

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система контролю якості електронних компонентів

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

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Панасенко С.М.
(підпис) (прізвище та ініціали)

Керівник Паляниця Ю.Б.
(підпис) (прізвище та ініціали)

Нормоконтроль Тиш Є.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Осухівська Г.М.
(підпис) (прізвище та ініціали)

Рецензент Литвиненко Я.В.
(підпис) (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)

« 25 » 04 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 Панасенко Станіслав Михайлович
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система контролю якості електронних компонентів

Керівник роботи Паляниця Юрій Богданович, к.т.н., ст.викладач
(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 28.06.24

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. Безпека життєдіяльності, основи охорони праці

Висновки

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

Схема електрична принципова

Діаграма послідовностей

Блок-схема алгоритму роботи програмного забезпечення

Схема структурна

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., проф. каф. 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>01.02.24-09.02.24</i>	
2	<i>Аналіз технічного завдання</i>	<i>05.02.24-11.02.24</i>	
3	<i>Розробка структурної схеми</i>	<i>25.04.24-02.05.24</i>	
4	<i>Вибір елементної бази</i>	<i>03.05.24-10.05.24</i>	
5	<i>Вибір програмного забезпечення</i>	<i>10.05.24-14.05.24</i>	
6	<i>Реалізація проектної складової</i>	<i>14.05.24-09.06.24</i>	
7	<i>Опрацювання розділу «Безпека життєдіяльності, основи охорони праці»</i>	<i>10.06.24-15.06.24</i>	
8	<i>Оформлення пояснювальної записки кваліфікаційної роботи</i>	<i>16.06.24-20.06.24</i>	
9	<i>Оформлення графічної частини</i>	<i>20.06.24-21.06.24</i>	
10	<i>Підготовка до захисту</i>	<i>14.06.24</i>	
11	<i>Захист кваліфікаційної роботи</i>	<i>24.06.24-28.06.24</i>	

Студент _____
(підпис)

Панасенко С.М.
_____ (прізвище та ініціали)

Керівник роботи _____
(підпис)

Паляниця Ю.Б.
_____ (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система контролю якості електронних компонентів. // Кваліфікаційна робота бакалавр // Панасенко Станіслав Михайлович // 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.- 72, рис.- 35, табл.- 2, кресл.- 4, додат.- 4, бібліогр.- 16.

Ключові слова: комп'ютеризована система контролю якості електронних компонентів, ESP32, LCD-дисплей, мікроконтролер, тестування.

Кваліфікаційна робота складається з чотирьох розділів. В першому розділі описується аналіз технічного завдання розробки, визначаються основні завдання та вимоги до неї, аналіз існуючих рішень та моделей тестерів. В другому розділі описується розробка і проектування системи та вибір інструментів для її реалізації. В третьому розділі висвітлюється розробка алгоритму роботи програмного забезпечення, процес впровадження налаштувань програмного забезпечення та розробка програмної частини системи. Крім того розділ містить тестування готової системи. В четвертому розділі описується безпека життєдіяльності та основи охорони праці.

ANNOTATION

Computerized quality control system for electronic components // Bachelor's thesis // Stanislav Panasenko Mykhailovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer and Information System and Software Engineering, Department of Computer Systems and Networks, Group CI-42 // Ternopil, 2024 // p.- 72, fig.- 35, tab.- 2, sheets A1.- 4, appendix- 4, bibliography- 16.

Keywords: computerized quality control system for electronic components, ESP32, LCD display, microcontroller, testing

The bachelor's thesis consists of four chapters. In the first chapter, the analysis of the technical task for development is described, outlining the main objectives and requirements for it, as well as the analysis of existing solutions and models of testers. The second chapter describes the development and design of the system and the selection of tools for its implementation. The third chapter covers the development of the software algorithm, the process of software configuration implementation, and the development of the system's software part. Additionally, the chapter includes the testing of the completed system. The fourth chapter describes life safety and occupational health basics.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ.....	7
ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	9
1.1 Визначення основних завдань та вимог до комп'ютеризованої системи контролю якості електронних компонентів	9
1.2 Аналіз існуючих рішень завдання.....	10
1.3 Аналіз існуючих моделей тестерів	13
РОЗДІЛ 2 РОЗРОБКА ТА ПРОЄКТУВАННЯ СИСТЕМИ.....	18
2.1 Розробка структурної схеми комп'ютеризованої системи контролю якості електронних компонентів.....	18
2.2 Опис вибраних компонентів системи	20
2.2.1 Опис вибраної плати	20
2.2.2 Опис вибраного дисплея.....	23
2.2.3 Опис шини I2C	25
2.3 Опис вибору програмних інструментів для написання ПЗ	26
2.3.1 Інструмент для розробки ПЗ мікроконтролера.....	27
2.3.2 Інструмент для розробки вебсайту.....	28
2.4 Створення загальної архітектури системи	29
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	33
3.1 Створення алгоритму роботи комп'ютеризованої системи контролю якості електронних компонентів.....	33

					КС КРБ 123.128.00.00 ПЗ						
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>	<i>Комп'ютеризована система контролю якості електронних компонентів</i>			<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>	
<i>Розроб.</i>		<i>Панасенко С.М.</i>								5	72
<i>Перевір.</i>		<i>Паляниця Ю.Б.</i>									
<i>Реценз.</i>		<i>Литвиненко Я.В.</i>									
<i>Н. контр.</i>		<i>Тиш С.В.</i>									
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						<i>ТНТУ, каф. КС, гр. СІ-42</i>			

3.2 Налаштування програмного забезпечення	35
3.3 Опис функціональних бібліотек реалізації.....	38
3.4 Розробка програмного забезпечення для ESP32.....	41
3.5 Розробка програмного забезпечення для сайту	44
3.6 Тестування системи	47
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	53
4.1. Шляхи підвищення життєдіяльності людини	53
4.2 Заходи щодо захисту системи контролю якості електронних компонентів від короткого замикання.....	54
4.3 Долікарська допомога при переломах	56
ВИСНОВОК.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
Додаток А Технічне завдання	61
Додаток Б Перелік елементів	64
Додаток В Лістинг програми	68
Додаток Д Лістинг сайту	71

СПИСОК СКОРОЧЕНЬ

I2C – Inter-Integrated Circuit

Wi-Fi – Wireless Fidelity

LCD – Liquid Crystal Display

IoT – Internet of Things

API – Application Programming Interface

LCR – Inductance (L), Capacitance (C), and Resistance (R)

					КС КРБ 123.128.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

На сьогоднішній день електронні компоненти відіграють значну роль у розвитку технологій та забезпечення функціонування різних електронних пристроїв. Від точності та якості цих компонентів залежить надійність і ефективність роботи кінцевих продуктів. У зв'язку з цим, контроль якості електронних компонентів є важливим етапом у виробничому процесі.

Традиційні методи перевірки електронних компонентів нерідко є трудомісткими та вимагають значних витрат часу. Ще з початку розвитку комп'ютерних технологій виникла потреба у створенні систем, які б забезпечували швидкий та точний контроль якості електронних компонентів. Дані системи можуть дозволити значно підвищити продуктивність та знизити кількість браку, що свою чергу, зменшує виробничі витрати.

Основною метою даної роботи є розробка та впровадження комп'ютеризованої системи контролю якості електронних компонентів. В ній було створено тестер електричних компонентів, який дозволяє перевірити параметри електронних компонентів.

Завдання дослідження:

- проектування апаратної частини тестера;
- розробка програмного забезпечення для управління тестером та обробки даних;
- тестування та калібрування системи для забезпечення точності вимірювання.

Розроблена система є універсальним інструментом для контролю якості різноманітних електронних компонентів, що може бути використана як у лабораторних умовах, так і у виробництві.

					КС КРБ 123.128.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Визначення основних завдань та вимог до комп'ютеризованої системи контролю якості електронних компонентів

Основною метою даного проекту є розробка та впровадження комп'ютеризованої системи контролю якості електронних компонентів. Ця система має забезпечити точний рівень та надійність при перевірці якості електронних компонентів, що використовуються у виробництві електронних пристроїв.

Основна ціль створення даної системи, це моніторинг і аналіз результатів при перевірці. Можливості обробки та аналізу результатів перевірок дозволять виявити проблемні зони у виробничому процесі. Це сприятиме постійному вдосконаленню якості продукції та процесу її виробництва.

Для досягнення цієї мети визначимо головні завдання та вимоги до системи, вона має бути забезпечена точністю вимірювання та аналізу. Створений інтерфейс для користувача має бути зручним та інтуїтивний. Має бути присутня візуалізація результатів перевірок у зрозумілому вигляді. Використовувати стандартні протоколи передачі даних для забезпечення безперебійної роботи. Має включати сервер, на який надсилаються дані із утворенням бази даних, це дозволить централізовано зберігати результати перевірок та здійснювати аналіз. Забезпечення надійного збереження даних та можливість оперативного доступу до них для подальшого аналізу та обробки.

					КС КРБ 123.128.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>				
<i>Розроб.</i>		<i>Панасенко С.М.</i>			<i>РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Паляниця Ю.Б.</i>					9	8
<i>Реценз.</i>		<i>Литвиненко Я.В.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. контр.</i>		<i>Тиш С.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

1.2 Аналіз існуючих рішень завдання

Системи тестування електронних компонентів почали розвиватися у 1950-х роках разом із розвитком електроніки та мікроелектроніки. Основними причинами розвитку таких систем були потреби в підвищенні надійності та якості електронних пристроїв і зростання складності електронних схем. З часом тестування електронних компонентів переходило із ручного у автоматизований процес. Ці системи включали використання комп'ютерів для автоматичного тестування параметрів компонентів, що забезпечило більш точні та надійні результати.

Технології контролю якості електронних компонентів постійно еволюціонують, щоб відповідати зростаючим вимогам та потребам ринку. Сучасні комп'ютеризовані системи контролю якості включають різноманітні рішення, що базуються на передових методах діагностики та аналізу даних. Розглянемо існуючі підходи до вирішення цього завдання, їх переваги та недоліки.

Автоматизовані оптичні системи є найбільш розповсюджені методи контролю якості електронних компонентів. Вони використовують високоякісні камери та спеціалізовані алгоритми обробки зображень та автоматичного виявлення дефектів, вже безпосередньо розміщені на платі. Ці системи дозволяють швидко і точно ідентифікувати дефекти.

Приклад компанії які пропонують рішення в даній області контролю електронних компонентів:

- Omron – компанія, яка пропонує автоматичну оптичну інспекцію, що забезпечує виявлення дефектів завдяки використанню передових алгоритмів обробки зображень;
- Koh Young Technology – компанія, яка спеціалізується на 3D автоматичній оптичній інспекції, дозволяє виявляти поверхневі дефекти, що значно підвищує якість контролю.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Автоматизовані рентгенівські інспекційні системи використовуються для виявлення прихованих дефектів, які не можуть бути виявлені за допомогою оптичного контролю. Дана система є особливо ефективна при перевірці якості пайки та з'єднань, оскільки допомагає виявити такі дефекти, як порожнеча, неповні з'єднання, а також мікротріщини.

Приклад компанії які пропонують рішення в даній області контролю електронних компонентів:

– Nordson DAGE – компанія, яка спеціалізується на виробництві рентгенівських систем для інспекції електронних компонентів та вузлів. Ці системи мають високу роздільну здатність та точність виявлення дефектів;

– Viscom – компанія, яка забезпечує високу роздільну здатність та точність виявлення дефектів, що дозволяють ефективно контролювати електронні компоненти з'єднані на друкованих платах.

Останні досягнення у галузі штучного інтелекту дозволило створити системи, які використовують алгоритми машинного навчання для аналізу якості електронних компонентів. Такі системи здатні адаптуватися до нових типів дефектів, постійно покращуючи свою точність завдяки самонавчанню на основі зворотного зв'язку.

Приклад компанії яка пропонують рішення в даній області контролю якості електронних компонентів є Kitov Systems. Вона використовує штучний інтелект для аналізу зображень електронних компонентів, забезпечуючи точні виявлення дефектів та автоматичне навчання системи на основі зворотного зв'язку.

Електричні тестові системи використовуються для перевірки функціональних характеристик електронних компонентів та зібраних друкованих плат. Ці методи включають функціональне тестування, векторне тестування, а також методи аналізу сигналів. Функціональне тестування передбачає перевірку працездатності компонентів у реальних умовах експлуатації, що дозволяє визначити їхню відповідність технічним вимогам.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Приклад компанії яка пропонує рішення в даній області контролю якості електронних компонентів є Teradyne. Дана компанія пропонує широкий спектр тестових систем для перевірки функціональних характеристик електронних компонентів. Системи автоматизованого функціонального тестування забезпечує високу точність та надійність результатів, а також має можливість тестування на різних етапах виробничого процесу.

Ознайомившись із різними методами тестування електронних компонентів, можна визначити їх переваги та недоліки зображені на табл. 1.1.

Таблиця 1.1 – Переваги та недоліки методів

Метод	Основні переваги	Обмеження
Автоматизований оптичний контроль	Висока точність виявлення поверхневих дефектів, швидкість перевірки	Не виявляє приховані дефекти, обмеження у виявленні дефектів на складних компонентах
Автоматизований рентгенівський контроль	Можливість виявлення прихованих дефектів, висока точність	Висока вартість обладнання, повільніший процес інспекції порівняно з автоматизованим оптичним контролем
Штучний інтелект	Адаптивність до нових типів дефектів, висока точність	Високі вимоги до обчислювальних ресурсів, залежить від якості навчальних даних

Метод	Основні переваги	Обмеження
Електричні тестові методи	Перевірка функціональних характеристик, висока точність	Не виявляє фізичні дефекти, потребує спеціальних тестових програм

Проаналізувавши можливі моделі рішення було обрано електричні тестові методи контролю якості електронних компонентів.

1.3 Аналіз існуючих моделей тестерів

Для ефективної роботи з радіодеталями необхідно використовувати спеціалізовані тестери, які дозволять оцінити їх параметри та працездатність. Тому розглянемо деякі типи тестерів, визначивши їх особливості, переваги та недоліки. Проаналізувавши дані моделі можна зрозуміти, які інструменти найкраще підійдуть до конкретного завдання [8].

Аналоговий тестер радіодеталей, один з найпростіших інструментів для вимірювання параметрів. Він використовується переважно для базових перевірок резисторів, конденсаторів та інших компонентів. Прикладом такого може бути тестер Sanwa YX-360TRF зображений на рис. 1.1.



Рисунок 1.1 – Тестер Sanwa YX-360TRF

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Його перевагами є простота у використанні, завдяки простому інтерфейсу та інтуїтивно зрозумілим показникам. Він є відносно недорогим, що робить його доступним. Відсутність використання батарейок чи зовнішнього джерела живлення.

Однак має недоліком меншу точність у порівнянні з цифровим тестером, адже аналогові моделі мають нижчу точність при вимірюванні. Здатний вимірювати лише основні параметри, що робить його менш корисним у складних завданнях.

Цифрові тестери забезпечують більшу точність при вимірюванні та мають більш широкий набір функцій для тестування. Прикладом такого може бути тестер Peak Atlas DCA55 Semiconductor Component Analyzer зображений на рис. 1.2.



Рисунок 1.2 – Тестер Peak Atlas DCA55

Його перевагою є висока точність результатів вимірювання завдяки цифровій обробці сигналів. Автоматичне визначення типу компонента і його

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

параметри, що спрощує процес тестування. Можливість тестування різних напівпровідникових компонентів.

До недоліків можна віднести його велику вартість та потребу у джерелі живлення, що може створити додаткові незручності при експлуатації.

Тестери еквівалентного послідовного опору, використовуються для вимірювання еквівалентного послідовного опору конденсаторів, що дозволяє визначити їх стан без демонтажу плати. Прикладом такого може бути тестер MESR-100 V2 зображений на рис. 1.3.



Рисунок 1.3 – Тестер MESR-100 V2

До переваг даного тестера можна віднести високу точність вимірювання опору конденсаторів. Можливість тестування конденсаторів без демонтажу, що значно спрощує процес тестування вбудованих компонентів.

Його недоліком є вузька спеціалізація, а саме підходить лише для вимірювання еквівалентного послідовного опору конденсаторів і не може використовуватись для інших типів компонентів. Також вони є дорожчі за універсальні тестери.

LCR-тестери, що використовуються для вимірювання індуктивності, ємності та опору компонентів, вони є часто використовувані для роботи з пасивними елементами. Прикладом такого може бути тестер DE-5000 Handheld LCR Meter зображений на рис. 1.4.



Рисунок 1.4 – Тестер DE-5000 Handheld

До його переваг відноситься висока точність вимірювання індуктивності, ємності та опору. Широкий діапазон для вимірювання параметрів, що робить його універсальним. Компактний розмір робить його легким при переносі та експлуатації в різних умовах.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Великим недоліком є його вартість, адже він значно дорожчий та прості тестери. Потребує використання батарейок або іншого джерела живлення. Складність при використанні, через високоточні параметри налаштування.

В результаті аналізу існуючих пристроїв, було визначено деякі можливі рішення для реалізації системи контролю якості електронних компонентів. Дані тестери працюють в основному ізольовано, демонструючи результати вимірювання на власному дисплеї. Однак, для власної розробки системи, значення отримані при етапі тестування, будуть виводитись на вебсайт. Та зберігатись на платформі, щоб забезпечити зберігання даних та їх аналіз при подальшому використанні.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

РОЗДІЛ 2 РОЗРОБКА ТА ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Розробка структурної схеми комп'ютеризованої системи контролю якості електронних компонентів.

Для кращого розуміння роботи комп'ютеризованої системи контролю якості електронних компонентів, розглянемо структурну схему зображену на рис. 2.1.

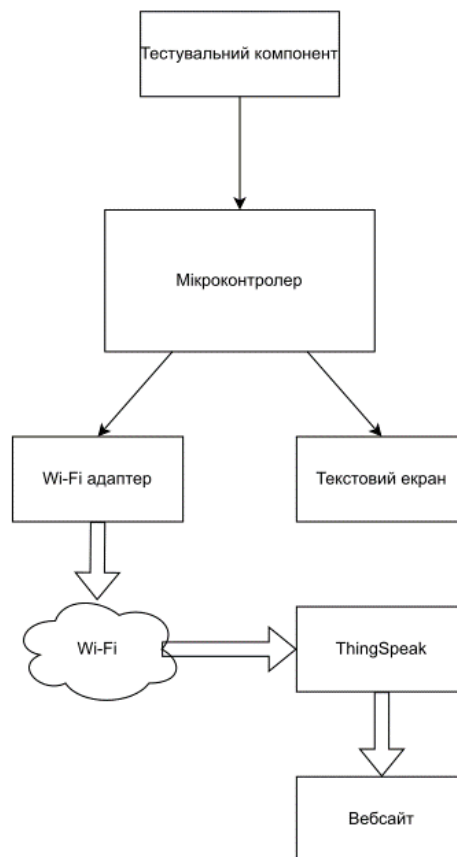


Рисунок 2.1 – Структурна схема комп'ютеризованої системи контролю якості електронних компонентів

					КС КРБ 123.128.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>				
<i>Розроб.</i>		<i>Панасенко С.М.</i>			<i>РОЗДІЛ 2 РОЗРОБКА ТА ПРОЄКТУВАННЯ СИСТЕМИ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Паляниця Ю.Б.</i>					18	15
<i>Реценз.</i>		<i>Литвиненко Я.В.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. контр.</i>		<i>Тиш С.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

Компонентами системи є тестувальний компонент, плата ESP32, Wi-Fi адаптер, текстовий екран, Wi-Fi мережа, база даних, вебсайт. Всі вони виконують свою функцію:

- тестувальний компонент, електронний компонент, який проходить тестування та якість та працездатність. Надає свої характеристики та параметри до мікроконтролера для аналізу;

- мікроконтролер, система управління, що здійснює обробку та передачу даних від тестувального компонента. Приймає сигнали від тестувального компонента, аналізує, обробляє та передає інформацію;

- текстовий екран, LCD I2C дисплей, відображає результати тестування у реальному часі. Надає миттєвий зворотній зв'язок оператору, що дозволяє відразу оцінити отриманий результат тестування;

- Wi-Fi адаптер, забезпечує бездротове з'єднання між мікроконтролером та вебсайтом. Зібравши отримані дані від мікроконтролера, відправляє дані на сервер та базу даних, через Wi-Fi мережу, забезпечивши віддалений доступ до даних;

- Wi-Fi мережа, забезпечує з'єднання між Wi-Fi адаптером і сервером/базою даних. Передає дані від системи контролю якості до віддаленої бази даних для зберігання та подальшого аналізу;

- ThingSpeak, зберігає інформацію, отриману від мікроконтролера через Wi-Fi адаптер. Збирає отримані дані про результати тестування, надає можливість їх зберігання, аналізу та доступ до них у майбутньому;

- вебсайт, інтерфейс користувача, забезпечує доступ до бази даних через інтернет. Дозволяє користувачу переглядати результати тестування, аналізу даних. У майбутньому можливе вдосконалення для різних рівнів доступу користувачів;

Отже система починає свою роботу від ініціалізації тестування, електронний компонент підключається до мікроконтролера. ESP32 ініціює процес тестування, зчитує необхідні параметри компоненту та формує

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

результати тестування. Після цього передає результати та текстовий екран, де вони відображаються для оператора. Це дозволяє оперативно отримати інформацію про стан компоненту. Паралельно мікроконтролер відправляє оброблені дані на Wi-Fi адаптер, який забезпечує їх передачу через Wi-Fi мережу на сервер. Дані отримуються від мікроконтролера та зберігаються до бази даних, для подальшого використання. Вебсайт, підключений до бази даних, надає користувачам можливість перегляду цих даних.

2.2 Опис вибраних компонентів системи

У даному розділі описується компонентна складова системи контролю якості електронних компонентів

2.2.1 Опис вибраної плати

Основною складовою комп'ютеризованої системи контролю якості електронних компонентів є мікроконтролер ESP32-WROOM-32, плата для розробки ESP32-DEVKIT v1.

Даний мікроконтролер є дуже популярним серед розробників, що займаються над проектами IoT та вбудованими системами. Він має в собі двоядерний процесор Tensilica LX6, який працює на частоті до 240 МГц. Це забезпечує проектам високу продуктивність та ефективне керування ресурсами [9].

Основною перевагою даного мікроконтролера є розширені можливості введення-виведення (I/O), також має різноманітні цифрові та аналогові входи-виходи, включаючи GPIO, I2C, UART, SPI.

ESP32 DevKit v1, містить в собі вбудовані модулі Bluetooth 4.2 та Wi-Fi 802.11 b/g/n. Це дозволяє використовувати мікроконтролер для з'єднання з іншими пристроями та мережами. Щодо різних інтерфейсів, містить Can,

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Ethernet, це робить його універсальним рішенням для застосування у різноманітних сферах промисловості [10].

Даний мікроконтролер є доволі гнучким у виборі середовища розробки, деякі із них Arduino IDE, MicroPython, Espressif IoT Development Framework, Simba Embedded Programming Platform, Espruino. Саме завдяки цьому мікроконтролер підходить для реалізації різних проектів, як по масштабу так і по рівню досвіду роботи з ним.

Основні компоненти плати ESP32-DevKit v1 зображено на рис. 2.2

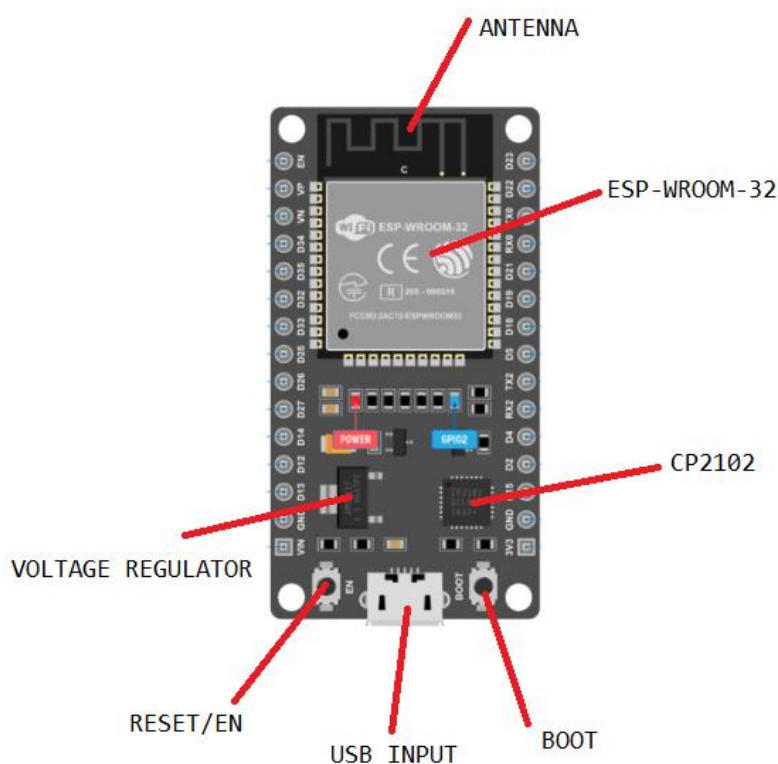


Рисунок 2.2 – Зображення основних компонентів плати

Опис основний компонентів:

- антена, використовується для безпроводного зв'язку, зокрема для Wi-Fi та Bluetooth. Вона дозволяє мікроконтролеру отримувати та відправляти бездротові сигнали з іншими пристроями та мережами, що є критично важливим для реалізації IoT проектів та бездротових мереж;

- ESP32-WROOM-32, це модуль, що містить сам мікроконтролер ESP32, його пам'ять та інші важливі компоненти. Він є серцем ESP32 DevKit v1, адже забезпечує виконання програмного забезпечення та взаємодію з іншими різними компонентами;

- CP2102, являє собою USB-Seria конвертер, який використовується для забезпечення зв'язку між мікроконтролером та комп'ютером через USB-порт. Саме він дозволяє програмувати мікроконтролер та передавати дані між ним і комп'ютером, що робить процес налаштування та відладки зручним і швидким;

- Boot, бутлоадер, це програмне забезпечення, яке відповідає за завантаження та виконання основного програмного забезпечення. Він дозволяє завантажити нове програмне забезпечення по USB-порту. Бутлоадер полегшує процес оновлення програмного забезпечення, дозволяючи швидко змінювати код без необхідності використання додаткового обладнання;

- USB INPUT, роз'єм, підключення кабеля для надання заряду у мікроконтролер, програмування і передачі даних між пристроями;

- RESET/EN, кнопка що використовується для скидання мікроконтролера у вихідний стан, ввімкнення його, якщо мікроконтролер був вимкнений. Це важливий інструмент для відладки, оскільки дозволяє швидко перезавантажити систему і продовжити роботу без переривань;

- Voltage Regulator, стабілізатор напруги, компонент, який забезпечує стабільну напругу живлення для різних компонентів мікроконтролера. Він гарантує стабільну роботу мікроконтролера незалежно від змін напруги живлення системи [11].

Завдяки цьому набору функцій, ESP32 є хорошим вибором для створення проектів, що вимагають надійного та ефективного мікроконтролера з можливістю безпроводного зв'язку. Характеристика мікроконтролера наведено у табл. 2.1.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

Таблиця 2.1 – Характеристика мікроконтролера

Модель	ESP32 DevKit v1
Мікроконтролер	ESP32
Процесор	2-ядерний Xtensa Dual-Core 32bit LX6
Тактова частота	80, 160, 240МГц
Флеш-пам'ять	448 Кбайт
Оперативна пам'ять	520 Кбайт
Кількість виводів на платі	30
Перетворювач USB to UART	CP2102
Антенa	PCB
Bluetooth	4.2 BR/EDR/BLE
Wi-Fi	IEEE 802.11b, безпека WPA, WPA/WPA2, WAPI, частота 2,4 ГГц, швидкість до 150 Мбіт/с, вбудований стек TCP/IP
Режим бездротового зв'язку	STA/AP/STA+AP
Відстань передачі/прийому	400м
Розмір та вага	51 x 28 x 13 мм, 9 г.

2.2.2 Опис вибраного дисплея

Для відображення отриманої інформації було обрано LCD 1602 дисплей, він відображає текстову інформацію у вигляді двоцифрових символів. Його технологія відображення базується на використанні рідкокристалічних матеріалів. Даний дисплей має розмір 16 символів у рядку та 2 рядки, у двох рядках може вмістити 32 ASCII символи. Кожен символ складається з матриці пікселів, яка може бути увімкнена чи вимкнена, щоб створити символи, букви або графічні елементи [12].

Даний дисплей значно легше використовувати та програмувати за допомогою I2C модуля.

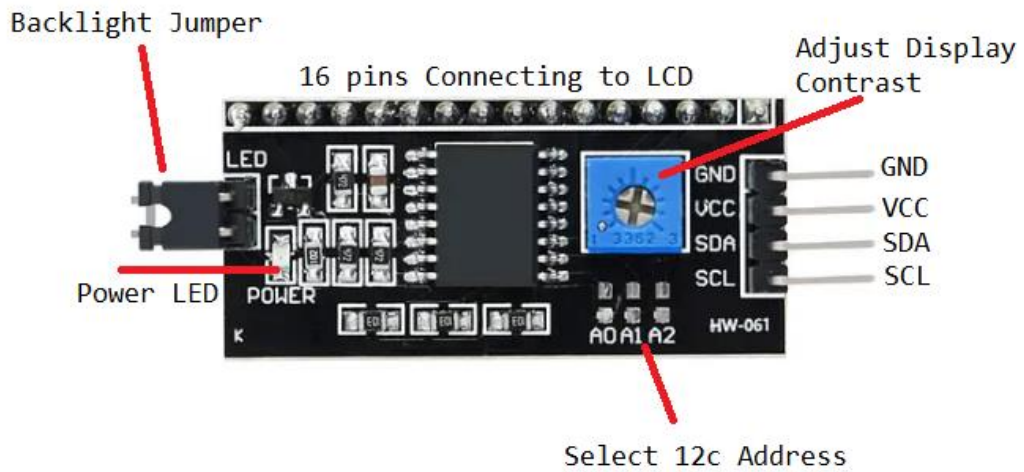


Рисунок 2.3 – Зображення I2C модуля

В основі даного модуля покладено 8-розрядну мікросхему розширення вводу та виводу, PCF8574T. Він призначений для роботи з напругою живлення від 2.5В до 6В. Пристрій має 8-бітний порт введення/виведення (P0-P7), включаючи виводи замикання з високою здатністю до великого струму для прямого управління світлодіодами.

Схема виводів PCF8574 зображено на рис. 2.4

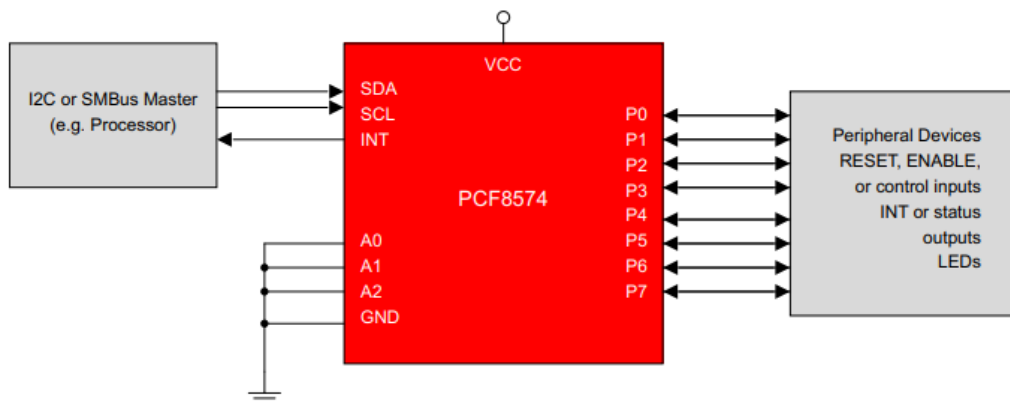


Рисунок 2.4 – Схема виводів PCF8574

Модуль I2C під'єднується до контактів LCD 1602, що спрощує використання дисплею. Отримана пристрій легко підключається до основної плати за допомогою контактів GND-заземлення, VCC-живлення, SDA-отримання даних, SCL-синхронізація [13]. Зображення підключення LCD та I2C модуля рис. 2.5.

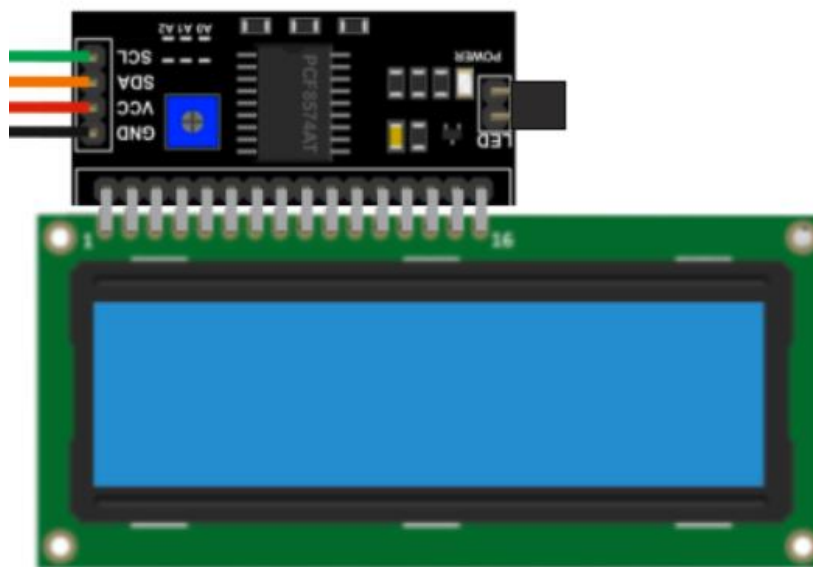


Рисунок 2.5 – Підключення модуля I2C до LCD

2.2.3 Опис шини I2C

I2C - це послідовна шина даних, призначена для обміну інформацією між інтегральними схемами, що використовує дві двосторонні лінії зв'язку (SDA і SCL). Вона використовується для підключення низькопродуктивних периферійних пристроїв до материнської плати, вбудованих систем і мобільних телефонів. I2C використовує дві двосторонні лінії, підняті до напруги живлення і керовані через відкритий колектор або відкритий стік - лінія послідовного передавання даних (SDA, Serial Data) і лінія послідовного тактування (SCL, Serial CLock). Стандартні напруги +5 В або +3,3 В, однак допускаються й інші.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Адресація включає 7-бітовий адресний простір з 16-ма зарезервованими адресами. Це означає до 112 вільних адрес для підключення на одну шину.

Синхронізація в шині I2C здійснюється за допомогою лінії SCL. Ведучий генерує сигнали на цій лінії, але ведений може утримувати її в низькому стані, якщо йому потрібно більше часу для обробки прийнятого біта.

Цей механізм дозволяє веденому контролювати швидкість передачі даних. На байтовому рівні ведений може утримувати лінію SCL в низькому стані після прийому байта, щоб дати головному час на передачу наступного байта. На бітовому рівні ведений може уповільнити частоту синхроімпульсів, продовжуючи їх низький період.

Переваги шини I2C:

- простота для управління шиною ,достатньо одного мікроконтролера;
- економічність для підключення пристроїв, використовуються лише два провідника;
- гнучкість, можлива одночасна робота декількох провідних пристроїв.
- мобільність, підтримує гаряче підключення і відключення пристроїв;
- надійність, вбудований фільтр зменшує імпульси в шині, забезпечуючи цілісність даних;

Недоліки шини I2C:

- складність програмування, через можливі нештатні ситуації на шині;
- трудність локалізації несправності, якщо один з пристроїв помилково встановлює на шині стан низького рівня;

2.3 Опис вибору програмних інструментів для написання ПЗ

У даному розділі описується програмне забезпечення для написання програмної складової системи.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

2.3.1 Інструмент для розробки ПЗ мікроконтролера

Для написання ПЗ для мікроконтролера мною було обрано програмне забезпечення Arduino IDE. Дане середовище написання програмного коду є доволі легким та зрозумілим у використанні. Воно включає в себе текстовий редактор для написання коду, область повідомлень, текстову консоль, панель з інструментами та додатковими функціями для налаштування програмного середовища. Код програми для Arduino пишеться на мові програмування C/C++, та має велику кількість бібліотек, що дозволить додати до проекту більший функціонал.

Основні інструменти для завантаження та компіляції коду, дозволять швидко тестувати та виправляти написаний код безпосередньо на апаратному забезпеченні. Arduino IDE є кросплатформерною, може працювати на різних операційних системах, та онлайн. Додатково можна зазначити, що середовище має підтримку широкого спектру плат Arduino та сумісних пристроїв, таких як ESP32, STM32, Teensy.

Однак в програмного забезпечення є й недоліки, воно може не підтримувати деякі новіші або більш спеціалізовані плати чи пристрої, що потребують спеціальних інструментів або фреймворків. Присутня проблема сумісності із деякими операційними системами або браузерами, якщо їх не оновлено або налаштовано належним чином.

Зображення головного інтерфейсного меню Arduino IDE зображено на рис. 2.6.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27



Рисунок 2.6 – Інтерфейс Arduino IDE

2.3.2 Інструмент для розробки вебсайту

Отримані дані, які будуть надходити при тестуванні, відобразатимуться на вебсайті. Для написання вебсайту було обрано програмний інструмент WebStorm. Він був розроблений компанією JetBrains, та є одним із популярних середовищ розробки веб-ресурсів. Даний інструмент містить розширені можливості редагування коду, що сприяє полегшенню написання та редагування CSS, HTML, JavaScript, TypeScript. За допомогою підсвічування синтаксису, автодоповнення та інтелектуальній рекомендації, дозволяє розробнику ефективно та швидко працювати над кодом.

Одна з основних функцій WebStorm є інструменти для редагування коду. Розробник може безпечно модифікувати структуру свого коду, використовуючи операції, такі як перейменування змінних, виділення фрагментів коду у функції та інші. Це допомагає забезпечити структурованість та легкість обслуговування коду під час подальшого розвитку програми.

Крім того містить вбудовану можливість тестування, яка дозволяє легко виявити та виправити помилки у коді. Виявлення помилок проводиться

безпосередньо у середовищі розробки, що дозволяє ефективно вирішувати проблеми та поліпшувати програму.

Таким чином, цей інструмент є надзвичайно потужним та універсальним інструментом для веб-розробки, забезпечуючи всі необхідні функції для створення, редагування, налаштування та тестування коду завдяки сучасним засобам та фреймворкам. На рис. 2.7 зображено меню інструмента WebStorm.

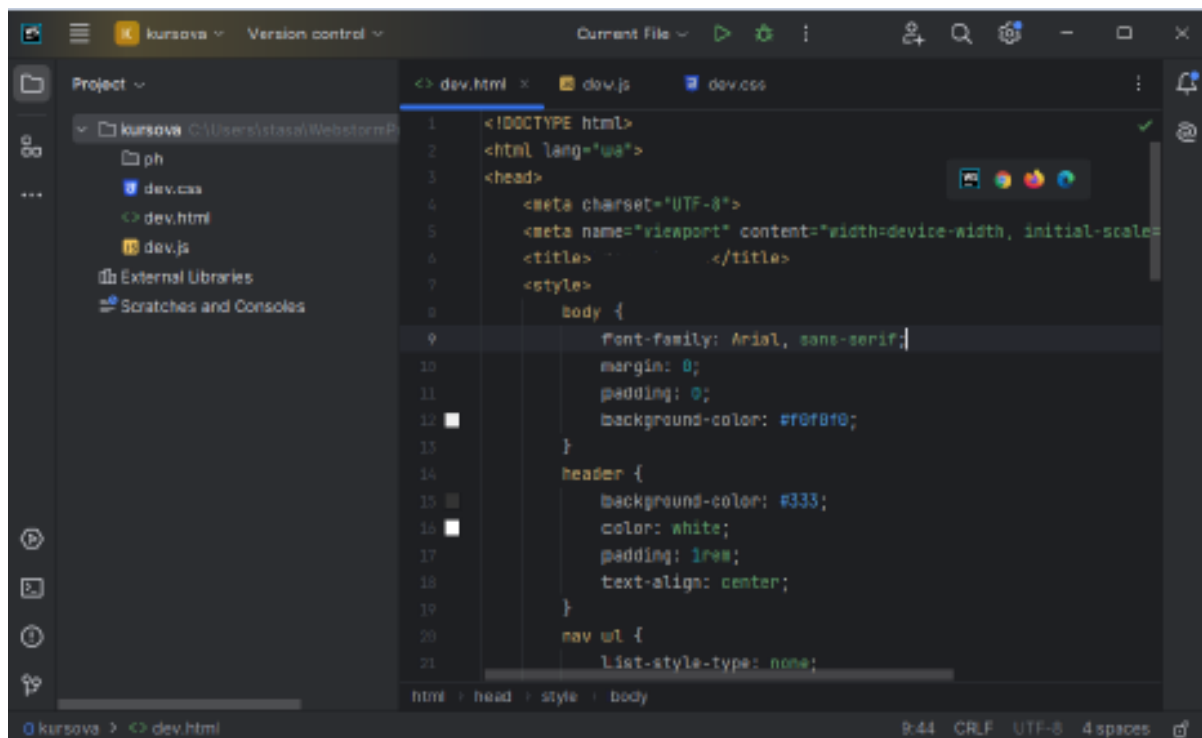


Рисунок 2.7 – Меню інструмента WebStorm

2.4 Створення загальної архітектури системи

Основним елементом схеми є мікроконтролер ESP32 (рис. 2.8), який виконує функцію керування всією системою.

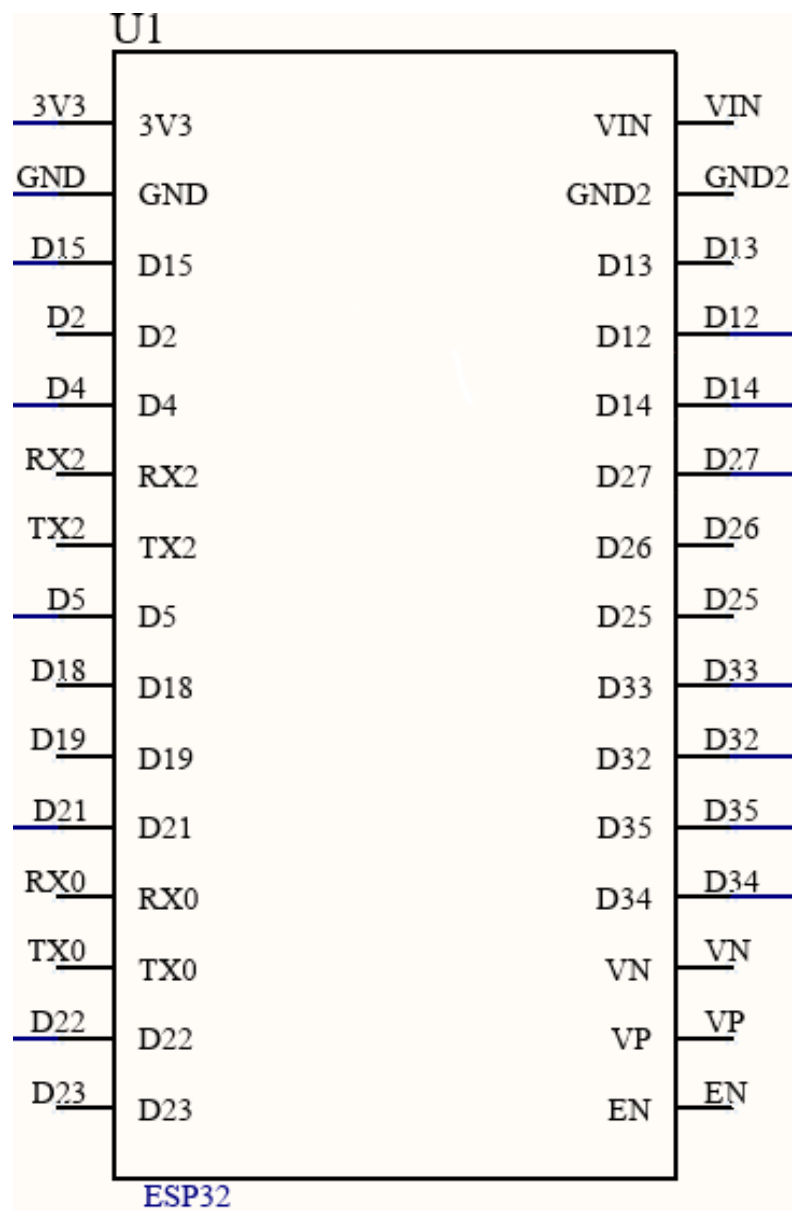


Рисунок 2.8 – Електрична принципова схема мікроконтролера

Розглянувши рис. 2.8 можна побачити спосіб під'єднання компонентів резисторів, кнопки, дисплея, роз'єм коннектор. Вони створюють загальну комп'ютеризовану систему контролю якості електронних компонентів. Загальну електрично принципову схему підключення компонентів зображено на рис. 2.9.

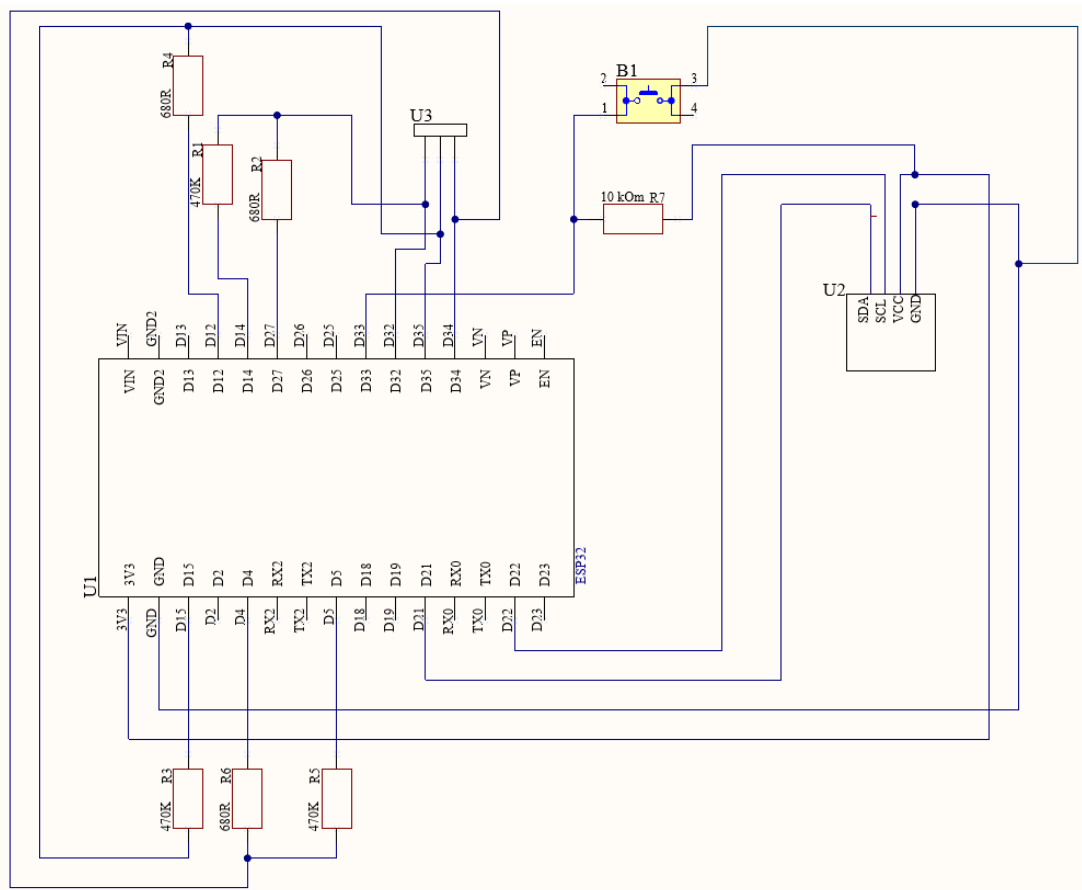


Рисунок 2.9 – Загальна електрично принципова схема підключення
КОМПОНЕНТІВ

Схема підключення, зображена на рис. 2.9, детально демонструє підключення електронних компонентів до мікроконтролера, який позначений символом U1. Центральним елементом даної схеми є мікроконтролер, який має ряд входів та виходів, позначені як D2-D35. Ці порти використовуються для підключення різних компонентів та модулів.

На лівій частині схеми зображено підключення декількох резисторів R1-R6, які підключені до входів мікроконтролера D15, D4, D5, D12, D14, D27. Дані резистори використовуються для обмеження струму та створення певного рівня напруги на входах мікроконтролера. Виходи D32, D35, D34, під'єднані до роз'єму U3, який має на собі задачу додавання до загальної схеми електронного компонента, для його тестування через повне коло системи.

У верхній частині схеми представлено підключення перемикача В1. Цей перемикач з'єднаний з одним із вхідних портів мікроконтролера через резистор R7, що має номінал 10кОм. Перемикач підключений таким чином, що при його активації сигнал подається на вхід мікроконтролера, пін D33. Це дозволяє здійснювати керування та вводити команди для контролю. З'єднання перемикача з землею GND та живленням VCC через резистор забезпечує стабільну роботу та запобігає можливим коливанням сигналу.

У правій частині схеми показано підключення LCD I2C дисплея, модуль U2. Він підключений до вихідних портів мікроконтролера D21, D22, та підключений до контактів живлення VCC та землі GND (рис. 2.10).

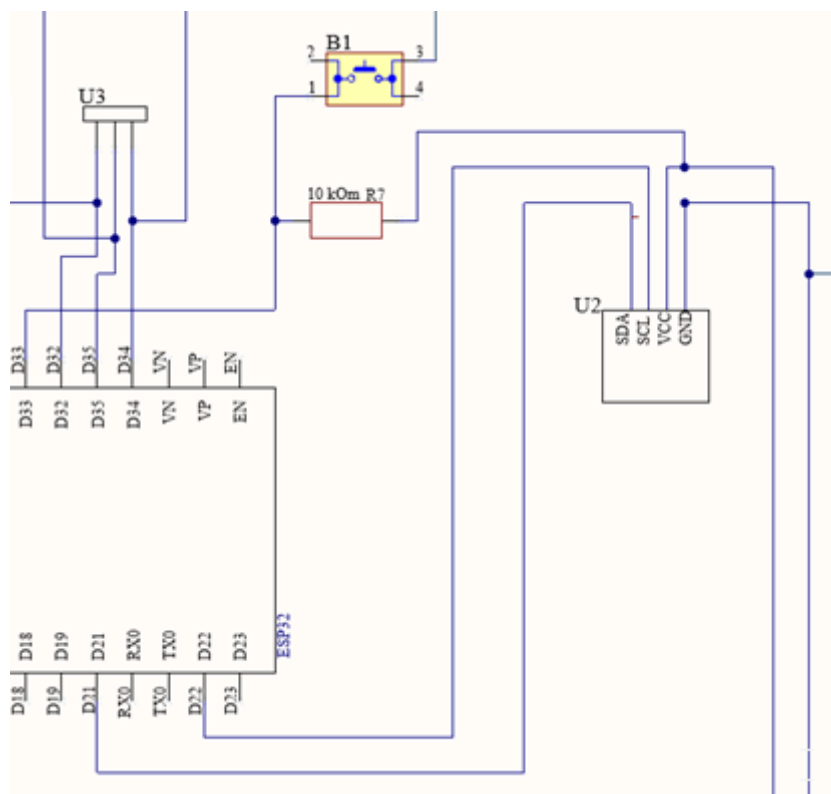


Рисунок 2.10 – Підключення LCD дисплея

Отже система побудована таким чином, щоб забезпечити стабільну і надійну роботу всіх компонентів, де кожен елемент виконує свою роль у забезпеченні функціонування системи.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Створення алгоритму роботи комп'ютеризованої системи контролю якості електронних компонентів

Блок-схема що зображена на рис. 3.1, представляє алгоритм роботи програмного забезпечення, що забезпечує збір та передачу даних з підключених елементів через Wi-Fi на платформу ThingSpeak та веб-сервер.

Детальний опис етапів блок-схеми:

а) програма починається з етапу початок, де відбувається запуск всіх необхідних процесів;

б) ініціалізація екрану на цьому етапі відбувається підготовка екрану для виведення інформації;

в) ініціалізація пінів, налаштування пінів для зв'язку з периферійними пристроями, задається режим роботи вхідних та вихідних пінів з їх початковим станом;

г) встановлення з'єднань з бездротовою мережею Wi-Fi. Це необхідно для передачі даних на віддалені сервери, такі як платформа ThinkSpeak, та доступу до веб-сервера;

г) ініціалізація ThingSpeak, налаштування параметрів для взаємодії з платформою ThingSpeak, встановлення API ключів і підготовку для відправки даних;

д) запуск веб-сервера, який дозволяє користувачу отримувати доступ до даних через веб-браузер;

					КС КРБ 123.128.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>				
<i>Розроб.</i>		<i>Панасенко С.М.</i>			<i>РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Паляниця Ю.Б.</i>					33	19
<i>Реценз.</i>		<i>Литвиненко Я.В.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. контр.</i>		<i>Тиш С.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

е) очікування тестування, програма знаходиться у стані очікування натискання кнопки та вставленого елемента, який ініціює процес тестування елемента;

е) кнопка перевірки елемента на даному етапі перевіряється, чи була натиснута кнопка перевірки елемента. Якщо кнопка натиснута, програма переходить до наступного етапу.

Якщо ні, повертається до попереднього стану очікування;

ж) підключення елемента, перевірка наявності підключеного елемента. Якщо елемент не підключено програма знову повертається до стану очікування натискання. Якщо елемент підключено, відбувається процес тестування;

з) виконується тестування підключеного елемента, зчитування даних;

и) вивід результату тестування про елемент на екран;

і) дані, зібрані під час тестування, відправляються на сервер ThingSpeak. Для зберігання та аналізу даних в режимі реального часу;

ї) отримані дані виводяться на веб-сервері, що дозволяє користувачам отримувати доступ до інформації через мережу;

й) завершення роботи програми.

					КС КРБ 123.128.00.00 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

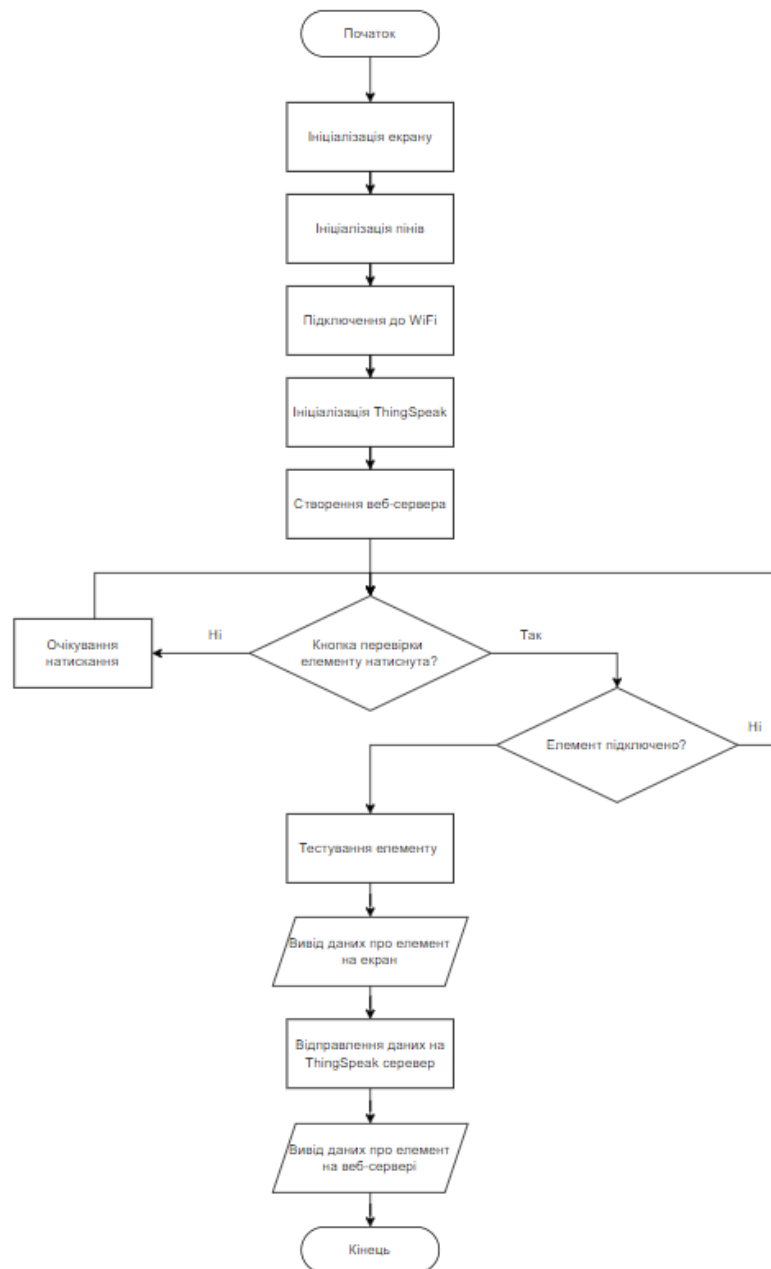


Рисунок 3.1 – Блок-схема алгоритму роботи програмного забезпечення

3.2 Налаштування програмного забезпечення

Основними платами для програмування у середовищі Arduino IDE є серія плат Arduino. Тому для подальшої роботи із ПЗ, встановимо необхідну бібліотеку для роботи із платою ESP32 DevKit v1.

У головному меню Arduino IDE натискаємо Файл > Налаштування, відкривається меню налаштувань для роботи із проектами. Переходимо до параметру із назвою "Additional Boards Manager URLs:" та додаємо посилання на бібліотеку:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Введення посилання для встановлення бібліотеки ESP32 зображено на рис. 3.2

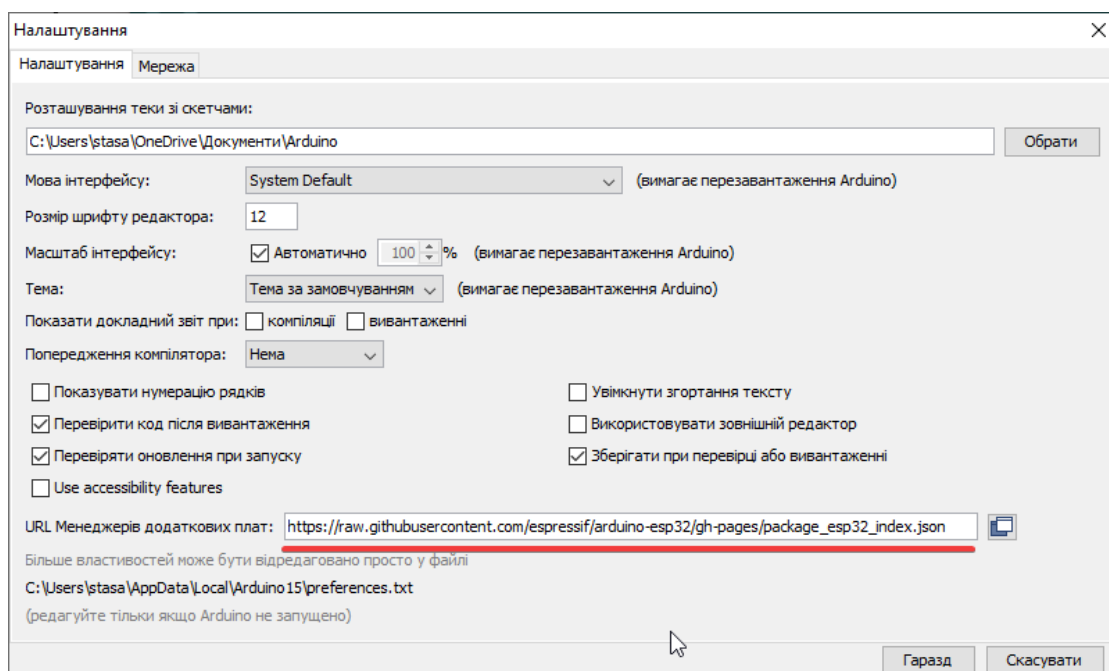


Рисунок 3.2 – Введення посилання для встановлення бібліотеки ESP32

В головному меню натискаємо на Інструменти > Плати > Менеджер плат. У відкритому вікні вписуємо у рядок ESP32 та качаємо потрібну нам версію прошивки. Кроки встановлення потрібної версії прошивки зображено на рис. 3.3 та 3.4.

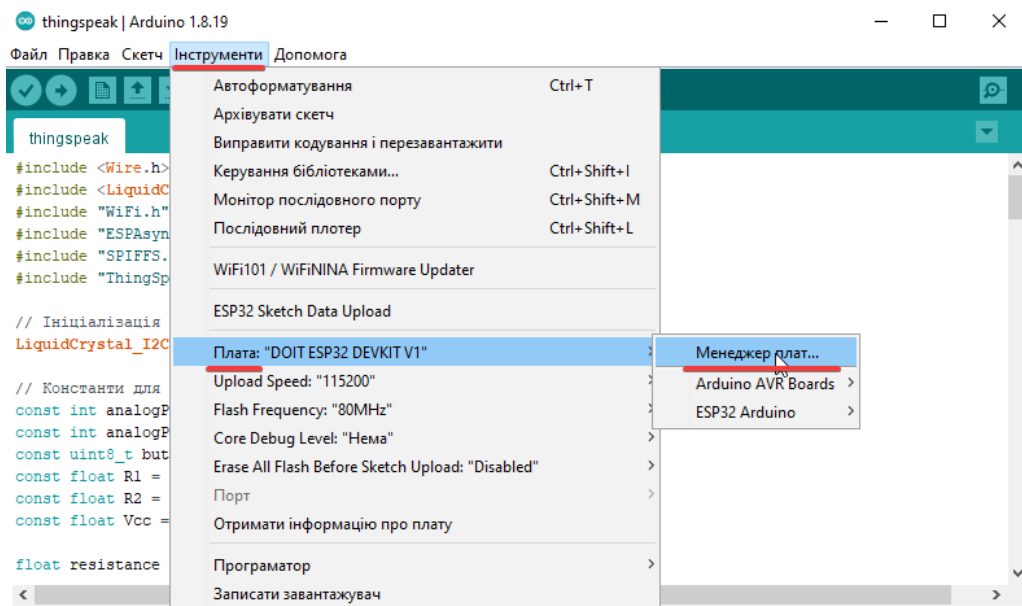


Рисунок 3.3 – Процес вибору прошивки плати

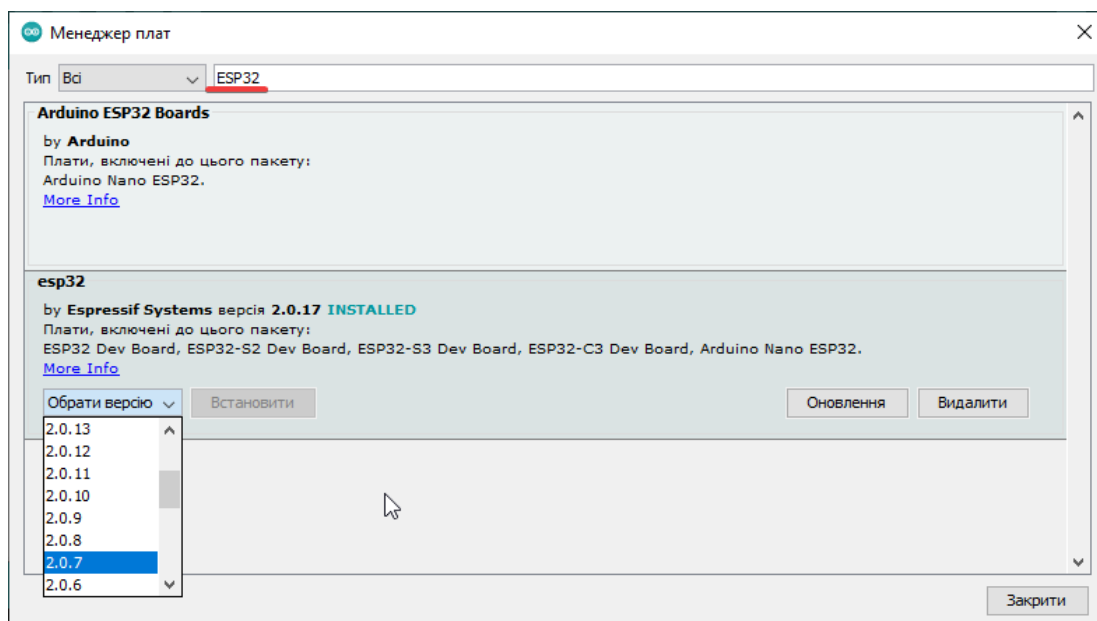


Рисунок 3.4 – Встановлення потрібної прошивки плати

Після виконання даних налаштувань програмне забезпечення готове до роботи із мікроконтролером ESP32 DevKit v1.

3.3 Опис функціональних бібліотек реалізації

Бібліотека “Wire.h” реалізує протокол I2C, який широко використовується для зв’язку між мікроконтролером та периферійними пристроями. Протокол I2C використовує двопровідну шину SDA та SCL для передачі даних, що дозволяє підключати кілька пристроїв до одного контролера. Ця бібліотека дозволяє мікроконтролеру виступати як у ролі майстра, так і у ролі слуги для комунікації I2C.

Основні функції:

- ініціалізація шини: встановлення мікроконтролера як майстра або слуги;
- запит даних від слуги: дозволяє майстру отримувати дані від периферійних пристроїв;
- початок та завершення передачі даних: керує послідовністю передачі даних між пристроями;
- запис даних у шину: надсилає байти даних на периферійні пристрої;
- читання даних з шини: отримуємо байти даних з периферійних пристроїв.

Бібліотека “LiquidCrystal_I2C.h” призначена для керування LCD-дисплеями через інтерфейс I2C. Використання даної шини для підключення дисплеїв значно зменшує кількість необхідних з’єднань, оскільки потрібно лише два дроти для передачі даних і команд. Бібліотека дозволяє легко відображати текстову інформацію на LCD-дисплеях різних розмірів.

Основні функції:

- ініціалізація дисплея та увімкнення підсвічування, що дозволяє пристрою працювати в різних умовах освітлення;
- встановлення курсору у конкретній позиції на дисплеї для точного відображення інформації;

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

- виведення тексту та числових значень на дисплей, що робить можливим динамічне оновлення інформації;

- очищення екрану та керування підсвічуванням, що забезпечує зручність використання та економію енергії.

Бібліотека “WiFi.h” є основним інструментом для підключення мікроконтролерів ESP32 або ESP8266 до мережі Wi-Fi. Бездротовий зв’язок є важливою складовою сучасних IoT-проектів, дозволяючи пристроям спілкуватися з сервером або іншими пристроями без використання проводів. Також забезпечує всі необхідні функції для налаштування та управління Wi-Fi-з’єднанням, включаючи підтримку різних типів безпеки та можливість автоматичного підключення до знайомих мереж.

Основні функції:

- підключення до мережі Wi-Fi з використанням SSID та пароллю, що дозволяє забезпечити захищене з’єднання;

- перевірка стану підключення, що дає можливість виявити проблеми з підключенням та вжити відповідні заходи;

- отримання IP-адреси та інших мережевих параметрів, необхідних для ідентифікації пристрою в мережі;

- обробку подій підключення та відключення від мережі, що дозволяє реагувати на зміни в стані з’єднання.

Бібліотека “ESPAsyncWebServer.h” дозволяє створювати асинхронні веб-сервери на базі мікроконтролерів ESP32 або ESP8266. Асинхронний підхід значно підвищує ефективність обробки HTTP-запитів, оскільки не блокує основний потік використання, дозволяючи пристрою одночасно виконувати інші завдання. Це особливо важливо для ресурсомістких або реального часу додатків [14].

Основні функції:

- створення та запуск веб-сервера, що дозволяє обслуговувати запит клієнта через браузер або інші засоби;
- обробка HTTP-запитів та формування відповідей, включаючи підтримку GET, POST, PUT, DELETE, запитів;
- підтримка обробки запитів до різних URL-адрес, що дозволяє створювати багато функціональні веб-інтерфейси для управління пристроєм;
- керування статичними файлами та сторінками, що дозволяє зберігати і віддавати HTML, CSS, JavaScript файли з флеш-пам'яті пристрою.

Бібліотека “SPIFFS.h” дозволяє використовувати SPI Flash File System для зберігання файлів у флеш-пам'яті мікроконтролерів серії ESP. SPIFFS є зручним засобом для зберігання конфігурації файлів, веб-сторінок та інших даних, які можуть бути необхідні для роботи пристрою. Вона забезпечує простий інтерфейс для роботи з файлами, дозволяючи створювати, читати, записувати і видаляти файли у файловій системі [15].

Основні функції:

- монтування файлової системи SPIFFS, що дозволяє використовувати флеш-пам'ять для зберігання даних;
- відкриття, створення, читання та запис файлів, що дозволяє працювати з даними безпосередньо з мікроконтролера;
- закриття файлів та управління файловою системою, що забезпечує надійність та ефективність.

Бібліотека ThingSpeak забезпечує інтеграцію з платформою IoT ThingSpeak, яка дозволяє зберігати, аналізувати та візуалізувати дані з сенсорів та інших пристроїв у режимі реального часу. Надає можливість віддаленого моніторингу даних через інтернет, що є критично важливим для багатьох IoT-додатків. Бібліотека дозволяє легко відправляти дані на платформу та отримувати їх назад для подальшого використання.

Основні функції:

					КС КРБ 123.128.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

- підключення до платформи з використанням API-ключів, що забезпечує безпеку і контроль доступу до даних;
- відправка даних для зберігання в канал, дозволяє вести історію вимірювань і аналізу їх у часі;
- робота з полями каналів та управління даними, дає змогу ефективно організувати і зберігати дані.

3.4 Розробка програмного забезпечення для ESP32

Підключивши всі необхідні бібліотеки для розробки програмного забезпечення, що надають певний функціонал, оголошуємо їх для роботи з ними (рис.3.5).

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include "SPIFFS.h"
#include "ThingSpeak.h"
```

Рисунок 3.5 – Лістинг оголошення бібліотек

Після оголошення бібліотек створюємо об'єкт LCD. Параметри, що надаються до конструктора визначають конфігурацію дисплея. Після чого оголошуємо константи піна мікроконтролера, до якого підключені піни для зчитування сигналу, та константи значення опорних резисторів. Змінна float використовується для зберігання значення опору, яке буде обчислене в програмі, із початковим значенням (рис.3.6).

					КС КРБ 123.128.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int analogPin1 = 32;
const int analogPin2 = 35;
const int analogPin3 = 34;
const uint8_t buttonPin = 33;
const float referenceResistancel = 680.0;
const float referenceResistance2 = 470000.0;
const float Vcc = 3.3;

float resistance = 0.0f;

```

Рисунок 3.6 – Лістинг изначення констант системи

Наступним етапом відбувається налаштування необхідних параметрів для підключення Wi-Fi мережі (рис. 3.7). Дана частина коду містить значення мережі, до якої підключається мікроконтролер. Після чого ініціалізуємо змінну паролю мережі Wi-Fi. Змінна “myChannelNumber” зберігає номер каналу на платформі ThingSpeak, до якого будуть передаватися дані. Далі описана змінна що має API ключ для запису даних у канал платформи.

```

const char* ssid = "Archangel";
const char* password = "as23qwdfzx";
unsigned long myChannelNumber = 2572154;
const char* myWriteAPIKey = "GH0XO6Z429TWSPQ6";

```

Рисунок 3.7 – Частина коду налаштування для підключення до мережі

Фрагмент коду для роботи з Wi-Fi та файловою системою SPIFFS (рис.3.8).

					КС КРБ 123.128.00.00 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if(!SPIFFS.begin(true)) {
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
}

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

```

Рисунок 3.8 – Код для роботи з Wi-Fi та SPIFFS

Процес обчислення опору невідомого резистора на основі вимірної напруги. Після цього отримані дані про опір передаються до платформи ThingSpeak. Якщо передача успішна, виводиться повідомлення про успішне надсилання даних, в іншому випадку виводиться повідомлення про помилку.

```

float calculateResistance() {
    int rawValue1 = analogRead(analogPin1);
    float voltage1 = (rawValue1 / 4095.0) * Vcc;
    float unknownResistance = referenceResistance1 * (Vcc / voltage1 - 1);
    return unknownResistance;
}

void sendToThingSpeak() {
    ThingSpeak.setField(1, resistance);
    int responseCode = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    if(responseCode == 200) {
        Serial.println("Channel update successful.");
    } else {
        Serial.println("Problem updating channel. HTTP error code " + String(responseCode));
    }
}

```

Рисунок 3.9 – Лістинг частини коду обчислення та надсилання даних

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

3.5 Розробка програмного забезпечення для сайту

Напишемо програмний код для вебсторінки, який буде містити в собі функції для взаємодії з сервером та оновлення інформації на сторінці.

```
<!DOCTYPE html>
<html>
<head>
  <title>ESP32 Web Server</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <link rel="icon" href="data:,">
  <link rel="stylesheet" type="text/css" href="style.css">
```

Рисунок 3.10 – Лістинг частини коду заголовку HTML

Даний HTML код починається з оголошення, що вказує на використання HTML5. Теги “<html>”, “<head>” та “<title>” відкривають документ, що містить метадані та встановлює заголовок сторінки. Також додає налаштування для коректного відображення веб сторінки за допомогою “<meta>”.

Напишемо скрипт для забезпечення динамічної взаємодії ESP32 з вебсервером. Він містить функції для перемикання стану кнопки, оновлення зовнішнього вигляду кнопки, оновлення поточного стану кнопки при завантаженні сторінки, також має функцію періодичного оновлення значення що отримуються. Дані функції використовують “fetch” для отримання даних з серверних маршрутів та оновлення відповідних елементів, що знаходяться на сторінці. Лістинг програмного коду скрипта зображено на рис. 3.11.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

```

<script>
  function toggleButton() {
    fetch('/toggle')
      .then(response => response.text())
      .then(state => updateButtonAppearance(state));
  }

  function updateButtonAppearance(state) {
    var button = document.getElementById('toggleButton');
    if (state === 'ON') {
      button.classList.remove('button2');
      button.classList.add('button');
      button.textContent = "ON";
    } else {
      button.classList.remove('button');
      button.classList.add('button2');
      button.textContent = "OFF";
    }
  }

  function updateButtonState() {
    fetch('/state')
      .then(response => response.text())
      .then(state => updateButtonAppearance(state));
  }

  window.onload = updateButtonState;
  function updateResistance() {
    fetch('/resistance')
      .then(response => response.text())
      .then(resistance =>
document.getElementById('resistanceValue').textContent =
resistance);
  }
  window.onload = () => {setInterval(updateResistance, 100)};
</script>

```

Рисунок 3.11 – Лістинг програмного коду скрипта

На рис. 3.12 зображено лістинг частини коду HTML сторінки з інтерактивними елементами.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

```

</head>
<body>
<button id="toggleButton" class="button"
onclick="toggleButton()">ON</button>
<h1>Test Components</h1>
<p>Resistance: <strong
id="resistanceValue">%RESISTANCE%</strong> ohms</p>
<iframe
src="https://thingspeak.com/channels/2572154/charts/1?bgcolor=%2
3ffffff&color=%23d62020&dynamic=true&results=60&type=line&update
=15" height="250" width="450" title="graph"></iframe>
</body>
</html>

```

Рисунок 3.12 – Лістинг частини коду HTML сторінки з інтерактивними елементами

Наведений код у лістингу створює вебсторінку з кнопкою стану безпосередньо для перемикання стану, також для відображення значення вимірювання та вбудованим графіком з сервісною платформою ThingSpeak. Стан кнопки змінюється при натисканні, а значення що тримуються в результаті тестування, оновлюються динамічно за допомогою JavaScript.

Для оформлення вебсторінки та її елементів, напишемо CSS код. Він стилізує вебсторінку додаючи на неї заднє зображення, вирівнювання вмісту по центру та зверху. Встановлення кольору для заголовків та додаткового тексту, кнопка що змінює свій колір згідно стану, та її розміщення на сторінці.

Приклад лістингу CSS коду зображено на рис. 3.13.

					КС КРБ 123.128.00.00 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		


```

body {
margin: 0;
padding: 0;
display: flex;
flex-direction: column;
align-items: center;
justify-content: flex-start;
height: 100vh;
background: url("https://backgrounds.jpg") no-repeat center
center fixed;
background-size: cover;
}
h1 {
margin-top: 50px;
text-align: center;
color: white;
}
p {
color: white;
}
.button {
background-color: blue;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border: none;
cursor: pointer;
position: absolute;
top: 10px;
left: 10px;
}
.button2 {
background-color: red;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border: none;
cursor: pointer;
position: absolute;
top: 10px;
left: 10px;
}

```

Рисунок 3.13 – Лістинг CSS коду

3.6 Тестування системи

Під’єднаємо систему до комп’ютера, у середовищі Arduino IDE. У меню інструментів обираємо порт: COM3, вивантажуємо код програми до мікроконтролера. Після завантаження програмного коду відбувається

					КС КРБ 123.128.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

ввімкнення LCD-дисплея, на нього виводить назва створеної системи, зображено на рис. 3.14.



Рисунок 3.14 – Інформація при ввімкненні системи

Наступним етапом у корінній папці Arduino IDE, де розміщений програмний код проекту, створюємо папку із назвою data. У створену папку додаємо файли конфігурації вебсайту для подальшої роботи із ним (рис 3.15).

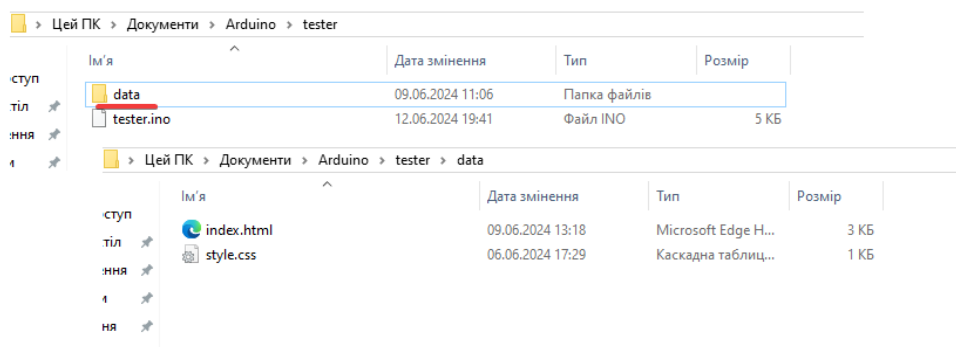


Рисунок 3.15 – Файли для розгортання вебсайту

Відкриваємо меню інструментів та натискаємо на пункт “ESP32 Sketch Data Upload” [16]. У вікні консолі відображається інформація про процесу прошивки плати мікроконтролера, завантаження файлів, параметри які використовуються для підключення до плати та процесу прошивки (рис.3.16).

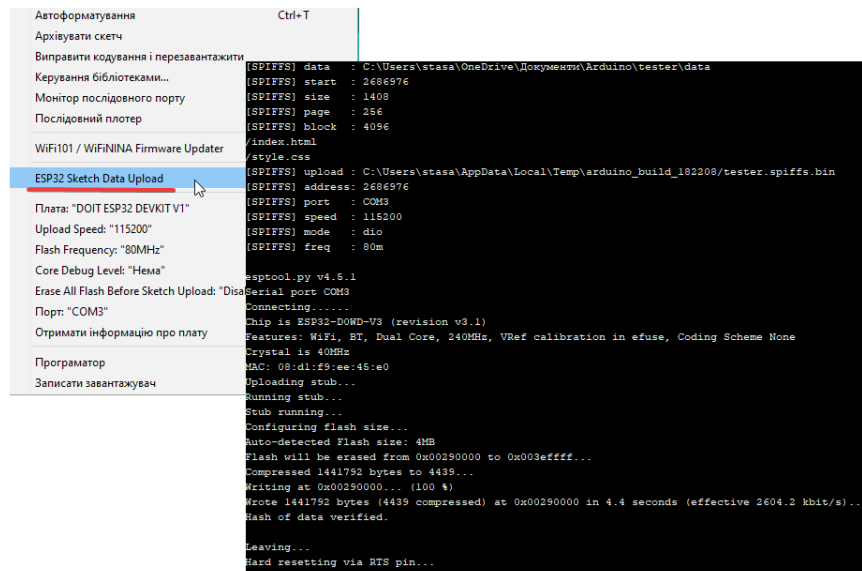


Рисунок 3.16 – Завантаження файлів до мікроконтролера

Відкриваємо серійний монітор у програмному середовищі, система має автоматично підключитися до Wi-Fi. Якщо у серійному моніторі нічого не відобразилось натискаємо на мікроконтролері кнопку RESET/EN. Після цього у серійному моніторі виводиться ір вебсервера, вводимо його у браузер та відкриваємо вебсторінку на якій відображається стан системи та результати вимірювання, зображено на рис. 3.17 та 3.18.

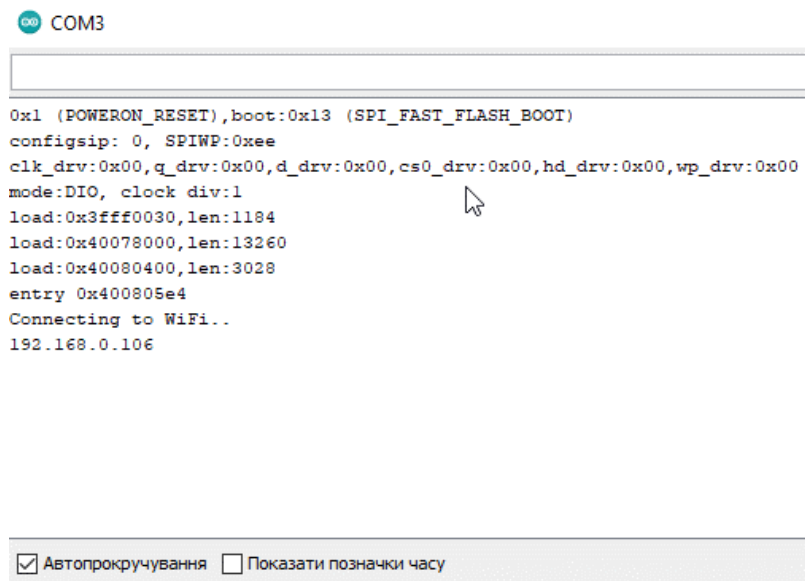


Рисунок 3.17 – Вивід ір у серійний монітор

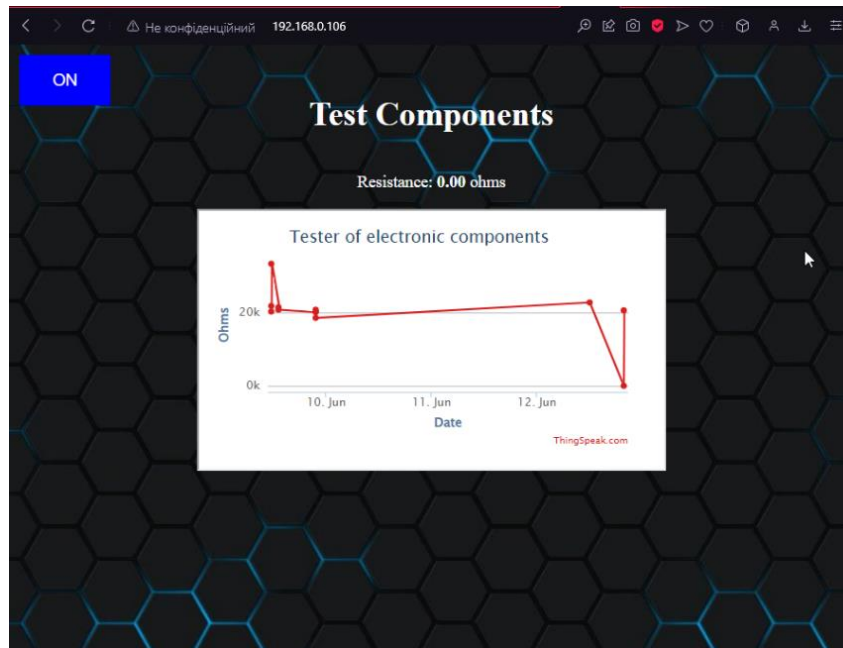


Рисунок 3.18 – Вебсторінка для відображення тестування

Вставимо резистор у роз’єм для тестування, після цього натискаємо на кнопку (рис. 3.19).

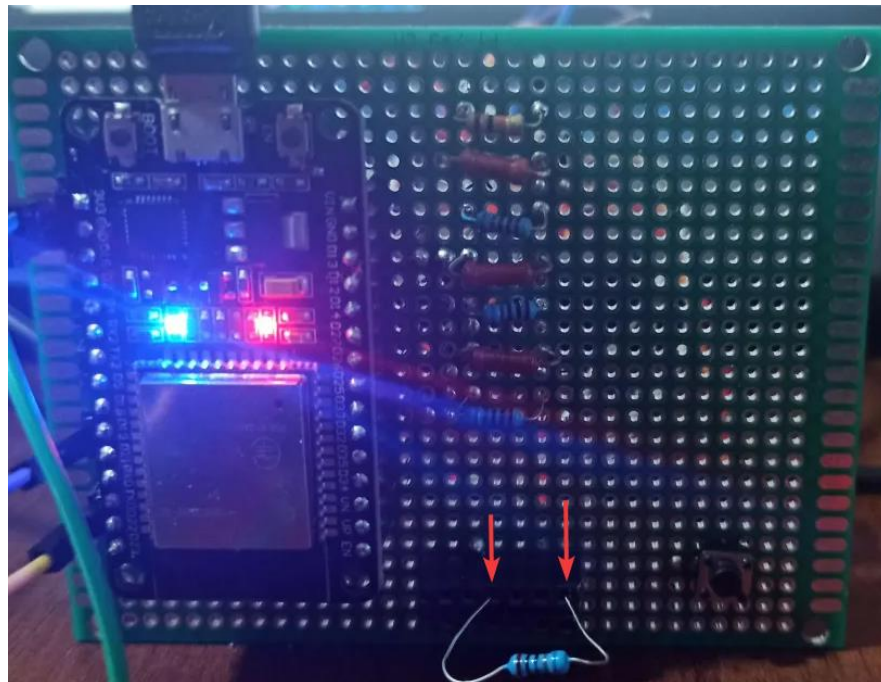


Рисунок 3.19 – Процес тестування

На LCD дисплей виводиться інформація результату тестування (рис. 3.20).



Рисунок 3.20 – Вивід інформації на дисплей

Паралельно із цим на вебсторінку надсилається результат тестування, він зберігається у ThingSpeak та записується у графік (рис. 3.21).

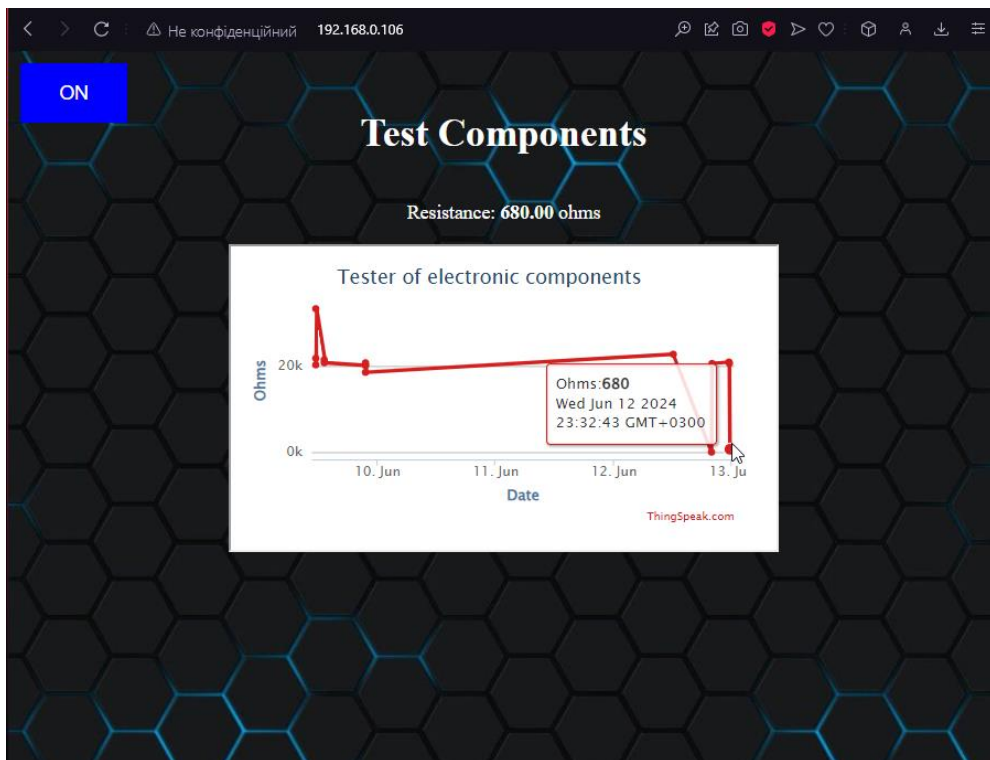


Рисунок 3.21 – Виведення значення у графік

Під час тестування системи, після вставлення резистора у роз'єм та натискання кнопки, на дисплей було виведено результат тестування. Паралельно з цим результатом тестування значення автоматично відправляються на вебсторінку, де зберігаються на платформі ThingSpeak. На вебсторінці результат тестування відобразився у вигляді графіка, що забезпечило зручний аналіз даних. Це дозволяє спостерігати за історією тестування та виявляти тенденції й аномалії в роботі системи.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1. Шляхи підвищення життєдіяльності людини

Життєдіяльність людини охоплює всі аспекти її існування, фізичне, психологічне та соціальне благополуччя. Підвищення життєдіяльності є багатогранним завданням, поліпшення умов праці, створення безпечного середовища, здорового способу життя та психологічної підтримки. У контексті комп'ютеризованих систем контролю якості електронних компонентів, підвищення життєдіяльності працівників є особливо важливим, оскільки ця сфера має свої специфічні ризики та виклики.

Основні шляхи підвищення життєдіяльності людини:

а) поліпшення умов праці:

- організація робочих місць з урахуванням ергономічних вимог знижує ризик професійних захворювань і травм. Це включає правильне розташування обладнання, зручні крісла, оптимальне освітлення та забезпечення відповідного рівня шуму;

- впровадження комп'ютеризованих систем знижує фізичне навантаження на працівників, зменшує ймовірність помилок і підвищує продуктивність праці. Автоматизовані системи контролю якості використовуються для того, щоб мінімізувати контакт людини з небезпечними матеріалами та умовами;

б) забезпечення безпеки на робочому місці:

- регулярні навчання з техніки безпеки, проведення інструктажів з використання обладнання та засобів індивідуального захисту допомагають працівникам уникати нещасних випадків;

					КС КРБ 123.128.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>				
<i>Розроб.</i>		<i>Панасенко С.М.</i>			<i>РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Паляниця Ю.Б.</i>					53	5
<i>Консульт.</i>		<i>Пилипець М.І.</i>				<i>ТНТУ, каф. КС, гр. СІ-42</i>		
<i>Н. контр.</i>		<i>Тиж Є.В.</i>						
<i>Затверд.</i>		<i>Осухівська Г.М.</i>						

- впровадження систем моніторингу для виявлення та попередження потенційно небезпечних ситуацій на ранніх стадіях. Для цього використовуються датчики, що контролюють рівень небезпечних речовин, системи відеоспостереження, автоматичні сигналізації тощо;

в) підтримка фізичного та психологічного здоров'я:

- регулярні медичні огляди, доступ до медичної допомоги та профілактичних заходів сприяють збереженню здоров'я працівників;

- створення служб психологічної допомоги та підтримки, організація тренінгів з управління стресом і подолання конфліктів на роботі;

г) заходи по покращенню соціального благополуччя:

- впровадження гнучкого графіка роботи, можливість дистанційної роботи, забезпечення регулярних перерв і відпусток;

- організація заходів для підвищення командного духу, корпоративні заходи, спортивні змагання, забезпечення соціальних пакетів, включаючи страхування та пільги;

Підвищення життєдіяльності людини є ключовим елементом забезпечення ефективної та безпечної праці, особливо у високотехнологічних галузях, таких як контроль якості електронних компонентів. Комплексний підхід, що включає поліпшення умов праці, забезпечення безпеки, підтримку фізичного та психологічного здоров'я, а також соціальне благополуччя, сприяє створенню гармонійного та продуктивного робочого середовища. Впровадження сучасних комп'ютеризованих систем контролю якості не тільки підвищує ефективність виробничих процесів, але й сприяє захисту здоров'я і добробуту працівників, що в кінцевому підсумку веде до підвищення загальної життєдіяльності людини.

					КС КРБ 123.128.00.00 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2 Заходи щодо захисту системи контролю якості електронних компонентів від короткого замикання.

Коротке замикання є однією з найпоширеніших причин виходу з ладу електронних систем, зокрема систем контролю якості електронних компонентів. Воно призводить до серйозних пошкоджень обладнання, втрати даних і спричиняє пожежу. Тому забезпечення захисту від короткого замикання є критично важливим завданням для будь-якої системи, яка працює з електричними компонентами. Дані заходи допоможуть захистити комп'ютеризовану систему контролю якості електронних компонентів від короткого замикання.

Причини короткого замикання:

- погана ізоляція проводів, ушкоджені або зношені дроти призводять до контакту між провідниками різних потенціалів;
- несправні, дефектні або пошкоджені електронні компоненти спричиняють коротке замикання;
- неправильне підключення, помилки при монтажі або обслуговуванні електричних схем спричиняють коротке замикання;
- скупчення пилу та вологи спричиняє утворенню провідного мосту між провідниками.

Заходи захисту від короткого замикання:

а) одним із найефективніших методів захисту від короткого замикання є використання запобіжників і автоматичних вимикачів. Ці пристрої автоматично відключають живлення у разі виникнення короткого замикання, що запобігає подальшому пошкодженню системи;

б) якісна ізоляція проводів і правильне їхнє прокладання є ключовими факторами для запобігання короткому замиканню. Використання високоякісних матеріалів для ізоляції та дотримання стандартів при монтажі допоможе знизити ризик пошкодження проводів. Варто уникати прокладання

					КС КРБ 123.128.00.00 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

проводів у місцях з підвищеною вологістю, місць які містять механічний вплив;

в) регулярні перевірки та технічне обслуговування системи дозволяє вчасно виявити потенційні проблеми, зношення ізоляції, пошкодження компонентів або накопичення пилу. Це дозволяє запобігти короткому замиканню до того, як воно спричинить серйозні проблеми. Важливо перевіряти стан роз'ємів і контактних з'єднань, які часто стають причиною коротких замикань;

г) перенапруга в мережі спричиняє коротке замикання. Для захисту від перенапруги використовують спеціальні пристрої, обмежувачі перенапруги або стабілізатори напруги. Вони захищають систему від різких змін напруги, які шкодять компоненти або призводять до короткого замикання;

г) якість використовуваних електронних компонентів безпосередньо впливає на надійність системи. Використання сертифікованих і перевірених компонентів знижує ризик виникнення коротких замикань. Важливо перевіряти компоненти перед їхнім встановленням у систему, для виявлення зовнішніх дефектів.

Захист комп'ютеризованої системи контролю якості електронних компонентів від короткого замикання є складним, але необхідним процесом. Відповідний підбір і установка захисних пристроїв, якісна ізоляція та регулярне обслуговування системи допомагає запобігти коротким замиканням і забезпечити безперебійну роботу системи. Виконання цих заходів знижує ризик пошкодження обладнання, та підвищує безпеку експлуатації системи загалом.

4.3 Долікарська допомога при переломах

Перелом – це серйозне порушення цілісності кістки, що виникає через травму, удар або падіння. Їх поділяють на закритий, коли шкіра не

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

пошкоджена, або відкритий, коли кістка проникає через порушену шкіру. Ознаки перелому включають постійний або збільшений біль, особливо під час навантаження на ушкоджену ділянку або при дотику до неї, обмеження рухів в пошкодженій області, зміну форми тіла в місці перелому, крововиливи та ненормальну рухливість кістки.

Загальний стан потерпілого залежить від тяжкості перелому. Температура тіла може підвищуватися, особливо у випадках переломів кісток черепа, таза або стегна. Важливо пам'ятати, що деякі з ознак перелому є неочевидними. Тому, якщо є підозра на перелом, слід трактувати травму як перелом і надавати відповідну допомогу.

Перша допомога полягає в забезпеченні спокою пошкодженої ділянки тіла та іммобілізації уламків кісток. Це можна зробити за допомогою утримуючих пов'язок або транспортних шин. Шини накладаються на оголене тіло або поверх одягу, забезпечуючи нерухомість уламків.

Основне правило іммобілізації – шина повинна охоплювати не менше ніж два суглоби нижче та вище від місця перелому. У випадку відкритого перелому перед накладанням шини накладаємо стерильну пов'язку.

Деякі переломи, такі як переломи ключиці, ребер, хребта або тазу, вимагають особливого підходу до надання допомоги. При переломі ключиці необхідно використовувати великий жмут вати, а руку прибинтовувати до тулуба. При переломі ребер може знадобитись туга пов'язка навколо грудної клітки.

Переломи хребта особливо небезпечні через можливість ураження спинного мозку. Необхідно забезпечити нерухомість хребта та обережно транспортувати потерпілого.

Переломи кісток тазу можуть супроводжуватися серйозним загальним станом, тому необхідно негайно доставити потерпілого до медичного закладу.

					КС КРБ 123.128.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВОК

Підсумовуючи результати створення комп'ютеризованої системи контролю якості електронних компонентів, можна стверджувати, що розроблена система значно покращує ефективність і точність процесу перевірки електронних компонентів.

Основними досягненнями створеної системи можна вважати точність вимірювання параметрів різних електронних компонентів, що забезпечує високу надійність отриманих результатів під час тестування, які є критично важливими для забезпечення якості кінцевих продуктів. Програмне забезпечення для управління тестером та обробки даних забезпечує простоту у використанні та дозволяє швидко аналізувати отримані результати, виявляти дефекти.

Розроблена комп'ютеризована система є ефективною для використання як у лабораторних умовах, так і у виробничих процесах. Впровадження цієї системи сприяє підвищенню надійності та якості електронних пристроїв.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. E – Learning TNTU. URL: <https://dl.tntu.edu.ua/> (дата звернення 24.04.24).
2. 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.
3. Лещишин Ю.З., Міська І.В., Назаревич Т.О. Структурне моделювання цифрових каналів зв'язку. Матеріали VIII науково-технічної конференції „Інформаційні моделі, системи та технології“. ТНТУ, 2020. С. 126.
4. Оконський М.В., Лупенко С.А., Паламар А.М. Комп'ютерна система для моніторингу метеорологічних параметрів на основі IoT. Актуальні задачі сучасних технологій: збірник тез доповідей X міжнародної науково-технічної конференції молодих учених та студентів. Тернопіль: ТНТУ. 2021. С. 112.
5. Паламар А.М. Комп'ютерна система для моніторингу параметрів джерел безперебійного живлення на основі технології Internet of Things. Матеріали IV Міжнародної науково-технічної конференції «Теоретичні та прикладні аспекти радіотехніки, приладобудування і комп'ютерних технологій», Тернопіль. 2019. С. 208–209.
6. Волоський В.П., Лещишин Ю.З., Романишин Н.Р. Комп'ютерна система контролю та балансування літій-іонних акумуляторних батарей, Тернопіль 2021. Том I. С. 87–88.
7. Осухівська Г.М., Волощук А.В. Технології передавання та опрацювання даних в комп'ютеризованих системах обліку електроенергії. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології“, 7–8 грудня 2022 року. Т. : ТНТУ, 2022. С. 136.

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

8. Guide to Testing Electronic Components Effectively. URL: <https://kunkune.co.uk/blog/testing-electronic-components/> (дата звернення 29.04.24).

9. Arduino UA плата для розробки ESP32. URL: <https://arduino.ua/prod3990-wi-fi-modyl-devkit-v1-s-esp-32> (дата звернення 01.05.24).

10. ESP32 DevKit v1. URL: <https://www.circuitstate.com/pinouts/doi-esp32-devkit-v1-wifi-development-board-pinout-diagram-and-reference/> (дата звернення 04.05.24).

11. Electronic Components Datasheet ESP32. URL: https://www.alldatasheet.com/view.jsp?Searchword=Esp32%20datasheet&gad_source=1&gclid=CjwKCAjwmrqzBhAoEiwAXVpgoiyM_W5EHygiETrNeUrsLbr5UOwtXsQlIZsz9ZH7eDA3IEVm-ITqgxoCgYgQAvD_BwE (дата звернення 12.05.24).

12. Arduino UA LCD-дисплей. URL: <https://arduino.ua/prod6188-lcd-1602-i2c-simvolnii-displei-16x2-sinii> (дата звернення 19.05.24).

13. LCD I2C 1602. URL: https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/21-component_i2c_lcd1602.html (дата звернення 26.05.24)

14. ESP32 Web Server. URL: <https://randomnerdtutorials.com/esp32-web-server-gauges/> (дата звернення 02.06.24).

15. SPIFFS Filesystem. URL: <https://docs.espressif.com/projects/espressif-esp-idf/en/stable/esp32/api-reference/storage/spiffs.html> (дата звернення 05.06.24).

16. ESP32 Filesystem Uploader. URL: <https://randomnerdtutorials.com/install-esp32-file-system-uploader-arduino-ide/> (дата звернення 09.06.24)

					КС КРБ 123.128.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

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.

"09.02" _____ 2024 р.

КОМП'ЮТЕРИЗОВАНА СИСТЕМА КОНТРОЛЮ ЯКОСТІ ЕЛЕКТРОННИХ
КОМПОНЕНТІВ

ТЕХНІЧНЕ ЗАВДАННЯ

на 4 листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____к.т.н., ст. викл. Паляниця Ю.Б.

«08.02» _____ 2024 р.

«ВИКОНАВЕЦЬ»

Студент групи СІ-42

_____Панасенко С.М.

«08.02» _____ 2024 р.

Тернопіль 2024

1. Повна назва та її умовне позначення.

Повна назва теми кваліфікаційної роботи «Комп'ютеризована система контролю якості електронних компонентів».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.128.00.00

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, Панасенко Станіслав Михайлович.

3. Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-408 від 24.04.2024 р.)

4. Планові терміни початку та завершення роботи

Плановий термін початку виконання роботи - 24.04.2024 р.

Плановий термін завершення виконання роботи - 28.06.2024 р.

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

5. Призначення і цілі створення системи

Система що розробляється призначена для контролю якості електронних компонентів, що основана на мікроконтролері ESP32.

6. Мета створення системи

Метою кваліфікаційної роботи є розробка система що вираховує значення електронних компонентів, передаючи отримані дані на вебсервер для їх зберігання та подальшого аналізу.

7. До складу виробу повинні входити:

а) Плата ESP32 DevKit v1;

б) Дисплей LCD I2C.

в) Резистори номіналом 680Ом, 470КОм, 10КОм.

г) Тактова кнопка

8. Конструктивні вимоги

Конструювання корпусу приладу системи контролю якості електронних компонентів не передбачено. Для побудови системи використана сучасна компонентна база.

9. Техніко-економічні показники

Планова собівартість проекту повинна становити не більше 2000 тисяч гривень.

10. Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної плати

№	Назва етапів роботи	Термін виконання етапів роботи
1	Затвердження технічного завдання	01.02.24-09.02.24
2	Аналіз технічного завдання	05.02.24-11.02.24
3	Розробка структурної та функціональної схеми	25.04.24-02.05.24
4	Вибір елементної бази	03.05.24-10.05.24
5	Вибір програмного забезпечення	10.05.24-14.05.24
6	Реалізація проектної складової	14.05.24-09.06.24
7	Опрацювання розділу «Безпека життєдіяльності, основи охорони праці»	10.06.24-15.06.24
8	Оформлення пояснювальної записки кваліфікаційної роботи	16.06.24-20.06.24
9	Оформлення графічної частини	14.06.24
10	Підготовка до захисту	24.06.24-28.06.24
11	Захист кваліфікаційної роботи	28.06.24

11. Під час виконання кваліфікаційної роботи у технічне завдання можуть вноситись зміни та доповнення

Додаток Б
Перелік елементів

Позначення	Найменування	Кіл.	Примітка
	Резистори		
R1, R3, R5	Резистор 470кОм	3	
R2, R4, R6	Резистор 680Ом	3	
R7	Резистор 10кОм	1	
	Кнопка		
SB1	Тактова кнопка	1	
	Модулі		
U1	ESP32 DevKit v1	1	
U2	LCD I2C дисплей	1	
	Роз'єми		
XS1	798-ST60-24P30 2-pin PCB socket	1	
XS2	Роз'єм коннектор 3pin	1	

					КС КРБ 123.128.00.00 ПЕ					
Змн.	Арк.	№ докум.	Підпис	Дата				Літ.	Арк.	Акрушів
Розроб.		Панасенко С.М.			<i>Комп'ютеризована система контролю якості електронних компонентів Перелік елементів</i>				1	1
Перевір.		Паляниця Ю.Б.								
Реценз.		Литвиненко Я.В.								
Н. контр.		Тиш Є.В						<i>ТНТУ, каф. КС, гр. СІ-42</i>		
Затверд.		Осухівська Г.М.								

Додаток В

Лістинг програми

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include "SPIFFS.h"
#include "ThingSpeak.h"
LiquidCrystal_I2C lcd(0x27, 16, 2);
const int analogPin1 = 32;
const int analogPin2 = 35;
const int analogPin3 = 34;
const uint8_t buttonPin = 33;
const float referenceResistance1 = 680.0;
const float referenceResistance2 = 470000.0;
const float Vcc = 3.3;
float resistance = 0.0f;
const char* ssid = "Archangel";
const char* password = "as23qwdfzx";
unsigned long myChannelNumber = 2572154;
const char* myWriteAPIKey = "GH0X06Z429TWSPQ6";
const int ledPin = 2;
String ledState = "OFF";
AsyncWebServer server(80);
float lastResistance = 0.0;
WiFiClient client;
String processor(const String& var){
  if(var == "STATE"){
    if(digitalRead(ledPin)){
      ledState = "ON";
    }
    else{
      ledState = "OFF";
    }
    return ledState;
  } else if (var == "RESISTANCE"){
    return String(lastResistance);
  }
  return String();
}
void setup() {
  lcd.init();
  lcd.backlight();
  Serial.begin(115200);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Tester");
  lcd.setCursor(0, 1);
```

```

lcd.print("components");
delay(3000);
lcd.clear();
if(!SPIFFS.begin(true)){
Serial.println("An Error has occurred while mounting SPIFFS");
return;
}
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
Serial.println("Connecting to WiFi..");
}
Serial.println(WiFi.localIP());
ThingSpeak.begin(client);
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
request->send(SPIFFS, "/index.html", String(), false, processor);
});

server.on("/style.css", HTTP_GET, [](AsyncWebServerRequest
*request){
request->send(SPIFFS, "/style.css", "text/css");
});
server.on("/toggle", HTTP_GET, [](AsyncWebServerRequest *request){
digitalWrite(ledPin, !digitalRead(ledPin));
request->send(200, "text/plain", digitalRead(ledPin) ? "ON" :
"OFF");
});
server.on("/state", HTTP_GET, [](AsyncWebServerRequest *request){
String state = digitalRead(ledPin) ? "ON" : "OFF";
request->send(200, "text/plain", state);
});
server.on("/resistance", HTTP_GET, [](AsyncWebServerRequest
*request){
request->send(200, "text/plain", String(resistance));
});
server.begin();
}
void loop() {
if (digitalRead(buttonPin) == LOW) {
measureAndDisplayResistance();
delay(200);
}
}
void measureAndDisplayResistance() {
resistance = calculateResistance();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Resistance: ");
lcd.setCursor(0, 1);
lcd.print(resistance);
lcd.print(" ohms");
Serial.print("Resistance: ");
Serial.print(resistance);
Serial.println(" ohms");
}

```

```

sendToThingSpeak();
}
float calculateResistance() {
int rawValue1 = analogRead(analogPin1);
float voltage1 = (rawValue1 / 4095.0) * Vcc;
float unknownResistance = referenceResistance1 * (Vcc / voltage1 -
1);
return unknownResistance;
}
void sendToThingSpeak() {
ThingSpeak.setField(1, resistance);
int responseCode = ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);
if(responseCode == 200) {
Serial.println("Channel update successful.");
} else {
Serial.println("Problem updating channel. HTTP error code " +
String(responseCode));
}
}
}

```


Додаток Д

Лістинг сайту

Index.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>ESP32 Web Server</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <link rel="icon" href="data:,">
  <link rel="stylesheet" type="text/css" href="style.css">
  <script>
    function toggleButton() {
      fetch('/toggle')
        .then(response => response.text())
        .then(state => updateButtonAppearance(state));
    }
    function updateButtonAppearance(state) {
      var button = document.getElementById('toggleButton');
      if (state === 'ON') {
        button.classList.remove('button2');
        button.classList.add('button');
        button.textContent = "ON";
      } else {
        button.classList.remove('button');
        button.classList.add('button2');
        button.textContent = "OFF";
      }
    }
    function updateButtonState() {
      fetch('/state')
        .then(response => response.text())
        .then(state => updateButtonAppearance(state));
    }
    window.onload = updateButtonState;
    function updateResistance() {
      fetch('/resistance')
        .then(response => response.text())
        .then(resistance =>
document.getElementById('resistanceValue').textContent =
resistance);
    }
    window.onload = () => {setInterval(updateResistance, 100)};
  </script>
</head>
<body>
<button id="toggleButton" class="button"
onclick="toggleButton()">ON</button>
<h1>Test Components</h1>
<p>Resistance: <strong id="resistanceValue">%RESISTANCE%</strong>
ohms</p>
```

```
<iframe
src="https://thingspeak.com/channels/2572154/charts/1?bgcolor=%23f
fffff&color=%23d62020&dynamic=true&results=60&type=line&update=15"
height="250" width="450" title="graph"></iframe>
</body>
</html>
```

Style.css:

```
body {
  margin: 0;
  padding: 0;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: flex-start;
  height: 100vh;
  background: url("https://background.jpg");
  background-size: cover;
}
h1 {
  margin-top: 50px;
  text-align: center;
  color: white;
}
.button {
  background-color: blue;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  border: none;
  cursor: pointer;
  position: absolute;
  top: 10px;
  left: 10px;
}
.button2 {
  background-color: red;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  border: none;
  cursor: pointer;
  position: absolute;
  top: 10px;
  left: 10px;
}
```