

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система вимірювання рівня рідини та об'єму в
циліндрі для ТОВ «Пром-Дорекс»

Виконав(ла): студент(ка) IV курсу, групи СІ-42

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Стецько В.Р.

(прізвище та ініціали)

Керівник

(підпис)

Стадник Н.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тиш Є.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Стоянов Ю. М.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Стецько Владислав Русланович
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс»

Керівник роботи к.т.н., Стадник Наталія Богданівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » 04 2024 року № 4/7-408

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпеки життєдіяльності, основи охорони праці

Висновок

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Структурна схема комп'ютеризованої системи

Схема електрична принципова

Блок-схема алгоритму роботи мікроконтролера

Блок-схема алгоритму роботи веб-інтерфейсу

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., д.т.н., професор кафедри МТ</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Затвердження технічного завдання</i>	<i>01.02-09.02</i>	<i>Виконано</i>
2	<i>Аналіз технічного завдання</i>	<i>05.02-11.02</i>	<i>Виконано</i>
3	<i>Розробка структурної схеми</i>	<i>25.04-06.05</i>	<i>Виконано</i>
4	<i>Вибір елементної бази</i>	<i>07.05-13.05 4</i>	<i>Виконано</i>
5	<i>Вибір програмного забезпечення</i>	<i>14.05-16.05 4</i>	<i>Виконано</i>
6	<i>Реалізація проектної складової</i>	<i>19.05-08.06</i>	<i>Виконано</i>
7	<i>Опрацювання розділу «Безпека життєдіяльності, основи охорони праці»</i>	<i>10.06-15.06</i>	<i>Виконано</i>
8	<i>Оформлення пояснювальної записки кваліфікаційної роботи</i>	<i>16.06-20.06</i>	<i>Виконано</i>
9	<i>Оформлення графічної частини</i>	<i>18.06-19.06</i>	<i>Виконано</i>
10	<i>Підготовка до захисту</i>	<i>24.06-28.06</i>	<i>Виконано</i>

Студент _____
(підпис)

Стецько В.Р. _____
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Стадник Н.Б. _____
(прізвище та ініціали)

АНОТАЦІЇ

Комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс». // Кваліфікаційна робота бакалавра // Стецько Владислав Русланович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІ-42 // Тернопіль, 2024 // с. – 56, рис. – 46 , табл. – 1 , кресл. – 4, додат. – 4 , бібліогр. – 24 .

Ключові слова: комп'ютеризована система вимірювання рівня рідини та об'єму для ТОВ «Пром-Дорекс», ESP32, мікроконтролер, РК-дисплей, ультразвуковий датчик відстані.

В кваліфікаційній роботі бакалавра розроблено комп'ютеризовану систему вимірювання рівня рідини та об'єму для ТОВ «Пром-Дорекс» на базі мікроконтролера ESP32.

Кваліфікаційна робота складається з чотирьох розділів.

В першому розділі проводиться аналіз технічного завдання та перегляд можливих рішень завдання.

Другий розділ присвячений розробці системи та інструментів для розробки.

В третьому розділі розробляється алгоритми програмного забезпечення реалізація та тестування проектних рішень.

Четвертий розділ описує безпеку життєдіяльності та основи охорони праці.

ABSTRACTS

Computerized system for measuring liquid level and cylinder volume for Prom-Dorex LLC. // Bachelor's thesis // Vladyslav Stetsko // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Networks, Group CI-42 // Ternopil, 2024 // p. – 56, figs. – 46, tables – 1, drawings – 4, appendices – 4, bibliography – 24.

Computerized system for measuring liquid level and cylinder volume for Prom-Dorex LLC, ESP32, microcontroller, LCD display, ultrasonic distance sensor.

The bachelor's qualification work developed a computerized system for measuring liquid level and volume for LLC "Prom-Dorex" based on the ESP32 microcontroller.

The qualification work consists of four sections.

The first section analyzes the technical task and reviews possible solutions to the problem.

The second section is dedicated to the development of the system and tools for development.

The third section develops software algorithms, implementation, and testing of design solutions.

The fourth section describes life safety and the basics of labor protection.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	8
1.1 Аналіз вимог до комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс».....	8
1.2 Аналіз можливих рішень поставленого завдання.....	10
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА	13
2.1 Розробка узагальненої структури комп'ютеризованої системи вимірювання рівня рідини та об'єму	13
2.2 Обґрунтування вибору апаратного забезпечення для комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі.....	14
2.2.1 Обґрунтування вибору мікроконтролера	14
2.2.2 Ультразвуковий датчик відстані	16
2.2.3 LCD дисплей I2C.....	19
2.3 Обґрунтування вибору програмного забезпечення для проєктування комп'ютеризованої системи	21
2.3.1 Вибір середовища для написання ПЗ для мікроконтролера.....	21
2.3.2 Вибір середовища для написання ПЗ для веб-сайту	23
2.4 Проєктування комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі.....	25
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА.....	29
3.1 Розробка алгоритму роботи комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі.....	29
3.1.1 Розробка алгоритму для програмного забезпечення мікроконтролера ..	29
3.1.2 Розробка алгоритму роботи веб-сайту	30
3.2 Реалізація проектних рішень	31
3.2.1 Реалізація проектних рішень програмного забезпечення мікроконтролера	31
3.2.2 Реалізація проектних рішень програмного забезпечення веб-сайту	39
3.3 Тестування системи	46

					КС КРБ 123.133.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дат			
Розроб.		Стецько В.Р.			РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ		
Перевір.		Стадник Н.Б.					
Реценз.		Стоянов Ю. М.					
Н. контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Акрушів
					ТНТУ, каф. КС, гр. СІ-42		

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ....	48
4.1 Долікарська допомога при задусі, утопленні.....	48
4.2 Заходи безпеки для захисту організму людини від шкідливого впливу ультразвуку.....	50
4.3 Пожежна профілактика на підприємстві.....	52
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
Додаток А Технічне завдання.....	58
Додаток Б Перелік елементів.....	66
Додаток В Лістинг коду мікроконтролера.....	68
Додаток Ж Лістинг код веб-сторінки.....	71

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасні виробничі процеси вимагають високого рівня точності та ефективності. Зокрема, підприємства, що займаються виробництвом і зберіганням рідин, потребують точних методів вимірювання рівня та об'єму рідин у резервуарах і циліндрах. Точність вимірювань є важливою частиною в оптимізації виробництва, моніторингу даних та мінімізації втрат ресурсів.

ТОВ «Пром-Дорекс» працює у сфері виробництва та зберігання різних рідин. Щоб підвищити ефективність і конкурентоспроможність, компанії потрібно впроваджувати сучасні технології, які автоматизують і покращують точність вимірювальних процесів. Одним з таких рішень є комп'ютеризована система для вимірювання рівня рідини та об'єму в циліндрах. Метою даної дипломної роботи є розробка комп'ютеризованої системи вимірювання рівня рідини та об'єму для експлуатації на підприємстві ТОВ «Пром-Дорекс». Ця система повинна забезпечити точність вимірювань, надійність та зручність у використанні, що суттєво підвищить ефективність виробничих процесів на підприємстві.

В даній роботі будуть розглянуті наступні особливості:

- аналіз існуючих методів та технологій вимірювання рівня та об'єму рідин в циліндрі.
- розробка архітектури комп'ютеризованої системи вимірювання.
- реалізація програмного забезпечення для системи.
- проведення тестування та оцінка ефективності впровадженої системи.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс»

Важливим етапом у розробці комп'ютеризованої системи вимірювання рівня та об'єму рідини в циліндрі для ТОВ «Пром-Дорекс» став детальний аналіз технічного завдання. Даний етап представляє собою основні вимоги до системи, її функціональність, обмеження та параметри. Аналіз даного завдання дозволяє визначити ключові аспекти проекту, які впливають на вибір технології, архітектуру системи та методи реалізації. Цей етап є критичним, оскільки від його якості залежить успішність всієї розробки, включаючи вибір відповідних компонентів, технологій та стратегій реалізації.

Система повинна бути оснащена сенсорами, які здатні точно вимірювати рівень рідини в циліндрі. Це можуть бути ультразвукові, ємнісні або гідростатичні сенсори. Вибір типу сенсорів залежить від характеристик рідини, таких як в'язкість, хімічний склад та інші фізико-хімічні властивості. Наприклад, ультразвукові сенсори можуть бути ефективні для неагресивних рідин з відносно низькою в'язкістю, тоді як ємнісні сенсори можуть бути кращими для рідин з високою електропровідністю. Гідростатичні сенсори, у свою чергу, можуть використовуватись у системах, де важливим є точне вимірювання тиску рідини на дно резервуару.

На основі даних про рівень рідини система повинна автоматично обчислювати об'єм рідини в резервуарі. Для цього необхідно враховувати геометричні параметри циліндра, такі як висота, діаметр, а також можливі варіації форми дна або внутрішніх конструкцій, які можуть впливати на

					КС КРБ 123.133.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>				
<i>Розроб.</i>		<i>Стецько В.Р.</i>			<i>РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Стадник Н.Б.</i>						
<i>Реценз.</i>		<i>Стоянов Ю. М.</i>						
<i>Н. контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						
						<i>ТНТУ, каф. КС, гр. СІ-42</i>		

точність вимірювань. Також варто враховувати температурні зміни, які можуть впливати на об'єм рідини, що також має бути враховано у алгоритмах обчислення.

Система повинна мати зручний і зрозумілий інтерфейс користувача, який дозволяє відображати поточні показники рівня та об'єму рідини. Інтерфейс може бути реалізований у вигляді додатку для комп'ютера або веб-інтерфейсу, який забезпечить доступ до системи з будь-якого пристрою, що підключений до Інтернету.

Для реалізації вимірювання рівня рідини необхідно обрати відповідне обладнання: сенсори, контролери, модулі збору даних та засоби передачі інформації. Важливо врахувати умови експлуатації, такі як температура, вологість, можливість контакту з агресивними речовинами, а також вимоги до точності вимірювань. Наприклад, у промислових умовах сенсори повинні бути стійкими до пилу та води, мати захист від механічних пошкоджень та бути здатними працювати у широкому діапазоні температур.

Розробка програмного забезпечення повинна включати створення алгоритмів для обробки даних з сенсорів, обчислення об'єму рідини, інтерфейсу користувача та системи сповіщень. Програмне забезпечення має бути надійним, легко масштабованим та забезпечувати швидку обробку даних. Важливо забезпечити можливість легкого оновлення програмного забезпечення для додавання нових функцій та виправлення можливих помилок. Для цього може використовуватись модульний підхід до розробки, що дозволяє додавати або змінювати окремі компоненти системи без необхідності повного перероблення всього програмного забезпечення.

Для забезпечення взаємодії між різними компонентами системи необхідно визначити комунікаційні протоколи, які будуть використовуватися для передачі даних. Це можуть бути дротові або бездротові протоколи, такі як Modbus, CAN, Zigbee або Wi-Fi. Вибір протоколу залежить від вимог до швидкості передачі даних, надійності з'єднання, відстані між компонентами та

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

умов експлуатації. Наприклад, у випадку великого виробничого підприємства може бути доцільним використання дротових протоколів для забезпечення стабільності зв'язку, тоді як у мобільних або тимчасових системах можуть бути більш зручними бездротові рішення.

1.2 Аналіз можливих рішень поставленого завдання

Розробка комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» вимагає всебічного аналізу можливих технічних рішень. Необхідно розглянути різні підходи до вимірювання, обробки та передачі даних, а також інтеграції системи з іншими компонентами інфраструктури підприємства. Це дозволить вибрати оптимальне рішення, яке відповідатиме вимогам замовника та забезпечить високу надійність і точність вимірювань.

Ультразвукові сенсори використовують звукові хвилі для визначення рівня рідини. Переваги цього методу включають високу точність та відсутність контакту з рідиною, що важливо для агресивних або забруднених середовищ. Ультразвукові сенсори можуть працювати в широкому діапазоні температур і тисків, що робить їх універсальними. Недоліки включають можливі помилки вимірювань у випадку наявності пінки на поверхні рідини або значних турбулентностей.

Ємнісні сенсори вимірюють зміну ємності, викликану зміною рівня рідини. Вони є відносно простими у виготовленні та обслуговуванні, а також забезпечують високу точність вимірювань. Однак, ємнісні сенсори можуть бути чутливими до забруднення поверхні сенсора та до змін хімічного складу рідини.

Гідростатичні сенсори вимірюють тиск, який створює рідина на певній глибині, і на основі цього обчислюють рівень рідини. Цей метод є дуже точним

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

і стабільним у часі. Основні обмеження пов'язані з необхідністю калібрування для різних типів рідин та впливом температури на точність вимірювань.

Реалізація даної комп'ютеризованої системи передбачає вибір із трьох методів обробки та передачі даних: локальна обробка даних, віддалена обробка даних та гібридний підхід.

Локальна обробка даних передбачає використання контролерів або вбудованих систем для обробки даних безпосередньо на місці вимірювання. Це дозволяє зменшити затримки у передачі даних і знизити залежність від мережевої інфраструктури. Основні переваги включають високу швидкість реакції та автономність. Недоліки можуть включати обмежені обчислювальні ресурси та складність оновлення програмного забезпечення.

Віддалена обробка даних передбачає передачу вимірюваних даних на централізований сервер або хмарну платформу для подальшої обробки та аналізу. Цей підхід забезпечує високу масштабованість та гнучкість в обробці великих обсягів даних. Однак, він може вимагати високої пропускну здатності мережі та стійкості до збоїв у зв'язку.

Гібридний підхід поєднує локальну та віддалену обробку даних, дозволяючи здійснювати попередню обробку на місці вимірювання і передавати тільки оброблені дані або важливі події на центральний сервер. Це дозволяє знизити вимоги до мережі та підвищити надійність системи.

Локальні інтерфейси можуть бути реалізовані у вигляді дисплеїв або панелей керування, розташованих безпосередньо на обладнанні. Вони забезпечують швидкий доступ до основних параметрів та налаштувань, але обмежують можливості віддаленого контролю.

Веб-інтерфейси дозволяють користувачам отримувати доступ до системи через браузер з будь-якого пристрою, підключеного до мережі. Це забезпечує високу гнучкість та зручність у використанні, а також спрощує інтеграцію з іншими веб-сервісами та додатками.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Мобільні додатки забезпечують користувачам можливість моніторингу та управління системою з мобільних пристроїв. Це особливо корисно для оперативного реагування на аварійні ситуації та отримання сповіщень у режимі реального часу.

Одним з основних критеріїв вибору технології є точність та надійність вимірювань. Ультразвукові сенсори забезпечують високу точність і надійність в різних умовах експлуатації. Гідростатичні сенсори також є дуже точними, але потребують регулярного калібрування. Ємнісні сенсори можуть бути менш надійними у випадку забруднення або змін хімічного складу рідини.

Вартість реалізації системи включає вартість обладнання, встановлення, обслуговування та програмного забезпечення. Гідростатичні сенсори зазвичай є найдешевшими, однак їх експлуатаційні витрати можуть бути вищими через необхідність частого калібрування. Ультразвукові сенсори можуть мати вищу початкову вартість, але менші витрати на обслуговування.

Вибір архітектури обробки даних значно впливає на гнучкість та масштабованість системи. Віддалена обробка даних і веб-інтерфейси забезпечують високу гнучкість та легкість масштабування, але вимагають надійного мережевого підключення. Локальна обробка даних може бути менш гнучкою, але забезпечує високу автономність.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Розробка узагальненої структури комп'ютеризованої системи вимірювання рівня рідини та об'єму

Для кращого розуміння роботи комп'ютеризованої системи розглянемо структурну схему на рис. 2.1.

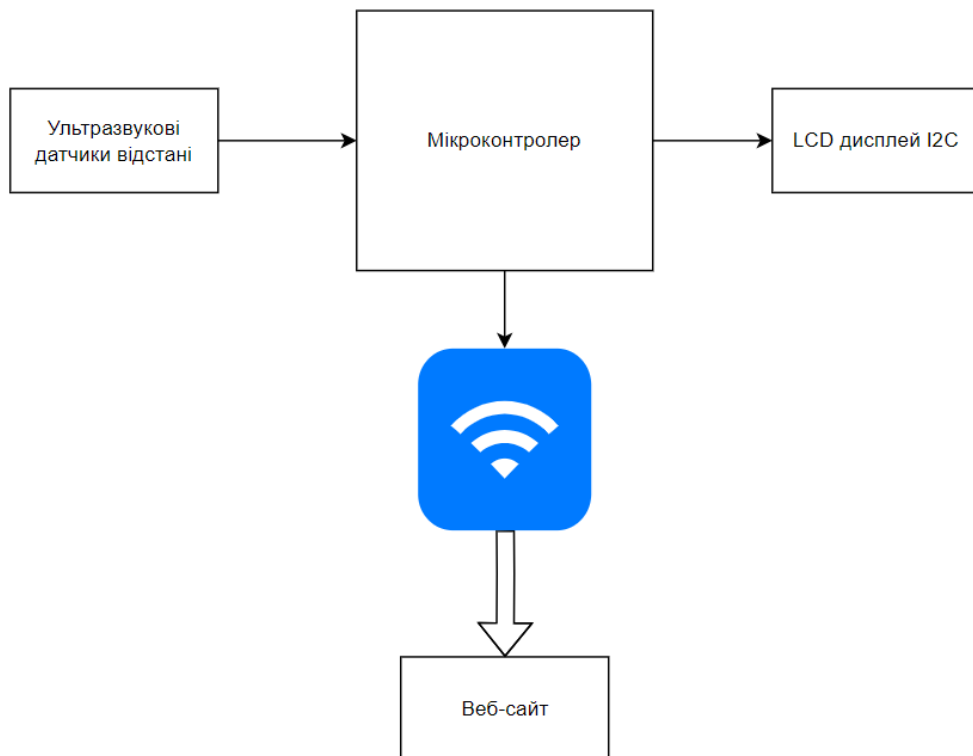


Рисунок 2.1 – Структурна схема комп'ютеризованої системи

З наведеної структурної схеми видно, що в основі системи лежить мікроконтролер ESP32 devkit v1. Дані з датчика надходять до мікроконтролера, який послідовно виводить їх на РК-дисплей за допомогою кнопки, а також відображає на веб-сайті через Wi-Fi-з'єднання [8].

					КС КРБ 123.133.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дат	РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА		
Розроб.		Стецько В.Р.			Літ.	Арк.	Акрушів
Перевір.		Стадник Н.Б.					
Реценз.		Стоянов Ю. М.			ТНТУ, каф. КС, гр. СІ-42		
Н. контр.		Тиш Є.В.					
Затверд.		Осухівська Г.М.					

Комп'ютеризована система складається з наступних елементів:

- Мікроконтролер ESP32 devkit v1.
- LCD 1602 I2C.
- Ультразвукові датчики відстані HC-SR04.

2.2 Обґрунтування вибору апаратного забезпечення для комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі

2.2.1 Обґрунтування вибору мікроконтролера

На основі детального аналізу та порівняння різних мікроконтролерів було прийнято рішення обрати плату ESP32 devkit v1. Цей мікроконтролер вирізняється рядом переваг, які роблять його оптимальним вибором для нашої системи (рис. 2.2).



Рисунок 2.2 – Плата мікроконтролера ESP32 devkit v1

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

ESP32 devkit v1 — це потужний та універсальний мікроконтролер, що має ряд ключових компонентів, які забезпечують його функціональність та гнучкість у застосуванні. Центральний процесор мікроконтролера представлений двоядерним процесором Tensilica Xtensa LX6 з тактовою частотою до 240 МГц, що дозволяє виконувати складні обчислювальні завдання та обробляти великі обсяги даних. Вбудована SRAM пам'ять обсягом до 520 КБ використовується для зберігання тимчасових даних та виконання програм, тоді як Flash-пам'ять, зазвичай обсягом 4 МБ, призначена для зберігання прошивки та постійних даних [9].

Однією з ключових особливостей ESP32 devkit v1 є вбудовані модулі Wi-Fi та Bluetooth. Wi-Fi підтримує стандарти 802.11 b/g/n, що дозволяє підключатися до бездротових мереж, тоді як Bluetooth підтримує як класичний режим, так і BLE (Bluetooth Low Energy), забезпечуючи підключення різноманітних пристроїв та передачу даних без використання проводів.

Мікроконтролер також має широкий спектр інтерфейсів вводу/виводу (I/O), включаючи численні цифрові та аналогові порти. Генеральні цифрові ввід/вивід (GPIO) порти дозволяють підключати різноманітні датчики та виконавчі механізми, забезпечуючи високу гнучкість у розробці систем. Важливою особливістю є наявність інтерфейсу I2C, що дозволяє підключати кілька периферійних пристроїв, використовуючи всього дві лінії зв'язку — SDA (дані) та SCL (синхронізація). Ці протоколи додатково розширюють можливості підключення та інтеграції різноманітних периферійних пристроїв, роблячи систему ще більш універсальною. Висока швидкість передачі даних і надійність цих інтерфейсів роблять його ідеальним вибором для широкого спектру застосувань, від простих сенсорних систем до складних автоматизованих рішень (рис. 2.3).

Окрім того, ESP32 devkit v1 підтримує популярні платформи для розробки програмного забезпечення, такі як Arduino IDE та PlatformIO. Це значно спрощує процес написання, налагодження та тестування програм,

						КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

дозволяючи розробникам швидко створювати прототипи та вносити зміни до програмного забезпечення. Доступна ціна мікроконтролера робить його економічно вигідним вибором, поєднуючи високу функціональність та продуктивність з помірною вартістю, що дозволяє знизити загальні витрати на розробку [10].

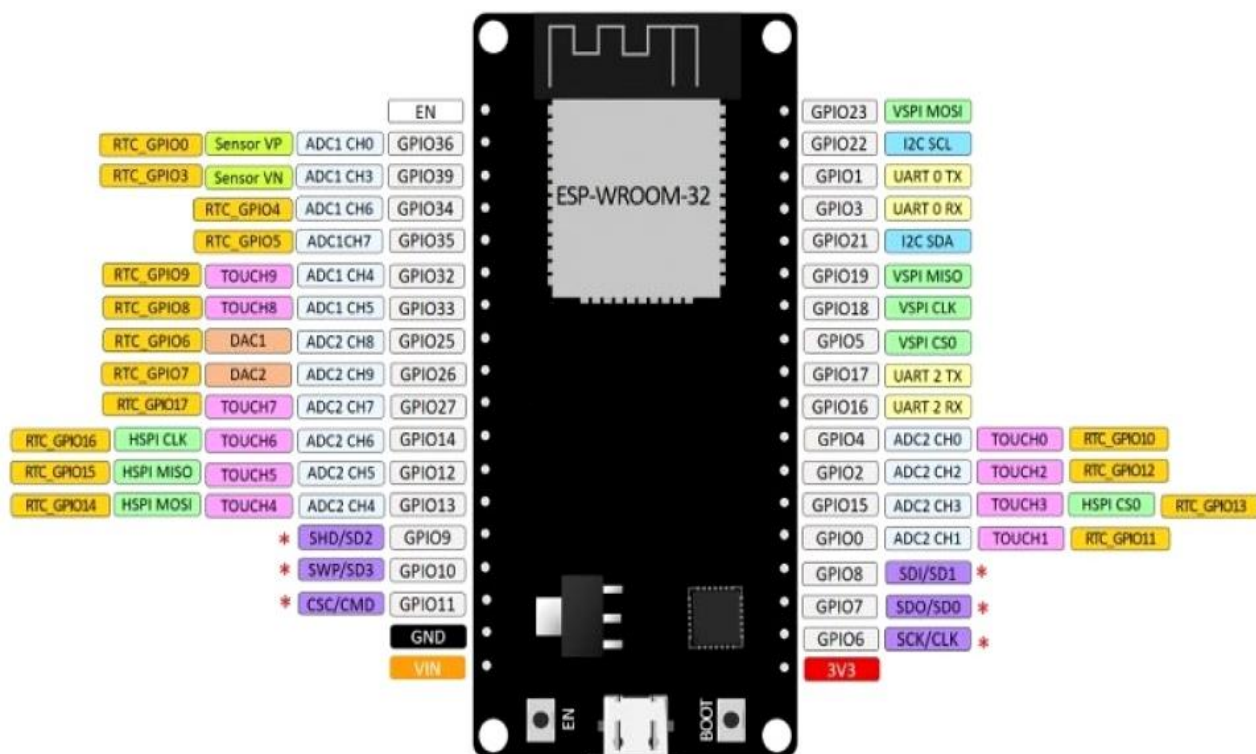


Рисунок 2.3 – Розпінування мікроконтролера ESP32 devkit v1

2.2.2 Ультразвуковий датчик відстані

Ультразвуковий датчик відстані HC-SR04 є одним із найпоширеніших сенсорів, що використовуються для вимірювання відстані в різноманітних автоматизованих системах. Він заснований на принципі ехолокації, що полягає у визначенні відстані до об'єкта за допомогою відбитих ультразвукових хвиль. Основні сфери застосування такого датчика включають робототехніку, системи автоматичного керування, системи безпеки, а також спеціалізовані вимірювальні системи, такі як комп'ютеризовані системи вимірювання рівня рідини та об'єму в циліндрі (рис. 2.4).

										Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						



Рисунок 2.4 – Вигляд ультразвукового датчика HC-SR04

HC-SR04 складається з двох основних компонентів: випромінювача та приймача, а також контролера для обробки сигналів. Випромінювач посилає ультразвукові хвилі з частотою 40 кГц. Після того, як хвиля досягає об'єкта, вона відбивається і повертається до датчика, де її захоплює приймач. Контролер обчислює час, який хвиля витрачає на подорож в обидва напрямки, і за допомогою цього значення визначає відстань до об'єкта.

Принцип роботи HC-SR04 можна розбити на декілька основних етапів. Перший етап – це ініціація вимірювання. Для цього на вхід Trig подається короткий імпульс високого рівня тривалістю не менше 10 мікросекунд. Це призводить до того, що випромінювач генерує ультразвуковий імпульс тривалістю 8 циклів ($8 * 1/40 \text{ кГц} = 200 \text{ мікросекунд}$). Після цього приймач очікує на відбитий сигнал. Коли сигнал повертається, на виході Echo з'являється імпульс, тривалість якого пропорційна відстані до об'єкта [11].

Другий етап – це очікування відбитого сигналу. Після генерації ультразвукового імпульсу, приймач знаходиться у режимі очікування, поки сигнал не відіб'ється від об'єкта і не повернеться назад.

Третій етап – це обчислення відстані. Контролер вимірює час між відправленням і прийомом сигналу і за допомогою цього значення розраховує відстань до об'єкта. Оскільки швидкість звуку у повітрі відома (близько 343 метри в секунду), контролер може легко перетворити час у відстань.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

HC-SR04 має наступні технічні характеристики, які наведено в табл. 2.1:

Таблиця 2.1 – Технічні характеристики HC-SR04

Назва характеристики	Значення
Напруга живлення	5 В
Споживаний струм	<2 мА
Діапазон вимірювання	2 см - 400 см
Точність вимірювання	±3 мм
Кут вимірювання	15°

Ці характеристики роблять датчик придатним для використання у багатьох різних застосуваннях, де потрібні точні вимірювання відстані на відносно невеликих відстанях.

HC-SR04 має чотири виводи: VCC - підключається до джерела живлення 5 В, Trig - підключається до цифрового виходу мікроконтролера для ініціалізації сигналу, Echo - підключається до цифрового входу мікроконтролера для отримання відбитого сигналу, GND - підключається до землі (рис. 2.5).

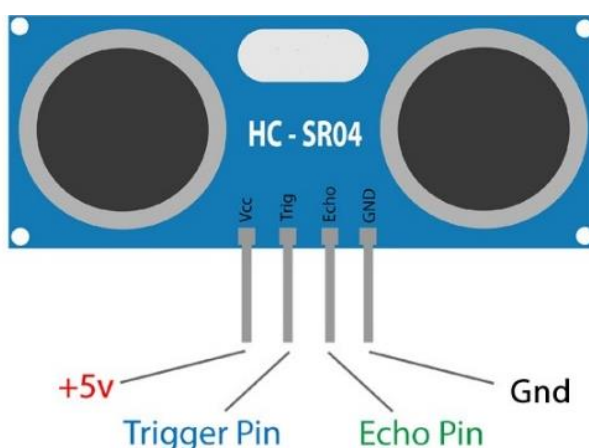


Рисунок 2.5 – Розпінування ультразвукового датчика HC-SR04

Процес підключення датчика до мікроконтролера є відносно простим і не вимагає спеціалізованих знань. Основне завдання полягає в правильному з'єднанні відповідних виводів та налаштуванні програмного забезпечення для роботи з датчиком.

HC-SR04 є ключовим компонентом комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс». Датчик встановлюється над циліндром і спрямовує ультразвуковий сигнал вниз до поверхні рідини. Відбитий сигнал повертається до датчика, що дозволяє обчислити відстань до поверхні рідини. Знаючи геометричні параметри циліндра, можна легко обчислити об'єм рідини в ньому

2.2.3 LCD дисплей I2C

LCD дисплеї (Liquid Crystal Display) є одними з найпоширеніших засобів відображення інформації у сучасних електронних пристроях. Їх популярність обумовлена низьким енергоспоживанням, високою якістю зображення та відносною простотою інтеграції в різноманітні системи. У рамках розробки комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс», використання LCD дисплею з інтерфейсом I2C дозволяє забезпечити ефективний та зручний спосіб візуалізації даних, які отримуються від сенсорів (рис. 2.6).

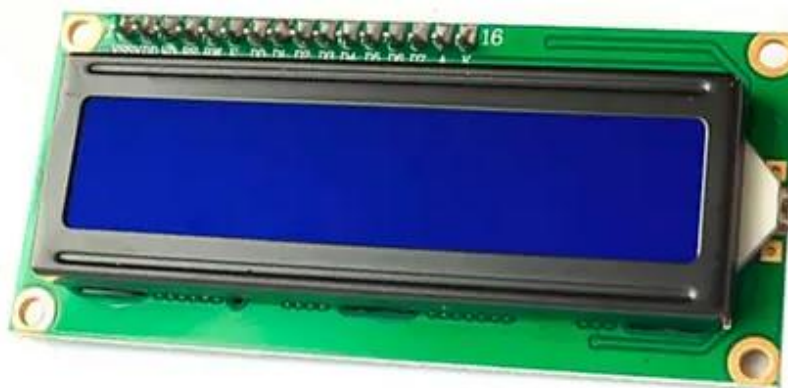


Рисунок 2.6 – Вигляд LCD дисплея

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

LCD дисплеї працюють на основі технології рідких кристалів, які змінюють свою прозорість під дією електричного поля. Пікселі дисплею складаються з рідких кристалів, розміщених між двома шарами скла. Коли до пікселя прикладається напруга, рідкі кристали змінюють своє положення, що змінює кількість світла, яке проходить через них. Це дозволяє формувати зображення на дисплеї [12].

Використання інтерфейсу I2C для керування LCD дисплеєм значно спрощує процес підключення та налаштування пристрою. I2C (Inter-Integrated Circuit) — це синхронний серійний інтерфейс, який дозволяє підключати до мікроконтролера велику кількість периферійних пристроїв, використовуючи лише дві лінії: SDA (Serial Data) і SCL (Serial Clock). Завдяки цьому знижується кількість необхідних з'єднань та спрощується схема підключення.

Інтеграція LCD дисплею з I2C інтерфейсом до системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» дозволяє забезпечити зручний користувацький інтерфейс для відображення результатів вимірювань. На дисплеї можуть відображатися поточні показники рівня рідини, об'єму.

Використання адаптера PCF8574 для підключення LCD дисплею до I2C інтерфейсу додатково спрощує процес інтеграції. PCF8574 є 8-бітовим розширювачем вводу-виводу для I2C шини, що дозволяє значно зменшити кількість проводів, необхідних для підключення дисплею. Це особливо актуально в системах з обмеженим простором для прокладки проводів або коли необхідно підключити кілька периферійних пристроїв до одного мікроконтролера. Крім того, використання адаптера полегшує процес програмування, оскільки більшість бібліотек для роботи з LCD дисплеями через I2C вже підтримують роботу з PCF8574. Завдяки цьому можна зконцентруватись на функціональності системи, а не на технічних деталях підключення що значно спрощує час розробки. Вигляд адаптера зображений на рис. 2.7.

						КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

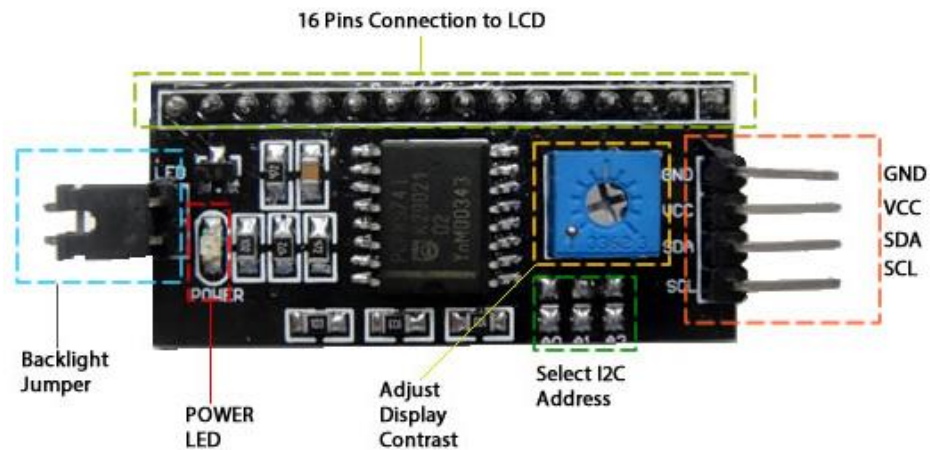


Рисунок 2.7 – Вигляд адаптера PCF8574

Підключення LCD дисплею до мікроконтролера через адаптер PCF8574 та I2C інтерфейс здійснюється шляхом використання відповідних бібліотек, які забезпечують високорівневе керування дисплеєм. Однією з найбільш поширених бібліотек для роботи з LCD дисплеями в Arduino є LiquidCrystal_I2C. Вона надає зручні методи для ініціалізації дисплею, виводу тексту, очищення екрану тощо. Це значно спрощує процес розробки програмного забезпечення для системи [13].

2.3 Обґрунтування вибору програмного забезпечення для проєктування комп'ютеризованої системи

2.3.1 Вибір середовища для написання ПЗ для мікроконтролера

Для розробки комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» було обрано програмне забезпечення ArduinoIDE. Система працює на базі мікроконтролера ESP32, який програмується за допомогою ArduinoIDE. Вибір даного середовища розробки ПЗ базується на таких ключових перевагах, як простота використання, доступність бібліотек та прикладів коду, а також сумісність з ESP32.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

ArduinoIDE має інтуїтивно зрозумілий інтерфейс, що робить його доступним для розробників різних рівнів підготовки. Це особливо важливо для роботи з мікроконтролером ESP32, оскільки розробники можуть швидко освоїти основи програмування і створювати складні системи завдяки зрозумілій структурі коду та простим прикладам. Наявність великої кількості навчальних матеріалів, форумів і спільнот допомагає новачкам швидко включитися в роботу і вирішувати виникаючі проблеми [14].

Однією з важливих особливостей цього середовища є підтримка великої кількості бібліотек, що значно спрощує роботу з різноманітними датчиками та компонентами. Наприклад, для вимірювання рівня рідини можна використовувати ультразвукові або ємнісні датчики, для яких вже існують готові бібліотеки та приклади коду. Це скорочує час розробки та знижує ймовірність помилок, дозволяючи зосередитися на специфічних завданнях проекту.

Крім того, ArduinoIDE підтримує мікроконтролер ESP32, що є важливою перевагою для нашої системи. ESP32 надає потужні можливості для роботи з бездротовими мережами, має високу продуктивність та підтримує широкий спектр периферійних пристроїв. Завдяки цьому, можна легко інтегрувати ESP32 в проект, використовуючи ArduinoIDE для програмування та налаштування. Підтримка Wi-Fi і Bluetooth дозволяє розробляти проекти з дистанційним моніторингом та керуванням, що додає додаткову цінність системі.

Середовище розробки ПЗ ArduinoIDE забезпечує високу гнучкість та можливість розширення функціоналу системи. Модульна структура коду дозволяє легко додавати нові функції та змінювати існуючі. Це важливо для проектів, які можуть змінюватися та вдосконалюватися з часом. Наприклад, систему можна доповнити функціями дистанційного моніторингу, автоматичного керування насосами та іншими можливостями. (рис. 2.8).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

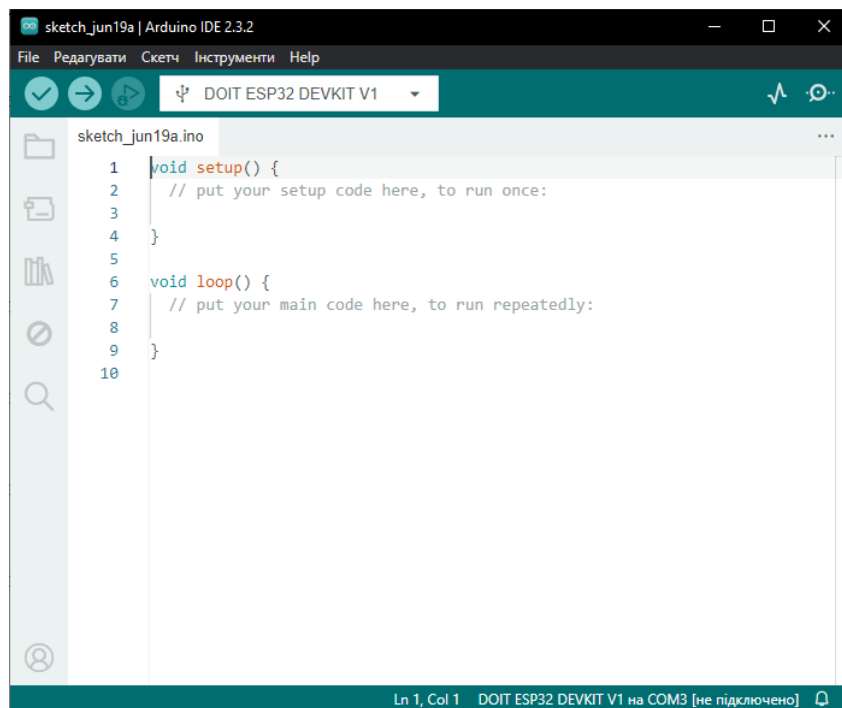


Рисунок 2.8 – Видгляд програмного середовища ArduinoIDE

2.3.2 Вибір середовища для написання ПЗ для веб-сайту

Для створення веб-сайту, що є частиною автоматизованої системи вимірювання рівня та об'єму рідини в циліндрі для ТОВ "Пром-Дорекс", було обрано Visual Studio Code (VS Code). Цей вибір зумовлений його багатофункціональністю, зручністю використання, широкою підтримкою мов програмування, інтеграцією з іншими інструментами та наявністю численних розширень, що роблять процес розробки ефективним та зручним [15].

VS Code пропонує розробникам зручне та багатофункціональне середовище з інтуїтивно зрозумілим інтерфейсом. Його функції автодоповнення коду, підсвічування синтаксису та вбудований термінал значно спрощують процес написання коду та зменшують кількість помилок. Завдяки таким можливостям, розробники можуть швидше та якісніше реалізовувати свої ідеї.

Однією з ключових переваг VS Code є підтримка широкого спектру мов програмування, включаючи HTML, CSS, JavaScript, а також серверні мови,

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

такі як Node.js та Python. Це забезпечує гнучкість у виборі інструментів для розробки веб-сайту та дозволяє інтегрувати різні технології в один проект.

Інтеграція VS Code з системами контролю версій, такими як Git, Docker та іншими інструментами, гарантує ефективне керування проектом та стабільність середовища розробки. Це дозволяє команді розробників зручно та безпечно співпрацювати над проектом, відстежувати зміни та контролювати якість коду.

Visual Studio Marketplace пропонує безліч розширень для форматування коду, аналізу, відлагодження та роботи з популярними фреймворками (React, Angular, Vue.js), що полегшує розробку та підвищує продуктивність. Кожне з цих розширень можна налаштувати відповідно до потреб проекту, що робить процес розробки більш адаптивним та зручним.

Наявність вбудованих засобів для виправлення помилок різних мов програмування дозволяє відлагоджувати код безпосередньо в середовищі розробки, роблячи програмне забезпечення більш надійним та стабільним. Це знижує час на пошук та виправлення помилок, забезпечуючи високу якість кінцевого продукту.

Конфігураційні файли VS Code дають змогу налаштувати середовище розробки під конкретні потреби проекту, забезпечуючи оптимальні умови для продуктивної роботи. Це дозволяє розробникам створювати зручне середовище для написання коду, що відповідає їхнім особистим потребам та вимогам проекту.

Важливо зазначити, що VS Code є безкоштовним програмним забезпеченням з відкритим кодом, що знижує витрати на розробку та робить його доступним для ТОВ "Пром-Дорекс". Це дозволяє компанії інвестувати ресурси в інші важливі аспекти проекту, не жертвуючи якістю інструментів для розробки.

Завдяки своїм численним функціям та можливостям, Visual Studio Code зарекомендував себе як універсальний інструмент для розробки веб-сайту,

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

роблячи його ідеальним вибором для проекту ТОВ "Пром-Дорекс". Його використання забезпечує високу якість, ефективність та зручність розробки, що є ключовими факторами для успішного впровадження автоматизованої системи вимірювання рівня та об'єму рідини. (рис. 2.9).

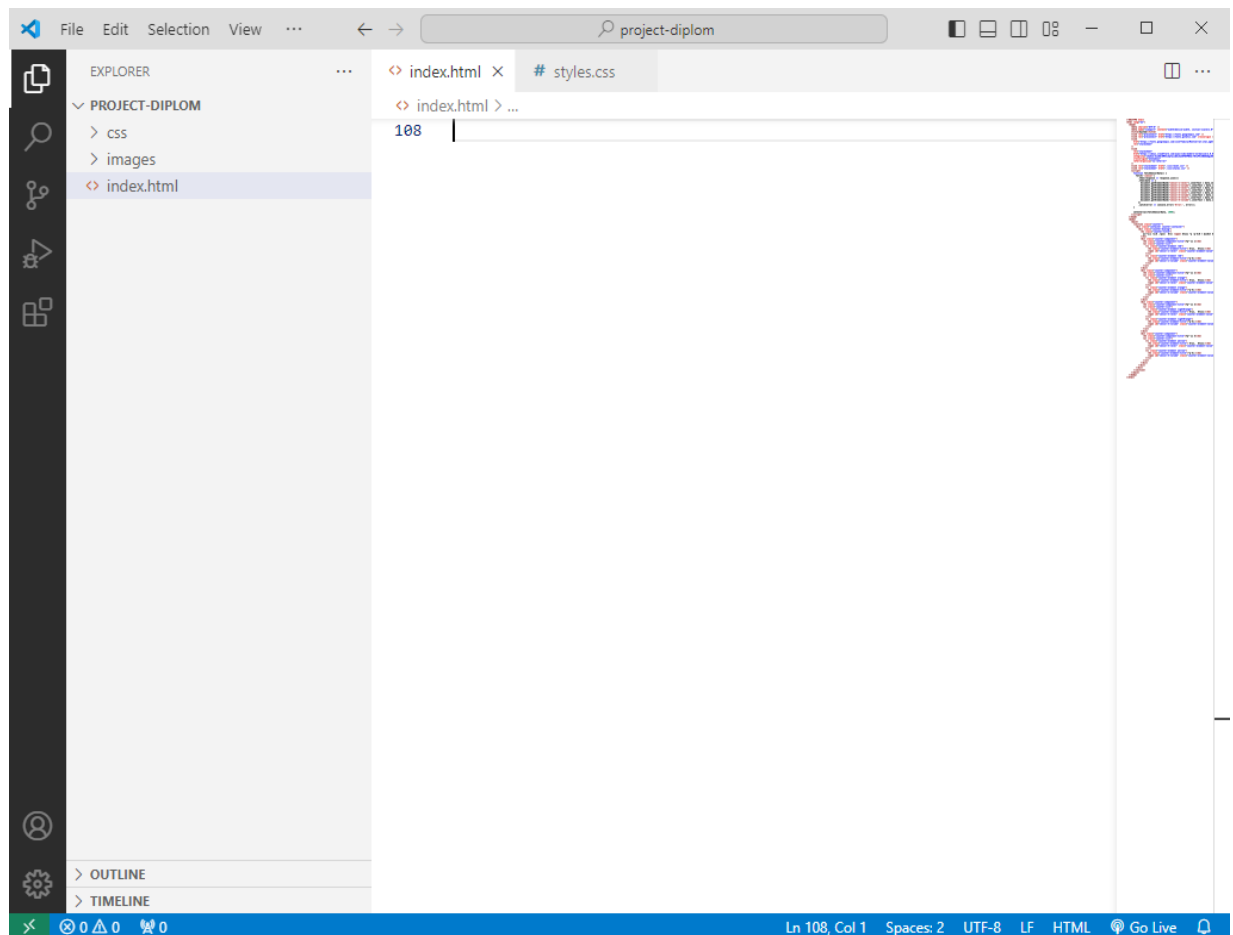


Рисунок 2.9 – Вигляд середовища Visual Studio Code

2.4 Проектування комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі

Основна платформа схеми – це плата ESP32 devkit v1, що відповідає за управління всіма модулями системи. Вона виконує центральні функції керування та координації іншими компонентами в системі. Плата включає порти GPIO, ADC і DAC (рис. 2.10).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

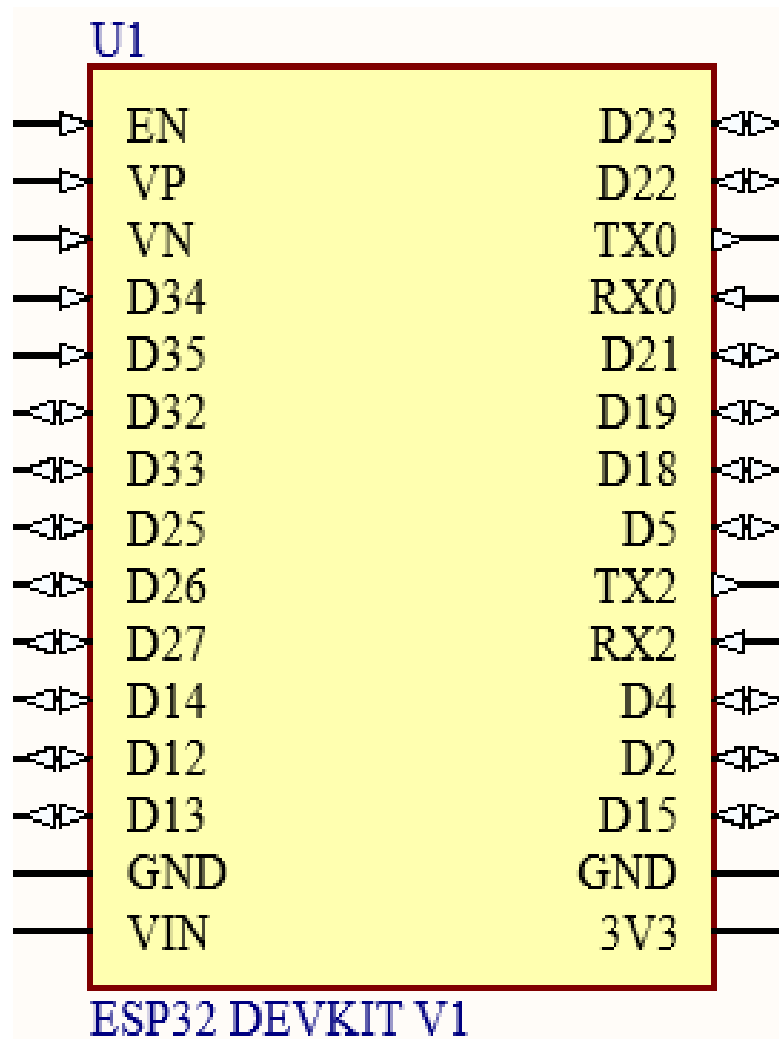


Рисунок 2.10 – Електрична принципова схема плати

На рис. 2.11 показано схему підключення ультразвукових датчиків та до плати ESP32 devkit v1. Ультразвукові датчики з'єднані через макетну плату з контактами живлення (V_{CC}) і землі (GND). Сигнальні контакти TRIG датчиків підключені до виводів 13, 12, 14, 27, а контакти ECHO - до виводів 26, 25, 33, 32.

Кнопка підключена до контакту 2 на основній платі, як зображено на рис. 2.12.

Ця конфігурація дозволяє ESP32 контролювати та зчитувати дані від ультразвукових датчиків і кнопки, що спрощує розробку системи інтеграції сенсорів і управління на основі мікроконтролера ESP32.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

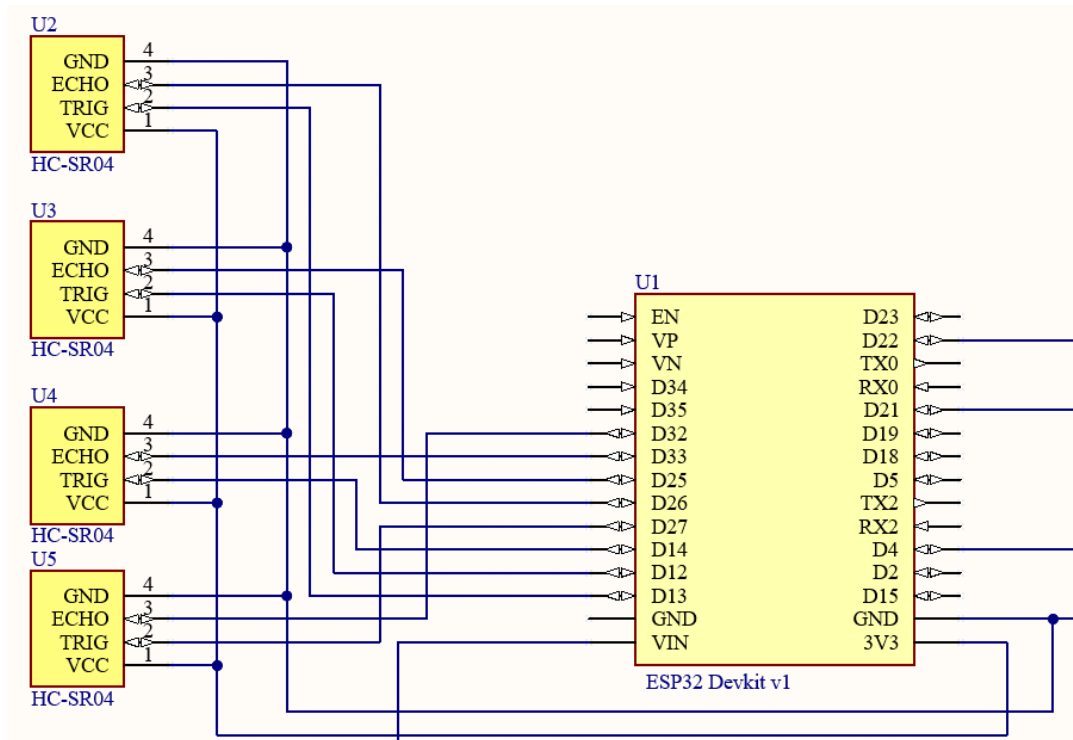


Рисунок 2.11 – Електрична принципова схема підключення ультразвукових датчиків

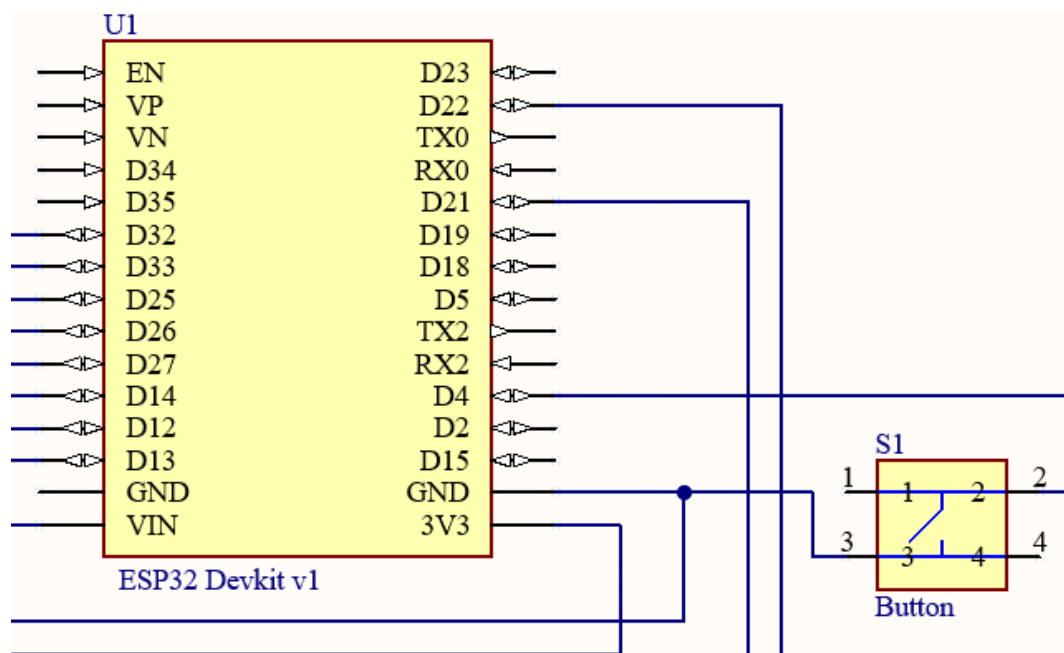


Рисунок 2.12 – Електрична принципова схема підключення та кнопки

Підключення адаптера PCF8574T до LCD 1602 дисплея зазвичай виконується через шину I2C. PCF8574T служить як розширювач виводів, що

дозволяє керувати LCD дисплеєм за допомогою всього двох проводів (SDA і SCL), що значно спрощує підключення. (рис. 2.13).

Виводи з дисплею LCD1602 DB4, DB5, DB6, DB7, підключені відповідно до адаптера в P4, P5, P6, P7, а RS, RW, CS, DB0, підключаються відповідно в P0, P1, P2, P3

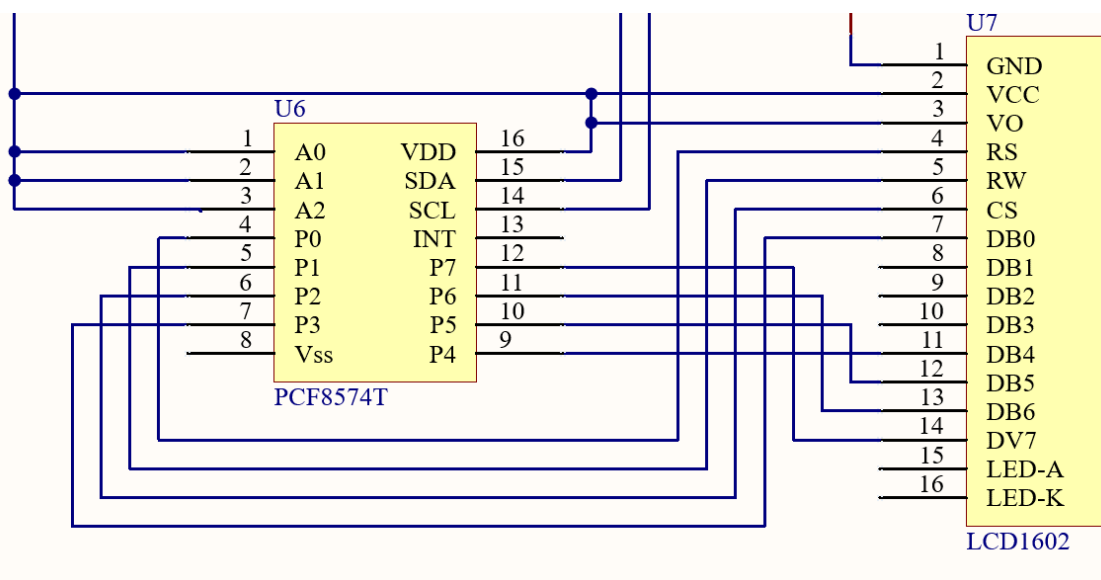


Рисунок 2.13 – Об'єднання адаптера та LCD дисплея

Це забезпечує зв'язок між мікроконтролером (наприклад, ESP32) і LCD дисплеєм через I2C, дозволяючи керувати виводами і відображенням інформації на дисплеї безпосередньо з ESP32 за допомогою PCF8574T.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Розробка алгоритму роботи комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі

3.1.1 Розробка алгоритму для програмного забезпечення мікроконтролера

Блок-схема алгоритму роботи програми представлена на рис. 3.1.

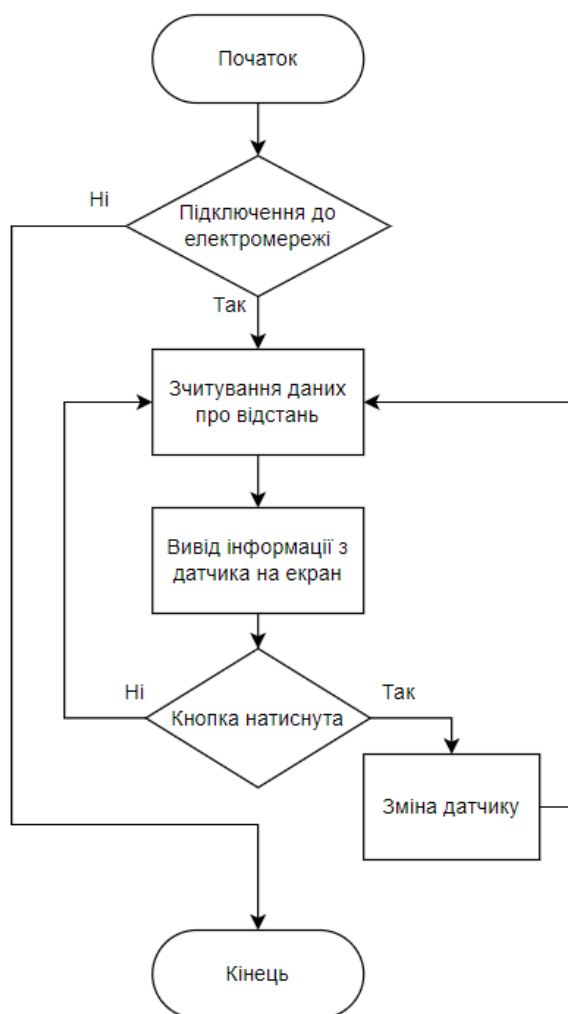


Рисунок 3.1 – Блок-схема алгоритму роботи програми

					КС КРБ 123.133.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>	<i>РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Розроб.</i>		<i>Стецько В.Р.</i>						
<i>Перевір.</i>		<i>Стадник Н.Б.</i>						
<i>Реценз.</i>		<i>Стоянов Ю. М.</i>						
<i>Н. контр.</i>		<i>Тиш Є.В.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

Якщо система підключена до електромережі, то здійснюється зчитування даних про відстань з датчиків. Після цього виводиться інформація про перший датчик. Далі йде перевірка стану кнопки, якщо вона натиснута то на дисплеї з'являється інформація про наступний датчик, якщо ні то на дисплеї залишається інформація про поточний датчик.

3.1.2 Розробка алгоритму роботи веб-сайту

Блок-схема алгоритму роботи веб-сайту представлена на рис. 3.2.

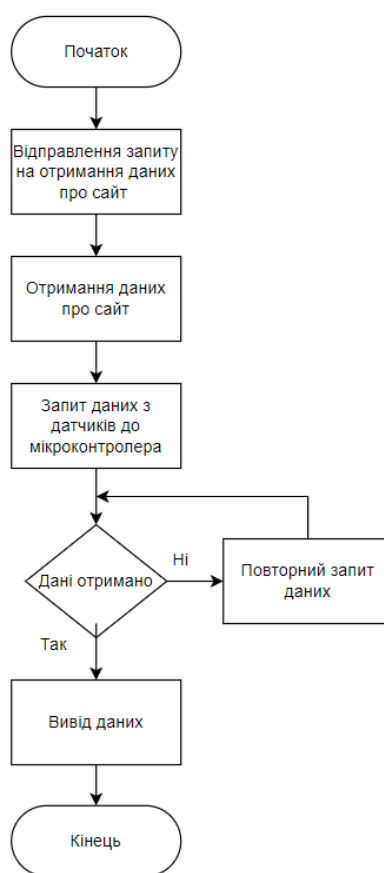


Рисунок 3.2 – Блок-схема алгоритму роботи веб-сайту

Спочатку клієнт відправляє запит на отримання даних про сайт. Після отримання даних про сайт надсилається запит даних з датчиків до мікроконтролера, якщо дані отримано то вони будуть відображатись на сайті, а якщо ні то клієнт повторно надсилає запит на отримання даних.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Реалізація проектних рішень

3.2.1 Реалізація проектних рішень програмного забезпечення мікроконтролера

Для початку нам необхідно додати необхідну плату для нашого проекту в програмному середовищі Arduino IDE. Для цього відкриваємо “Налаштування”, і вставляємо URL-адресу посилання для менеджера плат (рис. 3.3).

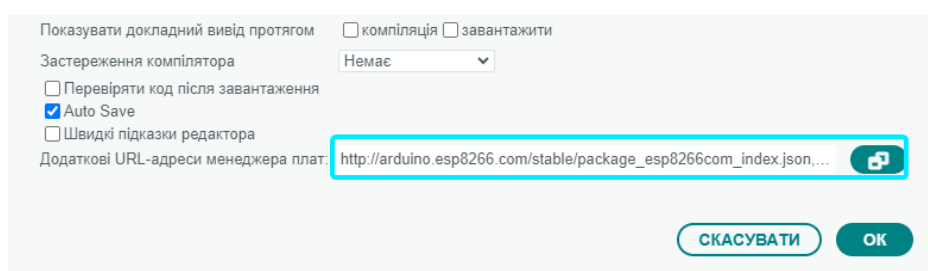


Рисунок 3.3 – Підключення менеджера плат

Далі переходимо в вибір плат для проекту та вибираємо плату “DOIT ESP32 DEVKIT V1”. Також необхідно задати порт “COM3 Serial Port(USB)” (рис. 3.4).

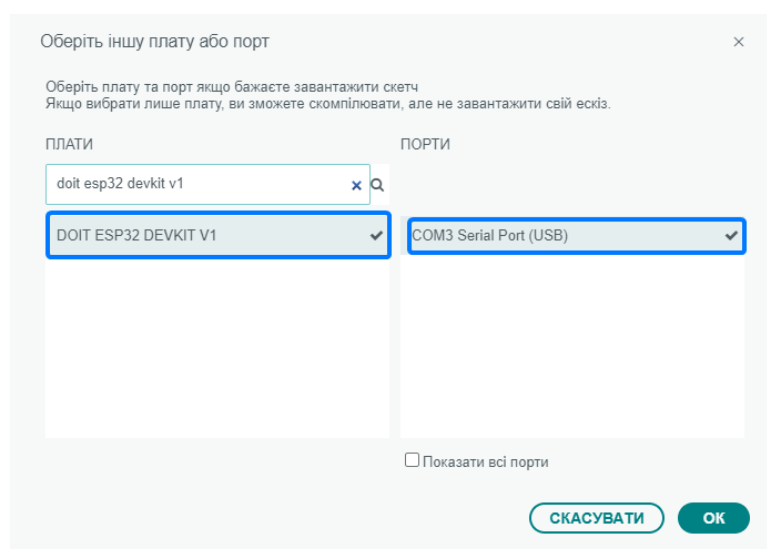


Рисунок 3.4 – Вибір плати для проекту

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Тепер необхідно встановити бібліотеки. Для цього відкриваємо “Менеджер бібліотек”, натиснувши на необхідну кнопку на боковій панелі, і в пошуковому полі вводимо назви бібліотек та встановлюємо їх (рис. 3.5).

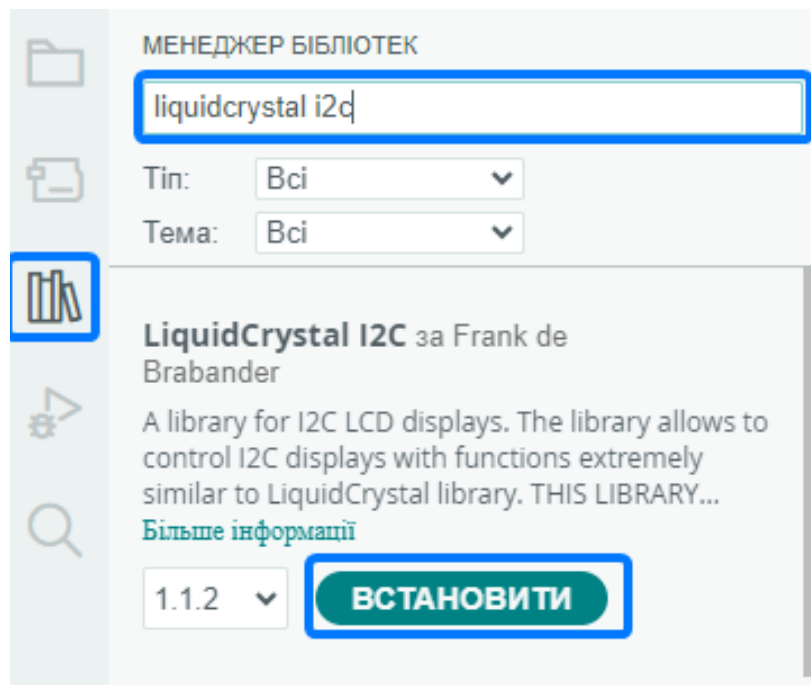


Рисунок 3.5 – Встановлення бібліотек

Тепер розглянемо бібліотеки необхідні для написання ПЗ:

- `LiquidCrystal_I2C.h` – дана бібліотека використовується для роботи з LCD дисплеєм. Вона спрощує взаємодію з дисплеями через шину I2C, дозволяючи легко керувати виводом тексту та підсвіткою.

- `Ultrasonic.h` – призначена для роботи з ультразвуковими датчиками відстані, в даній системі з HC-SR04. Основні можливості бібліотеки включають ініціалізацію датчика з зазначенням пінів TRIG і ECHO, вимірювання часу відлуння та обчислення відстані в сантиметрах або дюймах.

- `WiFi.h` – використовується для відключення до мережі Wi-Fi з використанням мікроконтролера ESP32 devkit v1. Дана бібліотека надає функції для сканування доступних мереж, підключення до них налаштування точок доступу.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

- `ESPAsyncWebServer.h` – створена для асинхронного управління веб-сервером на базі мікроконтролерів ESP32 або ESP8266. Вона дозволяє створювати веб-сервери, які можуть обробляти запити HTTP асинхронно, що підвищує ефективність та швидкість роботи, особливо при обробці великої кількості одночасних запитів.

- `SPIFFS.h` – призначена для роботи з SPIFFS (SPI Flash File System) на мікроконтролерах ESP32 та ESP8266. Вона дозволяє використовувати вбудовану флеш-пам'ять як файлову систему, що дає можливість зберігати та читати файли, такі як конфігураційні файли, веб-сторінки, зображення та інші дані.

```
const char* ssid = "WASD";  
const char* password = "34666616";  
AsyncWebServer server(80);
```

Рисунок 3.6 – Лістинг коду для підключення до Wi-Fi та налаштування веб-сервера

Цей фрагмент коду ініціалізує налаштування для підключення плати до Wi-Fi мережі та створення веб-сервера, де `AsyncWebServer server(80)` створює об'єкт `server` типу `AsyncWebServer`, який приймає HTTP-запити на порту 80, а `const char* ssid = "WASD"` та `const char* password = "34666616"` задає ім'я та пароль для підключення до мережі (рис. 3.6).

```
int lcdColumns = 16;  
int lcdRows = 2;  
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
```

Рисунок 3.7 – Лістинг коду для налаштування LCD дисплея

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Даний фрагмент коду налаштовує LCD-дисплей за допомогою бібліотеки LiquidCrystal_I2C. lcdColumns = 16 та lcdRows задають кількість колонок та рядків на дисплеї. LiquidCrystal_I2C, використовуючи I2C адресу 0x27 і вказані параметри для кількості колонок і рядків. Цей об'єкт lcd використовується для взаємодії з дисплеєм через I2C інтерфейс (рис. 3.7).

```
const int TRIG_PIN_1 = 13;
const int ECHO_PIN_1 = 26;
Ultrasonic ultrasonic_1(TRIG_PIN_1, ECHO_PIN_1);

const int TRIG_PIN_2 = 12;
const int ECHO_PIN_2 = 25;
Ultrasonic ultrasonic_2(TRIG_PIN_2, ECHO_PIN_2);

const int TRIG_PIN_3 = 14;
const int ECHO_PIN_3 = 33;
Ultrasonic ultrasonic_3(TRIG_PIN_3, ECHO_PIN_3);

const int TRIG_PIN_4 = 27;
const int ECHO_PIN_4 = 32;
Ultrasonic ultrasonic_4(TRIG_PIN_4, ECHO_PIN_4);

const int BUTTON_PIN = 4;
bool buttonPressed = false;
```

рис.

Рисунок 3.8 – Лістинг коду

В даному фрагменті коду визначаються до яких пінів на мікроконтролері підключені відповідні контакти кожного датчика. Далі, створюються об'єкти Ultrasonic ultrasonic_X(TRIG_PIN_X, ECHO_PIN_X); для кожного датчика, які використовують відповідні піни. Ці об'єкти дозволяють зручно взаємодіяти з датчиками для вимірювання відстані. Для кнопки const int BUTTON_PIN = 4; визначає, що кнопка підключена до піна 4 мікроконтролера. bool buttonPressed = false; оголошує змінну buttonPressed, яка використовується для зберігання стану кнопки (рис. 3.8).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```
long maxHeight = 8;
long distance;
float v;
float actual_height;
float r = 3.5/2;
float capacity;
int currentSensor = 1;
```

Рисунок – 3.9 – Лістинг оголошення змінних

Опис оголошених змінних:

- maxHeight: визначає максимальну висоту рідини в баках, виміряну в сантиметрах. У цьому випадку, максимальна висота встановлена на 8 см;
- distance: зберігає значення відстані, виміряної ультразвуковим датчиком. Ця відстань використовується для обчислення висоти рідини в циліндрах;
- v: зберігає об'єм рідини в циліндрах, який буде обчислений на основі виміряної висоти рідини;
- actual_height: зберігає фактичну висоту рідини в баках. Ця висота обчислюється як різниця між maxHeight і distance;
- r: визначає радіус циліндра. У цьому випадку, діаметр циліндра встановлений на 3.5 см, тому радіус обчислюється, як половина діаметра ($3.5/2 = 1.75$ см);
- capacity: зберігає об'єм рідини в циліндрах, виражений в літрах. Об'єм буде обчислений на основі значення v;
- currentSensor: визначає поточний активний датчик, який використовується для вимірювання. Ініціалізована значенням 1, що означає, що перший датчик використовується за замовчуванням (рис. 3.9).

```
void setup() {
  lcd.init();
  lcd.backlight();
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  Serial.begin(115200);
}
```

Рисунок 3.10 – Лістинг функції ініціалізації

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Опис функції `void setup()`:

- `lcd.init()`: ініціалізує LCD дисплей. Ця команда запускає LCD і підготовлює його до роботи;
- `lcd.backlight()`: вмикає підсвічування дисплея;
- `pinMode`: налаштовує режим піна `BUTTON_PIN` на вхід з підтягуванням до напруги;
- `Serial.begin(115200)`: ця команда задає швидкість передачі даних по послідовному інтерфейсу в бітах за секунду (рис. 3.10).

```
if(!SPIFFS.begin(true)){  
  Serial.println("Під час монтування SPIFFS сталася помилка");  
  return;  
}  
  
WiFi.begin(ssid, password);  
while (WiFi.status() != WL_CONNECTED) {  
  delay(1000);  
  Serial.println("Підключення до WiFi..");  
}  
  
Serial.println(WiFi.localIP());  
  
Serial.println("Підключено до мережі WiFi");  
  
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){  
  request->send(SPIFFS, "/index.html", "text/html");  
});  
  
server.on("/css/styles.css", HTTP_GET, [](AsyncWebServerRequest *request){  
  request->send(SPIFFS, "/css/styles.css", "text/css");  
});
```

Рисунок 3.11 – Лістинг ініціалізації SPIFFS

Цей фрагмент коду налаштовує мікроконтролер для роботи, як веб-сервер і забезпечує підключення до мережі Wi-Fi. Він також налаштовує обробку HTTP запитів для відправки вмісту файлів з файлової системи SPIFFS, таких як HTML сторінки та CSS стилів, на веб-клієнти, що з'єднуються з мікроконтролером через Wi-Fi (рис. 3.11).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

server.on("/sensor", HTTP_GET, [](AsyncWebServerRequest *request){
    float distance_1 = ultrasonic_1.read();
    float level_1 = maxHeight - distance_1;
    float capacity_1 = 3.14 * r * r * level_1 / 1000;

    float distance_2 = ultrasonic_2.read();
    float level_2 = maxHeight - distance_2;
    float capacity_2 = 3.14 * r * r * level_2 / 1000;

    float distance_3 = ultrasonic_3.read();
    float level_3 = maxHeight - distance_3;
    float capacity_3 = 3.14 * r * r * level_3 / 1000;

    float distance_4 = ultrasonic_4.read();
    float level_4 = maxHeight - distance_4;
    float capacity_4 = 3.14 * r * r * level_4 / 1000;
}

```

Рисунок 3.12 – Лістинг налаштування обробника HTTP GET

Цей фрагмент коду встановлює обробник для HTTP GET запитів на шляху /sensor для веб-сервера (рис. 3.12).

```

String json = "{\"sensor1\":{\"level\":" + String(level_1) + ",\"capacity\":" + String(capacity_1) + "},";
json += "\"sensor2\":{\"level\":" + String(level_2) + ",\"capacity\":" + String(capacity_2) + "},";
json += "\"sensor3\":{\"level\":" + String(level_3) + ",\"capacity\":" + String(capacity_3) + "},";
json += "\"sensor4\":{\"level\":" + String(level_4) + ",\"capacity\":" + String(capacity_4) + "}}";

request->send(200, "application/json", json);

```

Рисунок 3.13 – Лістинг коду, який створює рядок JSON

Цей код створює рядок JSON з даними про рівні рідини та їх об'єм у чотирьох резервуарах у форматі, придатному для відправлення через HTTP (рис. 3.13).

```

void loop() {
    switch (currentSensor) {
        case 1:
            distance = ultrasonic_1.read();
            break;
        case 2:
            distance = ultrasonic_2.read();
            break;
        case 3:
            distance = ultrasonic_3.read();
            break;
        case 4:
            distance = ultrasonic_4.read();
            break;
    }
}

```

Рисунок 3.14 – Лістинг реалізації циклу loop()

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Ця частина коду реалізує головний цикл `loop()`, де використовується конструкція `switch-case` для визначення, який ультразвуковий датчик буде опитаний для вимірювання відстані. Відповідно до значення змінної `currentSensor`, визначається номер датчика, і зчитується відстань за допомогою відповідного об'єкта `Ultrasonic` (рис. 3.14).

```
actual_height = maxHeight - distance;  
v = ((3.141592654*(r*r))*(actual_height));  
capacity = v/1000;
```

Рисунок 3.15 – Лістинг розрахунку даних

Ці розрахунки дозволяють отримувати значення об'єму рідини в резервуарі на основі відстані, яку виміряв ультразвуковий датчик, і параметрів резервуара, таких як максимальна висота і радіус (рис. 3.15).

```
byte buttonState = digitalRead(BUTTON_PIN);  
if (buttonState == LOW && !buttonPressed) {  
    currentSensor++;  
    if (currentSensor > 4) {  
        currentSensor = 1;  
    }  
    buttonPressed = true;  
} else if (buttonState == HIGH) {  
    buttonPressed = false;  
}
```

Рисунок 3.16 – Лістинг циклічного перемикання між датчиками

Даний код дозволяє використовувати кнопку для циклічного перемикання між чотирма ультразвуковими датчиками, що міряють рівень рідини в різних резервуарах. Кожне натискання кнопки змінює значення `currentSensor`, що потім використовується для вибору потрібного датчика в інших частинах програми (рис. 3.16).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		


```

lcd.setCursor(0, 0);
lcd.print("Sen:");
lcd.print(currentSensor);
lcd.setCursor(6, 0);
lcd.print("Dist:");
if (distance < 10)
| lcd.print("0");
if (distance < 100)
| lcd.print("0");
lcd.print(actual_height);
lcd.setCursor(14, 0);
lcd.print("cm");

lcd.setCursor(0, 1);
lcd.print("Capacity: ");
lcd.print(lcdString);
lcd.setCursor(14, 1);
lcd.print(" ");
lcd.setCursor(15, 1);
lcd.print("1");

delay(10);

```

Рисунок 3.17 – Лістинг розрахунку даних

Цей код використовує LCD дисплей для відображення інформації обраного датчика, номер датчика, рівень рідини і об'єм рідини в циліндрі (рис. 3.17).

3.2.2 Реалізація проектних рішень програмного забезпечення веб-сайту

Для початку ми створюємо папку в зручному для нас місці і відкриваємо її за допомогою Visual Studio Code. Далі в нашому проекті необхідно додати основні файли для нашого проекту: `index.html` та `style.css`. Також в папку з проектом завантажуюмо файл з фоном сайту і `reset.css`, який слугує для скидання стилів та ідентичного відображення веб-сторінки в різних браузерах(рис. 3.18).

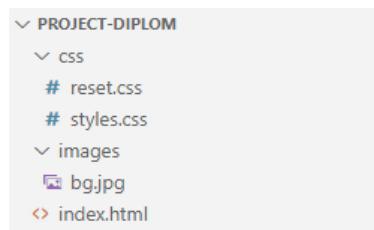


Рисунок 3.18 – Підготовка файлів для веб-сторінки

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Спочатку розберемо файл index.html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>System</title>
  <link rel="preconnect" href="https://fonts.googleapis.com" />
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
  <link
    href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&display=swap"
    rel="stylesheet"
  />
  <link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/modern-normalize/2.0.0/modern-normalize.min.css"
    integrity="sha512-4xo8b1kMVCiXpTaLzQSLSw3KF0VPWhm/TRtuPvc4wG6kUgjH6J03IBuG7JZPkclWmxJ5huwaBp0pnwYE1P/m6wg=="
    crossorigin="anonymous"
    refererpolicy="no-referrer"
  />
  <link rel="stylesheet" href="./css/reset.css" />
  <link rel="stylesheet" href="./css/styles.css" />
</head>
```

Рисунок 3.19 – Лістинг заголовку HTML-файлу

Даний фрагмент HTML-коду містить заголовок `head` документа в якому містяться різні мета-дані, підключаються зовнішні шрифти та CSS-стилі та задається назва сторінки (рис. 3.19).

```
<script>
  function fetchSensorData() {
    fetch('/sensor')
      .then(response => response.json())
      .then(data => {
        document.getElementById("sensor-1-level").innerText = data.sensor1.level + " cm";
        document.getElementById("sensor-1-volume").innerText = data.sensor1.capacity + " l";
        document.getElementById("sensor-2-level").innerText = data.sensor2.level + " cm";
        document.getElementById("sensor-2-volume").innerText = data.sensor2.capacity + " l";
        document.getElementById("sensor-3-level").innerText = data.sensor3.level + " cm";
        document.getElementById("sensor-3-volume").innerText = data.sensor3.capacity + " l";
        document.getElementById("sensor-4-level").innerText = data.sensor4.level + " cm";
        document.getElementById("sensor-4-volume").innerText = data.sensor4.capacity + " l";
      })
      .catch(error => console.error('Error:', error));
  }

  setInterval(fetchSensorData, 1000);
</script>
```

Рисунок 3.20 – Лістинг отримання даних та їх оновлення

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей фрагмент коду відповідає отримання даних про рівень рідини та об'єм, і автоматичне оновлення їх на сайті (рис. 3.20).

```
<body>
  <main>
    <section class="counter">
      <div class="container counter-container">
        <div class="counter-dialog">
          <h1 class="counter-title">
            Системи вимірювання рівня <span>рідини та об'єму в циліндрі</span>
          </h1>
        </div>
      </div>
    </section>
  </main>
</body>
```

Рисунок 3.21 – Лістинг відображення заголовку

В даному фрагменті коду зображено початок тіла HTML-коду `body` з основним вмістом `main`. За допомогою контейнерів з класами представлено заголовок сторінки (рис. 3.21).

```
<div class="counter-component">
  <h2 class="countet-component-title">Датчик 1</h2>
  <ul class="counter-list">
    <li class="counter-element red">
      <h2 class="counter-element-title">Рівень рідини:</h2>
      <span id="sensor-1-level" class="counter-element-value">0</span>
    </li>
    <li class="counter-element red">
      <h2 class="counter-element-title">Об'єм:</h2>
      <span id="sensor-1-volume" class="counter-element-value">0</span>
    </li>
  </ul>
</div>
```

Рисунок 3.22 – Лістинг секції для відображення даних першого датчика

В цьому фрагменті коду створюється блок, що відображає номер датчика та інформацію про рівень рідини і об'єм та оновляє ці значення (рис. 3.22). Таким чином представлені, ще три блоки з іншими датчиками.

Далі переходимо до заданих CSS-стилів.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

:root {
  --red-color: #bf2604;
  --red-gradient: linear-gradient(
    to right bottom,
    #bf2604,
    #6f0012,
    #4d0816,
    #2a0c12,
    #000000
  );
  --orange-gradient: linear-gradient(
    to right bottom,
    #9d1f00,
    #8a1c00,
    #7a1800,
    #571100,
    #2b0900
  );
  --light-orange-gradient: linear-gradient(
    to right bottom,
    #ff7700,
    #ba5700,
    #ab5000,
    #6b3200,
    #3d1c00
  );
  --yellow-gradient: linear-gradient(
    to right bottom,
    #f2a950,
    #cf9349,
    #9b6c34,
    #744e20,
    #583d1d
  );
}

```

Рисунок 3.23 – Лістинг змінних для градієнта

Тут зображено змінні, які дозволяють легко змінювати кольорову палітру і градієнти на всій сторінці, що дозволяє легко налаштувати стилі (рис. 3.23).

```

body {
  font-family: "Montserrat", sans-serif;
  font-weight: 600;
  font-size: 16px;
  line-height: 1;
  color: white;
  background-image: url(../images/bg.jpg);
  background-repeat: no-repeat;
  background-size: cover;
}

```

Рисунок 3.24 – Лістинг налаштувань вигляду всієї сторінки

						КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

Цей набір стилів допомагає налаштувати вигляд нашої сторінки для зручного читання, використовуючи шрифт Montserrat, текст білого кольору та заднє фонове зображення (рис. 3.24).

```
.container {  
  max-width: 1170px;  
  padding: 25px;  
  margin: 0 auto;  
}  
  
.counter {  
  position: fixed;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  background-color: #000000ac;  
}  
  
.counter-container {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

Рисунок 3.25 – Лістинг параметрів контейнера

На рис. 2.25 зображено лістинг таких класів:

- `container`: – задає параметри основного контейнера;
- `counter`: – в цьому класі фіксується основний блок та задаються параметри для затемнення фонового зображення сайту;
- `counter-container`: – розміщує контейнер відносно його батьківського блоку та центрує його.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

.counter-title {
  font-weight: 700;
  font-size: 25px;
  text-align: center;
  text-transform: uppercase;
  letter-spacing: 0.2em;
  line-height: 1.2;
  margin-bottom: 50px;
  text-shadow: 0px 0px 10px white;

  & span {
    color: var(--red-color);
    text-shadow: 0px 0px 10px var(--red-color);
  }
}

```

Рисунок 3.26 – Лістинг вигляду заголовка

Ці стилі використовуються для створення візуального заголовка з підсвічуванням елементів (рис. 3.26).

```

.countet-component-title {
  font-weight: 500;
  font-size: 20px;
  text-align: center;
  text-transform: uppercase;
  margin-bottom: 20px;
}

.counter-list {
  display: flex;
  align-items: center;
  justify-content: center;
  gap: 80px;
}

```

Рисунок 3.27 – Лістинг параметрів для заголовків та елементів

Дані стилі налаштовують розташування заголовків та елементів для коректного відображення на сайті (рис. 3.27).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

.counter-element {
  border-radius: 10px;
  animation: shadow 3s infinite;
  padding: 15px;
  text-align: center;

  &.red {
    background-image: var(--red-gradient);
  }
  &.orange {
    background-image: var(--orange-gradient);
  }
  &.lightOrange {
    background-image: var(--light-orange-gradient);
  }
  &.yellow {
    background-image: var(--yellow-gradient);
  }
}

```

Рисунок 3.28 – Лістинг параметрів блоків з даними

Представлений клас задає вигляд блокам з даними про рівень рідини та об'єм. Також додає анімацію з динамічним ефектом (рис. 3.28).

```

.counter-element-title {
  margin-bottom: 10px;
  font-weight: 300;
  font-size: 18px;
  color: □white;
}

.counter-element-value {
  font-weight: 700;
  font-size: 30px;
  color: □white;
}

```

Рисунок 3.29 – Лістинг параметрів елементів в середині блоків

Вище зображені стилі задають зовнішній вигляд для елементів всередині блоків (рис. 3.29).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Тестування системи

На даному етапі необхідно протестувати систему. Для цього підключаємо всі елементи між собою та розміщуємо датчики над резервуарами рис.3.30.

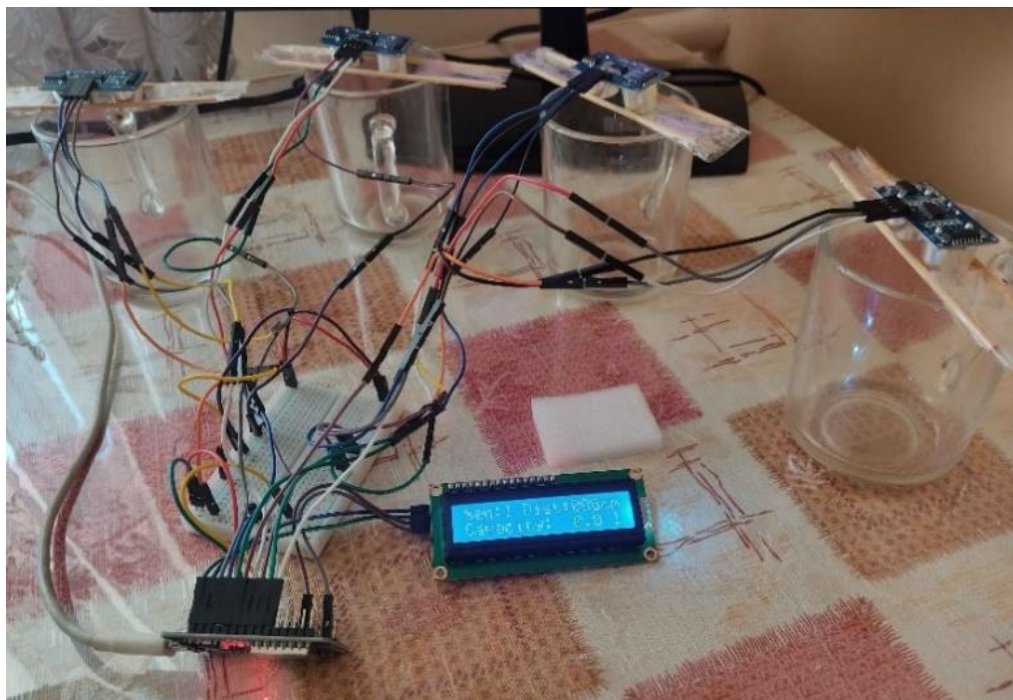


Рисунок 3.30 – Прототип системи вимірювання рівня рідини та об’єму

При натисканні на кнопку можемо побачити, що значення на дисплеї змінюються (рис. 3.31).



Рисунок 3.31 – Дані з різних датчиків

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Після написання коду для мікроконтролера та веб-сайту. Відкриваємо наш проект в програмному середовищі ArduinoIDE і завантажуюмо код в мікроконтролер. Після цього відкриваємо «Монітор порту» і бачимо IP-адресу завдяки якій ми можемо потрапити на наш сайт (рис. 3.32).

```
16:59:59.788 -> Підключення до WiFi..  
17:00:00.812 -> Підключення до WiFi..  
17:00:01.814 -> Підключення до WiFi..  
17:00:02.787 -> Підключення до WiFi..  
17:00:02.827 -> 192.168.1.120  
17:00:02.827 -> Підключено до мережі WiFi
```

Рисунок 3.32 – Отримання IP-адреси веб-сайту

Переходимо на сайт та бачимо результат компіляції коду і виведення даних (рис. 3.33).



Рисунок 3.33 – Результат роботи системи

Після проведення тестування можемо переконатися, що система функціонує коректно і відповідає попередньо заявленим вимогам.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Долікарська допомога при задусі, утопленні

Надання долікарської допомоги при задусі та утопленні є критично важливим для збереження життя постраждалого до прибуття медичних служб. Ці ситуації потребують швидкої реакції та знань основних методів реанімації. Ефективність долікарської допомоги визначається швидкістю та правильністю дій, тому важливо розуміти механізми виникнення задушення та утоплення, а також відповідні алгоритми надання першої допомоги.

Задушення, або асфіксія, виникає через обмеження потоку повітря в дихальні шляхи. Це може статися через обструкцію верхніх дихальних шляхів сторонніми предметами, компресію ший, алергічні реакції з набряком горла, або патологічні стани, що призводять до зменшення прохідності дихальних шляхів.

Утоплення, в свою чергу, є процесом утоплення у воді або іншій рідині, що призводить до гіпоксії (недостатнього постачання кисню до тканин). Утоплення може бути результатом випадкового потрапляння води в дихальні шляхи, тривалого перебування під водою або впливу інших факторів, які унеможливають дихання у водному середовищі.

При підозрі удушення важливо швидко оцінити стан постраждалого. Перш за все, необхідно перевірити наявність свідомості та здатність дихати. Якщо постраждалий без свідомості, негайно перевірте прохідність дихальних шляхів. Важливо також визначити, чи є обструкція дихальних шляхів, через наявність сторонніх предметів. Ці кроки дозволяють швидко зорієнтуватися у стані постраждалого та обрати правильний алгоритм дій для надання першої допомоги.

					КС КРБ 123.133.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>				
<i>Розроб.</i>		<i>Стецько В.Р.</i>			<i>РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Стадник Н.Б.</i>						
<i>Консульт.</i>		<i>Пилипець М.І.</i>						
<i>Н. контр.</i>		<i>Тиш Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						
						<i>ТНТУ, каф. КС, гр. СІ-42</i>		

Перед початком надання допомоги при задусі на підприємстві, необхідно переконатися у безпеці оточуючих умов. Важливо забезпечити, щоб місце було безпечним для рятувальника і постраждалого.

Наступним кроком є визначення ступеня обструкції дихальних шляхів. Якщо обструкція часткова, і постраждалий може кашляти, дихати або говорити, заохочуйте його до активного кашлю, оскільки це найефективніший спосіб видалення обструкції. У випадку повної обструкції, коли постраждалий не може говорити, дихати або кашляти, потрібно негайно приступити до активних дій.

Метод Геймліха ефективний для дорослих та дітей старше одного року. Для виконання цього методу станьте позаду постраждалого, обійміть його руками, розташувавши кулаки на животі між пупком і ребровою дугою. Виконуйте різкі поштовхи вгору і всередину, намагаючись виштовхнути обструкцію [21].

У випадку втрати свідомості постраждалого, необхідно обережно покласти його на спину і розпочати серцево-легеневу реанімацію (СЛР). Компресії грудної клітки можуть допомогти видалити обструкцію. Після кожних 30 компресій потрібно обережно відкрити рот постраждалого і перевірити наявність обструкції, повторюючи процедуру до відновлення дихання або прибуття медичних служб.

У випадку утоплення на підприємстві, першочерговим завданням є швидке витягнення постраждалого з води, дотримуючись правил безпеки, щоб уникнути ризику для рятувальника. Після витягнення необхідно оцінити стан постраждалого, перевіривши наявність свідомості, дихання та циркуляції крові. Якщо постраждалий не дихає, негайно розпочніть серцево-легеневу реанімацію (СЛР). У випадку, якщо постраждалий дихає, його потрібно розмістити у стабільному боковому положенні для підтримання прохідності дихальних шляхів і запобігання аспірації.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Перед наданням допомоги необхідно переконатися у безпеці оточуючих умов і забезпечити безпеку як рятувальника, так і постраждалого при витягуванні з води. Після цього проводиться початкова оцінка стану постраждалого: якщо він не дихає, негайно починайте серцево-легеневу реанімацію. Для цього необхідно виконувати 30 компресій грудної клітки на глибину приблизно 5-6 см з частотою 100-120 компресій на хвилину, після чого зробіть 2 рятувальні вдихи, забезпечивши герметичність навколо рота постраждалого. Продовжуйте реанімацію до прибуття медичних служб або відновлення самостійного дихання [22].

При відновленні дихання необхідно перевірити наявність ознак гіпотермії. Постраждалого потрібно вкривати теплим одягом або ковдрою, за можливості виміряти температуру тіла та вжити додаткових заходів для підтримання нормальної температури. Ці дії спрямовані на збереження життя постраждалого до прибуття професійних медичних служб і значно підвищують його шанси на виживання.

4.2 Заходи безпеки для захисту організму людини від шкідливого впливу ультразвуку

Ультразвук, що використовується в різних галузях медицини, промисловості та технологій, має як корисні, так і шкідливі ефекти на організм людини. Вплив ультразвукових хвиль може бути шкідливим при перевищенні допустимих рівнів інтенсивності та тривалості впливу, тому важливо розробити та впровадити ефективні заходи для зниження ризиків.

Перш за все, необхідно забезпечити належний моніторинг і контроль рівнів ультразвуку в робочих та житлових приміщеннях. Використання спеціалізованих ультразвукових дозиметрів дозволяє вчасно виявляти перевищення допустимих норм і вживати відповідних заходів. Крім того, важливо розробити і впровадити нормативні документи, що регулюють

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

гранично допустимі рівні ультразвукового випромінювання для різних середовищ та умов праці.

Для захисту працівників, які постійно перебувають в умовах підвищеного ультразвукового фону, необхідно застосовувати засоби індивідуального захисту (ЗІЗ). Це можуть бути спеціальні звукоізоляційні навушники або шоломи, які знижують рівень ультразвуку, що досягає органів слуху. Також використовуйте захисні костюми, які зменшують вплив ультразвукових хвиль на інші частини тіла.

Важливим аспектом є технічні заходи щодо зниження інтенсивності ультразвукового випромінювання на робочих місцях. Це може включати в себе встановлення захисних бар'єрів, екранів і обшивок, які поглинають або відбивають ультразвукові хвилі. Крім того, необхідно забезпечити регулярне технічне обслуговування і налаштування ультразвукових пристроїв, щоб уникнути їхнього ненавмисного випромінювання на підвищених рівнях.

Освітня та інформаційна робота серед працівників також є ключовою для зниження шкідливого впливу ультразвуку. Необхідно проводити навчальні заходи, семінари та тренінги, де роз'яснюватимуться основні ризики та методи захисту від ультразвукових хвиль. Інформування про можливі наслідки тривалого впливу ультразвуку та про заходи безпеки сприятиме підвищенню обізнаності та відповідальності працівників.

Крім індивідуального та технічного захисту, розглянуто організаційні заходи. Це може включати обмеження часу перебування працівників у зонах підвищеного ультразвукового випромінювання, ротацію працівників, щоб зменшити тривалість впливу, а також організацію регулярних медичних оглядів для раннього виявлення можливих порушень здоров'я, викликаних ультразвуком.

Враховано можливість використання альтернативних технологій, які мають менший вплив на організм людини. У випадках, коли це можливо, варто

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

замінювати ультразвукові методи на інші діагностичні чи технологічні підходи, які є безпечнішими для здоров'я.

Наукові дослідження у сфері впливу ультразвуку на організм людини повинні продовжуватися і розширюватися. Це дозволить глибше розуміти механізми впливу ультразвукових хвиль, розробляти нові методи захисту та оптимізувати існуючі заходи безпеки. Співпраця між науковцями, лікарями та технологами може призвести до створення більш ефективних стратегій боротьби з шкідливою дією ультразвуку [23].

Таким чином, заходи щодо боротьби з шкідливою дією ультразвуку на організм людини повинні бути комплексними і включати моніторинг, нормативне регулювання, використання засобів індивідуального та колективного захисту, освітні програми, організаційні зміни та наукові дослідження. Лише системний підхід дозволить ефективно мінімізувати ризики та забезпечити безпеку для здоров'я людей, що працюють в умовах підвищеного ультразвукового фону.

4.3 Пожежна профілактика на підприємстві

Пожежна профілактика є важливою складовою забезпечення безпеки на підприємствах, де концентрація легкозаймистих матеріалів та обладнання створює високий ризик виникнення пожежі. Основна мета пожежної профілактики полягає в попередженні виникнення пожеж, мінімізації їх наслідків та забезпеченні безпеки працівників.

Організаційні заходи є першочерговими в системі пожежної профілактики. До них належать розробка та впровадження локальних нормативних актів, інструкцій з пожежної безпеки, які регламентують порядок дій працівників у разі виникнення пожежі. Ці документи розробляють відповідно до діючих державних стандартів та законодавства.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

На підприємстві призначають відповідальні особи за пожежну безпеку, які здійснюють постійний контроль за дотриманням правил пожежної безпеки, проводять навчання та інструктажі з працівниками. Регулярні інструктажі та тренування з евакуації допомагають працівникам швидко та правильно діяти в разі пожежі.

Технічні заходи включають установку та обслуговування протипожежного обладнання, такого як автоматичні системи пожежогасіння, пожежні сигналізації, системи димовидалення та вентиляції. Важливо, щоб усі системи знаходились у справному стані та регулярно перевірялися. Також необхідно забезпечити наявність достатньої кількості вогнегасників у доступних місцях, відповідно до нормативних вимог[24].

Перевірка стану електропроводки, справність захисних пристроїв та контроль за експлуатацією електрообладнання є важливими елементами попередження пожеж. Регулярне технічне обслуговування та своєчасна заміна зношених або пошкоджених елементів електросистем дозволяє уникнути коротких замикань та перегріву проводки.

Архітектурно-будівельні заходи включають створення умов, що перешкоджають поширенню пожежі. Це досягається шляхом використання негорючих або важко горючих матеріалів при будівництві, а також поділом приміщень на протипожежні зони. Встановлення протипожежних перегородок, дверей та вікон, що перешкоджають поширенню вогню та диму, є важливим елементом пожежної профілактики.

Підтримка чистоти та порядку в цеху також має значення для пожежної безпеки. Скупчення горючих відходів, пилу, мастил та інших легкозаймистих матеріалів створює додаткові ризики. Регулярне прибирання робочих місць, видалення відходів та своєчасне очищення обладнання від пилу та бруду сприяють зниженню ймовірності виникнення пожежі.

На підприємстві працівники повинні мати необхідні знання та навички для дій у разі виникнення пожежі. Регулярне проведення навчань з пожежної

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

безпеки, тренувань з евакуації та практичних занять з використання первинних засобів пожежогасіння допомагають забезпечити готовність персоналу до надзвичайних ситуацій.

Ефективна пожежна профілактика неможлива без тісної взаємодії з місцевими підрозділами пожежної охорони. Підприємство забезпечується засобами зв'язку для оперативного виклику пожежних підрозділів. Важливо також проводити спільні тренування з пожежними службами, що дозволяє відпрацювати координацію дій у разі пожежі.

Постійний контроль за дотриманням правил пожежної безпеки є важливою складовою профілактичних заходів. Проводити регулярні аудити та перевірки допомагає виявляти та усувати потенційні порушення та недоліки в системі пожежної безпеки. Використання чек-листів та планів перевірок сприяє систематичному підходу до контролю пожежної безпеки.

Таким чином, комплексний підхід до пожежної профілактики, що включає організаційні, технічні, архітектурно-будівельні, санітарно-гігієнічні заходи та навчання працівників, є необхідною умовою для забезпечення безпеки на промислових підприємствах. Це дозволяє мінімізувати ризик виникнення пожеж, зберегти життя та здоров'я працівників, а також зменшити матеріальні збитки.

Важливою складовою пожежної профілактики є регулярне проведення практичних занять з пожежної безпеки. Такі заняття включають відпрацювання евакуаційних маршрутів, використання первинних засобів пожежогасіння, та взаємодію між працівниками у випадку надзвичайної ситуації. Це допомагає не лише закріпити теоретичні знання, але й сформувати у працівників навички швидкого та правильного реагування на пожежу, що значно підвищує рівень загальної безпеки на підприємстві.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була розроблена комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» і були зроблені такі висновки:

- розглянуто та опрацьовано ймовірні методи та засоби вимірювання рівня рідини та об'єму в різних ємкостях;
- розроблено структурну схему для комп'ютеризованої системи;
- спроектовано електричну принципову схему фізичної складової системи;
- було складено електричне коло для системи вимірювання рівня рідини та об'єму в циліндрі на базі мікроконтролера ESP32 devkit v1;
- розроблено програмне забезпечення для мікроконтролера та веб-сайту;
- проведено тестування, яке показало, що система є точною та ефективною для відслідковування вимірювань;
- створено веб-сайту, як систему моніторингу вимірювань, яка дозволяє легко та зручно відслідковувати дані про рівень рідини та об'єм в циліндрі на підприємстві ТОВ «Пром-Дорекс».

Проаналізувавши проведені дослідження рекомендується вдосконалити систему, додавши функцію сповіщення про критичні значення вимірювання, які включатимуть відправку сповіщень на електронну пошту користувачів.

У майбутньому це сприятиме розвитку системи вимірювання рівня рідини та об'єму в циліндрі для кращого забезпечення зв'язку з користувачами.

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Осухівська Г.М., Тиш Є.В., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.
2. Ларіоник Р., Луцик Н., Паламар А. Система для моніторингу якості атмосферного повітря на базі IoT: ТНТУ, 2021. с. 119.
3. Купратий І. Г., Паламар А. М. Комп'ютерна система для дистанційного моніторингу стану здоров'я пацієнтів: ТНТУ, 2022. с. 142.
4. Кунинець Д. В., Лещишин Ю. З. Застосунок для моніторингу даних розумного будинку: ФОП Паляниця В. А., 2021. Том I., с. 94.
5. Бігуняк А., Жаровський Р. О. Особливості протоколів маршрутизації в комп'ютерних мережах: ТНТУ, 2012 с. 40.
6. Radar sensor VEGAPULS 61 URL: <https://www.vega.com/en/products/product-catalog/level/radar/vegapuls-61/> (дата звернення: 29.04.2024).
7. Датчики для контролю рівня речовин. URL: academy.arbu.edu.ua/e-books/009/TA/rozd2_3.htm#:~:text=Технічні%20засоби%2C%20які%20призначені%20для,інші%2C%20що%20одержали%20незначне%20поширення (дата звернення: 29.04.2024).
8. Структурна схема URL: https://uk.wikipedia.org/wiki/Структурна_схема (дата звернення: 02.05.2024).
9. ESP32 Pinout Reference URL: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/> (дата звернення: 07.05.2024).
10. DOIT ESP32 DEV KIT v1, high resolution pinout and specs URL: <https://mischianti.org/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/> (дата звернення: 08.05.2024).
11. Ультразвуковий датчик відстані HC-SR04 URL: makerobot.com.ua/hc-sr04 (дата звернення: 10.05.2024).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

12. LCD дисплей 1602 URL: <https://ardushop.in.ua/arduino/lcd-display-1602-hd44780-with-blue-backlight> (дата звернення: 11.05.2024).

13. Адаптер I2C для LCD дисплеїв на PCF8574 URL: <https://www.minitech.com.ua/pcf8574-i2c-adapter> (дата звернення: 12.05.2024).

14. Arduino IDE URL: <https://www.arduino.cc/en/software> (дата звернення: 14.05.2024).

15. Visual Studio Code URL: <https://code.visualstudio.com> (дата звернення: 15.05.2024).

16. Arduino C++ URL: <https://docs.arduino.cc/arduino-cloud/guides/arduino-c/> (дата звернення: 17.05.2024).

17. ESP32 Web Server: Display Sensor Readings in Gauges URL: <https://randomnerdtutorials.com/esp32-web-server-gauges> (дата звернення: 18.05.2024).

18. ESP32 Useful Wi-Fi Library Functions (Arduino IDE) URL: <https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/> (дата звернення: 20.05.2024).

19. How to Use I2C LCD with ESP32 on Arduino IDE URL: <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/> (дата звернення: 22.05.2024).

20. ESP32 with HC-SR04 Ultrasonic Sensor with Arduino IDE URL: <https://randomnerdtutorials.com/esp32-hc-sr04-ultrasonic-arduino/> (дата звернення: 22.05.2024).

21. Метод Геймліха URL: <https://vseosvita.ua/lesson/pryiom-heimlikha186541.html> (дата звернення: 29.05.2024).

22. Перша допомога при утопленні URL: <https://paramedic.ua/ua/pervaya-pomoshh-pri-utoplenii> (дата звернення: 29.05.2024).

23. Література: 43 ДСН 3.3.6.037 – 99 „Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку” (дата звернення: 30.05.2024).

24. Пожежна профілактика URL: <https://nais-n.com.ua/news/pozhezhna-profilaktyka-ta-yiyi-zavdannya/> (дата звернення: 30.05.2024).

					КС КРБ 123.133.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

«ЗАТВЕРДЖУЮ»

Завідувач кафедри КС

Осухівська Г.М.

“ ___ ” _____ 2024р.

КОМП'ЮТЕРИЗОВАНА СИСТЕМА ВИМІРЮВАННЯ РІВНЯ РІДИНИ ТА
ОБ'ЄМУ В ЦИЛІНДРІ ДЛЯ ТОВ «ПРОМ-ДОРЕКС»

ТЕХНІЧНЕ ЗАВДАННЯ

на 7 листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

к.т.н. Стадник Н. Б.

“ ___ ” _____ 2024 р.

«ВИКОНАВЕЦЬ»

Студент групи СІ-42

Стецько В. Р.

“ ___ ” _____ 2024 р.

Тернопіль 2024

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи бакалавра: «Комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс»»

Умовне позначення дипломного проекту: КС КРБ 123 133.00.00.

1.2 Виконавець

Студент групи СІ-42, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Стецько Владислав Русланович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи бакалавра є наказ по університету № 4/7-408 від «24» квітня 2024 року

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи бакалавра – 24.04.2024р.

Плановий термін завершення виконання кваліфікаційної роботи бакалавра

– 26.06.2024р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Оформлення технічної документації до кваліфікаційної роботи бакалавра здійснюється згідно діючих вимог вітчизняних та міжнародних стандартів. Технічна документація до кваліфікаційної роботи бакалавра включає в себе текст пояснювальної записки та креслення, які максимально інформативно та стисло відображають основні результати розробки комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс». Основними регламентними документами при оформленні та пред'явленні результатів проектування є групи діючих стандартів ДСТУ, ГОСТ, ISO та ЄСКД, ЕСПД. Пред'явлення результатів кваліфікаційної роботи бакалавра відбувається шляхом захисту роботи на відповідному засіданні ДЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Система призначена для реалізації системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс».

2.2 Мета створення системи

Метою створення комп'ютеризованої системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» є розробка інноваційного рішення, яке дозволить користувачам зручно і точно взяти рівень рідини та об'єму в циліндрах. Це дозволить збільшити продуктивність за рахунок економії часу та швидко дізнатись інформацію з вимірювальних датчиків та зручно і швидко

переглянути її на веб-сайті.

2.3 Характеристика об'єкту

Система проектується для отримання вимірювань рівня рідини та об'єму в циліндрі, які будуть відображатись на дисплеї та на веб-сайті, і включає в себе:

- розробку схеми електричної принципової;
- розробку алгоритму роботи та програмного забезпечення для мікроконтролера.
- розробку алгоритму роботи та програмного забезпечення для веб-сайту

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» повинна забезпечити:

1. Мінімальну затримку при передачі;
2. Точність передачі даних;
3. Точність вимірювань;
4. Зручність в експлуатації.

3.1.1 Вимоги до структури та функціонування системи

Структура системи вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» включає в себе:

- мікроконтролер, який отримує і обробляє дані вимірювань;
- датчики вимірювання відстані;
- екран;

В загальному випадку, структура системи повинна реалізовувати функції

отримання та обробки даних і їх виведення на екран та веб-сайт. Основні функціональні вимоги характеризуються наступними критеріями:

- функціональність;
- точність;
- ефективність;
- надійність;
- зручність

3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Обмін даними між контролером та приймачем повинен здійснюватися з використанням бездротових технологій передачі інформації.

3.1.3 Вимоги до надійності системи

Система повинна бути захищена від фізичних чи механічних пошкоджень на рівні апаратного та програмного забезпечення. Надійність системи повинна забезпечувати відновлюваність функціонування у випадку збою апаратного чи програмного забезпечення.

3.1.7 Вимоги до апаратного забезпечення

Вимоги до елементної бази розробки:

- режими роботи і умови експлуатації вибраних елементів повинні відповідати вказаним в ТЗ;
- елементна база по можливості має бути широкоживаною, доступною і дешевою. Необхідно також враховувати можливість заміни вибраних елементів на аналогічні (вітчизняні чи імпортного виробництва).

Вимоги до мікроконтролера:

- мікроконтролер має підтримувати RISC архітектуру команд;
- мікроконтролер повинен містити потрібну кількість керованих портів

введення /виведення.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ.

Комплект конструкторської документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
 1. структурна схема;
 2. схема електрична принципова;
 3. блок-схема алгоритму програми.
 4. блок-схема алгоритму веб-сайту.

*Примітка: В комплект конструкторської документації можуть вноситися зміни та доповнення в процесі розробки.

5 Техніко-економічні показники

Собівартість розробки системи повинна становити не більше 2000 грн.

Термін експлуатації системи повинен бути не менший 10 років.

*Примітка: собівартість системи може змінюватись під час розрахунку в процесі розробки.

6 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання КРБ

№ етапу	Назва етапу виконання КРБ	Термін виконання
1	Затвердження технічного завдання	24.04-27.04.2024
2	Аналіз технічного завдання	28.04-01.05.2024
3	Розробка структурної схеми	02.05-06.05.2024
4	Вибір елементної бази	07.05-13.05.2024
5	Вибір програмного забезпечення	14.05-16.05.2024
6	Реалізація проектної складової	17.05-27.05.2024
7	Опрацювання розділу «Безпека життєдіяльності, основи охорони праці»	28.05-30.05.2024
8	Оформлення пояснювальної записки кваліфікаційної роботи	31.05-06.06.2024
9	Оформлення графічної частини	18.06.2024
10	Підготовка до захисту	

7 Додаткові умови виконання кваліфікаційної роботи бакалавра

Під час виконання кваліфікаційної роботи в дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б
Перелік елементів

<i>Поз. Позначення</i>	Найменування	Кіл.	Примітка
	Плата		
U1	ESP32 DEVKIT v1	1	
	Сенсори		
U2, U3, U4, U5	HC-SR04	4	
	Модуль		
U6	PCF8574T	1	
U7	LCD1602	1	
S1	Button	1	

						КС КРБ 123 133.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Комп'ютеризована система вимірювання рівня рідини та об'єму в циліндрі для ТОВ «Пром-Дорекс» Перелік елементів	Лім.	Арк.	Акрушіє
Розроб.	Стецько В.Р.							1
Перевір.	Стадник Н.Б.					ТНТУ, каф. КС, гр. СІ-42		
Реценз.								
Н. контр.	Тиш Є.В.							
Затверд.								

Додаток В

Лістинг коду мікроконтролера

```
#include <LiquidCrystal_I2C.h>
#include <Ultrasonic.h>
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <SPIFFS.h>

const char* ssid = "WASD";
const char* password = "34666616";
AsyncWebServer server(80);

int lcdColumns = 16;
int lcdRows = 2;
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);

const int TRIG_PIN_1 = 13;
const int ECHO_PIN_1 = 26;
Ultrasonic ultrasonic_1(TRIG_PIN_1, ECHO_PIN_1);
const int TRIG_PIN_2 = 12;
const int ECHO_PIN_2 = 25;
Ultrasonic ultrasonic_2(TRIG_PIN_2, ECHO_PIN_2);
const int TRIG_PIN_3 = 14;
const int ECHO_PIN_3 = 33;
Ultrasonic ultrasonic_3(TRIG_PIN_3, ECHO_PIN_3);
const int TRIG_PIN_4 = 27;
const int ECHO_PIN_4 = 32;
Ultrasonic ultrasonic_4(TRIG_PIN_4, ECHO_PIN_4);
const int BUTTON_PIN = 4;
bool buttonPressed = false;

long maxHeight = 8;
long distance;
float v;
float actual_height;
float r = 3.5/2;
float capacity;
int currentSensor = 1;

void setup() {
  lcd.init();
  lcd.backlight();
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  Serial.begin(115200);

  if(!SPIFFS.begin(true)){+
    Serial.println("Під час монтування SPIFFS сталася помилка");
    return;
  }
}
```

```

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Підключення до WiFi..");
}

Serial.println(WiFi.localIP());

Serial.println("Підключено до мережі WiFi");

server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send(SPIFFS, "/index.html", "text/html");
});

server.on("/css/styles.css", HTTP_GET, [] (AsyncWebServerRequest
*request) {
    request->send(SPIFFS, "/css/styles.css", "text/css");
});

server.on("/sensor", HTTP_GET, [] (AsyncWebServerRequest
*request) {
    float distance_1 = ultrasonic_1.read();
    float level_1 = maxHeight - distance_1;
    float capacity_1 = 3.14 * r * r * level_1 / 1000;
    float distance_2 = ultrasonic_2.read();
    float level_2 = maxHeight - distance_2;
    float capacity_2 = 3.14 * r * r * level_2 / 1000;
    float distance_3 = ultrasonic_3.read();
    float level_3 = maxHeight - distance_3;
    float capacity_3 = 3.14 * r * r * level_3 / 1000;
    float distance_4 = ultrasonic_4.read();
    float level_4 = maxHeight - distance_4;
    float capacity_4 = 3.14 * r * r * level_4 / 1000;

    String json = "{\"sensor1\":{\"level\":\"" + String(level_1) +
",\"capacity\":\"" + String(capacity_1) + "\"},\"";
    json += "\"sensor2\":{\"level\":\"" + String(level_2) +
",\"capacity\":\"" + String(capacity_2) + "\"},\"";
    json += "\"sensor3\":{\"level\":\"" + String(level_3) +
",\"capacity\":\"" + String(capacity_3) + "\"},\"";
    json += "\"sensor4\":{\"level\":\"" + String(level_4) +
",\"capacity\":\"" + String(capacity_4) + "\"}}";

    request->send(200, "application/json", json);
});
server.begin();
}

void loop() {
    switch (currentSensor) {

```

```

    case 1:
        distance = ultrasonic_1.read();
        break;
    case 2:
        distance = ultrasonic_2.read();
        break;
    case 3:
        distance = ultrasonic_3.read();
        break;
    case 4:
        distance = ultrasonic_4.read();
        break;
}
byte buttonState = digitalRead(BUTTON_PIN);
if (buttonState == LOW && !buttonPressed) {
    currentSensor++;
    if (currentSensor > 4) {
        currentSensor = 1;
    }
    buttonPressed = true;
} else if (buttonState == HIGH) {
    buttonPressed = false;
}

actual_height = maxHeight - distance;
v = ((3.141592654*(r*r))*(actual_height));
capacity = v/1000;
char lcdString[6];
dtostrf(capacity, 4, 1, lcdString);

lcd.setCursor(0, 0);
lcd.print("Sen:");
lcd.print(currentSensor);
lcd.setCursor(6, 0);
lcd.print("Dist:");
if (distance < 10)
    lcd.print("0");
if (distance < 100)
    lcd.print("0");
lcd.print(actual_height);
lcd.setCursor(14, 0);
lcd.print("cm");
lcd.setCursor(0, 1);
lcd.print("Capacity: ");
lcd.print(lcdString);
lcd.setCursor(14, 1);
lcd.print(" ");
lcd.setCursor(15, 1);
lcd.print("l");
delay(10);
}

```


Додаток Ж

Лістинг код веб-сторінки

Лістинг коду HTML-сторінки:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>System</title>
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin />
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wgh
t@0,100..900;1,100..900&display=swap"
rel="stylesheet"
/>
    <link
rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/modern-
normalize/2.0.0/modern-normalize.min.css"
integrity="sha512-
4xo8blKMVCiXpTaLzQSLSw3KFOVPWhm/TRtuPVc4WG6kUgjH6J03IBuG7JZPkcWMxJ
5huwaBpOpnWYElP/m6wg=="
crossorigin="anonymous"
referrerpolicy="no-referrer"
/>
    <link rel="stylesheet" href="./css/reset.css" />
    <link rel="stylesheet" href="./css/styles.css" />
    <script>
      function fetchSensorData() {
        fetch('/sensor')
          .then(response => response.json())
          .then(data => {
            document.getElementById("sensor-1-level").innerText =
data.sensor1.level + " cm";
            document.getElementById("sensor-1-volume").innerText =
data.sensor1.capacity + " l";
            document.getElementById("sensor-2-level").innerText =
data.sensor2.level + " cm";
            document.getElementById("sensor-2-volume").innerText =
data.sensor2.capacity + " l";
            document.getElementById("sensor-3-level").innerText =
data.sensor3.level + " cm";
            document.getElementById("sensor-3-volume").innerText =
data.sensor3.capacity + " l";
```

```

        document.getElementById("sensor-4-level").innerText =
data.sensor4.level + " cm";
        document.getElementById("sensor-4-volume").innerText =
data.sensor4.capacity + " l";
    })
    .catch(error => console.error('Error:', error));
}

    setInterval(fetchSensorData, 1000);
</script>
</head>
<body>
    <main>
        <section class="counter">
            <div class="container counter-container">
                <div class="counter-dialog">
                    <h1 class="counter-title">
                        Системи вимірювання рівня <span>рідини та об'єму в
циліндрі</span>
                    </h1>
                    <div class="counter-component">
                        <h2 class="countet-component-title">Датчик 1</h2>
                        <ul class="counter-list">
                            <li class="counter-element red">
                                <h2
                                    class="counter-element-title">Рівень
рідини:</h2>
                                <span id="sensor-1-level" class="counter-element-
value">0</span>
                            </li>
                            <li class="counter-element red">
                                <h2 class="counter-element-title">Об'єм:</h2>
                                <span
                                    id="sensor-1-volume"
                                    class="counter-
element-value">0</span>
                            </li>
                        </ul>
                    </div>
                    <div class="counter-component">
                        <h2 class="countet-component-title">Датчик 2</h2>
                        <ul class="counter-list">
                            <li class="counter-element orange">
                                <h2
                                    class="counter-element-title">Рівень
рідини:</h2>
                                <span id="sensor-2-level" class="counter-element-
value">0</span>
                            </li>
                            <li class="counter-element orange">
                                <h2 class="counter-element-title">Об'єм:</h2>
                                <span
                                    id="sensor-2-volume"
                                    class="counter-
element-value">0</span>
                            </li>
                        </ul>
                    </div>
                </div>
            </div>
        </section>
    </main>
</body>
</html>

```

```

    </div>
    <div class="counter-component">
      <h2 class="countet-component-title">Датчик 3</h2>
      <ul class="counter-list">
        <li class="counter-element lightOrange">
          <h2
            class="counter-element-title">Рівень
піддини:</h2>
          <span id="sensor-3-level" class="counter-element-
value">0</span>
        </li>
        <li class="counter-element lightOrange">
          <h2 class="counter-element-title">Об'єм:</h2>
          <span
            id="sensor-3-volume"
            class="counter-
element-value">0</span>
        </li>
      </ul>
    </div>
    <div class="counter-component">
      <h2 class="countet-component-title">Датчик 4</h2>
      <ul class="counter-list">
        <li class="counter-element yellow">
          <h2
            class="counter-element-title">Рівень
піддини:</h2>
          <span id="sensor-4-level" class="counter-element-
value">0</span>
        </li>
        <li class="counter-element yellow">
          <h2 class="counter-element-title">Об'єм:</h2>
          <span
            id="sensor-4-volume"
            class="counter-
element-value">0</span>
        </li>
      </ul>
    </div>
  </div>
</div>
</section>
</main>
</body>
</html>

```

Лістинг CSS-коду:

```

:root {
  --red-color: #bf2604;
  --red-gradient: linear-gradient(
    to right bottom,
    #bf2604,
    #6f0012,
    #4d0816,
    #2a0c12,
    #000000
  );
}

```

```
);
--orange-gradient: linear-gradient(
  to right bottom,
  #9d1f00,
  #8a1c00,
  #7a1800,
  #571100,
  #2b0900
);
--light-orange-gradient: linear-gradient(
  to right bottom,
  #ff7700,
  #ba5700,
  #ab5000,
  #6b3200,
  #3d1c00
);
--yellow-gradient: linear-gradient(
  to right bottom,
  #f2a950,
  #cf9349,
  #9b6c34,
  #744e20,
  #583d1d
);
}

body {
  font-family: "Montserrat", sans-serif;
  font-weight: 600;
  font-size: 16px;
  line-height: 1;
  color: white;
  background-image: url(../images/bg.jpg);
  background-repeat: no-repeat;
  background-size: cover;
}

.container {
  max-width: 1170px;
  padding: 25px;
  margin: 0 auto;
}

.counter {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: #000000ac;
}
```

```
}

.counter-container {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  display: flex;
  align-items: center;
  justify-content: center;
}

.counter-title {
  font-weight: 700;
  font-size: 25px;
  text-align: center;
  text-transform: uppercase;
  letter-spacing: 0.2em;
  line-height: 1.2;
  margin-bottom: 50px;
  text-shadow: 0px 0px 10px white;

  & span {
    color: var(--red-color);
    text-shadow: 0px 0px 10px var(--red-color);
  }
}

.counter-component {
  &:not(:last-child) {
    margin-bottom: 50px;
  }
}

.countet-component-title {
  font-weight: 500;
  font-size: 20px;
  text-align: center;
  text-transform: uppercase;
  margin-bottom: 20px;
}

.counter-list {
  display: flex;
  align-items: center;
  justify-content: center;
  gap: 80px;
}

.counter-element {
  border-radius: 10px;
}
```

```
animation: shadow 3s infinite;
padding: 15px;
text-align: center;

&.red {
  background-image: var(--red-gradient);
}
&.orange {
  background-image: var(--orange-gradient);
}
&.lightOrange {
  background-image: var(--light-orange-gradient);
}
&.yellow {
  background-image: var(--yellow-gradient);
}
}

.counter-element-title {
  margin-bottom: 10px;
  font-weight: 300;
  font-size: 18px;
  color: white;
}

.counter-element-value {
  font-weight: 700;
  font-size: 30px;
  color: white;
}

@keyframes shadow {
  0% {
    box-shadow: 0px 0px 10px rgb(192, 0, 0, 0.6);
  }
  50% {
    box-shadow: 0px 0px 30px rgba(192, 0, 0, 0.6);
  }
  100% {
    box-shadow: 0px 0px 10px rgb(192, 0, 0, 0.6);
  }
}
```