

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Кіберфізична система керування водопостачанням домашнього господарства  
Назва теми

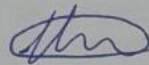
КвРКІ 022032.22.02.27 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва

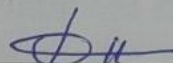
Освітня програма «Комп'ютерна інженерія та програмування»  
Назва

Виконав: студент III курсу, група КІ2с-22-2

  
Підпис

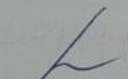
Маским ПАСІЧНИК  
Ініціали, прізвище

Керівник

  
Підпис, дата

Микола ФЕДУЛА  
Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

Тетяна КИСІЛЬ  
Ініціали, прізвище

До захисту допускаю:  
зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Ольга ПАВЛОВА  
Ініціали, прізвище

«13» червня 2025 р.

Хмельницький 2025

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Максиму ПАСІЧНИКУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система керування водопостачанням домашнього господарства

Керівник проекту (роботи) Микола ФЕДУЛА, к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Дослідження існуючих рішень питань постачання води для приватних домашніх господарств та потенційних економічних ніш доступних в цій області

Розроблення кіберфізичної системи керування водопостачанням домашнього господарства та програмного забезпечення для неї

Реалізація симуляції кіберфізичної системи керування водопостачанням домашнього господарства

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Структурна \_\_\_\_\_ схема \_\_\_\_\_ проекту

Діаграма діяльності ПЗ мікроконтролера \_\_\_\_\_

Симуляції \_\_\_\_\_ та \_\_\_\_\_ діаграма \_\_\_\_\_ діяльності

проекту \_\_\_\_\_

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагіат	Андрій Нічепорук, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів та розробка структурної схеми системи водопостачання домашнього господарства	01.04.2025	виконано
5	Робота над розділом 3 – розробка ПЗ та симулювання системи водопостачання домашнього господарства	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Керівник роботи

Підпис

Максим ПАСІЧНИК  
Ініціали, прізвище

Підпис

Микола ФЕДУЛА  
Ініціали, прізвище

№ Р я д к а	ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ ек з	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРКІ 022032.22.02.27 ПЗ	Пояснювальна записка	97		
			<u>Графічні матеріали</u>			
2		КвРКІ 022032.22.02.27 Е8	Структурна схема проєкту	1		
3		КвРКІ 022032.22.02.27 Е8	Діаграма діяльності ПЗ мікроконтролера	1		
4		КвРКІ 022032.22.02.27 Е8	Симуляції та діаграма діяльності проєкту	1		

КвРКІ 022032.22.02.27 ВП

Зм	Арк	№ докум	Підпис	Дата
Розробив		Пасічник		11.06.25
Перевір.		Федула		11.06.25
Н. контр.		Кисіль		26.6
Затв.		Павлова		11.06

Літера	Аркуш	Аркушів
У	1	1

Відомість проєкту

ХНУ, КІ2с-22-2

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Кіберфізична система керування водопостачання домашнього господарства».

Автор роботи: Пасічник Максим Костянтинович.

Керівник роботи: Федула Микола Васильович.

Пояснювальна записка: 73 с., 35 рис., 5 табл., 4 дод., 46 джерел.

Графічна частина: 3 креслення.

ВОДОПОСТАЧАННЯ, КІБЕРФІЗИЧНА СИСТЕМА, АРХІТЕКТУРА,  
МОНІТОРИНГ.

Метою роботи є розроблення кіберфізичної система керування водопостачанням домашнього господарства на основі аналізу та узгодження режимів роботи насосів і витрат води.

Під час проведення даної роботи було розроблено симульції кіберфізичної системи керування постачанням води домашнього господарства в середовищах Wokwi та Simulink. Було розроблено засоби визначення ймовірних значень вимірюваних величин базуючись на показаннях сенсорів та виявлення множин сенсорів, що вийшли з ладу, базуючись на протеріччах вимірюваних величин. Розроблене ПЗ мало здатність передавати про стан системи та отримувати данні необхідні для корегування функціонування системи з хмари



Підпис студента

30.05.2025

Дата

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>1 АНАЛІЗ ВІДОМИХ РІШЕНЬ ТА ВИБІР СТРАТЕГІЇ І ЗАСОБІВ РЕАЛІЗАЦІЇ ЗАВДАННЯ</b> .....	7
1.1 Огляд відомих систем водопостачання домашнього господарства.....	7
1.2 Огляд відомих методів автоматизованого керування водопостачанням у домашніх господарствах.....	14
1.3 Огляд відомих кіберфізичних систем для керування водопостачанням домашнього господарства .....	16
1.4 Висновки до першого розділу.....	21
<b>2 АРХІТЕКТУРА СИСТЕМИ КЕРУВАННЯ ВОДОПОСТАЧАННЯ ДОМАШНЬОГО ГОСПОДАРСТВА</b> .....	22
2.1 Розробка структурної схеми.....	22
2.2 Вибір мікроконтролера.....	23
2.3 Вибір сенсора температури.....	29
2.4 Вибір сенсора рівня води.....	32
2.5 Вибір сенсора тиску .....	37
2.6 Вибір дисплея .....	40
2.7 Висновки до другого розділу .....	43
<b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ ВОДОПОСТАЧАННЯ ДОМОГОСПОДАРСТВА</b> .....	44
3.1 Опис реалізації симуляції системи водопостачання в Wokwi.....	44
3.2 Опис програмного забезпечення ситеми водопостачання для збереження інформації про стан системи.....	59
3.3 Опис реалізації симуляції системи водопостачання в MATLAB .....	60
3.4. Висновки до третього розділу.....	65
<b>ВИСНОВКИ</b> .....	66

				КвРКІ.022032.22.02.27 ПЗ			
Зм. Арк.	№докум.	Підпис	Дата	Кіберфізична система керування водопостачанням домашнього господарства	Літера	Аркуш	Аркушів
Виконав	Максим ПАСІЧНИК		11.06.25		у	2	73
Перевір.	Микола ФЕДУЛА		11.06.25	Пояснювальна записка	ХНУ КІ2с-22-2		
Н.контр.	Тетяна КИСІЛЬ		11.06.25				
Затвер.	Ольга ПАВЛОВА		11.06.25				

<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....</b>	<b>69</b>
<b>ДОДАТОК А.....</b>	<b>74</b>
<b>ДОДАТОК Б.....</b>	<b>75</b>
<b>ДОДАТОК В.....</b>	<b>76</b>
<b>ДОДАТОК Г.....</b>	<b>77</b>

					КВРКІ.022032.22.02.27 ПЗ	Арк. 3
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Актуальність теми керування процесами домашніх господарств у сучасних умовах автоматизації процесів життєзабезпечення стає невід'ємною частиною розвитку розумних будинків та кіберфізичних систем. Однією з ключових сфер застосування таких технологій є керування водопостачанням, оскільки раціональне використання водних ресурсів безпосередньо впливає на економічну ефективність, стабільність та екологічну безпеку цілих регіонів та країн.

Ще з моменту зародження цивілізації та людського виду, яким ми його знаємо зараз, та до індустріальної революції абсолютна більшість людської праці була направлена на виготовлення їжі для підтримання популяції. В рамках агрокультурних методів розв'язання цих питань виникає потреба в стабільному та безперервному доступі до води. На це вказує той факт, що до недавнього часу усі міста з великою кількістю населення засновувалися на берегах річок та морів.

У сучасному світі, де технологічний та соціальний прогрес призвів до значного зменшення кількості людей, що знаходять своє покликання в секторі вирощування їжі, внаслідок надзвичайного збільшення ефективності людської праці у цій сфері. Підвищення ефективності мало ефект також на логістичні питання постачання води для різноманітних господарств – у першу чергу господарств державного масштабу і важливості, але також і приватних господарств різних розмірів, потреб та призначень від великого бізнесу до приватних будинків пересічних громадян. Виникає потреба в автоматизованих, надійних та багатофункціональних системах водопостачання, що відповідають все вищим стандартам.

Зростання попиту на ефективні рішення для керування водопостачанням обумовлене кількома факторами. По-перше, зростаюча урбанізація та зміни клімату призводять до нестачі водних ресурсів, що вимагає впровадження технологій їхнього раціонального використання. По-друге, розвиток Інтернету речей (IoT) створює нові можливості для автоматизації та інтелектуального аналізу

					КВРКІ.022032.22.02.27 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

даних у сфері водопостачання. По-третє, споживачі все більше зацікавлені у скороченні витрат на комунальні послуги, що стимулює розвиток автоматизованих рішень для контролю споживання води.

Кіберфізичні системи (CPS) об'єднують апаратні та програмні компоненти для моніторингу, керування та оптимізації складних інженерних процесів. У контексті водопостачання вони забезпечують інтеграцію сенсорів, пристроїв, що керують і аналітичного програмного забезпечення для автоматичного контролю подачі та розподілу води в домашньому господарстві. Такі системи дозволяють мінімізувати втрати води, підвищити її якість і знизити енергоспоживання за рахунок інтелектуального керування насосами та клапанами.

Сучасні кіберфізичні системи водопостачання дозволяють вирішувати широкий спектр завдань, зокрема прогнозування споживання, виявлення витоків, динамічне регулювання тиску в системі та інтеграцію з іншими елементами розумного будинку. Однак традиційні методи керування водопостачанням не завжди забезпечують необхідний рівень гнучкості, адаптивності та безпеки. Тому розробка та впровадження кіберфізичної системи, яка поєднує сучасні технології збору та аналізу даних, є актуальною науковою та практичною задачею.

Таким чином, актуальність дослідження полягає у необхідності створення ефективного та надійного рішення для керування водопостачанням, яке дозволить оптимізувати використання ресурсів, зменшити експлуатаційні витрати та підвищити рівень комфорту і безпеки у домашньому господарстві.

Впровадження такої системи сприятиме не лише економії води, але й підвищенню загальної ефективності функціонування інженерних мереж, що є важливим кроком у розвитку сучасних інтелектуальних технологій керування життєвими процесами.

Метою роботи є розроблення кіберфізичної система керування водопостачанням домашнього господарства на основі аналізу та узгодження режимів роботи насосів і витрат води.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

Для досягнення мети роботи необхідно виконати ряд часткових завдань: провести аналітичний огляд відомих кіберфізичних систем керування водопостачанням домашніх господарств, розробити архітектуру кіберфізичної системи керування водопостачанням домашнього господарства, реалізувати архітектуру кіберфізичної системи керування водопостачанням домашнього господарства з використанням імітаційного моделювання в Wokwi та Simulink.

					КВРКІ.022032.22.02.27 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

# 1 АНАЛІЗ ВІДОМИХ РІШЕНЬ ТА ВИБІР СТРАТЕГІЇ І ЗАСОБІВ РЕАЛІЗАЦІЇ ЗАВДАННЯ

## 1.1 Огляд відомих систем водопостачання домашнього господарства

Водопостачання є однією з ключових частин життєзабезпечення домашніх господарств. Його основна мета – забезпечення мешканців чистою водою для пиття, побутових потреб та технічного використання. Теперішні системи водопостачання можна поділити на централізовані, які підключені до муніципальної водної інфраструктури, та децентралізовані, що використовують власні джерела води [1].

Крім того, сучасні технології дозволяють значно підвищити ефективність водопостачання за допомогою автоматизованих та кіберфізичних систем керування. Розглянемо основні типи систем водопостачання, їхні переваги та недоліки.

Система водопостачання – це сукупність споруд та інженерних комунікацій, призначених для забору води з природних джерел, її можливого очищення, транспортування та розподілу між кінцевими споживачами. Основною функцією такої системи є автоматизоване постачання води через мережу трубопроводів, що забезпечує доступ до водних ресурсів у необхідних обсягах.

Водопостачальні системи поділяються за кількома критеріями. Залежно від охоплення території вони бувають локальними, які обслуговують один населений пункт чи окремий об'єкт, та груповими, що забезпечують водою кілька населених пунктів або навіть цілі регіони [1].

За джерелами води системи водопостачання поділяються на три типи: підземні, поверхневі та комбіновані.

Підземні системи водопостачання використовують водоносні горизонти, артезіанські свердловини або джерела. Поверхневі системи водопостачання отримують воду з річок, озер чи водосховищ. Комбіновані системи водопостачання поєднують підземні та поверхневі джерела для оптимального забезпечення водопостачання.

					КВРКІ.022032.22.02.27 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

Системи водопостачання також поділяються на три типи за способом транспортування води. Серед них розрізняють системи з примусовою подачею, самотічні (гравітаційні) системи та комбіновані системи [1].

Системи з примусовою подачею мають таку структуру, що для переміщення води застосовуються насосні станції. В самотічних (гравітаційних) системах вода надходить до споживачів завдяки природному перепаду висот між водозабором і точками розподілу. Комбіновані системи використовують як гравітаційне транспортування, так і механізовану подачу в різних частинах мережі.

Розглянемо систему водопостачання для приватного будинку, що надається компанією "Віола Аква-Сервіс" [2].

Автономна система водопостачання приватного будинку дозволяє забезпечити незалежність від централізованих комунікацій, використовуючи природні підземні води. Вона складається з кількох ключових елементів: водозабірної споруди (свердловини або колодязя), насосного обладнання, трубопроводів, системи очищення та накопичення води.

Водозабірні споруди є головним джерелом води в автономній системі. Двома основними видами водозабірних споруд є свердловини та колодязі.

Свердловини можуть мати різну глибину залежно від геологічних умов і типу водоносного горизонту. Глибокі артезіанські свердловини забезпечують стабільний доступ до води високої якості. Колодязі зазвичай мають глибину до 30 метрів і використовують верхні водоносні горизонти, що робить їх менш стабільними за якістю та кількістю води.

При виборі між свердловиною і колодязем необхідно враховувати хімічний склад води, продуктивність джерела та санітарні вимоги.

Насосне обладнання та транспортування води є дуже важливими оскільки вода з підземних джерел не має природного тиску, для її подачі застосовують насоси різних типів.

Глибинні насоси використовуються для свердловин глибиною понад 20-30 метрів, занурюються безпосередньо у воду та забезпечують стабільний напір.

					КВРКІ.022032.22.02.27 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

Насосні станції встановлюються в системах із невеликою глибиною водозабору (до 9 метрів). Вони містять насос, гідроаккумулятор і систему керування.

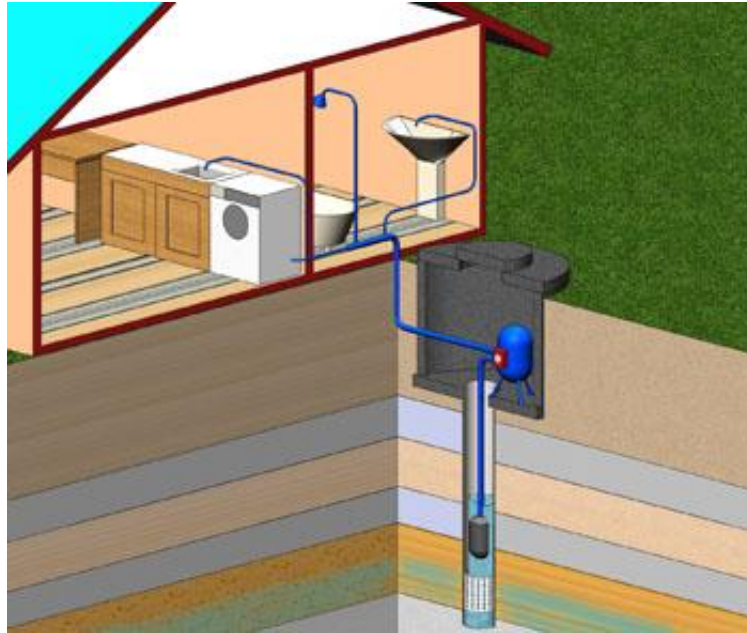


Рисунок 1.1 – Приклад системи водопостачання з глибинним насосом [2]

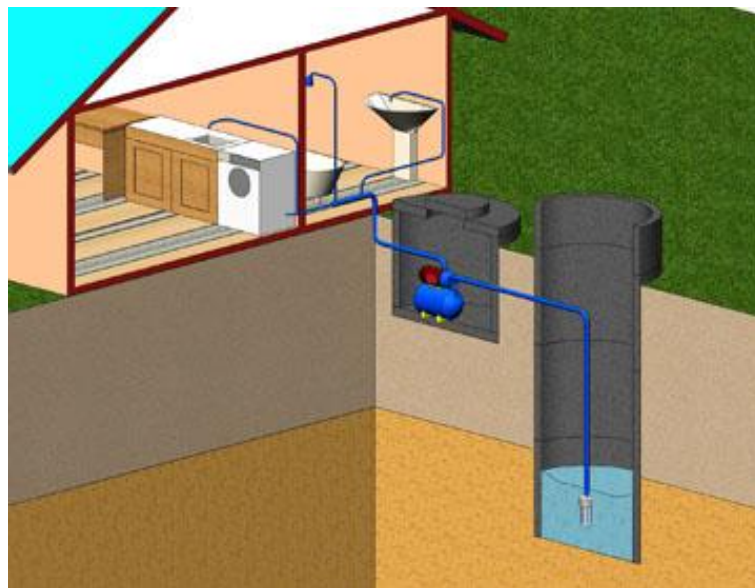


Рисунок 1.2 – Приклад системи водопостачання з використанням насосної станції [2]

Накопичувальні та регуляційні місткості зазвичай фігурують в системах водопостачання у двох видах. Система може включати гідроаккумулятор або

накопичувальний бак, які дозволяють регулювати тиск у трубопроводах і зменшувати навантаження на насосне обладнання. Це особливо важливо для стабільної роботи системи та зниження енергоспоживання.

Для забезпечення високої якості питної води автономні системи оснащуються багатоступневими фільтрами. У реальних системах водопостачання використовуються механічні фільтри які видаляють великі частинки піску, іржі та інших домішок, фільтри для пом'якшення води які знижують вміст кальцію та магнію, запобігаючи утворенню накипу та фільтри комплексної очищення які усувають домішки заліза, марганцю та органічні сполуки.

До переваг подібної системи можна віднести її відносну простоту, що збільшує надійність, зменшує ціну та складність встановлення, експлуатації та ремонту. Також система володіє високим ступенем гнучкості, оскільки не прив'язується до конкретних моделей обладнання, систему можна підлаштовувати до потреб господарства та властивостей місцевості на етапі встановлення та потенційного ремонту. Також незалежність від міських комунікацій, висока якість води при правильному облаштуванні та можливість використання великих об'ємів води без лімітів є позитивними сторонами подібних систем [1].

Однак високі початкові витрати на буріння свердловини та монтаж системи, необхідність регулярного технічного обслуговування, можливість вичерпання водоносного горизонту або його забруднення та відсутність використання сучасних кіберфізичних систем є суттєвими недоліками.

Розглянемо систему водопостачання для приватного будинку, що запропонована вітчизняної компанії "НасосВДом" [3].

					КВРКІ.022032.22.02.27 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.3 – Схема автономного водопостачання: основні елементи системи [3]

Система водопостачання, що рекомендується компанією, включає наступні компоненти (табл. 1.1).

Таблиця 1.1 – Таблиця основних компонентів системи водопостачання від вітчизняної компанії "НасосВДом"

Компонент	Опис призначення
Свердловина	Основне джерело води використовується для забору води з глибоких водоносних горизонтів. Вона забезпечує стабільне водопостачання з мінімальним ризиком забруднення
Кесон	Захисне приміщення для свердловини та захищає обладнання від впливу ґрунтових вод і морозів. Він зазвичай виконується з пластику, металу або бетонних кілець

Кінець таблиці 1.1

Глибинний насос	Занурюється у свердловину і подає воду до будинку (Водолій БЦПЕ). Його потужність підбирається залежно від глибини свердловини та потреби у воді
Зворотний клапан	Запобігає зворотному відтоку води та підтримує стабільний тиск у системі (клапан зворотний з латунним штоком)
Гідроакумулятор (гідробак)	Регулює тиск у системі та забезпечує резерв води. Місткість підбирається відповідно до витрат води (оптимально 80-100 л)
Фільтри для очищення води	Грубий фільтр, що затримує великі частинки бруду та піску та фільтр картриджного типу який видаляє дрібні домішки (глину, іржу, пісок). Додаткові фільтри встановлюються за потреби для видалення заліза чи пом'якшення води
Манометр	Контролює тиск у водопровідній системі (манометр 10 бар. D63)
Реле тиску	Автоматично вмикає та вимикає насос залежно від тиску в системі (Реле тиску РМ 5-G Італтехніка)
Електронне реле	Забезпечує захист від «сухого ходу» та стрибків напруги, наприклад SWITCHMATIC 1
Редуктор тиску	Стабілізує потік води на виході з системи



Рисунок 1.4 – Завершений варіант облаштування приямка (кесону) розглянутої системи [3]

Розглянута система водопостачання має схожий дизайн, переваги та недоліки в порівнянні з попередньою, але виявляє більше прив'язаності до конкретних моделей обладнання, що призводить до меншої гнучкості.

Також варто відзначити, що система має достатню простоту встановлення для того, щоб компанія надавала поради для коректного налаштування системи, спрямовані на кінцевого споживача. Важливою перевагою порівняно з попереднім дизайном є також більша увага до безпеки експлуатації системи, що пояснює високу популярність в більше ніж 100 тисяч зацікавлених користувачів згідно з інформацією компанії [3].

Але обидві розглянуті системи водопостачання не використовують у повній мірі інформаційні технології для полегшення та покращення експлуатації кінцевим користувачем, таким чином отримуючи значний недолік в сучасному світі інформаційних технологій та кіберфізичних систем в порівнянні з централізованими та значними за масштабом системами постачання води.

## 1.2 Огляд відомих методів автоматизованого керування водопостачанням у домашніх господарствах

Автоматизовані системи керування водопостачанням у домашніх господарствах спрямовані на підтримання стабільного рівня води та тиску в системі без потреби постійного втручання користувача. Основними методами автоматизації таких систем є дистанційне або телемеханічне керування, автоматичне керування за рівнем води та автоматичне керування за тиском у системі [4].

Дистанційне або телемеханічне керування полягає в тому, що насос вмикається і вимикається за командами з централізованого пункту керування. Такий метод часто застосовується у великих господарствах, де важливо координувати роботу кількох насосних станцій.

Автоматичне керування за рівнем води передбачає використання сенсорів рівня, встановлених у водонапірному резервуарі або башті. Коли рівень води падає до нижньої позначки, насос автоматично вмикається, а при досягненні верхнього рівня вимикається.

Автоматичне керування за тиском у системі передбачає використання електроконтактних манометрів (ЕКМ), які контролюють тиск у водопровідній мережі. Якщо тиск знижується нижче встановленого порогу, насос вмикається, а коли тиск досягає необхідного значення, то вимикається.

Для керування глибинними насосами часто застосовують автоматизовані станції типу «Каскад» або «Лоцман». Вони складаються з блоку керування та захисту, магнітного пускача та сенсорів контролю (струму, рівня води, тиску). Такі системи забезпечують автоматичне увімкнення та вимкнення насоса відповідно до рівня води або тиску, можливість ручного та дистанційного керування, захист від перевантажень, асиметрії напруги та короткого замикання, блокування роботи насоса у разі критичного зниження рівня води (захист від сухого ходу) та світлової сигналізацію і контроль навантаження у фазах живлення.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

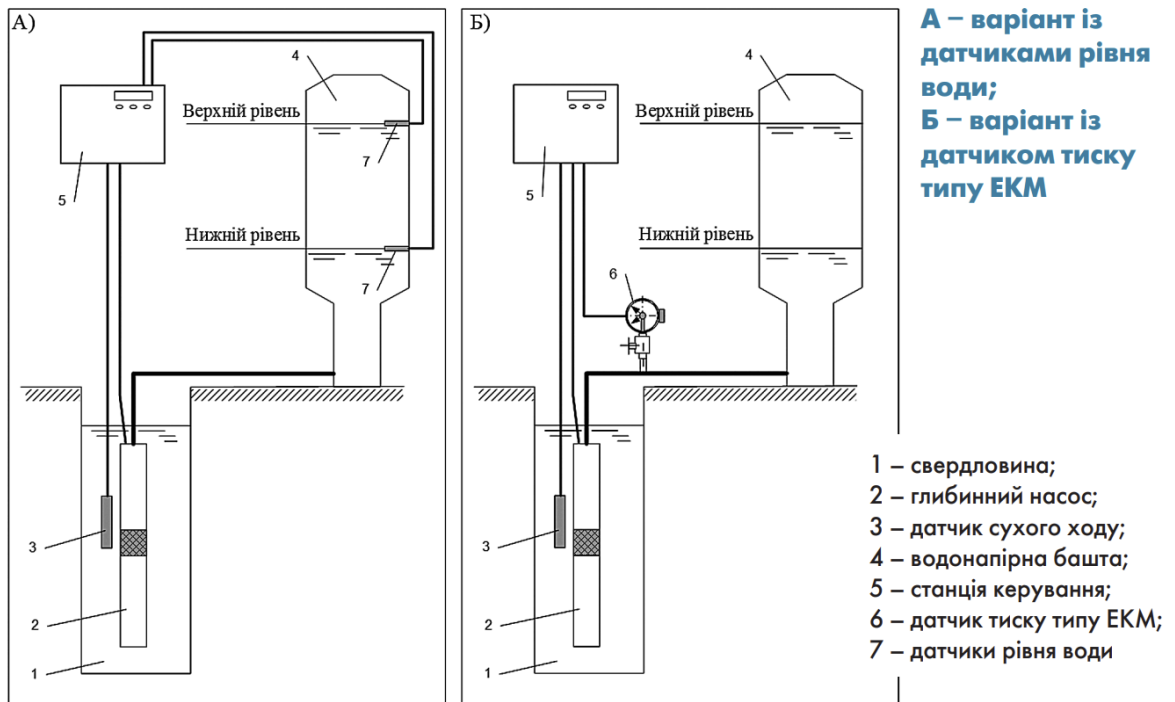


Рисунок 1.5 – Схема автоматизації роботи глибинного насоса [4]

Застосування таких методів дозволяє значно підвищити надійність роботи системи водопостачання, продовжити термін служби насосного обладнання та мінімізувати ризик аварійних ситуацій.

Окрім автоматизованого регулювання за рівнем води та тиском, сучасні системи водопостачання також забезпечують автоматизацію допоміжних процесів, таких як керування запірною арматурою, запуск резервних насосних агрегатів, а також аварійне відключення обладнання [4].

Для безперебійної роботи системи важливо, щоб усі дії, пов'язані з запуском і зупинкою насосів (відкриття й закриття засувки, заповнення насосів водою, охолодження підшипників тощо), виконувалися без участі оператора. Виняток становлять насосні станції з низькими вимогами до надійності, де частину операцій може здійснювати персонал.

Якщо насос зупиняється аварійно через спрацювання захисних механізмів, система автоматизації повинна забезпечити закриття засувки, щоб уникнути різких перепадів тиску та пошкодження трубопроводу. У випадку несправності засувки

передбачається автоматичне вимкнення насоса для запобігання подальшим аварійним ситуаціям.

Щоб підвищити ефективність і надійність роботи насосів, їх рекомендують встановлювати в умовах постійного заповнення водою, що мінімізує ризики повітряних пробок та полегшує запуск. Крім того, у разі затоплення машинного залу насосна станція має автоматично припинити роботу для уникнення пошкоджень електрообладнання.

Автоматизовані системи також дозволяють регулювати продуктивність насосних установок відповідно до змін водоспоживання. Це досягається шляхом керування кількістю справних насосних агрегатів, зміни швидкості їх обертання або регулювання потоку води дроселюванням. Використання частотних перетворювачів у приводах насосів дозволяє знизити енергоспоживання на 5–15%, а в окремих випадках – до 30%.

Додатково скоротити витрати на експлуатацію можна завдяки використанню накопичувальних місткостей (водонапірних башт) та оптимізації графіків роботи насосів відповідно до тарифних зон електроенергії. Наприклад, у період знижених тарифів (переважно вночі) здійснюється наповнення резервуарів, а в денний час подача води відбувається під дією гравітації, що зменшує навантаження на насосне обладнання.

### 1.3 Огляд відомих кіберфізичних систем для керування водопостачанням домашнього господарства

Сучасні кіберфізичні системи для водопостачання поєднують апаратні та програмні засоби для забезпечення ефективного керування водними ресурсами в реальному часі. Такі системи дозволяють автоматично контролювати тиск, рівень води, якість води, а також розподіл водних ресурсів залежно від поточного попиту.

Вони використовують ряд пристроїв для опрацювання даних, таких як мікроконтролери Arduino, ESP32, STM32 та інших лінійок [5-19]. Також

комп'ютерні системи подібного призначення використовують насосні станції різних потужностей та типів [20-22] і різноманітні сенсори для збору інформації про стан системи. До них можуть входити сенсори температури або LM35, або DS18B20, або SHT3x-DIS серії, або інші [23-28]. Також для досягнення своїх цілей такі системи можуть використовувати сенсори рівня води, які діють як напряду подібно до поплавкових сенсорів ZP4510 [29], так і непрямим підходом, як сенсори відстані HC-SR04 [30-31]. Додатково одним з надзвичайно широко використовуваних елементів систем постачання води є реле тиску які бувають різного функціоналу та складності [32-35].

Своєю чергою деякі сімейства та моделі мікроконтролерів можуть мати більш сумісну архітектуру до інших елементів системи [5-19, 36-37], що призводить до їх частішого використання з елементами які залежать від такої сумісності, як наприклад поширений дисплей I2C Serial Interface 20x4 LCD Module [38].

Однією з передових реалізацій кіберфізичних системи є комплексна система керування поливом та водопостачанням (КСКП), що була впроваджена на об'єкті Sunray. Вона забезпечує автоматизований контроль подачі води для різних цілей, включаючи побутове водопостачання та іригацію [39].

КСКП складається з кількох ключових компонентів, які взаємодіють між собою через локальну мережу, система керування поливом, система керування наповненням місткостей, система керування насосною станцією та система диспетчеризації та віддаленого доступу.

Система керування поливом (СКП) використовує прецизійні сенсори вологості ґрунту, які дозволяють визначати рівень зволоженості та коригувати об'єм подачі води в режимі реального часу. Основною перевагою цього є відмова від жорстких графіків поливу на користь гнучкої системи зрошення. Керування цим здійснюється за допомогою контролерів Acclima CS3500, які підключаються до мережі Ethernet для дистанційного моніторингу та налаштування параметрів

поливу. Використання декодерів в системі дозволяє автоматично відкривати та закривати клапани для подачі води за командами контролера.

Система керування наповненням ємностей (СКНЄ) відповідає за автоматичний вибір джерела води для поливу та водопостачання між резервуаром з очищеною стічною водою (пріоритетне джерело) і централізованим водопроводом (використовується лише за необхідності). Система аналізує рівень води у резервуарі та її якість. Якщо рівень достатній і вода відповідає встановленим нормам, вона використовується для поливу. У разі нестачі чи низької якості автоматично підключається водопровід.

Система керування насосною станцією (СКН) що здійснює контроль за роботою насосного обладнання та забезпечує підтримку оптимального рівня тиску в системі. Вона дозволяє гнучке керування насосами в шести режимах роботи, що включають як паралельне функціонування двох насосів, так і роботу одного насоса з резервним підключенням другого у разі аварії. Додатково система передбачає засоби захисту від деяких потенційних загроз: захист від “сухого ходу” (автоматичне вимкнення насосів у разі відсутності води) та захист від перегріву (контроль температури корпусу насосів).

Система диспетчеризації та віддаленого доступу (ЗДВД) що об'єднує всі підсистеми КСКП у єдину локальну мережу, що дозволяє централізовано контролювати та налаштовувати параметри роботи обладнання. Для цієї системи використовується диспетчерський персональний комп'ютер (ДПК) із встановленим спеціалізованим програмним забезпеченням. Оператор ДПК може переглядати поточний стан системи, отримувати добові звіти про витрати води, рівень вологості ґрунту, стан насосного обладнання та інші параметри.

					КвРКІ.022032.22.02.27 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.6 – Вигляд системи керування насосною станцією [39]

Впровадження такої системи призвело до скорочення витрат води, зниження споживання електроенергії, підвищення надійності роботи, збільшення гнучкості та простоти керування та забезпечення оперативного моніторингу стану системи і можливість дистанційного втручання.

Для досягнення таких результатів було використано ряд спеціалізованого обладнання (табл. 1.2).

Таблиця 1.2 – Основні компоненти комплексної системи керування поливом та водопостачанням на об’єкті Sunray

Компонент	Опис призначення
ПЛК160[U22]	Програмований логічний контролер з дискретними і аналоговими входами/виходами для автоматизації середніх систем в двох екземплярах. Контролери цієї лінійки рекомендуються для побудови розподілених систем керування і диспетчеризації з використанням як провідних, так і безпроводних технологій [40]

Кінець таблиці 1.2

БП15Б-Д2	Промисловий блок живлення призначений для широкого спектру радіоелектронних пристроїв стабілізованою напругою постійного струму – релейної автоматики, контролерів тощо в двох екземплярах [41]
МВ110-16Д	Пристрій призначений для збирання даних із вбудованих дискретних входів з подальшим передаванням їх в мережу RS-485. Вбудовані дискретні входи можуть працювати у режимі лічильників імпульсів частотою до 1 кГц МВ110 працює у мережі RS-485 за протоколами ModBus-RTU, ModBus-ASCII, DCON [42]
МУ110-8Р	Пристрій призначений для керування вбудованими дискретними виходами за сигналами із мережі RS-485, які використовуються для підмикання виконавчих механізмів з дискретним керуванням в одному екземплярі [43]
БСФ-Д2-0,6	Блоки мережевих фільтрів призначені для захисту двопровідної схеми змінного струму, що живить пристрої та сенсори, від імпульсних та високочастотних завад в двох екземплярах [44]
СП270	Кольорова графічна панель з сенсорним дисплеєм, підтримує спільну роботу з ОВЕН ПЛК, модулями ОВЕН Мх110, а також з контролерами та модулями інших виробників в одному екземплярі. Графічний дисплей з діагоналлю 7 дюймів та роздільною здатністю 480x234 пікселя [45]

Основним недоліком розглянутої системи в контексті водопостачання домашнього господарства є велика ціна та складність, що є бар'єром у використанні таких систем пересічними громадянами для фінансово доступного та безпечного водопостачання їхніх осель. Головною причиною значної ціни є той факт, що система розроблялася не для використання одним приватним будинком, а для заміського комплексу відпочинку у Дніпрі SunRay [46]; що вимагає більшого машабу системи і відповідно вищої ціни.

#### 1.4 Висновки до першого розділу

В процесі аналізу відомих систем водопостачання для приватних господарств було розглянуто прості системи водопостачання домашніх господарств, автоматизовані системи водопостачання та кіберфізичні системи керування водопостачанням домашніх господарств.

Під час огляду наявних рішень було виявлено відсутність або надзвичайно маленьку кількість систем водопостачання, які б використовували сучасні технологічні можливості для збільшення безпеки та складності алгоритмів експлуатації й одночасно знаходились в ціновій категорії, як необхідного обладнання, так і вартості його встановлення і розробки системи доступних для пересічного громадянина, в рамках сучасної економічної ситуації.

Оскільки ціни, енергоспоживання, розповсюдженість, ергономічність та надійність електронних засобів які можуть бути використані в проектах розумних будинків значно покращились на останні десятиліття, це вказує на можливу економічну нішу для порівняно економних кіберфізичних систем водопостачання приватних домогосподарств направлених на покращення експлуатації та надійності вже наявних систем. Таким чином, розроблення кіберфізичних систем водопостачання домашніх господарств є актуальним завданням сучасної комп'ютерної інженерії, вирішення якого надає можливість суттєво покращити характеристики поширених систем постачання води.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 АРХІТЕКТУРА СИСТЕМИ КЕРУВАННЯ ВОДОПОСТАЧАННЯ ДОМАШНЬОГО ГОСПОДАРСТВА

### 2.1 Розробка структурної схеми

Детальний огляд систем, методів керування та кіберфізичних систем керування водопостачанням домашнього господарства, який був проведений у попередньому розділі, дозволяє розробити структурну схему (додаток А).

Центральним елементом кіберфізичної системи водопостачання приватного домогосподарства є мікроконтролер, що відповідає за збір даних про стан системи від сенсорів та прийняття, на основі цих даних, рішень про керування насосом та відправлення повідомлень користувачеві.

Сенсори температури відповідають за перевірку знаходження насоса в допустимих температурних рамках для справної роботи. Для перевірки, та ввімкнення насоса, для передачі води з гідроакумулятора в труби для подальшого використання кранами на кухні та в ванній кімнатах та в системі поливу газону, використовується реле тиску. Сенсор рівня води використовується для перевірки рівня води в резервуарі для отримання інформації про необхідність ввімкнення свердловинного насоса.

Інформація зі всіх цих сенсорів передається до мікроконтролера після чого для підвищення надійності результати роботи однотипних сенсорів порівнюється з метою визначення безпечності використання даних. Якщо всі три показують однакове значення, в межах допустимої похибки вимірювання то система приймає рішення, що всі три сенсори справні. У випадку якщо показники одного з сенсорів суттєво відрізняються від інформації отриманої з інших двох, показники більшості вибираються як умовно істинні значення, але це також дає додаткову можливість попередити користувача про те, що принаймні один сенсор вийшов з ладу. У рідкісних випадках коли два або всі три сенсори вийдуть з ладу в достатньо короткий період, і користувач не може замінити сенсор, що вийшов з ладу першим,

					КвРКІ.022032.22.02.27 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

всі три сенсори будуть показувати різну інформацію, яку можна використати для виявлення несправностей.

Також сенсорна панель та модуль зв'язку можуть бути використані для комунікації з користувачем про стан системи та рекомендовані дії та або налаштування.

## 2.2 Вибір мікроконтролера

З метою роботи кіберфізичних систем водопостачання домогосподарства необхідно використати як елемент системи пристрій, що буде відповідати за логіку роботи та взаємодії всіх інших елементів між собою та системи загалом з користувачем. Таким елементом може виступати мікроконтролер.

Для ефективного використання в системі розумного водопостачання приватного домогосподарства, з використанням наявної економічної ніші, мікроконтролер повинен відповідати наступним вимогам: хороший баланс ціни і якості, для збереження економічної доцільності, достатня обчислювальна потужність для ефективного опрацювання даних сенсорів і виконання контролю над системою, низьке споживання електроенергії, для ефективної роботи в режимі очікування, підтримка можливості роботи з дисплеєм, підтримка можливості використання Wi-Fi задля моніторингу стану та передання команд від користувача на відстані.

ESP32 це потужна та універсальна мікроконтролерна платформа, розроблена компанією Espressif Systems (рис. 2.1). Вона відрізняється своєю низькою ціною та наявністю WiFi модуля, як і її попередник ESP8266, але потужним процесором Tensilica Xtensa LX6 та більшою кількістю периферійних пристроїв призвело до надзвичайної популярності [10].

					КВРКІ.022032.22.02.27 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

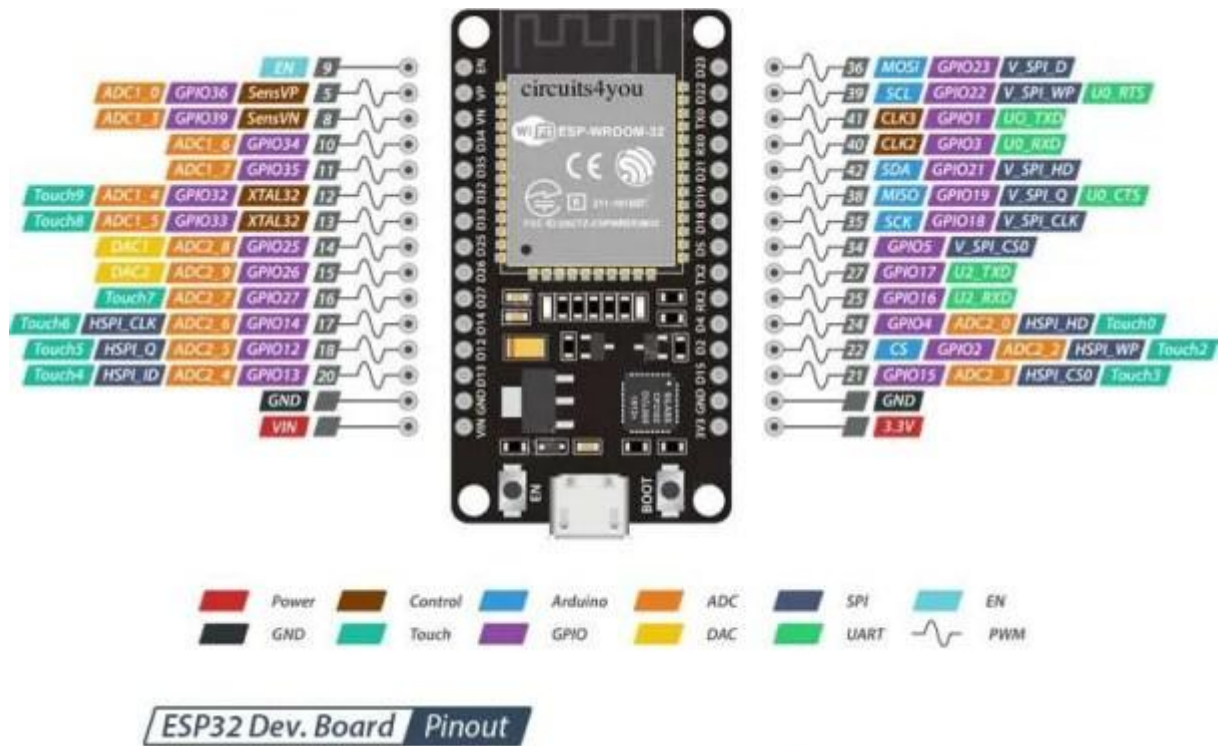


Рисунок 2.1 – Розташування виводів плати ESP32 [11]

Серія ESP32 поділяється на різноманітні мікроконтролери, до них входять ESP32-C3, ESP32-C6, ESP32-S2, ESP32-S3, ESP32-H, ESP32-H2FH2 та ін. Основною відмінністю між ними є розмір пам'яті та наявність деяких периферійних пристроїв. Серію ESP32, в широкому сенсі цього слова, можна розділити на ESP32-Cx, ESP32-Sx та ESP32-Hx.

ESP32-Cx є серією з порівняно невеликою кількістю периферійних пристроїв та енергоспоживанням, призначеною найкраще та найближче замінити пристрої на основі ESP8266.

ESP32-Sx – це версії з найкращими технічними характеристиками, наприклад ESP32-S3 був розроблений для систем машинного навчання, що, як відомо, потребують чимало обчислювальних ресурсів [12], але деколи можуть не мати підтримки Bluetooth.

ESP32-Hx спеціалізуються на використанні в сфері IoT та мають підтримку мережних протоколів Zigbee та Matter для використання в різноманітних проєктах розумних будинків які стали популярними в останні роки [13].

Порівняємо характеристики та основні параметри мікроконтролерів ESP32, ESP32-S2, ESP32-S3, ESP32-C3 та ESP32-C6 між собою та з ESP8266 в більших деталях [10, 14, 15] (табл. 2.1).

Таблиця 2.1 – Порівняльна таблиця ESP8266, ESP32, ESP32-S2, ESP32-S3, ESP32-C3, ESP32-C6

	ESP8266	ESP32	ESP32-S2	ESP32-S3	ESP32-C3	ESP32-C6
Головний процесор	Tensilica L106 32-bit	Tensilica Xtensa 32-bit LX6	Tensilica Xtensa 32-bit LX7	Tensilica Xtensa 32-bit LX7 dual core	RISC-V 32-bit	RISC-V 32-bit
Частота процесора	160 МГц	240 МГц	240 МГц	240 МГц	160 МГц	160 МГц
SRAM	160 Кбайт	520 Кбайт	320 Кбайт	512 Кбайт	400 Кбайт	400 Кбайт
ROM	0	448 Кбайт	128 Кбайт	384 Кбайт	384 Кбайт	384 Кбайт
Bluetooth	Немає	BLE 4.2	Немає	BLE 5.0	BLE 5.0	BLE 5.0
WiFi	Wi-Fi 4	Wi-Fi 4	Wi-Fi 4	Wi-Fi 4	Wi-Fi 4	Wi-Fi 6
PMU	Є	Є	Є	Є	Є	Є
I2C	1	2	2	2	1	1
I2S	1	2	1	2	1	1
UART	2	3	2	3	2	2
SPI	2	4	4	2	3	2
SDIO Host	0	1	0	2	0	0

Кінець таблиці 2.1

SDIO Slave	0	1	0	0	0	1
GPIO	17	43	43	44	22	22
LED PWM	5	16	8	8	6	6
MCPWM	0	6	0	2	0	0
Pulse counter	0	8	4	4	0	1
GDMA	0	0	0	5	3	6
Timer	2x 23-біт	4x 64-біт	4x 64-біт	4x 54-біт + 1x 52-біт	2x 54-біт + 1x 52-біт	1
DAC	Немає	2x 8-біт	2x 8-біт	Немає	Немає	Немає
ADC	1x 10-біт SAR	2x 12-біт SAR (18 каналів)	2x 13-bit SAR (20 каналів)	2x 12-bit SAR (20 каналів)	2x 12-біт SAR, (6 каналів)	12-біт SAR, (7 каналів)
TWAI	0	1	1	1	1	2
USB	Немає	Немає	USB OTG 1.1	Serial/JTAG	Serial/JTAG	Serial/JTAG
Температурний сенсор	Є	Є	Є	Є	Є	Є
Сенсор дотику	0	10	14	14	0	0

Для використання мікроконтролерів ESP32 існує декілька сумісних програмних середовищ розробки: Espressif SDK, Mongoose OS, MicroPython, PlatformIO, Arduino IDE [16 - 17].

Espressif SDK це пакет утиліт і бібліотек для мікроконтролерів Espressif. Для повноцінної розробки програмного забезпечення таким чином є фреймворк ESP-IDF, що доступний на всіх основних платформах.

Mongoose OS – це фреймворк для розробки програмного забезпечення для IoT, що підтримує C та JavaScript.

MicroPython є компілятором та середовищем виконання для мікроконтролерів. Він містить ряд бібліотек для взаємодії з низьким рівнем апаратного забезпечення.

PlatformIO – це програмне забезпечення для розробки застосунків для різноманітних мікроконтролерів, наприклад ESP32. PlatformIO може використовувати в якості IDE: Visual Studio Code та Atom.

Arduino IDE це середовище розробки з інтуїтивним інтерфейсом, що було згадано вище. Не зважаючи на те, що базова версія Arduino IDE не підтримує мікроконтролери ESP32, їх підтримку можна додатково встановити. Процес полегшується великою розповсюдженістю навчальних матеріалів на тему вказаних мікроконтролерів.

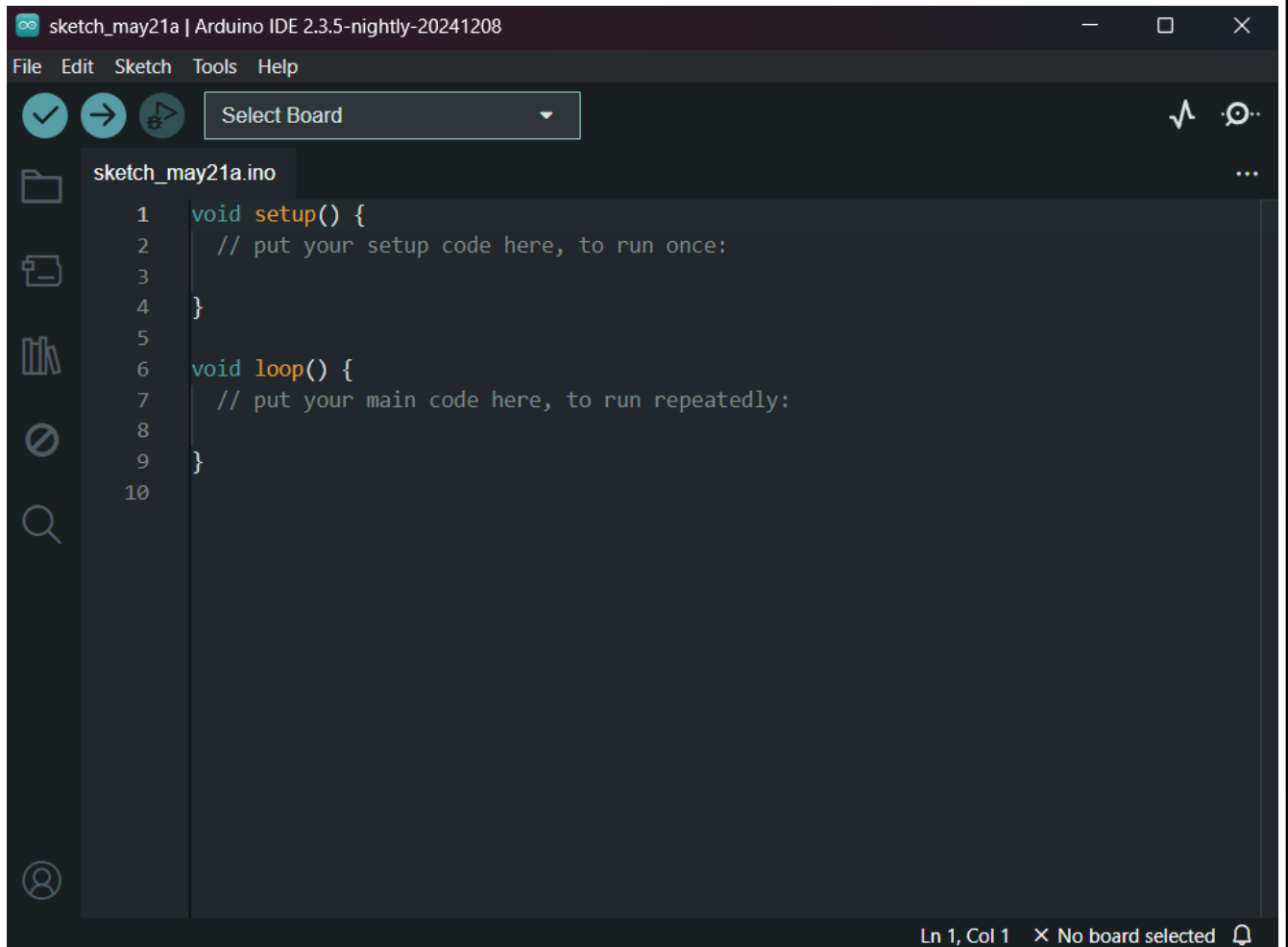


Рисунок 2.2 – Інтуїтивно зрозумілий інтерфейс Arduino IDE

Основними кандидатами для виконання функцій контролю над взаємодією з користувачем та елементами системи, сенсорами, насосними станціями та іншими є ESP32 та ESP32-S3.

При цьому ESP32-S3 є альтернативним, більш потужним рішенням яке може бути використано у випадку значного ускладнення логіки роботи програмного забезпечення проєкту у майбутньому, що призведе до підвищення вимог до потужності технічних характеристик мікроконтролера, його можна буде замінити на ESP32-S3 з порівняно невеликим збільшенням ціни [18 - 19] і збереження більшості переваг оригінального вибору.

Але враховуючи надлишковість технічних характеристик ESP32-S3 для поточних задач системи водопостачання перевагу варто надати ESP32.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

В результаті через наявність та підтримку модуля Wi-Fi, низьке енергоспоживання, хороші технічні характеристики, популярність, що спричиняє поширеність навчального матеріалу та інформації і легкість знаходження моделей, доступ до легкого в використанні програмного забезпечення для розробки достатню кількість виводів для роботи з необхідною кількістю сенсорів та інших пристроїв, після аналізу наявних рішень та альтернатив було обрано ESP32-DevKitC V4, що можна придбати за 427 грн [28].

### 2.3 Вибір сенсора температури

Перед сенсором температури, призначеним для використання в системі постачання води, що буде використовуватись для перевірки стану насосної станції висувуються різноманітні вимоги яким він повинен відповідати: стійкість до фізичних вібрацій, стійкість до вологості, можливість добре переймати тепло та достатньо широкий діапазон температури.

Стійкість до фізичних вібрацій необхідна тому, що сенсор температури буде вимірювати стан насосної станції, що має рухомі частини і може генерувати значні фізичні вібрації під час роботи, сенсор температури повинен бути стійким до подібного впливу. Стійкість до вологості є важливою оскільки сенсор буде використовуватись поряд з обладнанням, що взаємодіє з водою, він повинен бути пристосованим до такої експлуатації.

Сенсор повинен мати можливість добре приймати тепло від насосних станцій та достатньо широкий діапазон температури який він може зчитати, щоб могли отримати інформацію про перегрів. Згідно з документаціями для різноманітних насосних станцій температура навколишнього середовища та транспортованої води не повинна перевищувати приблизно 40 °C [20 - 22], але температура корпусу і охолоджувальних елементів самої насосної станції може бути більше, тому діапазон від близько нуля до більш ніж сотні градусів може бути потрібним в залежності в моделі.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

Розглянемо сенсори температури LM35 та DS18B20 (рис. 2.3).

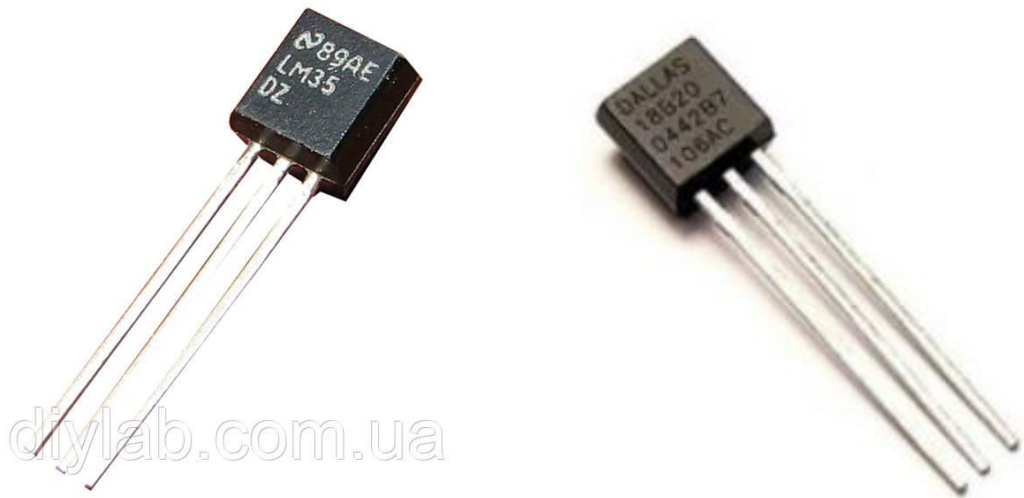


Рисунок 2.3 – Сенсори температури LM35 та DS18B20 [23 - 24]

Порівняємо їхні характеристики та основні параметри [23 -27] (табл. 2.2)

Таблиця 2.2 – Порівняльна таблиця LM35 та DS18B20

Параметр	LM35A	DS18B20
Напруга живлення	-0.2 - 35 V	+3.0 - +5.5 V
Діапазон вимірювання	-55 -150 °C	-55 - +125°C
Точність	± 0.25 °C	± 0.5 °C
Ціна	86 грн	25 грн

Своєю чергою сенсори подібної конструкції до сенсорів серії SHT; як, наприклад, SHT31 та SHT35 [28] всупереч своїх технічних характеристик, як вимірювальні прилади, на відміну від вище розглянутих температурних сенсорів, не придатні для використання в середовищі з значним рівнем вібрації як поверхня насосної станції.

Враховуючи те що обидва сенсори мають задовільні технічні характеристики та є досить поширеними, що дозволяє полегшити знаходження, закупівлю та придбання замінних екземплярів у випадку виходу з ладу одного з сенсорів, що

полегшує потенційний ремонт та наперекір того, що технічні характеристики сенсора DS18B20 трохи поступаються відповідним параметрам сенсора LM35A, майже втричі менша ціна натякає на кращий баланс з якістю для DS18B20. Розглянемо його розташування виводів (рис. 2.4).

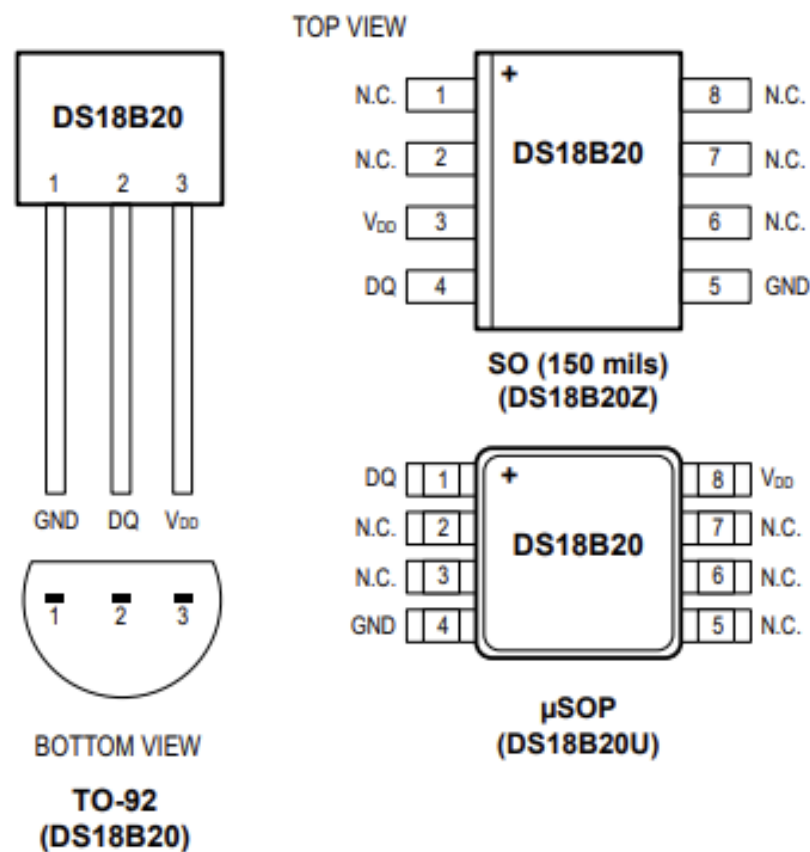


Рисунок 2.4 – Розташування виводів DS18B20 [25]

Інформація про температуру зовнішнього середовища яка вимірюється сенсором DS18B20 передається через 1-Wire інтерфейс. Таким чином зв'язок з мікроконтролером може бути реалізовано через один провід для передачі інформації та провід для заземлення, які будуть відповідати виводам GND та DQ сенсора.

Оскільки кожен сенсор DS18B20 містить унікальний серійний номер, що можна зчитати електронно, це дозволяє розміщення декількох сенсорів на одній

лінії [27]. Для найзручнішого використання цієї технології зазвичай використовують спеціалізовані бібліотеки.

Отож обираємо сенсор DS18B20 за задовільні технічні характеристики для виконання поставлених цілей та невелику ціну.

## 2.4 Вибір сенсора рівня води

Реалії системи постачання води для приватного господарства, вимагають застосування сенсорів, здатних виміряти рівень води в певному контейнері для її зберігання. Оскільки розташування свердловини для добування води для будівлі та здатність насоса передавати воду на певну висоту та відстань можуть створювати потребу в проміжному контейнері для зберігання води.

Додатково така споруда, в ролі якої може виступати звичайна пластикова або металева бочка для зберігання води виконує додаткову функцію розділення системи постачання на дві частини, з запасом води, що полегшує задоволення потреб в воді які мають значну по швидкості, але коротку по протяжності природу, оскільки миттєва швидкість споживання води не повинна бути рівною миттєвій швидкості її добування з свердловини.

Таким чином сенсори, здатні виміряти рівень води в резервуарі, дають можливість перевірити коли потрібно вмикати насоси та чи не працюють вони в холосту.

Таким чином перед сенсорами рівня води система її доставлення висуває вимоги до надійності роботи протягом довгого періоду оскільки у випадку їх несправної роботи один з насосів може працювати в холосту викликаючи перегрів, а інший може перевищити обсяг контейнера для зберігання води викликаючи затоплення приміщення.

В результаті важливим є те, щоб спосіб виміру або сенсор був стійким до забруднення води, оскільки її фільтрація зазвичай, як було розглянуто раніше, відбувається на більш пізніх стадіях обробки води.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

Для виміру рівня води існують різні фізичні методики, що породжує різні види сенсорів її рівня: поплавкові, ультразвукові, твердотілові, оптичні та провідникові.

Поплавкові сенсори є вимикачами, що плавають на поверхні води, змінюючи своє положення залежно від її рівня. Вони є дешевими надійними і простими в підключенні, але створюють труднощі при потребі виміру рівня води з певною точністю.

Ультразвукові сенсори вимірюють відстань від місця встановлення до води, таким чином чим більша відстань, тим менший рівень, за допомогою ультразвуку. Вони є безконтактними тому не залежать від чистоти води та дають можливість вимірювання рівня з високою точністю.

Твердотілові та оптичні сенсори використовують закони оптики для визначення рівня. Своєю чергою провідникові сенсори вимірюють провідність води між електродами. Мають простоту конструкції та використання.



Рисунок 2.5 – Сенсор рівня води ZP4510 [29]

Сенсор рівня води ZP4510 (рис. 2.5) має дуже просту структуру, яка забезпечує високу надійність. Його можна встановити як на верху так і в низу контейнера для води, а також він надає можливість перемкнути його з режиму відкритого за замовчуванням в режим закритого за замовчуванням.

Також через його простоту усі мікроконтролери можуть з ним успішно співпрацювати. Однак оскільки струм, що проходить через нього, не може перевищувати 0.5А, то випадку під'єднання одразу до насосної станції потрібно використати реле [29].

Простота конструкції такого типу сенсора запобігає вимірюванню води з точністю більше ніж один біт інформації, але їхня простота призводить до високого рівня надійності, що можна використовувати в комбінації з іншими типами сенсорів.

Однією з найпопулярніших моделей сенсорів відстані є HC-SR04 (рис. 2.6).



Рисунок 2.6 – Ультразвуковий сенсор відстані HC-SR04 [30]

Ультразвуковий сенсор відстані HC-SR04 має робочу напругу 3.8 - 5.5В, струм 8 мА, частоту 40 кГц, максимальну вимірювану дистанцію 1500 мм,

роздільну здатність 3 мм, ширину імпульсів 10 мкс, кут виміру 15 градусів, зовнішні габарити 37x20x15 мм та ціну 57 грн з ПДВ [30 - 31].

Для коректного виміру відстані за допомогою HC-SR04 необхідно надіслати короткий імпульс на вхідний Trig вхід сенсора. Після цього на виході Echo з'явиться сигнал відповідний виміряному відлунню раніше надісланого звукового сигналу [31].

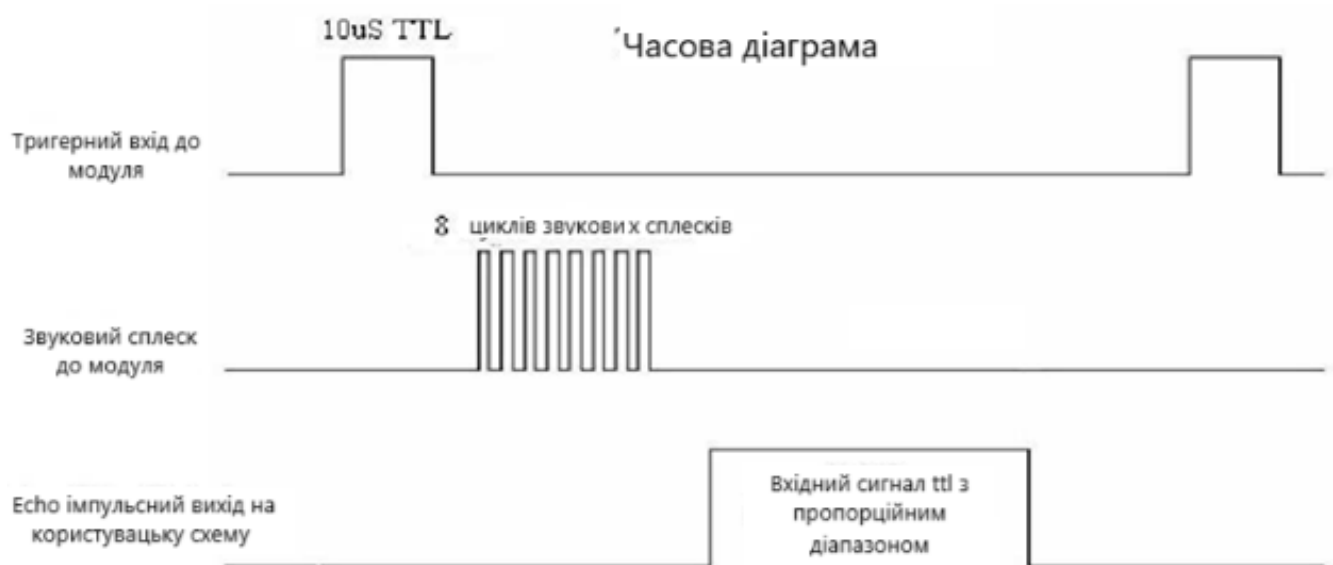


Рисунок 2.7 – Часова діаграма виміру HC-SR04 [31]

Це надає можливість виміру відстані оскільки швидкість звуку в повітрі відома, а отже:

$$r = tv/2, \tag{2.1}$$

де  $r$  – результат виміру відстані;

$t$  – часовий інтервал між надсиланням тригер сигналу та отриманням ехо ;

$v$  – швидкість звуку в повітрі, оскільки саме через повітря буде вимірюватись відстань.

Важливим при цьому є те, що довжина інтервалу буде в діапазоні мілісекунд, і залежно від засобу виміру часу інтервалу, може бути наданий в мілісекундах.

Для коректного виміру є важливим врахування відповідності вимірюваних величин, оскільки використання часу в мілісекундах, а швидкості згідно з міжнародною системою одиниць в метрах на секунду, призведе до результату у тисячу разів більшого від бажаного.

Стабільність та поширеність цієї моделі, разом з простотою використання, доступ до багатьох навчальних матеріалів і прикладів та достатньо великою максимальною вимірною дистанцією робить HC-SR04 хорошим вибором для цілей виміру рівня води.

З метою додаткової безпеки та функціональності можна використати обидва вище описаних сенсори рівня води. Оскільки поплавкові сенсори мають значно більшу надійність навіть в порівнянні з HC-SR04, але не можуть вимірювати рівень води з значною точністю, виникає можливість застосувати обидва сенсори.

У випадку використання ZP4510 в двох екземплярах разом з HC-SR04, система має можливість вимірювати рівень води з великою точністю. Але навіть у випадку якщо один з сенсорів вийде з ладу, два інших зможуть визначитись про правильне вимірювання.

Оскільки це надзвичайно мало ймовірно, що 2 сенсори, що використовують різні методи вимірювання, вийдуть з ладу в один і той самий момент, і будуть давати однакові показники.

Своєю чергою ZP4510, що можуть показувати лише невелику точність виміру, і через це потенційно могли б вийти з ладу і досі погоджуватись в своїх показниках компенсують це своєю надзвичайною надійністю та довговічністю.

У випадку ж якщо HC-SR04 вийде з ладу першим, система може порівняти його показники з ZP4510, і у випадку виявлення проблеми перейти в режим недовіри до нього, і перестати використовувати функції що потребують точного виміру.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

Альтернативним підходом може являтися використання 3 копій ZP4510 розташованих на різних рівнях резервуару, даючи підвищену точність вимірювання, хоч і меншу за HC-SR04, з використанням більш довговічного сенсора та дешевшою ціною. При цьому можливість порівнювати покази сенсорів між собою для визначення розбіжності залишається.

Отож для використання вище описаних сильних сторін сенсорів рівня води ZP4510 та ультразвукового сенсора відстані HC-SR04 у питанні виміру кількості рідини в місткості обрано використати два екземпляри ZP4510 та один екземпляр HC-SR04.

## 2.5 Вибір сенсора тиску

Для забезпечення якісного функціонування системи постачання води, тиск в трубах має підтримуватись на відповідному рівні. Для цього виникає потреба в сенсорі тиску для вимірювання тиску води в трубах і ввімкненню насосної станції за потреби.

Потенційним розв'язанням цієї проблеми могло б бути встановлення сенсора тиску води в трубі, зчитування його показників та порівняння з бажаними показниками тиску в трубах. Але такий метод додає складності, а звідси і потенційних помилок в роботі системи. На щастя існує легше і широко розповсюджене рішення цієї проблеми за допомогою реле тиску.

Реле тиску – це пристрій призначений для передавання електричного сигналу у випадку якщо робоча речовина досягла певного рівня тиску [32]. У випадку систем водопостачання реле тиску найчастіше використовується для ввімкнення подачі води в секцію труб при падінні тиску в ній нижче певного рівня.

Це значно спрощує використання подібних сенсорів [33] (рис. 2.8) в системах водопостачання, залишаючи також можливість не пропустити сигнал залежно від інших факторів системи.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

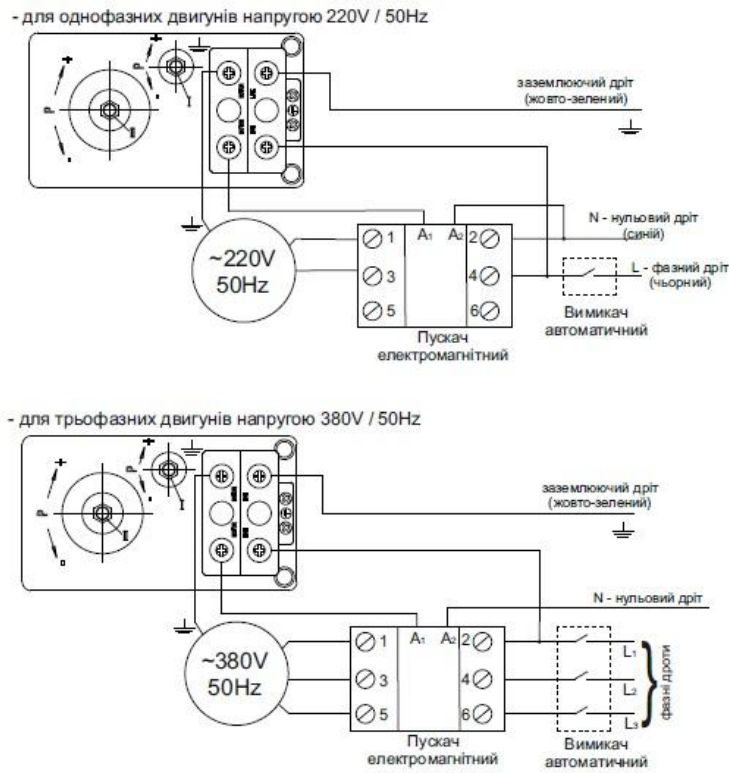


Рисунок 2.8 – Електрична схема підключення реле тиску [33]

Своєю чергою існують різні типи реле тиску: механічні пружинні, механічні мембранні, електронні базові, електронні з регульованим гістерезисом, електронні програмовані, електронні з частотним перетворювачем [34].

Механічне пружинне відрізняється простотою конструкції, низькою ціною та невеликою точністю вимірювання. Механічне мембранне має підвищену чутливість, схильність до зносу важливих компонентів, що призводить до меншої надійності та підвищену вартість.

Електронне базове реле має високу точність вимірювання, візуальну індикацію та захист від сухого ходу. Електронне з регульованим гістерезисом надає можливість більш детального встановлення параметрів вмикання та вимикання насосної станції. Електронне програмоване має розширені функції захисту, високу точність та можливість програмувати режими роботи. Електронне з частотним перетворювачем надає можливість плавного пуску насоса, економії енергії та регулювання продуктивністю.

Розглянемо можливості та характеристики реле тиску з частотним перетворювачем від одного з провідних виробників світу [34] (рис. 2.9).



Рисунок 2.9 – Електронний регулятор тиску EASYSMALL -2М 1.5 BAR Pedrollo [35]

Виробник Pedrollo має в рамках сучасного ринку різноманітних реле тиску має хороший баланс між доступністю, ціною та технічними характеристиками випусканих екземплярів продукції. До основних характеристик EASYSMALL -2М 1.5 BAR Pedrollo відноситься [35] (табл. 2.3)

Таблиця 2.3 – Таблиця характеристик EASYSMALL -2М 1.5 BAR Pedrollo

Параметр	Опис
Максимальна подача	200 л / хв. (12 м <sup>3</sup> / год.)
Робочий тиск	10 бар

### Кінець таблиці 2.3

Тиск включення (пусковий тиск)	1.5 бар
Температура рідини	до +50 ° С
Температура навколишнього середовища	до +40 ° С
Руйнівний внутрішній тиск	40 бар
Ступінь захисту	IP 65
Напруга	230 В
Частота	50/60 Гц
Максимальний струм	16 А
Ціна	2 448.00 грн

Враховуючи прекрасні технічні характеристики та репутацію виробника про надійність продукту, це реле тиску задовольняє потребам системи водопостачання приватного господарства. Також враховуючи, що при спробі модернізації наявного водопостачання існує можливість того, що користувач вже має реле тиску з перевіркою на сухий хід, заміна пристрою може бути не обов'язковою.

Оскільки через високу надійність сучасних електронних реле тиску та порівняно велику ціну деяких моделей можна використовувати тільки один сенсор, не хвилюючись за безпеку.

### 2.6 Вибір дисплея

Для ефективної, швидкої та надійної взаємодії системи з користувачем виникає потреба в передачі інформації про стан системи користувачу. Одним з підходів до розв'язання цього питання є використання дисплея для відображення необхідної інформації.

Особливості системи водопостачання формують ряд вимог до дисплею. Дисплей повинен бути невеликого розміру, мати невелике енергоспоживання та

передбачати можливість легкого використання в комбінації з обраним мікроконтролером.

Оскільки ESP32-DevKitC V4 має послідовну шину даних I2C це надає можливість використання сумісного з цим дисплея. Додатково використання цієї технології дозволяє зменшити потенційну кількість інформаційних з'єднань з дисплеєм до двох.

I2C використовує дві лінії передачі інформації, обидві з яких є двонапрямними. До них відносяться SDA послідовна лінія даних та SCL послідовна лінія тактування.

Важливість зменшення кількості необхідних виводів для передачі інформації між дисплеєм та мікроконтролером пов'язана з обмеженістю його можливостей до під'єднання великої кількості периферійних пристроїв без ускладнення системи різноманітними альтернативними розв'язання даного питання. Також це пов'язано з кількістю вже вище розглянутих периферійних пристроїв та необхідних зв'язків.

Одним з популярних рішень подібної задачі, що відповідає висунутим вимогам є 20×4 2004 LCD модуль з I2C послідовним інтерфейсом [38].

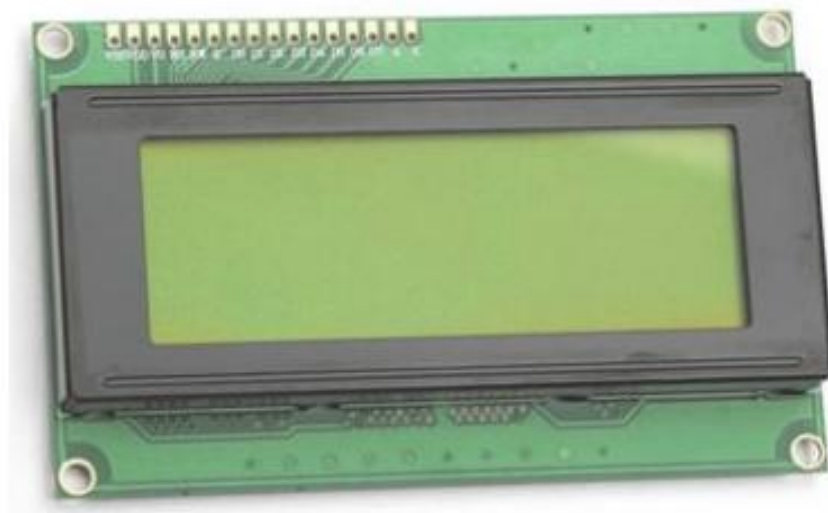


Рисунок 2.10 – Модуль LCD 20×4 2004 з I2C послідовним інтерфейсом [38]

Цей дисплей може використовувати джерело живлення з напругою 5В, має відносно просту конструкцію, що призводить до доступності та дешевизни елемента. Також важливою характеристикою є довговічність яка відповідно досягається цим дисплеєм за допомогою простоти конструкції та поширеності, що призводить до значної кількості інформації про використання доступної, як розробникам, так і користувачам.

Для повноцінного використання подібного дисплею використовується контролер монохромних рідкокристалічних знакосинтезуючих дисплеїв HD44780 розроблений компанією Hitachi [38]. Вони відзначаються надзвичайною поширеністю та низькою ціною.



Рисунок 2.11 – Контролер монохромних рідкокристалічних знакосинтезуючих дисплеїв HD44780 [38]

Для їх сумісного використання необхідно під'єднати відповідні 16 виводів обох пристроїв один до одного. Своєю чергою чотири виводи HD44780 мають бути під'єднаними в відповідності до їх призначення. Інформаційні виводи SDA та SCL під'єднуються до мікроконтролера для передачі інформації, а решта виводів відповідають за живлення. Але зазвичай обидва пристрої вже є коректно під'єднаними один до одного при закупівлі.

Отож для виконання функції надання необхідної інформації, як стан справності периферійних пристроїв згідно порівнянь вимірів сенсорів системою, користувачу за якісні технічні характеристики, задовільні можливості та сумісність з обраним мікроконтролером без надмірного використання його виводів обираємо 20×4 2004 LCD модуль з I2C послідовним інтерфейсом.

## 2.7 Висновки до другого розділу

У межах розділу було проведено аналіз наявних та часто використовуваних елементів звичайних та кіберфізичних систем водопостачання, які необхідні для реалізації системи згідно з розробленою структурною схемою.

Для мікроконтролера який буде виконувати функції керування всією системою було обрано ESP32-DevKitC V4 за хороший баланс ціни та технічних характеристик, низьке енергоспоживання та прийнятні розміри і наявність Wi-Fi модуля для зв'язку.

З метою виконання функції виміру температури насосної станції було обрано сенсор DS18B20 який дещо поступається технічними характеристиками розглянутим альтернативним рішенням, але має значно меншу ціну.

Для виміру рівня води елементів системи призначеного для її тимчасового зберігання та зменшення навантаження на насосну станцію при короткочасному різкому збільшенні потреби в воді було обрано використати комбінацію з двох сенсорів ZP4510 та ультразвукового сенсор відстані HC-SR04, що дозволить досягнути підвищену точність та надійність системи.

В якості реле тиску було обрано EASYSMALL -2M 1.5 BAR від компанії Pedrollo, що має можливість захисту від сухого ходу підвищуючи надійність системи та високу надійність з хорошими значеннями технічних характеристик.

В якості дисплея було обрано 20×4 2004 LCD модуль з I2C послідовним інтерфейсом з вбудованим контролером дисплеїв HD44780, що надає можливість виводити необхідну інформацію користувачу для покращення процесу експлуатації.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ ВОДОПОСТАЧАННЯ ДОМОГОСПОДАРСТВА

### 3.1 Опис реалізації симуляції системи водопостачання в Wokwi

Програмне забезпечення є не лише ефективним інструментом керування кіберфізичними системами, але і корисним інструментом для симуляції їх роботи. Використання такого підходу дозволяє зменшити фінансове навантаження на розробку завдяки витраті коштів лише після успішного проектування, що дозволяє перевірити можливі недоліки системи перед її реалізацією та експлуатацією.

Це дозволяє запобігти потенційних матеріальних збитків у випадку неправильного функціонування системи, та додатково дозволяє перевірити її роботу в небезпечних ситуаціях без суттєвого ризику. Оскільки система водопостачання взаємодіє з обладнанням, яке є чутливим до ситуації даний підхід є надзвичайно корисним. Для симуляції системи водопостачання було обрано симулятор Wokwi за зручний інтерфейс та широкі можливості симулювання конкретних моделей мікроконтролерів та їх взаємодію з елементами. Для зручного процесу розробки симуляції необхідно завантажити розширення PlatformIO (рис. 3.1). та Wokwi для VS code.

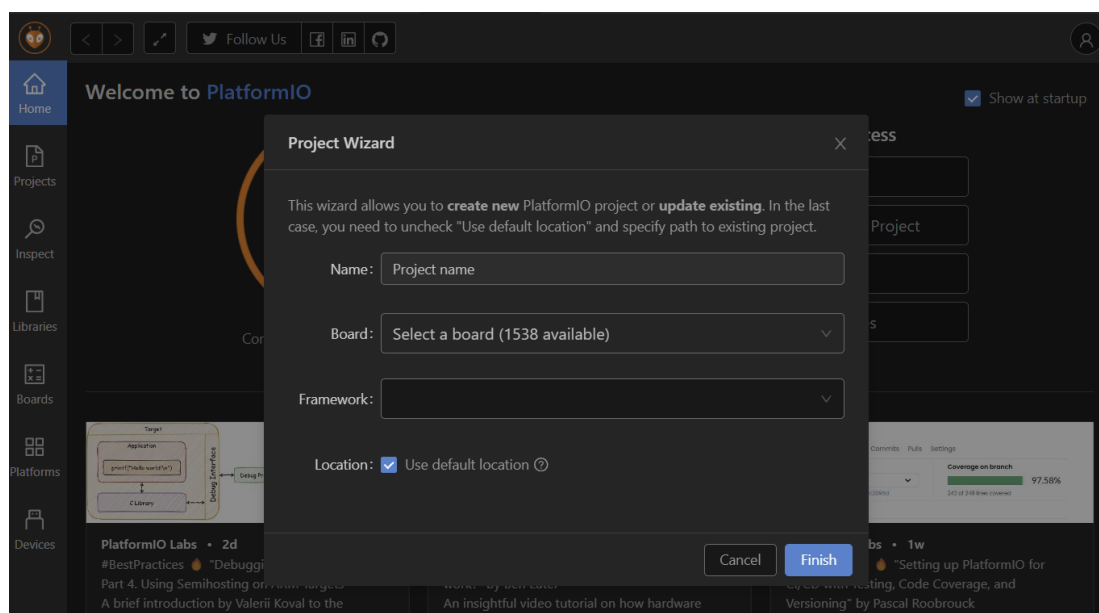


Рисунок 3.1 – Інтерфейс створення нового проекту за допомогою PlatformIO

Після вибору необхідної плати мікроконтролера PlatformIO згенерує проєкт до якого можна буде додати wokwi.toml та diagram.json файли з описом необхідних даних для виконання симуляції.

```
ESP32 Water supply system > wokwi.toml
1  [wokwi]
2  version = 1
3  firmware = '.pio/build/esp32dev/firmware.bin'
4  elf = '.pio/build/esp32dev/firmware.elf'
5  rfc2217ServerPort = 4000
```

Рисунок 3.2 – Необхідний вміст для файлу wokwi.toml

У файл wokwi.toml необхідно помістити інформацію про відносне розміщення скомпільованого файлу програми. Своєю чергою diagram.json повинен містити опис всіх елементів системи водопостачання включно з їх параметрами та зв'язками між ними, згідно з офіційною документацією від розробників симулятора. Але онлайн версія симулятора надає можливість модифікувати diagram.json автоматично з використанням зручного інтерфейсу.

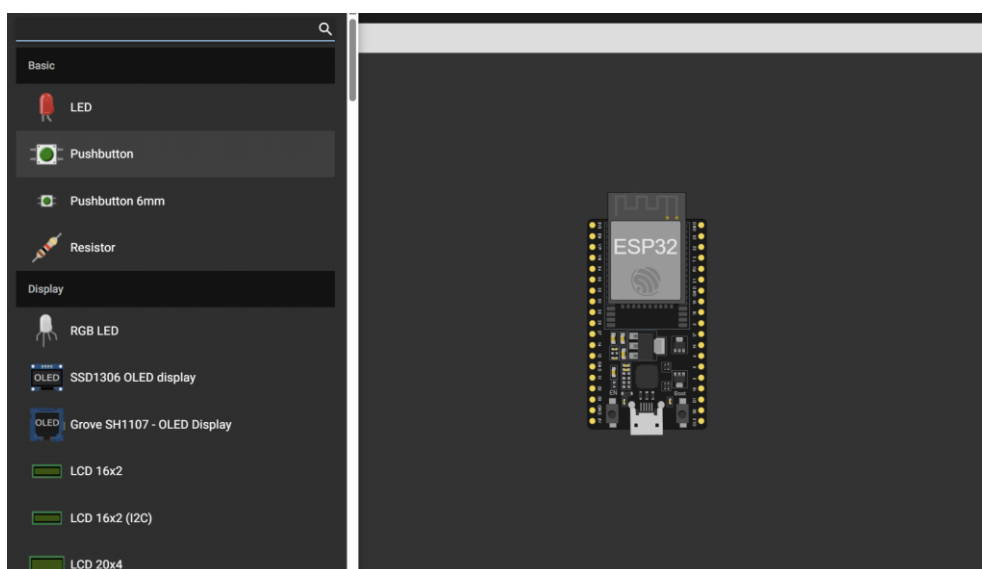


Рисунок 3.3 – Візуальний інтерфейс Wokwi

Після коректного налаштування зв'язків між елементами вміст файлу `diagram.json` може бути просто скопійованим у відповідника в VS code.

Не зважаючи на те, що безоплатна версія Wokwi в VS code не дозволяє використовувати візуальний інтерфейс для автоматичної зміни зв'язків в системі, їх перегляд є доступним. В результаті було розроблено систему (додаток Б, В), що реалізує структуру описану у розділі 2.

Отже, загальний функціонал кіберфізичної системи згідно з поставленим завдання описується формулою (3.1):

$$\{y_1, y_2, y_3\} = G[x_1, x_2, y_1, y_2, y_3], \quad (3.1)$$

де  $y_1, y_2, y_3$  – тиск вхідного і двох вихідних насосів;

$x_1, x_2$  – відсотки відкриття труб двома групами користувачів (умовно вважаємо, що це дім і город);

$G$  – функціонал системи автоматичного керування, яка включає математичні моделі насосів та логіку контролерів, описану в блок-схемах алгоритмів.

Перед будь-яким запуском системи відбувається період початкового її налаштування. В цей процес входить початкова ініціалізація LCD дисплея встановлення режимів роботи для необхідних виводів мікроконтролера, як INPUT та OUTPUT.

Після цього встановлюються початкові значення виходів таким чином, що усі насоси не отримують живлення, та відбувається під'єднання до мережі Wi-Fi та MQTT.

Фінальним етапом налаштування є визначення необхідних для роботи ідентифікаторів сенсорів температур. Увесь вище описаний процес відбувається автоматично, і символічно зображений на діаграмі діяльності на рисунку 3.4.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 46
Зм.	Арк.	№ докум.	Підпис	Дата		

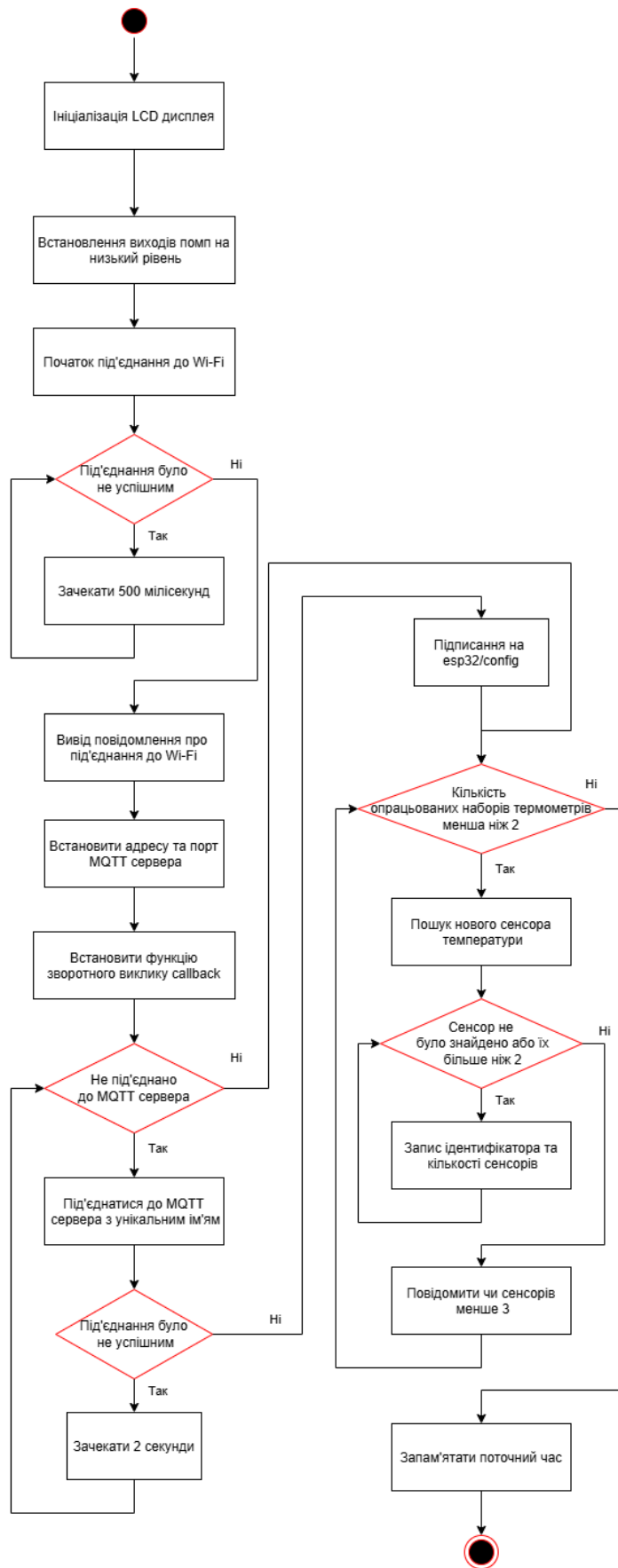


Рисунок 3.4 – Діаграма діяльності початкового процесу налаштування системи (додаток Б)

Розроблена система складається з декількох основних частин, що відповідають за взаємодію з користувачем, вимірювання рівня води в резервуарі, вимірювання температури насосної станції та контролю роботи як насосної станції, так і свердловинного насоса.

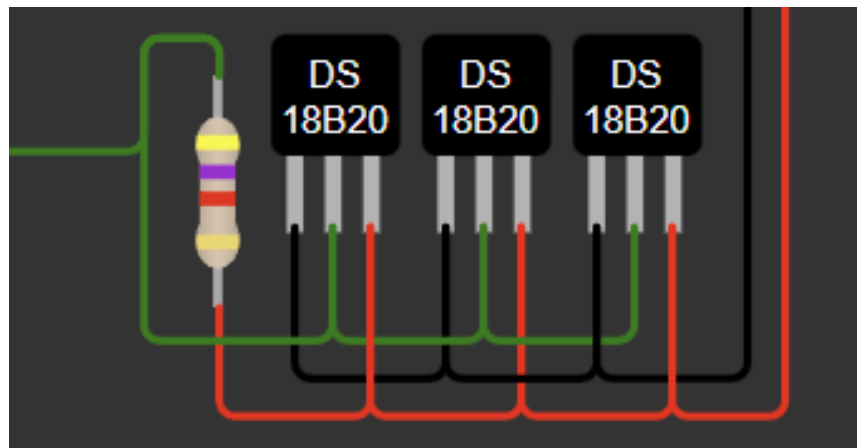


Рисунок 3.5 – Частина системи, що відповідає за вимірювання температури насосної станції (додаток В)

Частина системи, що відповідає за вимірювання температури насосної станції поєднує в собі три сенсори температури DS18B20, що під'єднуються до мікроконтролера для передачі даних через один провід використовуючи технологію OneWire. Завдяки бібліотекам OneWire.h та DallasTemperature.h було отримано можливість використати команди `OneWire.search()` для отримання адрес індивідуальних сенсорів температури. Після чого використання комбінації команд `DallasTemperature.requestTemperatures()` та `DallasTemperature.getTempC()` надає можливість отримати покази індивідуальних значень температури, що були виміряні кожним з сенсорів.

Такий підхід надає можливість помітно зменшити кількість використаних з'єднань. Однак ключовим моментом використання цієї технології є під'єднання спільного живлення сенсорів до спільного інформаційного виводу через резистор.

Наявність трьох сенсорів температури призводить до виникнення принципового питання щодо того, який із сенсорів вийшов з ладу, і яке результуюче значення температури потрібно обрати.

Для цього після отримання інформації від кожного сенсору про виміряну температуру, необхідно помістити відповідну інформацію у тимчасовий `std::vector` та відсортувати його.

Відсортований набір температур надає можливість зробити легкий пошук найбільшого набору температур, що були виміряні та не відрізняються один від одного на більше ніж похибка сенсорів. Це реалізовується через перебори циклом всіх можливих початкових температур в послідовностях. Для кожної ж початкової температури підбирається максимальна довжина послідовності в якій останній елемент задовольняє критерію того, що різниця температур не перевищує потенційну похибку сенсорів. Оскільки початковий `std::vector` був відсортованим, то нам відомо, що для всіх проміжних елементів послідовності різниця температур буде ще меншою.

У випадку якщо нова глобально найдовша послідовність знайдена, її індекс початкової температури та довжина запам'ятовується.

Таким чином в  $O(n^2)$  вдається знайти найбільшу підмножину множини усіх виміряних температур таку, що для будь-яких двох елементів підмножини модуль їх різниці буде меншим ніж похибка виміру сенсорів.

Після визначення температур, що узгоджуються між собою у своїх вимірах ми можемо визначити, що усі сенсори, що виміряли значення яке не було знайдено в найдовшій підмножині скоріш за все зіпсовані та не можуть бути надійними в майбутньому. Тому при наступних перевірках для розрахунків будуть використовуватись лише ті сенсори у яких ми ще не виявили несправність.

Вище описаний алгоритм зображено на діаграмі діяльності на рисунку 3.6.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

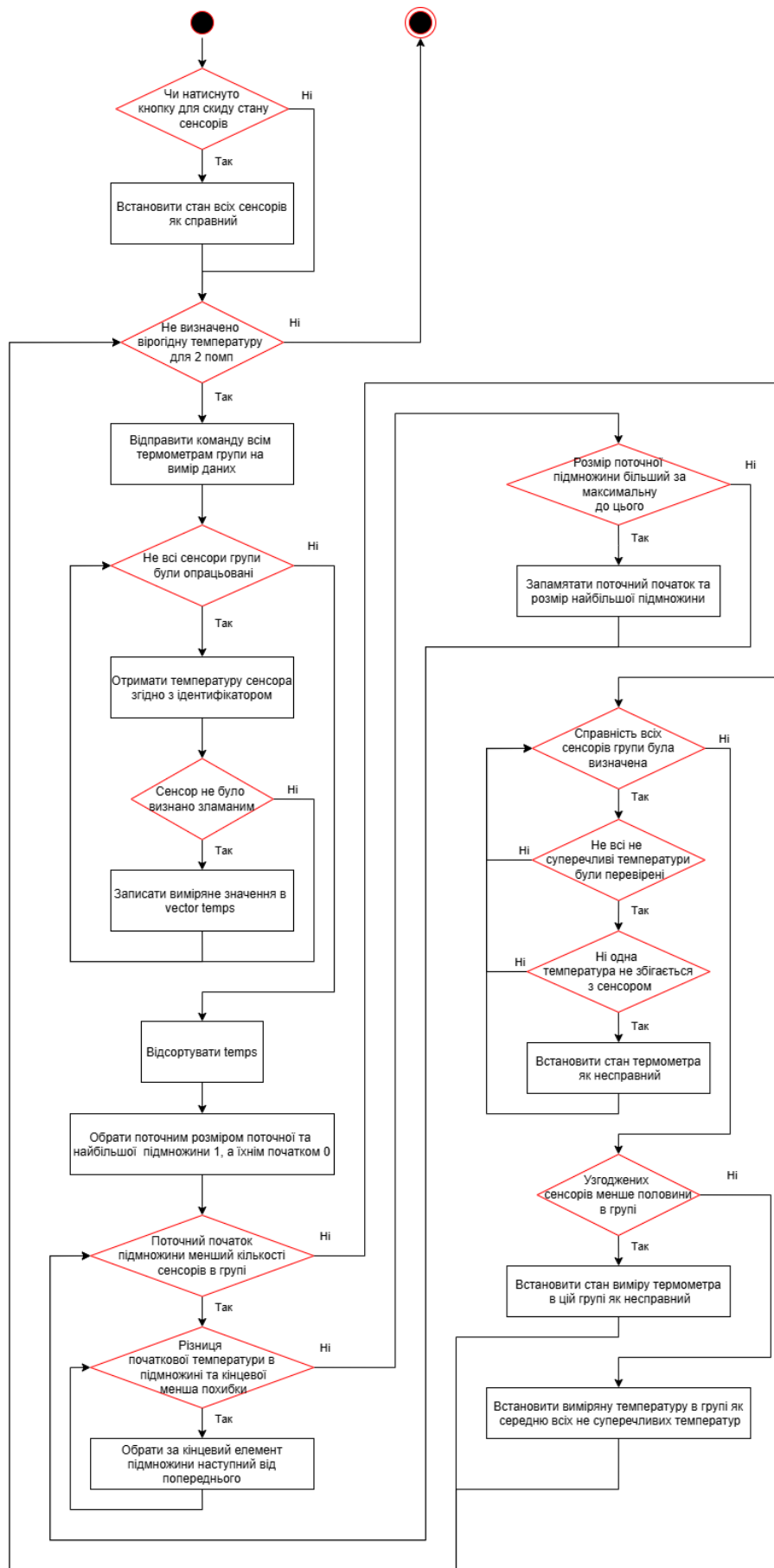


Рисунок 3.6 – Діаграма діяльності вимірювання температури (додаток Б)

Зм.	Арк.	№ докум.	Підпис	Дата

Отже, вище описаний алгоритм знаходить результат, що відповідає формулі (3.2):

$$(S_{max} \subseteq A) \cap (\forall x, y \in S_{max} (|x - y| \leq a)) \cap \\ \cap (\forall T \subseteq A ((\forall x, y \in T (|x - y| \leq a)) \rightarrow |T| \leq |S_{max}|)), \quad (3.2)$$

де  $S_{max}$  – найбільшого набору температур, що були виміряні та не відрізняються один від одного на більше ніж похибка сенсорів;

$a$  – максимальна можлива похибка термометрів, згідно їх технічних характеристик;

$A$  – початкова множина всіх виміряних температур;

$T$  – будь-який набір виміряних температур;

$x$  – будь-яка виміряна температура;

$y$  – будь-яка виміряна температура.

Після встановлення правдивості значень сенсорів, у випадку якщо найдовша підмножина містила більш ніж половину усіх сенсорів, ми можемо використати їх середнє арифметичне як ймовірну температуру.

Але якщо необхідна кількість сенсорів, що узгоджуються між собою в вимірах не було знайдено, або ймовірна температура занадто висока для безпечного функціонування насосної станції система прийме рішення про зупинення перекачування води незалежно від тиску в трубах.

Перевагою цього підходу є його незалежність від кількості сенсорів температури.

Також частина системи, що відповідає за вимірювання рівня води, вирішує аналогічну, але суттєво відмінну проблему (рис. 3.7).

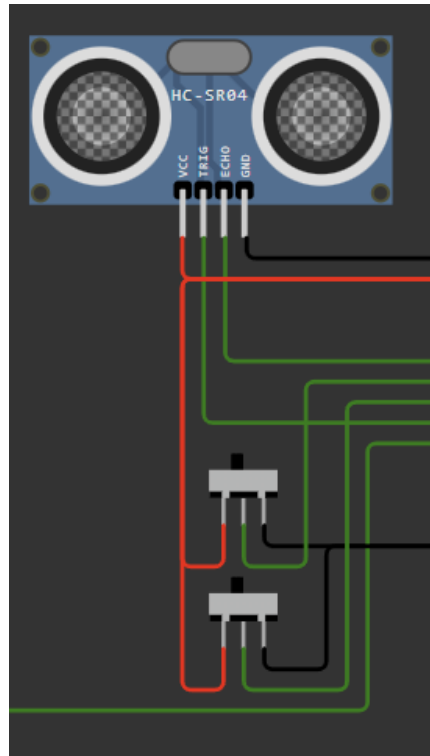


Рисунок 3.7 – Частина системи, що відповідає за вимірювання рівня води в резервуарі для її зберігання (додаток В)

Функціонування цієї частини системи базується на двох сенсорах рівня води ZP4510, що у зв'язку з їх відсутністю у Wokwi та надзвичайною простотою в рамках симуляції можуть бути замінені на перемикачі, та ультразвуковий сенсор відстані HC-SR04. HC-SR04 вимірює відстань до води за допомогою стандартного методу. Система генерує короткий сигнал низького рівня на вхід TRIG для того, щоб далі сформувати на 10 мілісекунд.

Після цього результат функції pulseIn() сигналу на ECHO сигналізує про час який був необхідний звуку для поширення у прямому та зворотному напрямку: до і від об'єкту до якого вимірюється відстань. Далі здійснюється перетворення цього результату у відстань до води, а потім у висоту води залежно від висоти встановлення сенсора відстані яке є тривіальним.

Але оскільки інформація від сенсорів ZP4510 надається у вигляді того, чи досяг рівень води місця їх встановлення, а HC-SR04 надає точну інформацію про її рівень, це ускладнює перевірку на відмінність результатів виміру між сенсорами.

В першу чергу система перевіряє кожну пару сенсорів на те, чи суперечать їхні покази один одному. У випадку двох ZP4510, це відбувається лише коли нижчий з них повідомляє про те що вода не досягла вимірюваного рівня, а вищий повідомляє що вода його вже досягла.

Натомість на суперечність між HC-SR04 та одним з ZP4510 вказує те, що або рівень води, виміряний HC-SR04, є меншим за рівень встановлення відповідного ZP4510, за умови що він повідомляє про досягнення цього рівня водою, або навпаки при відсутності повідомлення про досягнення рівня води згідно з показами ZP4510, сенсор відстані повідомляє про вищий рівень води.

Після отримання інформації про те які з сенсорів суперечать один одному система може прийняти рішення, що якщо попередньо усі сенсори були придатними для експлуатації, то у випадку якщо ZP4510 не суперечать один одному і хоча б один з них суперечить HC-SR04, то HC-SR04 скоріш за все є несправним. Але якщо ZP4510 суперечать один одному, то той з них який узгоджується з сенсором відстані скоріш за все показує правдиву інформацію.

Своєю чергою якщо двоє попередньо не зіпсованих сенсорів суперечать один одному, оскільки випадок коли третій сенсор для вирішення розбіжності був справним, вже було розглянуто, вони обидва запам'ятовуються системою як не надійні сенсори. Після цього етапу система визначила, яким показам можна довіряти в рамках виміру рівня води. Отже, далі вона визначає найбільш імовірний результуючий рівень води, базуючись на вимірах індивідуальних сенсорів.

Якщо ні один з сенсорів не є надійним, вимірювання води у загальному вважається ненадійним. Це призводить до зупинки усіх насосів. У випадку якщо лише два сенсори надають не суперечливі дані та один з них є сенсором відстані, то через його вищу точність, обирається його вимірювання як найбільш ймовірний.

Але якщо лише ZP4510 надають ймовірну інформацію, то обирається середнє арифметичне їх висот встановлення або їх максимум залежно від того чи показують вони обидва досягнення цільового рівня води.

Вище описаний алгоритм зображено на діаграмі діяльності на рисунку 3.8.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		

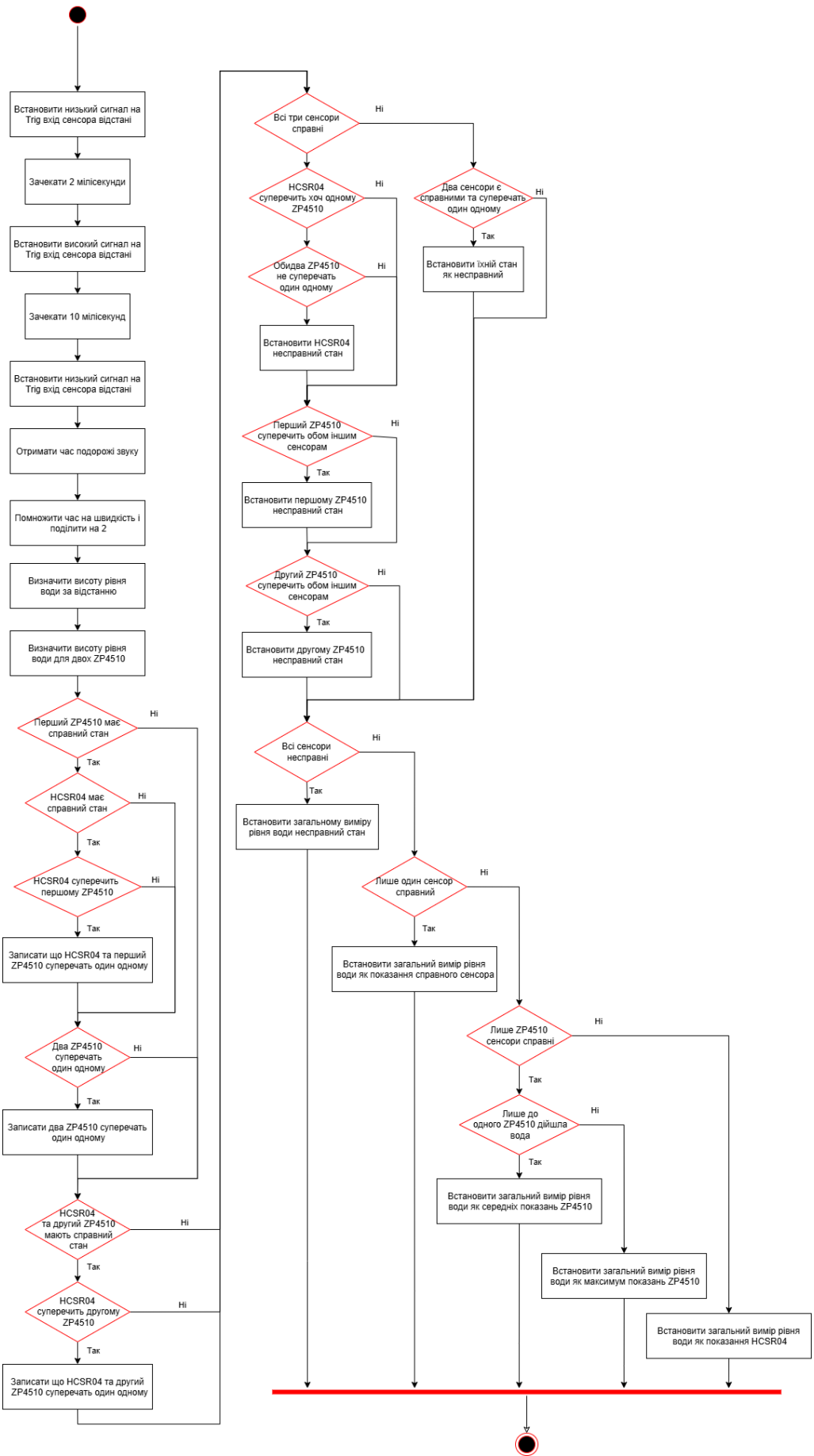


Рисунок 3.8 – Діаграма діяльності вимірювання рівня води (додаток Б)

Таким чином, перед вимірюванням показників сенсорів система повинна вирішити чи можна знову довіряти сенсорам, для яких попередньо були виявлені суперечності з іншими сенсорами. Такі функції виконує частина системи, яка відповідає за взаємодію з користувачем (рис. 3.9).

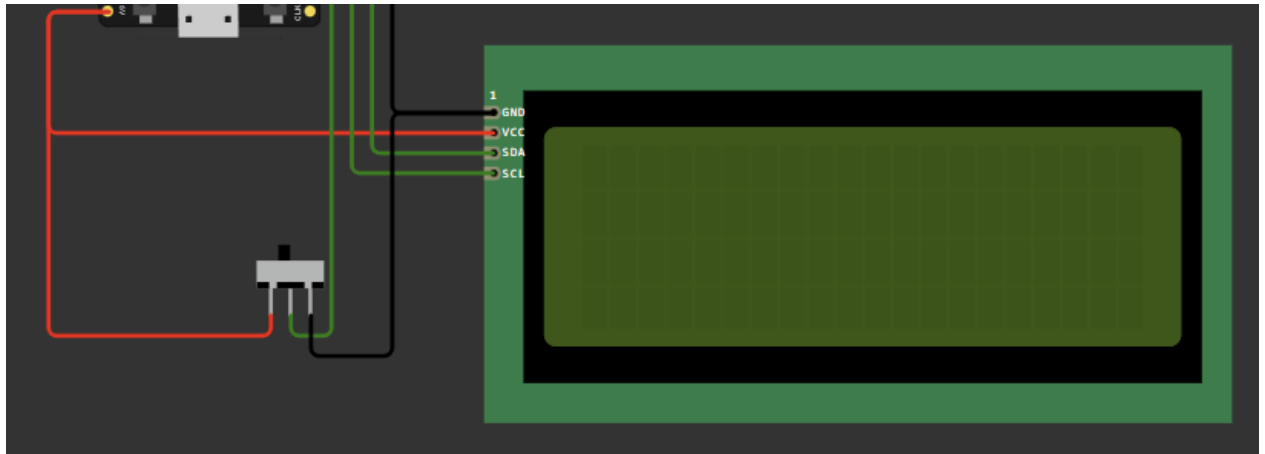


Рисунок 3.9 – Частина системи, що взаємодіє з користувачем (додаток В)

Функціонування цієї частини системи базується на перемикачі який у випадку натискання повідомляє систему про те, що користувач змінив сенсори які вийшли з ладу на нові, або з іншої причини, бажає видалити інформацію про те, які з сенсорів суперечать іншим і не можуть бути використані для отримання ймовірних даних. Також система виводить інформацію про те, які з сенсорів, скоріше за все, вийшли з ладу на дисплей, для зручного надання користувачу цієї інформації.

Система контролює вмикання та вимикання насосних станцій завдяки контролю їх живлення через реле. Безпечність вмикання для насосу призначеного для перекачування води з свердловини до місткості для її зберігання базується лише на ймовірну виміряному значенні рівня води в місткості. Якщо рівень води занадто великий, то насос вимикається. Своєю чергою насоси, що займаються перекачуванням води з місткості в труби беруть до уваги власну температуру разом з рівнем води. Але рівень води, на відміну від зовнішнього насоса, для увімкнення має бути вищим ніж певне значення, задане системою для кожного з внутрішніх насосів індивідуально. Цей алгоритм описано більш детально на рисунку 3.10.

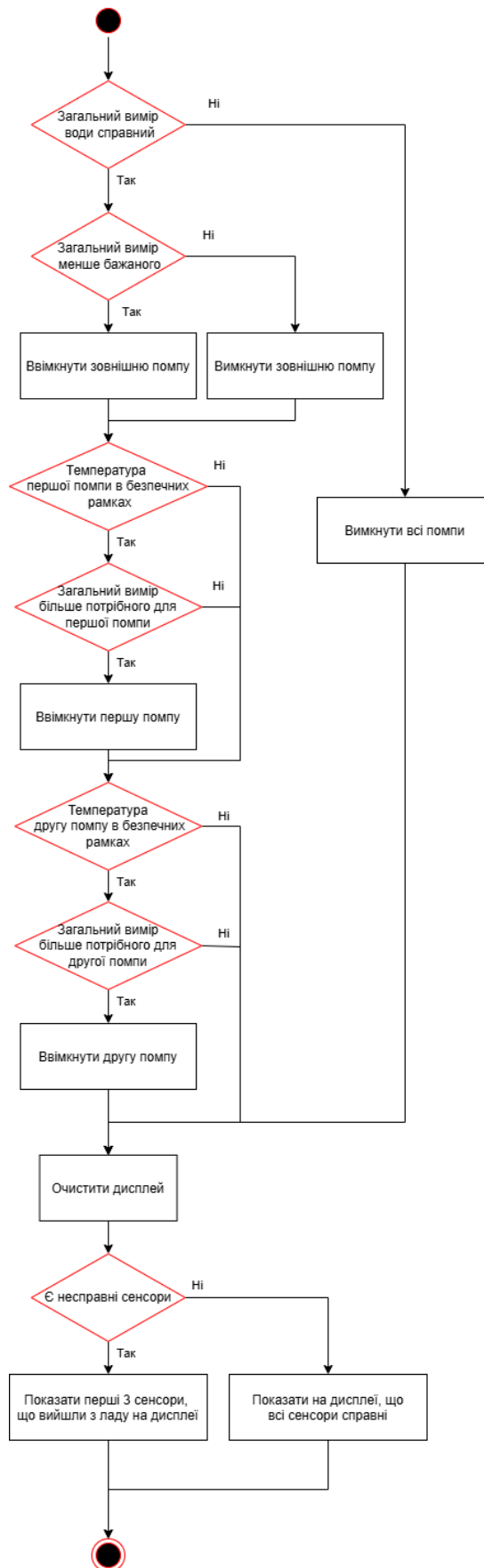


Рисунок 3.10 – Діаграма діяльності вимкнення насосу та виводу інформації  
(додаток Б)

Далі напруга живлення для внутрішніх насосів має пройти через реле тиску, яке в симуляції в зв'язку з обмеження програмного забезпечення Wokwi симулюється через комбінацію ще одного реле та вимикача.

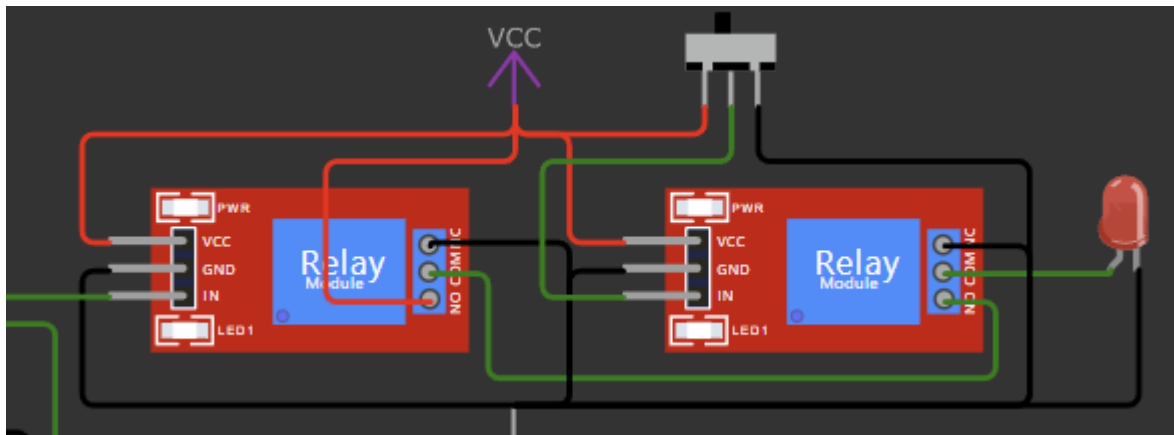


Рисунок 3.11 – Частина системи, що контролює роботу насосної станції (додаток В)

У випадку якщо насосну станцію безпечно ввімкнути, навіть якщо у цьому не має потреби, живлення буде подано системою через ліве реле. Своєю чергою праве реле та перемикач, імітують роботу реле тиску, не пропустять живлення для насосу якщо в цьому немає потреби. В результаті лише якщо для обидвох реле виконується умова ввімкнення насосної станції, то вона запуститься.

Фінальним етапом циклу роботи програмного забезпечення системи, що працює на мікроконтролері, є відправлення інформації про стан системи за допомогою MQTT протоколу. У випадку, якщо не пройшло години після попереднього часу коли він був збережений, то весь цей етап пропускається і система просто очікує 5 секунд. У зворотному випадку спочатку запам'ятовується новий час, з яким у майбутньому буде проводитись порівняння. Після цього надсилається за допомогою MQTT повідомлення «NEW», що сигналізує про початок нової сесії передавання даних сенсорів.

Одразу після цього протягом 5 секунд система намагається отримати відповідь про налаштування власних параметрів. При успішному отриманні

відповіді система призначає нові значення для таких параметрів, як висота встановлення сенсора відстані та його можлива максимальна похибка, висоти встановлення обох сенсорів ZP4510, максимальна похибка термометрів, рівень води в місткості який система буде намагатись досягти, максимально припустима температура насосів та рівень води, починаючи з якого насоси зможуть перекачувати воду з місткості у труби.

Після успішного або неуспішного отримання цієї інформації, система починає відправляти виміряне значення кожного з сенсорів, та інформацію про їхню справність, аналогічні дані для загальної ймовірної температури кожного з внутрішніх насосів та рівень води та чи можна довіряти цій інформації.

Діаграма діяльності цього процесу зображена на рисунку 3.12.

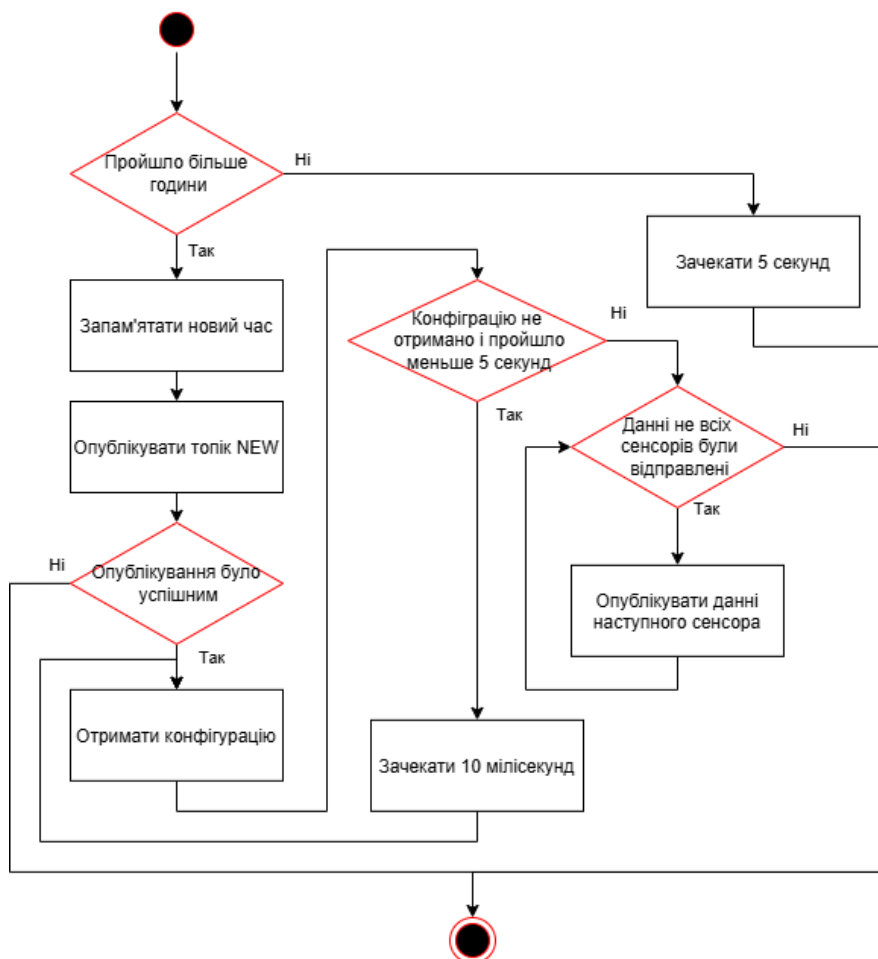


Рисунок 3.12 – Діаграма діяльності відправлення даних і про стан системи та отримання параметрів налаштування (додаток Б)

Після цього система повертається на етап перевірки необхідності оновлення інформації про стан сенсорів (рис. 3.6).

### 3.2 Опис програмного забезпечення ситеми водопостачання для збереження інформації про стан системи

Програмне забезпечення, призначене для запису інформації про стан системи, для коректної роботи повинно виконуватись на окремому пристрої. Воно виконує дії описані в відповідній діаграмі діяльності (додаток В).

На початку роботи програми, відбувається під'єднання до MQTT брокера, після чого при отриманні повідомлення на топик, на який підписана програма, відбувається основна робота.

Спочатку програма очікує повідомлення «NEW», що сигналізує про початок надсилання інформації для нового рядка. Після цього програма зчитує з таблиці налаштування системи та публікує їх через MQTT для того, щоб мікроконтролер міг оновити свої налаштування.

Своєю чергою усі наступні повідомлення які не є «NEW» повідомленням будуть інтерпретуватись як інформація про один сенсор. Цю інформацію буде записано згідно з ім'ям сенсора або виміру величини. У випадку отримання усієї інформації з очікуваного списку, її буде записано у поточний рядок таблиці. Після чого поточний рядок збільшиться на один, але при перевищенні максимального повернеться до першого запису станів системи у 4 рядку.

У випадку, якщо кількість записів про стан системи перевищує максимальну, яка вказана у самій таблиці, надлишкові рядки буде видалено.

У випадку коли інформація не про весь стан системи надійде до програми, то нове «NEW» повідомлення призведе до запису наявної інформації.

В результаті роботи програми для збереження інформації про стан системи, разом з самою системою за деякий час можна отримати таблицю, зображену на

рисунок 3.13, за винятком часу запису інформації, оскільки запис було прискорено з метою перевірки роботи програми.

Max rows	index	HCSR04 mount height	HCSR04 error	ZP4510_1 level	ZP4510_2 level	Temperature error	Desired water level	Pump max temp	Water level pump1	Water level pump2								
30	10	160.0	1.0	20.0	90.0	0.5	50.0	40.0	50.0	20.0								
Time Stamp	WaterLevel	ZP4510_1	ZP4510_2	HCSR04	Temperature1	DS18B20_1.0	DS18B20_1.1	DS18B20_1.2	Temperature2	DS18B20_2.0	DS18B20_2.1	DS18B20_2.2						
2025-05-31 08:49:03	TRUE	120.0500031	TRUE	20	TRUE	120.05000	TRUE	0	TRUE	0	TRUE	0	TRUE	0				
2025-05-31 08:49:21	TRUE	119.9819946	TRUE	20	TRUE	90	TRUE	119.98199	TRUE	-0.000528	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:49:39	TRUE	119.9649963	TRUE	20	TRUE	90	TRUE	119.96499	TRUE	-0.000528	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:49:57	TRUE	119.9649963	TRUE	20	TRUE	90	TRUE	119.96499	TRUE	-0.000528	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:50:15	TRUE	120.1009979	TRUE	20	TRUE	90	TRUE	120.10099	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-06-01 22:21:52	TRUE	120.0159988	TRUE	20	TRUE	100	TRUE	120.01599	TRUE	-0.000220	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 03:23:33	TRUE	120.0159988	TRUE	20	TRUE	90	TRUE	120.01599	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 03:23:40	TRUE	119.9649963	TRUE	20	TRUE	90	TRUE	119.96499	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 03:25:51	TRUE	120.2200012	TRUE	20	TRUE	90	TRUE	120.22000	TRUE	-0.000220	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 03:25:58	TRUE	119.9819946	TRUE	20	TRUE	90	TRUE	119.98199	TRUE	-0.000220	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 06:04:15	TRUE	120.0330048	TRUE	20	TRUE	100	TRUE	120.03300	TRUE	-0.000146	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 06:04:22	TRUE	120.0330048	TRUE	20	TRUE	100	TRUE	120.03300	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 06:04:28	TRUE	120.0159988	TRUE	20	TRUE	90	TRUE	120.01599	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 06:04:35	TRUE	119.9649963	TRUE	20	TRUE	90	TRUE	119.96499	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 06:04:42	TRUE	120.0159988	TRUE	20	TRUE	90	TRUE	120.01599	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 06:05:33	TRUE	120.0330048	TRUE	20	TRUE	100	TRUE	120.03300	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 06:05:51	TRUE	120.0159988	TRUE	20	TRUE	100	TRUE	120.01599	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:27:49	TRUE	120.0500031	TRUE	20	TRUE	100	TRUE	120.05000	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:28:07	TRUE	120.0670013	TRUE	20	TRUE	100	TRUE	120.06700	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:28:25	TRUE	120.0839996	TRUE	20	TRUE	100	TRUE	120.08399	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:29:20	TRUE	120.0330048	TRUE	20	TRUE	100	TRUE	120.03300	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:29:38	TRUE	120.0159988	TRUE	20	TRUE	100	TRUE	120.01599	TRUE	-0.000528	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:29:56	TRUE	119.9649963	TRUE	20	TRUE	90	TRUE	119.96499	TRUE	-0.000220	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:30:31	TRUE	119.9990005	TRUE	20	TRUE	100	TRUE	119.99900	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0
2025-05-31 08:37:57	TRUE	120.1009979	TRUE	20	TRUE	100	TRUE	120.10099	TRUE	0	TRUE	0	TRUE	0	TRUE	0	TRUE	0

Рисунок 3.13 – Таблиця з інформацією про стан системи

Надалі ці дані можна використовувати для виявлення небезпечних ситуацій, повідомлення користувача про сенсори, які вийшли з ладу, зміну параметрів, підготовку до режимів інтенсивного споживання води або навіть прогнозування потреби в воді, базуючись на її використанні в минулому і виявлення протікання водопроводу.

### 3.3 Опис реалізації симуляції системи водопостачання в MATLAB

Попередньо розроблена симуляція кіберфізичної системи водопостачання в Wokwi має один значний недолік. Незважаючи на те, що програмне забезпечення симулює роботу сенсорів та мікроконтролера відповідно до їх екземплярів в реальному світі, симуляція в програмному забезпеченні Wokwi не має зручного засобу замикання циклу виміру величин, прийняття рішень, зміни відповідних вимірюваних величин, без втручання користувача.

Таким чином Wokwi може успішно визначити, яке рішення система буде приймати при будь-якому стані вимірюваних величин, але не надає можливості

динамічно симулювати зміну цих вимірних величин базуючись на реакції системи. Таким чином виникає потреба в використанні програмного забезпечення, що здатне симулювати логіку взаємодії та зміни станів з часом насосів, систем труб, місткостей для води, її споживачів, свердловин та інших елементів. Зручним програмним забезпеченням для виконання таких операцій є MATLAB/Simulink.

Розроблена симуляція системи водопостачання у MATLAB/Simulink складається з декількох підсистем. Ці підсистеми можна віднести до декількох видів в залежності від типу реального пристрою, який вони симулюють. Такий підхід надає можливість легше розробляти, модифікувати та використовувати симуляцію.

Одним з найважливіших типів елементів в симуляції, яким відповідають підсистеми, є елементи що накопичують в собі воду. До таких елементів відносяться місткість для води (рис. 3.14) та системи труб.

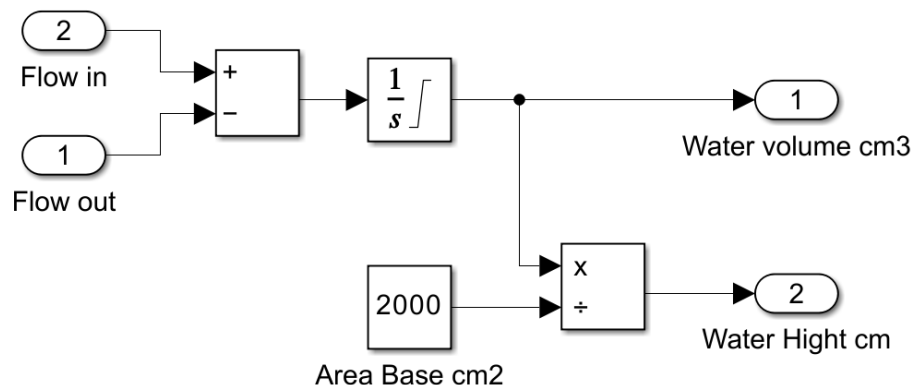


Рисунок 3.14 – Підсистема, що симулює місткість для води

Симуляція систем труб організована ідентично до симуляції місткості для води, але замість рівня води, як другого вихідного параметра, вона використовує тиск води у найнижчій точці відповідної системи труб. В рамках симуляції відношення між кількістю води та тиском вважається лінійним.

Отже, вище описаний алгоритм знаходить результат, що відповідає формулі (3.3):

$$z(t) = k \int_0^t x(\tau) - y(\tau) d\tau, \quad (3.3)$$

де  $z(t)$  – значення виводу підсистеми у певний момент;

$x(\tau)$  – швидкість надходження води до підсистеми у певний момент;

$y(\tau)$  – швидкість потоку води, що покидає підсистему у певний момент;

$t$  – поточний момент часу симуляції;

$k$  – коефіцієнт для переведення з кількості води в висоту рівня води, або тиску.

Другим важливим типом підсистем є підсистеми, що симулюють пристрої для перекачування води з однієї місткості в іншу (рис. 3.15).

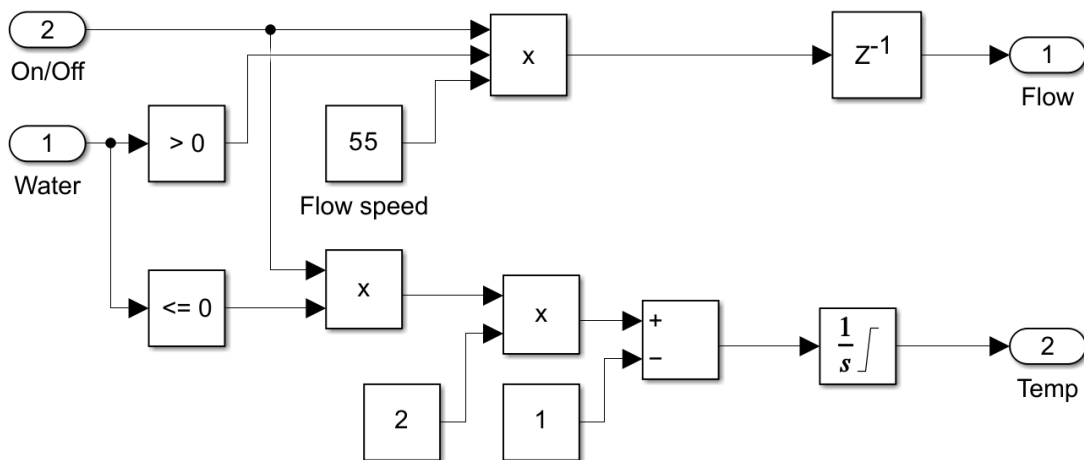


Рисунок 3.15 – Підсистема, що симулює елемент для перекачування води

Будова таких підсистем передбачає у випадку наявності води у її джерелі: криниці, місткості чи трубах, та увімкненості пристрою для перекачування, а також відправлення про кількість води, яку було переміщено, а у випадку відсутності води в джерелі відстежування поступового збільшення температури.

Зміна температури насосу визначається за формулою (3.4):

$$z(t) = \int_0^t 2x(\tau)y(\tau) - 1d\tau, \quad (3.4)$$

де  $z(t)$  – значення температури насосу у певний момент;

$x(\tau)$  – функція, що має значення 1, якщо насос ввімкнути та 0 якщо вимкнуто;

$y(\tau)$  – функція, що має значення 1, якщо насос не має води, яку він міг би перекачувати та 0 якщо він має цю воду;

$t$  – поточний момент часу симуляції.

Таким чином досягається можливість симулювати свердловини та насосні станції. Аналогічним чином реалізовані споживачі води, із зовнішніми входами для керування.

Керування споживанням води у симуляції показано на рисунку 3.16.

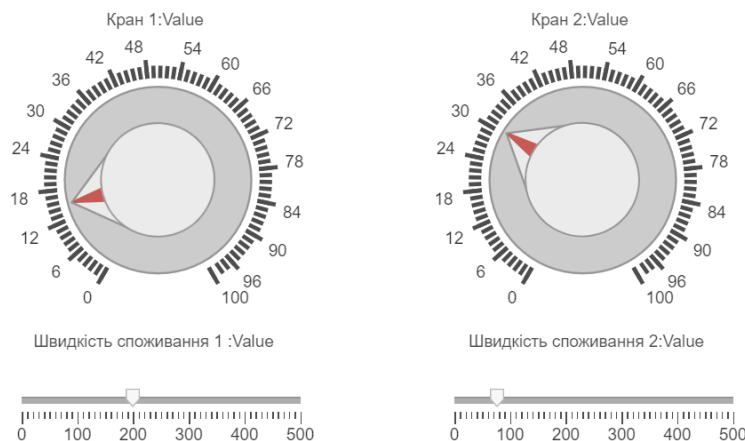


Рисунок 3.16 – Засоби для керування користувачами води

Своєю чергою інші підсистеми є простішими у своїх принципах роботи. Підсистеми для реле тиску пропускають сигнал про безпечність ввімкнення, якщо потрібний рівень тиску було виявлено, а підсистема для симуляції мікроконтролера вирішує безпечність ввімкнення свердловини та насосних станцій базуючись на релевантних налаштуваннях системи, виміряних температур та рівні води у

місткості. Відносна простота підсистеми, яка симулює мікроконтролер, досягається, відсутністю необхідності інтерпретації вимірів з відповідних сенсорів та відсутністю комунікації з хмарою в рамках симуляції призначеної для перевірки динамічних змін параметрів рівня води, тиску та температури, замість симуляції апаратного забезпечення. Під'єднання елементів та підсистем для загальної симуляції відбувається відповідно до їх призначення та розробленої системи в Wokwi (додаток В). Таким чином вдається виміряти та записати інформацію про рівень води в місткості, тиск в трубах, температуру насосних станцій та ввімкненість пристроїв перекачування води залежно від часу (рис. 3.17 - 3.18).

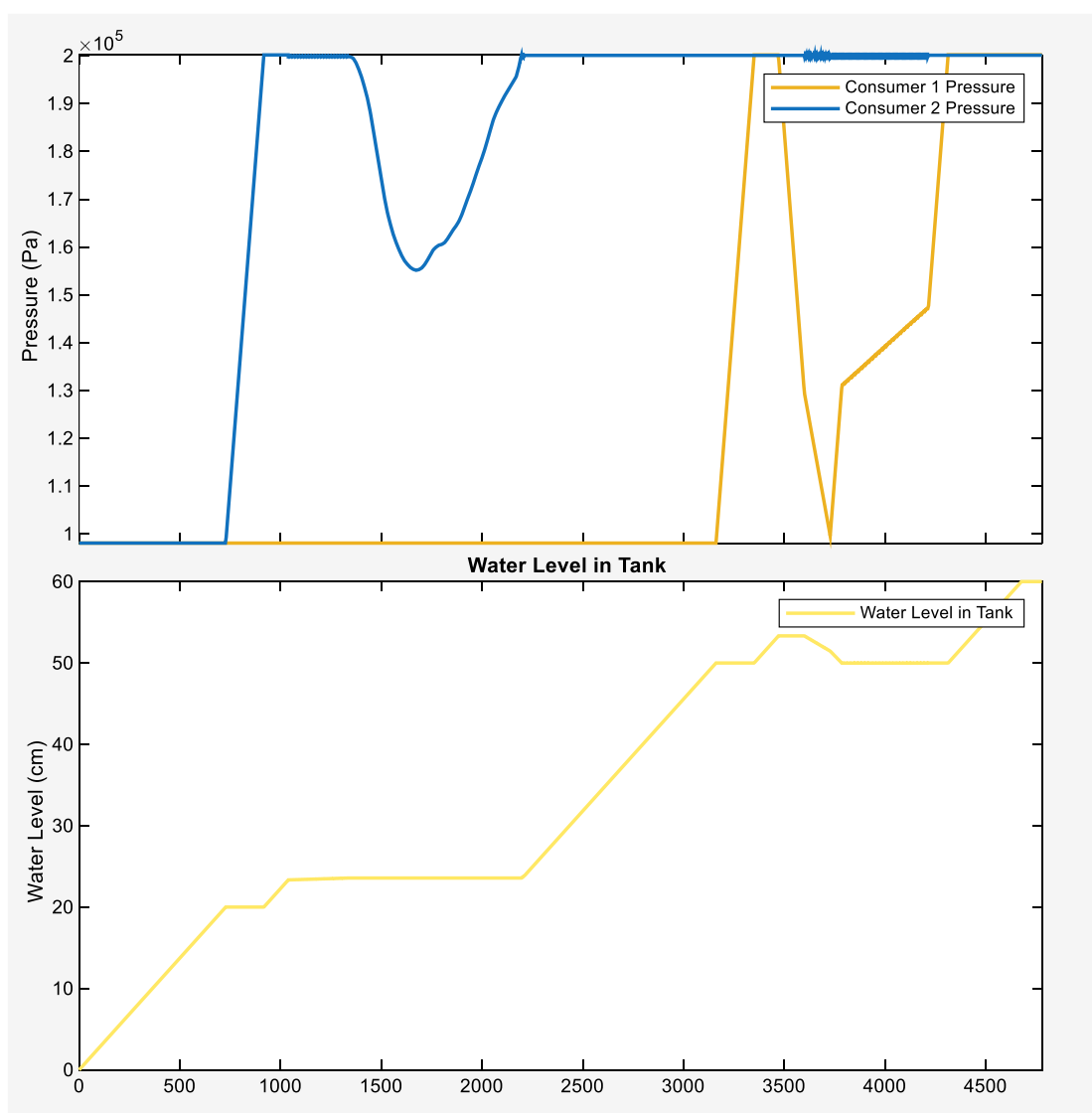


Рисунок 3.17 – Графік зміни тиску в трубах та рівня води в місткості для її зберігання

Зм.	Арк.	№ докум.	Підпис	Дата

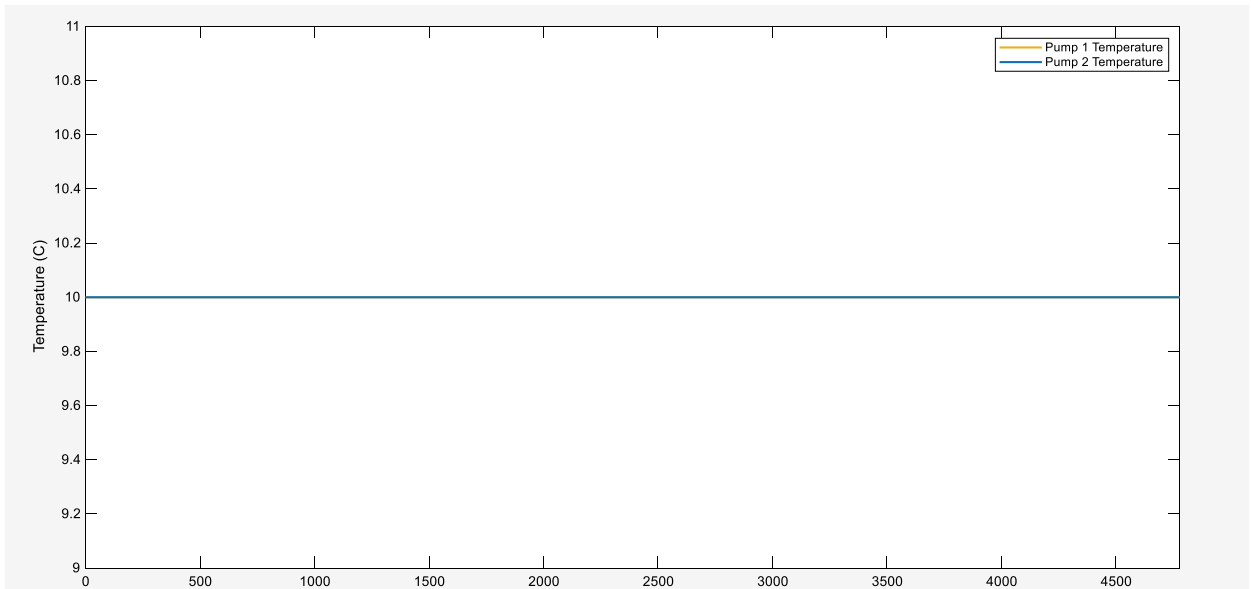


Рисунок 3.18 – Графік зміни температури насосних станцій

З приведених графіків можна спостерігати, що тиск тримається в безпечних межах, а ввімкнення пристроїв для перекачування води та рівень води система підтримує в відповідності до поставленого завдання. Своєю чергою температура у робочому режимі, оскільки насоси не працюють в холосту.

#### 3.4. Висновки до третього розділу

У межах розділу було реалізовано дві симуляції розробленої кіберфізичної системи водопостачання приватного домогосподарства. Симуляція, виконана за допомогою програмного забезпечення Wokwi, описує взаємодію апаратних елементів системи. Вона підтримує можливість для будь-яких вимірних параметрів з 9 сенсорів визначити реалістичні значення вимірюваних величин та які з сенсорів могли вийти з ладу. Також вона надає можливість взаємодії з хмарою для збереження показників та поновлення інформації про налаштування системи.

Друга симуляція була розроблена в MATLAB/Simulink та демонструє в реальному часі динамічну зміну рівня води в місткості для її збереження, тиск в трубах, температуру насосних станцій та керування їхньою роботою з часом в залежності один від одного.

Зм.	Арк.	№ докум.	Підпис	Дата

## ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було згідно з завданням розроблено кіберфізичну систему керування водопостачанням домашнього господарства.

В процесі роботи було проведено аналіз відомих систем водопостачання для приватних господарств було розглянуто прості системи водопостачання домашніх господарств, автоматизовані системи водопостачання та кіберфізичні системи керування водопостачанням домашніх господарств.

Під час огляду наявних рішень було виявлено відсутність або надзвичайно маленьку кількість систем водопостачання, які б використовували сучасні технологічні можливості для збільшення безпеки та складності алгоритмів експлуатації й одночасно знаходились в цій категорії, як необхідного обладнання, так і вартості його встановлення і розробки системи доступних для пересічного громадянина, в рамках сучасної економічної ситуації.

Оскільки ціни, енергоспоживання, розповсюдженість, ергономічність та надійність електронних засобів які можуть бути використані в проектах розумних будинків значно покращились на останні десятиліття, це вказує на можливу економічну нішу для порівняно економних кіберфізичних систем водопостачання приватних домогосподарств направлених на покращення експлуатації та надійності вже наявних систем. Таким чином, розроблення кіберфізичних систем водопостачання домашніх господарств є актуальним завданням сучасної комп'ютерної інженерії, вирішення якого надає можливість суттєво покращити характеристики поширених систем постачання води.

У другому розділі було проведено аналіз наявних та часто використовуваних елементів звичайних та кіберфізичних систем водопостачання, які необхідні для реалізації системи згідно з розробленою структурною схемою.

					КВРКІ.022032.22.02.27 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

Для мікроконтролера який буде виконувати функції керування всією системою було обрано ESP32-DevKitC V4 за хороший баланс ціни та технічних характеристик, низьке енергоспоживання та прийнятні розміри і наявність Wi-Fi модуля для зв'язку.

З метою виконання функції виміру температури насосної станції було обрано сенсор DS18B20 який дещо поступається технічними характеристиками розглянутим альтернативним рішенням, але має значно меншу ціну.

Для виміру рівня води елементів системи призначеного для її тимчасового зберігання та зменшення навантаження на насосну станцію при короткочасному різкому збільшенні потреби в воді було обрано використати комбінацію з двох сенсорів ZP4510 та ультразвукового сенсор відстані HC-SR04, що дозволить досягнути підвищену точність та надійність системи.

В якості реле тиску було обрано EASYSMALL -2M 1.5 BAR від компанії Pedrollo, що має можливість захисту від сухого ходу підвищуючи надійність системи та високу надійність з хорошими значеннями технічних характеристик.

В якості дисплея було обрано 20×4 2004 LCD модуль з I2C послідовним інтерфейсом з вбудованим контролером дисплеїв HD44780, що надає можливість виводити необхідну інформацію користувачу для покращення процесу експлуатації

У третьому розділі реалізовано дві симуляції розробленої кіберфізичної системи водопостачання приватного домогосподарства. Симуляція виконана за допомогою програмного забезпечення Wokwi симулює взаємодію апаратних елементів системи. Розроблене програмне забезпечення при перевірці у симульованому середовищі успішно відстежує значення вимірюваних сенсорами величин, визначає набір сенсорів яким можна довіряти базуючись на їхній суперечливості один одному та попереднім перевіркам. Також програмне забезпечення успішно взаємодіє зі збереженими у хмарі даними за допомогою MQTT протоколу, підтримуючи зв'язок з користувачем навіть при його відсутності у приміщенні з системою.

Також було розроблено симуляцію основних елементів для взаємодії з водою у системі у середовищі MATLAB/Simulink. Після розробки та під'єднання усіх

					КВРКІ.022032.22.02.27 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

свердловин, насосних станцій, систем труб, місткостей для зберігання води та реле тиску, було виявлено задовільне підтримання температури засобів перекачування води, рівня води та її тиску у системі протягом симульованого періоду часу.

Отже, усі поставлені завдання у роботі було вирішено. Надалі система має перспективи розвитку у напрямку розширення функціоналу передбачення потреб у воді базуючись на зібраній за період часу інформації.

Оскільки система зберігає усю інформацію необхідну для її функціонування та інформацію про стан системи протягом обраного часу, у майбутньому можна розробити додаток для смартфона, який буде надавати можливість зручного моніторингу інформації про стан системи, використання різних підходів для динамічного встановлення бажаного рівня води та попередження користувача про критичний стан системи без необхідності перевірки інформації в ручну.

					КВРКІ.022032.22.02.27 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Сільськогосподарське водопостачання: лекція 7. URL: <http://www.tsatu.edu.ua/tsst/wp-content/uploads/sites/6/lekciya-7-silskohospodarske-vodopostachannja.pdf> (дата звернення: 02.04.2025).
2. Автономне водопостачання приватного будинку. URL: <https://kolodec.com.ua/ua/avtonomnaya-kanalizaciya.html> (дата звернення: 03.04.2025).
3. Що потрібно купити, щоб зробити систему водопостачання. URL: <https://nasosvdom.com.ua/uk/-/korisna-informaciya/ssho-potribno-kupiti-sshob-zrobiti-sistemu-vodopostachannya> (дата звернення: 03.04.2025).
4. Сільське питне водопостачання: від ідеї до реалізації : методичний посібник. URL: [https://hromady.org/wp-content/uploads/2020/11/Despro\\_voda\\_print.pdf](https://hromady.org/wp-content/uploads/2020/11/Despro_voda_print.pdf) (дата звернення: 05.04.2025).
5. Insider's Guide to STM32 Microcontroller. URL: <https://www.hitex.com/fileadmin/assets/download/insiders-guides/stm32/isg-stm32-v18d-scr.pdf> (дата звернення: 16.04.2025).
6. STM32F103C8T6 board, alias Bluepill. URL: [https://os.mbed.com/users/hudakz/code/STM32F103C8T6\\_Hello/](https://os.mbed.com/users/hudakz/code/STM32F103C8T6_Hello/) (дата звернення: 16.04.2025)
7. Плата розробника STM32F103C8T6 ARM STM32. URL: <https://arduino.ua/prod1328-plata-razrabotchika-stm32f103c8t6-arm-stm32-minimalnaya-konfiguraciya> (дата звернення: 17.04.2025).
8. Uno R3 ATmega328P CH340G. URL: <https://www.roboter-bausatz.de/p/uno-r3-atmega328p-ch340g> (дата звернення: 17.04.2025).
9. Arduino UNO (A000066) Datasheet. URL: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf> (дата звернення: 17.04.2025).
10. Огляд ESP32. URL: <https://itmaster.biz.ua/electronics/esp32/esp32-s2.html> (дата звернення: 18.04.2025).

					КВРКІ.022032.22.02.27 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

11. WiFi Bluetooth модуль NodeMcu ESP32-WROOM-32S на Prom. URL: <https://ardu.prom.ua/ua/p1286879985-wifi-bluetooth-modul.html> (дата звернення: 27.04.2025).

12. AI-на краю: апаратні рішення. URL: <https://itmaster.biz.ua/electronics/ai/ai-edge.html> (дата звернення: 18.04.2025).

13. Matter стандарт для розумного будинку. URL: <https://itmaster.biz.ua/directory/standarts/matter-standart.html> (дата звернення: 18.04.2025).

14. ESP32-S3 Datasheet. URL: [https://www.espressif.com/sites/default/files/documentation/esp32-s3\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf) (дата звернення: 19.04.2025).

15. ESP32-C6 Datasheet. URL: [https://www.espressif.com/sites/default/files/documentation/esp32-c6\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf) (дата звернення: 19.04.2025).

16. Огляд мікроконтролера ESP32. URL: <https://itmaster.biz.ua/directory/microcontrollers/esp32.html> (дата звернення: 20.04.2025).

17. Підключення ESP32 до Arduino IDE. URL: <https://itmaster.biz.ua/electronics/esp32/esp32-arduino.html> (дата звернення: 20.04.2025).

18. ESP32-DevKitC V4 WROOM-32U на PROM. URL: <https://prom.ua/ua/p1440574523-plata-razrabotki-esp32.html> (дата звернення: 20.04.2025).

19. Продаж ESP32-S3 на OLX. URL: <https://www.olx.ua/uk/elektronika/q-esp32-s3/> (дата звернення: 21.04.2025).

20. Інструкція з експлуатації та технічного обслуговування. Насосна станція. Моделі: WP97801/WP97111/WP97131. URL: [https://kulibin.com.ua/upload/iblock/626/WP97801\\_111\\_131.pdf](https://kulibin.com.ua/upload/iblock/626/WP97801_111_131.pdf) (дата звернення: 21.04.2025).

21. Насосні станції DAB. URL: <https://nasosvdom.com.ua/uk/-/korisna-informaciya/nasosnye-stancii-dab-dab> (дата звернення: 22.04.2025).

22. Автоматична насосна станція WS 3519/WS 4024/WS4050/WS 5524P. URL: <https://content.rozetka.com.ua/goods/documents/original/175027270.pdf> (дата звернення: 22.04.2025).

					КВРКІ.022032.22.02.27 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

23. Сенсор температури LM35. URL: <https://prom.ua/ua/p81695570-datchik-temperaturi-lm35.html> (дата звернення: 24.04.2025).

24. Сенсор температури DS18B20. URL: <https://www.mini-tech.com.ua/ua/datchik-temperature-ds18b20> (дата звернення: 24.04.2025).

25. DS18B20 Digital Thermometer Datasheet. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf> (дата звернення: 23.04.2025).

26. LM35 Precision Centigrade Temperature Sensors. URL: <https://www.ti.com/lit/ds/symlink/lm35.pdf> (дата звернення: 23.04.2025).

27. DS18B20 Programmable Resolution 1-Wire Digital Thermometer Datasheet. URL: <https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf> (дата звернення: 23.04.2025).

28. Sensirion SHT3x-DIS Series Datasheet. URL: [https://sensirion.com/media/documents/213E6A3B/63A5A569/Datasheet\\_SHT3x\\_DIS.pdf](https://sensirion.com/media/documents/213E6A3B/63A5A569/Datasheet_SHT3x_DIS.pdf) (дата звернення: 24.04.2025).

29. Liquid Water Level Sensor Vertical Float Switches. URL: <https://www.igelectronics.com/products/ac487b0b21/1726810000000496077> (дата звернення: 25.04.2025).

30. Ultraschallsensor HC-SR04. URL: [https://wiki.hshl.de/wiki/index.php/Ultraschallsensor\\_HC-SR04](https://wiki.hshl.de/wiki/index.php/Ultraschallsensor_HC-SR04) (дата звернення: 25.04.2025).

31. Ultrasonic Ranging Module HC-SR04 Datasheet. URL: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> (дата звернення: 25.04.2025).

32. Реле тиску Wiki ТНТУ. URL: [https://wiki.tntu.edu.ua/Реле\\_тиску](https://wiki.tntu.edu.ua/Реле_тиску) (дата звернення: 26.04.2025).

33. Електрична схема підключення реле тиску. URL: [https://wiki.tntu.edu.ua/Файл:Електрична\\_схема\\_підключення\\_реле\\_тиску.jpg](https://wiki.tntu.edu.ua/Файл:Електрична_схема_підключення_реле_тиску.jpg) (дата звернення: 26.04.2025).

					КВРКІ.022032.22.02.27 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

34. Механічне або електронне реле тиску: що вибрати для насосної станції. URL: <https://termocom.com.ua/ua/mechanichne-abo-elektronne-rele-tisku-shho-vibrati-dlja-nasosnoi-stancii/> (дата звернення: 27.04.2025).

35. Електронне реле тиску EASYSMALL-2М 1.5 BAR Pedrollo. URL: <https://nasosplus.com.ua/ua/elektronnyy-regulyator-davleniya-easysmall-2m-1.5-bar-pedrollo/> (дата звернення: 27.04.2025).

36. STM32 Families. URL: [https://stm32world.com/wiki/STM32\\_Families](https://stm32world.com/wiki/STM32_Families) (дата звернення: 27.04.2025).

37. Stm32f103 pinout diagram. URL: [https://commons.wikimedia.org/wiki/File:Stm32f103\\_pinout\\_diagram.png](https://commons.wikimedia.org/wiki/File:Stm32f103_pinout_diagram.png) (дата звернення: 27.04.2025).

38. User Guide I2C Serial Interface 20x4 LCD Module. URL: [https://www.handsontec.com/dataspecs/I2C\\_2004\\_LCD.pdf](https://www.handsontec.com/dataspecs/I2C_2004_LCD.pdf) (дата звернення: 27.04.2025).

39. Комплексна система керування поливом та водопостачанням. URL: <https://aqteck.com.ua/ua/projects/kompleksna-systema-keruvannja-polyvom> (дата звернення: 06.04.2025).

40. ПЛК160-U02 Програмований логічний контролер. URL: <https://aqteck.com.ua/ua/programovani-logichni-kontrolery/plk160-m02-programovanyj-logichnyj-kontroler> (дата звернення: 06.04.2025).

41. Блоки живлення для пристроїв та промислової автоматики. URL: <https://aqteck.com.ua/ua/arhiv-produkciji/bloky-zhyvlennja-dlja-prystrojiv-ta-promyslovoji-avtomatyky> (дата звернення: 07.04.2025).

42. MB110-16Д модуль вводу дискретних сигналів. URL: <https://aqteck.com.ua/ua/moduli-vvodu-vyvodu/mv110-16d-modul-vvodu-dyskretnyh-sygnaliv> (дата звернення: 08.04.2025).

43. МУ110-8Р модуль дискретного виводу. URL: <https://aqteck.com.ua/ua/moduli-vvodu-vyvodu/mu110-8r-modul-dyskretnogo-vyvodu> (дата звернення: 08.04.2025).

					КВРКІ.022032.22.02.27 ПЗ	Арк. 72
Зм.	Арк.	№ докум.	Підпис	Дата		

44. БСФ блок мережевого фільтру. URL: <https://aqteck.com.ua/ua/arhiv-produkciji/bsf-blok-merzhevogo-filtru> (дата звернення: 10.04.2025).

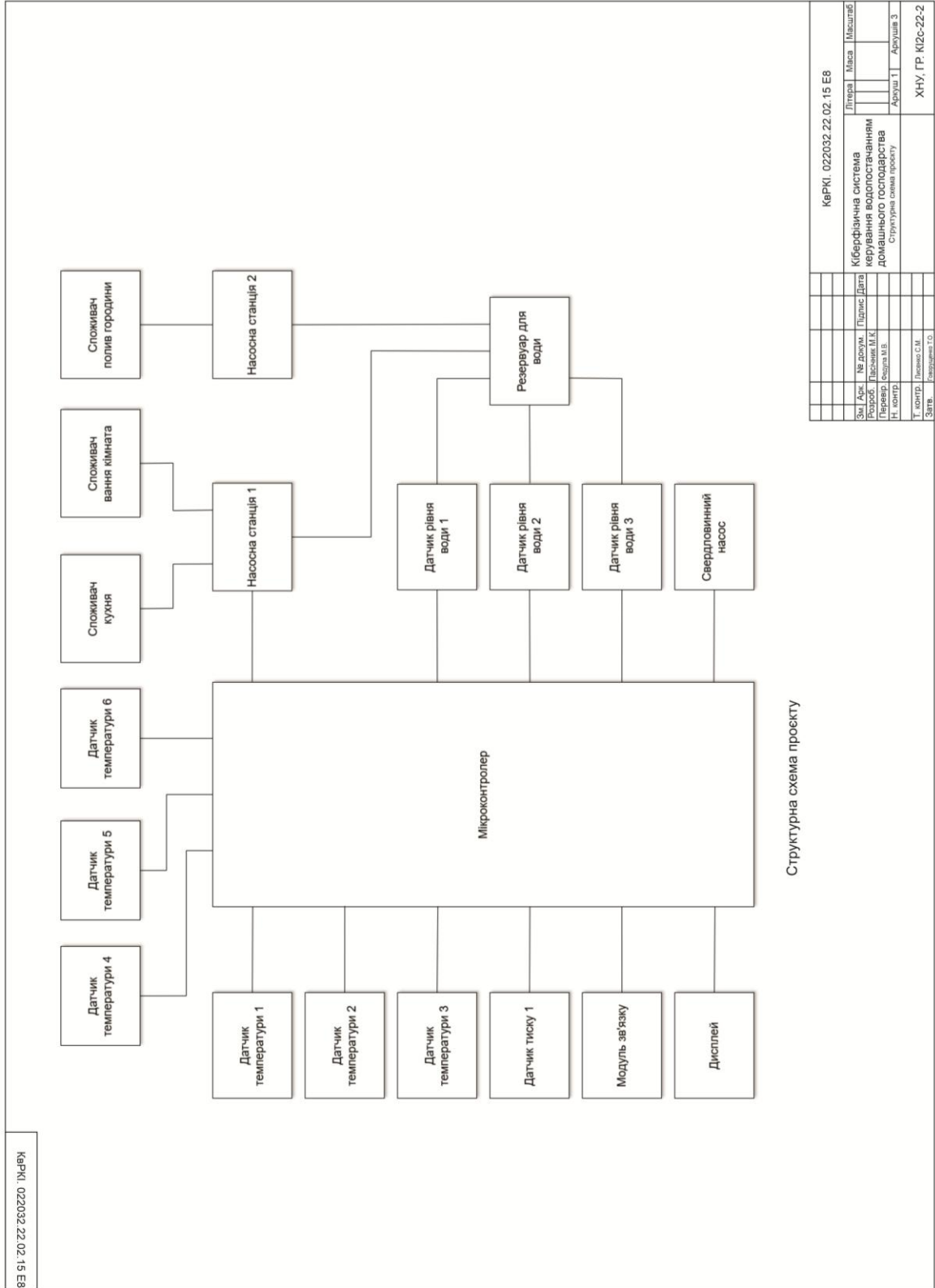
45. СП270 графічна панель оператора з сенсорним керуванням. URL: <https://aqteck.com.ua/ua/arhiv-produkciji/sp270-grafichna-panel-operatora-z-censornym-keruvannjam> (дата звернення: 11.04.2025).

46. Готельні системи опалення та водопостачання Sunray. URL: <https://sunray.ua/ukr/hotel> (дата звернення: 12.04.2025).

					КВРКІ.022032.22.02.27 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

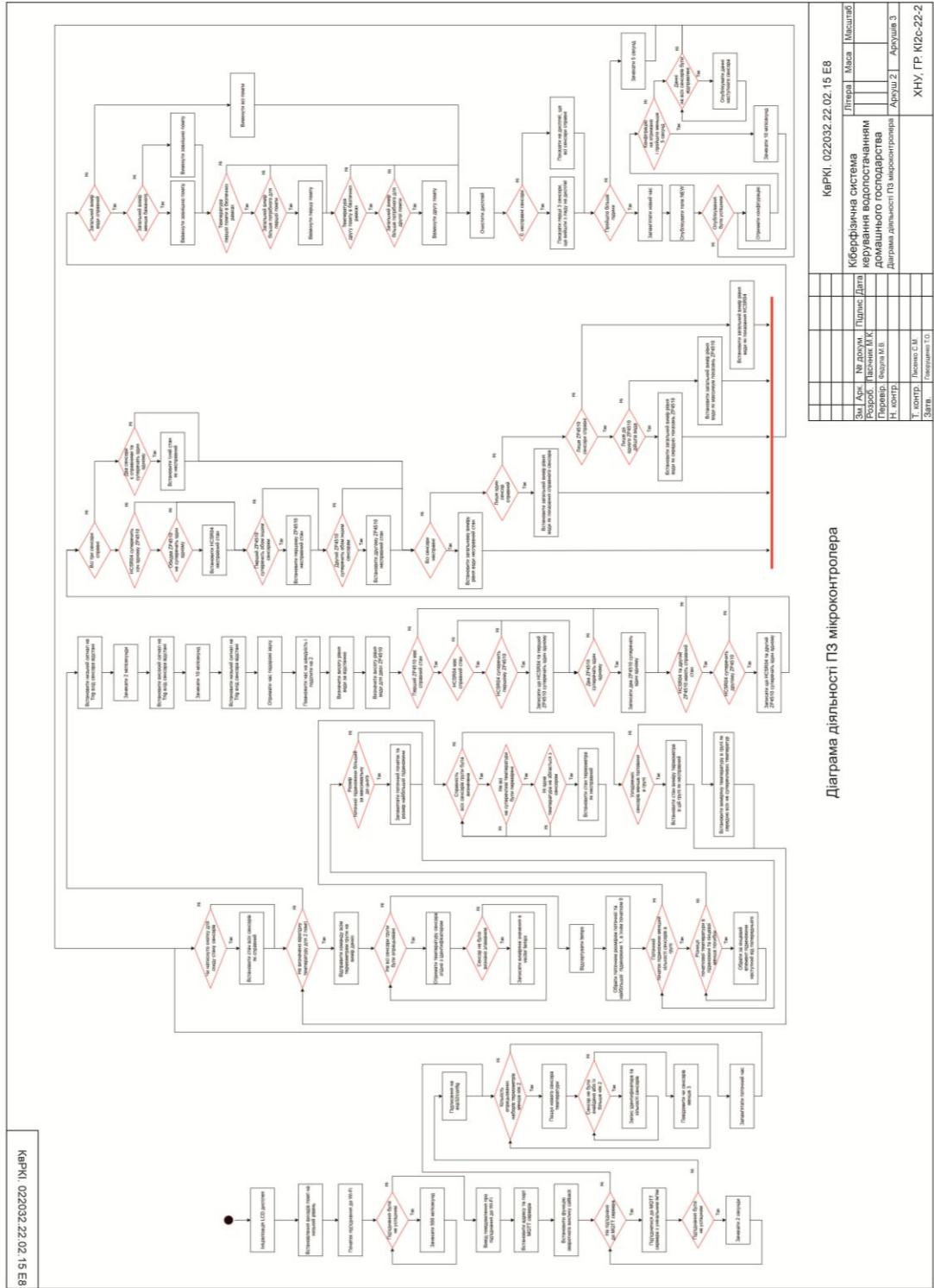
**Додаток А**  
**(обов'язковий)**

**КОПІЯ КРЕСЛЕННЯ «СТРУКТУРНА СХЕМА ПРОЄКТУ»**



## Додаток Б (обов'язковий)

# КОПІЯ КРЕСЛЕННЯ «ДІАГРАМА ДІЯЛЬНОСТІ ПЗ МІКРОКОНТРОЛЕРА»



КВРКІ\_022032.22.02.15.E8

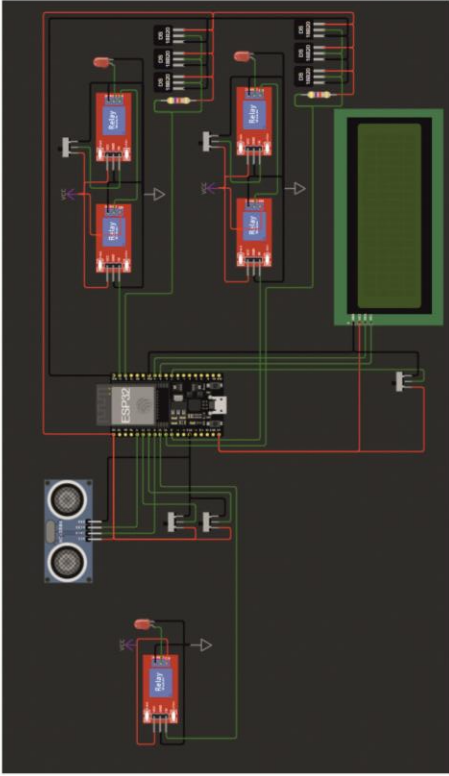
КВРКІ_022032.22.02.15.E8			
Літера	Місяц	Місця	
Кіберфізична система керування водопостачанням домашнього господарства			
Архив 1	Архив 2	Архив 3	
Діяльність діяльності ПЗ мікроконтролера			
ХНУ, ГР. КІЗС-22-2			

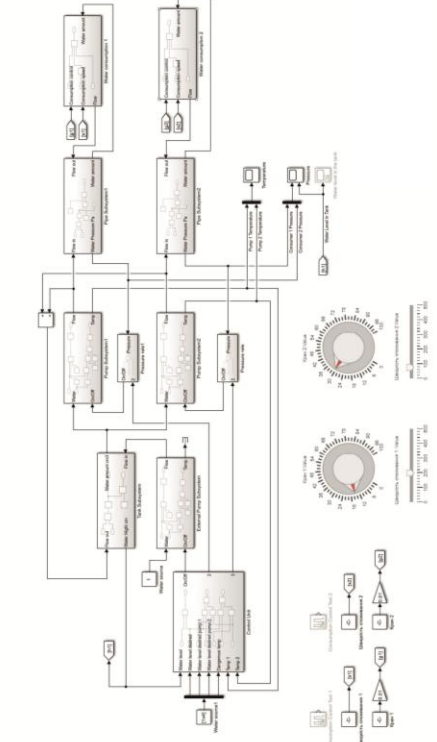
Діаграма діяльності ПЗ мікроконтролера

Від	На	Листів	Дата
Розробник	Начальник МК		
Н. констр.	Сергій А.В.		
Т. констр.	Лілія С.В.		
Затв.	Григорій О.О.		

## Додаток В (обов'язковий)

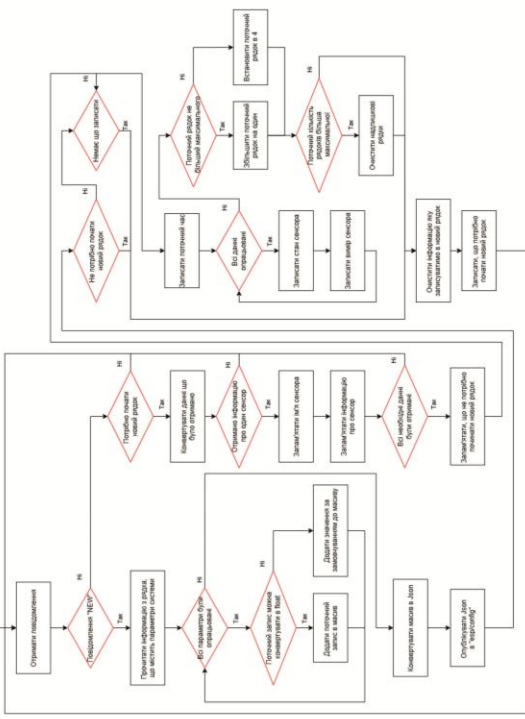
# КОПІЯ КРЕСЛЕННЯ «СИМУЛЯЦІЇ ТА ДІАГРАМА ДІЯЛЬНОСТІ ПРОЄКТУ»





Симуляція проєкту в ПЗ Wokwi

Симуляція проєкту в ПЗ Mathlab Simulink



Діаграма діяльності ПЗ зберігання інформації про стан системи

КерРКІ_0220332_22_02_15_E8		Літера	Масштаб
Скі. Арх.	МВ. Вектор	Підпис	Дата
Розроб.	Підпис А.М.		
Н. контр.	Вектор М.В.		
Т. комп.	Личко С.М.		
Затв.	Вектор М.В.		

КерРКІ_0220332_22_02_15_E8		Літера	Масштаб
Керівнича система керування водостачанням домашнього господарства			
Симуляції та діаграми діяльності проєкту			
			Архив 3
			Архив 3
			ХНУ, ГР. КІ2С-22-2

## Додаток Г (обов'язковий)

### КОД РОЗРОБЛЕНОГО ПЗ

```
main.cpp

#include <OneWire.h>
#include <DallasTemperature.h>
#include <unordered_map>
#include <string>
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <algorithm>
#include <vector>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define EXTERNAL_PUMP_PIN 25
#define INTERNAL_PUMP_PIN_1 23
#define INTERNAL_PUMP_PIN_2 27
#define TEMPERATURE_ONE_WIRE_BUS 22
#define TEMPERATURE_ONE_WIRE_BUS_2 26
#define ZP4510_1_PIN 32
#define ZP4510_2_PIN 35
#define TRIG_PIN 33
#define ECHO_PIN 34
#define ERROR_RESET_PIN 17
#define SDA_PIN 19
#define SCL_PIN 18

const char* ssid = "Wokwi-GUEST";
const char* password = "";
const char* mqtt_server = "broker.mqttdashboard.com";
const char* topic = "ESP32SensorMap";

WiFiClient espClient;
PubSubClient client(espClient);

float HCSR04_MOUNT_HEIGHT_CM = 160.0;
float ULTRASONIC_MAX_ERROR_CM = 1.0;
```

```

float ZP4510_1_LEVEL_CM = 20.0;
float ZP4510_2_LEVEL_CM = 100.0;
float TEMPERATURE_ERROR_MARGIN = 0.5;
float desiredWaterLevelCm = 50.0;
float internalPumpMaxTemperature = 50.0;
float waterLevelForInternalPump1 = 50;
float waterLevelForInternalPump2 = 20;

class Sensor{
    public:
    bool isSensorNotBroken = true;
    float measuredValue;
};

std::unordered_map<std::string, Sensor> sensorData = {
    {"DS18B20_1.0", Sensor()},
    {"DS18B20_1.1", Sensor()},
    {"DS18B20_1.2", Sensor()},
    {"Temperature1", Sensor()},
    {"DS18B20_2.0", Sensor()},
    {"DS18B20_2.1", Sensor()},
    {"DS18B20_2.2", Sensor()},
    {"Temperature2", Sensor()},
    {"ZP4510_1", Sensor()},
    {"ZP4510_2", Sensor()},
    {"HCSR04", Sensor()},
    {"WaterLevel", Sensor()}
};

class TempBusData{
    public:
    float waterLevelMin;
    uint8_t pin;
    OneWire oneWire;
    DallasTemperature sensors;
    DeviceAddress sensorAddresses[3];
    int temperatureSensorCount = 0;

    TempBusData(uint8_t busPin, int waterLevel): oneWire(busPin),
sensors(&oneWire) {
        pin = busPin;
        waterLevelMin = waterLevel;

```

```

    }
};

TempBusData temperatureData[] = {
    TempBusData(TEMPERATURE_ONE_WIRE_BUS,waterLevelForInternalPump1),
    TempBusData(TEMPERATURE_ONE_WIRE_BUS_2,waterLevelForInternalPump2)
};

LiquidCrystal_I2C lcd(0x27, 20, 4);
unsigned long startTime;

void printAddress(DeviceAddress deviceAddress) {
    for (uint8_t i = 0; i < 8; i++) {
        if (deviceAddress[i] < 16) Serial.print("0");
        Serial.print(deviceAddress[i], HEX);
    }
}

void sensorSearch(){

    for(int i = 0; i < 2; i++){
        while
(tempertureData[i].oneWire.search(tempertureData[i].sensorAddresses[tempertureD
ata[i].temperatureSensorCount]) && tempertureData[i].temperatureSensorCount < 3)
{
            Serial.print("Sensor ");
            Serial.print(tempertureData[i].temperatureSensorCount);
            Serial.print(" address: ");
            printAddress(tempertureData[i].sensorAddresses[tempertureData[i].tempe
ratureSensorCount]);
            Serial.println();
            tempertureData[i].temperatureSensorCount++;
        }

        if (tempertureData[i].temperatureSensorCount < 3) {
            Serial.println("Less than 3 DS18B20 sensors found at " +
tempertureData[i].pin);
        }
    }
}

void getAgreedTemperature(TempBusData temperatureSensors, int pumpNumber){

```

```

temperatureSensors.sensors.requestTemperatures();
std::vector<float> temps;

for (int i = 0; i < temperatureSensors.temperatureSensorCount; i++) {
    sensorData["DS18B20_" + std::to_string(pumpNumber) + "." +
std::to_string(i)].measuredValue =
temperatureSensors.sensors.getTempC(temperatureSensors.sensorAddresses[i]);

    if (sensorData["DS18B20_" + std::to_string(pumpNumber) + "." +
std::to_string(i)].isSensorNotBroken) {
        temps.push_back(sensorData["DS18B20_" + std::to_string(pumpNumber) +
"." + std::to_string(i)].measuredValue);
    }
}

std::sort(temps.begin(), temps.end());

int k = 1;
int maxLen = 0;
int startIndex = 0;

for (int j = 0; j < temps.size(); j++) {
    while (fabs(temps[j] - temps[k]) <= TEMPERATURE_ERROR_MARGIN) {
        k++;
    }

    int currentLen = k - j;
    if (currentLen > maxLen) {
        maxLen = currentLen;
        startIndex = j;
    }
}

for (int i = 0; i < temperatureSensors.temperatureSensorCount; i++) {
    bool check = false;
    for(int j = 0; j < maxLen; j++)
        if(sensorData["DS18B20_" + std::to_string(pumpNumber) + "." +
std::to_string(i)].measuredValue == temps[startIndex + j])
            check = true;

    if(!check)

```

```

        sensorData["DS18B20_" + std::to_string(pumpNumber) + "." +
std::to_string(i)].isSensorNotBroken = false;
    }

    if(maxLen < (temperatureSensors.temperatureSensorCount + 1) / 2){
        sensorData["Temperature" + std::to_string(pumpNumber)].isSensorNotBroken
= false;
        return;
    }

    float result = 0;
    for(int j = 0; j < maxLen; j++)
        result += temps[startIndex + j];
    result /= maxLen;

    sensorData["Temperature" + std::to_string(pumpNumber)].isSensorNotBroken =
true;
    sensorData["Temperature" + std::to_string(pumpNumber)].measuredValue =
result;
}

float readDistanceCM() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    int duration = pulseIn(ECHO_PIN, HIGH);
    return duration * 0.034 / 2;
}

float getWaterLevelFromUltrasonic() {
    return HCSR04_MOUNT_HEIGHT_CM - readDistanceCM();
}

void estimateWaterLevel() {
    float levelFromUltrasonic = getWaterLevelFromUltrasonic();
    sensorData["HCSR04"].measuredValue = levelFromUltrasonic;
    sensorData["ZP4510_1"].measuredValue = digitalRead(ZP4510_1_PIN) *
ZP4510_1_LEVEL_CM;
    sensorData["ZP4510_2"].measuredValue = digitalRead(ZP4510_2_PIN) *
ZP4510_2_LEVEL_CM;
}

```

```

sensorData["WaterLevel"].isSensorNotBroken = true;

bool ZP1vsZP2 = false, ZP1vsHCSR = false, ZP2vsHCSR = false;

if(sensorData["ZP4510_1"].isSensorNotBroken){
    if(sensorData["HCSR04"].isSensorNotBroken && (digitalRead(ZP4510_1_PIN)
?
        sensorData["HCSR04"].measuredValue - ZP4510_1_LEVEL_CM <
ULTRASONIC_MAX_ERROR_CM :
        sensorData["HCSR04"].measuredValue - ZP4510_1_LEVEL_CM >
ULTRASONIC_MAX_ERROR_CM))
        ZP1vsHCSR = true;
        if(sensorData["ZP4510_2"].isSensorNotBroken &&
(digitalRead(ZP4510_1_PIN) != digitalRead(ZP4510_2_PIN) &&
(ZP4510_1_LEVEL_CM > ZP4510_2_LEVEL_CM ? digitalRead(ZP4510_1_PIN) :
digitalRead(ZP4510_2_PIN))))
            ZP1vsZP2 = true;
    }

    if(sensorData["HCSR04"].isSensorNotBroken &&
sensorData["ZP4510_2"].isSensorNotBroken &&
(digitalRead(ZP4510_2_PIN) ? sensorData["HCSR04"].measuredValue -
ZP4510_2_LEVEL_CM < ULTRASONIC_MAX_ERROR_CM :
    sensorData["HCSR04"].measuredValue - ZP4510_2_LEVEL_CM >
ULTRASONIC_MAX_ERROR_CM))
        ZP2vsHCSR = true;

    if(sensorData["ZP4510_1"].isSensorNotBroken &&
sensorData["ZP4510_2"].isSensorNotBroken &&
sensorData["HCSR04"].isSensorNotBroken){
        if(!ZP1vsZP2 && (ZP1vsHCSR || ZP2vsHCSR))
            sensorData["HCSR04"].isSensorNotBroken = false;
        if(ZP1vsHCSR && ZP1vsZP2)
            sensorData["ZP4510_1"].isSensorNotBroken = false;
        if(ZP1vsZP2 && ZP2vsHCSR)
            sensorData["ZP4510_2"].isSensorNotBroken = false;
    }
    else if(sensorData["ZP4510_1"].isSensorNotBroken &&
sensorData["ZP4510_2"].isSensorNotBroken && ZP1vsZP2) {
        sensorData["ZP4510_1"].isSensorNotBroken = false;

```

```

        sensorData["ZP4510_2"].isSensorNotBroken = false;
    }
    else if(sensorData["ZP4510_1"].isSensorNotBroken &&
sensorData["HCSR04"].isSensorNotBroken && ZP1vsHCSR){
        sensorData["ZP4510_1"].isSensorNotBroken = false;
        sensorData["HCSR04"].isSensorNotBroken = false;
    }
    else if(sensorData["ZP4510_2"].isSensorNotBroken &&
sensorData["HCSR04"].isSensorNotBroken && ZP2vsHCSR){
        sensorData["HCSR04"].isSensorNotBroken = false;
        sensorData["ZP4510_2"].isSensorNotBroken = false;
    }

    if(!sensorData["ZP4510_1"].isSensorNotBroken &&
!sensorData["ZP4510_2"].isSensorNotBroken &&
!sensorData["HCSR04"].isSensorNotBroken){
        sensorData["WaterLevel"].isSensorNotBroken = false;
    }
    else if(sensorData["ZP4510_1"].isSensorNotBroken &&
!sensorData["ZP4510_2"].isSensorNotBroken &&
!sensorData["HCSR04"].isSensorNotBroken){
        sensorData["WaterLevel"].isSensorNotBroken = true;
        sensorData["WaterLevel"].measuredValue =
sensorData["ZP4510_1"].measuredValue;
    }
    else if(!sensorData["ZP4510_1"].isSensorNotBroken &&
sensorData["ZP4510_2"].isSensorNotBroken &&
!sensorData["HCSR04"].isSensorNotBroken){
        sensorData["WaterLevel"].isSensorNotBroken = true;
        sensorData["WaterLevel"].measuredValue =
sensorData["ZP4510_2"].measuredValue;
    }
    else if(sensorData["ZP4510_1"].isSensorNotBroken &&
sensorData["ZP4510_2"].isSensorNotBroken &&
!sensorData["HCSR04"].isSensorNotBroken){
        sensorData["WaterLevel"].isSensorNotBroken = true;
        if(digitalRead(ZP4510_1_PIN) != digitalRead(ZP4510_2_PIN))
            sensorData["WaterLevel"].measuredValue =
(sensorData["ZP4510_1"].measuredValue + sensorData["ZP4510_2"].measuredValue) / 2
;
        else

```

```

        sensorData["WaterLevel"].measuredValue =
max(sensorData["ZP4510_1"].measuredValue, sensorData["ZP4510_2"].measuredValue) *
digitalRead(ZP4510_1_PIN);
    }
    else {
        sensorData["WaterLevel"].isSensorNotBroken = true;
        sensorData["WaterLevel"].measuredValue =
sensorData["HCSR04"].measuredValue ;
    }
}

void resetAllSensorsStatus() {
    for (auto& pair : sensorData) {
        pair.second.isSensorNotBroken = true;
    }
}

void printSensorData() {
    for (const auto& pair : sensorData) {
        Serial.print("Sensor Name: ");
        Serial.print(pair.first.c_str());
        Serial.print(" | isSensorNotBroken: ");
        Serial.print(pair.second.isSensorNotBroken);
        Serial.print(" | Measured Value: ");
        Serial.println(pair.second.measuredValue);
    }
    Serial.println();
}

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

void debugDNS() {
    IPAddress mqttIP;
    if (WiFi.hostByName("broker.mqttdashboard.com", mqttIP)) {

```

```

        Serial.print("Resolved MQTT broker IP: ");
        Serial.println(mqttIP);
    } else {
        Serial.println("DNS resolution failed.");
    }
}

void callback(char* topic, byte* payload, unsigned int length) {
    StaticJsonDocument<512> doc;
    DeserializationError error = deserializeJson(doc, payload, length);

    if (error) {
        Serial.print("JSON parse failed: ");
        Serial.println(error.f_str());
        return;
    }

    if (!doc.is<JsonArray>()) {
        Serial.println("Expected JSON array.");
        return;
    }

    JsonArray arr = doc.as<JsonArray>();

    if (arr.size() >= 9) {
        HCSR04_MOUNT_HEIGHT_CM      = arr[0];
        ULTRASONIC_MAX_ERROR_CM     = arr[1];
        ZP4510_1_LEVEL_CM           = arr[2];
        ZP4510_2_LEVEL_CM           = arr[3];
        TEMPERATURE_ERROR_MARGIN    = arr[4];
        desiredWaterLevelCm         = arr[5];
        internalPumpMaxTemperature   = arr[6];
        waterLevelForInternalPump1   = arr[7];
        waterLevelForInternalPump2   = arr[8];

        Serial.println("Config values updated.");
    } else {
        Serial.println("Received config array too short.");
    }
}

void connectToMQTT() {

```

```

client.setCallback(callback);
while (!client.connected()) {
    Serial.print("Connecting to MQTT...");
    if (client.connect(("ESP32Publisher-" + WiFi.macAddress()).c_str())) {
        Serial.println(" connected.");
        client.subscribe("esp32/config");
    } else {
        Serial.print(" failed, rc=");
        Serial.print(client.state());
        Serial.println(" retrying in 2s");
        delay(2000);
    }
}
}

void publishSensorData() {
    bool startResult = client.publish(topic, "NEW");

    Serial.print("Published NEW message to start row. Result: ");
    Serial.println(startResult ? "Success" : "Failed");

    if(startResult){
        while (millis() - startTime < 5000 && client.loop()) {
            delay(10);
        }
        for (const auto& pair : sensorData) {
            StaticJsonDocument<128> doc;

            const std::string& key = pair.first;
            const Sensor& sensor = pair.second;

            JsonObject obj = doc.createNestedObject(String(key.c_str()));
            obj["isSensorNotBroken"] = sensor.isSensorNotBroken;
            obj["measuredValue"] = sensor.measuredValue;

            char buffer[128];
            size_t len = serializeJson(doc, buffer);

            bool result = client.publish(topic, buffer, len);
            Serial.print("Published sensor: ");
            Serial.print(key.c_str());
            Serial.print(" Result: ");

```

```

        Serial.println(result ? "Success" : "Failed");
        Serial.println(buffer);

        delay(100);
    }
    Serial.println();
}
else {
    delay(5000);
}
}

void displayBrokenSensors() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Broken Sensors:");

    int row = 1;
    for (const auto& pair : sensorData) {
        if (!pair.second.isSensorNotBroken && pair.first != "Temperature1" &&
pair.first != "Temperature2" && pair.first != "WaterLevel") {
            lcd.setCursor(0, row);
            lcd.print(pair.first.c_str());

            row++;
            if (row >= 4) break; // Only 4 rows on display
        }
    }

    if (row == 1) {
        lcd.setCursor(0, 1);
        lcd.print("Everything works");
    }
}

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

    Serial.println("System Booting...");
    Wire.begin(SDA_PIN, SCL_PIN);
    lcd.init();
    lcd.backlight();
}

```

```

lcd.setCursor(0, 0);
lcd.print("System Booting...");

pinMode(EXTERNAL_PUMP_PIN, OUTPUT);
pinMode(INTERNAL_PUMP_PIN_1, OUTPUT);
pinMode(INTERNAL_PUMP_PIN_2, OUTPUT);
pinMode(ZP4510_1_PIN, INPUT);
pinMode(ZP4510_2_PIN, INPUT);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
pinMode(ERROR_RESET_PIN, INPUT);

digitalWrite(EXTERNAL_PUMP_PIN, LOW);
digitalWrite(INTERNAL_PUMP_PIN_1, LOW);
digitalWrite(INTERNAL_PUMP_PIN_2, LOW);

connectToWiFi();
debugDNS();
client.setServer(mqtt_server, 1883);
connectToMQTT();

sensorSearch();

}

void loop() {
    if(digitalRead(ERROR_RESET_PIN))
        resetAllSensorsStatus();

    getAgreedTemperature(temperatureData[0], 1);
    getAgreedTemperature(temperatureData[1], 2);
    estimateWaterLevel();

    digitalWrite(EXTERNAL_PUMP_PIN,
sensorData["WaterLevel"].isSensorNotBroken &&
        sensorData["WaterLevel"].measuredValue < desiredWaterLevelCm);
    digitalWrite(INTERNAL_PUMP_PIN_1, sensorData["WaterLevel"].isSensorNotBro
ken && sensorData["WaterLevel"].measuredValue >= temperatureData[0].waterLevelMin
        && sensorData["Temperature1"].isSensorNotBroken &&
sensorData["Temperature1"].measuredValue < internalPumpMaxTemperature);
    digitalWrite(INTERNAL_PUMP_PIN_2, sensorData["WaterLevel"].isSensorNotBro
ken && sensorData["WaterLevel"].measuredValue >= temperatureData[1].waterLevelMin

```

```

        && sensorData["Temperature2"].isSensorNotBroken &&
sensorData["Temperature2"].measuredValue < internalPumpMaxTemperature);

    printSensorData();
    displayBrokenSensors();

    if (millis() - startTime >= 15000){
        startTime = millis();
        if (!client.connected()) {
            connectToMQTT();
        }
        publishSensorData();
    }
    else delay(5000);
}

```

## Subscriber.py

```

import json
import os
from datetime import datetime, UTC
import paho.mqtt.client as mqtt
import gspread
from oauth2client.service_account import ServiceAccountCredentials

# Google Sheets setup
scope = [
    "https://spreadsheets.google.com/feeds",
    "https://www.googleapis.com/auth/drive"
]
CREDENTIALAL_PATH = os.path.join(os.path.dirname(__file__), "credential.json")
creds = ServiceAccountCredentials.from_json_keyfile_name(CREDENTIALAL_PATH,
scope)
gs_client = gspread.authorize(creds)
sheet = gs_client.open("ESP32WaterSuplyData").sheet1 # Must match exact
name

# MQTT setup
MQTT_BROKER = "broker.mqttdashboard.com"
MQTT_TOPIC = "ESP32SensorMap"

# Define the desired sensor order here:

```

```

expected_sensors = [
    "WaterLevel",
    "ZP4510_1",
    "ZP4510_2",
    "HCSR04",
    "Temperature1",
    "DS18B20_1.0",
    "DS18B20_1.1",
    "DS18B20_1.2",
    "Temperature2",
    "DS18B20_2.0",
    "DS18B20_2.1",
    "DS18B20_2.2"
]

current_row_data = {} # Temporary store for sensor data of the current row
row_started = False

def get_row_limit():
    return int(sheet.acell("A2").value)

def get_current_index():
    return int(sheet.acell("B2").value)

def set_current_index(index):
    sheet.update_acell("B2", str(index))

def trim_excess_rows(limit):
    existing_data = sheet.get_all_values()
    if len(existing_data) > limit:
        sheet.batch_clear([f"A{limit+1}:Z{len(existing_data)}"])

def read_config_values_ordered():
    values = sheet.range("C2:S2")
    config_floats = []

    for i, val in enumerate(values):
        if i%2 == 0:
            try:
                config_floats.append(float(val.value))
            except ValueError:
                print(f"Warning: Invalid float at row {i+1}, skipping.")

```

```

        config_floats.append(0.0)

    return config_floats

def publish_config_values(mqtt_client):
    payload = json.dumps(read_config_values_ordered())
    mqtt_client.publish("esp32/config", payload)
    print("Published config array to ESP32:", payload)

def write_row_to_sheet():
    if not current_row_data:
        return # Nothing to write
    timestamp = datetime.now(UTC).strftime("%Y-%m-%d %H:%M:%S")

    row = [timestamp]
    for sensor in expected_sensors:
        val = current_row_data.get(sensor, {})
        status = "TRUE" if val.get("isSensorNotBroken", False) else "FALSE"
        measured = val.get("measuredValue", "N/A")
        row.append(status)
        row.append(str(measured))

    # Determine which row to write (starting from row 2 due to headers)
    target_row = get_current_index()
    num_columns = len(row)
    cell_range = f"A{target_row}:{chr(64 + num_columns)}{target_row}"

    # Write data
    sheet.update(range_name=cell_range, values=[row]) # Nested list = one
row
    print(f"Wrote row to position {target_row}: {row}")

    # Update index for next write
    set_current_index(get_current_index() + 1 if get_current_index() <
get_row_limit() else 4)
    trim_excess_rows(get_row_limit())

def on_connect(client, userdata, flags, rc):
    print(f"Connected to MQTT broker with code {rc}")
    client.subscribe(MQTT_TOPIC)

def on_message(client, userdata, msg):

```

```

global current_row_data, row_started
payload = msg.payload.decode().strip()
print(f"\nMessage received:\n{payload}")

# Check for NEW message to start a new row
if payload == "NEW":
    publish_config_values(mqtt_client)
    if row_started:
        # Save previous row before starting a new one
        write_row_to_sheet()
    current_row_data = {}
    row_started = True
    print("Started a new row.")
    return

if not row_started:
    print("Ignoring sensor data before first NEW message.")
    return

# Parse sensor data message
try:
    data = json.loads(payload)
    # Expecting one sensor per message
    if len(data) != 1:
        print("Expected one sensor per message.")
        return

    sensor_name = list(data.keys())[0]
    sensor_info = data[sensor_name]

    current_row_data[sensor_name] = sensor_info
    print(f"Stored data for sensor: {sensor_name}")

    # Check if all expected sensors have been received
    if all(sensor in current_row_data for sensor in expected_sensors):
        write_row_to_sheet()
        current_row_data = {}
        row_started = False # Wait for next NEW message to start

except Exception as e:
    print(f"Error processing sensor message: {e}")

```

```
mqtt_client = mqtt.Client()
mqtt_client.on_connect = on_connect
mqtt_client.on_message = on_message

print("Connecting to broker...")
mqtt_client.connect(MQTT_BROKER, 1883, 60)
mqtt_client.loop_forever()
```

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Максим ПАСІЧНИК

Співавтор:

Назва: Пасічник\_Кіберфізична система керування водопостачанням домашнього господарства

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:5.6%

Коефіцієнт подібності 2:1.7%

Мікропробіли: 6

Заміна букв: 3

Інтервали: 0

Білі знаки: 11

Дата створення звіту: 2025-06-12 07:12:43.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-12

Дата



Доцент Андрій Нічепорук

експерт

Thu Jun 12 08:05:56 EEST 2025, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

# Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 16.0%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 12%

ID: 245238 Title: БКР Кіберфізична система керування водопостачанням домашнього господарства Added in a DB: 2025-06-12 Authors: Маским ПАСІЧНИК Heads: Микола ФЕДУЛА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	81838	572	13849 (17%)	120 (21%)

## Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes
240745	Title: Звіт з ПДП Кіберфізична система керування водопостачанням домашнього господарства Added in a DB: 2025-05-02 Authors: Пасічник М. Heads: Федула М.В. Consultants: Opponents:	13409 (16.0%)	119 (21.0%)

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Пасічник Максим Костянтинович

Тема: Кіберфізична система керування водопостачанням домашнього господарства

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень  3  Кількість сторінок записки  76

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення кіберфізичної система керування водопостачанням домашнього господарства на основі аналізу та узгодження режимів роботи насосів і витрат води.
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено аналіз відомих систем водопостачання для приватних господарств, було розглянуто прості системи водопостачання домашніх господарств, автоматизовані системи водопостачання та кіберфізичні системи керування водопостачанням домашніх господарств. Під час огляду наявних рішень було виявлено відсутність або надзвичайно маленьку кількість систем водопостачання, які б використовували сучасні технологічні можливості для збільшення безпеки та складності алгоритмів експлуатації й одночасно мали доступну для пересічного громадянина ціну. В другому розділі кваліфікаційної роботи проведено аналіз наявних та часто використовуваних елементів звичайних та кіберфізичних систем водопостачання, які необхідні для реалізації системи згідно з розробленою структурною схемою. Для виконання поставлених завдань було обрано мікроконтролер ESP32-DevKitC V4, сенсори температури DS18B20, ультразвуковий сенсор відстані HC-SR04, сенсори рівня води ZP4510, реле тиску EASYSMALL -2M 1.5 BAR та дисплей модуля LCD 20×4 2004 з I2C. В третьому розділі кваліфікаційної

роботи було реалізовано дві симуляції розробленої кіберфізичної системи водопостачання приватного домогосподарства, у програмному забезпеченні Wokwi та MATLAB/Simulink. Симуляція, виконана за допомогою програмного забезпечення Wokwi, описує взаємодію апаратних елементів системи, а симуляція в MATLAB/Simulink демонструє в реальному часі динамічну зміну стану системи.

4. Позитивні сторони роботи: висока надійність розробленої системи.

5. Негативні сторони роботи: симуляція, виконана за допомогою програмного забезпечення MATLAB/Simulink, припускає лінійне відношення між кількістю води та тиском в трубах.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

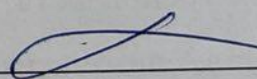
8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: відмінно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Бедриш Д. П. Зав. сек. ІІІ ХЦУ

"13" 06 2025 р.

 (підпис)

Завідувачу кафедри КПС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Максима ПАСІЧНИКА

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-22-2

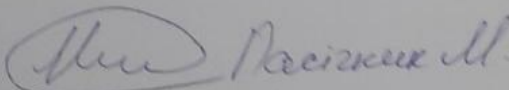
### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

11.06. 2025 року

 Максим Пасічник М.

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Кіберфізична система керування водопостачанням домашнього господарства

Автор: Максим ПАСІЧНИК

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Микола ФЕДУЛА, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:



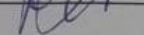
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами, виразами, або власними назвами про що свідчить посилання системи на збіг на однотипні фрагменти речення;
- 4) якість запозичень відповідає технічним особливостям дослідження: виявлено збіги в формулах та термінах, які не можуть вважатися авторським порушенням;
- 5) система зафіксувала технічні модифікації тексту, зокрема заміну окремих символів. Це є наслідком форматування або експорту документа і не є цілеспрямованим уникненням перевірки.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 5.65% і адресується до 25 першоджерела; та системою Anti-Plagiarism складає 16%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

  
\_\_\_\_\_  
  
\_\_\_\_\_  
  
\_\_\_\_\_

Микола ФЕДУЛА

Андрій Нічепорук

Ольга ПАВЛОВА