

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Розробка ігрової приставки на базі мікроконтролера ESP32
Назва теми

КвРКІ 210373.21.03.52 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент IV курсу, група KI2-21-3 СВ Іван СТЕПАНОВ
Підпис Ім'я, прізвище

Керівник [Підпис] Марія КАПУСТЯН
Підпис, дата Ім'я, прізвище

Нормоконтролер [Підпис] Тетяна КИСІЛЬ
Підпис, дата Ім'я, прізвище

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

[Підпис] Ольга ПАВЛОВА
Підпис Ініціали, прізвище

«19» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Іван СТЕПАНОВ

Прізвище, ім'я, по-батькові студента

1. Тема проекту (роботи) Розробка ігрової приставки на базі мікроконтролера ESP32

Керівник проекту (роботи) Капустян М.В., к.т.н., доцент.

Прізвище, ім'я, по-батькові, науковий ступінь, к.т.н., доцент, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

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

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

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

Огляд існуючих ігрових приставок на базі ESP32

Проектування ігрової приставки на базі ESP32

Програмно-апаратна реалізація

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

Схема портативної ігрової приставки

Принципова схема приставки

Схема підключення мікроконтролера

6. Консультанти розділів дипломного проекту (роботи)

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

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

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Збір та аналіз літературних джерел	01.02.2025	виконано
3	Робота над розділом 1 – Дослідження апаратної частини мікроконтролера ESP32	01.03.2025	виконано
4	Робота над розділом 2 – Підбір бібліотек для графіки та звуку (FabGL та інші)	01.04.2025	виконано
5	Робота над розділом 3 – Розробка структури програмного забезпечення приставки	29.04.2025	виконано
6	Реалізація графічного інтерфейсу	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Керівник роботи

Підпис

Іван СТЕПАНОВ
Ім'я, прізвище

Марія КАПУСТЯН
Ім'я, прізвище

№	Ф	Позначення	Найменування	К	№	П
р	о			і	е	р
я	р			л	к	и
д	м			л	з	м
х	а			н		і
в	т			е		т
				с		к
				т		а
				і		
				в		
			<u>Текстові документи</u>			
1		КвРКІ 210373.21.03.52 ПЗ	Пояснювальна записка	57		
			<u>Графічні матеріали</u>			
2		КвРКІ 210373.21.03.52 Е8	Архітектура ПЗ проекту	1		
3		КвРКІ 210373.21.03.52 Е8	Архітектура ПЗ для Ігрової приставки	1		
4		КвРКІ 210373.21.03.52 Е8	Апаратне забезпечення проекту	1		

КвРКІ 210373.21.03.52 ВП				
Зм	Арж	№ докум	Підпис	Дата
Розробив	Степанов		<i>СБ</i>	
Перевір.	Капустян		<i>КК</i>	
Н. контр.	Кисіль		<i>КК</i>	19.10
Затв.	Павлова		<i>КК</i>	19.10.15
Відомість проекту			Літера	Аркуш
			У	1
			ХНУ, КІ2-21-3	

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Розробка ігрової приставки на базі мікроконтролера ESP32».

Автор роботи: Степанов Іван Романович.


Керівник роботи: Капустян Марія Вікторівна.

Пожиювальна записка: 57 с., 10 рис., 2 табл., 3 дод., 9 джерел.

Графічна частина: 3 креслення.

ESP32, МІКРОКОНТРОЛЕР, SPI ДИСПЛЕЙ, I2S АУДІО,
МІКРОКОНТРОЛЕРНА СИСТЕМА.

У даній дипломній роботі розглядається процес розробки ігрової приставки на базі мікроконтролера ESP32. Основним призначенням дослідження є створення функціонального ігрового пристрою з підтримкою ретро-ігор, з використанням обмежених апаратних ресурсів мікроконтролера. Робота містить аналіз апаратної архітектури ESP32, вибір оптимального програмного забезпечення та бібліотек для створення графіків та звуку, а також розробку інтерфейсу керування. У процесі реалізації було використано мову програмування C++, середовище розробки Arduino IDE та бібліотеку FabGL для візуалізації графіки. Результатом роботи стала працездатна ігрова приставка, яка підтримує основні функції обробки введення, виведення зображення на екран і генерації звуку. Запропоновані рішення можуть бути використані для подальших досліджень у сфері розробки інтегрованих систем та ретроігор.






Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ АПАРАТНОЇ ЧАСТИНИ	6
1.1 Огляд архітектури.....	6
1.2 Порівняння ESP32 з іншими мікроконтролерами для розробки ігрових приставок.....	8
1.3 Постановка завдання.....	20
1.4 Висновок.....	20
2 ПРОЕКТУВАННЯ ІГРОВОЇ ПРИСТАВКИ	22
2.1 Вибір апаратних компонентів (екран, контролери, живлення).....	22
2.2 Архітектура програмного забезпечення.....	29
2.3 Використання бібліотек для графіки та звуку (FabGL, Arduino GFX)..	32
2.4 Проектування інтерфейсу користувача.....	37
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	44
3.1 Налаштування середовища розробки та вибір мови програмування	44
3.3 Обробка введення від геймпада/кнопок.....	50
3.3.1 Апаратна реалізація.....	50
3.3.2 Програмна обробка введення.....	50
3.3.3 Призначення кнопок.....	50
3.4 Генерація звуку та робота з динаміком.....	51
3.4.1 Технічна реалізація.....	52
3.4.2 Генерація звуку.....	52
3.4.3 Звукові ефекти.....	52
3.5 Оптимізація продуктивності та використання пам'яті.....	52
3.5.1 Оптимізація пам'яті.....	53
3.5.2 Оптимізація продуктивності.....	54

КвРКІ 210373.21.03.52 ПЗ				
Зм.	Арк.	№ док.ум.	Підпис	Дата
Виконав		Іван Степанов		6.06.15
Перевір.		Марія КАПУСТЯН		6.06.15
Н.контр.		Тетяна КИСІЛЬ		6.06.15
Затвер.		Ольга ПАВЛОВА		10.06.15
Розробка ігрової приставки на базі мікроконтролера ESP32».				
		Літера	Аркуш	Аркушів
		у	2	57
ХНУ КІ2-21-3				

ВСТУП

Сучасні технології вбудованих систем дозволяють створити компактні та ефективні пристрої для різних сфер застосування, включаючи розважальну індустрію. Одним із популярних напрямків є розробка ігрових приставок, зокрема ретро-орієнтованих пристроїв, що імітують класичні аркадні та консольні ігри. У цьому контексті мікроконтролер ESP32 виступає привабливим рішенням за своїми технічними характеристиками, низьким енергоспоживанням та широкими можливостями підключення. Зростаюча популярність ретро-ігор та мініатюрних ігрових консолей спричинила підвищений інтерес до розробки доступних та функціональних пристроїв, які можуть запускати прості 2D-ігри. ESP32, завдяки вбудованій підтримці графіки, звуку та підключенню периферійних пристроїв, є ідеальною платформою для створення ігрової приставки. Методом цієї роботи є розробка та реалізація прототипу ігрової приставки на базі ESP32 з можливістю запуску простих ретро-ігор.

Для досягнення цієї мети необхідно вирішити такі завдання:

- дослідити апаратні можливості ESP32 для побудови ігрового пристрою;
- обрати оптимальні компоненти для керування, відображення графіки та відтворення звуку;
- розробити програмне забезпечення, яке забезпечить взаємодію користувача з пристроєм;
- протестувати та оцінити працездатність системи;
- проаналізувати можливості вдосконалення та розширення функціоналу приставки.

У процесі роботи використовувалися методи аналізу апаратного забезпечення, порівняння бібліотек для графіки та звуку, моделювання та тестування роботи мікроконтролера. Для програмування використовувалися мова C++, середовище Arduino IDE, а також бібліотеки FabGL та Arduino GFX для реалізації графіки. Наукова новизна дослідження полягає в розробці методики

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

оптимізації роботи ESP32 для створення ігрових пристроїв. Практичне значення роботи полягає у можливості використання отриманих результатів для подальших досліджень у сфері вбудованих систем, а також для створення навчальних і комерційних проектів у сфері електроніки та програмування. Робота складається з вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглянуто теоретичні основи та особливості мікроконтролера ESP32. У другому розділі описано процес проектування апаратної та програмної частини ігрової приставки. Третій розділ присвячений реалізації програмного забезпечення та тестування.

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ АПАРАТНОЇ ЧАСТИНИ

1.1 Огляд архітектури

ESP32 - це високопродуктивний мікроконтролер із підтримкою Wi-Fi та Bluetooth, розроблений компанією Espressif Systems. Він є наступником популярного ESP8266 і володіє розширеним функціоналом, що робить його ідеальним рішенням для IoT-проектів, бездротових мереж, промислової автоматизації та вбудованих систем, зокрема, ігрових приставок. Основні переваги ESP32 - він оснащений двоядерним процесором із тактовою частотою до 240 МГц, має вбудовану підтримку Wi-Fi 802.11 b/g/n та Bluetooth 4.2. У цього модуля оперативна пам'ять до 520кб, також є можливість використання зовнішньої пам'яті. Підтримайте кілька контролерів таких як: SPI, I2C, UART, PWN, ADC, DAC. Також забезпечений сенсорними датчиками, має низьке енергоспоживання та підтримує різні режими сну. Архітектура процесора EPS32 побудована на основі двоядерного 32-бітного процесора Tensilica Xtensa LX6, який працює на тактовій частоті до 240 МГц. Є також версії з одноядерним процесором, наприклад ESP32-S0WD. Основні особливості процесора - Його подвійне ядро дозволяє виконувати незалежні задачі, наприклад: одне ядро може обробляти введення, а інше ядро – графічну частину. ESP32 підтримує FreeRTOS, що дозволяє ефективно керувати завданнями – це потрібно для більш ефективного управління задачами, а також оснащений інструкцією DSP для швидкої обробки сигналів, що корисно для аудіо та графіки. ESP32 має декілька типів пам'яті:

- SRAM (520КБ) – його необхідність для роботи програм та обробки даних;
- ROM (448КБ) – містить стартовий код та системні бібліотеки;
- FLASH - пам'ять (4-16 МБ, залежно від моделі) – для зберігання програм, файлів та графічних ресурсів;
- зовнішній PSRAM (до 8 МБ, у деяких великих моделях) – вибір для графічних буферів та ресурсомістких додатків;

					КВРКІ 210373.21.03.52 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

– графічні можливості та відображення зображення, оскільки ESP32 не має апаратного GPU, графіка обробляється на програмному рівні. Для візуалізації виберіть декілька бібліотек.

ESP32 є універсальним мікроконтролером із його широкими можливостями для роботи з графікою, звуком та пристроями введення, що робиться придатним для створення ігрових пристроїв.

Підтримка дисплеїв та графіки:

- FabGL – потужна бібліотека для роботи з VGA-виводом, яка підтримує як текстовий, так і графічний режими;
- Arduino GFX – універсальна бібліотека, яка використовується для керування дисплеями через SPI та I2S інтерфейси;
- TFT_eSPI – спеціалізована бібліотека для роботи з TFT-дисплеями через SPI, що забезпечує високошвидкісне оновлення зображення.

Типи сумісних дисплеїв:

- OLED-дисплеї (наприклад, SSD1306, SH1106) – енергоефективні чорно-білі дисплеї з інтерфейсами SPI або I2C;
- TFT-дисплеї (наприклад, ILI9341, ST7735, ST7789) – кольорові дисплеї з підключенням через SPI, здатні відображати графіку з високою деталізацією.

Підтримка аудіо:

ESP32 має кілька способів генерації та виведення звуку:

- I2S – цифровий аудіоінтерфейс для підключення зовнішніх ЦАП або аудіочіпів, таких як MAX98357;
- DAC – два вбудованих 8-бітних цифро-аналогових перетворювача для простого аналогового виводу звуку.
- PWM – широтно-імпульсна модуляція, яка використовується для генерації простих звукових ефектів.

Підключення пристроїв введення:

- ESP32 має до 36 універсальних GPIO-портів, які можна підключити;
- кнопки та геймпади напряму (через GPIO або інтерфейс I2C);

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

- ємнісні сенсори для безконтактного введення;
- SPI/I2C-контролери, зокрема сумісні з NES та SNES.

Режими енергоспоживання:

ESP32 підтримує кілька режимів енергозбереження, які особливо важливі для портативних пристроїв:

- Активний режим (до 240 МГц) – максимальна продуктивність;
- Light Sleep Mode – центральний процесор зупиняється, але периферійні пристрої продовжують працювати;
- Deep Sleep Mode – активні залишаються лише критично пошкоджені функції, споживання знижується до 10 мкА;
- Глибокий сон із пробудженням – пристрій майже не споживає енергію, пробудження можливо за таймером або сигналом з кнопки.

1.2 Порівняння ESP32 з іншими мікроконтролерами для розробки ігрових приставок

У цьому розділі ми порівнюємо кілька контролерів: ESP32, ESP8266.

ESP32 та ESP8266 – це не дорогі SOC на базі Wi-Fi (System on Chip), вони ідеально підходять для проектів DIY в інтернет-речах. Обидва контролера мають 32-розрядні процесори. Візьмемо до уваги ESP32 – двоядерний процесор з частотою від 80МГц до 240МГц, а ESP8266 – має тільки одноядерний процесор з частотою 80МГц. Ці модулі мають кілька GPIO (введення/виведення загального призначення), вони підтримують різні протоколи, такі як SPI, I2C, UART, ADC, DAC та PWM. Щоб ці два мікроконтролера працювали коректно, вони потребують напруги 3.3В. Відлагоджувальних плат на базі ESP8266 існує безліч. Вони між собою відрізняються розмірами та додатковою зв'язкою – типом USB об'ємом та стабілізацією напруги, такими мікросхемами, як ESP8266, та кількома доступними виводами GPIO. В даному порівнянні візьмемо одну із популярних у накладних платі на базі ESP8266 NodeMCU. Модуль ESP-12E на платі містить мікросхему

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ESP8266 з 32-розрядним RISC-мікропроцесором Tensilica Xtensa LX106, який підтримує RTOS і працює з тактовою частотою від 80 до 160 МГц. Nodemcu ESP8266 включає приймач Wi-Fi стандарту 802.11b/g/n HT40, що дозволяє підключатися до мережі Wi-Fi і взаємодіяти з Інтернет-модулем, який може функціонувати як точка доступу, так і станція. Цей модуль має досить потужні можливості вбудованої обробки та зберігання даних, що дозволяє інтегрувати його з датчиками та іншими пристроями, спеціальними для конкретної програми, через GPIO з мінімальною попередньою розробкою та мінімальним завантаженням під час виконання. Він має 128 КБ ОЗУ та 4 МБ флеш-пам'яті (для зберігання додатків та даних), що більш ніж достатньо для обробки величезних рядків, які створюють веб-сторінки, дані у форматі JSON/XML та все інше, що ми використовуємо в пристроях Інтернету речей у наші дні. Модуль ESP8266 NodeMCU доступний у двох варіантах, де один побудований з мостом CP2102 USB to UART, а інший з мостом CH340 USB to UART. Плата оснащена регулятором напруги LDO для підтримки стабільної напруги на рівні 3,3 В, у той час як робочий елемент напруги ESP8266 становить від 3 до 3,6 В. Коли ESP8266 споживає до 80 мА під час радіочастотної передачі, регулятор може надійно подавати до 600 мА, чого має бути більш ніж достатньо. Вихід регулятора також наведено на одній із сторінок плати та позначений як 3V3. Через цей висновок можна подавати живлення зовнішніх компонентів. Вбудований роз'єм MicroB USB забезпечує живлення ESP8266 NodeMCU. Ви можете використовувати висновок VIN для живлення ESP8266 та його периферійних пристроїв разом, якщо живлення дається з джерела 5В. Для зв'язку ESP8266 потрібно джерело живлення 3,3 В та логічні рівні 3,3 В. Контакти GPIO не мають допуску 5 В. Модуль Wi-Fi ESP8266 має загалом 17 висновків GPIO, доступних по обидва боки накладної плати. Ці контакти можуть бути призначені для всіх видів периферійних функцій.

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

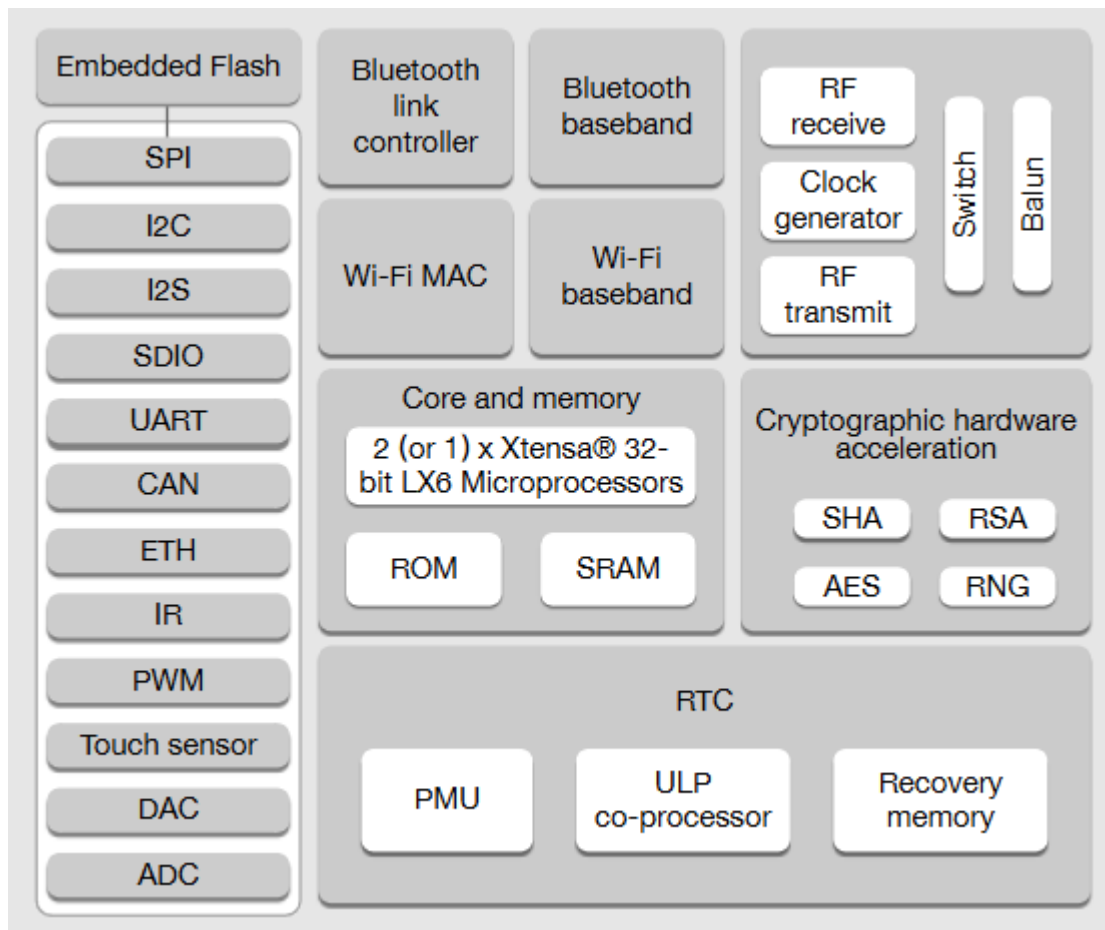


Рисунок 1.1 – Блок діаграма ESP32

Мікроконтролер має один вивід VIN та три виводи 3,3 В. Якщо джерело живлення дає 5 В, для живлення ESP8266 та периферійних пристроїв слід використовувати VIN. Виводи 3,3 Включені до вбудованого стабілізатора напруги та можуть використовуватися для подачі живлення на зовнішні компоненти. Контакти I2C: вони використовують для підключення всіх датчиків I2C вашого проекту та периферійних пристроїв. Підтримується як провідний, і ведений інтерфейси I2C.

Контакти GPIO: ESP8266 NodeMCU зображена на Рисунку 1.2 та містить 17 контактів GPIO, які можуть бути програмно призначені для різних завдань, таких як I2C, I2S, UART, PWM, ІЧ-пульт дистанційного керування, світлодіодне підсвічування та кнопки.

– Заземлення: це контактне заземлення накладної плати ESP8266 NodeMCU;

- Канал АЦП: У NodeMCU вбудований АЦП SAR із 10-розрядною точністю. Тестування напруги живлення на виводі VDD3P3 та тестування вхідної напруги на виводі TOUT – це дві функції, які можуть бути реалізовані за допомогою АЦП. Вони не можуть бути реалізовані одночасно;

- контакти UART: ESP8266 NodeMCU оснащений двома портами UART, UART0 та UART1, які можуть взаємодіяти зі швидкістю до 4,5 Мбіт/с та забезпечують асинхронний зв'язок (RS232 та RS485). Для зв'язку можна використовувати UART0 (контакти TXD0, RXD0, RST0 та CTS0). Він підтримує контроль рідини. Проте наступний UART1 (висновок TXD1) передає лише дані, він зазвичай використовується для друку журналів;

- контакти SPI: ESP8266 має два SPI (SPI та HSPI), які можна використовувати в режимах slave та master;

- контакти ШІМ: на платі є чотири канали широтно-імпульсної модуляції (ШІМ). Вихід ШІМ може бути програмно реалізований та використаний для управління цифровими двигунами та світлодіодами. Діапазон частот ШІМ регулюється від 1000 до 10000 мкс, тобто від 100 Гц до 1 кГц;

- контакти управління: використання для керування мікроконтролером ESP8266. Серед цих контактів - виведення включення чіпа (EN), виведення скидання (RST) та виведення пробудження;

ESP8266 NodeMCU має дві кнопки. Кнопка скидання, позначена RST і розташована у верхньому лівому кутку, використовується для скидання чіпа ESP8266. Кнопка FLASH розташована в нижньому лівому кутку та використовується для оновлення прошивки. На платі також є запрограмований світлодіодний індикатор, який підключається до контакту D0.

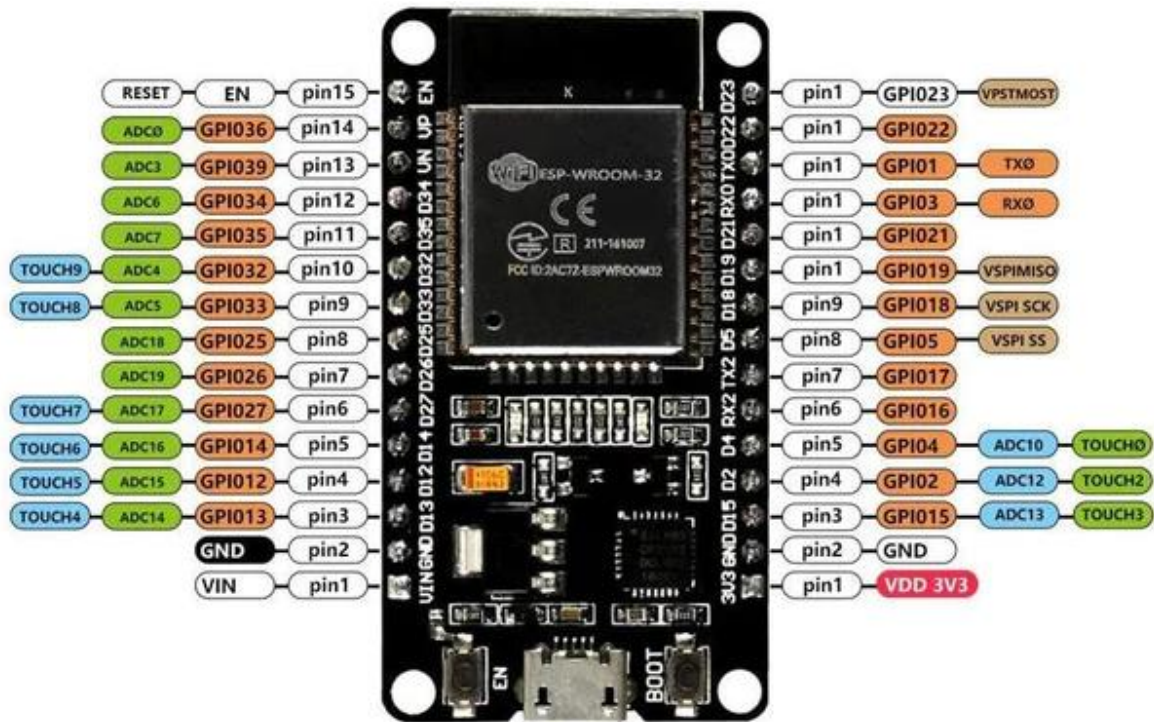


Рисунок 1.2 – Плата ESP8266

ESP8266 – дуже популярна та доступна платформа для реалізації енергоефективних програм Інтернет-речей, які працюють на основі підключення Wi-Fi. У своєму зверненні Espressif ESP32 є більш новим і більш просунутим рішенням, в якому розробники збільшили швидкість Wi-Fi, додали підтримку Bluetooth 4.2 і Bluetooth Low Energy, а також збільшили кількість входів/виходів. ESP32 має більше GPIO, ніж ESP8266, і ви можете вказати, які контакти використовують для UART, I2C та SPI в кодї. Це можливо з можливістю мультиплексування чіпа ESP32, що дозволяє відображати багато функцій одному контакту. Сигнали PWM можуть бути встановлені в будь-якому GPIO з частотами, що настроюються, і робочими циклами коду. Аналогові контакти статичні, але ESP32 підтримує вимірювання на 18 каналах (аналогові контакти), тоді як ESP8266 має тільки один 10-розрядний висновок АЦП. Два 8-розрядні канали ЦАП також підтримуються ESP32. Крім того, ESP32 містить 10 GPIO з ємним датчиком, які розпізнають дотик і можуть використовуватися для запуску подій або пробудження ESP32 від глибокого сну. ESP32 підтримує протокол Bluetooth для промовчання, в той час як ESP8266 цього не може. Помітною відмінністю ESP32 є його функція

Bluetooth, яка дозволяє ESP32 не обмежуватися тільки зв'язком Wi-Fi, дозволяючи інтегрувати його в більшу кількість проектів. Він підтримує як класичний Bluetooth, так і Bluetooth із низьким енергоспоживанням. У той час, як ESP8266 не підтримує Bluetooth. ESP8266 має вбудований процесор, але через багатозадачність, пов'язану з оновленням стека Wi-Fi, більшість додатків використовує окремий мікроконтролер для взаємодії з датчиками, цифрового введення-виведення та обробки даних. При використанні ESP32 вам не знадобитися використовувати додатковий мікроконтролер, залишивши ESP32 оснащений двома 32-розрядними мікропроцесорами і працювати на розподільчих платах і модулях з частотою від 160 МГц до 240 МГц. Це забезпечує достатню швидкість для будь-якої програми, яка потребує мікроконтролера з можливістю підключення:

- ESP32 швидше, ніж ESP8266;
- ESP32 поставляється з великою кількістю GPIO з багатьма функціями;
- ESP32 підтримує аналогові вимірювання на 18 каналах (аналогові контакти) відповідно до одного 10-розрядним виведенням АЦП на ESP8266; ESP32 підтримує аналогові вимірювання на 18 каналах (аналогові контакти) відповідно до одного 10-розрядним виведенням АЦП на ESP8266;
- ESP32 підтримує Bluetooth, а ESP8266 - ні;
- ESP32 підтримує Bluetooth, а ESP8266 - немає;
- ESP8266 дешевше, ніж ESP32;
- у ESP8266 ширша підтримка спільноти;
- для багатьох проектів IoT та Wi-Fi ESP8266 можуть виконувати цю роботу за нижчою ціною;
- обидві плати можна запрограмувати за допомогою Arduino IDE або інших підтримуваних IDE.

Обидві плати підтримують MicroPython.

					КВРКІ 210373.21.03.52 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1 – Порівняння мікроконтролерів

СПЕЦИФІКАЦІЯ	ESP8266	ESP32
Мікроконтролер	Одноядерний 32-розрядний процесор Xtensa L106	двоядерний 32-бітний процесор Tensilica Xtensa LX6
802.11 b / g / n Wi-Fi	Так, HT20	Так, HT40
АЦП	10-розрядний	12-розрядний
Апаратний/програмний ШІМ	Ні / 8 каналів	1/16 канала
Типова частота	80МГц	160МГц
SRAM	160 КБ	512 КБ
GPIO	17	36
Сенсорний датчик	Ні	Так
Bluetooth	Ні	Bluetooth 4,2
SPI / I2C / I2S/UART	2/1/2/2	4/2/2/3
CAN	Ні	Так
ПЗУ	Немає програмованого	448 КБ ПЗУ для завантаження та основних функцій
Робоча температура	від -40 °С до 125 °С	від -40 °С до 125 °С

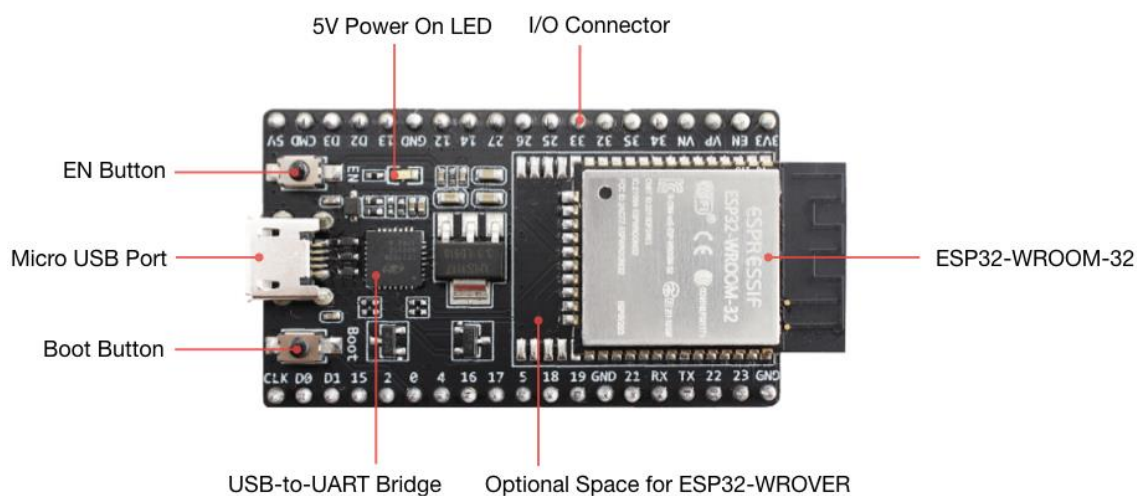


Рисунок 1.3 – Плата ESP32

Підсумовуючи цей короткий огляд, можна зробити висновок, що мікроконтролер ESP8266 є доступним та функціональним рішенням для реалізації бездротового зв'язку через Wi-Fi. Однак у разі потреби в більшій енергоефективності та підтримці Bluetooth доцільно враховувати його наступника — модуль ESP32. Обидва мікроконтролери, ESP8266 та ESP32, забезпечують можливість підключення пристроїв до Інтернету, проте ESP32 пропонує розширені функціональні можливості, що робить його більш оптимальним вибором у багатьох випадках. Водночас обидва модулі залишаються гідними представниками своєї категорії. Серія інтелектуальних модулів дисплея на ESP32, розроблена та виготовлена компанією 4D Systems. Ці модулі відображення доступні в розмірах 2,4", 2,8", 3,2" і 3,5", що пропонують інтерфейс SPI між процесором ESP32-S3R8 і РК-дисплеями TFT. Дисплеї 2,4", 2,8" і 3,5" є IPS TFT LCD, а 3,2" є TN TFT LCD. Доступний у безсенсорному режимі, ємнісному сенсорному та ємнісному сенсорному з рамкою об'єктива (CLB). Resistive Touch наразі недоступний через обмеження місця на друкованій платі для контролера RTP. Інтерфейс користувача серії ESP32 — це 30-контактний роз'єм FPC/ZIF, призначений для 30-контактного кабелю FFC із кроком 0,5 мм для легкого та простого підключення до програми чи материнської плати або для підключення до додаткової плати для цілого ряду вдосконалених функцій. Ця серія плат сумісна з IDE 4D Systems Workshop4, використовуючи компілятор Espressif і бібліотеки спеціального призначення 4D Systems, що забезпечує багатфункціональний дизайн і досвід програмування. Будь-який код, розроблений і написаний для роботи на інших модулях відображення 4D Systems, таких як модулі з графічними процесорами Goldelox, Picaso, Pixxi або Diablo16, на жаль, не сумісні з функцією gen4-ESP32 через те, що це зовсім інше сімейство процесорів. Системи, після яких між ними є певна схожість, наприклад графіка, проте більшу частину кодування всього адаптувати. З механічної точки зору ці модулі gen4-ESP32 фізично мають той самий монтажний розмір, який і інші продукти gen4 від 4D Systems. Єдина відмінність порушена у використаній схемі. Загальна товщина цих модулів gen4-ESP32 більша, ніж в інших

					КВРКІ 210373.21.03.52 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

продуктах gen4, завдяки роз'єму USB-C. Як правило, там був встановлений інший модуль gen4, модуль gen4-ESP32 міг підійти в те саме місце.

Типи дисплеїв, які підтримує ESP32:

- TFT-дисплеї - дозволяє малювати в кольорі, використовувати анімацію;
- OLED-дисплеї - підтримують графіку, шрифти, анімацію;
- E-Ink (E-Paper) дисплеї - використовуються для створення енергоефективних інформаційних панелей;
- RGB LED-матриці (WS2812, HUB75) - ESP32 може генерувати VGA-сигнал для підключення до монітора.

Звук за своєю природою є аналоговим, і до розвитку цифрового звуку аудіообладнання також було аналоговим. Коливання звуку на перетворювачі, наприклад мікрофоні, можна посилити, а потім посилити до динаміки, дифузор якого відтворює ці вібрації. Передача звуку з мікрофона на динаміку, особливо ви хочете записати його та відтворити на динаміку пізніше, потребувала багато аналогової електроніки. Крім того, найкраща аналогова електроніка викликає електричний шум і спотворення сигналу, хоча сучасні конструкції вражають чистіше. Цифрове аудіо рекламувалося як спосіб усунути ці спотворення звуку. За допомогою цифрового аудіо звук надсилається до аналого-цифрового перетворювача (АЦП), щоб створити його цифрове представлення. Потім це можна зберегти або передати без будь-якого недоліку. З іншого боку, цифровий сигнал проходить через еквівалентний цифро-аналоговий перетворювач (DAC), який створює аналоговий вхід, а потім використовується для керування динамікою. Звичайно, цифровий аудіосигнал не є ідеальним відображенням оригінального сигналу. Якість сигналу залежить від двох параметрів, які застосовуються як до АЦП, так і до ЦАП. Роздільна здатність – кількість бітів, використаних у виборі. Більше бітів означає кращу якість. Частота вибору – скільки вибору ми беремо в секунду. Це має бути принаймні вдвічі вище найвищої частоти, яку ми хочемо відібрати. Очевидно, що чим вище роздільна здатність і частота дискретизації, тим більше буде файл цифрових даних. Цифровий аудіо компакт-диск має роздільну

					КВРКІ 210373.21.03.52 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

здатність 16 біт і частоту дискретизації 44,1 кГц, тоді як якість цифрового аудіо телефону має 8 біт і дискретизацію 8 кГц. Коли цифрове аудіо передається по всьому світу або між інтегральними схемами, це робиться в системному форматі. Існує кілька форматів, розширених є імпульсно-кодова модуляція або PCM. I2S працює з цифровими даними PCM, використовуючи будь-яку роздільну здатність і частоту дискретизації. Його можна використовувати для маршрутизації джерела звуку до місця через різні сигнальні процесори та еквайзери в деяких випадках. Протокол I2S керує даними PCM на шині, яка складається з наступних трьох ліній підключення:

- SCK – лінія послідовного тактового сигналу, яку іноді називають «лінією бітового тактового сигналу»;
- WS – Word Select, який вибирає між левим і правим аудіоканалами;
- SD – Наступні дані, аудіодані PCM;

Якість аудіосигналу визначається після такої частоти та збільшення за такою формулою: частота дискретизації * біт на канал * кількість каналів – $44,1 \text{ кГц} * 16 * 2 = 1,4112 \text{ МГц}$. Отже, якщо потрібно надіслати два канали високоякісного звуку, знадобиться така частота 1,4112 МГц. Пристрої, підключені до шини I2S, можна розділити на дві категорії: контролер – керує сигналами SCK і WS; ціль – потім сигнали SCK і WS.. На малюнку 4 аудіопередавача також є контролером I2S і забезпечує сигнали SCK, WS і SD. Приймач є I2S Target.



Рисунок 1.4 – Схема аудіопередавача

ESP32 має два периферійних пристрої I2S, I2S0 та I2S1. Кожен з них може бути налаштований як контролер або ціль, і кожен може бути аудіопередавачем або приймачем. INMP441 — розширений і недорогий мікрофонний модуль I2S. Він використовує мікрофон MEMS (мікроелектромеханічних систем) і має внутрішній 24-розрядний аналого-цифровий перетворювач та інтерфейс I2S. Мікрофонний модуль INMP441 має такі характеристики:

- всеспрямований відгук;
- 24-розрядний інтерфейс I2S;
- співвідношення сигнал/шум 61 дБА;
- АЧХ 60 Гц – 15 кГц.
- він має такі зв'язки:
- SD – послідовне з'єднання даних I2S;
- VDD – вхідна напруга від 3 до 6 вольт;
- GND – земля;
- L/R – вибір каналу;
- WS – Вибір слова;
- SCK – серійний годинник;
- вибір каналу (контакт L/R) працює таким чином;
- ліво – Л/П підключено до GND;
- право – L/R підключений до VDD.

Одна особливість цієї функції та створення мікрофонних модулів I2S MEMS утворюється в тому, що звук надходить у мікрофон з нижньої частини друкованої плати. Ви побачите маленьку піктограму мікрофона поруч із невеликим отвором, це місце, де звук надходить у мікрофон. Переконайтеся, що ви монтуєте свій модуль так, щоб звук міг проникнути в мікрофон, зазвичай для цього потрібно прикріпити контакти «догори дном». Ця конструктивна особливість пов'язана з тим, що мікрофон MEMS (Micro-Electro-Mechanical Systems) використовує невеликий діафрагмовий отвір, розташований на нижній стороні корпусу, для

більш точного відтворення звукових хвиль. Завдяки цьому досягається краща чутливість та захист від шумів, які можуть виникати через повітряні потоки або механічні перешкоди на верхній частині плати. При встановленні таких модулів важливо:

- забезпечити відкритий простір під отвором мікрофона – модуль припаяти занадто щільно до поверхні або закрити отвір, звук не зможе потрапити в середину, якщо значно зменшить якість запису;

- уникати контакту з пилом та вологістю – наступний відвір мікрофона відкритий, частки пилу або конденсат можуть увійти в його роботу;

- необхідно звернути увагу на правильне розташування під час монтажу – у деяких випадках, наприклад, у вбудованих аудіосистемах або портативних пристроях, можна знадобитися спрямований звуковий канал (акустичний порт) для покращення спрямованості запису;

- слід також отримати, що такі мікрофони часто мають два варіанти виходу – лівий (L) та правий (R) канал, що дозволяє використовувати їх у стереосистемах. При роботі з протоколом I²S варто правильно налаштувати сигнал WS (Word Select), щоб ESP32 правильно отримував дані з потрібного мікрофона або обробляв стереопотік.

- сигнал WS визначає, який канал (лівий чи правий) передається в поточний момент: зазвичай, низький рівень (LOW) сигналу відповідає лівому каналу, а високий рівень (HIGH) - правому. Залежно від типу мікрофона (наприклад, MEMS мікрофонів з підтримкою I²S), канал визначається апаратно або через конфігураційний пін (наприклад, L/R select);

- якщо використовується лише один мікрофон, важливо переконатися, що його канал відповідає налаштуванням I²S периферії ESP32, щоб уникнути ситуації, коли дані читаються з порожнього каналу. У разі стереоконфігурації необхідно реалізувати обробку обох каналів – наприклад, зберігаючи або аналізуючи їх окремо для подальшої обробки аудіосигналу або просторового позиціонування звуку.

1.3 Постановка завдання

У рамках цього дослідження поставлено завдання проаналізувати можливість використання мікроконтролера ESP32 як апаратної платформи для створення простої ігрової приставки. Для досягнення цієї мети необхідно здійснити технічний огляд архітектури ESP32, зокрема його процесорного ядра, периферії, можливостей енергоспоживання та комунікаційних інтерфейсів. Також потрібен порівняльний аналіз ESP32 з іншими популярними мікроконтролерами, які можна використовувати для створення подібних пристроїв, та дослідити вбудовані можливості ESP32 в обробці контексту, генерації графіків звуку та роботи з пристроями введення, що критично створюється для реалізації ігрового функціоналу. Результати даного аналізу не дозволяють обґрунтувати доцільність вибору ESP32 як основу для проектування функціональної, доступної та енергоефективної ігрової програми.

1.4 Висновок

У першому розділі було проведено детальний огляд апаратної платформи ESP32, її архітектурних особливостей, а також можливостей у порівнянні з іншими мікроконтролерами, які використовують у подібних проектах. Аналіз показав, що ESP32 має достатню потужність для реалізації базових графічних і звукових функцій, а також підтримує широкий вибір периферійних інтерфейсів, деякі для роботи з елементами керування. Його перевагами є наявність двох ядер, підтримка Wi-Fi і Bluetooth, вбудовані DAC і PWM, а також підтримка популярних графічних бібліотек, таких як LVGL або FabGL. Усе це робить ESP32 привабливим вибором для побудови прототипу недорогої ігрової приставки. На основі проведеного аналізу можна зробити висновок, що ESP32 є ефективною платформою для створення інтерактивних мікроконтролерних ігрових систем середнього рівня складності.

					КВРКІ 210373.21.03.52 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ ІГРОВОЇ ПРИСТАВКИ

2.1 Вибір апаратних компонентів (екран, контролери, живлення)

Найкращим варіантом для розробки ігрової приставки є ESP32-WROVER, оскільки він має PSRAM (Pseudo Static Random Access Memory) - це тип оперативної пам'яті, який впливає на особливості DRAM (Dynamic RAM) і SRAM (Static RAM). Основна ідея PSRAM виникає в тому, щоб використати переваги обох типів пам'яті, забезпечуючи більш високу продуктивність та енергоефективність.

Основні особливості PSRAM:

PSRAM використовує комірки пам'яті, аналогічні DRAM, що робить її компактнішою і дешевшою у виробництві в порівнянні з SRAM;

як і DRAM, дані в PSRAM вимагають періодичного оновлення (регенерації), щоб зберегти їх;

- На відміну від DRAM, де оновлення даних має керуватись зовнішнім контролером, PSRAM має вбудований контролер оновлення. Це робить її більш схожою на SRAM з погляду простоти використання, оскільки не потрібне зовнішнє керування оновленням;

- PSRAM має інтерфейс, аналогічний SRAM, що спрощує її інтеграцію до системи, де потрібна висока швидкість доступу до пам'яті;

- PSRAM зазвичай споживає менше енергії, ніж DRAM, завдяки вбудованому контролеру оновлення та оптимізованій архітектурі;

- PSRAM часто використовується у пристроях з обмеженими ресурсами, таких як мобільні телефони, IoT-пристрої, вбудовані системи та інші портативні пристрої, де важливе низьке енергоспоживання та компактність.

- вища щільність пам'яті порівняно з SRAM;

- нижча вартість у порівнянні з SRAM;

- спрощене керування в порівнянні з DRAM;

- найменша швидкість доступу в порівнянні з SRAM;

- вимагає більше енергії, ніж SRAM, але менше, ніж DRAM.

					КВРКІ 210373.21.03.52 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Тип пам'яті PSRAM є компромісним рішенням, яке підходить для пристроїв, де важливі баланс між продуктивністю та енергоспоживанням. Завдяки поєднанню характеристик динамічної (DRAM) та статичної (SRAM) пам'яті, PSRAM забезпечує помірну швидкість при зниженому енергоспоживанні, що робить її оптимальною для портативних пристроїв, таких як мобільні телефони, носімі гаджети, а також мікроконтролери в системах реальної години. Крім того, вона не потребує складної логіки для оновлення, як DRAM, та має менші розміри кристала в порівнянні із SRAM, що дозволяє зменшити загальні витрати на апаратне забезпечення при збереженні достатнього рівня ефективності для багатьох вбудованих рішень. PSRAM зазвичай використовують у пристроях, де традиційна DRAM була б надмірною з точки зору енергоспоживання, а SRAM — надто дорогою з точки зору вартості на один біт. Це стосується, зокрема, систем типу ESP32, де PSRAM застосовується як додаткова оперативна пам'ять для зберігання графіки, буферів звуку, стеків для багатопотокової обробки тощо. Найкращий варіант буде використовувати дисплей - TFT LCD (SPI, ILI9341 320x240)

Характеристики дисплея:

Його роздільна здатність і глибина кольору – від 128x128 до 320x380 пікселів, він підтримує глибину кольору до 16біт – 65.536 кольорів або 18біт – 262.144 кольорів, дисплей забезпечує чітке та деталізоване зображення. Ці дисплеї зазвичай використовують SPI (Serial Peripheral Interface) або паралельні інтерфейси для зв'язку з ESP32, однак SPI надає перевагу через його простоту та меншу кількість контактів, що полегшує інтеграцію в проекти з обмеженою кількістю контактів GPIO.

Багато РК-дисплеїв TFT ESP32 оснащені ємними або резистивними сенсорними екранами, які дозволяють користувачу взаємодіяти за допомогою сенсорного введення - ця функція особливо корисна для створення інтуїтивно зрозумілих інтерфейсів користувача.

					КВРКІ 210373.21.03.52 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

TFT РК-дисплей пропонує високі рівні яскравості та коефіцієнти контрастності, забезпечуючи чітку видимість навіть у хорошому освітленому середовищі.

Однак регулювання підсвічування дозволяє оптимізувати його залежно від умов навколишнього освітлення.

Під час створення ретроігрової консолі з використанням мікроконтролера ESP32 або подібного, вибір елементів керування має вирішальне значення для відтворення автентичного досвіду гри. Ретро-ігри зараз складаються на кнопках чи джойстиках, або їх комбінації, щоб забезпечити інтуїтивно зрозуміле та швидке введення. Нижче наведено детальний опис цих пристроїв керування та способів їх реалізації в ретро-проекті ігрової консолі.

У стандартній ігровій приставці потрібно аналоговий джойстик — це універсальний пристрій введення, який забезпечує точне керування шляхом вимірювання ступеня нахилу за двома осями (X і Y). Він традиційно використовується в ігрових контролерах, робототехніці та інших програмах, які потребують детального введення. У поєднанні з аналоговим джойстиком ESP32 можна використовувати для створення інтерактивних проєктів, таких як ретро-ігрові консолі, дистанційно керовані роботи або пристрої спеціального введення.

Характеристики аналогової джойстики:

Аналоговий характер джойстика забезпечує плавне та безперервне введення, що ідеально підходить для таких програм, як ігри, робототехніка та дистанційне керування.

- джойстик забезпечує незалежні аналогові виходи як для осі X (горизонтальна), так і для осі Y (вертикальна);
- забезпечує двовимірне керування напрямком, що робить його придатним для ігор, навігації по меню та роботизованого руху;
- коли джойстик знаходиться по центру, значення X і Y знаходяться в середній точці (наприклад, ~2047 для 12-розрядного АЦП);

- аналогових джойстиків мають вбудовану кнопку (вісь Z), яку можна натиснути клацанням джойстика;
- кнопку можна використовувати для таких дій, як вибір параметрів, стрибки в іграх або активація спеціальних функцій;
- кнопку можна підключити до будь-якого цифрового контакту GPIO;
- аналогові виходи джойстика можна відкалібрувати для врахування коливань нейтрального положення та діапазону руху;
- необроблені значення АЦП можна відобразити в потрібному діапазоні (наприклад, від -100 до 100) для легшої інтерпретації та використання в кодї;
- аналогові джойстики споживають дуже мало електроенергії, що робить їх придатними для проектів, що працюють від батарейок;
- допускається використання джойстика як керуючий контролер для ретро-ігрових консолей або індивідуальних ігрових проектів;
- він забезпечує точне керування переміщеннями робота, наприклад керуванням роботом або рукою робота;
- за допомогою джойстика можна керувати дронами, автомобілями на радіоуправлінні та іншими пристроями з дистанційним керуванням.
- аналогові джойстики компактні і їх можна легко інтегрувати в невеликі проекти;
- вони розроблені таким чином, щоб витримувати часте використання, що робить їх надійними для тривалого використання.

Мікроконтролер ESP32 має достатньо GPIO (до 34 виводів у різних моделях), що дозволяє підключати кнопки, джойстики, дисплеї, динаміки та інші периферійні пристрої. Він підтримує аналогові входи (ADC) для зчитування аналогових сигналів, наприклад, від потенціометрів або аналогових джойстиків, що робить його ідеальним для побудови інтерактивного ігрового керування.

Завдяки апаратним перериванням, ESP32 може миттєво реагувати на дії користувача — натискання кнопки, зміну сигналу, що критично важливо у динамічній середовищі, а саме у відеоіграх. Підтримка PWM (широко-імпульсної

модуляції) дозволяє плавно регулювати яскравість підсвічування дисплея, керувати двигунами або створювати звукові сигнали.

Крім того, ESP32 має вбудовані модулі Bluetooth та Wi-Fi, що відкриває широкі можливості для створення бездротових геймпадів, локального або онлайн мультіплеєра, хмарного збереження грового прогресу або оновлення прошивки «по повітрю» (OTA).

Його двоядерний процесор Tensilica Xtensa LX6 із тактовою частотою до 240 МГц, а також підтримка багатозадачності через FreeRTOS, дозволяють ефективно розділяти завдання: обробку введення, рендеринг графіки, відтворення звуку, обмін даними по мережі — все це може відбуватися паралельно, без затримок для користувача.

Таблиця 2.1 – Основні сигнали I²S та їх призначення

Назва сигналу	Призначення	Напрямок (ESP32)	Рівні сигналу / Стани	Опис
BCLK (Bit Clock)	Бітова синхронізація	Вихід	Частота залежить від частоти дискретизації	Синхронізує передачу кожного біта даних
DOUT (Data Out)	Вихід даних з мікрофона	Вхід	Потік цифрових бітів	ESP32 приймає цифрові аудіодані від мікрофона
DIN (Data In)	Вхід даних до ЦАП	Вихід	Потік цифрових бітів	Якщо передається аудіо

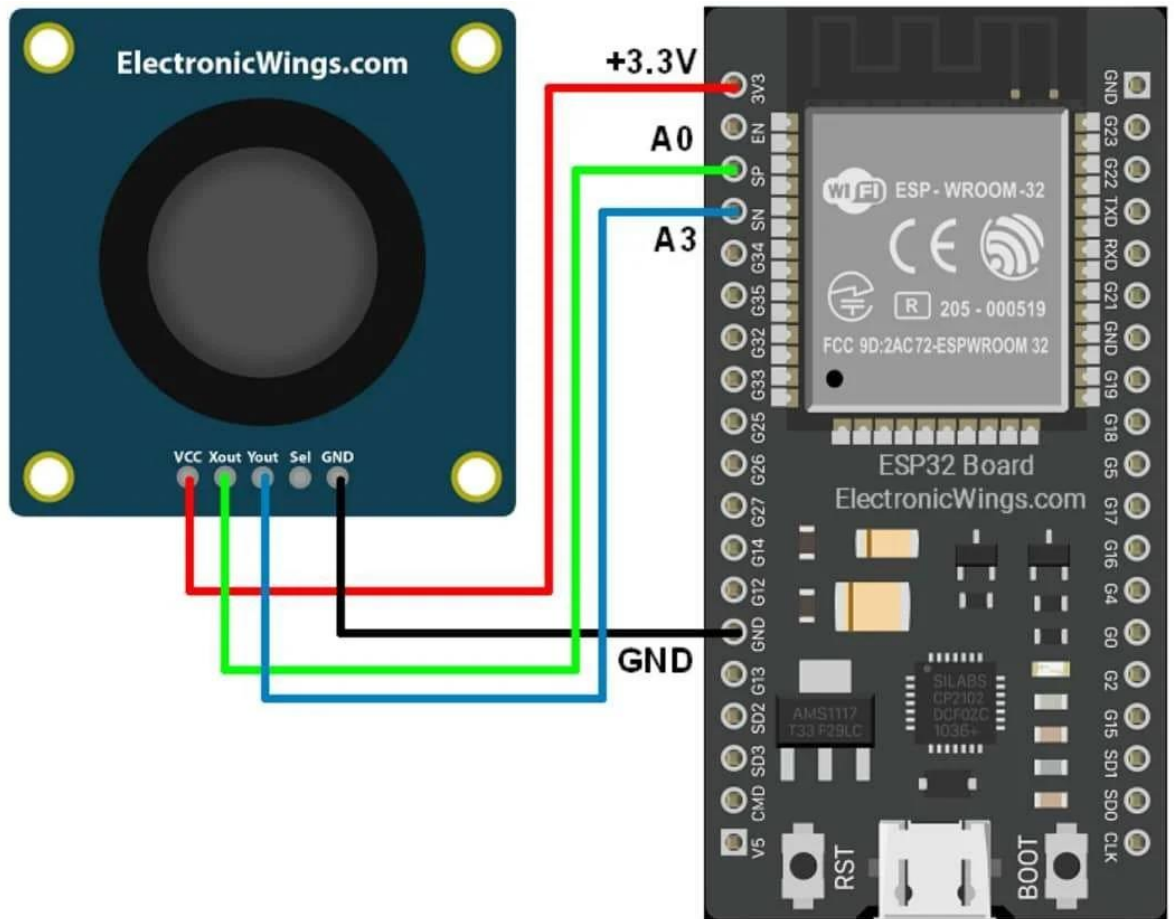


Рисунок 2.1 – Взаємодія ESP32 з аналоговим джойстиком

Під час створення портативної ретро-ігрової консолі вибору правильного акумулятора має вирішальне значення для забезпечення достатньої потужності, портативності та тривалості роботи. Батарея має забезпечити достатню напругу та ємність для живлення ESP32 (або іншого мікроконтролера), TFT РК-дисплея та будь-яких додаткових компонентів, таких як динаміки чи підсилювачі.

1. Li-Ion (літій-іонні)

- Напруга: номінально 3.7 (макс. ~4.2 В)
- Ємність: зазвичай від 1000 до 3000+ мА·рік
- Висока щільність енергії, компактність, багаторазова зарядка
- Недоліки: потребує захисту від перезаряду/перерозряду

2. Li-Po (літій-полімерні)

- Подібні до Li-Ion, але з гнучким форм-фактором

- Ідеально підходять для компактних пристроїв завдяки тонкому корпусу.
- Чутливіші до механічних пошкоджень

3. 18650 (формат Li-Ion)

- Циліндричні акумулятори високої ємності (до 3500 мА·год)
- Часто використовують у DIY-проектах через наявність готових тримачів

і модулів живлення

Безпека та експлуатація:

- завжди використовуйте захисні плати BMS або готові модулі, які запобігають коротким замиканням, глибокому розряду та перегріву;
- забезпечте правильну вентиляцію та термоізоляцію, особливо якщо пристрій використовується тривалу годину;
- для зручності заряджання можна інтегрувати індикатори заряду або навіть OLED-дисплей для відображення рівня батареї.

Поради щодо реалізації живлення:

- для заряджання використовуйте зарядні модулі (наприклад, TP4056) із вбудованим захистом і USB-портом;
- щоб збільшити напругу до стабільних 5 В (потрібних для деяких дисплеїв або модулів), використовуйте підвищувальні перетворювачі (MT3608, SY8008 тощо);
- для стабільного джерела живлення 3,3 В для ESP32 використовуйте лінійні стабілізатори з низьким падінням напруги або високоефективні перетворювачі напруги.

Приклад розрахунку автономності:

- акумулятор Li-Ion 2000 мА·год;
- середнє споживання: 400 мА;
- очікувана автономність $\approx 2000 / 400 = 5$ годин

					КВРКІ 210373.21.03.52 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

Нижче наведено найпоширеніші варіанти акумуляторів для ретро-ігрових консолей разом із їхніми характеристиками та міркуваннями.

Найкращим варіантом буде Lithium-ion (Li-Ion) батарея, Li-Ion батареї схожі на LiPo батареї, але традиційно мають циліндричну форму (наприклад, 18650 елементів). Вони широко застосовуються в портативних пристроях і павербанках.

Таблиця 2.2 – Характеристики акумулятора

Напруга	3,7 В номінально (4,2 В при повній зарядці, 3,0 В при розрядженні).
Ємність	Діапазон від 1000mAh до 3500mAh для 18650 елементів.
Форм-фактор	Циліндрична форма (наприклад, 18650, 21700).

Переваги цього акумулятора:

- висока щільність енергії;
- довший термін служби в порівнянні з акумуляторами LiPo;
- широкодоступний і економічно ефективний;

Недоліки цього акумулятора:

- більший і важчий, ніж акумулятори LiPo;
- потрібна схема захисту та правильна зарядка.

Візьмемо до уваги, що літій-іонний акумулятор ємністю 2500 мАг 18650 може забезпечити кілька годин роботи ретро-ігрової консолі.

2.2 Архітектура програмного забезпечення

Програмне забезпечення можна розділити на декілька рівнів:

1.1 Низькорівневе ПО:

- драйвери для дисплею: SPI/I2C;
- драйвери вводу: GPIO, I2C для джойстиків/кнопок;
- драйвери звуку: I2C, DAC;

- робота з пам'яттю: SPI Flash, SD – карта.

1.2 Системне ПЗ:

- RTOS: FreeRTOS, який входить в ESP-IDF;
- менеджер ресурсів: відео, звук, введення;
- файлова система: LittleFS, SPIFFS, FAT для SD-карти);
- бібліотеки для графіки: TFT_eSPI, LovyanGFX;
- бібліотеки для звуку: ESP-ADF.

1.3 Ігровий рушій:

- обробка кадрів: візуалізація спрайтів та анімацій;
- фізика: рух об'єктів та колізії;
- введення: натискання кнопок, дії гравця;
- ігрова логіка.

1.4 Ігровий контент:

- графіка: спрайти, текстури;
- звуки: ефекти, музика;
- сценарій гри: логіка, рівні.

2. Технічні аспекти

Дисплей – ESP32 може працювати з TFT LCD, OLED, або E-paper дисплеями через SPI/I2C.

2.1 Найпопулярніші бібліотеки:

- TFT_eSPI: швидкий рендеринг на SPI;
- LovyanGFX: підтримка багатьох дисплеїв;
- Adafruit GFX: універсальна, але повільніша.

2.2 ESP32 має I2S, DAC, тому можна використовувати:

- вивід на динамік через I2S/I2C DAC;
- вивід через вбудований DAC;
- бібліотеки: ESP-ADF, Audio Kit.

2.3 Ввід – Керування:

- фізичні кнопки(GPIO + pull-up резистор);

					КВРКІ 210373.21.03.52 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

- джойстик I2C;
- сенсорний екран;
- Bluetooth – контролери.

2.4 Файлова система

- SPIFFS: для зберігання текстур та налаштувань;
- SD – карта: для зберігання великих ресурсів – музика та рівні.

Низькорівневе ПО:

- драйвери для дисплею: SPI/I2C;
- драйвери вводу: GPIO, I2C для джойстиків/кнопок;
- драйвери звуку: I2C, DAC;
- робота з пам'яттю: SPI Flash, SD – карта.

2.2 Системне ПЗ:

- RTOS: FreeRTOS, який входить в ESP-IDF;
- менеджер ресурсів: відео, звук, введення;
- файлова система: LittleFS, SPIFFS, FAT для SD-карти);
- бібліотеки для графіки: TFT_eSPI, LovyanGFX;
- бібліотеки для звуку: ESP-ADF.

2.3 Ігровий рушій:

- обробка кадрів: рендерінг спрайтів та анімацій;
- фізика: рух об'єктів та колізії;
- введення: натискання кнопок, дії гравця;
- логіка гри.

2.4 Ігровий контент:

- графіка: спрайти, тексттури;
- звуки: ефекти, музика;
- сценарій гри: логіка, рівні.

Таблиця 2.3 – Стек технологій

Рівень	Компоненти
ОС/RTOS	FreeRTOS (ESP-ODF)
Графіка	TFT_eSPI, LovyanGFX
Звук	ESP-ADF, I2S, DAC
Ввід	GPIO, I2C, Bluetooth
Збереження даних	SPIFFS, SD-карта
Мереже	WiFi, ESP-NOW

2.3 Використання бібліотек для графіки та звуку (FabGL, Arduino GFX)

FabGL - це переважно графічна бібліотека для ESP32. Він реалізує кілька драйверів дисплеїв (вихід VGA, PAL/NTSC Color Composite, дисплеї I2C і SPI). FabGL також може підтримувати дані з клавіатури PS/2 і миші. FabGL також реалізує: звуковий механізм (CAP і сигма-дельта), графічний інтерфейс користувача (GUI), ігровий механізм і термінал ANSI/VT. Вихід VGA вимагає зовнішнього цифро-аналогового перетворювача (DAC): це можна зробити за допомогою трьох резисторів на 270 Ом, щоб отримати 8 кольорів, або 6 резисторів, щоб отримати 64 кольори. Композитний вихід не потребує зовнішніх компонентів (можливо, фільтр низьких частот 5 МГц). Є кілька вбудованих шрифтів, що фіксують змінену ширину. Підтримуватися необмежена кількість спрайтів. Однак великі спрайти та їх велика кількість зменшують частоту кадрів і можуть спричинити вимірювання. Коли пам'яті достатньо (на низькій роздільній здатності, як-от 320x200), можна виділити два екранні буфери, щоб реалізувати подвійну буферизацію. У цьому випадку примітиви завжди малюються на задньому буфері. За допомогою подвійної буферизації або явного виділення всі малюнки забезпечуються з вертикальним трасуванням, тому вимірювання не видно. Якщо черга примітивів для малювання не оброблена до завершення вертикального повторного трасування, то воно переривається та продовжується під час наступного повторного трасування. Існує графічний інтерфейс користувача (GUI) із вікнами, які перекриваються, і керування мишкою, а також багато віджетів

(кнопки, поля редагування, прапорці, поля зі списком, списки тощо). Нарешті, є звуковий механізм із мікшуванням кількох каналів до моновиходу. Кожен канал може генерувати синусоїдні сигнали, квадрати тощо або використовувати дані вибору. Подвійна буферизація є ключовим компонентом для забезпечення плавності графіки в системах з обмеженими ресурсами. Вона мінімізує візуальні артефакти, такі як мерехтіння або розриви кадрів, що виникають, коли зображення оновлюється без синхронізації з вертикальним оновленням дисплея (vsync). Це особливо важливо для інтерактивних додатків, таких як ігри або динамічні інтерфейси, де чіткість та стабільність виведення безпосередньо впливають на зручність користування.

FabGL - це потужна бібліотека для розробників, які працюють з мікроконтролерами ESP32 та хочуть створити інтерактивні графічні інтерфейси, ігри чи мультимедійні додатки. Її унікальність полягає в тому, що вона інтегрує в собі не лише графічні можливості, а й підтримку периферійних пристроїв, таких як клавіатури та миші, а також звуковий відображення, що робить її ідеальним інструментом для створення комплексних проектів. Безкоштовних базових функцій малювання ліній, прямокутників, ключів та виведення тексту, FabGL підтримує роботу з растровими зображеннями, що дозволяє використовувати попередньо створені графічні елементи. Бібліотека також має оптимізовані алгоритми для швидкого заповнення областей (заповнення) та виконання альфа-злиття (alpha blending), що дозволяє створювати напівпрозорі елементи інтерфейсу. Для розробників ігор особливо корисними будуть функції спрайтів. FabGL дозволяє працювати з великою кількістю спрайтів одночасно, але, як зазначалося, їх розмір і кількість впливають на продуктивність. Для мінімізації вимірювання можна використовувати подвійну буферизацію, якщо це дозволяє обсяг доступної пам'яті. Це особливо актуально для динамічних сцен, де частина оновлення екрану може призвести до візуальних артефактів.

Графічний інтерфейс користувача (GUI).

					КВРКІ 210373.21.03.52 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

Однією з найсильніших сторін FabGL є вбудована підтримка GUI. Бібліотека надає широкий набір віджетів, які можна легко інтегрувати в інтерфейс:

- кнопки(з підтримкою натискання, фокусу та стилізації);
- текстові поля (з можливістю редагування та обробки подій вводу);
- списки та випадаючі меню (для навігації по даним) ;
- прапорці та радіокнопки (для вибору параметрів) ;
- смуги прокрутки (для роботи з великими обсягами інформації) ;

Вікна у FabGL підтримують перекриття, що дозволяє створювати складні багаторівневі інтерфейси. Крім того, є можливість обробки подій миші (натискання, перетягування, прокрутка), що значно розширює можливості взаємодії з користувачем.

FabGL включає механізм генерації та мікшування звуку, що дозволяє створювати просту музику або звукові ефекти в іграх. Підтримуються різні типи хвиль:

- синусоїдальні (для плавних звуків);
- прямокутні (для 8-бітної стилістики);
- пилочасті та трикутні (для синтезу різних тембрів);
- користувацькі семпли (для відтворення записаних звуків).

Мікшування дозволяє одночасно відтворювати кілька звукових каналів, що корисно для створення багатоголосої аудіодоріжки.

Підтримка периферії

FabGL підтримує клавіатури PS/2, що дозволяє підключати стандартні комп'ютерні клавіатури до ESP32. Також реалізовано обробку подій від миші PS/2, що робить бібліотеку зручною для розробки інтерактивних додатків.

Для досягнення максимальної швидкодії бібліотека FabGL дозволяє гнучко налаштовувати параметри відображення — зокрема, частоту оновлення, роздільну здатність, глибину кольору, що дозволяє розробнику збалансувати якість зображення та продуктивність відповідно до можливостей пристрою. Крім того, бібліотека надає засоби керування чергою рендерингу, що дає змогу уникати збоїв

та «мерехтіння» під час оновлення кадру, зокрема при використанні подвійної буферизації. FabGL — це кросфункціональна бібліотека для ESP32, яка значно розширює можливості мікроконтролера в напрямках графіки, звуку, введення та побудови інтерфейсів користувача. Вона створена спеціально для проектів, що потребують виводу графіки на екрани VGA або TFT, генерації звуку, обробки вводу з клавіатури, миші чи джойстика, а також створення віконного GUI-середовища.

Arduino GFX - оптимізована бібліотека для рендеринга графіки на TFT, OLED та IPS дисплеях. Вона підтримує швидкий відображення з DMA, апаратне прискорення (на деяких дисплеях) і ефективніше працює на стандартних бібліотеках (Adafruit GFX, TFT_eSPI).

Основні можливості:

- підтримка інтерфейсів SPI/I2C/Parallel для дисплеїв;
- робота з різними драйверами: ILI9341, ST7789, SSD1306, ILI9488;
- апаратне прискорення на ESP32 (DMA, подвійна буферизація);
- графічні примітиви: лінії, коли, прямокутники, спрайти;
- робота зі шрифтами та BMP-зображеннями;
- підтримка спеціальних шрифтів: дозволяє завантажувати та використовувати власні шрифти для створення більш інтуїтивно зрозумілих і візуально привабливих інтерфейсів;
- візуалізація растрових зображень: підтримує малювання зображень BMP без необхідності складних перетворень або додаткових бібліотек;
- швидке оновлення екрана: завдяки DMA (прямий доступ до пам'яті) і подвійній буферизації мерехтіння екрана зведено до мінімуму, а анімація виглядає плавною;
- сумісність із кількома контролерами дисплеїв: дозволяє легко адаптувати той самий код для різних моделей дисплеїв без переписування логіки відтворення;
- малий слід пам'яті: особливо цінний для мікроконтролерів з обмеженими ресурсами, таких як AVR або STM32.

Переваги Arduino GFX у проєкті ретро-ігрової консолі:

- дає змогу реалізовувати швидке оновлення графіки в іграх;
- легко адаптується до різних моделей дисплеїв;
- підтримує мультимовні шрифти, що корисно для інтерфейсів;
- ідеально підходить для роботи в парі з ESP32, використовуючи його

апаратні ресурси на повну.

Таблиця 2.4 – Порівняння бібліотек

Параметр	Arduino GFX	TFT_eSPI	Adafruit GFX
Швидкодія (з DMA)	Висока	Висока	Середній
Підтримка інтерфейсів	SPI/I2C/Parallel	SPI (лише)	SPI/I2C
Сумісність із дисплеями	Широка	Обмежена	Обмежена
Подвійна буферизація	Так	Частково	Ні
Простота налаштування	Середня	Висока	Висока
Розмір бібліотеки	Малий	Середній	Великий

Бібліотека Arduino GFX забезпечує високу продуктивність і універсальність, тому є ідеальним вибором для розробки графічного рушія в проєкті ретро-консолі на ESP32.

Таблиця 2.5 – Порівняння дисплеїв

Arduino GFX	TFT_eSPI
-------------	----------

Підтримка кількох типів інтерфейсів: SPI, I2C, Parallel.	Дуже швидка робота з дисплеями на ESP32/ESP8266 (DMA, SPI частота до 80MHz).
Багатолатформність: працює з різними мікроконтролерами (ESP32, STM32, AVR тощо).	Проста інтеграція з популярними дисплеями: особливо ILI9341, ST7735, ST7789.
Більш гнучка структура: можна легко переключатися між різними драйверами дисплеїв	Підтримка графічних шрифтів, спрайтів, об'ємного тексту тощо.
Оптимізовано для ESP32: підтримка DMA, подвійної буферизації.	Тонке налаштування
Мінімальне використання пам'яті: хороша оптимізація для слабких MCU.	Менш гнучка у плані мультидисплейних проєктів.

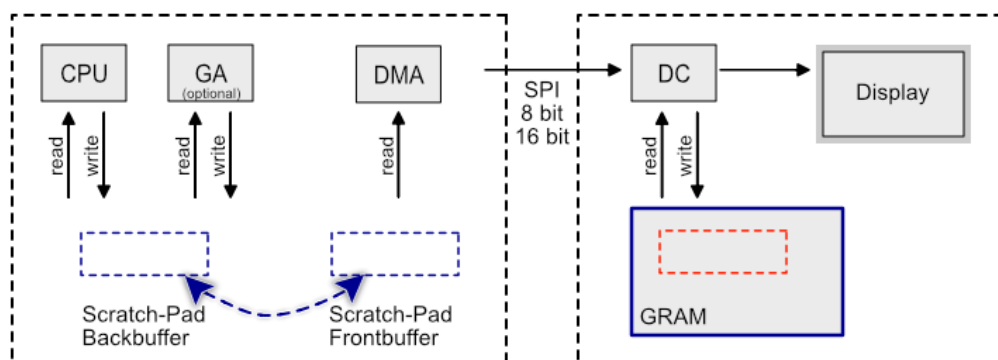


Рисунок 2.2 - Буфер блоку з подвійною буферизацією та зовнішній контролер дисплея з пам'яттю

2.4 Проектування інтерфейсу користувача

ESP32-S3-BOX разом з мікроконтролером ESP32-S3 (MCU) з дисплеєм 320x240, підключеним через SPI. Дисплей керує зовнішнім контролером дисплея, який містить власну пам'ять дисплея. У результаті весь кадровий буфер можна розташувати в центральному контролері дисплея, а в центральному

мікроконтролері (MCU) потрібен лише невеликий буферний блокнота. З цим призначенням Embedded Wizard підтримує часткове оновлення дисплея, що дає змогу оновити дисплей у наступних невеликих об'єктах. Це дає можливість працювати з буфером блокнота розміром кількох кілобайт замість повнокранного буфера кадрів у просторі пам'яті MCU. Зверніть увагу: часткове оновлення призначене для використання в системах із надзвичайно обмеженим обсягом пам'яті. Через те, що оновлення дисплея виконується під час наступних оновлень невеликих областей, переміщення графічних об'єктів може спричинити певний ефект розриву. Дизайн інтерфейсу користувача повинен отримати цей аспект. Щоб комбінувати ESP32-S3-BOX з багатьма ефективними контролерами дисплея - потрібно почати з наступних апаратних компонентів, щоб гарантувати, що запуститься програмне забезпечення.

Апаратні компоненти:

- ESP32-S3-BOX від Espressif;
- USB кабель для підключення плати до ПК;
- програмні компоненти;
- вбудована безкоштовна версія Wizard Studio або професійна версія Wizard Studio.

Встановлення інструментів і програмного забезпечення

Перед початком розробки GUI за допомогою Embedded Wizard переконайтеся, що встановлено середовище розробки програмного забезпечення ESP32 (ESP-IDF) та запущено перші програми на ESP32-S3-BOX. Середовище зборів Embedded Wizard, що надається, для ESP32-S3-BOX містить усе необхідне для створення програми Embedded Wizard UI для ESP32-S3-BOX. Після розпакування знайдете наступні підкаталоги та файли в середині:

Application\\GeneratedCode — ця папка, яка використовується для створення згенерованого коду з інтерфейсу користувача Embedded Wizard. Шаблонний проект будує програму інтерфейсу користувача з цієї папки.

Application\\Source - ця папка містить файли main.c і ewmain.c. Там ви знайдете системи ініціалізації та основний цикл для управління додатком Embedded Wizard GUI. Файл ewconfig.h містить загальні параметри конфігурації для цільової системи, такі як параметри пам'яті та параметри відображення, а також параметри конфігурації для Embedded Wizard Graphics Engine та Runtime Environment. Крім того, ця папка містить файли драйверів пристроїв C/H, які використовують для прикладу DeviceIntegration.

Examples\\ScreenSize - Ця папка містить набір демонстраційних програм, підготовлених для віддаленого розміру екрана (320x240 пікселів). Кожен приклад зберігається в окремій папці, що містить весь проект Embedded Wizard UI. Кожен проект містить необхідні параметри профілю для цільового ESP32.

HelloWorld - дуже простий проект, який можна використовувати як відправну точку і для перевірки правильності роботи всіх інструментів ланцюжка, вашої установки та плати.

ColorFormats - цей проект демонструє, що кожен додаток інтерфейсу користувача може бути створений для різних кольорних форматів: RGB565, Index8 і LumA44.

ScreenOrientation - ця демонстрація показує, що орієнтація інтерфейсу користувача не залежить від фізичної орієнтації дисплея.

DeviceIntegration - у цьому прикладі показана інтеграція пристроїв в UI-додаток і розглядаються типові питання: як запустити певну дію на цілі? як отримати дані із пристрою?

GraphicsAccelerator - ця програма демонструє графічну продуктивність мети за допомогою набору базових операцій малювання, які забезпечуються постійно та безперервно.

AnimatedList - ця демонстрація демонструє реалізацію деяких химерних віджетів списку, які прокручуються, для встановлення часу і дня тижня. Особливістю цього прикладу є ефект підвищення центрованих елементів списку і м'які ефекти збільшення з'яви/загасання.

					КВРКІ 210373.21.03.52 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

BrickGame - приклад програми BrickGame реалізує класичну гру «ракетка та м'яч». У грі у верхній частині екрана розташовано кілька рядів цегли. М'яч рухається по екрану, відскакуючи від верхньої та бічних стінок екрану. При ударі по цегли м'яч відскакує, і цегла руйнується. У гравця є рухлива ракетка, щоб підкидати м'яч догори, утримуючи його в грі.

Технічні особливості:

- Графіка реалізована через FabGL або Arduino GFX.
- Керування: GPIO-вхід із кнопок / аналоговий джойстик.
- Реакція в реальному часі за допомогою використання апаратних переривань та таймерів.
- Подвійна буферизація усуває вимірювання та забезпечує плавну анімацію.

ClimateCabinet - демо ClimateCabinet демонструє реалізацію панельного керування для випробувальної камери кліматичного впливу. Користувач може задати час нагріву, номінальну температуру та вологість, час витримки та кінцевий час охолодження.

Графічний інтерфейс:

- Вікна, поля введення, повзунки.
- Відображення значно в реальному часі.

WaveformGenerator - це демонстраційний додаток WaveformGenerator для створення форми хвилі з широкими амплітудами та частотами. Реалізація показує використання векторної графіки для малювання кривої з урахуванням списку координат.

SnakeGame — це реалізація легендарної гри «Змійка», знайомих користувачам мобільних телефонів і портативних приставок минулих років. Гравець керує змійкою, яка пересувається на екран та збирає «їжу», подовжуючи своє тіло з кожним зібраним об'єктом. Основна мета — протриматися якомога довше, уникаючи зіткнення зі стінами або власним тілом.

Основні елементи гри:

- Ігрове поле: прямокутна сітка, на якій змійка рухається по клітинках.
- Змійка: керована стрілками або джойстиком, кожен з'єднений елемент підтримує її.
- Їжа: випадково з'являється на полі, додатково довжину змійки та очки.
- Кінець гри: настає при зіткненні змійки зі стіною або самою собою.

PlatformPackage - в цій папці містяться вихідні коди та/або бібліотеки пакету платформи ESP32: кілька графічних рухів для різних кольорових форматів (RGB565, Index8 і LumA44) та середовище виконання (підкаталогі RTE).

TargetSpecific - ця папка містить усі файли конфігурації та вихідні коди, специфічні для платформи. Різні файли ew_bsp_xxx реалізують міст між програмою Embedded Wizard UI та базовим пакетом підтримуваної плати (драйверне обладнання ESP) для доступу до дисплея.

ThirdParty - ця папка містить інші вихідні коди (BSP).

Створення прикладів інтерфейсу користувача:

Приклад використання команди *"HelloWorld"*

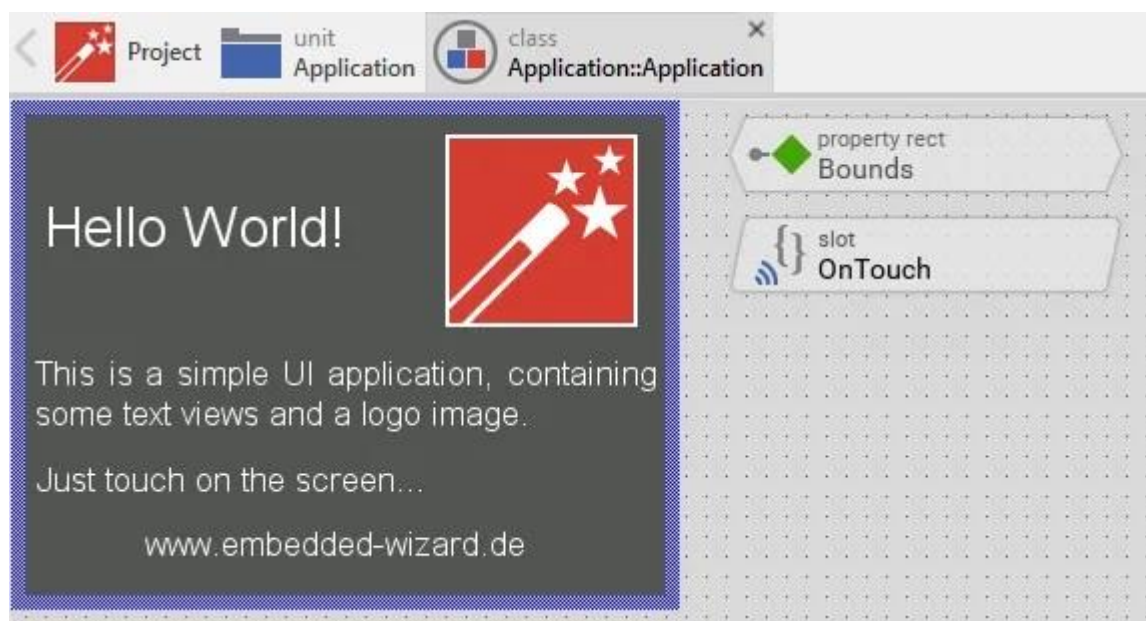


Рисунок 2.3 – Приклад *"HelloWorld"*

Для генерації вихідного коду цього прикладу програми необхідно виконати такі кроки:

- Перейдіть до каталогу `\\main\\Examples\\ScreenSize\\HelloWorld`.
- Відкривши файл проекту `HelloWorld.ewp` за допомогою попередньо встановленої `Embedded Wizard Studio`. Весь проект добре документований у рядку. Буде можливість запустити програму UI в `Prototyper`, натиснувши `Ctrl+F5`. Для того щоб запустити генератор коду, потрібно вибрати пункти меню `Build` ► `Build this profile` - або просто натисніть `F8`. `Embedded Wizard Studio` тепер генерує вихідні файли прикладу проекту в каталозі `\\main\\Application\\GeneratedCode`.

Компіляція, компоування та прошивка

Для складання та прошивки прикладу програми `Embedded Wizard UI` з використанням набору інструментів `MSYS2 MINGW32` необхідно виконати такі кроки:

Необхідно відкрити консоль (з усіма необхідними налаштуваннями для складання проекту `ESP-IDF`) та перейти на верхній рівень середовища складання `ESP32-S3-BOX`.

Якщо потрібно або змінити поточні параметри, введіть: `idf.py menuconfig`

Починається компіляція та компоування: `idf.py build`

На цьому етапі можна прошити програму за допомогою відповідного COM-порту: `idf.py -p COMxx flash`

Щоб отримати вихідні дані від програми та надати ключові вхідні дані, запустіть монітор: `idf.py -p COMxx monitor`

Більшість налаштувань проекту виконується завдяки згенерованому коду, наприклад, формату кольору або орієнтації екрана. Усі інші параметри можна зробити разом із файлом `ewconfig.h`, який містить загальні параметри конфігурації для цільової системи. Для отримання повідомлень про помилки або відображення простих повідомлень налагодження або трасування з програмою `Embedded Wizard`

					КВРКІ 210373.21.03.52 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

UI слід використати останній термінал, наприклад «Putty» або «TeraTerm», або монітор, який запускається разом з `idf.py -p COMxx monitor`.

Це термінальне з'єднання можна використовувати для всіх операторів трасування з Embedded Wizard U.

Після запуску монітора `idf.py -p COMxx monitor`` ви зможете побачити повідомлення, яке виводить прошиту програму на ESP32-S3-BOX у реальному часі. Це дуже корисно під час налагодження, оскільки дозволяє відстежувати поведінку додатка, бачити значення змінених, повідомлення про помилки, проблеми з функціями чи інші події, які ви передбачили у своєму коді за допомогою `printf()` або `EW_TRACE()` у вбудованому майстрі. Якщо під час роботи програми потрібно внести зміни до її параметрів або конфігурації, їх можна відредагувати одночасно у файлі `ewconfig.h``. Наприклад, ви можете змінити:

- роздільну здатність екрану;
- глибину кольору (дълбочина кольору);
- частоту оновлення (частота оновлення);
- пін-коди підключення до зовнішніх пристроїв;
- налаштування живлення або таймерів.

Після зміни у файлі `ewconfig.h`` слід знову вибрати проект команди `idf.py build``, прошити його команду `idf.py -p COMxx flash`` та, за потреби, переглянути результат через `idf.py -p COMxx monitor``. Якщо ви використовуєте середовище розробки на базі Visual Studio Code або інше IDE, інтеграція з ESP-IDF дозволяє запускати ці команди одночасно з інтерфейсом, що завершує цикл розробки. Також важливо не забувати перевіряти, що порт COM показано правильно, і що він не зайнятий іншим програмним забезпеченням (наприклад, відкритою сесією Putty). У разі зміни порту — не забудьте оновити команду `idf.py -p COMxx ...`` відповідно до нового значення.

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Налаштування середовища розробки та вибір мови програмування

У розробці програмного забезпечення для ігрової приставки на базі мікроконтролера ESP32 було обрано мову програмування C++, яка є стандартом для розробки вбудованих систем за рахунок високої продуктивності, можливості низького доступу до апаратних ресурсів та більшої кількості готових бібліотек. Основним середовищем розробки було обрано Arduino IDE, яке надає зручний графічний інтерфейс та має офіційну підтримку мікроконтролера ESP32 через встановлення відповідного платіжного пакету. Arduino IDE дозволяє швидко налагоджувати код, демонструвати прошивку мікроконтролера через USB та використовувати велику кількість сторінкових бібліотек, що значно пришвидшує процес розробки. Для більшого гнучкого управління проектом і підтримки великих кодових баз також використовувалося середовище PlatformIO, інтегроване з редактором Visual Studio Code. PlatformIO забезпечило:

- зручне управління проектною структурою;
- просте підключення та оновлення зовнішніх бібліотек;
- підтримку системи контролю версій (Git);
- автоматизацію процесів компіляції та завантаження прошивок;
- багатоплатформену підтримку різних мікроконтролерів;
- можливість роботи з конфігураційними файлами (`platformio.ini`) для точного налаштування параметрів проекту.

Також зазначимо, що використання Visual Studio Code з PlatformIO надало розширені можливості для налагодження, включаючи підтримку підсвічування синтаксису сліду, автодоповнення, інтеграцію з терміналом та видимим серійним монітором, що є місцем інструментом діагностики прийому та тестування вбудованої системи. Завдяки комбінації Arduino IDE і PlatformIO розробка програмного забезпечення вийшла ефективною, з можливістю швидкого

					КВРКІ 210373.21.03.52 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

тестування окремих функціональних блоків та подальшої інтеграції в загальний код проекту.

3.2 Реалізація виведення графіки на екран

Одним із ключових елементів ігрової приставки є графічний інтерфейс, що забезпечує візуальний зв'язок між користувачем і грою. Для реалізації виведення графіки було обрано OLED-дисплей з роздільною здатністю 128×64 пікселів, який побудований на базі контролера SSD1306. Цей дисплей є енергоефективним, компактним і чудово підходить для виведення як текстової, так і простої монохромної графіки.

3.2.1 Обґрунтування вибору дисплея

Серед основних причин вибору дисплея з SSD1306 можна виділити:

- невелике енергоспоживання, що дозволяє використовувати приставку з живленням від акумулятора;
- чітке відображення зображення навіть при низькому освітленні;
- наявність великої кількості готових бібліотек для мікроконтролерів;
- простий інтерфейс підключення I2C (використання лише двох дротів: SDA і SCL), що дає можливість захистити виводи мікроконтролера.

3.2.2 Використані бібліотеки

Для взаємодії з дисплеєм використовувалися дві основні бібліотеки:

- Adafruit SSD1306 - драйвер для ініціалізації дисплея, буфера керування і виведення на екран;

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

- Adafruit GFX - бібліотека для графічних операцій, яка забезпечує функціональне малювання точок, ліній, геометричних фігур, зображень та виведення тексту з вибором шрифтів;

- ці бібліотеки інтегровані в проект за допомогою менеджера бібліотек у PlatformIO. Їх використання значно спростило процес розробки графіки, дозволивши зосередитися на логіці гри.

3.2.3 Побудова системи координат

На основі розмірів дисплея (128 пікселів по горизонталі та 64 пікселів по вертикалі) була створена двовимірна система координат, у якій точка (0, 0) відповідає верхньому лівому куту екрана, а координати збільшуються праворуч (для осі X) та вниз (для осі Y). Усі ігрові об'єкти позиціонуються відповідно до цієї системи координат як зображена на (Рисунку 8), використовуючи змінні x та y, які змінюються динамічно під час виконання гри залежно від логіки подій. Для покращення графічної логіки та зі зменшенням кількості обчислень було впроваджено явну сітку (сітку), яка накладається на екран. Кожен об'єкт у грі прив'язаний до певного розміру клітинки сітки, що дозволяє легко позиціонувати та виявляти колізії між об'єктами. наприклад:

- гравець відображається як прямокутник розміром 8×8 пікселів, що дозволяє йому переміщуватись по сітці з кроком у 8 пікселів;

- Перешкоди чи проекти можуть мати розмір 4×4 пікселів, що забезпечує більшу деталізацію та дозволяє реалізувати складніші елементи ігрового процесу;

Фонові сітка не виводиться на екран, але використовується внутрішньо для спрощення логіки розміщення та зіставлення об'єктів.

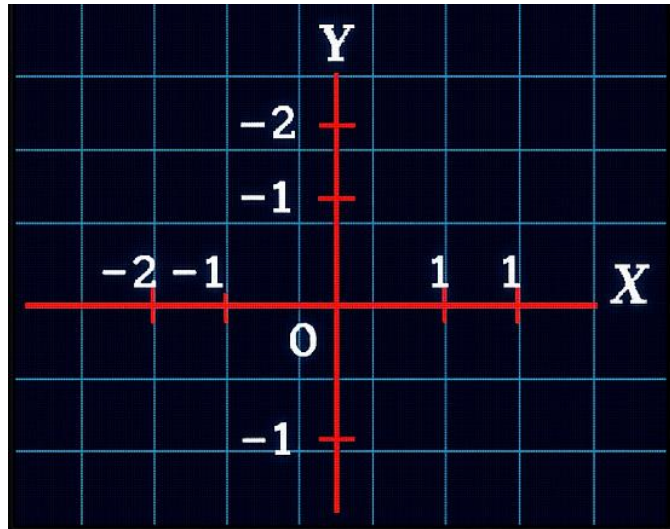


Рисунок 3.1 - будова системи координат

Такий підхід дозволяє:

- полегшити обчислення при перевірці зіткнення (колізій), після чого всі об'єкти розміщуються в кратних позиціях;
- уніфікувати розміри та розміщення спрайтів, що полегшує розробку анімації;
- економічні ресурси мікроконтролера, якщо менше координатних обчислень, використовуються в реальному часі.

3.2.4 Малювання графічних елементів

На основі бібліотеки Adafruit GFX були реалізовані функції для відображення основних графічних елементів:

- гравець: квадрат або інша фігура, що змінює координати відповідно до введення з кнопок;
- перешкоди: прямокутники або інші фігури, що рухаються на екрані;
- лінії, межі, рамки - для візуального поділу ігрового поля.

Кожна з цих функцій реалізована таким чином, щоб вона оперувала координатами та могла бути легко викликана з основного циклу гри.

3.2.5 Буферизований вивід

Оскільки дисплей оновлюється досить повільно, було реалізовано буферизовану систему оновлення екрана. Спочатку всі графічні елементи малюються у framebuffer (внутрішній буфер бібліотеки), і лише після завершення малювання викликається метод `display.display()`, який виводить зображення на екран. Такий підхід дозволяє:

- уникнути мерехтіння при оновленні екрана;
 - зменшити кількість звернень до дисплея;
- підвищити швидкодію і плавність анімацій.

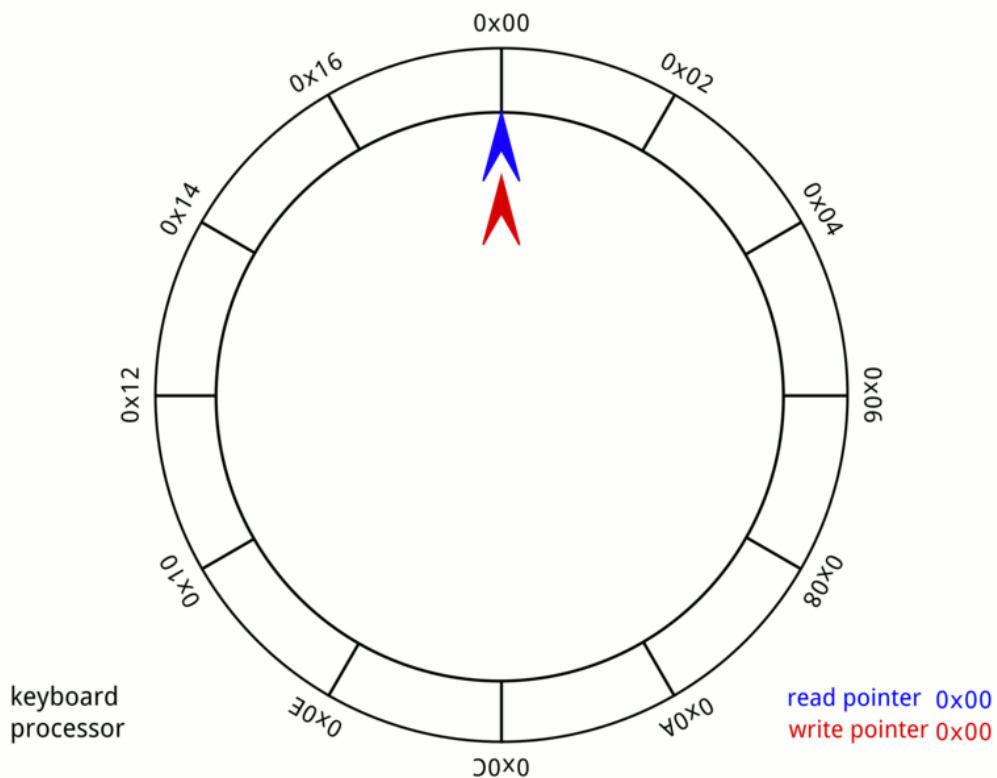


Рисунок 3.2 - Буферизований вивід

3.2.6 Часткове оновлення екрана

З метою підвищення продуктивності та оптимізації використання ресурсів ESP32 було реалізовано механізм часткового оновлення дисплея. Наприклад, якщо змінюється лише позиція гравця, оновлюється тільки область, в якій він

знаходиться. Це дозволяє не очищати та не перемальовувати весь екран повністю, зберігаючи ресурси мікроконтролера.

– На практиці оновлення виробляються тільки для тих ділянок екрану, де відбулися зміни. наприклад:

– Якщо змінюється лише позиція гравця, перерисовується лише стара і нова область його розміщення;

– Якщо з'являється новий об'єкт або зникає попередній, оновлюється тільки відповідна частина екрана;

– Для анімованих елементів, таких як ефекти вибуху або переміщення ворогів, оновлюється лише область навколо них.

Такий підхід має кілька переваг:

– зменшення кількості переданих даних до дисплея через SPI-інтерфейс, що значно збільшує відображення кадру;

– скорочення обробки кадру часу, завдяки чому зростає загальна частота оновлення екрана (FPS);

– економія енергії, що особливо актуально для портативних пристроїв;

– менше миготіння зображення, відсутність фону та інші незмінні елементи залишаються стабільними, а не перемальовуються кожному кадру.

3.2.7 Тестування графічного виводу

Для перевірки коректності роботи графічного інтерфейсу було створено окремі функції тестування, які перевіряли:

– вивід тексту на різних позиціях дисплея;

– роботу з різними розмірами шрифтів;

– малювання геометричних фігур;

– відображення анімованих об'єктів.

Усі тести були проведені на реальному апаратному забезпеченні, що дало змогу перевірити час оновлення дисплея, стабільність роботи та зручність виводу.

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

3.3 Обробка введення від геймпада/кнопок

Для забезпечення взаємодії користувача з ігровою приставкою було реалізовано систему обробки введення за допомогою механічних кнопок, підключених до цифрових входів мікроконтролера ESP32.

3.3.1 Апаратна реалізація

Кожна кнопка з'єднується з цифровим входом ESP32 через резистор підтягування (pull-up). У стані спокою контакт кнопки розімкнутий, і на вхід подається високий рівень логіки. Під час натискання — контакт замикається на землю, і мікроконтролер зчитує низький рівень (LOW), що трактує як активне натискання.

3.3.2 Програмна обробка введення

Щоб уникнути хибних спрацювань, викликаних механічним відскоком контактів (явищем «дребезгу»), була реалізована програмна фільтрація натискань (debounce). Це досягалося шляхом перевірки стану кнопки кілька разів із затримкою в кілька мілісекунд (зазвичай 10–50 мс), і фіксації зміни лише після стабілізації сигналу.

Для кожної кнопки був реалізований стан натиснуто/відпущено, а також підтримка утримання кнопки (long press), що дозволяє гравцю неодноразово змінювати положення об'єкта при тривалому натисканні.

3.3.3 Призначення кнопок

У рамках проєкту було передбачено кілька функціональних кнопок:

- ліво/право/вгору/вниз - рух ігрового персонажа по екрану;
- start/select - запуск або перезапуск гри, активація меню;

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

- pause - зупинка гри;
- action - взаємодія з об'єктами, постріли тощо.

Таким чином, реалізована система зображена на (Рисунку 10) дозволяє вводити точно й швидко керувати ігровим процесом, забезпечуючи інтуїтивно зрозуміле управління для користувача.

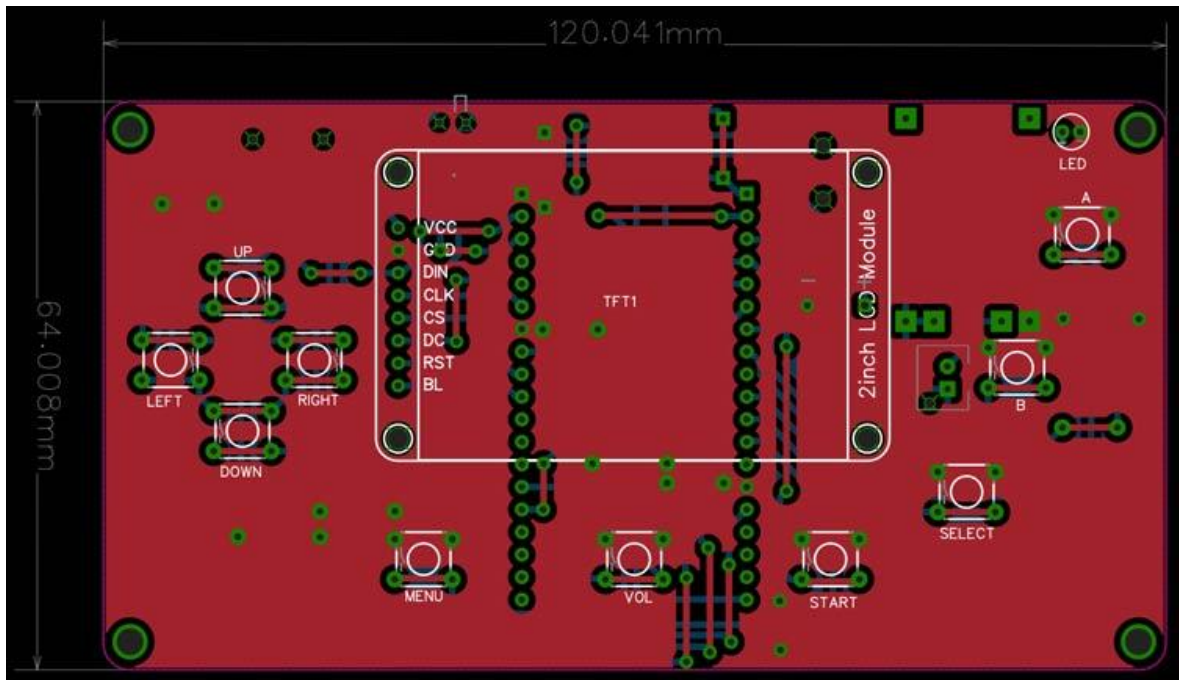


Рисунок 3.3 - Призначення кнопок

3.4 Генерація звуку та робота з динаміком

Щоб зробити ігрову приставку більш інтерактивною та динамічною, було реалізовано систему генерації звуку за допомогою п'єзоелектричного динаміка, підключеного до одного з цифрових виходів ESP32.

Для створення звукових ефектів використовується ШІМ-сигнал (широкоімпульсна модуляція), який генерується апаратними засобами ESP32. Такий підхід дозволяє формувати звуки різної частоти без суттєвого навантаження на центральні обчислювальні ядра.

3.4.1 Технічна реалізація

П'єзоелемент був з'єднаний з мікроконтролером через резистор, що обмежує струм, і не потребує підсилювача, оскільки він може працювати з цифровими сигналами прямо з виводу ESP32.

3.4.2 Генерація звуку

Для відтворення звукових ефектів використовувалася вбудована функція `tone(pin, frequency, duration)`, яка генерує на заданому піні прямокутний сигнал із вказаною частотою та тривалістю. Також застосовувалась функція `noTone(pin)`, щоб зупинити звук, коли це необхідно.

3.4.3 Звукові ефекти

У грі були реалізовані прості звукові ефекти, що відповідають певним подіям:

- старт гри - короткий сигнал з підвищенням частоти;
- рух або дія гравця - тихий короткий клік;
- постріл - імпульс середньої частоти;
- зіткнення або поразка - низькочастотний сигнал;
- перемога - послідовність нот або міні-мелодія.

Завдяки цьому гра стала більш емоційною, а користувач отримує додатковий зворотній зв'язок від пристрою.

3.5 Оптимізація продуктивності та використання пам'яті

Мікроконтролер ESP32, попри доволі потужний процесор і наявність кількох сотень кілобайт оперативної пам'яті, все одно має обмеження щодо ресурсів, особливо при виведенні графіки та одночасній обробці логіки гри, введення і звуку. У зв'язку з цим була проведена цілеспрямована оптимізація. Оптимізація охоплювала кілька ключових напрямків:

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

- зменшення обсягу використовуваних графічних ресурсів: були використані спрайти з невеликою роздільною здатністю, а також обмежена кількість кольорів для збереження обсягу пам'яті відеобуфера;
- ефективне використання кешу та пам'яті SPI RAM (у разі її наявності): неосновні дані (наприклад, фонові елементи) зберігалися у зовнішній пам'яті, тоді як використані об'єкти - у внутрішній SRAM;
- використання апаратного прискорення: було задіяно можливості апаратного PWM для звукового виводу та DMA (Direct Memory Access) для виконання операцій з передачі графіки на дисплей;
- оптимізація логіки гри: цикли та операції було переписано з урахуванням обмежень на тактову частоту та кількість доступної пам'яті. Було зменшено кількість перевірок стану в кожному кадрі та реалізовано буферизацію введення;
- розподіл навантаження між ядрами: одне ядро ESP32 відповідало для обробки ігрової логіки, потім як інше - для оновлення графіки або звуку, що дозволяло уникнути «просидання» кадрів і затримок у керуванні.

3.5.1 Оптимізація пам'яті

- Глобальні буфери: замість динамічного виділення пам'яті в циклах були створені глобальні масиви для зберігання даних гри (наприклад, позицій об'єктів).
- Мінімізація графіки: використовувалися лише монохромні спрайти невеликого розміру.
- Програмна економія RAM: фонові зображення або повторювані графічні елементи зберігалися в PROGMEM (Flash пам'яті), а не в оперативній пам'яті
- Використання зовнішньої SPI RAM (за наявності): деякі версії ESP32 мають підтримку зовнішньої оперативної пам'яті, яка використовувалася для зберігання великих структур, зокрема попередніх оброблених тайлів, таблиць колізій або звукових семплів;

					КВРКІ 210373.21.03.52 ПЗ	Арк. 51
Зм.	Арк.	№ докум.	Підпис	Дата		

- Уникнення дублювання даних: змінені, що мають однакову інформацію в кількох модулях, були перенесені в єдиний модуль і передавались через покажчики або посилання, що зменшило загальний обсяг споживаної пам'яті;
- Оптимізація структури даних: були замінені важкі типи (наприклад, int, float) на легші (uint8_t, bool) там, де це було можливо без втрати функціональності. Структури вирівнювалися вручну для уникнення зайвого заповнення байтами.

Memory Type	Size (in KB)	Details
ROM	448	<ul style="list-style-type: none"> • First stage bootloader • ROM libraries
IRAM	192	<ul style="list-style-type: none"> • Instruction RAM • 32KB cache goes for each core • Only 32-bit aligned access
DRAM	328	<ul style="list-style-type: none"> • Data RAM • DMA capable • Fastest possible access
RTC Fast Memory	8	<ul style="list-style-type: none"> • Only PRO CPU accessible • Operates at APB clock (typical value 80MHz)
RTC Slow Memory	8	<ul style="list-style-type: none"> • Both CPU accessible • Operates at RTC fast clock (typical value 8MHz)

Рисунок 3.4 - Типи пам'яті.

3.5.2 Оптимізація продуктивності

- Часткове оновлення дисплея: оновлювалися лише змінені частини екрана, що дозволяло уникнути зайвого навантаження на процесор та інтерфейс I2C.
- Спрощення обчислень: замість використання "важких" математичних функцій (наприклад, float, sin(), sqrt()) перевагу надавали простим арифметичним операціям та цілочисельній математиці.
- Оптимізований цикл гри: основний цикл (loop) був побудований з урахуванням пріоритетів - перевірка введення, оновлення логіки, оновлення графіки й лише потім - звук.

Завдяки цьому приставка демонструє стабільну роботу навіть у випадках одночасного відтворення графіки, обробки кнопок і звукових ефектів.

3.7 Тестування та аналіз результатів

3.7.1 Методика тестування працездатності системи

Для перевірки працездатності створеної ігрової приставки було застосовано функціональну методику тестування, яка передбачала поетапну перевірку кожного з окремих модулів:

- OLED-дисплей: перевірялась якість виведення графіки та тексту, швидкість оновлення, відсутність артефактів під час руху об'єктів;
- кнопки керування: тестувались реакція на натискання, правильність обробки команд, робота debounce-механізму;
- звуковий модуль: перевірялась точність відтворення тонів та синхронізація зі сценаріями гри;

Після модульного тестування було проведено інтеграційне тестування, під час якого перевірялась коректна взаємодія всіх компонентів у реальному ігровому процесі. Особливу увагу приділено реакції системи на часті дії гравця, включаючи швидкі переміщення, багаторазові натискання кнопок, запуск звукових ефектів тощо.

Окремо проводилось тестування стабільності роботи пристрою протягом тривалого часу. Пристрій залишався увімкненим понад 6 годин із повторним запуском гри, і за цей час не спостерігалось збоїв, зависань чи некоректної поведінки.

3.7.2 Аналіз продуктивності та обмежень ESP32

В процесі тестування було підтверджено, що мікроконтролер ESP32 забезпечує достатню обчислювальну потужність для роботи 2D-ігор з базовою графікою та звуковим супроводом.

					КВРКІ 210373.21.03.52 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		

Позитивні характеристики ESP32:

- частота оновлення екрана до 30 кадрів/с без помітних затримок;
- плавна обробка введення з кнопок та миттєве реагування на дії гравця;
- можливість одночасної роботи з графікою, кнопками та звуком.

Виявлені обмеження:

- відсутність апаратного прискорення графіки, що ускладнює використання складних візуальних ефектів.
- обмежений обсяг оперативної пам'яті (до 520 КБ), що впливає на можливість зберігати велику кількість спрайтів або фонів.
- труднощі з роботою при високій деталізації графіки або складних анімаціях без зниження частоти кадрів.

Загалом, обчислювальні можливості ESP32 є цілком достатніми для реалізації класичних аркадних ігор у стилі 8-біт або 16-біт.

3.7.3 Порівняння отриманих результатів з аналогічними рішеннями

Для кращого розуміння ефективності реалізованої системи було проведено порівняння з аналогічними платформами. Це наведено в таблиці 4.1.

У порівнянні з Arduino Uno/Nano, мікроконтролер ESP32 істотно перевершує їх за параметрами обчислювальної потужності, частоти тактування, об'єму оперативної пам'яті, а також завдяки вбудованим інтерфейсам Wi-Fi та Bluetooth, що дозволяє реалізувати більш функціональні та інтерактивні об'єкти без потреби в зовнішніх модулях. Крім того, ESP32 підтримує DMA, апаратне прискорення для синхронізації та аудіо, що критично важливо для плавної анімації та мультимедійних компонентів у грі. У порівнянні з комерційними ігровими платформами (наприклад, Nintendo Switch, PlayStation Portable, Game Boy Advance), створена система має базовий рівень графіки та звуку, обмежену продуктивність мікроконтролера та низьку роздільну здатність дисплея. Цей протест не компенсується:

					КВРКІ 210373.21.03.52 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

- низькою сім'єю компонентів - проект доступний для реалізації навіть студентами або ентузіастами;
- простотою в розробці - платформа Arduino й сумісні бібліотеки можуть швидко створювати прототипи;
- відкритичністю коду - користувачі можуть вільно змінювати, розширювати та адаптувати функціонал під свої потреби;
- енергоефективністю - що дозволяє створювати портативні рішення з живленням від акумулятора;
- навчальну цінність - цей проект чудово підходить для вивчення основ мікроконтролерного програмування, графіків, електроніки та системи реального часу.

Таблиця 3.1 - Порівняння з аналогічними платформами

Параметр	Arduino Uno/Nano	ESP32 (наш проєкт)	Комерційні приставки
Частота процесора	16 МГц	до 240 МГц	>1 ГГц
РАМ	2 КБ / 8 КБ	до 520 КБ	>1 ГБ
Графіка	Обмежена (без екрана)	OLED/можливо TFT	Повноколірна
Звук	Простий, без DMA	Простий, частотно-кодований	Багатоканальний
Підключення	Потрібен модуль	Вбудований Wi-Fi / BT	Wi-Fi / Bluetooth
Ціна системи	Низька	Низька/середня	Висока
Модифікація	Обмежена	Відкрите ПЗ/залізо	Обмежена

3.8 Висновки щодо можливості подальшого розвитку проєкту

На основі проведеного тестування та аналізу можна зробити висновок, що проєкт має значний потенціал для подальшого вдосконалення та розвитку як з технічної, так і з функціональної точки зору. Реалізована ігрова приставка демонструє на практиці можливості мікроконтролера ESP32 у сфері графіки, звуку та взаємодії з користувачем, і вже на поточному етапі забезпечує базову ігрову функціональність.

Можливі напрями модернізації:

– Багатокористувацький режим — реалізація спільної гри між двома (або більше) пристроями з використанням Bluetooth або Wi-Fi, що відкриває можливості для кооперативного або змагального геймплею.

– Власний геймпад — розробка окремого контролера з бездротовим підключенням, зокрема через Bluetooth або інфрачервоний канал, дозволяє підвищити зручність використання та наблизити пристрій до реальних консолей.

– Поліпшення графіки — використання TFT-дисплеїв із більшою роздільною здатністю (наприклад, 480x320 або вище) та кращим відображенням збільшує значно покращити видимість складової ігор.

– Емуляція класичних ігор — можлива інтеграція емуляторів ретро-консолей, таких як NES, GameBoy або Atari, що дозволяє запускати класичні ігри без переписування.

– Інтуїтивне меню — створення графічного меню для вибору гри, налаштування, перегляду статистики, прогресу збереження, що зробить користувацький досвід повноцінним та приємним.

– Поліпшення звуку — впровадження стереовиходу, мікшування кількох аудіоканалів, а також підтримка форматів аудіофайлів (наприклад, WAV) для музичного супроводу та ефектів.

Таким чином, реалізована ігрова приставка є не лише завершеним функціональним проєктом, а й гнучкою платформою для досліджень,

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

експериментів та навчання. Вона дозволяє розвивати теми мікроелектроніки, програмування вбудованих систем, графічних інтерфейсів, передачі даних, цифрової обробки сигналів та багато іншого, що робить її перспективною як для індивідуальних розробників, так і для освітнього середовища.

					КВРКІ 210373.21.03.52 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

ВИСНОВКИ

У кваліфікаційній роботі було досліджено, спроектовано та реалізовано прототип ігрової приставки на основі мікроконтролера ESP32, що має на меті створення компактного, енергоефективного та функціонального пристрою для запуску простих 2D-ретроігор.

У процесі виконання роботи:

- проведено аналіз технічних можливостей мікроконтролера ESP32, включаючи його архітектуру, периферійні інтерфейси, режими енергоспоживання та графічні можливості;

- здійснено порівняння ESP32 з альтернативними мікроконтролерами (ESP8266 та іншими), що дозволило обґрунтувати вибір саме цієї платформи для реалізації проекту;

- обрано та інтегровано оптимальні апаратні компоненти: TFT-дисплей, джойстик, акумулятор, елементи живлення;

- розроблено архітектуру програмного забезпечення, що включає графічні драйвери, обробку введення, звуковий модуль та ігровий рушій;

- реалізовано графічний інтерфейс користувача за допомогою бібліотек FabGL та Arduino GFX, забезпечено обробку введення з геймпада та генерацію звуку;

- проведено тестування працездатності системи, визначено основні обмеження, пов'язані з продуктивністю та обсягами доступної пам'яті;

- запропоновано напрями удосконалення, зокрема: використання більш продуктивних дисплеїв, впровадження Bluetooth-контролерів, розширення грового контенту, оптимізацію обробки графіки.

Отримані результати свідчать про доцільність використання ESP32 для створення бюджетних ігрових пристроїв із мінімальним енергоспоживанням.

					КВРКІ 210373.21.03.52 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Офіційний технічний опис мікроконтролера ESP32 від Espressif.
URL:https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf - ESP32 Series /(дата звернення: 12.06.2025)
2. Посібник для початківців із роботою з ESP8266, з прикладами підключення до Wi-Fi, використанням GPIO, базовим програмуванням.
URL:<https://www.esploradores.com/wp-content/uploads/2018/04/A-Beginners-Guide-to-the-ESP8266-1.pdf> - ESP ESP8266 /(дата звернення: 12.06.2025)
3. Розгляд ESP32 як платформи для обробки даних, з акцентом на його мультимедійні можливості: аудіо, графіка, обробка введення (Вбудовані можливості графіки, звуку та обробки введення).
URL:https://www.researchgate.net/publication/334572715_Using_the_ESP32_Microcontroller_for_Data_Processing (дата звернення: 12.06.2025)
4. Покроковий гайд по налаштуванню I²S інтерфейсу на ESP32 для виведення звуку, з прикладами коду та схемами.
URL:<https://dronebotworkshop.com/esp32-i2s/> (дата звернення: 12.06.2025)
5. Приклад підключення аналогового джойстика до ESP32, читання аналогових значень з осей X та Y (Інтерфейс аналогового джойстика з ESP32).
URL:<https://www.electronicwings.com/esp32/analog-joystick-interfacing-with-esp32> - (дата звернення: 12.06.2025)
6. Стаття від розробника Espressif щодо побудови UI-додатків для ESP32-S3 з використанням бібліотек типу LVGL (Проектування інтерфейсу користувача).
URL:<https://developer.espressif.com/blog/how-to-create-an-ui-application-for-the-esp32-s3/> (дата звернення: 12.06.2025)
7. Версія даташиту на ESP32 для швидкого перегляду, відформатована як класичний даташит. URL:https://www.alldatasheet.com/datasheet-pdf/view/1242995/ESPRESSIF/ESP32_V01.html. /(Дата звернення: 12.06.2025)

					КВРКІ 210373.21.03.52 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

8. Код і приклад реалізації кільцевого буфера на ESP32 в середовищі Arduino (Буферизований вивід). URL: <https://github.com/CDFER/Ring-Buffer-Demo-ESP32-Arduino?tab=readme-ov-file> (дата звернення: 12.06.2025)

9. Практичний приклад створення ретро-ігрової консолі з ESP32 (Призначення кнопок) URL: <https://circuitdigest.com/microcontroller-projects/esp32-based-retro-game-console> (дата звернення: 12.06.2025)

10. Smith J. Game Development on Microcontrollers. *Journal of Embedded Systems*. 2022. Vol. 15, No. 2. P. 88–97.

11. Докладний інструктаж зі створення ретро-комп'ютера з VGA, звуком та введенням через PS/2, використовуючи бібліотеку FabGL URL: <https://www.instructables.com/A-VGA-Computer-Based-on-ESP32-and-FabGL-With-Sound/> -/ (дата звернення: 12.06.2025)

12. Кодування 2D ігрового движка для ESP32 URL: https://www.theseus.fi/bitstream/handle/10024/863283/Gianadda_David_Jean_Raymond.pdf? / (дата звернення: 12.06.2025)

13. Навчальний матеріал з підключенням аналогової джойстики до ESP32 URL: <https://esp32io.com/tutorials/esp32-joystick?> (дата звернення: 12.06.2025)

14. Розробка вбудованих ігор за допомогою Arduino GFX. Hackster.io, 2024. Приклад використання Arduino GFX для створення гри. (дата звернення: 12.06.2025)

15. PWM-based Audio Synthesis on ESP32. MakerAdvisor, 2023. Стаття про створення звуку PWM на ESP32. (дата звернення: 12.06.2025)

16. TFT_eSPI vs Arduino_GFX Performance Benchmarks. esp32.com forum, 2024. Оцінка продуктивності порівняльно між бібліотеками. (дата звернення: 12.06.2025)

17. Retro Style GUI with FabGL. GitHub examples, 2023. Приклади ретро GUI з FabGL. (дата звернення: 12.06.2025)

					КВРКІ 210373.21.03.52 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

18. FreeRTOS Task Scheduling on ESP32 for Games. Espressif blog, 2023. Стаття нараощена можливості FreeRTOS для ігрових задач. (дата звернення: 12.06.2025)

19. Графіка в реальному часі на ESP32. Hackaday, 2022. Опис графіка реального часу на ESP32. (дата звернення: 12.06.2025)

20. OLED Graphics with SSD1306 and Arduino_GFX. Adafruit guide, 2023. Інструкція по графіці для OLED. (дата звернення: 12.06.2025)

21. Double Buffering Techniques in Embedded Graphics. EmbeddedArtistry, 2022. Теорія подвійної буферизації. (дата звернення: 12.06.2025)

22. PSRAM Memory Optimization for ESP32. Espressif technical notes, 2022. Рекомендації з оптимізації PSRAM. (дата звернення: 12.06.2025)

23. ADC Sampling and Filtering Analog Inputs. Analog Devices app note, 2021. Теорія АЦП, корисна для джойстиків. (12.06.2025)

24. BLE Gamepad Protocol Implementation. Bluetooth SIG, 2024. Стандарт HID для Bluetooth-геймпадів. (дата звернення: 12.06.2025)

25. ESP32 Battery Management and Power Saving. BatteryUniversity.com, 2023. Рекомендації з енергозбереження. (дата звернення: 12.06.2025)

26. Designing Portable Consoles with Li-Po. Instructables, 2024. Інструкція зі схемами живлення. (дата звернення: 12.06.2025)

27. Latency Optimization in Embedded Audio. Texas Instruments whitepaper, 2023. Практики зниження затримок у звуці. (дата звернення: 12.06.2025)

28. Arm-based Alternatives to ESP32 Performance Analysis. Maker.space, 2022. Порівняльне дослідження для конкурентів. (дата звернення: 12.06.2025)

29. ILI9341 Drive Timing and Optimization Guide. 2021. Технічні поради для ІЛІ9341-дисплеїв. (дата звернення: 12.06.2025)

30. Efficient BMP Rendering on Microcontrollers. Microcontroller.com blog, 2022. Оптимізація рендерингу BMP. (дата звернення: 12.06.2025)

31. MicroPython on ESP32: Graphics and Interrupts. 2024. Документація по MicroPython для графіки та периривань. (дата звернення: 12.06.2025)

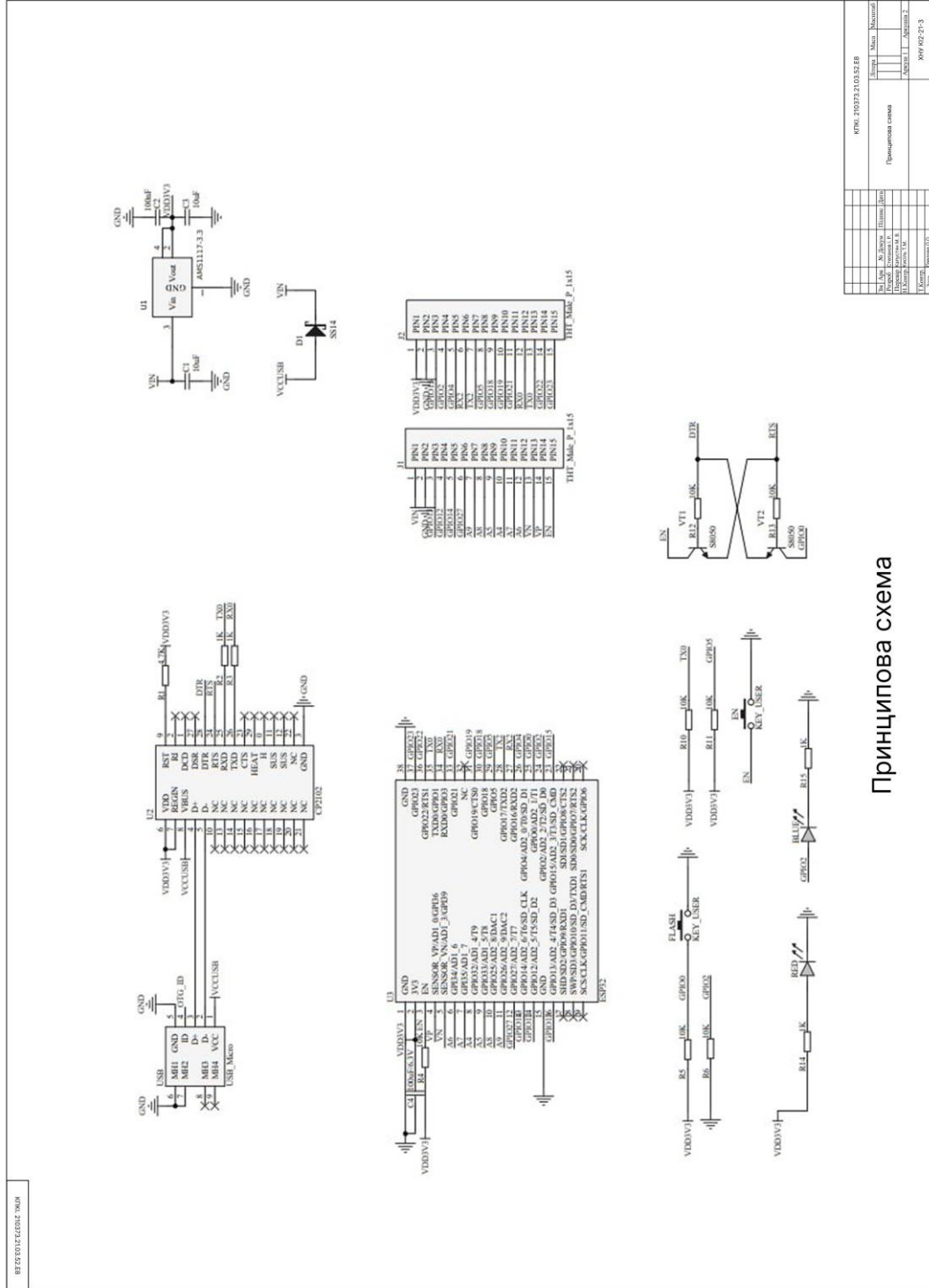
					КВРКІ 210373.21.03.52 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

32. Wi-Fi Mesh Multiplayer on ESP32. Espressif example, 2023. Приклад Mesh-проекту. (дата звернення: 12.06.2025)
33. Using NeoPixel LED for HUD on Game Console. Adafruit, 2022. Приклади підсвітки ігрового інтерфейсу. (дата звернення: 12.06.2025)
34. Infrared Remote Protocols for Microcontrollers. Sony developer guide, 2021. Підключення ІЧ-пульта. (дата звернення: 12.06.2025)
35. HDMI-like VGA Output on ESP32. FabGL advanced projects, 2024. Розширена робота з VGA. (дата звернення: 12.06.2025)
36. Modular Game Architecture Patterns in C++. GameDev.net, 2023. Патерни архітектури ігрових двигунів. (дата звернення: 12.06.2025)
37. Using FATFS on SD-Card Game Loading. ChaN, 2019. Інтеграція SD та файлової системи. (дата звернення: 12.06.2025)
38. Retrogaming on ESP32-S3: Audio Synchronization. 2025. Практичний проект зі звуком у ретроіграх на S3. (дата звернення: 12.06.2025)
39. Bluetooth LE HID Profile for Game Controllers. Bluetooth SIG whitepaper, 2023. Стандарт для бездротових контролерів. (дата звернення: 12.06.2025)
40. Unicode/Font Handling on Embedded Devices. UTF-8 info whitepaper, 2022. Підтримка кирилиці/Unicode. (дата звернення: 12.06.2025)
41. ESP32 Watchdog & Fail-safe Routines. Espressif app note, 2023. Практики стабільності системи. (дата звернення: 12.06.2025)
42. Big Buffer Memory Allocation Strategies. Embedded-tech.com, 2024. Алгоритми розподілу пам'яті. (дата звернення: 12.06.2025)

					КВРКІ 210373.21.03.52 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «ПРИНЦИПОВА СХЕМА»



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

Іван Степанов

ІІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-3

ЗАЯВА

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

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

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

16.06 2025 року

ІВ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Степанов Іван Романович

Тема: Розробка ігрової приставки на базі мікроконтролера ESP32

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 71

1. Короткий зміст роботи та прийнятих рішень: Метою даної дипломної роботи є розробка портативної ігрової приставки на базі мікроконтролера ESP32 з підтримкою ретро-ігор, що поєднує обробку введення, виведення графіки та генерацію звуку при обмежених апаратних ресурсах.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: першому розділі кваліфікаційної роботи здійснено аналітичний огляд сучасних мікроконтролерів, зосереджено увагу на апаратних та програмних можливостях ESP32, розглянуто особливості його архітектури, підтримку бездротових інтерфейсів, графіки, звуку та обробки введення. Також виконано огляд сучасних підходів до створення вбудованих систем та ретроігрових пристроїв, проаналізовано популярні бібліотеки (FabGL, Arduino GFX) та їх придатність до розробки графічних інтерфейсів і аудіообробки. У другому розділі детально описано процес проектування апаратної частини ігрової приставки: обрано TFT-дисплей, аналоговий джойстик, кнопки керування, динамік та акумулятор. Розроблено схему підключення усіх компонентів до мікроконтролера ESP32. Підібрано оптимальні компоненти з урахуванням енергоспоживання, габаритів та функціональних потреб. У третьому розділі описано реалізацію програмної частини, зокрема налаштування графіки з використанням бібліотеки FabGL, обробку введення з джойстика та кнопок, а також генерацію звуку через I2S Реалізовано

демонстраційні ігри, зокрема BrickGame, WaveformGenerator і ClimateCabinet.
Проведено тестування системи та оцінку її продуктивності.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Обмеженість графічних та звукових можливостей через апаратні обмеження ESP32.

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

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

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Гегула
Микола Васильович К.Т.Н., доцент кафедри АКІТ та Р

"16" 06 2025 р.

Гегула (підпис)

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

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

Назва: Розробка Ігрової приставки на базі мікроконтролера EPS32

Автор: Іван СТЕПАНОВ

Спеціальність: 123- Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Марія КАПУСТЯН, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 4,27% і адресується до 9 першоджерела; та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

Марія КАПУСТЯН

Андрій Ніченорук

Ольга ПАВЛОВА

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 12%

ID: 246859 Title: БКР Розробка ігрової приставки на базі мікроконтролера ESP32 Added in a DB: 2025-06-19 Authors: Іван СТЕПАНОВ Heads: Марія КАПУСТЯН Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	77056	696	562 (1%)	8 (1%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності експертом

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

Автор: Іван СТЕПАНОВ

Співавтор:

Назва: Стеанов_Розробка ігрової приставки на базі мікроконтролера ESP32

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:4.3%

Коефіцієнт подібності 2:1%

Мікропробіли: 9

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-19 08:44:36.0

Після аналізу Звіту подібності констатую наступне:

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

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

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

Обґрунтування:

2025-06-19

Дата



Доцент Андрій Нічепорук

експерт