

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Система тестування GLSL коду на базі Raspberry PI

Виконав(ла): студент(ка) IV курсу, групи CI-41

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Музичук А.А.

(прізвище та ініціали)

Керівник

(підпис)

Луцик Н.С.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тиш Є.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Мудрик І. Я.

(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 » 06 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 Музичук Андрій Андрійович
(прізвище, ім'я, по батькові)

1. Тема роботи Система тестування GLSL коду на базі Raspberry PI

Керівник роботи Луцик Надія Степанівна

(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 28.06.2024

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 електрично принципова

Схема алгоритму програми

Схема діаграми послідовностей

Схема підключень

АНОТАЦІЯ

Система тестування GLSL коду на базі Raspberry PI// Кваліфікаційна робота бакалавра // Музичук Андрій Андрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІ-41 // Тернопіль, 2024 // с.– 55 , рис. – 22 , додат. – 2, бібліогр. – 17.

Ключові слова: комп'ютерна система, Raspberry PI, OpenGL Shader Language, Веб-інтерфейс, тестування коду.

Метою даної кваліфікаційної роботи бакалавра є проектування та реалізація системи тестування GLSL коду на базі Raspberry PI.

У кваліфікаційній роботі бакалавра спроектовано архітектуру системи тестування GLSL коду на базі Raspberry PI, розроблено програмне забезпечення, веб-інтерфейс для реалізації необхідного функціоналу.

Налаштовано операційну систему Raspberry PI OS, встановлено необхідні бібліотеки і залежності, що дало можливість відправляти та тестувати код через веб-інтерфейс, щоб визначати його роботоздатність і продуктивність.

ABSTRACT

Raspberry PI-based GLSL code testing system // Bachelor's thesis // Muzychuk Andriy // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Systems and Networks, group CI-41 // Ternopil, 2024 // p. – 55, fig. – 22, , bibliog. – 17.

Keywords: computer system, Raspberry PI, OpenGL Shader Language, web interface, code testing.

The aim of this bachelor's qualification work is the design and implementation of a GLSL code testing system based on Raspberry PI.

In the bachelor's qualification work, the architecture of the GLSL code testing system based on Raspberry PI is designed, and software and a web interface for implementing the necessary functionality are developed.

The Raspberry PI OS operating system has been configured, and the necessary libraries and dependencies have been installed, allowing code to be sent and tested through the web interface to determine its functionality and performance.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	9
1.1 Аналіз вимог до комп'ютерної системи	9
1.2 Аналіз можливих рішень поставленого завдання.....	11
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА.....	15
2.1 Розробка узагальненої структури комп'ютерної системи	15
2.2 Обґрунтування вибору апаратного забезпечення	17
2.3 Обґрунтування вибору програмного забезпечення проєктованого комп'ютерного засобу	19
2.4 Проєктування комп'ютерної системи	23
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА.....	29
3.1 Реалізація проєктних рішень.....	29
3.2 Тестування	43
БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	48
4.1 Долікарська допомога при ураженні електричним струмом.....	48
4.2 Вплив електромагнітних полів (ЕМП) на людину та заходи щодо зменшення їх впливу на обслуговуючий персонал.....	48
4.3 Заходи захисту обладнання від статичної електрики.....	48
ВИСНОВКИ.....	53
ПЕРЕЛІК ПОСИЛАНЬ	54
ДОДАТКИ	
Додаток А. Технічне завдання	
Додаток Б. Перелік елементів	

					КС КРБ 123.122.00.00 ПЗ			
Зм.	Аркуш	№ докум.	Підпис	Дата	<i>Система тестування GLSL коду на базі Raspberry PI</i>	Літ.	Аркуш	Аркушів
Розробив	Музичук А.						6	55
Перевірила	Луцик Н.С.					ТНТУ, каф. КС,		
Реценз.	Мудрик І.Я.					гр. СІ-41		
Н. контр.	Тиш Є.В.							
Затверд.	Осухівська Г.М.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

GLSL – OpenGL Shading Language

ПЗ – Програмне забезпечення

PSB – Printed circuit board

SPI – Serial Peripheral Interface

GLM – OpenGL Mathematics

GLFW – Graphics Library Framework

SSH – Secure Shell

					КС КРБ 123.122.00.00 ПЗ	Акр.
						7
Зм.	Аркуш	№ докум.	Підпис	Дата		

ВСТУП

Розробка та тестування графічних шейдерів у середовищі сучасних обчислювальних систем є важливим етапом у створенні високоякісного візуального контенту для ігор, анімацій та віртуальної реальності. GLSL [1] — це одна з найпопулярніших мов програмування шейдерів, яка використовується для опису поведінки графічних об'єктів у тривимірному просторі. Однак, тестування GLSL коду може бути викликом, особливо для розробників-початківців та маленьких студій, які не мають доступу до потужного обладнання.

Raspberry Pi, будучи доступним та компактним комп'ютером, здатним виконувати широкий спектр завдань, відкриває нові можливості для тестування графічних шейдерів у різних умовах. Використання Raspberry Pi як платформи для розробки та тестування GLSL коду дозволяє створити бюджетну та портативну систему, що є особливо актуальним у контексті освіти та малих підприємств.

Наявність системи тестування GLSL коду на базі Raspberry Pi надає низку важливих переваг. По-перше, це значно знижує вартість тестування, що є критичним для стартапів та невеликих команд розробників. По-друге, компактні розміри та низьке енергоспоживання Raspberry Pi роблять його ідеальним інструментом для мобільного тестування та навчальних лабораторій. Крім того, використання такої системи може сприяти підвищенню рівня обізнаності та навичок студентів у роботі з графічними шейдерами.

Комп'ютерна система тестування GLSL коду на базі Raspberry Pi також має велике значення у контексті популяризації та доступності сучасних технологій. Він дозволяє розширити можливості для творчості та інновацій, надаючи розробникам інструмент, який дозволяє швидко перевіряти і вдосконалювати свої ідеї. Крім того, ця система може слугувати основою для подальших досліджень та розробок у галузі графічного програмування та оптимізації продуктивності обчислювальних систем.

					КС КРБ 123.122.00.00 ПЗ	Акр.
						8
Зм.	Аркуш	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютерної системи

Система тестування GLSL коду на базі Raspberry Pi відкриває широкі можливості для різноманітних застосувань. Вона може стати незамінним інструментом для навчальних закладів, де студенти вивчають основи комп'ютерної графіки та програмування шейдерів. Завдяки доступності та простоті використання, Raspberry Pi дозволяє створити невеликі навчальні лабораторії, де кожен студент може працювати з власною системою, вивчаючи та експериментуючи з графічним програмуванням без значних витрат на обладнання.

У сфері малого бізнесу та стартапів така система може використовуватися для прототипування та тестування нових графічних рішень. Невеликі команди розробників можуть швидко і ефективно перевіряти свої ідеї, оптимізувати продуктивність шейдерів та відшліфувати візуальні ефекти перед їх впровадженням у кінцевий продукт. Це особливо важливо в умовах обмежених бюджетів та ресурсів, коли кожна помилка може обійтися дуже дорого [2].

Дослідники та ентузіасти також знайдуть застосування такій системі. Вона надає можливість проведення експериментів з новими алгоритмами опрацювання зображень, розробки інноваційних методів рендерингу та вивчення поведінки графічних процесорів у різних умовах. Raspberry Pi дозволяє створювати тестові стенди для проведення таких досліджень, не витрачаючи значні кошти на спеціалізоване обладнання.

Така комп'ютерна система може бути корисною навіть у сфері мистецтва та творчості. Художники та дизайнери, які працюють з цифровими інсталяціями та інтерактивними медіа, можуть використовувати її для створення унікальних візуальних ефектів та інтерактивних проектів. Компактність та портативність Raspberry Pi дозволяє інтегрувати його у різноманітні інсталяції,

					КС КРБ 123.122.00.00 ПЗ			
Зм.	Аркуш	№ докум.	Підпис	Дата	АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	Літ.	Аркуш	Аркушів
Розробив	Музичук А.						9	6
Перевірила	Луцик Н.С.					ТНТУ, каф. КС,		
Реценз.	Мудрик І.Я.					гр. СІ-41		
Н. контр.	Тиш Є.В.							
Затверд.	Осухівська Г.М.							

забезпечуючи високу якість графіки та взаємодії з користувачами.

Для успішної розробки та реалізації системи тестування GLSL коду необхідно детально проаналізувати вимоги до комп'ютерної системи. Основними критеріями для такого аналізу є апаратні характеристики, програмне забезпечення, а також сумісність та оптимізація роботи у контексті графічного програмування.

Проаналізувавши методи і області застосування можна виділити такі ключові параметри системи:

- можливість обробляти код в реальному часі;
- веб-інтерфейс для відправлення коду на систему;
- вивід зображення на дисплей;
- зображення параметрів і швидкості обробки коду;
- сумісність зі стандартом OpenGL.

Ця функція дозволяє системі швидко реагувати на введені зміни, забезпечуючи миттєвий зворотний зв'язок для користувача. Це особливо важливо для інтерактивних застосунків, таких як розробка ігор або візуалізація даних, де затримка може серйозно вплинути на користувацький досвід. Система повинна бути оптимізована для мінімізації затримок і забезпечення стабільної продуктивності навіть під час складних обчислень.

Наявність зручного веб-інтерфейсу значно полегшує процес взаємодії з системою. Користувачі можуть відправляти свої скрипти, отримувати результати і переглядати статус виконання завдань через інтуїтивно зрозумілий графічний інтерфейс. Веб-інтерфейс також дозволяє інтеграцію з іншими онлайн-сервісами та інструментами, розширюючи можливості системи.

Ця функція є критичною для додатків, які включають візуалізацію даних, графіку або будь-які інші форми візуального представлення інформації. Система повинна підтримувати високу роздільну здатність і забезпечувати чіткість зображень. Важливо, щоб зображення відображались без затримок, що дозволяє користувачам оперативно реагувати на отримані дані.

Інформація про параметри та швидкість обробки коду дозволяє користувачам оцінювати продуктивність системи та ефективність своїх алгоритмів. Ця функція може включати графічне відображення метрик, таких як час виконання,

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		10

використання ресурсів, кількість оброблених операцій тощо. Наявність таких даних допомагає виявляти "вузькі місця" та оптимізувати процеси.

OpenGL є одним із найбільш розповсюджених стандартів для роботи з графікою. Сумісність із цим стандартом дозволяє розробникам використовувати вже наявні бібліотеки та інструменти, що значно спрощує процес створення та інтеграції графічних компонентів. Вона також забезпечує високий рівень кросплатформеності, що дозволяє запускати розроблені програми на різних операційних системах без значних змін у коді.

Отже, необхідно розробити систему яка дозволить користувачу відправляти код на Raspberry Pi і перевіряти його роботоздатність.

1.2 Аналіз можливих рішень поставленого завдання

Першочерговою задачею виступає вибір апаратної частини системи. Серед параметрів, які необхідно враховувати при виборі апаратного забезпечення, є можливість опрацьовувати графіку та підтримувати веб-сервер. Враховуючи ці вимоги, були розглянуті різні варіанти апаратної платформи:

- arduino;
- beagleBone;
- intel NUC;
- raspberry Pi.

Arduino є популярним вибором для простих завдань автоматизації та контролю. Однак, обмежена продуктивність і відсутність підтримки графіки роблять його менш підходящим для задач, що вимагають обробки графіки та запуску веб-сервера.

BeagleBone має потужніший процесор і кращі можливості для обробки графіки, ніж Arduino, але його вартість є вищою, а спільнота менш чисельна в порівнянні з Raspberry Pi. Проте це також хороший варіант для більш складних задач.

Intel NUC пропонує високий рівень продуктивності і може працювати як повноцінний настільний комп'ютер. Це дозволяє виконувати навіть

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		11

найвимогливіші задачі, включаючи обробку складної графіки і підтримку високонавантажених веб-серверів. Однак, висока вартість і більші розміри можуть бути обмежуючими факторами.

Незважаючи на характеристики інших платформ мікрокомп'ютер Raspberry Pi став найкращим варіантом для реалізації даної комп'ютерної системи.

Переваги Raspberry Pi для виконання завдання:

- можливість обробки графіки;
- підтримка веб-сервера;
- низька вартість;
- компактність і енергоефективність;
- велика спільнота та підтримка;
- різноманіття моделей.

Raspberry Pi оснащений потужним графічним процесором, який підтримує OpenGL ES, що забезпечує ефективну обробку 3D-графіки. Це дозволяє реалізувати складні візуальні ефекти та інтерактивні графічні інтерфейси, що є важливим для задач, які потребують візуалізації даних у реальному часі.

Raspberry Pi може працювати як повноцінний веб-сервер завдяки наявності потужного процесора та достатньої кількості оперативної пам'яті. Існує безліч доступних веб-серверів, таких як Apache, Nginx, або lighttpd, які можна налаштувати на Raspberry Pi для обробки запитів та взаємодії з клієнтами через веб-інтерфейс.

Однією з ключових переваг Raspberry Pi є його доступна ціна. Це робить його ідеальним вибором для проектів з обмеженим бюджетом, забезпечуючи при цьому достатню продуктивність для виконання поставлених задач.

Raspberry Pi є компактним пристроєм, який споживає дуже мало енергії. Це робить його ідеальним для використання в проектах, де важливі розміри та енергоспоживання, наприклад, для вбудованих систем або мобільних пристроїв.

Навколо Raspberry Pi існує велика спільнота розробників і користувачів, що означає наявність широкого спектру ресурсів, включаючи документацію, форуми та приклади коду. Це спрощує процес розробки та вирішення можливих проблем, з якими можна зіткнутися під час роботи.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		12

Raspberry Pi представлений в різних моделях з різними технічними характеристиками. Це дозволяє вибрати найбільш підходящу модель залежно від конкретних вимог проекту, будь то продуктивність, об'єм пам'яті або наявність додаткових портів.

Після аналізу можливих варіантів було прийнято рішення використовувати Raspberry Pi як апаратну платформу для реалізації проекту. Цей вибір обумовлений оптимальним співвідношенням продуктивності, вартості, компактності та підтримки з боку спільноти, що робить Raspberry Pi ідеальним рішенням для задач, що вимагають обробки графіки та підтримки веб-сервера [3].

Також, не менш важливим є програмне забезпечення, яке відіграє ключову роль у забезпеченні функціональності системи, її продуктивності та зручності використання.

Для поставленої задачі було обрано "glslViewer" як засіб відображення GLSL коду [4]. Вибір цього інструменту обумовлений такими ключовими причинами:

- функціональність;
- простота використання;
- кросплатформенність;
- легка інтеграція.

GlslViewer є потужним інструментом для рендерингу та відображення коду GLSL у реальному часі. Він підтримує різноманітні шейдери, включаючи вершинні, фрагментні та геометричні шейдери, що дозволяє реалізовувати складні графічні ефекти.

GlslViewer має інтуїтивно зрозумілий інтерфейс, що робить його доступним для користувачів різного рівня підготовки. Це важливо для швидкого освоєння та ефективного використання інструменту в рамках проекту.

Завдяки своїй архітектурі, glslViewer легко інтегрується з іншими інструментами та системами, що дозволяє створювати комплексні рішення без значних зусиль.

GlslViewer працює на різних операційних системах, включаючи Linux, macOS та Windows, що забезпечує гнучкість у виборі платформи для розробки та тестування.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		13

Для забезпечення роботи веб-інтерфейсу було обрано Apache2 в якості веб-сервера [5]. Цей вибір обумовлений наступними факторами:

- надійність і продуктивність;
- гнучкість конфігурації;
- підтримка спільноти та документація;
- сумісність з Raspberry Pi.

Apache2 легко встановлюється і налаштовується на Raspberry Pi, що забезпечує безперебійну інтеграцію апаратної та програмної частин проекту. Це дозволяє швидко розгорнути веб-сервер та почати роботу без значних зусиль.

Велика спільнота користувачів і розробників забезпечує наявність численних ресурсів для навчання та вирішення проблем. Це включає офіційну документацію, форуми, блоги та інші джерела, які допомагають швидко знаходити відповіді на питання та вирішувати проблеми.

Apache2 пропонує широкий спектр налаштувань і модулів, що дозволяє налаштувати сервер під конкретні потреби проекту. Це включає підтримку SSL для забезпечення безпеки, різні методи аутентифікації та багато іншого.

Apache2 є одним з найбільш популярних і надійних веб-серверів, який забезпечує високу продуктивність та стабільність роботи. Він добре відомий своєю здатністю обробляти велику кількість одночасних запитів, що важливо для забезпечення безперебійної роботи веб-системи.

Також були використані додаткові компоненти та бібліотеки програмного забезпечення, такі як OpenGL і бібліотеки GLM та GLFW. Для роботи з графікою та створення візуальних ефектів система буде використовувати OpenGL, що забезпечує ефективне рендеринг графічних об'єктів. Бібліотеки GLM і GLFW додадуть зручність у роботі з математичними обчисленнями та управлінням вікнами відповідно.

Аналіз можливих рішень поставленого завдання показав, що вибір програмного забезпечення є критично важливим для успішної реалізації проекту. Використання glslViewer для відображення GLSL коду забезпечить високу якість візуалізації та зручність роботи з графічними шейдерами. Вибір Apache2 в якості веб-сервера гарантує надійність та продуктивність веб-інтерфейсу.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		14

РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

2.1 Розробка узагальненої структури комп'ютерної системи

Після аналізу вимог і параметрів майбутньої системи, була сформована комп'ютерна система, який базується на використанні Raspberry Pi. Це компактний і потужний мікрокомп'ютер, до якого підключений дисплей для відображення GLSL коду, що завантажується через веб-інтерфейс. Для обробки цього коду використовується glslViewer, оскільки він простий у використанні і має широку доступність. Веб-інтерфейс підтримується за допомогою веб-сервера Apache2 [6].

Огляд обраних компонентів спроектованої системи:

Raspberry Pi – це недорогий, але потужний мікрокомп'ютер, який може виконувати багато завдань, починаючи від простих обчислень до складної обробки графіки. У нашому проекті він відповідає за управління всіма підключеними пристроями та виконання коду GLSL. Вибір на користь Raspberry Pi зроблено завдяки його компактності, енергоефективності та достатній обчислювальній потужності для виконання наших завдань.

Дисплей є основним засобом візуалізації результатів виконання GLSL коду. Він підключений до Raspberry Pi і показує графічні шейдери в режимі реального часу. Вибір дисплея може змінюватися в залежності від потреб проекту: від маленьких LCD екранів до великих моніторів. В цьому випадку був обраний компактний TFC дисплей ili9486 розширенням 480x320.

Цей інструмент використовується для відображення і налагодження GLSL шейдерів. glslViewer обраний через його доступність та легкість інтеграції з іншими компонентами системи. Він дозволяє розробникам легко візуалізувати і тестувати результати своєї роботи, що є важливим для швидкого розвитку та налагодження проекту.

Веб інтерфейс – це ключовий компонент системи, який забезпечує

					КС КРБ 123.122.00.00 ПЗ			
Зм.	Аркуш	№ докум.	Підпис	Дата	ПРОЕКТНА ЧАСТИНА	Літ.	Аркуш	Аркушів
Розробив	Музичук А.						15	14
Перевірила	Луцук Н.С.					ТНТУ, каф. КС,		
Реценз.	Мудрик І.Я.					гр. СІ-41		
Н. контр.	Тиш Є.В.							
Затверд.	Осухівська Г.М.							

користувачам можливість завантажувати і редагувати GLSL код через інтернет. Веб-інтерфейс реалізований на базі веб-сервера Apache2, що гарантує стабільність і безпеку. Завдяки цьому інтерфейсу користувачі можуть взаємодіяти із системою з будь-якого пристрою, підключеного до інтернету.

Веб-сервер Apache2 використовується для підтримки веб-інтерфейсу. Це один з найпоширеніших і найнадійніших веб-серверів, який забезпечує масштабованість і легкість налаштування. Apache2 гарантує, що веб-інтерфейс буде працювати стабільно і безпечно, дозволяючи користувачам безперешкодно взаємодіяти із системою [7].

На рисунку нижче представлена узагальнена схема комп'ютерної системи, яка демонструє взаємодію між компонентами системи (рис.2.1).

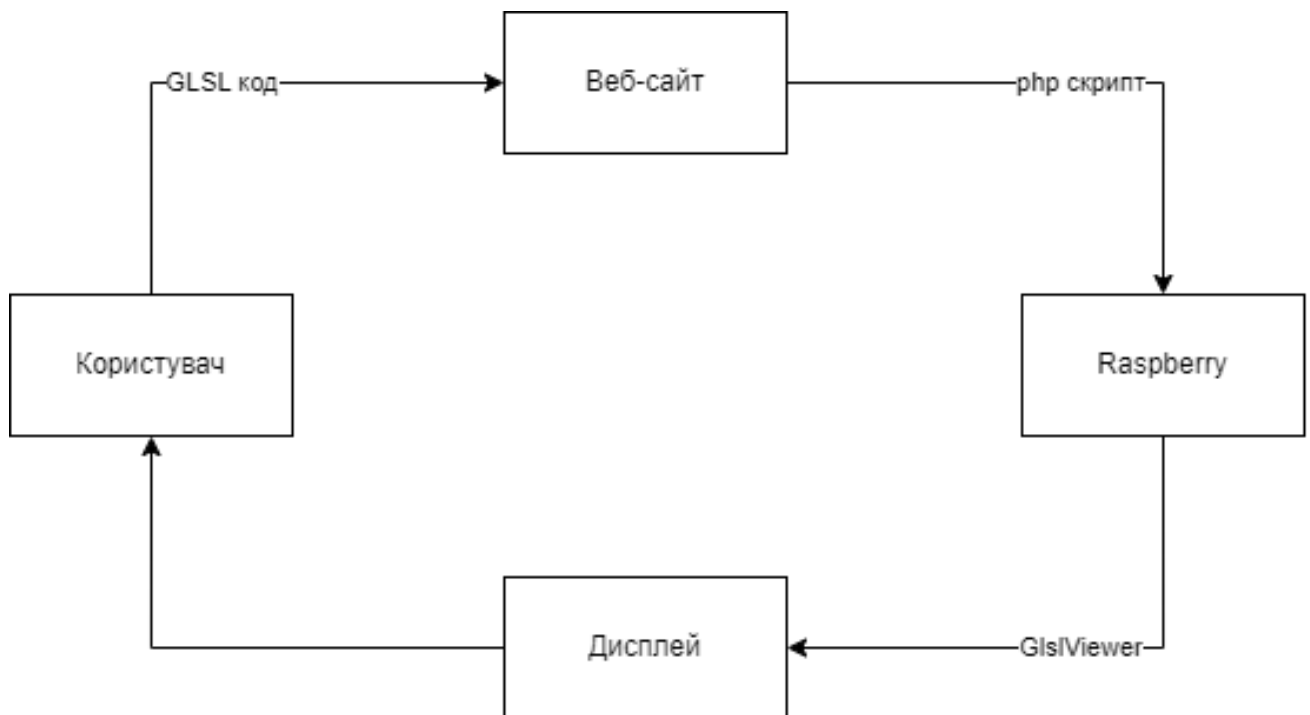


Рисунок 2.1 – Узагальнена схема системи

Використання такої архітектури системи має ряд значних переваг:

- мобільність і компактність;
- гнучкість і масштабованість;
- зручність використання.

Завдяки використанню Raspberry Pi, система залишається компактною і легко переносною. Це дає можливість розміщувати її в різних місцях і умовах, не обмежуючись одним стаціонарним місцем.

Можливість підключення різних типів дисплеїв і розширення функціональності веб-інтерфейсу відкриває широкі перспективи для подальшого розвитку проекту. Це дозволяє адаптувати систему під різні вимоги і задачі.

Проста інтеграція і налаштування компонентів дозволяють швидко адаптувати систему під конкретні потреби користувачів. Веб-інтерфейс забезпечує легкий доступ до функцій системи, роблячи її використання інтуїтивно зрозумілим.

2.2 Обґрунтування вибору апаратного забезпечення

При розробці проекту ключовим питанням стало обрання апаратного забезпечення, яке б найкраще відповідало поставленим задачам. Після ретельного аналізу обрано мікрокомп'ютер Raspberry Pi 4 та дисплей ili9486. Нижче детально розглянуто причини цього вибору, включаючи технічні параметри та переваги кожного компонента [8].

Однією з головних переваг Raspberry Pi 4 є його компактність. Це мініатюрний комп'ютер розміром 85.6 мм × 56.5 мм що дозволяє інтегрувати його в проекти з обмеженим простором. Компактність важлива для проектів, що передбачають створення мобільних або вбудованих рішень, де кожен сантиметр має значення. Дисплей ili9486 також є досить компактним, зазвичай розміром 3.5 дюйми, що забезпечує зручне відображення інформації без потреби у великому екрані.

Raspberry Pi 4 володіє потужним набором характеристик відносно свого розміру, що дозволяє виконувати різноманітні задачі, такі як обрахування графіки.

Raspberry Pi 4 має наступні технічні параметри:

- Процесор: Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.
- Оперативна пам'ять: 2GB, 4GB або 8GB LPDDR4-3200 SDRAM.
- Порти: 2 x USB 3.0, 2 x USB 2.0, 2 x micro-HDMI (до 4K), Gigabit Ethernet, 40 GPIO pins [9].

					КС КРБ 123.122.00.00 ПЗ	Акр.
						17
Зм.	Аркуш	№ докум.	Підпис	Дата		

– Підтримка бездротових з'єднань: 2.4/5.0 GHz 802.11ac wireless, Bluetooth 5.0, BLE.

Дисплей ili9486 легко підключається до Raspberry Pi через GPIO-піни або SPI-інтерфейс, забезпечуючи стабільну і безперебійну роботу. Це спрощує налаштування системи та інтеграцію компонентів, що важливо для досягнення швидкого результату. Дисплей ili9486 має такі технічні характеристики:

- роздільна здатність: 320 x 480 пікселів;
- інтерфейс: SPI;
- тип дисплея: TFT LCD;
- розмір: 3.5 дюйми;
- колірна гама: 262К кольорів;
- існуюче програмне забезпечення.

Raspberry Pi 4 пропонує величезну кількість готового програмного забезпечення, що значно спрощує розробку та тестування. На базі Raspberry Pi існує безліч проектів та бібліотек, зокрема операційна система Raspbian (тепер відома як Raspberry Pi OS), а також численні інструменти для програмування та налаштування пристроїв. Дисплей ili9486 підтримується багатьма бібліотеками, що дозволяє легко керувати ним з Raspberry Pi, використовуючи популярні мови програмування, такі як Python і C++.

Raspberry Pi має одну з найбільших і найактивніших спільнот розробників, що забезпечує величезну кількість ресурсів для навчання та вирішення проблем. На форумах, блогах та спеціалізованих сайтах можна знайти відповіді на будь-які питання, що виникають під час роботи з Raspberry Pi. Дисплей ili9486 також широко використовується, і для нього існує безліч посібників і прикладів коду, що дозволяє швидко освоїти роботу з ним. Така підтримка спільноти значно полегшує процес розробки та вирішення можливих проблем.

Raspberry Pi 4 відомий своєю простотою у використанні, що робить його ідеальним для швидкого старту проекту. Легкість налаштування та інтуїтивно зрозумілий інтерфейс дозволяють швидко розпочати роботу навіть новачкам. Встановлення операційної системи та налаштування необхідного програмного забезпечення займає мінімальний час. Дисплей ili9486 також легко налаштовується

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		18

і інтегрується в проект, що дозволяє зосередитися на основних задачах без відволікань на складні технічні деталі.

Не менш важливими перевагами Raspberry Pi, які варто виділити це:

- масштабованість;
- енергоефективність;
- висока продуктивність.

Raspberry Pi 4 можна використовувати в різних конфігураціях, залежно від потреб проекту, додаючи або видаляючи периферійні пристрої.

Raspberry Pi 4 споживає небагато енергії, що робить його економічно вигідним рішенням для тривалого використання.

Завдяки потужному процесору та великій кількості оперативної пам'яті, Raspberry Pi 4 здатний виконувати навіть складні обчислювальні задачі, що розширює його можливості використання в різних проектах.

Вибір Raspberry Pi 4 та дисплею ili9486 обумовлений їхньою компактністю, відмінною сумісністю, наявністю багатого програмного забезпечення, великою та активною спільнотою розробників, а також простотою у використанні. Ці фактори роблять їх ідеальними компонентами для реалізації проекту, дозволяючи швидко і ефективно виконувати поставлені задачі.

2.3 Обґрунтування вибору програмного забезпечення проектного комп'ютерного засобу

При розробці комп'ютерної системи особливу увагу було приділено вибору програмного забезпечення, яке б забезпечило надійність, ефективність та зручність у використанні. Після ретельного аналізу обрано glslViewer для відображення GLSL-коду, веб-сервер Apache2 для підтримки веб-інтерфейсу, мову програмування PHP для реалізації веб-інтерфейсу та операційну систему Raspberry Pi OS. Нижче детально розглянуто причини цього вибору, включаючи особливості, переваги та недоліки кожного компонента.

GlsViewer було обрано як засіб відображення GLSL-коду завдяки його численним перевагам:

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		19

- відображення в реальному часі;
- кросплатформенність;
- легка інтеграція;
- підтримка різних шейдерів;
- простота використання.

GslViewer має інтуїтивно зрозумілий інтерфейс і добре документовану API, що спрощує його освоєння навіть для новачків. Це дозволяє швидко розпочати роботу та ефективно використовувати інструмент.

GslViewer підтримує різні типи шейдерів, включаючи вершинні, фрагментні та геометричні шейдери. Це дозволяє реалізовувати складні графічні ефекти та забезпечує гнучкість у розробці візуальних рішень.

Завдяки своїй архітектурі, gslViewer легко інтегрується з іншими інструментами та системами, що дозволяє створювати комплексні рішення без значних зусиль. Це важливо для проектів, де необхідно швидко налаштувати і запустити систему.

GslViewer працює на різних операційних системах, включаючи Linux, macOS та Windows, що забезпечує гнучкість у виборі платформи для розробки та тестування.

GslViewer дозволяє рендерити GLSL-шейдери в реальному часі, що є критично важливим для інтерактивних графічних застосунків. Це забезпечує миттєвий зворотний зв'язок, що дозволяє швидко виявляти та виправляти помилки.

Одним з недоліків gslViewer є залежність від командного рядка. Багато функцій gslViewer керуються через командний рядок, що може бути незручним для користувачів, які звикли до графічного інтерфейсу.

Далі вказано особливості Apache2. Його було обрано як веб-сервер завдяки його різноманітним перевагам і характеристикам:

- надійність і продуктивність;
- гнучкість конфігурації;
- підтримка різних платформ;
- велика спільнота та документація.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		20

Apache2 є одним з найбільш популярних і надійних веб-серверів, який забезпечує високу продуктивність та стабільність роботи. Він здатний обробляти велику кількість одночасних запитів, що важливо для забезпечення безперебійної роботи веб-системи.

Apache2 пропонує широкий спектр налаштувань і модулів, що дозволяє налаштувати сервер під конкретні потреби проекту. Це включає підтримку SSL для забезпечення безпеки, різні методи аутентифікації, модулі для кешування та багато іншого.

Apache2 працює на різних операційних системах, включаючи Linux, Windows та macOS, що забезпечує гнучкість у виборі платформи для розгортання веб-сервера.

Apache2 має велику спільноту користувачів і розробників, що забезпечує наявність численних ресурсів для навчання та вирішення проблем. Це включає офіційну документацію, форуми, блоги та інші джерела, які допомагають швидко знаходити відповіді на питання та вирішувати проблеми.

Недоліки Apache2:

- відносно високе споживання ресурсів;
- складність налаштування для новачків.

Apache2 може споживати більше системних ресурсів у порівнянні з деякими іншими легковаговими веб-серверами, такими як Nginx, особливо при обробці великої кількості одночасних запитів.

Різноманітність налаштувань і модулів може бути занадто складною для новачків, які тільки починають працювати з веб-серверами.

Далі буде розглянуто особливості мови PHP [10], яку було обрано як мову програмування для реалізації веб-інтерфейсу завдяки його широким можливостям та зручності у використанні:

- широка підтримка веб-серверів;
- простота і зручність у використанні;
- висока продуктивність;
- велика спільнота та підтримка;
- гнучкість і масштабованість.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		21

PHP підтримується практично всіма веб-серверами, включаючи Apache2, що забезпечує легку інтеграцію та налаштування. Це дозволяє швидко створювати веб-додатки та забезпечувати їх стабільну роботу.

PHP відомий своєю простотою у використанні та низьким порогом входу. Це дозволяє швидко розпочати розробку веб-додатків навіть для новачків. Велика кількість готових бібліотек та фреймворків робить розробку ще зручнішою та ефективнішою.

PHP забезпечує високу продуктивність роботи веб-додатків, що важливо для забезпечення швидкого відгуку та обробки запитів користувачів. Завдяки використанню кешування та інших оптимізацій, PHP додатки можуть обробляти велику кількість одночасних запитів.

PHP має одну з найбільших спільнот розробників, що забезпечує доступ до численних ресурсів для навчання та підтримки. Це включає документацію, форуми, блоги, приклади коду та багато іншого, що допомагає швидко вирішувати проблеми та знаходити відповіді на питання.

PHP дозволяє створювати як прості веб-сайти, так і складні веб-додатки з високими вимогами до продуктивності та масштабованості. Це забезпечує гнучкість у реалізації різних проектів та дозволяє розширювати функціональність додатків у міру зростання потреб.

Одним з недоліків PHP є безпека. PHP, як мова, схильна до вразливостей, таких як SQL-ін'єкції, якщо код не написаний належним чином. Потрібно дотримуватися кращих практик безпеки, щоб запобігти можливим атакам.

Raspberry Pi OS було обрано як операційну систему для Raspberry Pi завдяки її спеціалізованим можливостям і перевагам:

- оптимізація для апаратури Raspberry Pi;
- зручний процес встановлення та налаштування;
- велика спільнота та підтримка;
- наявність графічного інтерфейсу.

Raspberry Pi OS спеціально розроблена для роботи з апаратним забезпеченням Raspberry Pi, що забезпечує оптимальну продуктивність і стабільність роботи.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		22

Raspberry Pi OS пропонує зручний процес встановлення та налаштування, що дозволяє швидко розпочати роботу. Операційна система постачається з попередньо встановленими інструментами та програмами, що спрощує процес налаштування.

Raspberry Pi OS має велику спільноту користувачів і розробників, що забезпечує доступ до численних ресурсів для навчання та підтримки. Це включає офіційну документацію, форуми, блоги та інші джерела.

Raspberry Pi OS пропонує графічний інтерфейс користувача (GUI), що робить її зручною для користувачів, які не мають досвіду роботи з командним рядком.

Недоліки Raspberry Pi OS:

- обмежена продуктивність;
- обмежена сумісність програмного забезпечення.

Хоча Raspberry Pi OS добре оптимізована для Raspberry Pi, її продуктивність може бути обмеженою у порівнянні з повноцінними настільними операційними системами, особливо при виконанні важких завдань.

Деяке програмне забезпечення, доступне для інших операційних систем, може бути недоступним або працювати не так добре на Raspberry Pi OS.

Вибір програмного забезпечення gLsViewer, Apache2, PHP та Raspberry Pi OS обумовлений їхніми численними перевагами, такими як надійність, продуктивність, гнучкість налаштування, простота використання та підтримка з боку спільноти розробників. Враховуючи переваги та недоліки кожного компонента, ці інструменти забезпечують ефективну роботу системи, легкість у налаштуванні та інтеграції, а також можливість швидкого розвитку та масштабування проекту.

2.4 Проектування комп'ютерної системи

Після оцінки компонентів можна перейти до формування електрично-принципової схеми проекту. Створення електрично-принципової схеми проекту є одним з ключових етапів у розробці електронних систем. Ця схема відображає всі електричні з'єднання та компоненти, необхідні для функціонування пристрою.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		23

Нижче розглянуто основні причини, чому створюють електрично-принципову схему:

- візуалізація архітектури системи;
- забезпечення правильності підключень;
- документація для виробництва;
- тестування та налагодження;
- модернізація та розширення;
- спрощення комунікації;
- відповідність стандартам та регламентам;
- економія часу та ресурсів.

Схема електрично-принципова надає зрозумілу візуалізацію всієї архітектури системи. Вона показує, як з'єднані між собою різні компоненти, такі як мікроконтролери, сенсори, дисплеї, живлення. Це дозволяє розробникам і інженерам легко зрозуміти структуру проекту та його взаємозв'язки.

Схема допомагає забезпечити правильність підключень всіх електричних компонентів. Вона відображає, які пінів мікроконтролера підключаються до яких елементів (наприклад, сенсорів або дисплеїв), що допомагає уникнути помилок при монтажі та пайці.

Електрично-принципова схема є важливим документом для виробництва пристрою. Вона використовується для створення друкованих плат (PCB), де компоненти будуть фізично розміщені і з'єднані. Виробники використовують ці схеми для точного виготовлення пристроїв згідно з вимогами проекту.

Схема є необхідною для процесу тестування та налагодження. Вона допомагає інженерам і технікам швидко знаходити можливі проблеми і неполадки в електричних з'єднаннях або компонентах. Завдяки схемі можна легко відслідковувати сигнали і перевіряти правильність роботи кожного елемента.

Електрично-принципова схема дозволяє легко модифікувати і розширювати систему. Вона дає чітке уявлення про те, які компоненти можна додати або замінити, а також які з'єднання потрібно змінити для впровадження нових функцій або покращення існуючих.

					КС КРБ 123.122.00.00 ПЗ	Акр.
						24
Зм.	Аркуш	№ докум.	Підпис	Дата		

За допомогою цієї схеми буде сформовано апаратна реалізація комп'ютерної системи.

Не менш важливою графічною документацією проекту є схема алгоритму роботи програми. Вона допоможе налаштувати ПЗ системи і уникнути логічних помилок в роботі програми.

Переваги схеми алгоритму роботи програми:

- візуалізація логіки роботи;
- полегшення розробки та налагодження;
- спрощення комунікації між членами команди;
- документування процесу розробки;
- підвищення якості програмного забезпечення;
- планування тестування;
- можливість модифікації та розширення.

Схема алгоритму надає наочне представлення послідовності кроків і логічних умов, що дозволяє краще зрозуміти, як саме працює програма.

Завдяки схемі розробники можуть легко відстежувати процес виконання програми і виявляти потенційні місця, де можуть виникнути помилки або неефективності.

Схема алгоритму є універсальним способом передачі інформації між різними учасниками проекту, включаючи розробників, тестувальників і менеджерів. Вона забезпечує загальне розуміння функціональності та логіки роботи програми.

Схема є важливою частиною технічної документації проекту, що допомагає зберегти інформацію про прийняті рішення і використовувані методи. Це особливо корисно для подальшого обслуговування і модернізації системи.

Чітке визначення алгоритму роботи дозволяє виявити і усунути логічні помилки ще на стадії проектування, що сприяє створенню надійного та якісного програмного забезпечення.

Схема алгоритму допомагає розробити детальний план тестування, охоплюючи всі можливі сценарії та умови роботи програми. Це забезпечує більш комплексне і ефективне тестування системи.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		26

Зі схемою алгоритму легше вносити зміни та додавати нові функції. Вона дозволяє зрозуміти, як нові компоненти інтегруються в існуючу систему і які зміни необхідно внести для підтримки нових вимог.

Результатом аналізу вхідної документації проекту стала схема алгоритму роботи програми, (рис.2.3).

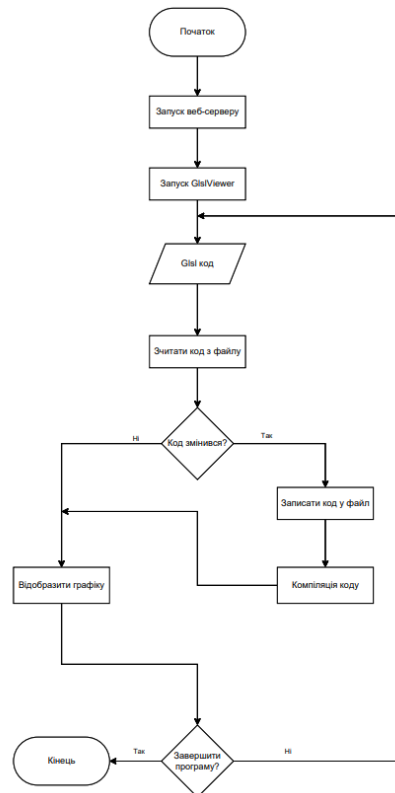


Рисунок 2.3 – Схема алгоритму роботи програми

Створена схема алгоритму програми, допоможе створити якісне програмне забезпечення, уникнути логічних помилок підчас розробки і оптимізувати процес реалізації ПЗ.

У розділі було детально розглянуто кілька ключових аспектів, необхідних для успішної реалізації проекту. Розробка узагальненої структури комп'ютерної системи, обґрунтування вибору апаратного та програмного забезпечення, а також проектування комп'ютерного засобу забезпечили цілісне і всебічне розуміння створюваної системи.

Узагальнена структура комп'ютерної системи включає всі основні компоненти та їхні взаємозв'язки. Вона забезпечує чітке уявлення про архітектуру

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		27

системи, що дозволяє ефективно планувати подальшу розробку та інтеграцію всіх елементів. Визначення ключових параметрів системи, таких як обробка коду в реальному часі, веб-інтерфейс для відправлення коду, відображення зображень і параметрів, а також сумісність з OpenGL, стали основою для подальших рішень.

Вибір апаратного забезпечення був обґрунтований з урахуванням потреб проекту в компактності, сумісності, існуючому програмному забезпеченні, об'єднання розробників і простоті використання. Raspberry Pi 4 і дисплей ILI9486 були обрані через їхню високу продуктивність, гнучкість і широку підтримку спільнотою розробників. Основні технічні характеристики Raspberry Pi 4 і дисплея ILI9486 підтвердили доцільність цього вибору, забезпечуючи надійну і ефективну основу для проекту.

Вибір програмного забезпечення також був ретельно обґрунтований. glslViewer був обраний для відображення GLSL коду завдяки своїй ефективності та зручності використання. Apache2 був обраний як веб-сервер через його надійність і поширеність, що забезпечує легке налаштування і масштабування. PHP додано для забезпечення динамічного оброблення запитів і інтеграції з веб-інтерфейсом. Всі ці програмні компоненти працюють разом, забезпечуючи стабільну і продуктивну роботу системи.

Процес проектування комп'ютерного засобу включав створення електрично-принципової схеми, розгляд інтерфейсу SPI, опис роботи дисплея з фрейм буфером [11] та аналіз недоліків системи. Електрично-принципова схема допомогла візуалізувати і налаштувати всі з'єднання, а розгляд SPI інтерфейсу забезпечив ефективну передачу даних між компонентами. Опис роботи дисплея з фрейм буфером дав можливість забезпечити коректне відображення інформації, а аналіз недоліків системи дозволив виявити та усунути потенційні проблеми на ранніх етапах розробки.

Даний розділ забезпечив всебічне планування та обґрунтування вибору всіх ключових компонентів проекту, як апаратних, так і програмних. Це створило міцну основу для подальшої реалізації і тестування системи, забезпечуючи її надійність, ефективність і відповідність всім поставленим вимогам.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		28

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Реалізація проектних рішень

На основі вхідної документації і сформованому графічному матеріалу можна приступити до фізичної реалізації комп'ютерної системи.

Першочергова задача це зібрати весь необхідний перелік компонентів та інструментів, які знадобляться в подальшому. В даному випадку це:

- raspberry PI 4 1gb;
- дисплей 3.5" TFT LCD ILI9486;
- sd карта пам'яті Samsung 64 gb EVO plus;
- блок живлення 5V, 3A;
- кардрідер.

Raspberry PI 4 1gb - основний мікрокомп'ютер, який буде виконувати обчислювальні функції і забезпечувати роботу програмного забезпечення.

Сенсорний дисплей 3.5" TFT LCD ILI9486 - використовується для виводу графічної інформації і забезпечення сенсорного введення від користувача [12].

Sd карта пам'яті Samsung 64 gb EVO plus - слугуватиме для зберігання операційної системи, програмного забезпечення та інших необхідних даних.

Блок живлення 5V, 3A - забезпечить стабільне живлення для Raspberry Pi 4 і підключеного до нього дисплея.

Кардрідер необхідний для запису операційної системи та інших файлів на SD карту пам'яті.

Усі перелічені компоненти є незамінними в розробці комп'ютеризованої системи, для досягнення поставлених цілей і забезпечення необхідного функціоналу.

Нижче зображено усі елементи системи які будуть приймати участь в розробці системи (рис.3.1).

					КС КРБ 123.122.00.00 ПЗ			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Розробив		Музичук А.			ПРАКТИЧНА ЧАСТИНА	Літ.	Аркуш	Аркушів
Перевірила		Луцук Н.С.					29	19
Реценз.		Мудрик І.Я.				ТНТУ, каф. КС,		
Н. контр.		Тиш Є.В.				гр. СІ-41		
Затверд.		Осухівська Г.М.						

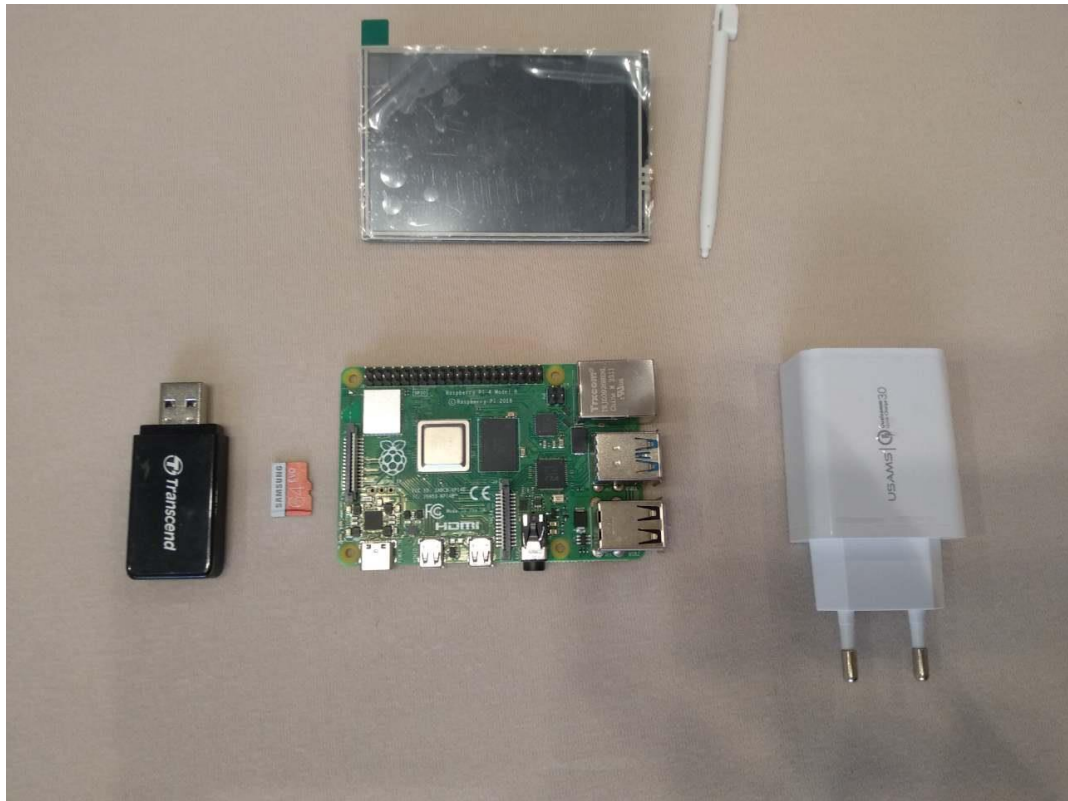


Рисунок 3.1 – Компоненти системи

Наступний етап – підключення дисплея, який є ключовим елементом системи. Він обмінюється даними по SPI з Raspberry PI.

SPI – це протокол послідовного периферійного інтерфейсу, розроблений для забезпечення високошвидкісної передачі даних між мікроконтролерами і периферійними пристроями [13]. Цей інтерфейс широко використовується в різних електронних пристроях завдяки своїй простоті та ефективності.

Основні характеристики SPI:

- Синхронна передача даних.
- Архітектура Master-Slave.
- Повнодуплексний режим.

SPI є синхронним протоколом, що означає, що передача даних керується тактовим сигналом (SCK, Serial Clock), який генерується головним пристроєм (Master).

В системі SPI завжди є один головний пристрій (Master) і один або кілька підлеглих пристроїв (Slaves). Master контролює обмін даними і генерує тактовий сигнал.

SPI підтримує повнодуплексний режим передачі даних, що означає, що дані можуть передаватися в обох напрямках одночасно. Для цього використовуються дві окремі лінії. MOSI (Master Out Slave In) і MISO (Master In Slave Out).

SPI використовує кілька ліній для передачі даних і керування:

- SCK (Serial Clock): тактовий сигнал, генерується Master;
- MOSI (Master Out Slave In): лінія передачі даних від Master до Slave;
- MISO (Master In Slave Out): лінія передачі даних від Slave до Master;
- SS/CS (Slave Select/Chip Select): лінія вибору підлеглого пристрою, використовується Master для вибору конкретного Slave.

Переваги SPI:

- Висока швидкість передачі даних.
- Простота протоколу.
- Гнучкість.
- Повнодуплексний режим.

SPI забезпечує дуже високу швидкість передачі даних у порівнянні з іншими послідовними інтерфейсами, такими як I2C або UART.

Протокол SPI є відносно простим, що спрощує його апаратну і програмну реалізацію.

SPI дозволяє підключати кілька підлеглих пристроїв до одного головного, що забезпечує гнучкість у побудові системи.

Можливість одночасної передачі і прийому даних підвищує ефективність обміну інформацією.

Недоліки SPI:

- велика кількість ліній керування;
- обмежена відстань передачі;
- відсутність стандарту протоколу.

Для кожного підлеглого пристрою потрібна окрема лінія SS/CS, що може збільшити кількість необхідних пінів на мікроконтролері.

SPI не підходить для передачі даних на великі відстані через високу чутливість до електромагнітних завад.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		31

На відміну від I2C, SPI не має стандартного протоколу для конфігурації та управління, що може призводити до несумісності між різними пристроями.

В проєкті з Raspberry Pi 4 і 3.5" TFT LCD дисплеєм ILI9486, SPI інтерфейс використовується для передачі даних між мікрокомп'ютером і дисплеєм. Це дозволяє Raspberry Pi відправляти графічну інформацію на дисплей з високою швидкістю, забезпечуючи швидке оновлення зображення і реакцію на сенсорні команди.

SPI є потужним і ефективним інтерфейсом для передачі даних між мікроконтролерами і периферійними пристроями. Його висока швидкість, повнодуплексний режим і простота реалізації роблять його ідеальним вибором для багатьох вбудованих систем, хоча для використання з великою кількістю підлеглих пристроїв і на великих відстанях можуть знадобитися альтернативні рішення (рис.3.2).

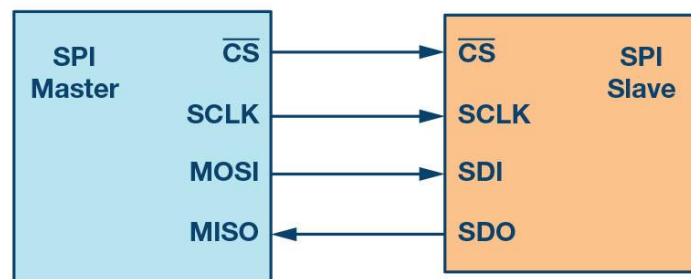


Рисунок 3.2 – Піни SPI інтерфейсу

Нижче наведено процес підключення дисплею ili9486 до Raspberry PI (рис.3.3).



Рисунок 3.3 – Піни підключення дисплею

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		32

Підключивши дисплей до Raspberry можна перевірити його працездатність давши напругу на інтерфейс power in на платі (рис.3.4).



Рисунок 3.4 – Перевірка роботоздатності дисплею

На фото видно, що дисплей працює, однак зображення не виводиться. Це пов'язано з тим що на Raspberry все ще не завантажена жодна операційна система.

Кроки, які необхідно виконати, для завантаження операційної системи на Raspberry Pi:

- підготовка SD карти;
- запис операційної системи на SD карту;
- встановлення SD карти в Raspberry Pi;
- підключіть блок живлення до Raspberry Pi;
- початкове налаштування операційної системи;
- дочекайтеся завантаження операційної системи.

Виконуючи перший пункт списку було сформовано загрузочний образ на SD накопичувачі. Для досягнення цього був підключений накопичувач через кардрідер так як зображено на рисунку нижче (рис.3.5).

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		33



Рисунок 3.5 – SD накопичувач в кардрідері

Формування образу буде проводитися за допомогою утиліти Raspberry Pi imager. Raspberry Pi Imager – це офіційна програма, розроблена командою Raspberry Pi Foundation, яка дозволяє легко записувати образи операційних систем на SD-карти для використання з мікрокомп'ютерами Raspberry Pi. Цей інструмент спрощує процес підготовки SD-карти, надаючи користувачам інтуїтивно зрозумілий інтерфейс і широкий вибір операційних систем. Основні функції Raspberry Pi Imager:

- запис образів операційних систем;
- підтримка широкого спектра ОС;
- вбудований каталог операційних систем;
- форматування SD-карт.

Доступ до функцій програми здійснюється за допомогою графічного інтерфейсу (рис.3.6).



Рисунок 3.6 – Інтерфейс засобу запису образів

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		34

Встановивши параметри утиліти на:

- пристрій Raspberry Pi: Raspberry Pi 4;
- операційна система: Raspberry Pi OS;
- записуючий пристрій: TS-RDF5A usb 64 gb.

Не менш важливим є встановлення параметрів системи. Вони дозволять налаштувати поведінку Raspberry Pi під час і після запуску. Було встановлено SSID і пароль wi-fi мережі, створений користувач ОС Linux і включений SSH протокол. Данні підключення дозволять Raspberry Pi в автоматичному порядку встановити зв'язок з мережею і обмінюватися даними з нею. Протокол SSH дозволить керувати Raspberry Pi через віддалений термінал що дуже корисно при початковій ініціалізації комп'ютерної системи (встановлення ПЗ, драйверів) (рис.3.7).

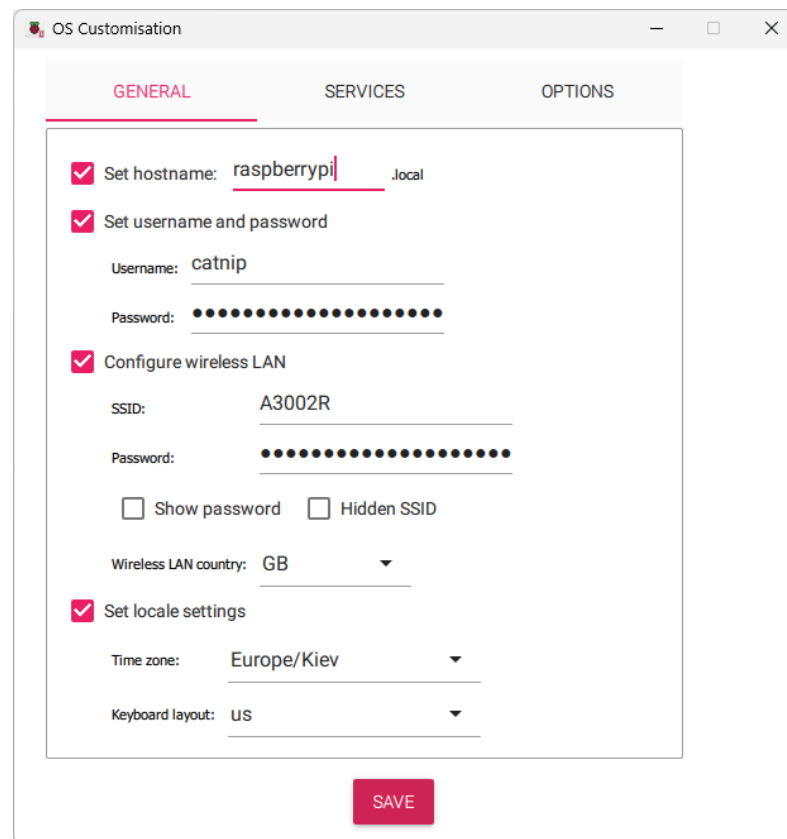


Рисунок 3.7 – Вікно кастомізації системи

Записавши образ на SD накопичувач за допомогою утиліти Raspberry Pi imager і запустивши мікрокомп'ютер, система розгорнеться і сотворить системні файли необхідні для функціонування.

Не зважаючи на те що дисплей не демонструє змін в поведінці системи, активність можна помітити в поведінці мережі, Raspberry Pi автоматично підключився до Wi-fi мережі, користуючись введеними раніше даними. Перевірити це можливо в інтерфейсі роутера, в вкладці підключені пристрої (рис.3.8).

raspberrypi	d8:3a:dd:9b:9d:17	192.168.0.3		
-------------	-------------------	-------------	---	---

Рисунок 3.8 – Присвоєне IP Raspberry Pi в локальній мережі

Визначивши ір системи було налагоджено комунікації через SSH. SSH – це мережевий протокол, який використовується для безпечного адміністрування і керування віддаленими серверами та іншими мережевими пристроями. SSH забезпечує шифроване з'єднання між клієнтом і сервером, що дозволяє виконувати команди, передавати файли і виконувати інші адміністративні задачі з високим рівнем безпеки. Основні характеристики SSH:

- безпека;
- аутентифікація;
- можливість тунелювання;
- переадресація портів;
- віддалене виконання команд.

SSH використовує шифрування для захисту даних, що передаються між клієнтом і сервером. Це запобігає перехопленню і несанкціонованому доступу до переданої інформації.

SSH підтримує кілька методів аутентифікації, включаючи паролі, ключі SSH (асиметричне шифрування) і сертифікати. Аутентифікація за ключами SSH є більш безпечною і рекомендується для використання.

SSH дозволяє створювати зашифровані тунелі для передачі різноманітних даних, включаючи віддалені робочі стільниці, бази даних та інші сервіси, через захищений канал.

SSH підтримує переадресацію портів (port forwarding), що дозволяє перенаправляти мережевий трафік через SSH-з'єднання, забезпечуючи додатковий

рівень безпеки для доступу до сервісів всередині захищених мереж.

За допомогою SSH можна виконувати команди на віддаленому сервері, що дозволяє адмініструвати систему і запускати програми дистанційно.

Як працює SSH:

- встановлення з'єднання;
- аутентифікація;
- передача даних.

Клієнт ініціює з'єднання з сервером через порт 22 (стандартний порт для SSH). Сервер відповідає, встановлюючи з'єднання і починаючи процес аутентифікації.

Клієнт і сервер обмінюються ключами шифрування для встановлення захищеного каналу. Потім клієнт проходить аутентифікацію за допомогою пароля або ключа SSH.

Після успішної аутентифікації всі дані, передані між клієнтом і сервером, шифруються, забезпечуючи їх конфіденційність і цілісність.

Переваги SSH:

- високий рівень безпеки;
- зручність використання;
- гнучкість;
- кросплатформеність.

Шифрування даних і надійні методи аутентифікації забезпечують захист від перехоплення і атак.

SSH дозволяє легко і швидко підключатися до віддалених систем для їх адміністрування і керування.

Підтримка тунелювання і переадресації портів дозволяє використовувати SSH для широкого спектра завдань.

SSH доступний для всіх основних операційних систем, включаючи Unix/Linux, Windows і macOS.

SSH є потужним і надійним інструментом для безпечного доступу до віддалених систем і мережевих пристроїв. Його використання забезпечує високий рівень безпеки і конфіденційності, що робить його незамінним в адмініструванні

					КС КРБ 123.122.00.00 ПЗ	Акр.
						37
Зм.	Аркуш	№ докум.	Підпис	Дата		

серверів, передачі даних і віддаленому виконанні команд. Незважаючи на деяку складність налаштування для новачків, переваги SSH значно перевищують його недоліки, роблячи його важливим компонентом сучасних ІТ-систем.

Для встановлення зв'язку через SSH також необхідно використовувати SSH клієнт, який дозволить відправляти команди. Було розглянуто декілька застосунків.

Termius — це сучасний SSH-клієнт, призначений для керування серверами з мобільних пристроїв, комп'ютерів та веб-браузерів. Він надає зручний інтерфейс для підключення до віддалених машин, підтримує різні протоколи (SSH, Mosh, Telnet) і пропонує такі функції, як синхронізація налаштувань між пристроями, збереження паролів та ключів, а також можливість створення скриптів для автоматизації задач. Termius також включає функції для керування хостами, групами і тегами, що дозволяє легко організувати та керувати великими мережами серверів.

Ще одним розглянутим клієнтом є OpenSSH (Open Secure Shell) — це набір програм для захищеного віддаленого доступу до комп'ютерів по мережі, що використовує протокол SSH (Secure Shell). OpenSSH забезпечує шифрування з'єднання, що захищає від перехоплення даних, а також дозволяє автентифікацію користувачів за допомогою паролів або ключів. До складу OpenSSH входять такі інструменти, як ssh (для підключення до віддалених машин), scp (для безпечної передачі файлів) та sftp (для роботи з файлами через SSH). OpenSSH широко використовується для адміністрування серверів та забезпечення безпеки віддаленого доступу.

MobaXterm — це розширений термінальний емулятор для Windows, який об'єднує в собі численні мережеві інструменти та функції для віддаленого адміністрування. Він підтримує протоколи SSH, X11, RDP, VNC, FTP, MOSH та інші, що дозволяє користувачам зручно керувати віддаленими серверами та системами. MobaXterm також включає вбудований X-сервер для запуску графічних додатків Linux на Windows, текстовий редактор, засоби для роботи з мережами та скриптами, а також можливості для збереження та управління сесіями. Цей інструмент є популярним серед системних адміністраторів і розробників завдяки своїй універсальності та зручності використання.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		38

В результаті був обраний клієнт PuTTY [14]. PuTTY — це безкоштовний та відкритий термінальний емулятор для Windows, який підтримує кілька мережевих протоколів, включаючи SSH, Telnet, SCP та rlogin. Він широко використовується для безпечного віддаленого доступу до серверів і мережевих пристроїв. PuTTY надає простий інтерфейс для підключення до віддалених систем, дозволяючи виконувати команди в терміналі. Його ключові особливості включають генерацію та управління ключами SSH, збереження сесій та налаштувань, а також підтримку різних видів аутентифікації. PuTTY є популярним серед системних адміністраторів і користувачів, які потребують надійного інструменту для віддаленого доступу. Серед переваг які стали ключовими в виборі утиліти були:

- легкість використання;
- широка підтримка протоколів;
- кросплатформеність;
- легка конфігурація;
- безкоштовність.

Інтуїтивно зрозумілий інтерфейс, який легко освоїти навіть новачкам. Крім SSH, підтримує Telnet, SCP, SFTP та інші протоколи. Основна версія розроблена для Windows, але також є версії для Unix/Linux і macOS. Великий вибір налаштувань для підключення, зокрема підтримка різних методів аутентифікації, включаючи ключі SSH. PuTTY є безкоштовним і відкритим програмним забезпеченням.

Таким чином, вибір PuTTY був обумовлений його широкою поширеністю, простотою у використанні, багатофункціональністю і безкоштовністю, що робить його оптимальним вибором для більшості користувачів, які шукають надійний SSH-клієнт для своїх потреб.

Встановивши зв'язок через SSH і авторизувавшись в системі через клієнт, необхідно скачати програми для реалізації функціоналу комп'ютерної системи.

За допомогою вбудованої команди apt-get було встановлено:

- Apache2;
- PHP.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		39

Програми, які не входять в офіціальний репозиторій, не можливо встановити через Apt-get. Тому, програму GlsViewer необхідно клонувати самостійно через git clone. Також для роботи застосунку необхідно встановити залежні бібліотеки. Для виконання процесу встановлення був сформований перелік команд (рис.3.9).

```
sudo apt install git cmake xorg-dev libglu1-mesa-dev
libncurses5-dev libncursesw5-dev
git clone https://github.com/patriciogonzalezvivo/glsViewer.git
cd glsViewer
git submodule init
git submodule update
mkdir build
cd build
cmake ..
make
sudo make install
```

Рисунок 3.9 – Стандартна сторінка Apache2

Після цього в систему буде встановлено GlsViewer доступ до якого можна отримати через однойменну команду. Щоб запустити програму, потрібно також ввести перемінну дисплею, за допомогою команди `export DISPLAY=:0`. Ввівши команду `glsViewer -f` Буде відкрито програму на весь екран, так як на рисунку нижче (рис.3.9).

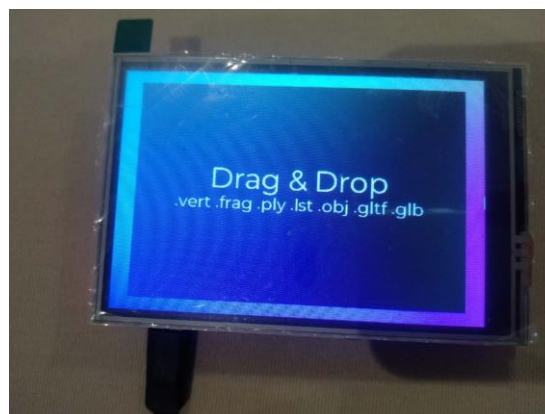


Рисунок 3.10 – Вступне вікно програми glsViewer

Маючи усі необхідні програми можна приступити до реалізації веб-серверу. Після встановлення Apache2, він автоматично розгортається і стає доступний за адресою пристрою. Якщо звернутися до адреси за допомогою браузера буде відкрито стандартна сторінка Apache2 (рис.3.10).

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		40

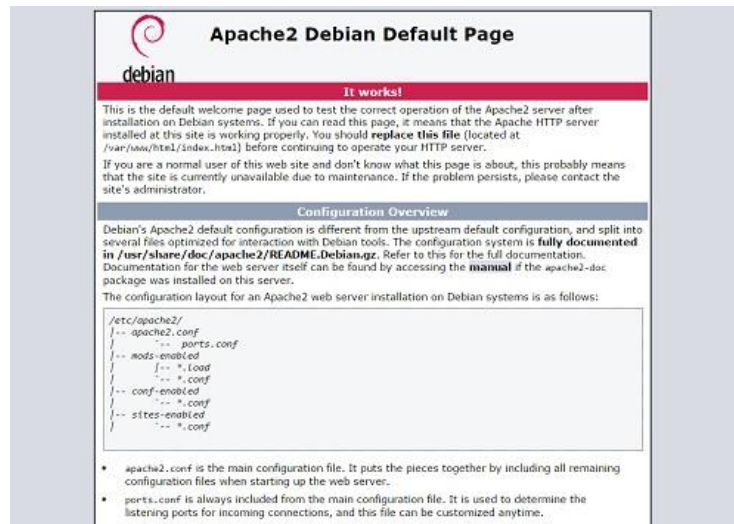


Рисунок 3.11 – Стандартна сторінка Apache2

Переконавшись в роботоздатності Apache2 можна налаштувати свою персональну сторінку, переписавши код html файлу який знаходиться в директорії var/www/html/index.html.

Таким чином було сформовано нову сторінку в яку входять заголовок, поле вводу тексту і кнопка відправлення.

Також був сформований PHP скрипт, який дозволяє зберігати текст поля з сайту в текстовий файл в директорії var/www/html. Лістинг коду наведений нижче (рис.3.11).

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $text = $_POST['text'] ?? '';

    if (!empty($text)) {
        $filename = 'submitted_gls1.frag';
        if (file_put_contents($filename, $text)) {
            echo "Код успішно оброблено";
        } else {
            echo "Помилка при збереженні коду у файл";
        }
    } else {
        echo "Текст не може бути порожнім";
    }
} else {
    echo "Неправильний метод запиту";
}
?>
```

Рисунок 3.12 – Лістинг PHP скрипту формування файлів

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		41

Існуюча логіка була зв'язана через JavaScript який формував текст з поля введення і відправляв на сервер. Лістинг коду наведений нижче (рис.3.12).

```
<script>
function sendText() {
    const text =
document.getElementById('textInput').value;
    const xhr = new XMLHttpRequest();
    xhr.open('POST', 'save_text.php', true);
    xhr.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');

    xhr.onload = function () {
        if (xhr.status === 200) {

document.getElementById('response').innerText = xhr.responseText;
        } else {

document.getElementById('response').innerText = 'Виникла помилка при
відправці тексту';
        }
    };

    xhr.send('text=' + encodeURIComponent(text));
}
</script>
```

Рисунок 3.13 – Лістинг JavaScript для відправки тексту

В результаті було отримано сайт який дозволяє формувати текст з поля введення у файл Submitted_gls1.frag. Вигляд сайту наведено нижче (рис.3.13).

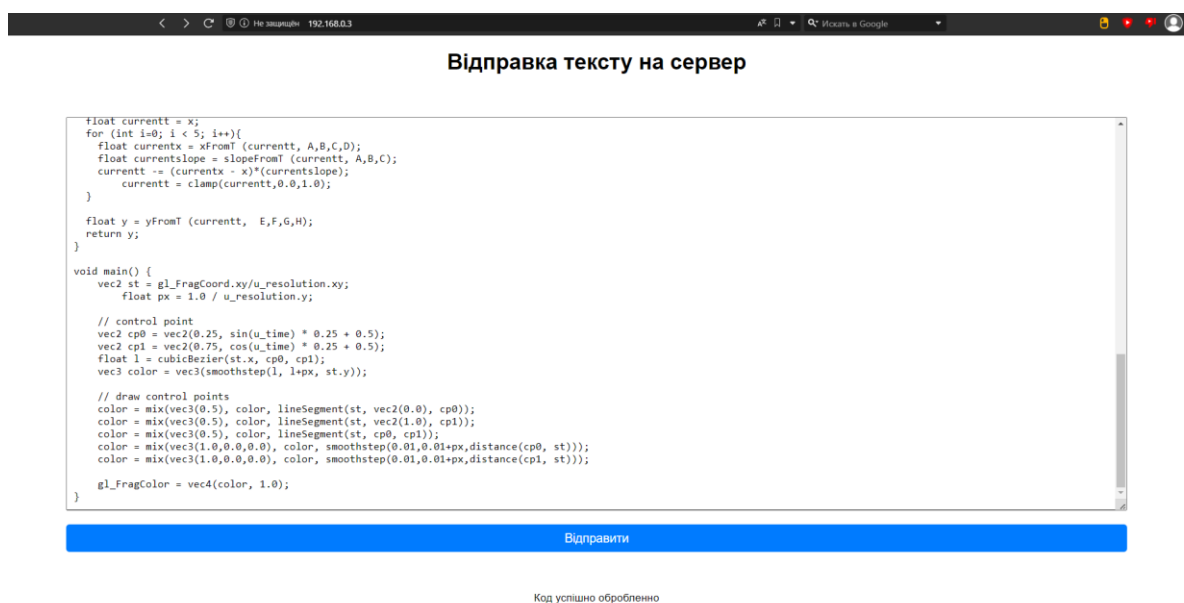


Рисунок 3.14 – Вигляд сторінки відправки коду

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		42

Тепер відправимо короткий GLSL код, який виведе кольоровий круг на дисплей. Отримавши код, залишається запустити його, за допомогою glslViewer увівши команду:

```
glslViewer submitted_glsl.frag -f -nocursor
```

Аргумент -f відкриває застосунок на повний екран, а -nocursor ховає курсор підчас рендеру зображення. В результаті відкрилось вікно, яке демонструє кольоровий круг, так як передбачено кодом (рис.3.14).

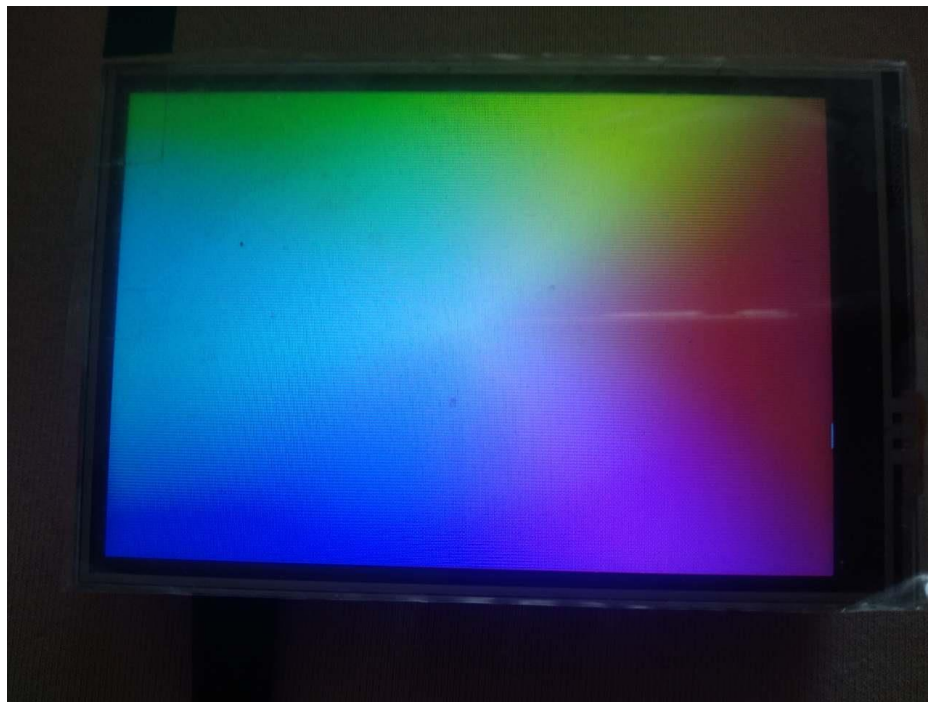


Рисунок 3.15 – Вивід GLSL коду на дисплей

На цьому, розгортання комп'ютерної системи тестування GLSL коду, завершено. Тепер система повністю готова до використання для виконання тестів і розробки графічних шейдерів.

3.2 Тестування

Завершивши розробку системи, залишається протестувати її роботоздатність. Тестування складається з відправки 3 різних GLSL кодів, які будуть передаватися в систему і вливати на вивід зображення.

					КС КРБ 123.122.00.00 ПЗ	Акр.
						43
Зм.	Аркуш	№ докум.	Підпис	Дата		

Перший код, після відправлення має відобразити градієнт, який переливається в залежності від перемінної часу(рис.3.15)

```

#ifdef GL_ES
precision mediump float;
#endif

uniform vec2 u_mouse;
uniform vec2 u_resolution;
uniform float u_time;

uniform sampler2D u_tex0;
uniform vec2 u_tex0Resolution;

void main (void) {
    vec2 st = gl_FragCoord.xy/u_resolution.xy;
    float aspect = u_resolution.x/u_resolution.y;
    st.x *= aspect;

    vec3 color = vec3(0.0);
    color = vec3(st.x, st.y, (1.0+sin(u_time))*0.5);

    if ( u_tex0Resolution != vec2(0.0) ){
        float imgAspect = u_tex0Resolution.x/u_tex0Resolution.y;
        vec4 img = texture2D(u_tex0,st*vec2(1.,imgAspect));
        color = mix(color,img.rgb,img.a);
    }

    gl_FragColor = vec4(color,1.0);
}

```

Рисунок 3.16 – Лістинг першого тестового GLSL коду

Результат виконання першого тестового коду на рисунку 3.16.

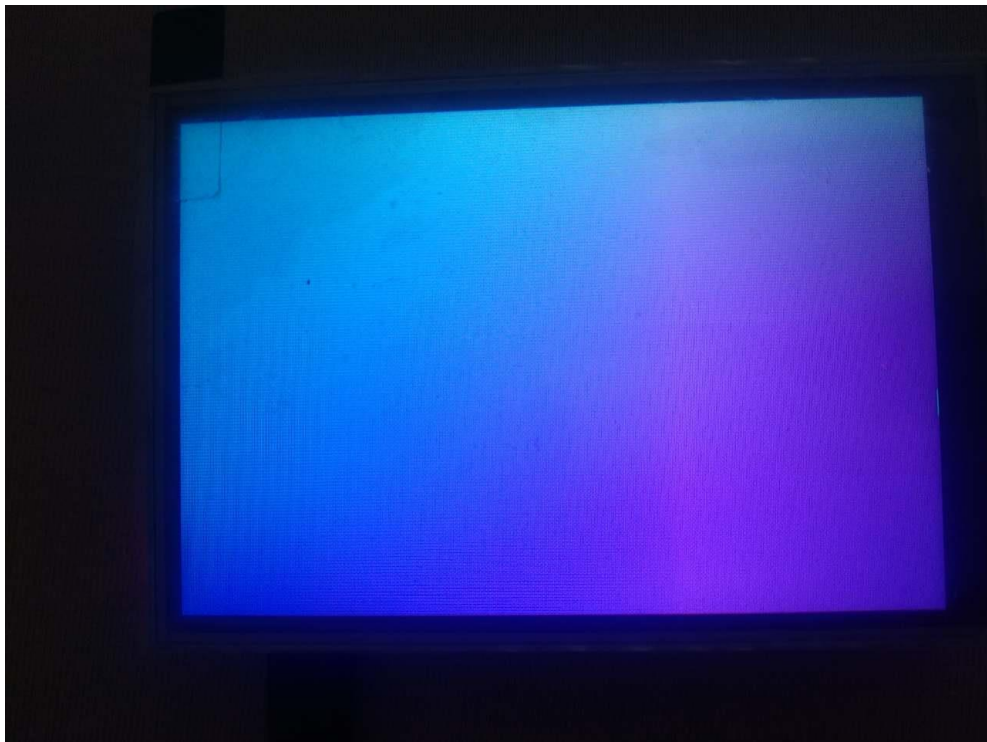


Рисунок 3.17 – Результат виконання першого тестового GLSL коду

					КС КРБ 123.122.00.00 ПЗ	Акр.
						44
Зм.	Аркуш	№ докум.	Підпис	Дата		

Другий код, після відправлення має відобразити патерн схожий на гістограму, який буде наростати по мірі плину часу (рис.3.17)

```
#ifdef GL_ES
precision mediump float;
#endif

#define IVORY vec3(1.0, 0.9, 0.8)
#define SUNSET vec3(0.9, 0.3, 0.3)
#define NAVY vec3(0.0, 0.1, 0.2)

uniform vec2 u_resolution;
uniform vec2 u_mouse;
uniform float u_time;

float plot(vec2 st, float pct){
    return smoothstep( pct - 0.005, pct, st.y) -
           smoothstep( pct, pct + 0.005, st.y);
}

float easeInQuad(float t) {
    return t * t;
}

float easeOutQuad(float t) {
    return -1.0 * t * (t - 2.0);
}

float easeInOutQuad(float t) {
    if ((t *= 2.0) < 1.0) {
        return 0.5 * t * t;
    } else {
        return -0.5 * ((t - 1.0) * (t - 3.0) - 1.0);
    }
}

float easeInCubic(float t) {
    return t * t * t;
}

float easeOutCubic(float t) {
    return (t = t - 1.0) * t * t + 1.0;
}

float easeInOutCubic(float t) {
    if ((t *= 2.0) < 1.0) {
        return 0.5 * t * t * t;
    } else {
        return 0.5 * ((t -= 2.0) * t * t + 2.0);
    }
}
}
```

Рисунок 3.18 – Лістинг другого тестового GLSL коду

Результат виконання другого тестового GLSL коду на рисунку 3.18.



Рисунок 3.19 – Результат виконання другого тестового GLSL коду

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		45

Останній, третій тестовий код, після відправлення, має відображати криву Безьє яка буде змінювати свою кривизну відповідно до перемінної часу (рис.3.19).

```

#ifdef GL_ES
precision mediump float;
#endif

uniform vec2 u_resolution;
uniform vec2 u_mouse;
uniform float u_time;

// Helper functions:
float slopeFromT (float t, float A, float B, float C) {
    float dtdx = 1.0/(3.0*A*t*t + 2.0*B*t + C);
    return dtdx;
}

float xFromT (float t, float A, float B, float C, float D) {
    float x = A*(t*t*t) + B*(t*t) + C*t + D;
    return x;
}

float yFromT (float t, float E, float F, float G, float H) {
    float y = E*(t*t*t) + F*(t*t) + G*t + H;
    return y;
}

float lineSegment(vec2 p, vec2 a, vec2 b) {
    vec2 pa = p - a, ba = b - a;
    float h = clamp( dot(pa,ba)/dot(ba,ba), 0.0, 1.0 );
    return smoothstep(0.0, 1.0 / u_resolution.x, length(pa - ba*h));
}

float cubicBezier(float x, vec2 a, vec2 b) {
    float y0a = 0.0; // initial y
    float x0a = 0.0; // initial x
    float y1a = a.y; // 1st influence y
    float x1a = a.x; // 1st influence x
    float y2a = b.y; // 2nd influence y
    float x2a = b.x; // 2nd influence x
    float y3a = 1.0; // final y
    float x3a = 1.0; // final x

    float A = x3a - 3.0*x2a + 3.0*x1a - x0a;
    float B = 3.0*x2a - 6.0*x1a + 3.0*x0a;
    float C = 3.0*x1a - 3.0*x0a;
    float D = x0a;

    float E = y3a - 3.0*y2a + 3.0*y1a - y0a;
    float F = 3.0*y2a - 6.0*y1a + 3.0*y0a;
    float G = 3.0*y1a - 3.0*y0a;
    float H = y0a;

    // Solve for t given x (using Newton-Raphelson), then solve for y
    given t.
    // Assume for the first guess that t = x.
    float currentt = x;
    for (int i=0; i < 5; i++){

```

Рисунок 3.19 – Лістинг третього тестового GLSL коду

Результат виконання третього тестового GLSL коду на рисунку 3.20.

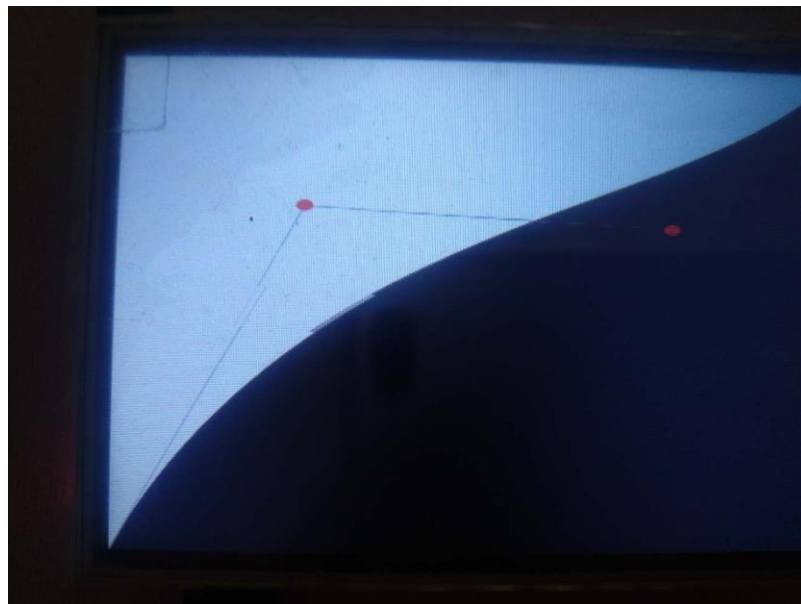


Рисунок 3.20 – Результат виконання третього тестового GLSL коду

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		46

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при ураженні електричним струмом

Ураження електричним струмом є серйозною небезпекою, яка може призвести до тяжких наслідків, включаючи серйозні опіки, зупинку серця та навіть смерть. Негайне надання долікарської допомоги є критично важливим для збереження життя та мінімізації ушкоджень. У цьому розділі розглянуто основні кроки, які виконуються для надання допомоги постраждалому від ураження електричним струмом.

Перед тим як наблизитись до постраждалого, оцініть ситуацію та забезпечте власну безпеку:

- вимкнути джерело електричного струму, якщо це можливо;
- якщо неможливо вимкнути струм, відсунути постраждалого від джерела електрики за допомогою непровідних предметів, уникати використання металевих або вологих матеріалів;
- переконатися, що ви самі не знаходитесь у небезпечній зоні під дією електричного струму.

Після того як постраждалий був відключений від джерела електричного струму, оцініть його стан:

- перевірити свідомість. Обережно струснути за плече та голосно запитати, чи постраждалий вас чує;
- перевірити дихання. Нахилити голову постраждалого назад, підняти підборіддя та перевірити, чи є дихальні рухи та звуки;
- перевірити пульс. Якщо постраждалий не дихає, спробувати знайти пульс на сонній артерії.

Залежно від стану постраждалого, надати допомогу. Якщо постраждалий у свідомості:

КС КРБ 123.122.00.00 ПЗ				
Зм.	Аркуш	№ докум.	Підпис	Дата
Розробив	Музичук А.			
Перевірив	Луцик Н.С.			
Консульт.	Пилипець М.І.			
Н. контр.	Тиш Є.В.			
Затверд.	Осухівська Г.М.			

БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ			Літ.	Аркуш	Аркушів
				47	6
			ТНТУ, каф. КС, єр. СІ-41		

– заспокоїти постраждалого, переконатися, що він знаходиться у безпечному місці;

– перевірити наявність опіків або інших видимих ушкоджень та надати першу допомогу (накласти стерильну пов'язку на опіки, при потребі охолодити уражене місце холодною водою, але не льодом);

– викликати швидку допомогу, описати ситуацію та стан постраждалого.

Якщо постраждалий непритомний, але дихає:

– покласти постраждалого на бік у стабільне положення для підтримки прохідності дихальних шляхів та запобігання аспірації (на випадок якщо постраждалий почне блювати);

– накрити постраждалого ковдрою або одягом для запобігання переохолодженню;

– негайно викликати швидку допомогу.

Якщо постраждалий не дихає або не має пульсу:

– негайно розпочати серцево-легеневу реанімацію (СЛР). виконувати компресії грудної клітини з частотою 100-120 натискань на хвилину, глибина компресії повинна бути близько 5-6 см. по можливості здійснювати штучне дихання (2 вдихи на кожні 30 компресій);

– продовжувати СЛР до прибуття медичної допомоги або відновлення нормального дихання та пульсу постраждалого;

– виклик медичної допомоги та подальші дії.

Навіть якщо постраждалий здається у стабільному стані, ураження електричним струмом може мати відкладені наслідки. Тому обов'язково:

– викликати швидку допомогу, повідомити оператору про ураження електричним струмом;

– не залишати постраждалого одного до прибуття медичних працівників;

– надавати постійну підтримку та спостерігати за станом постраждалого, бути готовим знову розпочати СЛР у разі погіршення стану.

У випадку ураження електричним струмом швидкість та правильність надання першої допомоги є критичними факторами, що можуть врятувати життя та мінімізувати наслідки для здоров'я постраждалого. Знання основних принципів

					КС КРБ 123.122.00.00 ПЗ	Акр.
						48
Зм.	Аркуш	№ докум.	Підпис	Дата		

долікарської допомоги допомагає ефективно діяти у таких надзвичайних ситуаціях [15].

4.2 Вплив електромагнітних полів (ЕМП) на людину та заходи щодо зменшення їх впливу на обслуговуючий персонал

Електромагнітні поля (ЕМП) є невід'ємною частиною сучасного технологічного середовища. Вони створюються багатьма пристроями, включаючи мобільні телефони, комп'ютери, побутову техніку, лінії електропередач та інші електронні системи. Вплив ЕМП на здоров'я людини є предметом активних досліджень, оскільки високі рівні електромагнітного випромінювання можуть мати негативні наслідки. Тому важливо знати про потенційні ризики та заходи щодо зменшення впливу ЕМП на обслуговуючий персонал.

ЕМП можуть впливати на людину по-різному, залежно від частоти, інтенсивності та тривалості впливу. Серед основних потенційних впливів виділяють:

- тепловий ефект. Високоінтенсивні електромагнітні поля можуть викликати нагрівання тканин тіла, що може призвести до термічних ушкоджень. Це особливо актуально для високочастотних полів;

- нетермічні ефекти. Низькочастотні електромагнітні поля можуть впливати на нервову систему, серцево-судинну систему та інші біологічні процеси. Серед можливих наслідків – головні болі, втома, порушення сну та інші функціональні зміни;

- хронічний вплив. Довготривалий вплив навіть низьких рівнів ЕМП може призводити до хронічних захворювань. Деякі дослідження вказують на можливий зв'язок між довготривалим впливом ЕМП та підвищеним ризиком розвитку раку.

Заходи щодо зменшення впливу ЕМП. Для захисту обслуговуючого персоналу від негативного впливу ЕМП вживають заходи, що допоможуть знизити рівень опромінення:

– зонування робочих місць. Організація робочих зон з урахуванням розташування джерел ЕМП. Мінімізація перебування працівників у зонах з високим рівнем випромінювання;

– екранування джерел випромінювання. Використання екрануючих матеріалів та конструкцій для зменшення рівня ЕМП. Це можуть бути металеві екрани, сітки або спеціальні поглинаючі матеріали;

– збільшення відстані від джерел ЕМП. Зменшення впливу електромагнітних полів можна досягти за рахунок збільшення відстані між працівниками та джерелами випромінювання. Інтенсивність ЕМП зменшується пропорційно до квадрата відстані від джерела;

– часові обмеження. Встановлення обмежень на час перебування працівників у зонах з високим рівнем ЕМП. Це допоможе знизити загальний вплив на організм;

– особисті захисні засоби. Використання засобів індивідуального захисту, таких як захисні екрани, спеціальний одяг та інші пристрої, що знижують вплив ЕМП;

– моніторинг рівнів ЕМП. Регулярний контроль та вимірювання рівнів електромагнітних полів на робочих місцях. Використання спеціальних приладів для моніторингу та аналізу ЕМП, що дозволяє своєчасно виявляти перевищення допустимих норм;

– освітні програми та інструктажі. Підвищення обізнаності працівників про потенційні ризики ЕМП та навчання методам зменшення впливу. Регулярне проведення інструктажів та тренінгів з охорони праці;

Захист працівників від впливу ЕМП регулюється відповідними нормативними актами та стандартами. Національні та міжнародні організації, такі як Міжнародна комісія з захисту від неіонізуючого випромінювання (ICNIRP) та Всесвітня організація охорони здоров'я (ВООЗ), встановлюють допустимі рівні впливу та рекомендації щодо захисту від електромагнітного випромінювання. Важливо дотримуватися цих норм та впроваджувати рекомендації на робочих місцях для забезпечення безпеки працівників.

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		50

Вплив електромагнітних полів на людину є важливою проблемою, яка потребує належної уваги та заходів щодо зменшення ризиків. Дотримання рекомендацій щодо зонування робочих місць, екранування джерел випромінювання, використання захисних засобів та проведення освітніх програм допоможе мінімізувати вплив ЕМП на обслуговуючий персонал, забезпечуючи безпечні умови праці [16].

4.3 Заходи захисту обладнання від статичної електрики

Статична електрика може стати причиною серйозних проблем при роботі з електронним обладнанням, включаючи вихід з ладу компонентів, пошкодження мікросхем та переривання роботи систем. Захист від статичної електрики є важливим аспектом забезпечення надійності та довговічності обладнання. Нижче наведено основні заходи, які допомагають запобігти пошкодженням, викликаним статичною електрикою.

Одним з ключових методів захисту від статичної електрики є використання антистатичних матеріалів:

- антистатичні підлоги. Використання підлогових покриттів, виготовлених з антистатичних матеріалів, допомагає знизити накопичення статичної електрики на поверхні;
- антистатичний одяг. Спеціальний антистатичний одяг та взуття, які носять працівники, дозволяють знизити ризик накопичення та передачі статичної електрики на обладнання;
- антистатичні пакети та коробки. Для зберігання та транспортування електронних компонентів використовуються антистатичні пакети, коробки та контейнери, що захищають їх від впливу ESD.

Також одним із методів захисту виступає Заземлення. Воно є ефективним методом для розсіювання статичної електрики. Робочі столи та поверхні повинні бути заземлені, щоб забезпечити відведення статичної електрики, яка може накопичуватися під час роботи з електронними компонентами. Заземлення

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		51

обладнання. Всі електронні пристрої та обладнання повинні бути надійно заземлені для запобігання накопичення статичного заряду.

Працівники, що працюють з чутливими електронними компонентами, повинні носити антистатичні браслети, які заземлюються через дроти, знижуючи таким чином ризик передачі статичної електрики. Використання антистатичних килимків на робочих столах допомагає знизити накопичення статичної електрики на робочій поверхні.

Низька вологість повітря сприяє накопиченню статичної електрики. Контроль за вологістю в приміщеннях, де працюють з електронними компонентами, допомагає знизити ризик ESD. Тому використовують зволожувачі повітря. Використання зволожувачів повітря дозволяє підтримувати оптимальний рівень вологості в приміщенні, знижуючи таким чином можливість накопичення статичної електрики.

Для ефективного захисту від статичної електрики також впроваджують організаційні заходи:

- регулярні інструктажі та навчання персоналу щодо правил роботи з електронними компонентами та методів захисту від статичної електрики;
- встановлення процедур контролю за дотриманням правил захисту від статичної електрики, проведення перевірок та аудитів.

Для зниження рівня статичної електрики можуть використовуватись спеціальні пристрої:

- Іонізатори допомагають нейтралізувати статичні заряди в повітрі, забезпечуючи захист електронних компонентів;
- Використання антистатичних спреїв на поверхнях допомагає зменшити накопичення статичної електрики.

Розглянуті заходи підвищують рівень безпеки на підприємстві і забезпечать захист працівників.

					КС КРБ 123.122.00.00 ПЗ	Акр.
						52
Зм.	Аркуш	№ докум.	Підпис	Дата		

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра було проведено аналіз поставленого завдання, що дозволило визначити ключові параметри та вимоги до системи. На основі проведеного аналізу було розроблено узагальнену структуру комп'ютерної системи, яка включає вибір апаратного та програмного забезпечення, проектування електрично-принципової схеми та інтеграцію всіх компонентів.

Було обґрунтовано вибір апаратного забезпечення, зокрема мікрокомп'ютера Raspberry Pi 4 та сенсорного дисплея ІІІ9486, що забезпечило необхідну компактність, сумісність та простоту використання. Переваги Raspberry Pi 4, такі як потужність, гнучкість налаштувань та широка підтримка з боку спільноти розробників, зробили його оптимальним вибором для цього проекту. Дисплей ІІІ9486, завдяки своїй сумісності та високій якості зображення, забезпечив надійне виведення графічної інформації.

Проаналізовано та обрано програмне забезпечення. Для відображення GLSL коду було використано glslViewer, який дозволив ефективно працювати з графічними шейдерами. Веб-сервер Apache2 та мова програмування PHP забезпечили стабільність і гнучкість веб-інтерфейсу.

У процесі проектування комп'ютерного засобу було створено електрично-принципову схему, розглянуто інтерфейс SPI для взаємодії з дисплеєм. Було також проаналізовано можливі недоліки створеної системи, що дозволило врахувати всі потенційні проблеми на етапі реалізації.

Завершальною стадією роботи стала фізична реалізація проекту. Після збору всіх необхідних компонентів і налаштування програмного забезпечення, система була успішно зібрана.

Комп'ютерна система була протестована на відображення довільного GLSL коду, що підтвердило правильність вибору компонентів і налаштувань. Дана комп'ютерна система тестування GLSL коду на базі Raspberry PI досягла поставлених цілей і може бути використаний як основа для подальших розробок у сфері обробки графіки.

					КС КРБ 123.122.00.00 ПЗ	Акр.
						53
Зм.	Аркуш	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

1. GLSL документація URL: https://www.khronos.org/opengl/wiki/Fragment_Shader (дата звернення 14.05.2024)
2. Zozulia A., Lytvynenko I., Lutsyk N., Lupenko S., Yasniy O. Method of vector rhythmcardiosignal automatic generation in computer-based systems of heart rhythm analysis. Visnyk of TNTU. 2020. Vol. 97, P. 122-132.
3. Raspberry Pi Documentation URL: <https://www.raspberrypi.com/documentation/> (дата звернення 28.04.2024)
4. GlsIViewer for linux URL: <https://github.com/patriciogonzalezvivo/glsIViewer> (дата звернення 20.05.2024)
5. Apache2 документація URL: <https://httpd.apache.org/docs/2.4/> (дата звернення 22.05.2024)
6. 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.
7. Харченко О., Яцишин В. Розробка та керування вимогами до програмного забезпечення на основі моделі якості. Вісник ТДТУ. Тернопіль, 2009. Т. 14. №1. С. 201-207.
8. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
9. Rapberry Pi 4 model B datasheet URL: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-reduced-schematics.pdf> (дата звернення 25.05.2024)
10. Php documentation URL: <https://www.php.net/manual/en/> (дата звернення 28.05.2024)
11. Робота з фреймбуфером Linux URL: <https://docs.lvgl.io/master/integration/driver/display/fbdev.html> (дата звернення 01.06.2024).

					КС КРБ 123.122.00.00 ПЗ	Акр.
Зм.	Аркуш	№ докум.	Підпис	Дата		54

12. 3.5inch RPi LCD (A) Documentation URL:
[https://www.waveshare.com/wiki/3.5inch_RPi_LCD_\(A\)](https://www.waveshare.com/wiki/3.5inch_RPi_LCD_(A)) (дата звернення 03.06.2024)

13. Introduction to the SPI Interface URL:
<https://embeddedcomputing.com/technology/software-and-os/ides-application-programming/introduction-to-spi-interface> (дата звернення 05.06.2024)

14. Офіційний сайт PuTTY URL: <https://www.putty.org> (дата звернення 07.06.2024)

15. Мохняк С.М. Безпека життєдіяльності. Навчальний посібник. – Львів: вид. НУ „Львівська політехніка”, 2009. 264 с.

16. ДСанПін 3.3.2.007-98 „Державні санітарні правила і норми роботи з візуальними дисплейними терміналами (ВДП) електронно-обчислювальних машин”.

17. Larning TNTU URL: <https://dl.tntu.edu.ua/> (дата звернення 10.06.2024)

					КС КРБ 123.122.00.00 ПЗ	Акр.
						55
Зм.	Аркуш	№ 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.

“ 09 ” 02 2024р.

Система тестування GLSL коду на базі Raspberry PI

ТЕХНІЧНЕ ЗАВДАННЯ

на 6 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня

«Бакалавр» Спеціальність 123

«Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

____ д-р. філос. наук Луцик Н.С.

“ 08 ” 02 2024р

«ВИКОНАВЕЦЬ»

Студент групи СІ-41

____ Музичук А.А.

“ 08 ” 02 2024р

Тернопіль 2024

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Система тестування GLSL коду на базі Raspberry PI».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.122.00.00

1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерної інженерії, Тернопільського національного технічного університету імені Івана Пулюя, Музичук Андрій Андрійович.

1.3 Підстава для виконання роботи

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

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання роботи – 24.04.2024р.

Плановий термін завершення виконання роботи – 28.06.2024р.

1.5 Порядок оформлення та представлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ЄСКД, ЄСПД 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.

2 Призначенні і цілі створення розробки

2.1 Призначення розробки

Система тестування GLSL коду на базі Raspberry PI є дуже простою у користуванні та інтуїтивно зрозумілою.

До складу системи повинні входити як апаратна складова, так і програмна.

2.2 Мета створення системи

Метою розробленої системи є забезпечення виведення і візуалізації GLSL коду в реальному часі за допомогою мікрокомп'ютера Raspberry Pi 4 і сенсорного дисплея ILI9486, що дозволяє інтерактивно взаємодіяти з графічними шейдерами через веб-інтерфейс.

2.3 Характеристика об'єкту

Система проектується для інформування користувача, що включає в себе:

- Розробку структурної схеми;

- Розробку електричної принципової схеми;
- Розробку блок-схеми алгоритму роботи;

3 Вимоги до системи

3.1 Загальні вимоги

Система тестування GLSL коду повинна забезпечити:

- компактність, система не повинна займати багато простору;
- низьку собівартість;
- обробку графіки в реальному часі.

3.1.1 Вимоги до структури та функціонування системи

Структура системи тестування GLSL коду:

- Raspberry PI – Основний компонент, який виконує обчисленням, зчитуванням обробкою та виводом даних.
- ili9486 – Дисплей, який відображає графіку.

3.1.2 Вимоги в функціонуванні системи

Система має безвідмовно виконувати код, якщо він сформований вірно зберігаючи синтаксис і правильно включені перемінні.

3.1.4 Вимоги до функцій та задач, які виконує система

Система має бути сумісна з мовою програмування GLSL, коректно обробляти код і відправляти оброблений код в виді графіки на дислей.

3.1.5 Вимоги до апаратного забезпечення

- Режими роботи і умови експлуатації вибраних елементів повинні відповідати вказаним в ТЗ
- Елементна база має забезпечувати необхідні режими роботи системи

4 Вимоги до документації

Документація повинна відповідати вимогам

ЄСКД та ДСТУ комплект документації складається з:

- Пояснювальної записки
- Графічного матеріалу:
 - Електрично-принципової схеми;
 - Електрично-принципової схеми GPIO пінів Raspberry PI;
 - Блок-схеми алгоритму роботи;
 - Діаграма послідовностей;

*Примітка: у комплект документації можуть вноситися зміни та доповнення в процесі розробки

5 Техніко-економічні показники

Планована собівартість проєкту повинна становити не більше 3 тисяч гривень.

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

6 Стадії та етапи проєктування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи

№ етапу	Назва виконання кваліфікаційної роботи	Термін виконання
1	Розробка та затвердження технічного завдання	01.02 – 09.02
2	Аналіз можливих рішень поставленого завдання	05.02 – 11.02
3	Розробка структурної, функціональної та електричної принципової схеми	26.04 – 04.05
4	Вибір компонентів та програмного забезпечення для комп'ютерної системи	05.05 – 29.05
5	Проектування апаратного забезпечення системи тестування GLSL коду	30.05 – 03.06
6	Реалізація програмного забезпечення комп'ютерної системи	04.06 – 09.06
7	Тестування системи тестування GLSL коду	10.06 – 15.06
8	Оформлення кваліфікаційної роботи	16.06 – 20.06
9	Попередній захист кваліфікаційної роботи	14.06
10	Захист кваліфікаційної роботи	24.06 – 28.06

7 Додаткові умови виконання кваліфікаційної роботи

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

Додаток Б
Перелік елементів

