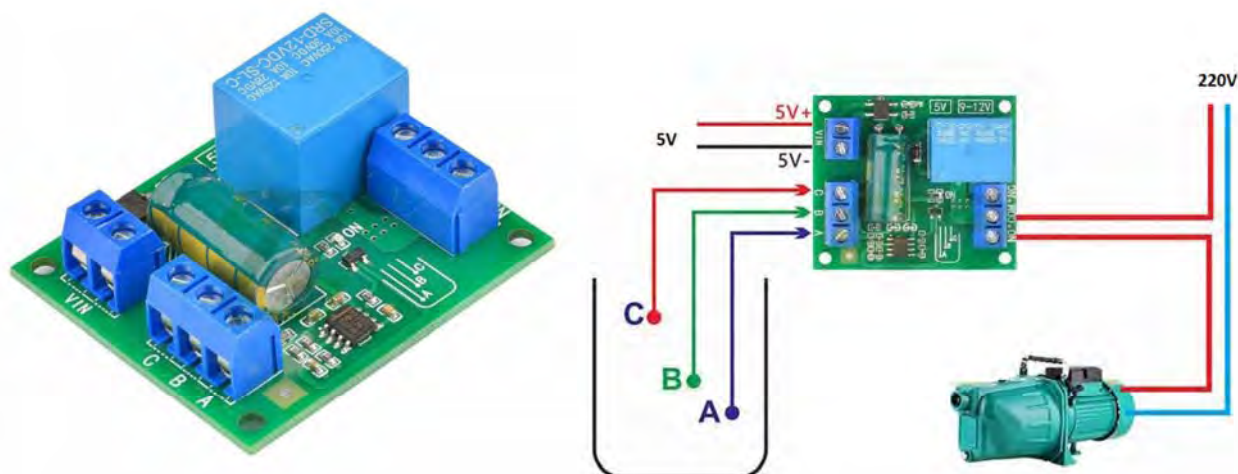


Рисунок 1.10 – Реле контролю рівня води ХН-М203

1.3 Постановка завдання

Провівши аналіз предметної області та здійснивши огляд наявних рішень можна сформулювати список практичних задач, які необхідно виконати в ході кваліфікаційної роботи:

- Розробити апаратну частину системи, включаючи датчики рівня води та електронасос, яка забезпечить точне та надійне вимірювання рівня води та контроль його наповненості.
- Реалізувати інтерфейс системи для взаємодії з користувачем, що забезпечить зручний та інтуїтивно зрозумілий доступ до інформації про рівень води та можливість керування системою.
- Забезпечити можливість віддаленого моніторингу та управління системою через Інтернет, що дозволить користувачам отримувати доступ до даних про рівень води з будь-якого місця та в будь-який час.
- Забезпечити надійну роботу системи в різних умовах експлуатації, включаючи забезпечення стійкості до зовнішніх впливів та аварійних ситуацій.
- Забезпечити можливість збереження та аналізу історичних даних про рівень води для подальшого використання та аналізу.

- Провести тестування системи для перевірки її функціональності, надійності та відповідності вимогам.
- Запропонувати можливі шляхи подальшого вдосконалення системи та її застосування в практичних умовах.

					<i>КР.КІ 24.534.07.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

2 ПРОЄКТУВАННЯ СИСТЕМИ

В попередньому розділі було визначено загальні принципи роботи систем контролю рівня наповненості водяних резервуарів та розглянута типова структура системи контролю рівня води (рис. 1.1). Крім того, було проаналізовано найпопулярніші види датчиків, які можуть використовуватися в таких системах: ультразвукові, оптичні, плаваючі перемикачі та з резистивними або конденсаторними зондами, кожен з яких має як свої переваги, так і недоліки.

Також було сформульовано список практичних задач, які необхідно виконати в ході реалізації проєкту.

Наступним етапом роботи є процес проєктування системи, який передбачає визначення усіх компонентів системи та способу їх взаємодії, а також вибір загальної архітектури системи. Розглянемо ці кроки більше детально.

2.1 Визначення компонентів системи

Беручи до уваги проведений раніше аналіз було прийнято рішення використати в даному проєкті датчик води з резистивним або конденсаторним зондом (рис. 1.5), оскільки такі датчики не мають рухомих елементів конструкції, прості у використанні та доволі дешеві. Усі ці фактори разом роблять цей тип датчиків найоптимальнішим для даного проєкту.

Окрім безпосередньо датчика рівня води система включатиме наступні компоненти:

Електронасос – використовуватиметься для коригування рівня води у резервуарі.

Реле – використовуватиметься для вмикання та вимикання живлення електронасоса.

Світлодіоди – використовуватимуться для візуальної індикації різних рівнів води в резервуарі.

Бuzzer – використовуватиметься для звукового сповіщення в разі досягнення певного рівня води у резервуарі.

					КР.КІ 24.534.07.000 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

OLED-дисплей – використовуватимуться для відображення інформації про стан системи (поточний рівень води, стан електронасоса) та виводу будь-яких попереджень або повідомлень про помилки у роботі системи.

Мікроконтролер – це центральний елемент системи, який відповідатиме за керування всіма вище переліченими компонентами. Мікроконтролер забезпечуватиме зчитування даних з датчика рівня води, включення/виключення електронасоса в залежності від рівня води, а також відтворення звукових сигналів через буюер та управління світлодіодами та OLED-дисплеєм для візуального сповіщення про поточний стан системи. Крім цього, мікроконтролер забезпечуватиме можливість віддаленого керування системою.

2.2 Проєктування структури системи

Визначившись з компонентами системи перейдемо до проєктування її структури.

Загалом система складатиметься з умовних підсистем:

1. Підсистема вимірювання рівня води в резервуарі.
2. Підсистема наповнення резервуару водою.
3. Підсистема індикації та сповіщення про стан наповненості резервуару.
4. Підсистема керування вузлами системи.
5. Підсистема віддаленого моніторингу та контролю.

Структурна схема системи наведена на рисунку 2.1.

Як видно зі структурної схеми, центральним вузлом усієї системи є підсистема керування, яка забезпечуватиме прийом інформації від підсистеми вимірювання рівня води в резервуарі, її зручне відображення засобами підсистеми індикації та сповіщення, надсилання керуючих сигналів до підсистеми наповнення резервуару водою. Підсистема віддаленого моніторингу та контролю буде окремим вузлом системи, який не матиме фізичного представлення (як решту підсистем), а реалізовуватиме безпроводовий зв'язок системи з «хмарою» та мобільним застосунком.

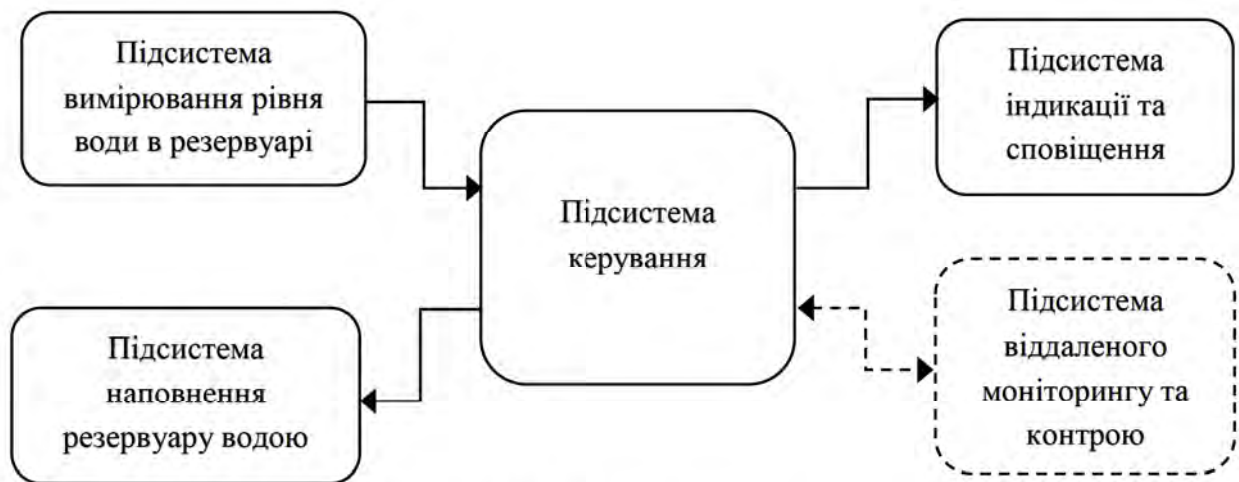


Рисунок 2.1 – Структурна схема системи

Така структура дозволить забезпечити ефективне та просте управління системою контролю рівня води в резервуарі без необхідності використання окремих мікроконтролерів для кожної підсистеми. Всі компоненти будуть підконтрольні одному мікроконтролеру, який координуватиме їхню роботу відповідно до встановлених правил та умов, а зв'язок з «хмарою» та мобільним застосунком дозволить в реальному часі моніторити та контролювати роботу системи загалом.

2.3 Алгоритм роботи системи

Процес роботи системи можна описати наступними кроками, які відбуватимуться циклічно після її ініціалізації.

- 1) Датчик рівня води зчитує поточний рівень води в резервуарі.
- 2) Отримане значення передається до мікроконтролера.
- 3) Мікроконтролер надсилає його в підсистему віддаленого моніторингу та контролю.
- 4) Мікроконтролер порівнює отримане значення із зазначеними у програмі мінімальним та максимальним значеннями рівня води в резервуарі.
- 5) Якщо отримане значення рівне мініимальному – активується водяний електронасос та здійснюється відповідне звукове сповіщення про початок процесу наповнення резервуару водою.

- 6) Якщо отримане значення рівне максимальному – водяний електронасос вимикається.
- 7) Якщо отримане значення менше максимального та більше мінімального значення – водяний електронасос залишається в попередньому стані.
- 8) Мікроконтролер надсилає отримане значення на OLED-дисплей та індикаторні світлодіоди.

Блок-схема алгоритму роботи системи наведена в додатку А.

					<i>КР.КІ 24.534.07.000 ПЗ</i>	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

В попередньому розділі було визначено основні компоненти для реалізації даної системи, спроектовано структурну схему та алгоритм роботи системи.

Наступним етапом роботи є процес реалізації та тестування системи, який передбачає вибір компонентів, які задовольняють поставленим вимогам, реалізацію електричної принципової схеми системи та написання програмного коду відповідно до алгоритму роботи системи.

3.1 Вибір компонентів системи

Як було визначено раніше, система складатиметься мікроконтролера, датчика рівня води, електронасоса, реле, OLED-дисплея та світлодіодів для візуальної індикації рівнів води та бузера для звукових сповіщень. Розглянемо детальніше основні компоненти з перелічених.

Мікроконтролер

Оскільки система повинна забезпечувати віддалений доступ через Інтернет, який найзручніше реалізувати через WiFi, то найоптимальнішим для даного проєкту вбачається використання мікроконтролера серії ESP. Враховуючи значну кількість необхідних портів вводу-виводу для підключення зовнішніх модулів було прийнято рішення використати в ролі центрального мікроконтролера ESP8266-12E, короткі технічні характеристики якого наведені в таблиці 3.1 [5].

Таблиця 3.1 – Технічні характеристики мікроконтролера ESP8266 12-E

Мікроконтролер	ESP8266EX з інтегрованим WiFi
Частота процесора	До 80 МГц
Флеш-пам'ять	4 МБ
Оперативна пам'ять	80 КБ
Вбудований WiFi	802.11 b/g/n
Інтерфейси	UART, I2C, SPI, GPIO
Напруга живлення	3.3 В
Системи розробки	Arduino IDE, NodeMCU Lua, MicroPython та інші

Для зручності роботи з даним мікроконтролером скористаємось платою розробки NodeMCU Kit (рис. 3.1), яка оснащена USB-UART конвертером на основі мікросхеми Silicon Labs CP2102 та роз'ємом micro-USB, що дозволяє дуже легко підключати її до комп'ютера для завантаження програмного коду та його відлагодження [6].

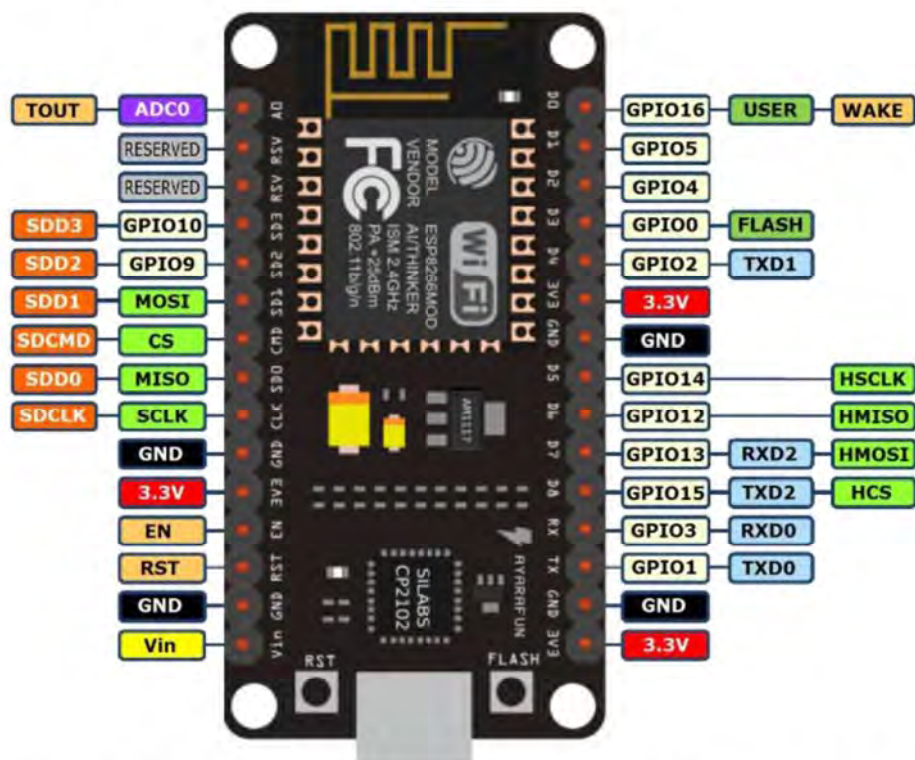


Рисунок 3.1 – Плата розробки ESP8266 12-E NodeMCU Kit

Індикатор рівня води LC1BD04

LC1BD04 – це цифровий індикатор рівня води з чотирма інтерфейсами для підключення електродів (рис. 3.2). Напруга живлення індикатора – від 3,3 до 5 вольт. Індикатор має 4 цифрові виходи, кожен з яких пов'язаний з відповідним інтерфейсом для підключення електродів. При наявності води між будь-якою парою електродів на відповідному цифровому виході формується логічний сигнал високого рівня (логічна «1»). В іншому випадку – на цифровому виході формується логічний сигнал низького рівня (логічний «0»).

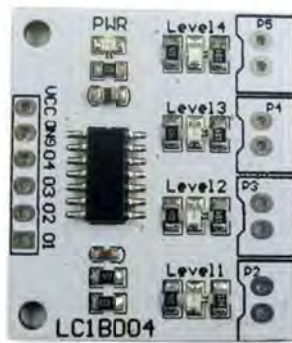


Рисунок 3.2 – Цифровий індикатор рівня води LC1BD04

Приклад використання індикатора наведений на рисунку 3.3.

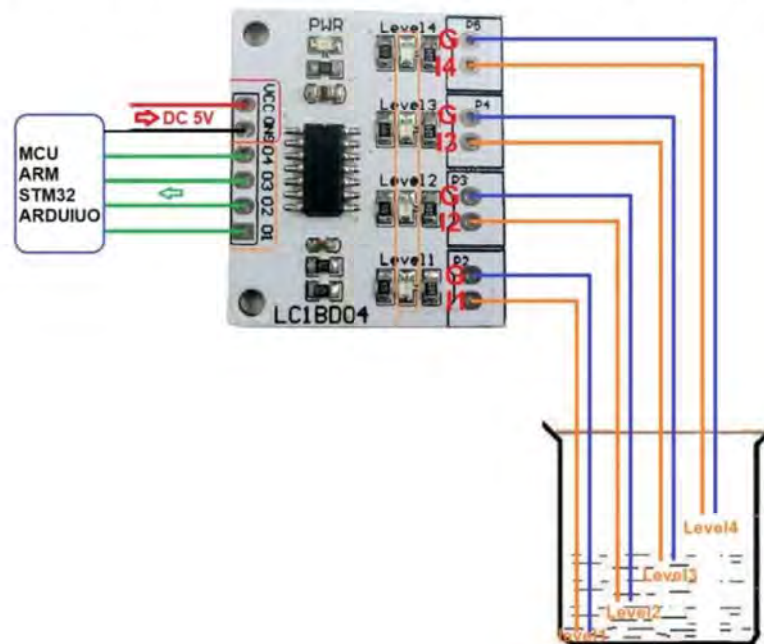


Рисунок 3.3 – Схема використання індикатор рівня води LC1BD04

Електронасос водяний

Електронасос призначений для примусового прокачування води по відповідних каналах (трубах). Він може бути будь-якої конструкції та з будь-якими характеристиками (напруга живлення, швидкість прокачування води тощо). В реальному проєкті використовуватиметься електронасос з напругою живлення 220 вольт змінного струму. Однак, для демонстраційного варіанту буде використано електронасос з напругою живлення 5 вольт постійного струму (рис. 3.4).



Рисунок 3.4 – Електронасос водяний (з живленням 5 вольт)

Модуль реле

Модуль реле використовується для керування високовольтними електричними пристроями за допомогою низьковольтних сигналів з мікроконтролерів (наприклад, ESP8266). Він складається безпосередньо з електромагнітного реле і необхідних елементів для його роботи: транзистора або оптопари, діода захисту та світлодіодів для індикації його роботи (рис. 3.4). В даному проєкті модуль реле використовуватиметься для вмикання та вимикання електронасоса.



Рисунок 3.4 – Модуль реле

OLED-дисплей

SSD1306 – це популярний завдяки своїм компактним розмірам, низькому енергоспоживанню і високій якості зображення малий дисплей, який використовується в різних електронних проєктах (рис. 3.5).



Рисунок 3.5 – OLED-дисплей SSD1306

Основні характеристики дисплея наведені в таблиці 3.2 [1]:

Таблиця 3.2 – Технічні характеристики дисплея SSD1306

Тип	OLED
Контролер	SSD1306
Розмір	0.96 дюйма
Роздільна здатність	128x32 пікселі
Інтерфейс підключення	I2C
Напруга живлення	Від 3.3 до 5 вольт

3.2 Реалізація принципової електричної схеми та монтаж пристрою

Розглянувши технічні характеристики та принципи роботи основних компонентів системи перейдемо до реалізації її електричної схеми.

Принципову електричну схему зручно реалізувати за допомогою програмного середовища Fritzing, яке призначене для проектування електронних схем та їх прототипів. Fritzing дозволяє створювати схеми з використанням віртуальних компонентів, макетувати друковані плати та візуалізувати з'єднання. Програма має простий інтерфейс, що підходить для новачків та освітніх цілей. Крім того, Fritzing підтримує експорт схем для подальшого виготовлення друкованих плат.

Принципова електрична схема системи представлена на рисунку 3.6, монтажна схема – на рисунку 3.7, а змонтований пристрій – на рисунку 3.8.

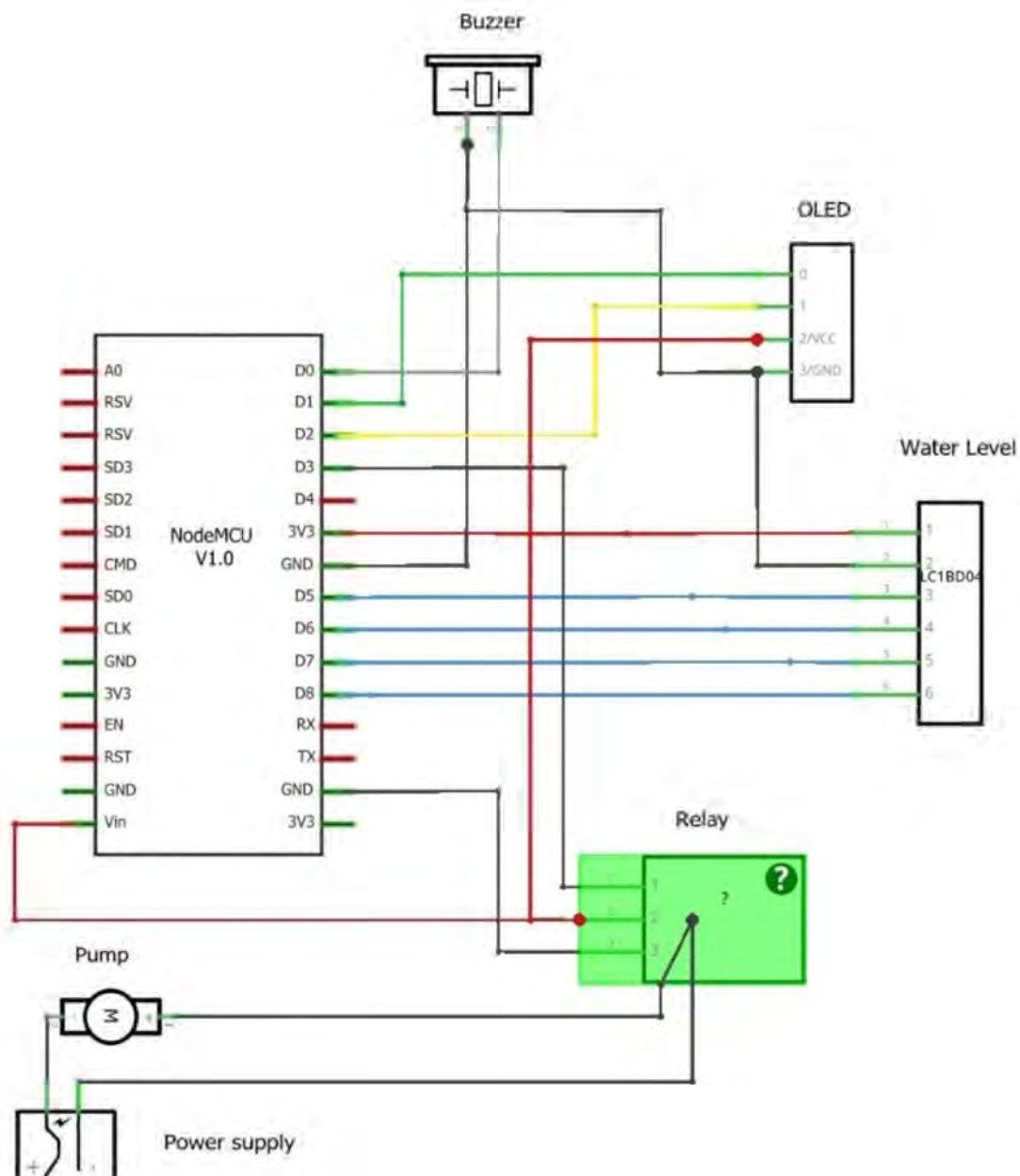


Рисунок 3.6 – Принципова електрична схема системи

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 24.534.07.000 ПЗ

Арк.

26

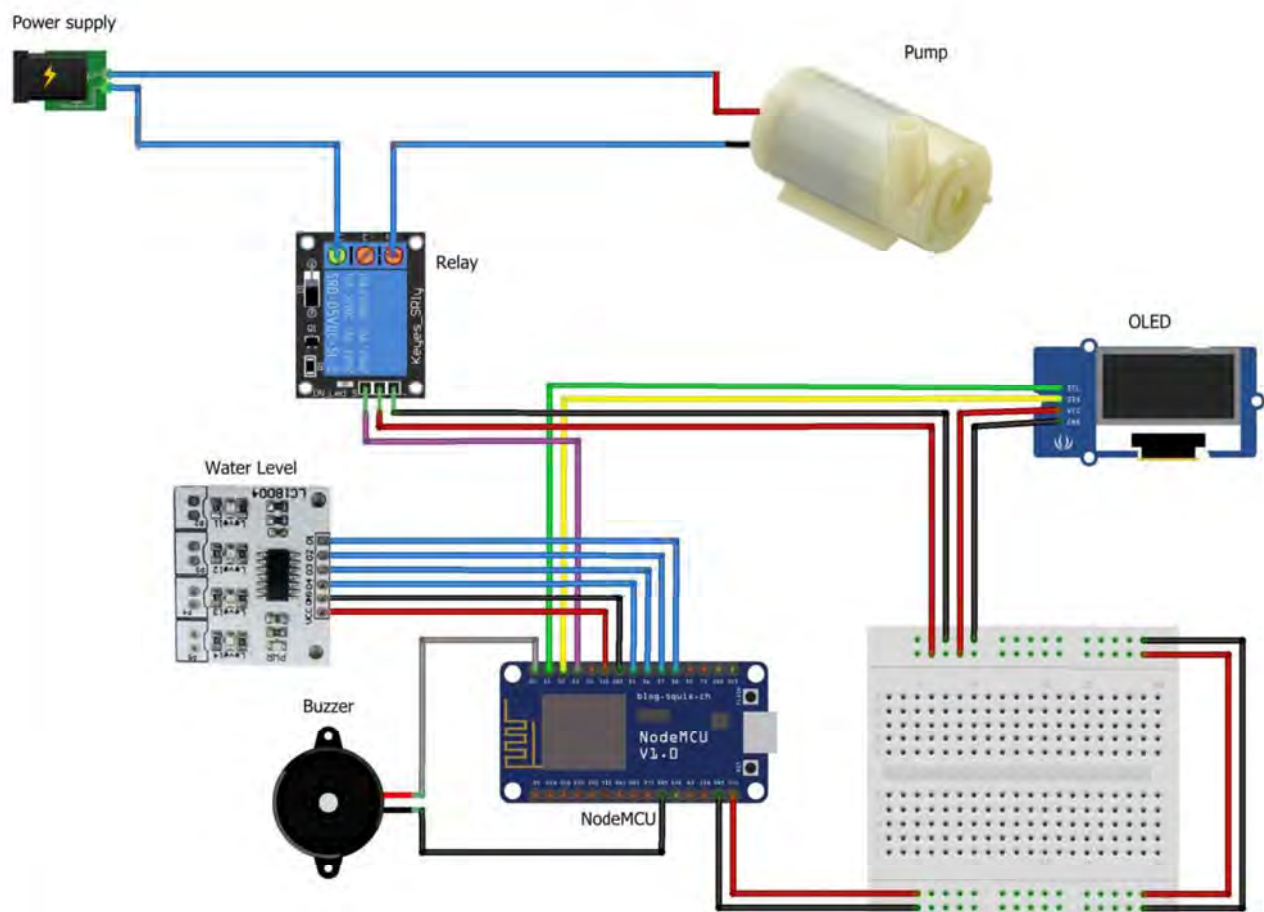


Рисунок 3.7 – Монтажна схема системи

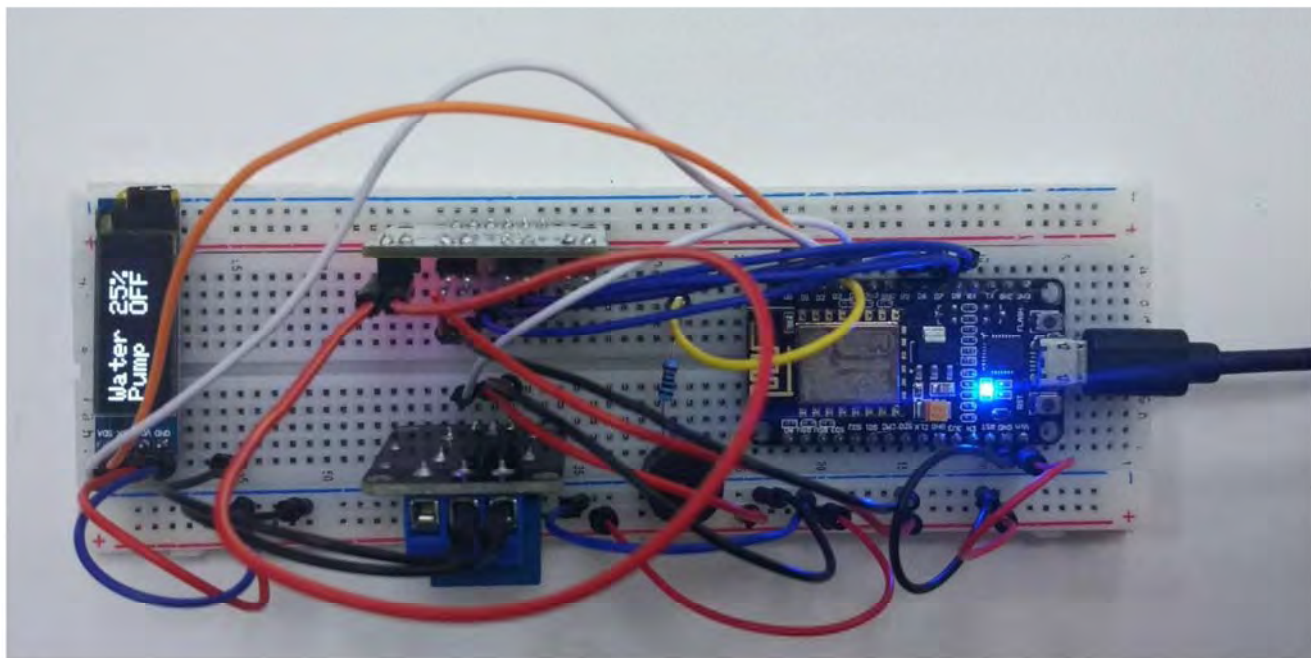


Рисунок 3.8 – Змонтований пристрій на макетній платі

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 24.534.07.000 ПЗ

Арк.

27

3.3 Реалізація програмного коду

Відповідно до спроектованого алгоритму роботи системи (рис. 2.2) реалізуємо програмний код роботи системи.

Спершу підключимо необхідні бібліотеки, які нам знадобляться для роботи OLED-дисплея. У разі їх відсутності спершу встановимо їх через менеджер бібліотек Arduino IDE.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

Далі визначимо піни мікроконтролера ESP8266, які будуть використані в проєктів та зазначимо для чого саме. Слід зауважити, що нумерація пінів на платі розробки ESP8266 12-E NodeMCU відрізняється від нумерації пінів самого мікроконтролера ESP8266 (рис. 3.1).

```
#define WATER_LEVEL_PIN_1 14 // Пін 1-го виводу датчика рівня води (D5)
#define WATER_LEVEL_PIN_2 12 // Пін 2-го виводу датчика рівня води (D6)
#define WATER_LEVEL_PIN_3 13 // Пін 3-го виводу датчика рівня води (D7)
#define WATER_LEVEL_PIN_4 15 // Пін 4-го виводу датчика рівня води (D8)
#define RELAY_PIN 0 // Пін управління реле (D3)
#define BUZZER_PIN 16 // Пін підключення бузера (D0)
```

Встановимо параметри OLED-дисплея SSD1306. Згідно до визначених характеристик (табл. 3.2) його роздільна здатність становить 128*32 пікселі.

```
#define SCREEN_WIDTH 128 // Ширина дисплею в пікселях
#define SCREEN_HEIGHT 32 // Висота дисплею в пікселях
#define OLED_RESET -1 // Пін скидання OLED-дисплея
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
```

Після цього ініціалізуємо змінні, які використовуватимемо у проєкті, та встановимо їм початкові значення.

```
int waterLevel = 0; // Змінна для зберігання поточного рівня води
int lastWaterLevel = -1; // Змінна для зберігання попереднього рівня
води
bool pumpState = false; // Змінна для зберігання стану насоса
```

Тепер в функції setup() ініціалізуємо параметри роботи Serial Monitor, ініціалізуємо OLED-дисплей та задамо режими роботи обраних для проєкту пінів мікроконтролера.

					КР.КІ 24.534.07.000 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

```

void setup() {
    // Ініціалізація SerialMonitor на швидкості 9600
    Serial.begin(9600);

    // Ініціалізація OLED-дисплея
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println(F("SSD1306 allocation failed"));
        for (;;)
    }

    // Встановлення режимів роботи пінів
    pinMode(WATER_LEVEL_PIN_1, INPUT);
    pinMode(WATER_LEVEL_PIN_2, INPUT);
    pinMode(WATER_LEVEL_PIN_3, INPUT);
    pinMode(WATER_LEVEL_PIN_4, INPUT);
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);
}

```

Переходимо до функції loop(), яка циклічно виконуватиметься увесь час, поки увімкнуте живлення системи. У ній нам необхідно постійно визначати поточні значення пінів датчика рівня води, на їх основі обчислювати рівень води в резервуарі, в залежності від обчисленого рівня вмикати та вимикати насос і генерувати звукові сигнали на бужері. Окрім цього, обчислений рівень води в резервуарі та поточний стан електронасоса (включений чи виключений) виводитимемо на OLED-дисплей. Після виконання кожної ітерації циклу функції loop() робитимемо затримку тривалістю 1 секунда для зменшення навантаження на мікроконтролер.

```

void loop() {
    // Отримання значень пінів датчика рівня води
    int level1 = digitalRead(WATER_LEVEL_PIN_1);
    int level2 = digitalRead(WATER_LEVEL_PIN_2);
    int level3 = digitalRead(WATER_LEVEL_PIN_3);
    int level4 = digitalRead(WATER_LEVEL_PIN_4);

    // Обчислення поточного рівня води
    int currentWaterLevel = level1*25 + level2*25 + level3*25 + level4*25;

    // Визначення стану насоса залежно від рівня води
    if (currentWaterLevel < 25) {
        // Увімкнення насоса
        digitalWrite(RELAY_PIN, HIGH);
        pumpState = true;
        // Бужер видає 3 короткі сигнали на частоті 1000Гц
        for (int i = 0; i < 3; i++) {
            tone(BUZZER_PIN, 1000);
            delay(200);
            noTone(BUZZER_PIN);
            delay(200);
        }
    }
}

```



```

    }
} else if (currentWaterLevel >= 100) {
    // Вимкнення насоса
    digitalWrite(RELAY_PIN, LOW);
    pumpState = false;
}

// Перевірка, чи змінився рівень води
if (currentWaterLevel != lastWaterLevel) {
    waterLevel = currentWaterLevel;
    lastWaterLevel = currentWaterLevel;

    // Виведення поточного рівня води та стану насоса на OLED-дисплей
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.print("Water ");
    display.print(waterLevel);
    display.println("%");
    display.print("Pump ");
    display.println(pumpState ? "ON" : "OFF");
    display.display();

    // Виведення поточного рівня води та стану насоса в Serial Monitor
    Serial.print("Water Level: ");
    Serial.print(waterLevel);
    Serial.print("% | ");
    Serial.print("Pump State: ");
    Serial.println(pumpState ? "ON" : "OFF");
}

// Затримка 1 сек. перед наступною ітерацією
delay(1000);
}

```

3.4 Підключення та налаштування сервісу thinger.io

Опис сервісу thinger.io

Згідно поставленої до проєкту задачі створювана система повинна забезпечити можливість віддаленого моніторингу та управління системою через Інтернет, що дозволить користувачам отримувати доступ до даних про рівень води з будь-якого місця та в будь-який час. Для реалізації даної вимоги скористаємось онлайн-сервісом thinger.io.

Thingier.io – це платформа для Інтернету речей, що надає інструменти для підключення пристроїв до Інтернету, моніторингу та керування ними через вебінтерфейс або мобільний застосунок.

					<i>КР.КІ 24.534.07.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

Основні можливості Thinger.io:

- підключення пристроїв: платформа дозволяє легко підключати різноманітні IoT-пристрої через стандартні протоколи;
- моніторинг в реальному часі: користувачі можуть переглядати дані з датчиків та стан пристроїв у реальному часі через зручний інтерфейс;
- керування пристроями: дає змогу дистанційно керувати підключеними пристроями, такими як насоси, освітлення тощо;
- аналітика та звіти: надає інструменти для зберігання та аналізу даних, що допомагає приймати обґрунтовані рішення;
- інтеграція з іншими сервісами: легко інтегрується з іншими вебсервісами та застосунками для розширення функціоналу.

Thinger.io спрощує процес створення IoT-проектів, надаючи готову інфраструктуру для підключення, моніторингу та управління пристроями, що економить час та ресурси розробників.

Реєстрація облікового запису та пристрою в сервісі Thinger.io

Процес реєстрації в сервісі дуже простий та займає буквально кілька хвилин. Для його здійснення необхідно вказати свій активний емейл, придумати собі унікальне ім'я користувача та сформулювати пароль (рис. 3.9).

					<i>КР.КІ 24.534.07.000 ПЗ</i>	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

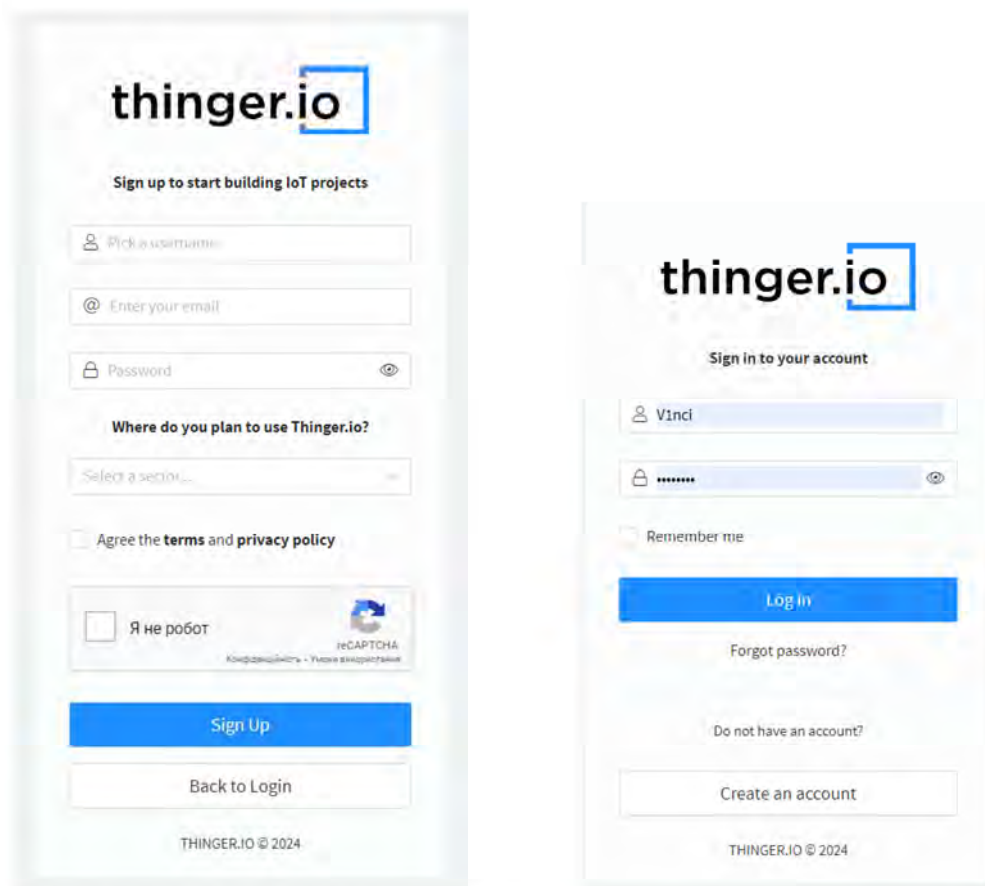


Рисунок 3.9 – Створення облікового запису та авторизація в сервісі Thinger.io

Після успішної реєстрації та авторизації в сервісі користувач потрапляє на сторінку Statistics (рис. 3.10).



Рисунок 3.10 – Стартова сторінка сервісу після авторизації

Для даного проєкту нам знадобиться створити один пристрій. Для цього потрібно перейти на сторінку Devices, натиснути кнопку Create a Device, заповнити необхідні параметри та натиснути кнопку Add Device (рис. 3.11).

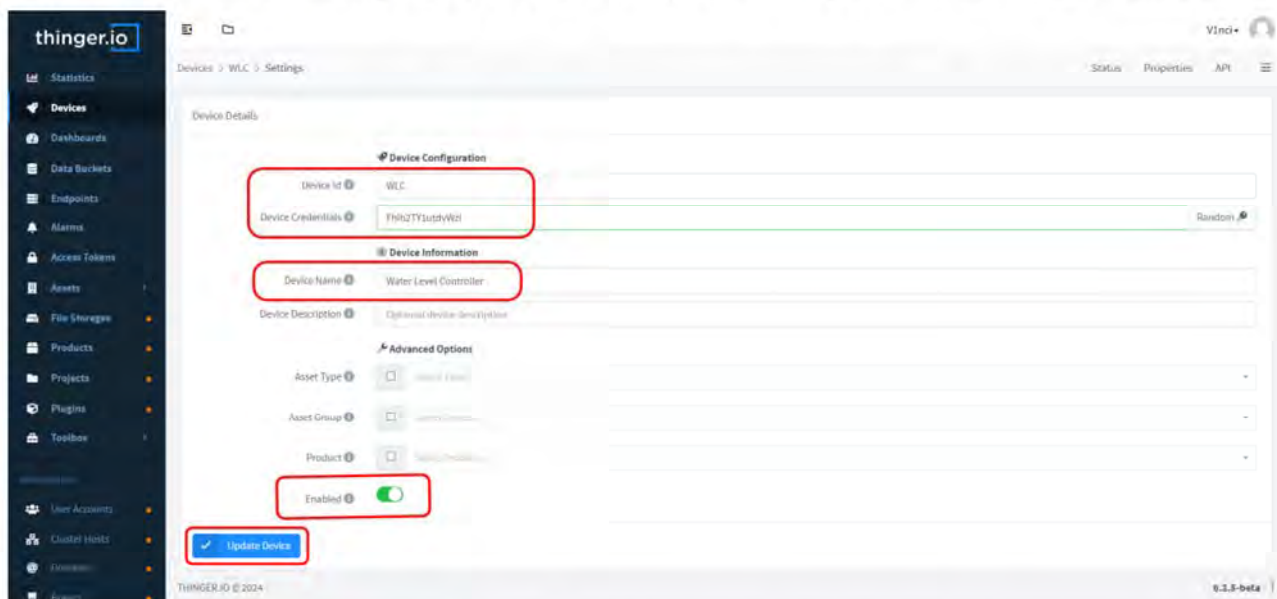


Рисунок 3.11 – Створення нового пристрою

Адаптація програмного коду для взаємодії з сервісом Thinger.io

Отримавши значення параметрів Device Id та Device Credentials, а також маючи ім'я користувача, адаптуємо програмний код системи для роботи з сервісом Thinger.io.

Спершу підключаємо додаткові бібліотеки, які знадобляться нам для роботи з сервісом.

```
#include <ESP8266WiFi.h>
#include <ThingerESP8266.h>
```

Визначимо змінні для підключення до мережі WiFi

```
char ssid[] = "iot_21"; // Назва Wi-Fi мережі
char password[] = "smarthome"; // Пароль Wi-Fi мережі
ThingerESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
```

Задамо параметри підключення до сервісу Thinger.io.

```
#define USERNAME "Vlnci" // Ім'я користувача
#define DEVICE_ID "WLC" // Ідентифікатор пристрою
#define DEVICE_CREDENTIAL "Fh1h2TYlutyWzi" // Ключ пристрою
```


В функції `setup()` створимо два ресурси (Resources) для використання в сервісі: `water_level` – для передачі в сервіс поточного значення рівня води в резервуарі, та `pump_state` – для передачі поточного стану електронасоса.

```
thing["water_level"] >> outputValue(waterLevel);
thing["pump_state"] >> outputValue(pumpState);
```

Тепер нам залишилось реалізувати в функції `loop()` надсилення значень ресурсів `water_level` та `pump_state` в Thingier.io.

```
waterLevel = currentWaterLevel;
lastWaterLevel = currentWaterLevel;
thing.handle();
```

Повна версія оновленого лістингу програмного коду наведена в додатку Б.

Після оновлення програмного коду та його завантаження в мікроконтролер підключаємо пристрій до WiFi та перевіряємо його роботу в Thingier.io. Якщо все зроблено правильно, то створений пристрій повинен відображатися з зеленим кружечком, який свідчить, що пристрій успішно підключився (рис. 3.12).

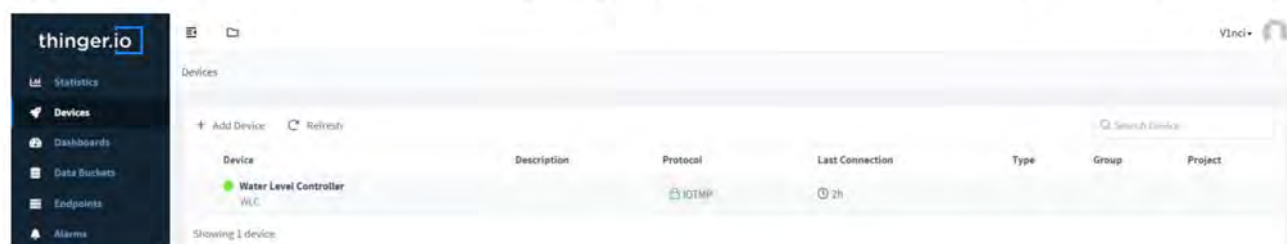


Рисунок 3.12 – Пристрій активний після успішного підключення

Створення та налаштування панелі керування в сервісі Thingier.io

Тепер переходимо до створення панелі керування (Dashboard). На сторінці Dashboards натискаємо кнопку Create Dashboard, заповнюємо необхідні параметри та натискаємо кнопку Add Dashboard (рис. 3.13).

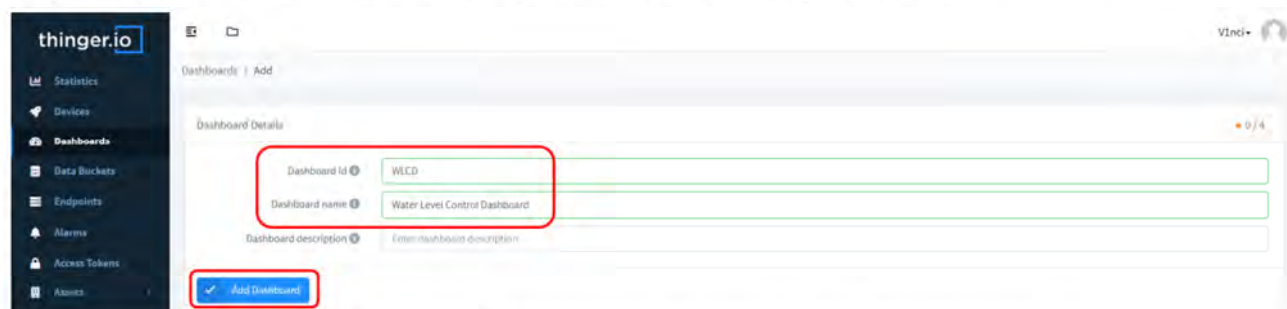


Рисунок 3.13 – Створення нової панелі керування

Після цього потрібно наповнити панель керування відповідними віджетами.

Додамо на панель керування 3 наступні віджети:

1. Віджет, який відображає поточний стан насоса (ввімкнути чи вимкнутий).

Для цього додамо віджет типу Led Indicator та встановимо йому наступні значення параметрів (рис. 3.14):

- Title: Pump State;
- Type: Led Indicator;
- Data Source: From Device Source;
- Device: Water Level Controller;
- Resource Name: pump_state;
- Refresh Mode: Sampling Interval = 1 second;
- Led Size: 100px;
- Colors: Red (вимкнутий) = 0, Green (увімкнутий) = 1.

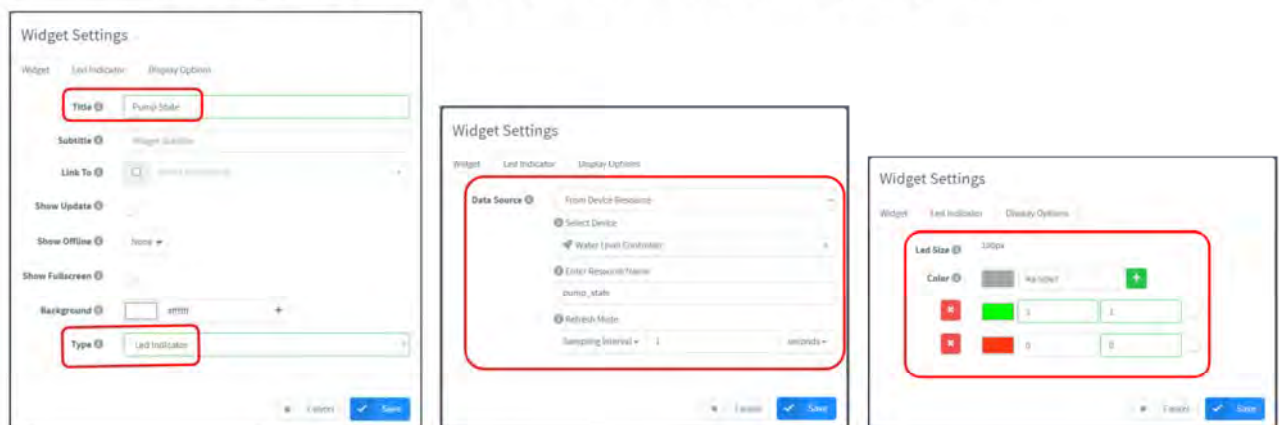


Рисунок 3.14 – Налаштування віджета стану насоса

2. Віджет, який відображає поточний рівень наповненості водяного резервуара.

Для цього додамо віджет типу Donut Chart та встановимо йому наступні значення параметрів (рис. 3.15):

- Title: Current Water Level;
- Type: Donut Chart;

- Data Source: From Device Source;
- Device: Water Level Controller;
- Resource Name: water_level;
- Refresh Mode: Sampling Interval = 1 second;

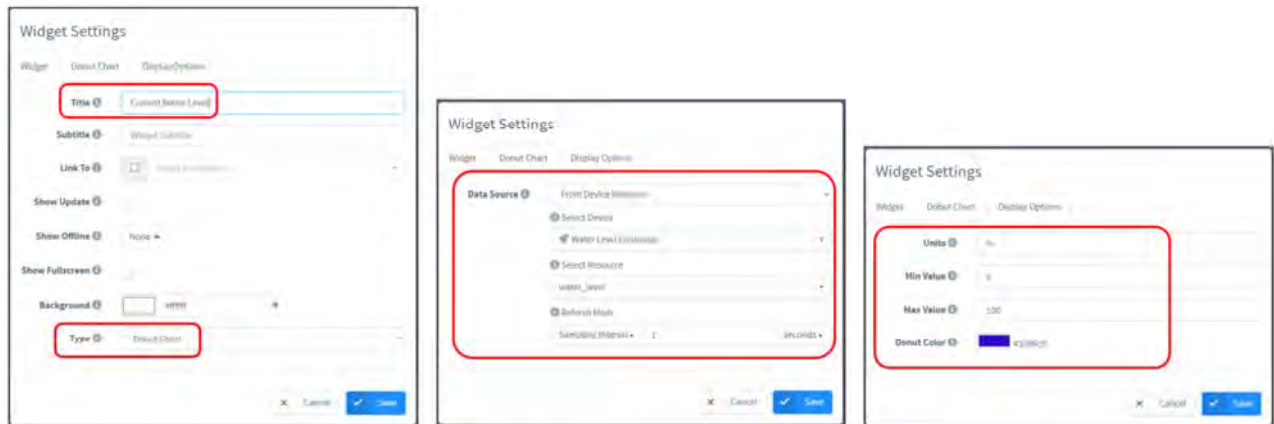


Рисунок 3.15 – Налаштування віджета рівня води

3. Віджет, який відображає динаміку споживання води з резервуара.

Це останній з запланованих віджетів. Його функція – показувати на графіку хронологію споживання води. Для його реалізації додамо віджет типу Time Series Chart та встановимо йому наступні значення параметрів (рис. 3.16):

- Title: Water Level Diagram;
- Type: Time Series Chart;
- Data Source: From Device Source;
- Device: Water Level Controller;
- Source: water_level;
- Refresh Mode: Sampling Interval = 5 second;
- Display Options: Axis, X Axis = 1, Y Axis = 0 ... 100 ticks.

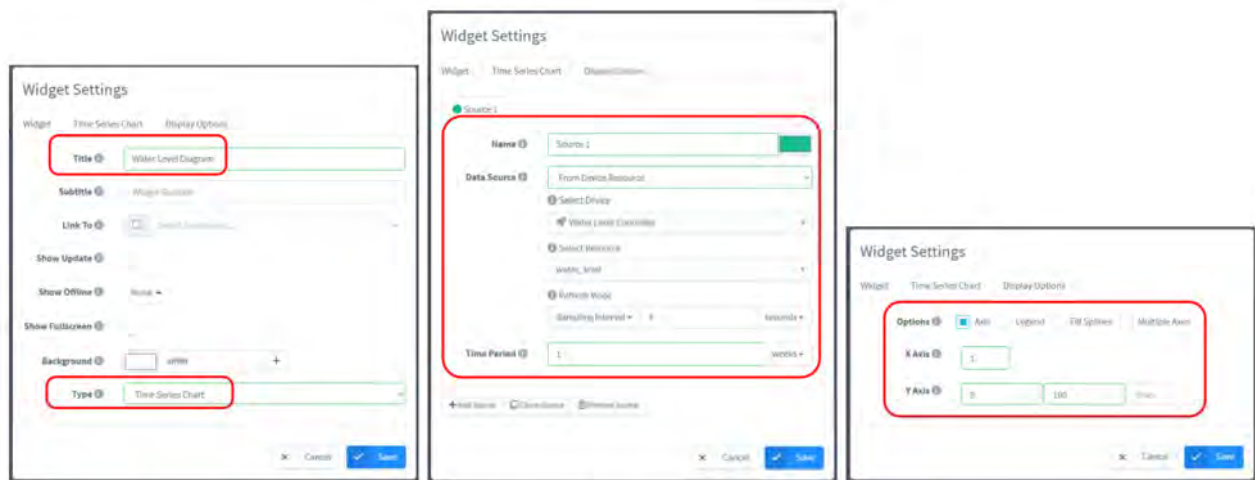


Рисунок 3.16 – Налаштування віджета динаміки споживання води

Після встановлення та налаштування усіх віджетів панель керування виглядає як показано на рисунку 3.17, а стартова сторінка сервісу після здійснення всіх налаштувань – на рисунку 3.18.

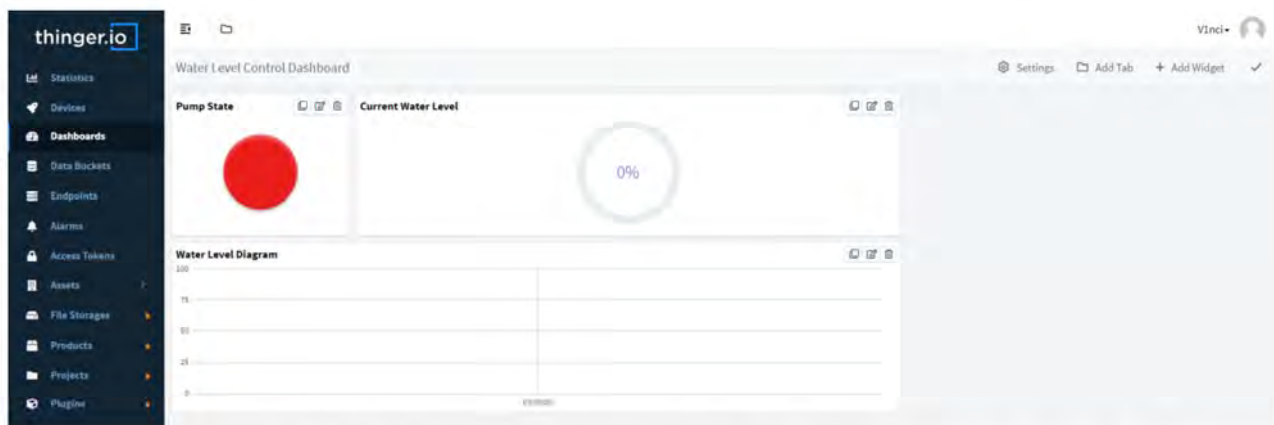


Рисунок 3.17 – Панель керування після встановлення віджетів



Рисунок 3.18 – Стартова сторінка сервісу після здійснення всіх налаштувань

3.5 Тестування роботи системи

Для проведення тестування складемо план тестування та перевіримо роботу системи відповідно до нього.

План тестування:

1. Перевірити підключення пристрою до мережі WiFi.
2. Перевірити підключення пристрою до сервісу Thinger.io.
3. Перевірити відображення рівня води в резервуарі та стану роботи насоса на OLED-дисплеї.
4. Перевірити увімкнення насоса при рівні води в резервуарі менше 25 відсотків.
5. Перевірити увімкнення звукового сповіщення при рівні води в резервуарі менше 25 відсотків.
6. Перевірити вимкнення насоса при рівні води в резервуарі 100 відсотків.
7. Перевірити відображення рівня води в резервуарі, стану роботи насоса та статистики використання води на панелі керування в Thinger.io.

Результати тестування відображені в таблиці 3.3 та на рисунку 3.19.

Таблиця 3.3 – Результати тестування роботи системи

№	Дія	Очікуваний результат	Фактичний результат
1.	Підключення пристрою до мережі WiFi	Пристрій підключився після увімкнення живлення	Пристрій підключився після увімкнення живлення
2.	Підключення пристрою до сервісу Thinger.io	Пристрій підключився до сервісу	Пристрій підключився до сервісу
3.	Відображення рівня води в резервуарі на OLED-дисплеї	Коректний рівень води відображається на OLED-дисплеї	Коректний рівень води відображається на OLED-дисплеї
4.	Відображення стану роботи насоса на OLED-дисплеї	Стан роботи насоса відображається на OLED-дисплеї	Стан роботи насоса відображається на OLED-дисплеї
5.	Увімкнення насоса при рівні води в резервуарі менше 25%	При рівні води в резервуарі менше 25% насос увімкнувся	При рівні води в резервуарі менше 25% насос увімкнувся

6.	Звукове сповіщення при рівні води в резервуарі менше 25%	При рівні води в резервуарі менше 25% лунає звуковий сигнал	При рівні води в резервуарі менше 25% лунає звуковий сигнал
7.	Вимкнення насоса при рівні води в резервуарі вище 100%	При рівні води в резервуарі вище 100% насос вимкнувся	При рівні води в резервуарі вище 100% насос вимкнувся
8.	Відображення стану роботи насоса на панелі керування Thinger.io	Стан роботи насоса коректно відображається на панелі керування	Стан роботи насоса коректно відображається на панелі керування
9.	Відображення рівня води в резервуарі на панелі керування Thinger.io	Рівень води в резервуарі коректно відображається на панелі керування	Рівень води в резервуарі коректно відображається на панелі керування
10.	Відображення статистики використання води на панелі керування Thinger.io	Статистика використання води відображається на графіку панелі керування	Статистика використання води відображається на графіку панелі керування

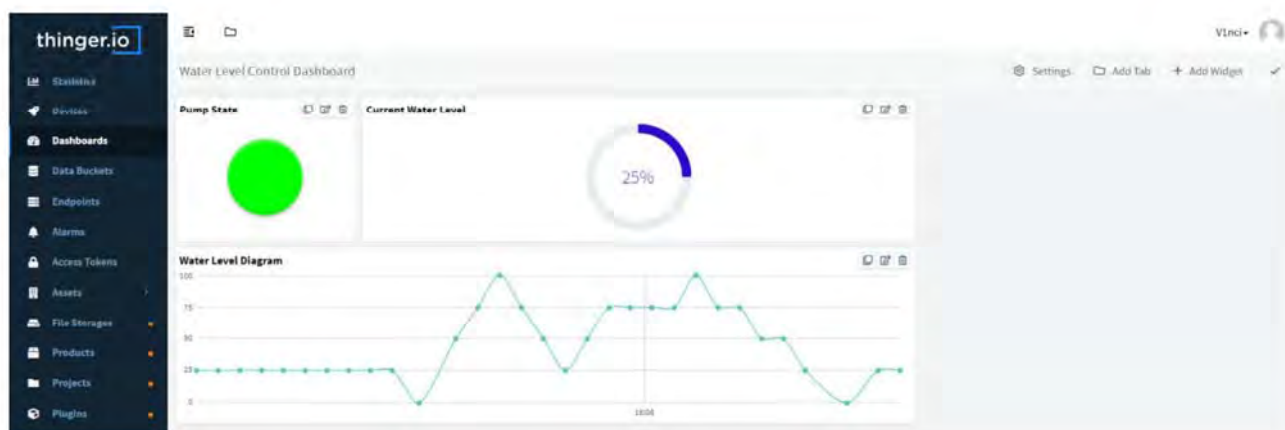


Рисунок 3.19 – Панель керування після встановлення віджетів

В процесі тестування роботи системи не було виявлено жодних відхилень в її роботі. Система працює відповідно до поставленого завдання.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ПРОЄКТУ

4.1 Аналіз ринку

Системи контролю рівня наповненості водяних резервуарів є важливою складовою для багатьох галузей, включаючи сільське господарство, промисловість, житлово-комунальне господарство та інші. Зокрема, такі системи необхідні для забезпечення ефективного управління водними ресурсами, запобігання переливу або пересихання резервуарів, а також для забезпечення надійного постачання води.

Основні сегменти ринку, які потребують таких систем, включають:

- Приватні будинки та котеджі: Власники приватних будинків з автономними системами водопостачання.
- Фермерські господарства: Використання для контролю рівня води у резервуарах для поливу та водопою тварин.
- Готелі та гостьові будинки: Забезпечення безперебійного водопостачання та автоматичний контроль за рівнем води в резервуарах.
- Малі підприємства: Використання в системах охолодження, виробничих процесах, де необхідний контроль рівня води.

Як було відмічено в питанні 1.2, сучасний ринок пропонує широкий спектр рішень для контролю рівня води: від простих механічних датчиків до складних автоматизованих систем з дистанційним управлінням.

Загалом, можна виділити два тренди в таких рішеннях:

1. Використання Інтернету речей для підключення систем до мережі Інтернет, що дозволяє здійснювати моніторинг та управління такими системами в режимі реального часу.

2. Зростання попиту на автономні та енергоефективні рішення, які можуть працювати з використанням сонячної енергії або інших альтернативних джерел енергії.

Основні компанії, які пропонують такі рішення, включають такі бренди, як Siemens, Honeywell і Grundfos, що пропонують високонадійні рішення для промислових та побутових потреб.

Ціновий діапазон для систем контролю рівня води в резервуарі може значно варіюватися в залежності від складності та функціональних можливостей. Нижче наведено приблизні діапазони цін для різних сегментів ринку.

1. Прості датчики, які можуть інтегруватися в існуючі рішення:

- Механічні поплавкові датчики: від 100 до 500 грн.
- Ультразвукові датчики рівня: від 800 до 2000 грн.
- Ємнісні та тензOMETричні датчики: від 500 до 1500 грн.

2. Складні системні рішення з використанням Інтернету речей (IoT):

- IoT-рішення для контролю рівня води: від 5000 до 10000 грн.
- Комплексні системи моніторингу та управління: від 10000 до 30000 грн і більше.

4.2 Розрахунок витрат на реалізацію

Для оцінки вартості реалізації системи контролю рівня наповненості водяного резервуару необхідно вирахувати витрати на компоненти та витрати на оплату праці.

Проведемо детальний розрахунок витрат на матеріали.

Як було зазначено раніше, система складається з таких компонентів:

1. Мікроконтролер ESP8266-E12.
2. Індикатор рівня води LC1BD04.
3. Електронасос водяний.
4. Модуль реле.
5. OLED-дисплей SSD1306.
6. Бузер.

Тут варто відмітити, що створювана система (як і наявні рішення, що представлені на ринку) не передбачає придбання водяного насоса, з'єднувальних електропроводів та водяних резервуарів, оскільки це все, як правило, уже є в наявності і функціонує в ручному режимі. Створювана система необхідна виключно для заміни ручного режиму на автоматизований чи автоматичний. Саме тому в розрахунки вартості витрат на матеріали будуть включені лише дійсно використані в системі компоненти (табл. 4.1). Для визначення вартості зазначених компонентів було обчислено середні значення цін з кількох онлайн магазинів.

Таблиця 4.1 – Вартість компонентів системи

№	Назва компонента	К-сть, шт.	Ціна за одиницю, грн.
1	Мікроконтролер ESP8266-E12	1	250
2	Індикатор рівня води LC1BD04	1	150
3	Модуль реле	1	50
4	OLED-дисплей SSD1306	1	130
5	Бuzzer	1	20
	Разом		600

Тепер здійснимо розрахунок витрат на оплату праці.

Спершу визначимо перелік осіб, задіяних в розробці системи. В ідеальному випадку команда розробників може складатись з трьох осіб: проєктувальника, інженера-програміста та інженера з налаштування системи. Однак, на практиці усі ці функції може виконувати інженер-програміст. Для зменшення витрат на розробку системи розрахуємо витрати на оплату праці одного працівника – інженера-програміста.

Середня місячна заробітна плата інженера-програміста становить близько 30 000грн. Тривалість розробки системи не перевищує одного місяця. Вирахуємо кошторис витрат з врахуванням зазначених вихідних даних та дійсних податкових відрахувань з заробітної плати.

					КР.КІ 24.534.07.000 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

- Податок на доходи фізичних осіб (18%):
 $30000 \text{ грн.} * 18\% = 5400 \text{ грн.}$
- Військовий збір (1.5%):
 $30000 \text{ грн.} * 1.5\% = 450 \text{ грн.}$
- Єдиний внесок (22%):
 $30000 \text{ грн.} * 22\% = 6600 \text{ грн.}$
- Загальні відрахування:
 $5400 \text{ грн.} + 450 \text{ грн.} = 5850 \text{ грн.}$
- Чиста зарплата:
 $30000 \text{ грн.} - 5850 \text{ грн.} = 24150 \text{ грн.}$

Загальна вартість реалізації системи розраховуватиметься як сума витрат на придбання компонентів системи та витрат на заробітну плату інженера-програміста:

$$30000 \text{ грн.} + 600 \text{ грн.} = 30600 \text{ грн.}$$

4.3 Обґрунтування доцільності розробки

Розробка власної системи контролю рівня наповненості водяного резервуару має декілька ключових переваг, незважаючи на схожість вартості з ринковими аналогами:

1. Адаптація до специфічних потреб.

Розробка власної системи дозволяє створити рішення, яке повністю відповідає конкретним потребам та вимогам користувача, що може бути складно досягти з комерційними продуктами.

2. Гнучкість та можливість модернізації.

Власна система може бути легко змінена та модернізована відповідно до нових вимог або змін у технологіях, без необхідності повної заміни обладнання.

3. Зниження довгострокових витрат.

Хоча початкові витрати на розробку можуть бути високими, в довгостроковій перспективі це може знизити витрати на підтримку та

					КР.КІ 24.534.07.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

модернізацію системи, порівняно з комерційними продуктами, що потребують регулярних оновлень та ліцензійних платежів.

Таким чином, розробка власної системи контролю рівня наповненості водяного резервуару є обґрунтованою інвестицією, що може забезпечити більшу гнучкість, адаптивність та довгострокову економічну ефективність у порівнянні з комерційними аналогами.

					<i>КР.КІ 24.534.07.000 ПЗ</i>	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У даній роботі було досліджено, спроектовано, реалізовано та протестовано систему контролю рівня води на основі мікроконтролера ESP8266 та сервісу Thingier.io.

В процесі аналізу предметної області було визначено, що контроль рівня води є важливим аспектом у багатьох галузях, включаючи сільське господарство, промисловість та побутові системи. Метою роботи було визначено створення автоматизованої системи, яка здатна відстежувати рівень води в резервуарі та керувати насосом для підтримки оптимального рівня.

На етапі проектування було розроблено архітектуру системи, що включає датчики рівня води, модуль керування насосом, дисплей для локального відображення інформації та ESP8266 для обробки даних і зв'язку з сервісом Thingier.io. Було вибрано апаратні компоненти, які забезпечують необхідні функціональні можливості. Система була спроектована таким чином, щоб надати можливість віддаленого моніторингу та керування через Інтернет.

На етапі реалізації було розроблено програмне забезпечення для мікроконтролера ESP8266. Інтеграція з Thingier.io дозволила забезпечити віддалений моніторинг та керування системою. Після завершення розробки було проведено тестування системи, результати якого показали, що система працює стабільно, точно вимірює рівень води та своєчасно керує насосом.

Економічний аналіз показав, що розробка та впровадження системи контролю рівня води є економічно доцільною. Використання доступних апаратних компонентів дозволило значно знизити витрати на розробку.

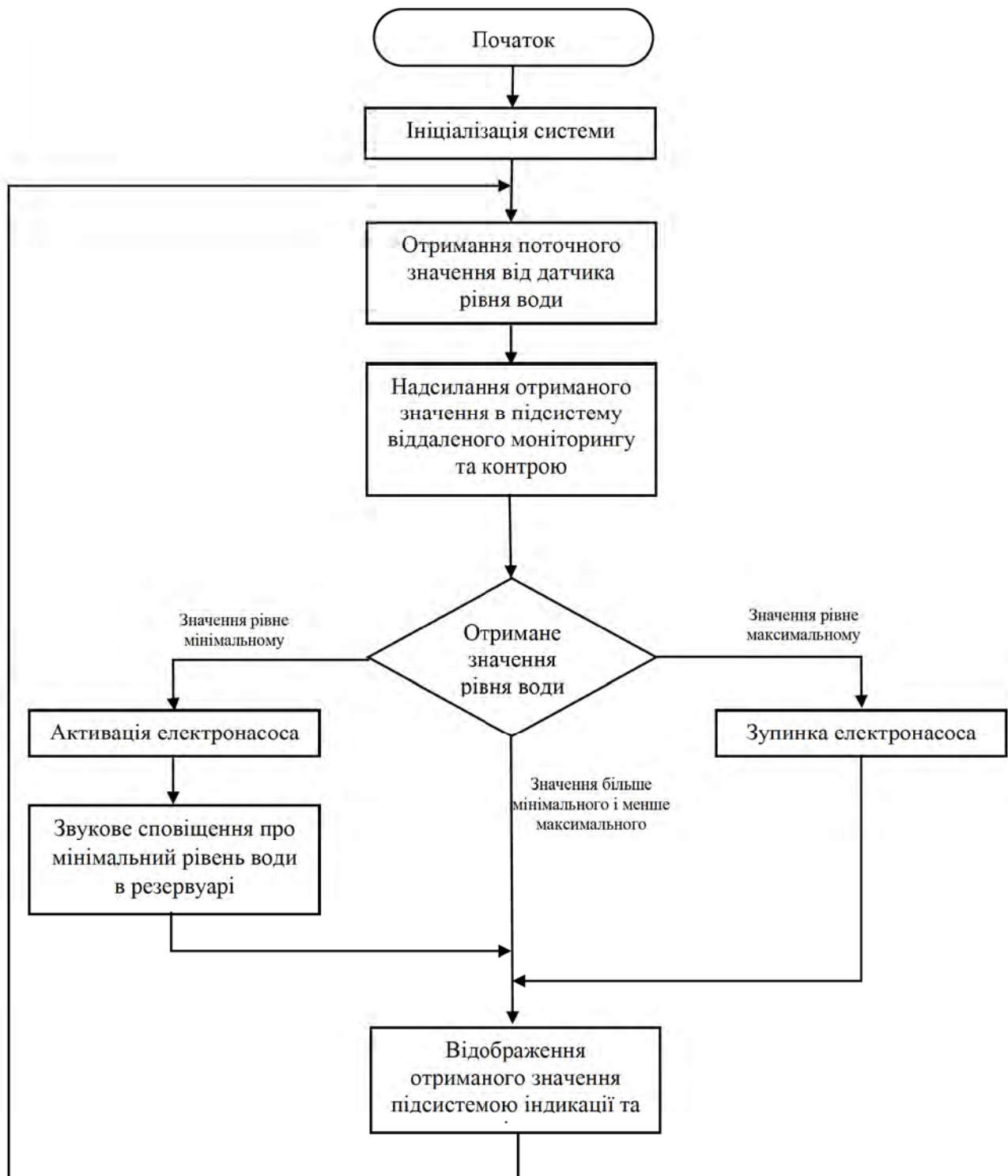
У подальшому систему можна вдосконалити шляхом інтеграції з наявними системами розумного дому. Також, за потреби, можна розглянути ідею створення мобільного застосунку для зручнішого керування системою.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. OLED-дисплей SSD1306: веб-сайт. URL: <https://arduino.ua/prod1795-oled-displei-modyl-0-91-i2c-128x32-beli> (дата звернення: 10.03.2024).
2. Датчики контролю рівня води Lovato: веб-сайт. URL: https://www.svaltera.ua/catalog/rele_kontrolyu_rivnya_ridini_ta_prioritetu_v_klyuchennya_lovato_electric (дата звернення: 11.04.2024).
3. Датчики рівня води: веб-сайт. URL: [https://ukrayinska.libretexts.org/Інженерна/Промислове_та_системного_машинобудування/Книга%3А_Динаміка_та_контроль_хімічних_процесів_\(Woolf\)/03%3А_Датчики_та_в_иконавчі_механізми/3.04%3А_Датчики_рівня](https://ukrayinska.libretexts.org/Інженерна/Промислове_та_системного_машинобудування/Книга%3А_Динаміка_та_контроль_хімічних_процесів_(Woolf)/03%3А_Датчики_та_в_иконавчі_механізми/3.04%3А_Датчики_рівня) (дата звернення: 10.04.2024).
4. Методичні рекомендації до виконання кваліфікаційної роботи для студентів спеціальності 123 «Комп'ютерна інженерія» освітньої програми «Інженерія Інтернету речей» / Павлюс В.П., Посвятовська О.Б., Кульчинська Н.З. – Галицький фаховий коледж імені В'ячеслава Чорновола, Тернопіль, 2023. 52с.
5. Мікроконтролер ESP8266: веб-сайт. URL: <https://uk.wikipedia.org/wiki/ESP8266> (дата звернення: 12.04.2024).
6. Налаштовуємо NodeMCU V3 Arduino IDE: веб-сайт. URL: <https://oxorona.com/nodemcu-v3-arduino> (дата звернення: 05.05.2024).

ДОДАТКИ

Додаток А. Блок-схема алгоритму роботи системи



Додаток Б. Лістинг програмного коду

```
#include <ESP8266WiFi.h>
#include <ThingrESP8266.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

char ssid[] = "iot_21";           // Назва Wi-Fi мережі
char password[] = "smarthome";    // Пароль Wi-Fi мережі

#define USERNAME "Vlnci"           // Ім'я користувача
thinger.io
#define DEVICE_ID "WLC"            // Ідентифікатор
пристрою thinger.io
#define DEVICE_CREDENTIAL "Fh1h2TY1utdvWzi" // Ключ пристрою
thinger.io

#define WATER_LEVEL_PIN_1 14       // Пін для першого виводу датчика
рівня води (D5)
#define WATER_LEVEL_PIN_2 12       // Пін для другого виводу датчика
рівня води (D6)
#define WATER_LEVEL_PIN_3 13       // Пін для третього виводу датчика
рівня води (D7)
#define WATER_LEVEL_PIN_4 15       // Пін для четвертого виводу датчика
рівня води (D8)
#define RELAY_PIN 0                // Пін для управління реле
(D3)
#define BUZZER_PIN 16              // Пін для підключення бузера
(D0)

#define SCREEN_WIDTH 128           // Ширина дисплею в пікселях
#define SCREEN_HEIGHT 32           // Висота дисплею в пікселях
#define OLED_RESET -1              // Пін скидання OLED дисплею, -1, якщо
не використовується

int waterLevel = 0;                // Змінна для зберігання поточного рівня
води
int lastWaterLevel = -1;           // Змінна для зберігання попереднього
рівня води
bool pumpState = false;           // Змінна для зберігання стану насоса
(увімкнено/вимкнено)

ThingrESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

void setup() {
```

```

Serial.begin(9600); // Ініціалізація SerialMonitor на швидкості
9600

// Ініціалізація OLED дисплея
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
}

pinMode(WATER_LEVEL_PIN_1, INPUT);
pinMode(WATER_LEVEL_PIN_2, INPUT);
pinMode(WATER_LEVEL_PIN_3, INPUT);
pinMode(WATER_LEVEL_PIN_4, INPUT);
pinMode(RELAY_PIN, OUTPUT);
pinMode(BUZZER_PIN, OUTPUT);

thing.add_wifi(ssid, password);
//thing["pump"] << digitalPin(RELAY_PIN);
thing["water_level"] >> outputValue(waterLevel);
thing["pump_state"] >> outputValue(pumpState);
}

void loop() {
    // Отримання стану датчиків рівня води
    int level1 = digitalRead(WATER_LEVEL_PIN_1);
    int level2 = digitalRead(WATER_LEVEL_PIN_2);
    int level3 = digitalRead(WATER_LEVEL_PIN_3);
    int level4 = digitalRead(WATER_LEVEL_PIN_4);

    // Обчислення поточного рівня води
    int currentWaterLevel = level1 * 25 + level2 * 25 + level3 * 25
+ level4 * 25;

    // Визначення стану насоса залежно від рівня води
    if (currentWaterLevel < 25) {
        digitalWrite(RELAY_PIN, HIGH); // Увімкнення насоса
        pumpState = true;
        for (int i = 0; i < 3; i++) {
            tone(BUZZER_PIN, 1000); // Бузер видає сигнал на частоті
1000 Гц
            delay(200);
            noTone(BUZZER_PIN);
            delay(200);
        }
    } else if (currentWaterLevel >= 100) {
        digitalWrite(RELAY_PIN, LOW); // Вимкнення насоса
        pumpState = false;
    }

    // Перевірка, чи змінився рівень води
    if (currentWaterLevel != lastWaterLevel) {
        waterLevel = currentWaterLevel;
    }
}

```



```

    lastWaterLevel = currentWaterLevel;

    // Виведення поточного рівня води та стану насоса на OLED
    дисплей
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.print("Water ");
    display.print(waterLevel);
    display.println("%");
    display.print("Pump ");
    display.println(pumpState ? "ON" : "OFF");
    display.display();

    // Виведення поточного рівня води та стану насоса в Serial
    Monitor
    Serial.print("Water Level: ");
    Serial.print(waterLevel);
    Serial.print("% | ");
    Serial.print("Pump State: ");
    Serial.println(pumpState ? "ON" : "OFF");

}

// Надсилання даних на thinger.io
thing.handle();

// Затримка перед наступною ітерацією
delay(1000); // Невелика затримка для зменшення навантаження на
мікроконтролер
}

```