

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
д-р техн. наук, проф.

_____ І. М. Журавська

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Бездротовий контролер для аркадних відеоігор

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.01 – 405.22010515

Студент

_____ Б. В. Павленко

підпис

«__» _____ 2024 р.

Керівник ст. викладач

_____ І. С. Бурлаченко

підпис

«__» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

« _____ » _____ 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної бакалаврської роботи

Видано студенту групи 405 факультету комп'ютерних наук

_____ Павленко Богдан Вікторович _____

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

_____ Бездротовий контролер для аркадних відеоігор _____

Затверджена наказом по ЧНУ ім. Петра Могили від 30.01.2024 № 17.

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

_____ Розроблений прототип бездротового контролеру для аркадних відеоігор
на основі мікроконтролера, з реалізацією підключення до смартфона. Для
реалізації задачі було зібрано усі необхідні
компоненти.

4. Перелік питань, що підлягають розробці

_____ Аналіз інформації стосовно технологій бездротового з'єднання і роботи
контролерів, їх переваги та недоліки. Порівняльний аналіз існуючих рішень.
Підбір необхідних для реалізації компонентів. Проектування прототипу
контролера на базі мікроконтролера ESP32. Реалізація програмної частини
контролера

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Бездротовий контролер для аркадних відеоігор

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КР	11.01.24	12.01.24	Виконав
2	Огляд літератури за темою роботи	15.01.24	22.02.24	Виконав
3	Складання календарного плану БКР	23.02.24	25.02.24	Виконав
4	Аналіз предметної області	26.02.24	19.03.24	Виконав
5	Розробка проєктних рішень	20.03.24	01.04.24	Виконав
6	Моделювання та конструювання АПЗ	02.04.24	09.04.24	Виконав
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	10.04.24	14.04.24	Виконав
8	Відгук керівника КР	15.04.24	21.05.24	Виконав
9	Оформлення БКР та презентації	15.04.24	21.05.24	Виконав
10	Попередній захист №1	28.05.24	28.05.24	Виконав
11	Попередній захист №2	06.06.24	06.06.24	Виконав
13	Рецензування	06.06.24	15.06.24	Виконав
14	Завершення оформлення КР та презентації	06.06.24	15.06.24	Виконав
15	Захист бакалаврської кваліфікаційної роботи	.06.24	.06.24	Виконав

Розробив здобувач ВО Павленко Богдан Вікторович _____
(прізвище, ім'я, по батькові) (підпис)
«_____» _____ 2024 р.

Керівник роботи ст. викладач Бурлаченко Іван Сергійович _____
(посада, прізвище, ім'я, по батькові) (підпис)
«_____» _____ 2024 р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«Бездротовий контролер для аркадних відеоігор»
Студент 405 гр.: Павленко Богдан Вікторович
Керівник: ст. викладач Бурлаченко Іван Сергійович

Ця робота присвячена створенню і проектуванню периферійного пристрою, який являє собою бездротовий контролер для аркадних відеоігор, який складається з мікроконтролера як основного компоненту пристрою, і декількох функціональних елементів. Головною метою проекту є створення пристрою, котрий має забезпечити просте і комфортне занурення в ігровий процес.

При виконанні проекту головне напрямлення було закладено на використанні бездротових технологій передачі даних для забезпечення зручності при використанні пристрою.

Розроблена система включає в себе мікроконтролер ESP32 з підтримкою вбудованого Bluetooth модуля, для забезпечення бездротової складової контролера, тактильні кнопки для імітації натискання по екрану смартфона і стік, котрий імітує свайпи по екрану смартфона. Приймач сигналу у вигляді девайсу на базі ОС Android отримує від контролера уже опрацьовані дані і використовує ці дані для виконання певних дій на девайсі. Для виконання програмної частини було використано середовище розробки Arduino IDE, котре було оснащено певним набором бібліотек, які використовувалися під час процесу програмування апаратної частини пристрою.

Результатом роботи став розроблений прототип периферійного пристрою, який являє собою бездротовий контролер для аркадних відеоігор, котрий своїми ергономічними рішеннями покращує процес ігрового занурення, забезпечуючи бездротове з'єднання і необхідні функції для зручного і комфортного користування пристроєм.

Пояснювальна записка бакалаврської роботи складається зі вступу, трьох розділів основної фахової частини, висновків і трьох додатків. У вступі визначається актуальність теми, її мета, об'єкт та предмет дослідження, та закладено основні завдання на бакалаврську роботу. У першому розділі проведено аналіз системи і готових рішень, підібрано компоненти і описано вимоги до АПЗ. У другому розділі описано покроковий процес підключення всіх компонентів в єдину систему. В третьому розділі описано процес створення програмного забезпечення, проведено тестування готової системи та вказано на можливі покращення. У висновках наведено аналіз виконаної роботи та отримані результати дослідження. У додатку А наведено перевірку на унікальність роботи. У додатку Б наведено лістинг коду програмної частини. У додатку В наведено усі використані в роботі графічні матеріали.

Загалом, бакалаврська робота без додатків складає 61 с., містить 50 рис., 5 табл., і 19 джерел посилань.

Ключові слова: *Мікроконтролер, ESP32, Android, Arduino, Bluetooth, периферійний пристрій, ігрове занурення.*

ABSTRACT

of the Bachelor's Thesis

"Wireless controller for arcade video games"

Student: Bohdan Pavlenko

Supervisor: Senior lecturer Ivan Burlachenko

This bachelor's thesis is devoted to the creation and design of a peripheral device, which is a wireless controller for arcade video games, consisting of a microcontroller as the main component of the device, and several functional elements. The main goal of the project is to create a device that should provide a simple and comfortable immersion in the gameplay.

The project was focused on the use of wireless data transmission technologies to ensure the convenience of using the device.

The developed system includes an ESP32 microcontroller with support for a built-in Bluetooth module to provide the wireless component of the controller, tactile buttons to simulate tapping on the smartphone screen, and a stick that simulates swiping on the smartphone screen. The signal receiver in the form of an Android device receives already processed data from the controller and uses this data to perform certain actions on the device. The Arduino IDE development environment was used to execute the software part, which was equipped with a certain set of libraries that were used during the programming process of the device's hardware

The result of the work was a prototype of a peripheral device, which is a wireless controller for arcade video games, which with its ergonomic solutions improves the process of gaming immersion, providing a wireless connection and the necessary functions for convenient and comfortable use of the device.

The explanatory note of a bachelor's thesis consists of an introduction, three chapters of the main professional part, conclusions and three appendices. The introduction defines the relevance of the topic, its purpose, object and subject of research, and lays down the basis for the bachelor's thesis assignment. The first chapter analyses the system and off-the-shelf solutions, selects components and describes the requirements for an automated control system. The second section describes the step-by-step process of connecting all components into a single system. The third section describes the software development process, tests the finished system and points out possible improvements. The conclusions provide an analysis of the work performed and the results of the study. Appendix A contains a check for the uniqueness of the work. Appendix B contains the code listing of the software part. Appendix C contains all the graphic materials used in the work.

In total, the bachelor's thesis without appendices is 61 pages, contains 50 figures, 5 tables, and 19 references.

Keywords: *Microcontroller, ESP32, Android, Arduino, Bluetooth, peripheral device, game immersion.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ІСНУЮЧИХ КОНТРОЛЕРІВ ДЛЯ АРКАДНИХ ВІДЕОІГОР ..	6
1.1 Загальна інформація про контролери	6
1.2 Приклади ігрових контролерів.....	7
1.3 Інформація про аркадні відеоігри	14
1.4 Опис комплектуючих	15
1.5 Вимоги до АПЗ	19
Висновки до першого розділу	21
2 ПРОЄКТУВАННЯ КОНТРОЛЕРА ДЛЯ АРКАДНИХ ВІДЕОІГОР	23
2.1 Будова контролера	23
2.2 Передавання даних між девайсом і смартфоном	25
2.3 Опис роботи функціональних компонентів	31
2.4 Підключення елементів	33
2.5 Опис системи живлення	37
2.6 Алгоритм роботи.....	39
Висновки до другого розділу.....	40
3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АРКАДНИХ ВІДЕОІГОР.....	41
3.1 Підготування середовища програмування	41
3.2 Програмування контролеру.....	50
3.3 Перевірка роботи контролера.....	53
3.4 Методи покращення пристрою	57
Висновки до третього розділу	58
ВИСНОВКИ	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	60
ДОДАТОК А Довідка про унікальність пояснювальної записки.....	63
ДОДАТОК Б Повний код розробленого програмного забезпечення	64
ДОДАТОК В Графічні матеріали КБР.....	66

ПЕРЕЛІК СКОРОЧЕНЬ

АПЗ – апаратно-програмне забезпечення

ОС – операційна система

Пад – геймпад

ПК – персональний комп'ютер

5V – живлення в 5 вольт

BLE – Bluetooth Low Energy

FHSS – Frequency-Hopping Spread Spectrum

FTDI – Future Technology Devices International

GND – заземлення

IDE – Середовище розробки

IoT – Інтернет речей

USB – Universal Serial Bus

HBA – Host Bus Adapter

ВСТУП

Темою кваліфікаційної роботи було обрано «Бездротовий контролер для аркадних відеоігор». В наш час велика кількість людей у вільних для себе час, щоб впоратися з нудьгою або просто для розваги використовує відеоігри різних жанрів, найпопулярнішим з яких є казуальні і аркадні, які за своїм визначенням можуть бути єдиним жанром.

Так само велика кількість людей відходить від стандартних засобів гри, таких як клавіатура і мишка до більш спрощених і конкретно напрямлених на ігри засобів, таких як контролери. Адже клавіатура з мишею для свого використання займають більше місця ніж контролер, в той же час коли кількість діючих кнопок в іграх може бути розтягнута по всій довжині клавіатури, в контролері всі діючі кнопки завжди знаходяться в положенні в якому до них легко дотягнутися тримаючи контролер, їх менше і простіше запам'ятати розташування, тим самим використовуючи контролер можна швидше звикнути до керування гри і отримувати повне задоволення від процесу швидше аніж на клавіатурі.

Мета: удосконалити контролери певних типів для взаємодії з ігровими платформами за рахунок розроблення АПЗ для бездротової передачі даних до Android ОС.

Об'єкт дослідження: електронна гра на аркадному ігровому автоматі.

Предмет дослідження: бездротові контролери для аркадних відеоігор на базі Android ОС.

Завдання:

- ознайомитися з різними видами контролерів, детально опрацювавши їхні характеристики, процесом створення таких контролерів і організацією їх роботи з різними платформами;
- визначити вимоги до технічної і програмної частини для розробки бездротового пристрою для аркадних відеоігор;

- змодельовати і спроектувати пристрій використовуючи функціональні моделі проектування;
- розробити програмне забезпечення для роботи з пристроєм, обґрунтувати вибір використаних в процесі засобів і технологій, описати отриманий результат.

Практичне значення: практичним значенням в результаті розробки має стати покращення процесу занурення в ігрову галузь для людей не знайомих з іграми, так само і для геймерів, котрі хочуть отримувати повне задоволення від процесу гри. Розроблений в процесі контролер можна буде брати з собою в різні ситуації, де буде доцільним грати в аркадні відеоігри. Окрім того цей контролер можна використовувати для проведення турнірів та змагань в аркадах.

1 АНАЛІЗ ІСНУЮЧИХ КОНТРОЛЕРІВ ДЛЯ АРКАДНИХ ВІДЕОІГОР

1.1 Загальна інформація про контролери

Контролер – спеціалізований компонент системи, котрий використовується для керування зовнішніми пристроями: принтерами, дисплеями, накопичувачами, відеосистемами, тощо.

За весь час існування контролерів вони пройшли великий процес вдосконалення. Спочатку контролери використовувалися як перетворювачі одного інтерфейсу в інший, таких як зовнішній і внутрішній інтерфейси. Наступним кроком в еволюції контролерів стало вдосконалення функцій, таких як пришвидшення роботи, додавання сумісності до різних систем і забезпечення безпеки під час роботи з контролером. Останнім на даний час кроком в вдосконаленні контролерів стало те, що більшість контролерів котрі раніше відповідали лише за одну певну дію або процес зараз можуть стати частиною єдиного контролера, котрий виконує усі функції вбудованих у нього контролерів і є універсальним.

Серед контролерів можна виділити певні «сімейства», котрі розвиваються окремо один від одного, проте мають спільне призначення як контролери:

- мережеві плати;
- адаптери (електричні, інтерфейсні, НВА);
- мікроконтролери;
- контролери доменів;
- логічні контролери;
- контролери бездротових мереж;
- ігрові контролери.

Окремо розберемо ігрові контролери і що вони з себе представляють. Ігровий контролер – пристрій введення, який використовується у відеоіграх, розважальних системах, який передає вхідні данні в ці системи для керування

процесом. Найперші ігрові контролери з'явилися лише для людей, котрі володіли консолями і така тенденція продовжувалася до моменту поки їх не обладнали USB інтерфейсом, що надало можливості під'єднувати контролери як до консолей, так і до ПК, що надало більшості людей можливості спробувати отримати нові враження від ігрового процесу. На даний момент ігрові контролери розробляються з бездротовим підключенням, що надає ще більшої гнучкості у їх використанні і облегшує процес підключення до різних платформ.

Такі пристрої введення як: клавіатура, миша, геймпад, джойстик є ігровими контролерами, так само як і пристрої спеціального призначення, такі як: кермо або світловий пістолет. Різниця між цими контролерами лише в тому, що перші можна використовувати для загального призначення, в той час як другі використовуються лише в певних типах ігор для покращення якості і заглиблення в гру, і можуть бути легко замінені першими для менш досконалого заглиблення.

1.2 Приклади ігрових контролерів

Далі детальніше розберемо різні варіації ігрових контролерів.

Геймпад (рис. 1.1) – пристрій котрий тримається обома руками з участю усіх пальців крім великих, бо вони використовуються для керування стіками і натискання кнопок пада. Стандартний сучасний геймпад обладнаний:

- парою аналогових стіків, котрі являють собою спрощений джойстик і які використовуються для орієнтування в просторі методами керування персонажем (лівий) і камерою (правий), і додатково обладнанні кнопкою, котра спрацьовує при натисканні на сам стік, тим самим розширюючи функціонал;

- чотирма кнопками дії, котрі відповідають за виконання певний дій в грі;

- D-pad, кнопка у вигляді хрестовини що може натискатися у чотири або вісім напрямів, тим самим додаючи додаткових опцій у використанні, загалом для пересування інтерфейсом гри;
- двома службовими кнопками, котрі знаходяться в середині пада і виконують функції паузи, виклику опцій або зміну режиму роботи;
- спусковими гачками (бампер і тригер) котрі знаходяться з верхньої частини контролера і з обох сторін. Зазвичай обидві кнопки натискаюся вказівним пальцем і знаходяться в місці де до них легко дотягнутися через конструкцію геймпаду.



Рисунок 1.1 – Сучасний геймпад

Весло (рис. 1.2) – контролер котрий має лише колесо для керування та одну або кілька кнопок. Використовується для пересування вздовж однієї осі екрану. Були першими з усіх ігрових контролерів, але швидко втратили свою популярність і актуальність з розвитком ігрової індустрії.



Рисунок 1.2 – Контролер типу весло

Джойстик (рис. 1.3) – пристрій котрий складається з одного або двох ручних джойстиків, котрий нахиляється вздовж двох осей і може обертатися навколо третьої. Може мати кнопки керування як на самому джойстику так і на платформі на якій він встановлений.



Рисунок 1.3 – Приклад джойстику з різними функціональними кнопками

Трекбол (рис. 1.4) та тачпад (рис. 1.5) – це сфера та панель відповідно котрі мають єдине призначення – керування курсором миші. Вбудовані в ноутбуки за замовчуванням і виконують функції комп'ютерної миші, займаючи менше місця. Трекбол може бути швидше миші в залежності від конструкції самого контролера, проте в більшості сучасних ноутбуків замінений на тачпад. Деякі сучасні трекболи є окремими контролерами (рис. 1.6), котрі замінюють комп'ютерну мишу.



Рисунок 1.4 – Вбудований у ноутбук трекбол

Кермо (рис. 1.7) – теж саме що і звичайне кермо котре використовується в машинах для їзди, тільки адаптоване для роботи з комп'ютером і іграми. В середині рульового колеса знаходяться такі самі функціональні кнопки як і на геймпаді, D-pad і системні кнопки. Окрім того на основі керма може знаходитися шифтер для імітування ручного гальма автомобіля, і за самим кермом можуть знаходитися «пелюстки» як в сучасних гоночних автомобілях, котрі використовуються для переключення передач.



Рисунок 1.5 – Вбудований у ноутбук тачпад



Рисунок 1.6 – Варіація трекболу як окремого контролера

Різні види керма мають різний кут оберту від 200 до 1080 градусів, це означає що кермо може крутитися в кожную сторону на 100 і 540 градусів відповідно, це використовується для різних ігрових симуляторів які копіюють кути оберту реальних гоночних автомобілів. Кожне кермо обладнане системою зворотнього зв'язку, щоб реалістичніше імітувати водіння автомобіля, прирівнюючи його до реального. Однією з версій керма є ярмо (рис. 1.8), яке імітує керування літаком, окрім обертального руху має вісь вперед-назад для імітації функцій тангажу літака.



Рисунок 1.7 – Контролер типу кермо



Рисунок 1.8 – Контролер типу ярмо з додатковою панеллю шифтерів

Невід'ємною частиною керування реальним автомобілем є педалі, тому в деяких комплектах до керма додають педалі, в деяких їх треба докуповувати окремо, але такий контролер теж присутній і нічим не відрізняється від реальних педалей. Буває в двох варіаціях, для ручного і механічного керування, єдиною різницею є відсутність педалі зчеплення в другому (рис. 1.9).



Рисунок 1.9 – Контролер типу педалі у варіації ручного (зліва)
і механічного (справа) керування

Миша (рис. 1.10) та комп'ютерна клавіатура (рис. 1.11) є типовими пристроями введення для персонального комп'ютера і в даний час є основними ігровими контролерами для комп'ютерних ігор . Миша часто використовується разом із килимком для досягнення більшої швидкості, комфорту, точності та більш плавного руху для гравця. Деякі ігрові консолі також мають можливість працювати з клавіатурою та мишею. Клавіатура комп'ютера створена за зразком клавіатури друкарської машинки і призначена для введення письмового тексту. Миша – це портативний вказівний пристрій, який використовується на додаток до клавіатури.

У іграх клавіатура зазвичай керує рухом персонажа, тоді як миша використовується для керування ігровою камерою або використовується для прицілювання. Хоча спочатку було розроблено для загального введення комп'ютера, є кілька доступних периферійних пристроїв для клавіатури та миші, які розроблені спеціально для ігор, часто з вбудованими функціями, що стосуються ігор. Приклади включають периферійні пристрої від Razer, серію клавіатур «Zboard» і серію «G» Logitech .



Рисунок 1.10 – Комп'ютерна миша



Рисунок 1.11 – Комп'ютерна клавіатура

Контролери віртуальної реальності (рис. 1.12) з'явилися відносно недавно, проте швидко набули популярності, адже з їх допомогою можна швидко зануритися в ігровий світ і отримати повне задоволення від часу проведеного в віртуальній реальності. Являють собою «палку» яку зручно тримати в руці, котра оснащена додатковими кнопками зверху і на сторонах контролера, дозволяючи виконувати додаткові функції. Основною функцією контролера є відстеження рухів руками гравця і проектування цих рухів в гру.



Рисунок 1.12 – Контролери віртуальної реальності

Існує ще велика кількість різних контролерів які менш популярні, ніж усі зазначені вище, проте все ще користуються популярністю певної аудиторії. Такі контролери дозволяють заглибитися гравцям в ігровий процес і отримувати від нього повні враження.

1.3 Інформація про аркадні відеоігри

Аркадні відеоігри – це різновид відеоігор, які на початку свого впровадження пропонувалися в громадських місцях, таких як торгові центри, бари та ресторани. Більшість ігор, дуже популярних у 1970-х і 1980-х роках, грали на спеціальних автоматах, які називалися монетоприймачами (рис. 1.13), оснащених джойстиком, кнопками та іншими елементами керування. Зазвичай вони характеризуються простим і легким ігровим процесом, доступністю та інтенсивністю коротких сеансів гри, які є дуже інтенсивними. Більшість аркадних ігор мали дуже яскраву графіку та захопливий звук, що привертало увагу гравців і створювало атмосферу змагання та розваги. Ігрова механіка в аркадних відеоіграх зазвичай була дуже зрозумілою і легкою для вивчення, але важкою для освоєння, що спонукало гравців спробувати ще раз і витратити додаткові гроші.



Рисунок 1.13 – Приклад аркадних автоматів

Однією з головних особливостей аркадних відеоігор була можливість зберігати рекорди гравців, що створювало елемент змагання і соціальної взаємодії. Гравці часто змагалися за встановлення нових рекордів і занесення своїх імен до таблиці лідерів, тому з'являлася додаткова мотивація до гри.

1.4 Опис комплектуючих

1.4.1 Вибір мікроконтролеру

Галузь мікроконтролерів зустрічає великим вибором для різних сфер застосування і використання, але найбільше за все з цього вибору нас цікавлять контролери на базі Arduino і ESP.

Arduino – це електронна платформа з відкритим вихідним кодом, що базується на простому у використанні апаратному та програмному забезпеченні. Плати Arduino здатні зчитувати вхідні дані і перетворювати такі дані на вихідні. Для цього плати Arduino програмуються вільно користувачем, щоб вони підходили для певних задач і використань. Для програмування таких контролерів використовується однойменна мова програмування, яка базується на мові програмування C++, та однойменне програмне забезпечення Arduino IDE.

Китайська компанія Espressif створила ESP як доступну платформу для легкого створення розумних пристроїв, підключених до Wi-Fi, Bluetooth, LoraWan тощо. Крихітні мікроконтролери поєднують дивовижну функціональність у міцному та недорогому корпусі, ставши ще більш популярними після того, як хакери почали використовувати дружнє для початківців середовище розробки Arduino IDE на ESP.

Просте прототипування продуктів безпеки Wi-Fi або BLE стало можливим завдяки створенню плат для розробки з мікроконтролером ESP, встановленим на друкованій платі з USB-інтерфейсом для легкого програмування. До них відносяться NodeMCU і D1 mini, мабуть, найпопулярніші плати для розробки ESP. Ці плати дозволяють початківцям

легко розпочати роботу з такими популярними мовами, як Arduino, MicroPython та Lua.

Протягом багатьох років ESP мала гарну репутацію. Це досить недорогий модуль для розробників, зацікавлених в одноразових проєктах. У багатьох системах IoT значний відсоток вартості матеріалів йде безпосередньо на підключення та сам мікроконтролер. Модулі ESP є досить недорогими в порівнянні з багатьма іншими рішеннями, які доступні на ринку сьогодні.

В результаті порівняння існуючих мікроконтролерів обох компаній, основні характеристики яких представлені в табл. 1.1, для виконання поставленої задачі було обрано мікроконтролер ESP32 (рис. 1.14). Мікроконтролер ESP32 є вдосконаленою версією ESP8266, надаючи більше можливостей та функцій для IoT проєктів.

Таблиця 1.1 – Порівняння проаналізованих мікроконтролерів

Характеристи ка	Arduino UNO	Arduino Nano	Arduino Pro Mini	ESP8266	ESP32
Робоча напруга	5В	5В	5В	3.3В	3.3В
Кількість входів і виходів	14	14	14	14	34
Флеш пам'ять	32Кб	32Кб	16–32Кб	до 16Мб	До 4Мб
Оперативна пам'ять	2Кб	2Кб	1Кб	32Кб + 80Кб	520Кб + 80Кб
Тактова частота процесора	16МГц	16МГц	8– 16МГц	80МГц	80–240МГц
Споживання струму	50мА– 200мА	40мА– 150мА	35мА– 110мА	70мА– 215мА(15мк А в режимі сну)	70мА– 300мА(10мк А в режимі сну)
Габарити	69 x 53	43 x 18	33 x 18	58 x 31	51 x 27
Джерело живлення	USB або зовнішній блок живлення	USB або зовнішній блок живлення	USB- UART конверте р	USB або зовнішній блок живлення	USB або зовнішній блок живлення

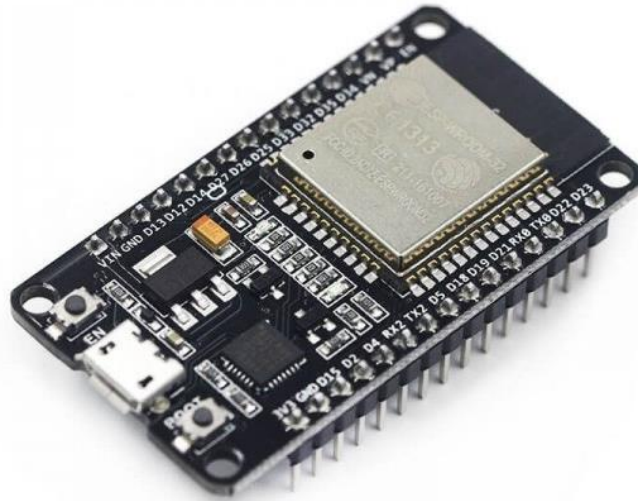


Рисунок 1.14 – Мікроконтролер ESP32

Таблиця 1.2 – Характеристика мікроконтролера ESP32

Характеристика	Опис
Мікроконтролер	Xtensa LX6 32-bit dual-core
Флеш-пам'ять	до 4 Мб
Оперативна пам'ять	520 Кб + 80 Кб
Входи/Виходи	34 цифрових входів/виходів (з яких 13 ШІМ-виходів)
Робоча/Максимальна напруга	3.3/3.6 В
Частота тактового генератора	до 240 МГц
Інтерфейси	UART, SPI, I ² C
Розміри	Довжина 51 мм, Ширина 27 мм
Особливості	Bluetooth v4.2 BR/EDR і BLE, підтримка SD-карт, SPI флеш

Так само як і ESP8266, ESP32 підтримує Wi-Fi стандартів 802.11 b/g/n, але основним фактором котрим ці дві плати відрізняються є наявність вбудованого в ESP32 Bluetooth модуля, котрий підтримує використання Bluetooth двох стандартів – Bluetooth 4.2 і Bluetooth Low Energy(BLE). Обидва мікроконтролера мають декілька режимів роботи, серед яких є режими споживання електроенергії в діапазоні від 10 мкА в режимі сну, до 300 мА в режимі передачі даних. Характеристики мікроконтролера наведено в табл. 1.2.

1.4.2 Вибір елементів керування

Оскільки контролер як ігровий девайс має вміщувати в себе декілька функціональних кнопок, котрі будуть відповідати за певні конкретні дії в грі, при взаємодії з ними на контролері, і ще одного елемента керування, котрий буде імітувати свайпання в чотири напрямки, як це відбувається при взаємодії користувача з смартфоном.

Для проєктування бездротового контролера було обрано тактильні кнопки (рис. 1.15), адже вони невеликих розмірів і чудово вміщуються в модель невеликого контролера. Додатком до цього стане чіткий звук «клацання» при натисканні на кнопку, що надаватиме користувачу зворотній зв'язок при натисканні на кнопку.



Рисунок 1.15 – Обрана тактильна кнопка

Для функції імітації свайпів на екрану смартфона було обрано пристрій типу стік (рис. 1.16), котрим можна керувати одним пальцем, при цьому зберігаючи повний функціонал і імітацію свайпів по екрану, котрої ми і намагаємося досягти.



Рисунок 1.16 – Обраний імітатор свайпів по екрану смартфона

Окрім того такий компонент обладнаний додатковою кнопкою, котру можна легко натискати під час взаємодії і поворотів компоненту, і котра може бути додатково прив'язана до виконання певної функції за потреби.

1.5 Вимоги до АПЗ

Ігрові контролери знаходять своє місце в усіх бюджетних нішах, варіюючись від невеликої ціни в декілька доларів до гігантів індустрії ціною в декілька сотень доларів. Таким самим чином ігрові контролери захоплюють усі жанрові ніші для різноманітних ігор, будучи то будь-які симулятори, по типу гоночних чи для літаків, котрі імітують і передають враження від керування реальними літальними апаратами чи машинами в грі, до найпростіших аркадних ігор, в котрих основним керуванням мають бути рухи в усі сторони компаса та натискання пари клавіш чи кнопок.

Контролер для аркадних відеоігор має бути невеликого розміру і ваги, щоб його зручно було брати з собою будь-де разом зі смартфоном, мати функціонал, котрий відповідає потребам користувача під час ігрового процесу, та буде не складний в освоєнні для користувачів через свою навантаженість різноманітними сенсорами і кнопками.

Для розробки такого контролера необхідно обрати низку деталей. Основним і найнеобхіднішим стане вибір мікроконтролера для основи проєкту. Наступним етапом буде вибір основних функціональних компонентів – різноманітних кнопок котрі будуть реагувати на натискання або інші дії при взаємодії з контролером та відповідати на ці дії певними керуваннями на екрані смартфона. Важливо також подумати про батарею приладу, адже бездротовий контролер не може бути бездротовим якщо він постійно знаходиться під живленням з кабелю, тому другою по важливості задачею буде вибір акумулятора котрий підтримуватиме контролер в зарядці на період декількох ігрових сесій, і при цьому буде вписуватися в конструкцію невеликого за габаритами і вагою контролера.

Проаналізувавши інформацію вище були зроблені вимоги для апаратно-програмного забезпечення у вигляді бездротового контролера для аркадних відеоігор, описаними в табл. 1.3, для правильного виконання своїх функцій і забезпечення ефективної роботи для користувача.

Таблиця 1.3 – Вимоги для АПЗ

ESP32	Мікроконтролер для виконання функції передавання даних між пристроєм і смартфоном
Тактильні кнопки	Забезпечення імітації натискання на екран смартфона, додавання окремих функцій при натисканні
Стік	Забезпечення імітації свайпів на екрані смартфона
Елемент живлення	Забезпечення безперервної роботи мікроконтролера та інших компонентів пристрою

Мікроконтролер ESP32 на ядрі Xtensa LX6 32-bit dual-core буде використовуватися для опрацювання і передавання даних між бездротовим контролером і смартфоном, з чим йому допоможуть 34 цифрових входи/виходи. Підтримка кількох режимів енергоспоживання і невеликі розміри дозволять інтегрувати цей мікроконтролер в пристрій без зайвих проблем.

Для імітації натискання на екран смартфона пристрій буде обладнаний тактильними кнопками невеликих розмірів, котрі будуть під'єднані до мікроконтролера. Невеликі розміри допоможуть зручно інтегрувати кнопки в пристрій, а функція тактильності допоможе підтримувати зворотній зв'язок, видаючи чіткий звук при натисканні на кнопку.

Для імітації свайпів по екрану смартфона пристрій буде обладнаний стіком, котрий зручно використовувати одним пальцем, і який ідеально підходить для поставленої для нього задачі. Найбільший за розмірами компонент проєкту, проте достатньо невеликих розмірів щоб підтримувати тенденцію невеликого, ергономічного пристрою.

Для забезпечення безперервної роботи основних компонентів пристрою потрібне джерело постійного живлення. На етапі проєктування пристрій можна жити через USB кабель, але під час тестування готового пристрою буде використовуватися джерело живлення типу акумулятор, з номінальною напругою в 3.7 В.

Висновки до першого розділу

Проаналізовано галузі використання контролерів як допоміжних засобів для роботи певних пристроїв, розглянуто процес становлення контролерів як окремої одиниці та о тенденції розвитку контролерів в сучасному світі.

Досліджено готові рішення для ігрових контролерів різноманітних типів, проаналізовано їх недоліки і плюси використання та конкретні напрямки використання певних ігрових контролерів.

Наслідком проведеного аналізу стало визначення основних функціональних компонентів для ігрових контролерів. Розроблено критерії вибору компонентів для виконання певних функцій і сумісності цих компонентів в різних контролерах. Було проведено ознайомлення з жанром аркадних ігор, процесом їх становлення і переходом до відеоігор.

Основним моментом розділу був вибір комплектуючих для проєктування бездротового контролеру. Певна увага була приділена вибору

мікроконтролера серед існуючих варіантів, в результаті аналізу котрих був обраний мікроконтролер ESP32, з вбудованими модулями для передачі Wi-Fi і Bluetooth, що облегшить подальшу роботу при проєктуванні.

Для виконання функцій імітації натискання і свайпання по екрану смартфона було обрано тактильні кнопки і стік відповідно. Такі компоненти забезпечать ефективну роботу користувача з пристроєм і не стануть завадою в процесі проєктування пристрою. У вимогах до АПЗ вказано компоненти для створення ергономічного пристрою, котрий дозволить ефективніше зануритися в ігровий процес аркадних відеоігор.

2 ПРОЄКТУВАННЯ КОНТРОЛЕРА ДЛЯ АРКАДНИХ ВІДЕОІГОР

2.1 Будова контролера

Для кращого розуміння будови ігрових контролерів гарним рішенням є розібрати одне з готових рішень і ознайомитися з підбором комплектуючих компонентів для цього рішення, їх з'єднанням між собою в готовому контролері. Для прикладу буде використовуватися контролер типу геймпад, адже такий вид контролеру найбільш популярний серед геймерів в усьому світі.

Розібравши один з таких контролерів, стало відомо, що основною частиною є плата (рис. 2.1), на котрій знаходяться системні кнопки, функціональні кнопки та кнопки D-pady, сигнали з яких передаються по кабелю USB до комп'ютера де з допомогою спеціального програмного забезпечення (драйверу) опрацьовуються і транспортуються в дії в грі. Кабель котрим під'єднано геймпад до ПК має чотири проводи, котрі під'єднані до відповідних комірок на платі, це GND, 5V, D-1 і D-2. Останні дві відповідають за передачу цифрового сигналу.



Рисунок 2.1 – Основна плата геймпаду

Кнопки спускових гачків з обох сторін знаходяться на окремих незалежних платах (рис. 2.2) і відповідають тільки за натискання двох кнопок, триггера і бампера відповідної сторони. Під'єднуються ці плати до основної за допомогою шлейфу.



Рисунок 2.2 – Плата спускових гачків

Так само за допомогою шлейфу до основної плати під'єднується плата котра відповідає за захоплення сигналів зі стіків (рис. 2.3). До відповідної плати припаяні блоки стіків.

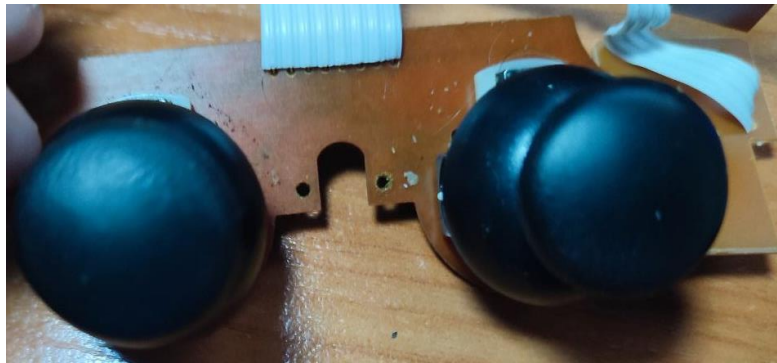


Рисунок 2.3 – Плата блоку стіків

Додатково до основної плати приєднані два вібромоторчики (рис. 2.4) які створюють вібрацію в залежності від виконаних дій в грі, організовуючи тим самим функцію зворотнього зв'язку. Приєднуються вони двома проводами які відповідають за живлення і передачу сигналу про те що час застосувати функцію зворотнього зв'язку. Останнім але не менш важливим є резинові вставки між кожною кнопкою на корпусі пада і відповідною кнопкою на платі, адже без цього шару резини пристрій швидко вийшов би з ладу.

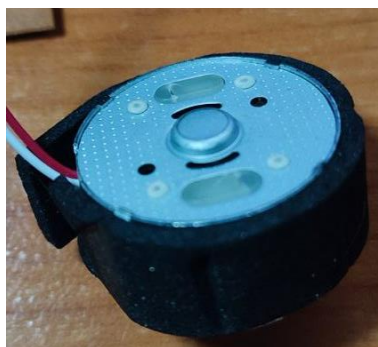


Рисунок 2.4 – Вібромоторчик

Відмінністю провідного паду від безпроводного буде вбудований слот під акумулятор (батарею) яка буде під'єднана до тих самих комірок GND і 5V, та блютуз модуля, котрий буде отримувати цифрові сигнали і передавати їх через блютуз хвилі на комп'ютер де в програмного забезпеченні ці хвилі приймаються і перетворюються на ті самі дії в грі. Єдина відмінність у зручності такого геймпаду це те, що його можна використовувати знаходячись набагато далі від екрану комп'ютера, адже гравець не обмежений довжиною кабелю.

2.2 Передавання даних між девайсом і смартфоном

Головним компонентом в проектуванні бездротового контролеру є забезпечення з'єднання між самим контролером і смартфоном з допомогою функцій бездротової передачі. Для цього була обрана функція Bluetooth, адже обраний мікроконтролер ESP32 обладнаний вбудованим передавачем з підтримкою Bluetooth, так само як і будь-який сучасний смартфон обладнаний вбудованим Bluetooth приймачем.

Мікроконтролер ESP32 підтримує використання Bluetooth двох стандартів – Bluetooth 4.2 і Bluetooth Low Energy. Bluetooth 4.2 і Bluetooth Low Energy – це два різних аспекти технології Bluetooth, які варто розглянути окремо, але вони часто співіснують і можуть використовуватися разом в різних пристроях.

Bluetooth 4.2 є версією базового стандарту Bluetooth. Це остання версія до Bluetooth 5.0 і була випущена у 2014 році. Bluetooth 4.2 має покращену швидкість передачі даних, покращену конфіденційність та безпеку з'єднання, оптимізацію енергоспоживання.

BLE – це розширення стандарту Bluetooth, яке було додане до специфікації з Bluetooth 4.0. В основному використовується для забезпечення низького енергоспоживання, що робить його ідеальним для низькоенергетичних пристроїв, таких як датчики, вимірні прилади,

взуття та одяг з підключенням до Інтернету (IoT). BLE працює на версіях Bluetooth 4.0 і вище, включаючи 4.2.

Отже, основна різниця між Bluetooth 4.2 і BLE полягає в тому, що Bluetooth 4.2 є версією основного стандарту Bluetooth з різними покращеннями, в той час як BLE є підмножиною Bluetooth 4.0 і вище, спеціально розробленою для забезпечення низького енергоспоживання. Багато сучасних пристроїв, які підтримують Bluetooth 4.2, можуть також підтримувати BLE для підтримки низькопотужних сенсорів та інших пристроїв IoT.

Розглянемо що собою являє технологія Bluetooth, і якими методами вона передає дані, щоб визначити для себе який з стандартів запропонованих мікроконтролером ESP32 краще використовувати для проєктування ігрового контролера. До того ж, проаналізувавши технологію Bluetooth з середини програмування девайсу стане значно простішим.

2.2.1 Опис технології

Технологія бездротового зв'язку малого радіусу дії Bluetooth дозволяє двом пристроям з'єднуватися безпосередньо, не потребуючи додаткової мережевої інфраструктури, наприклад, бездротового маршрутизатора або точки доступу. Сьогодні технологія Bluetooth найчастіше використовується людьми по всьому світу для підключення різноманітних периферійних пристроїв, таких як бездротові навушники, клавіатури, миші та колонки, як до ПК, так і до мобільних пристроїв.

Технологія Bluetooth працює на радіочастотах у діапазоні 2,4 ГГц. Сьогодні використовуються два стандарти Bluetooth: Bluetooth Classic та Bluetooth Low Energy.

Bluetooth Classic варіант котрий підтримує дві різні швидкості передачі даних: базову швидкість і підвищену швидкість передачі даних.

Bluetooth Low Energy варіант котрий оптимізований для низького енергоспоживання і в основному використовується для додатків, які обмежені

часом роботи від акумулятора. Bluetooth Low Energy зазвичай не використовується для обміну великими обсягами даних, але пропонує підтримку більш високої якості звуку і більш різноманітні можливості прослуховування, ніж Bluetooth Classic.

У перші дні багато комп'ютерів поклалися на зовнішні приймачі для підключення до периферійних пристроїв Bluetooth, часто використовуючи порт USB. Такий підхід забезпечував необхідну функціональність Bluetooth, але часто був громіздким для кінцевого користувача. Сьогодні більшість ПК оснащені вбудованими мережевими картами з функціями Wi-Fi і Bluetooth. Вони забезпечують кращу продуктивність і координацію роботи обох радіотехнологій, допомагаючи запобігти перешкодам і забезпечити достатню швидкість передачі даних.

Технологія Bluetooth в основному використовується для бездротового підключення периферійних пристроїв до смартфонів, настільних комп'ютерів і ноутбуків. До найпоширеніших аксесуарів Bluetooth належать миші, клавіатури, колонки та навушники. Багато ігрових контролерів також використовують технологію Bluetooth для бездротового підключення.

Щоб врахувати велику кількість пристроїв, які користувачі можуть захотіти підключити за допомогою технології Bluetooth, багато сучасних пристроїв підтримують декілька Bluetooth-з'єднань одночасно. Наприклад, у сучасному сценарії роботи вдома користувачі можуть синхронізувати такі пристрої, як клавіатури та миші, з кількома ПК одночасно і перемикатися між ними одним натисканням кнопки.

З'єднання двох Bluetooth-пристроїв – це простий процес, який часто називають паруванням. Багато сучасних Bluetooth-пристроїв автоматично переходять у режим парування при першому ввімкненні. В інших випадках, можливо, доведеться активувати цей режим вручну.

Для Bluetooth-пристроїв парування схоже на обмін контактною інформацією. Спочатку кожен пристрій реєструє інформацію про отримання пари, включно з ключем безпеки, на своєму пристрої-партнері. Ця інформація

зберігається на обох пристроях, тому вони можуть легко підключатися або автоматично повторно підключатися без повторення початкового процесу.

Залежно від пристрою, початковий процес сполучення може також включати порівняння рядка чисел, що відображаються на обох пристроях, щоб переконатися, що ви встановили правильне з'єднання, або введення PIN-коду з одного пристрою на інший. Це робиться для того, щоб запобігти випадковому з'єднанню з неправильним пристроєм і не дати небажаним особам отримати доступ до ваших Bluetooth-пристроїв.

Фізичне розміщення пристрою та навколишні умови мають вирішальне значення для роботи Bluetooth. Технологію Bluetooth можна використовувати для з'єднання пристроїв, які знаходяться на відстані приблизно від 30 до 300 футів один від одного. Радіус дії Bluetooth вашого пристрою залежить від таких факторів, як навколишнє середовище, перешкоди та потужність передачі. Технології, що працюють у діапазоні 2,4 ГГц, такі як старі ПК, планшети, смартфони та мережі Wi-Fi, а також мікрохвильові печі та радіояні, можуть перешкоджати роботі Bluetooth.

В основному, технологія Bluetooth використовується для передачі невеликих пакетів даних і зв'язку між пристроями з низькою затримкою; однак, вона не є кращою для передачі великих файлів через обмежену швидкість передачі даних у 2 Мбіт/с.

Як і у випадку з усіма іншими видами бездротового зв'язку, безпека є головним міркуванням для розробників і користувачів Bluetooth. Протягом довгого життя технології Bluetooth, її можливості безпеки розвивалися разом з покращенням продуктивності та розширенням сфер використання.

Щоб захистити дані під час передачі, технологія Bluetooth покладається на технології шифрування та автентифікації. Процеси сполучення, такі як введення PIN-коду, також допомагають захистити від небажаного доступу або підключення. Порівняно з Bluetooth Classic, Bluetooth Low Energy пропонує розширені функції безпеки, відомі як Low Energy Secure Connections, які

включають вдосконалену алгоритмічну генерацію ключів безпеки, що ускладнює зловмисникам доступ до пристроїв.

2.2.2 Bluetooth Classic

Технологія Bluetooth Classic, також відома як Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR), – це малопотужний модуль, який передає дані по 79 каналах у неліцензованому промисловому, науковому та медичному (ISM) діапазоні частот 2,4 ГГц. Підтримуючи зв'язок між пристроями типу "точка-точка", Bluetooth Classic в основному використовується для бездротового потокового аудіо і став стандартним радіопротоколом для бездротових динаміків, навушників та автомобільних розважальних систем. За допомогою Bluetooth Classic також можна використовувати програми для передачі даних, зокрема мобільний друк.

2.2.3 Bluetooth Low Energy

Технологія Bluetooth Low Energy призначена для роботи з дуже низьким енергоспоживанням. Передаючи дані по 40 каналах у неліцензованому ISM-діапазоні частот 2,4 ГГц, ця технологія надає розробникам величезну гнучкість у створенні продуктів, які відповідають унікальним вимогам до зв'язку на їхньому ринку. Bluetooth Low Energy підтримує кілька топологій зв'язку, розширюючись від "точка-точка" до широкомовної і, останнім часом, комірчастої, що дозволяє підтримувати створення надійних, великомасштабних мереж пристроїв. Спочатку відома своїми можливостями зв'язку між пристроями, зараз ця технологія широко використовується як технологія позиціонування пристроїв для задоволення зростаючого попиту на послуги високоточного визначення місцезнаходження в приміщенні, включаючи функції, які дозволяють одному пристрою визначати присутність, відстань і напрямок руху іншого пристрою.

2.2.4 Порівняльна таблиця

Для остаточного ознайомлення і порівняння технологій була створена таблиця 2.1, котра описує усі основні характеристики і недоліки технологій.

Таблиця 2.1 – Порівняння стандартів Bluetooth

Характеристика	Bluetooth Low Energy	Bluetooth Classic
Діапазон частот	2,4 ГГц ISM діапазон (2,402 – 2,480 ГГц)	2,4 ГГц ISM діапазон (2,402 – 2,480 ГГц)
Канали	40 каналів з інтервалом 2 МГц (3 рекламні канали/37 каналів передачі даних)	79 каналів з інтервалом 1 МГц
Використання каналів	Частотно-стрибкоподібне розширення спектру (FHSS)	Частотно-стрибкоподібне розширення спектру (FHSS)
Модуляція	GFSK	GFSK, $\pi/4$ DQPSK, 8DPSK
Швидкість передачі даних	LE 2M PHY: 2 Мбіт/с LE 1M PHY: 1 Мбіт/с LE Coded PHY (S=2): 500 Кбіт/с LE Coded PHY (S=8): 125 Кбіт/с	EDR PHY (8DPSK): 3 Мбіт/с EDR PHY ($\pi/4$ DQPSK): 2 Мбіт/с BR PHY (GFSK): 1 Мбіт/с
Потужність передавача	≤ 100 мВт (+20 дБм)	≤ 100 мВт (+20 дБм)
Чутливість приймача	LE 2M PHY: ≤ -70 дБм LE 1M PHY: ≤ -70 дБм LE Coded PHY (S=2): ≤ -75 дБм LE Coded PHY (S=8): ≤ -82 дБм	≤ -70 дБм
Транспортування даних	Асинхронний з з'єднанням Ізохронний з з'єднанням Асинхронний без підключення Синхронні без підключення Ізохронні з'єднання	Асинхронний, орієнтований на з'єднання Синхронний, орієнтований на з'єднання
Комунікаційні топології	Точка-точка (включаючи піконет) Трансляція Мережевий	Точка-точка (включаючи піконет)

Проаналізувавши зібрані дані обраним стандартом передачі даних був обраний Bluetooth Low Energy, адже він споживає менше енергії під час процесу передачі даних, а за іншими характеристиками в яких він «програє» стандарту Bluetooth Classic немає потреби.

2.3 Опис роботи функціональних компонентів

Першим і основним функціональним компонентом є стік, котрий буде імітувати свайпи по екрану Смартфона. Стік має 5 виводів: VCC (+ 5D), GND, X, Y і Button. Підключивши живлення до стіку можна знімати з виводів X, Y і Button статуси поточного становища стіка. Вивід X відповідає за зміщення стіка в горизонтальному положенні, вивід Y відповідає за зміщення стіка в вертикальному положенні. Конструкція стіку дозволяє одночасне зміщення як по горизонталі, так і по вертикалі. Вивід Button відповідає за натискання вбудованої в конструкцію стіка кнопки. На виходах X і Y сигнал диференційований, тобто змінюється від кута нахилу ручки.

При використанні напруги живлення 5 В, за замовчуванням на аналогових виводах X і Y буде 2.5 В. При переміщенні джойстика в одну сторону напруга буде наростати до 5 В, при русі джойстика в іншу сторону напруга буде падати до 0В. Виводи підключаються до аналогових входів ESP32. Таким чином можна отримувати точне положення ручки джойстика і реагувати на кут нахилу, а не тільки на сам факт нахилу ручки.

Розміри плати джойстика – 24 x 36 мм, відстань між кріпильними отворами – 27 x 21мм, діаметр отворів для кріплення – 3 мм.

Розпіновка стіка зображена на рис. 2.5.

Другим функціональним компонентом є тактильна кнопка розмірами 6 x 6 x 7 мм. Кнопка двоконтактна на 4 виводи, в нормальному стані розімкнута. Передає сигнал тільки при натисканні на кнопку, функції затискання немає. Характеристики:

- діапазон температур: -20 °С ... +60 °С;
- опір контактів: 30 мОм;

- максимальна напруга: 250В (протягом 1 хвилини);
- ліміт натисків: 10 000.

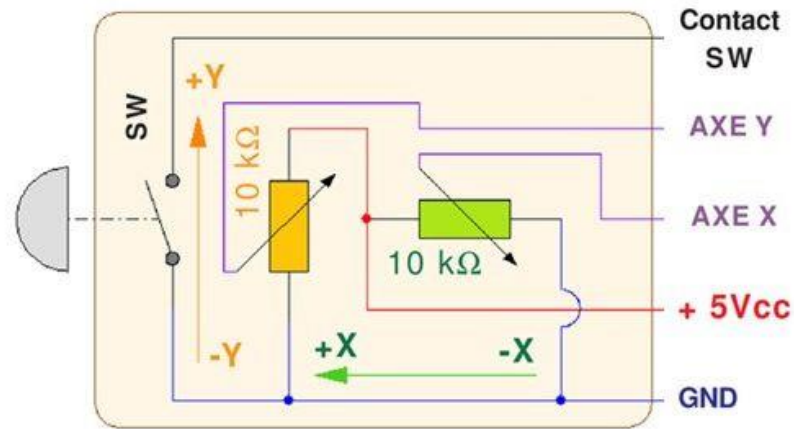


Рисунок 2.5 – Розпіновка компоненту стік

Підключення такої кнопки до мікроконтролера ESP32 є простим процесом, так як за конструкцією такої кнопки дві пари ніжок електрично з'єднані між собою і натискання на саму кнопку замикає контакти і з'єднує одну пару з іншою. Одна пара ніжок підключається до будь-якого цифрового виходу на ESP32. Друга пара ніжок підключається до загальної землі. Додатково для стабілізації роботи кнопки може бути підключений підтягувальний резистор. Схема кнопки зображена на рис. 2.6.

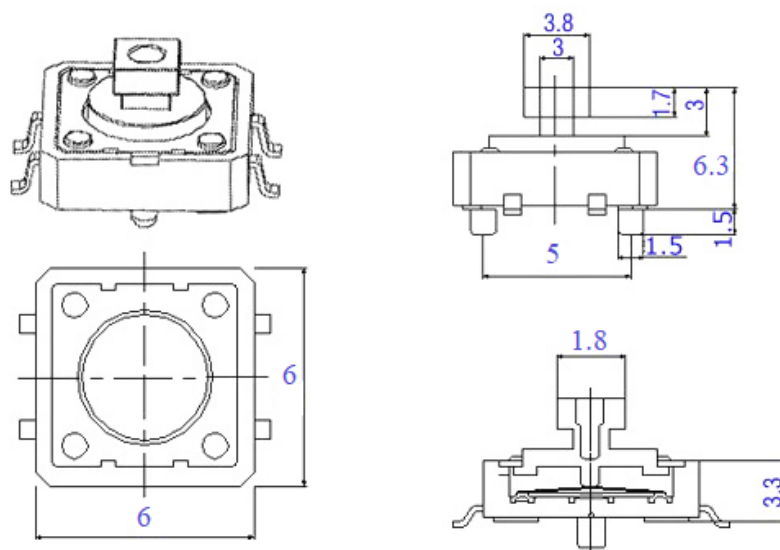


Рисунок 2.6 – Схематичне зображення кнопки

Використання такої кнопки в пристрої дозволить додавати певні функціональні дії при натисканні на кнопку, додатково сигналізуючи користувачу про натискання, видаючи конкретний звук клацання.

2.4 Підключення елементів

Для початку підключення елементів слід ознайомитися з розпіновкою мікроконтролера ESP32 (рис. 2.7), адже надалі в роботі будуть описуватися підключення до певних аналогових і цифрових входів і виходів різних елементів системи, тому важливо мати уявлення про піни до яких будуть під'єднуватися різні компоненти.

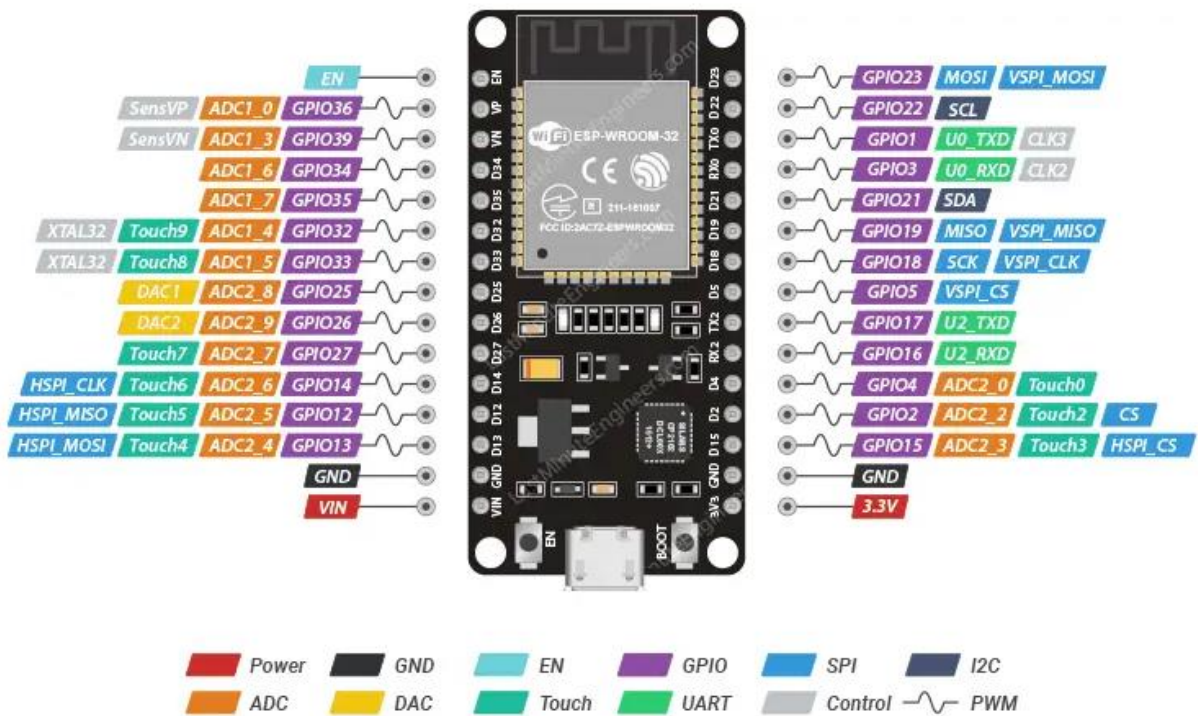


Рисунок 2.7 – Розпіновка мікроконтролера ESP32

Мікросхема ESP32 має 48 виводів з різними функціями. Не всі виводи доступні на всіх варіантах плат ESP32, а деякі виводи взагалі не можуть бути використані. На малюнку вище показано розводку виводів ESP-WROOM-32. Цей малюнок можна використовувати як довідник при проектуванні власних систем використовуючи ESP32.

2.4.1 GPIO піни

На платі ESP32 є 25 виводів GPIO, яким можна призначити різні функції, попередньо їх запрограмувавши. Існує кілька типів GPIO: тільки цифрові, з підтримкою аналогових сигналів, з підтримкою ємнісних сигналів тощо. Аналогові та ємнісні GPIO можуть бути налаштовані як цифрові GPIO. Більшість таких цифрових GPIO можна налаштувати з внутрішнім підтягуванням або відтягуванням, або налаштувати на високий опір.

Хоча ESP32 має багато контактів з різними функціями, деякі з них можуть не підходити для використання в певних проєктах. У табл. 2.2 нижче показано, які контакти безпечні для використання, а які слід використовувати з обережністю.

Таблиця 2.2 – Функціональність GPIO пінів

Позначення піну	GPIO	Чи можна використовувати	Причина
D0	0	Треба бути обережним	Має бути значення HIGH під час завантаження та LOW для програмування
TX0	1	Ні	Тх вивід, використовується для прошивки та налагодження
D2	2	Треба бути обережним	Під час завантаження має бути значення LOW, а також підключений до бортового світлодіода
RX0	3	Ні	Вивід Rx, використовується для прошивки та налагодження
D5	5	Треба бути обережним	Під час завантаження має бути значення HIGH
D6	6	Ні	Підключений до флеш-пам'яті
D7	7	Ні	Підключений до флеш-пам'яті
D8	8	Ні	Підключений до флеш-пам'яті
D9	9	Ні	Підключений до флеш-пам'яті
D10	10	Ні	Підключений до флеш-пам'яті
D11	11	Ні	Підключений до флеш-пам'яті
D12	12	Треба бути обережним	Під час завантаження має бути значення LOW

Позначення піну	GPIO	Чи можна використовувати	Причина
D15	15	Треба бути обережним	Використовується тільки для входу, не може бути налаштований як вихід
D34	34	Треба бути обережним	Використовується тільки для входу, не може бути налаштований як вихід
D35	35	Треба бути обережним	Використовується тільки для входу, не може бути налаштований як вихід
VP	36	Треба бути обережним	Використовується тільки для входу, не може бути налаштований як вихід
VN	39	Треба бути обережним	Використовується тільки для входу, не може бути налаштований як вихід

Інші піни які не описані в таблиці можна вільно використовувати для своїх потреб не хвилюючись про якісь наслідки чи перебої в роботі.

2.4.2 Безпосереднє підключення компонентів

Для забезпечення правильної роботи компоненту типу стік, його треба під'єднати до ESP32. Першочергово забезпечуємо живлення компоненту. Це відбувається шляхом з'єднання виходу + 5 В на компоненті і виходу 3.3 В на мікроконтролері. Слід зазначити, що використання різних напруг на компонентах в таких системах може негативно вплинути на подальшу роботу компонентів, проте, якщо наш компонент стік потребує 5В на своєму вході для коректної роботи, це не означає що він не буде працювати при подачі 3.3В від мікроконтролера. Ми живимо стік від мікроконтролера, а не навпаки, тому він жоден з компонентів не має вийти з ладу, єдиною проблемою буде потреба в програмному налаштуванні нехватки напруги на компоненті.

Наступним кроком є забезпечення спільного заземлення компонентів, що забезпечується з'єднанням відповідних контактів GND на обох компонентах. Забезпечення передачі даних про зміну положення осей X та Y на стіку відбувається шляхом підключення однойменних виходів на

компоненті до будь-яких вільних аналогових пінів на мікроконтролері. В нашому випадку такими пінами будуть піни GPIO34 і GPIO35. Попри описані вище складнощі і небезпеки у використанні певних пінів, до складу яких входять обидва цих піни, основними їх недоліками є те, що вони не можуть бути налаштованими як виходи, тобто не мають змоги передавати данні з мікроконтролера назад на стік, що нам власне і не потрібно, адже ми будемо тільки зчитувати дані зі стіку. Вихід кнопки стіку має бути під'єднаний до будь-якого цифрового піну, в нашому випадку це є пін GPIO32. Він забезпечить правильне функціонування кнопки при натисканні на неї.

Для забезпечення правильної роботи тактильної кнопки її з'єднання з ESP32 відбувається певним чином. Одна пара ніжок кнопки під'єднується до загальної землі (GND) на мікроконтролері. Інша пара ніжок під'єднується до цифрового входу на мікроконтролері. В нашому випадку це пін GPIO33.

Для розуміння системи і порядку підключення компонентів між собою було використано сервіс Wokwi для онлайн проєктування мікроконтролерів ESP (рис. 2.8).

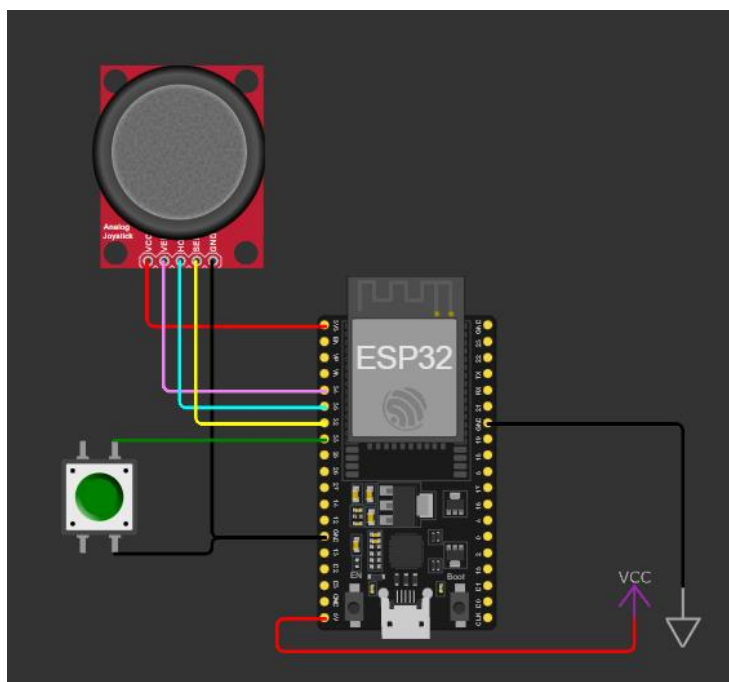


Рисунок 2.8 – Схематичне зображення під'єднання компонентів в системі

Під час проєктування пристрою живлення забезпечувалося через USB порт з персонального комп'ютера, для забезпечення стабільного живлення в процесі тестування роботи пристрою і облегшення процесу програмування. Для забезпечення бездротового живлення контролера буде використовуватися батарейка, позитивний контакт якої буде під'єднано до входу живлення 3.3В на мікроконтролері, а негативний буде під'єднано до загальної землі GND.

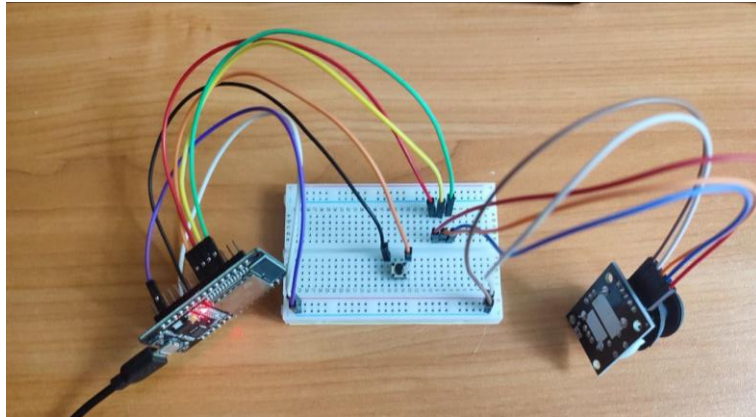


Рисунок 2.9 – Зібраний пристрій

Під'єднання всіх компонентів в одну систему відбувалося за рахунок макетної плати (рис. 2.9), для спрощення процесу проєктування і програмування пристрою, і за відсутності відповідних проводів типу мама-мама для підключення компонентів між собою без використання макетної плати.

2.5 Опис системи живлення

Так як контролер має бути бездротовим, то варіанти вибору елемента живлення зменшуються до використання акумуляторів різних типів. Важливо зазначити ESP32 працює від напруги 2,5–3,6 В. Найпоширенішим джерелом живлення є блок живлення 3,3 В.

Серед варіантів акумуляторів доступні три види для живлення ESP32. Це літій-іонні акумулятори (Li-Ion), літій-полімерні акумулятори (Li-Pol), літій-фератні акумулятори (LiFePO₄). Літій-іонні акумулятори являються найпоширенішим типом акумуляторів для ESP32. Вони мають високу

щільність енергії, довгий термін служби та доступні за ціною. Літій-полімерні акумулятори схожі на попередній тип, але вони більш гнучкі та легкі. Літій-фератні акумулятори більш безпечні та стійкі до пожеж, ніж обидва минулих варіанти, проте вони мають меншу щільність енергії.

В зв'язку з меншими номінальними показниками напруги при роботі з обраним мікроконтролером нам знадобиться акумулятор з номінальною напругою 3,7 В якщо це літій-іонний та літій-полімерний або 3,3 В якщо це літій-фератний акумулятор.

Ємність акумулятора вимірюється в міліампер-годинах (мАг), і чим більша ємність, тим довше акумулятор буде працювати. Вибір ємності акумулятора залежить від того, як довго контролер в середньому буде використовуватися під час ігрового процесу. Загалом, якщо брати роботу контролера весь час з максимальним споживанням в 300 мА, то акумулятора ємністю в 1500–2500 мАч буде вистачати від трьох до восьми годин безперервного користування. Тому вибір акумулятора припадає на літій-іонний з ємністю в 2000 мАч(рис. 2.10), якого вистачить на тривалу роботу в 6 годин без перезарядження.



Рисунок 2.10 – Приклад акумулятора

Ще одним фактором є процес зарядження вище зазначеного акумулятора, адже постійно купувати новий для роботи в 6 годин не є оптимальним рішенням. Тому додатково для користування цим контролером треба мати зарядний пристрій для відповідного акумулятора щоб отримувати повний доступ до функціонування контролеру.

2.6 Алгоритм роботи

Після ввімкнення пристрою до джерела живлення починається процес ініціалізації підключених компонентів. Після ініціалізації усіх компонентів пристрій починає роботу Bluetooth модуля, шукаючи девайс з яким можна зробити пару.



Рисунок 2.11 – Блок-схема алгоритму роботи пристрою

Після успішного парування девайсів починається процес зчитування інформації з тактильної кнопки та стіку під'єданого до мікроконтролера. Якщо буде зафіксовано будь-які зміни в положенні осей стіка, натискання на вбудовану в нього кнопку, або натискання на тактильну кнопку.

ESP32 отримує ці зміни в значеннях, обробляє і передає їх через Bluetooth до смартфона як уже готові команди для керування. В той час смартфон отримує оброблені дані і виконує передані команди без безпосередньої потреби в контакті з самим девайсом.

Після моменту ввімкнення пристрою і з'єднання по технології Bluetooth з будь-яким мобільним девайсом, пристрій починає безперервно зчитувати данні, обробляючи і передаючи ці данні на смартфон до моменту поки не буде розірвано Bluetooth з'єднання з девайсом, або поки не розрядиться акумулятор пристрою. В першому випадку пристрій продовжить роботу в режимі пошуку пари для з'єднання, в другому просто завершить свою роботу. Блок-схема роботи пристрою представлена на рис. 2.12.

Висновки до другого розділу

В другому розділі було розглянуто процес покрокового підключення всіх компонентів системи в єдиний пристрій, включаючи ознайомлення з мікроконтролером ESP32 і його схемою розпіновки, з компонентом імітації свайпів по екрану смартфона стік і тактильною кнопкою. Було описано процес живлення пристрою в процесі моделювання і програмування, так само як і розраховано потрібне живлення для пристрою під час бездротового з'єднання.

Було проведено ознайомлення з функціоналом технології передачі даних Bluetooth, проведено аналіз стандартів технології і в результаті аналізу було обрано найбільш підходящий для проєктування бездротового контролера стандарт – Bluetooth Low Energy. Це виконувалося з розрахунком облегшення роботи при написанні програмної частини пристрою.

Було побудовано і описано повний алгоритм роботи пристрою від початку підключення до живлення, процесу парування через технологію Bluetooth, зчитування та опрацювання даних, прийому цих даних смартфоном і переведення цих даних в команди для роботи з ним. За таким алгоритмом роботи користувач отримуватиме повну і безперервну взаємодію з контролером.

3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АРКАДНИХ ВІДЕОІГОР

3.1 Підготування середовища програмування

3.1.1 Обрання середовища для розробки

ESP32 підтримує декілька середовищ розробки (IDE), що робить його гнучким та зручним для різних типів проєктів. Основними середовищами, котрі дозволяють програмувати ESP32 є:

- **Arduino IDE**, одне з найпопулярніших середовищ для програмування ESP32. Воно має простий інтерфейс і велике співтовариство користувачів, що дозволяє легко знайти допомогу та приклади кодів. Перевагами цього середовища є простий і зручний інтерфейс, велика кількість бібліотек та прикладів для розробки різноманітних проєктів, широка підтримка спільноти. Мінусом є відсутність вбудованої в середовище моделі для ESP32, яку потрібно додатково встановлювати власноруч;

- **PlatformIO**, розширення для Visual Studio Code та інших редакторів коду, що забезпечує професійні можливості для розробки. Перевагами цього варіанту є підтримка різних середовищ та платформ, потужні інструменти для налагодження та аналізу коду, гнучкість у налаштуванні проєктів;

- **Espressif IoT Development Framework (ESP-IDF)**, є офіційним середовищем розробки від Espressif, яке забезпечує повний контроль над всіма аспектами роботи ESP32. Перевагами цього середовища є повний контроль над апаратним забезпеченням, офіційна підтримка від Espressif, розширені можливості для налагодження та профілювання;

- **MicroPython**, дозволяє програмувати ESP32 на Python, що робить його доступним для новачків та тих, хто віддає перевагу скриптовим мовам програмування. Перевагами цього методу є використання простого синтаксису Python для програмування, використання інтерактивного

середовище для швидкого тестування, легка інтеграція з іншими Python-бібліотеками. Мінусами є потреба в встановленні прошивки MicroPython на ESP32;

- Eclipse IDE, може бути налаштований для розробки з ESP-IDF, надаючи професійні інструменти для налагодження та управління проектами. Перевагами є розширені можливості для розробки та налагодження, підтримка великих проєктів;

- програмування на Lua через NodeMCU, середовище, що дозволяє програмувати ESP32. Це забезпечує простоту та легкість у використанні, схоже на MicroPython. Перевагами є простий та зрозумілий синтаксис Lua, швидка розробка прототипів. Мінусами є потреба в встановленні прошивки NodeMCU на ESP32;

В результаті порівняння існуючих варіантів програмування мікроконтролеру ESP32 вибір зупинився на використанні Arduino IDE, через простоту інтерфейсу середовища та зручний синтаксис.

3.1.2 Підготовка Arduino IDE до роботи

Arduino IDE (Integrated Development Environment) є офіційним середовищем розробки для програмування та завантаження програмного забезпечення на платформи Arduino і не тільки. Воно надає користувачам зручний інтерфейс для написання, компіляції та завантаження коду на мікроконтролери Arduino.

Arduino IDE містить простий текстовий редактор (рис. 3.1) для створення програм на мові програмування, яка базується на C/C++. Середовище забезпечує компіляцію написаного коду в машинний код, зрозумілий мікроконтролеру. Вбудований компілятор перевіряє код на наявність помилок і створює виконуваний файл.

Після компіляції, Arduino IDE дозволяє завантажити програму безпосередньо на мікроконтролер через USB-з'єднання, що забезпечує швидкий і зручний спосіб тестування та налагодження коду.

Arduino IDE містить велику кількість вбудованих бібліотек, що спрощують роботу з різними периферійними пристроями, такими як сенсори, модулі зв'язку та дисплеї. Крім того, середовище надає безліч прикладів коду, які можна використовувати як основу для власних проєктів.

Однією з ключових функцій Arduino IDE є монітор послідовного порту (рис. 3.2), який дозволяє відслідковувати дані, що надходять з мікроконтролера через послідовний порт. Це дуже зручно для налагодження та аналізу роботи програм. Встановлення середовища є безкоштовним і відбувається на основному сайті виробника.

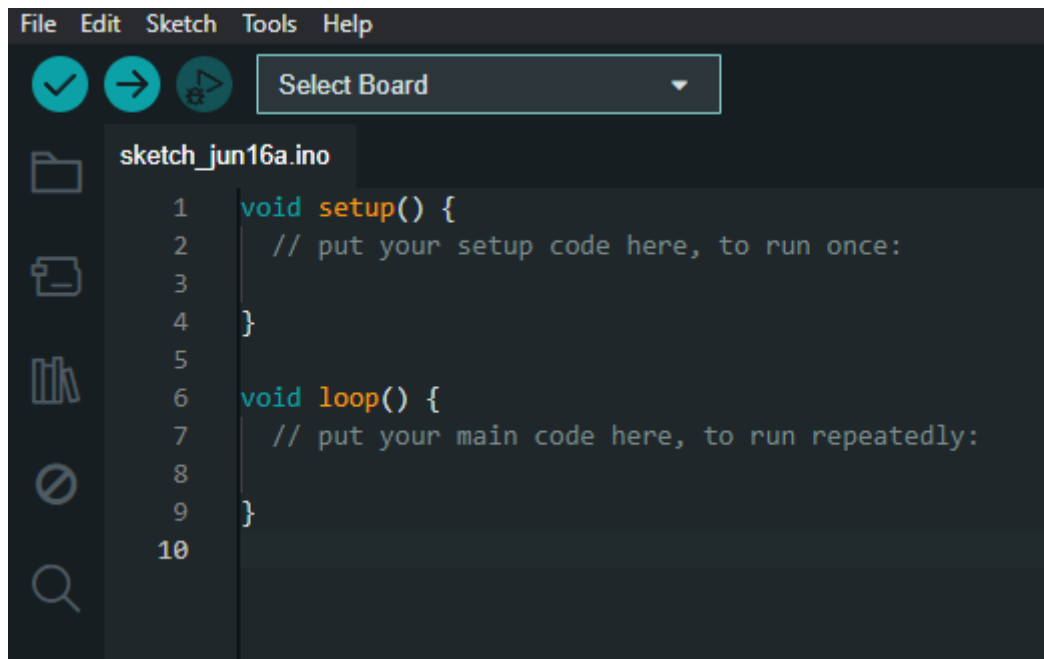


Рисунок 3.1 – Інтерфейс середовища розробки

Після завантаження та встановлення середовища, при першому його запуску відбувається налаштування та конфігурація середовища під систему. Середовище декілька разів просило дозволи на встановлення певних системних компонентів для налаштування його правильної роботи на персональному комп'ютері.


```
Output
Downloading packages
arduino:arduinoOTA@1.3.0
arduino:avr-gcc@7.3.0-atmel3.6.1-arduino7
arduino:avrdude@6.3.0-arduino17
arduino:avr@1.8.6
Installing arduino:arduinoOTA@1.3.0
Configuring tool.
arduino:arduinoOTA@1.3.0 installed
Installing arduino:avr-gcc@7.3.0-atmel3.6.1-arduino7
Configuring tool.
```

Рисунок 3.2 – Вікно монітора послідовного порту з логами налаштування при першому запуску середовища

Наступним кроком після першого налаштування середовища є додавання підтримки мікроконтролера ESP32 в цьому середовищі. Для цього треба дотримувати кроків з офіційної документації щодо використання ESP мікроконтролерів в Arduino IDE.

Першим кроком є запуск Arduino IDE та відкриття вікна Preferences (рис. 3.3)

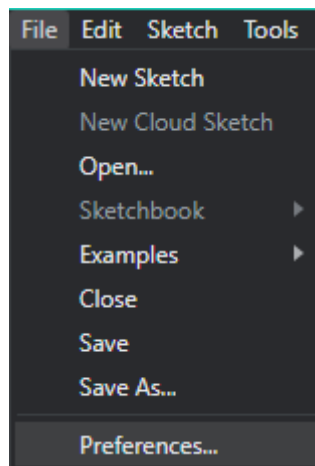


Рисунок 3.3 – Розположення вікна Preferences

Наступним кроком, у відкритому вікні треба знайти поле для вставлення Additional Board Manager URLs (рис. 3.4) і вставити туди посилання на встановлення різних версій. Можна додавати декілька посилань одночасно, перераховуючи їх через кому. Для цієї роботи була обрана остання стабільна

версія підтримки ESP32 в середовищі Arduino IDE, посилання на котру можна знайти в офіційній документації.

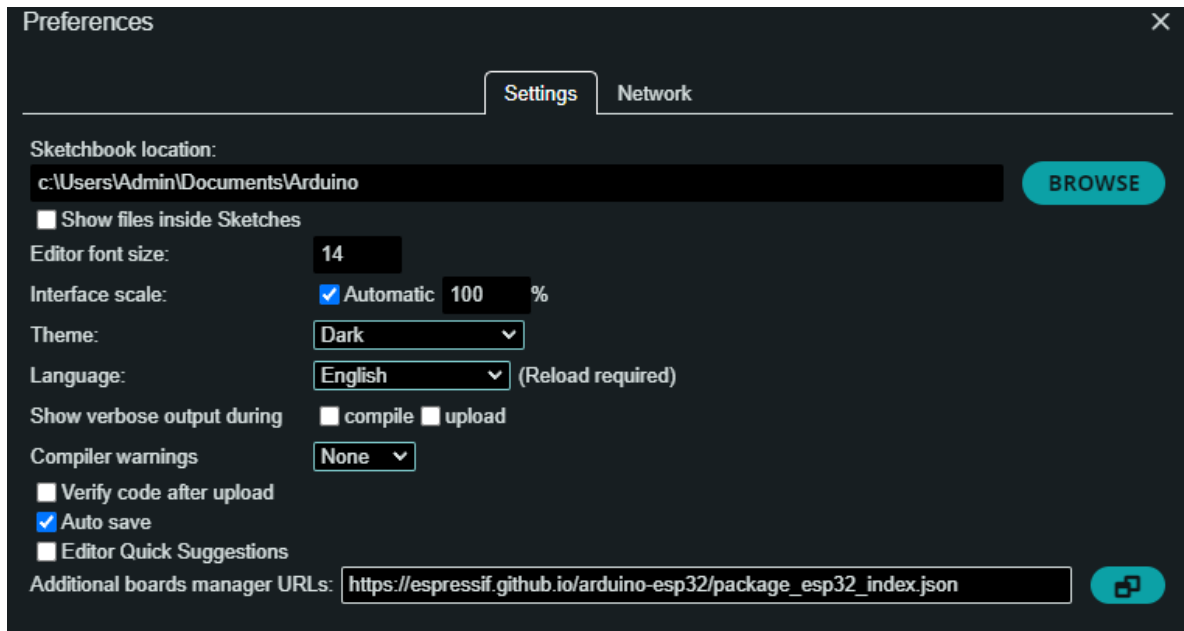


Рисунок 3.4 – Вікно Preferences з вставленим посиланням на останню стабільну версію підтримки ESP32

Після натискання кнопки ОК в цьому ж вікні почнеться процес завантаження відповідних бібліотек, котрий не має зайняти багато часу. Наступним кроком є встановлення платформи ESP32 в середовищі, для цього треба відкрити вікно Boards manager котре можна знайти за шляхом Tools > Board > Boards manager (рис. 3.5).

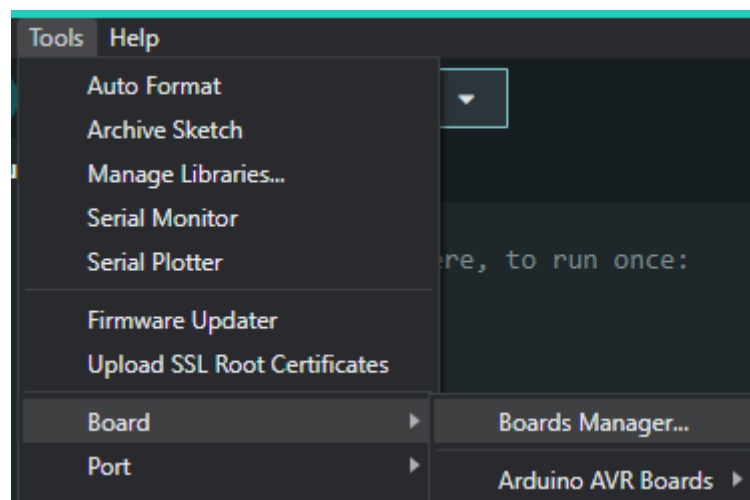


Рисунок 3.5 – Шлях до вікна Boards manager

У відкритому вікні в полі пошуку потрібно ввести «esp32», та натиснути кнопку «install» навпроти з'явившогося у вікні варіанту платформи (рис. 3.6).

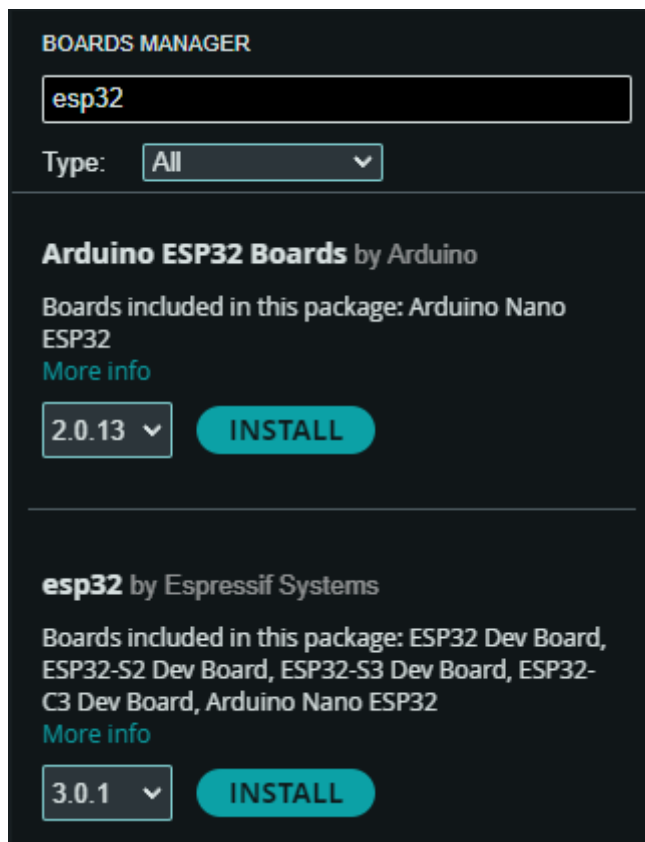


Рисунок 3.6 – Результат пошуку платформи ESP32

В результаті пошуку ми отримали два варіанти для встановлення платформи ESP32 в середовище Arduino IDE. Оберемо варіант від Espressif Systems, тому що саме цей варіант встановлювався в документації.

Після натискання кнопки завантаження, починається сам процес завантаження, по закінченню якого можна починати працювати з ESP32 в середовищі Arduino IDE, проте обов'язковим пунктом треба перезапустити середовище після встановлення платформи, для її коректного відображення і роботи.

3.1.3 Встановлення відповідних бібліотек

Наступним кроком в програмуванні бездротового контролера буде встановлення відповідних існуючих бібліотек для полегшення і покращення самого процесу програмування, адже вбудованих бібліотек середовища

Arduino IDE не вистачає для повного опрацювання і виконання поставленого завдання.

Для виконання поставленого завдання буде використовуватися бібліотека ESP32-BLE-Gamepad, доступ до якої відбувається через систему github. Користуючись документацію щодо встановлення цієї бібліотеки, пройдемося по необхідним крокам.

Першочергово треба встановити останню версію бібліотеки, посилання на котру прикріплене до сторінки документації. Наступним кроком треба відкрити середовище програмування Arduino IDE і перейти у вікно «Add .Zip Library», що можна зробити дотримуючись шляху Sketch > Include Library > Add .ZIP Library (рис. 3.7).

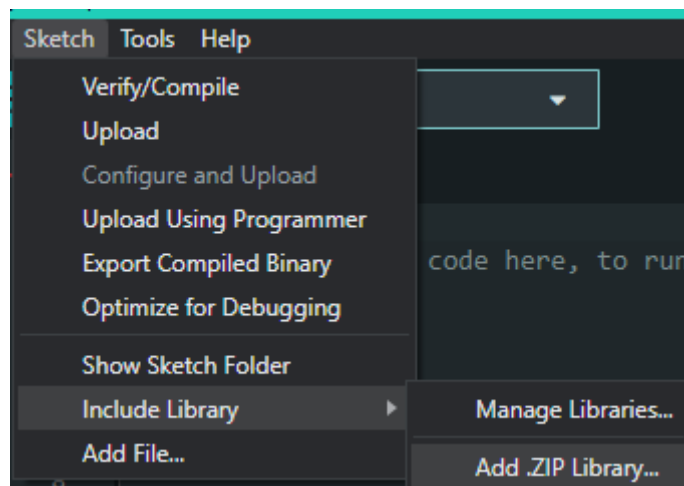


Рисунок 3.7 – Шлях до вікна Add .ZIP Library

Після переходу до цього вікна відривається стандартне вікно пошуку файлів у Windows, в котрому нам треба знайти завантажений раніше архів з бібліотекою і вибрати його (рис. 3.8).

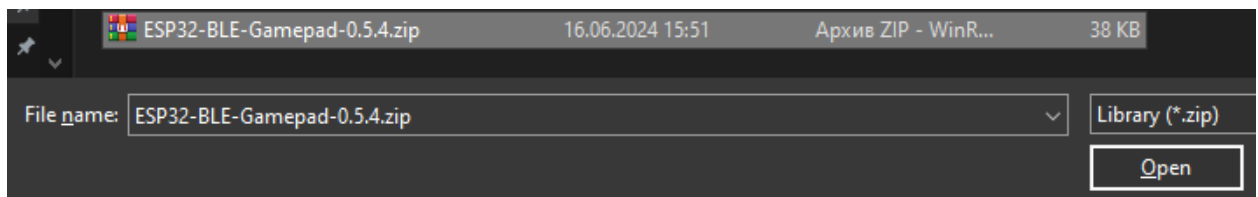


Рисунок 3.8 – Вибір бібліотеки

Наступним кроком треба встановити ще одну додаткову бібліотеку через менеджер бібліотек в середовищі Arduino IDE. Для цього треба перейти за шляхом Tools > Manage Libraries (рис. 3.9) і у вікні що відкриється в полі пошуку ввести «NimBLE-Arduino» (рис. 3.10), і встановити цю бібліотеку, виконуючи такі самі кроки як і при встановленні платформи ESP32.

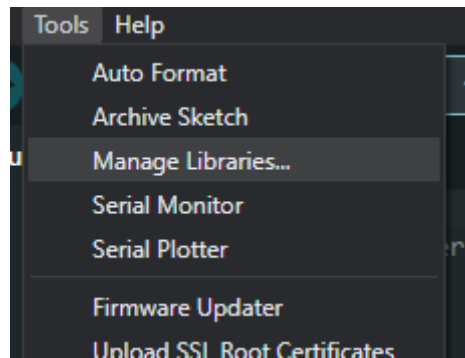


Рисунок 3.9 –Пошук вікна Manage Libraries

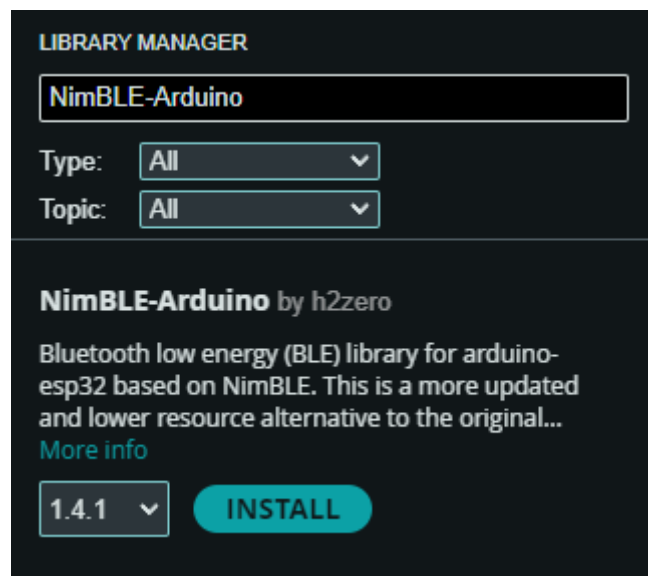


Рисунок 3.10 –Результат пошуку бібліотеки

Після встановлення бібліотеки і перезапуску середовища можна приступати до програмування контролера. Для облегшення завдання, встановленні бібліотеки мають готові екземпляри для роботи з контролерами.

Щоб відкрити один з таких екземплярів потрібно перейти за шляхом File > Examples > ESP32 BLE Gamepad (рис. 3.11) і обрати один з

запропонованих екземплярів. Для виконання цієї роботи було обрано екземпляр з назвою «Gamepad».

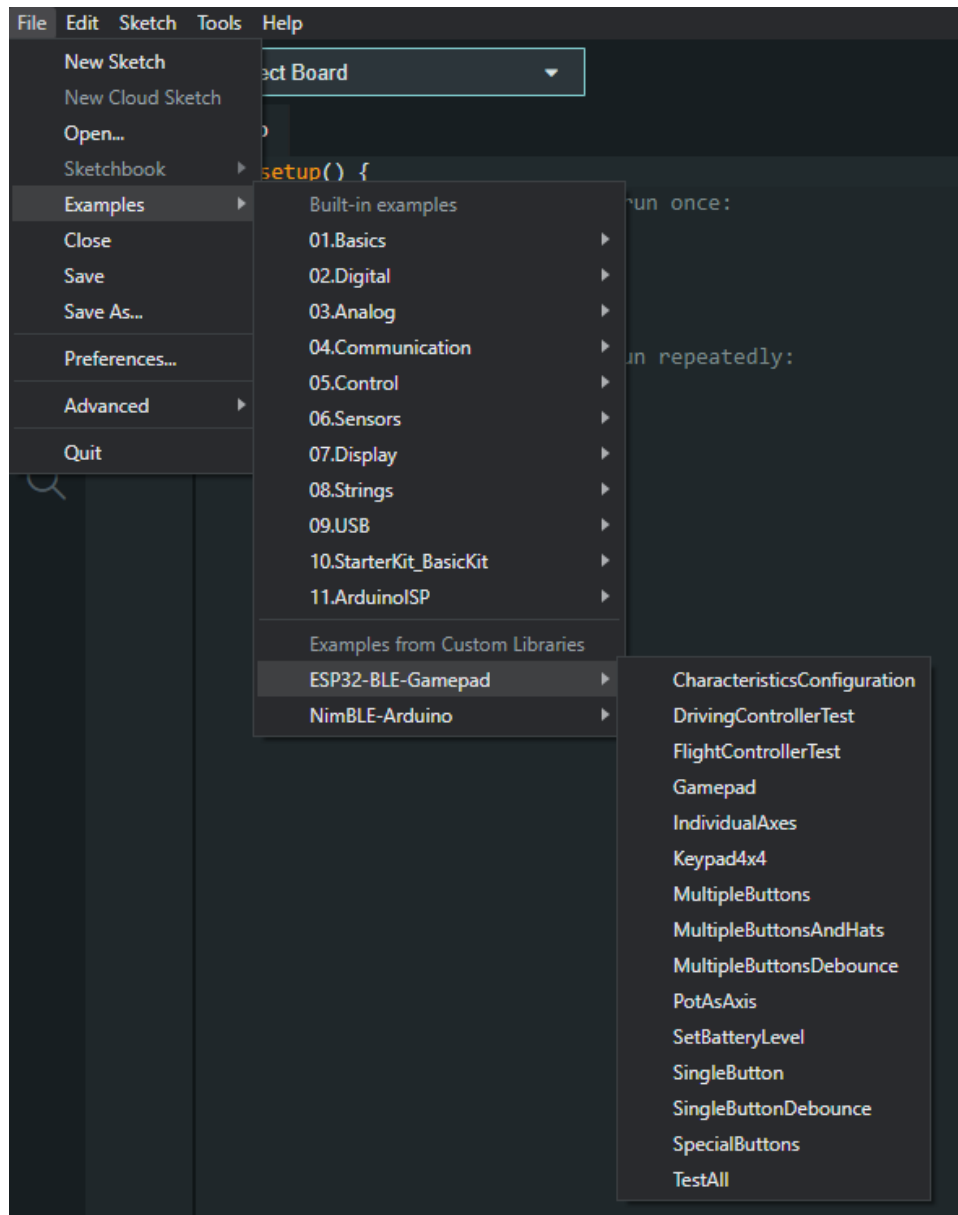


Рисунок 3.11 – Шлях до запропонованих бібліотеками екземплярів

Приклад використання цього екземпляру описаний в документації, тому для спрощення процесу програмування пристрою доцільно буде використовувати цей екземпляр. В документації детально описано правила використання усіх методів котрі є в обраному екземплярі, що пришвидшить процес.

3.2 Програмування контролера

Переходимо до проєктування програмної частини контролера. Основним компонентом у нас є мікроконтролер ESP32, котрий буде опрацьовувати дані і передавати їх по технології блютуз. Саме тому першочергово ми ініціалізуємо цей модуль. Ініціалізація компонента представляє собою налаштування самого компонента і його програмування на виконання певних функцій. В нашому проєкті ініціалізування блютуз модулю відбувається за рахунок бібліотеки ESP32 BLE Gamepad, з підключення котрої ми і почнемо, додатково описуючи параметри нашого контролера при підключенні за технологією Bluetooth, такі як назва, виробник і заряд батареї підключеного девайса (рис. 3.12). В нашому випадку ці значення є «ESP Controller» для назви контролера, «BohdanPavlenko» як ім'я виробника контролера.

```
#include <Arduino.h>
#include <BleGamepad.h>

BleGamepad bleGamepad("ESP Controller", "BohdanPavlenko");
```

Рисунок 3.12 – Підключення головної бібліотеки і опис параметрів контролера

Наступним кроком в ініціалізації модуля бездротової передачі даних є запуск його роботи, котрий відбувається після початку запуску самого мікроконтролера (рис. 3.13).

```
void setup() {
  Serial.begin(115200);
  Serial.println("Starting BLE work!");
  bleGamepad.begin();
}
```

Рисунок 3.13 – Ініціалізація роботи модуля Bluetooth разом з ініціалізацією роботи контролера

Функція «void setup» визначає основну функцію котра ініціалізує усі компоненти підключенні до мікроконтролера і виконується один раз при запуску системи. «Serial.begin(115200);» ініціалізує серійний зв'язок з бітовою швидкістю 115200 біт/сек. Це використовується для відлагодження та виведення повідомлень на серійний монітор. «Serial.println("Starting BLE work!");» виводить повідомлення «Starting BLE work!» на серійний монітор. Це корисно для відлагодження, щоб знати, що ініціалізація BLE почалася.

«bleGamepad.begin();» цей рядок починає роботу з об'єктом bleGamepad. Цей об'єкт є екземпляром класу встановленої раніше однойменної бібліотеки. Цей метод готує геймпад до роботи та починає регламувати його для підключення з іншими пристроями.

Наступним кроком стало визначення GPIO пінів в системі (рис. 3.14), для цього використовується функція «#define» котра визначає номер GPIO піна і надає йому певне значення, визначене користувачем. В нашому випадку ми використовуємо відповідні позначення для підключених до нашого контролера складових. До піна 33, до якого підключена тактильна кнопка ми надаємо значення А, адже за стандартами проектування контролерів, кнопки з маркуванням А відповідають за найголовніші функції. Так само і кнопки з маркуванням В відповідають за найголовніші функціональні значення. Кнопка В прив'язується до піна 32, до якого підключена вбудована на контролері кнопка. Аналогічно до цього підключені піни 34 і 35, які відповідають за зчитування даних з нахилу осей стіка. Маркування LX і LY означає що функціонал стіка буде прив'язано до функціоналу лівого стіку будь-якого геймпада, котрий відповідає за пересування в іграх. X і Y це відповідні осі.

```
#define A 33
#define LX 34
#define LY 35
#define B 32
```

Рисунок 3.14 – Визначення GPIO пінів

Наступним кроком стало програмування і опису процесу, що відбувається в основному циклі loop(). Рядок if (bleGamepad.isConnected()) перевіряє, чи підключений геймпад до смартфона або іншого пристрою через BLE. Тільки після успішно пройденої перевірки мікроконтролер починає зчитувати дані з функціональних компонентів контролера.

Далі відбувається обробка кнопки з маркуванням А (рис. 3.15). Якщо кнопка А натиснута (значення LOW), відправляється сигнал натискання кнопки BUTTON_1 через бібліотеку BLE Gamepad. В документації про цю бібліотеку написано що значення BUTTON_1 відповідає за кнопку А на звичайних геймпадах. Після короткої затримки кнопка відпускається, і відправляється відповідний звіт.

```
if (bleGamepad.isConnected()) {  
  if (digitalRead(A) == LOW) {  
    Serial.println(F("A Button"));  
    bleGamepad.press(BUTTON_1);  
    bleGamepad.sendReport();  
    delay(10);  
    bleGamepad.release(BUTTON_1);  
    bleGamepad.sendReport();  
  }  
}
```

Рисунок 3.15 – Обробка кнопки А

За аналогічним змістом відбувається обробка кнопки В (рис.3.16). З документації бібліотеки BLE Gamepad значення BUTTON_2 відповідає за кнопку В на звичайних геймпадах.

```
if (digitalRead(B == LOW)) {  
  Serial.println(F("B Button"));  
  bleGamepad.press(BUTTON_2);  
  bleGamepad.sendReport();  
  delay(10);  
  bleGamepad.release(BUTTON_2);  
  bleGamepad.sendReport();  
}
```

Рисунок 3.16 – Обробка кнопки В

Наступним кроком є зчитування і масштабування значень стіка, при повороті однойменного компоненту вздовж обох його осей (рис. 3.17). Значення з аналогових пінів LX і LY зчитуються та масштабуються з діапазону 0 ~ 4095 до 32767 ~ -32767. Це підходить для використання цих значень як осей геймпада. В строчці «bleGamepad.setLeftThumb(value1, value2);» значення осей джойстика встановлюються у контролері і передаються по

```
int value1 = map(analogRead(LX), 0, 4095, 32767, -32767);  
int value2 = map(analogRead(LY), 0, 4095, 32767, -32767);  
bleGamepad.setLeftThumb(value1, value2);  
  
bleGamepad.sendReport();  
delay(1);
```

Рисунок 3.17 – Зчитування і масштабування значень стіка

Bluetooth в строчці «bleGamepad.sendReport();». Значення затримки між роботою основного циклу було виставлено на 1 мілісекунду, для впровадження плавної роботи контролера.

3.3 Перевірка роботи контролера

Після успішного створення програмної частини для контролера, наступним кроком буде тестування роботи створеного девайсу. Для цього спочатку в середовищі Arduino IDE нам треба верифікувати створений код, а потім завантажити його в мікроконтролер використовуючи певні кнопки (рис. 3.18), розташовані в лівій верхній частині програми.



Рисунок 3.18 – Зображення кнопок верифікації, завантаження і дебагу

Після натискання на кнопку завантаження починається відповідний процес, під час якого контролер виводить дані в спеціально виділений для цього термінал. Значення зображені на рисунках 3.19 і 3.20 означають що

мікроконтролер успішно пройшов етапи верифікації і завантаження коду і готовий до початку роботи.

```
Sketch uses 625533 bytes (47%) of program storage space. Maximum is 1310720 bytes.  
Global variables use 31884 bytes (9%) of dynamic memory, leaving 295796 bytes for local variables.
```

Рисунок 3.19 – Відображення успішного проходження верифікації

```
Wrote 632048 bytes (381342 compressed) at 0x00010000 in 5.9 seconds (effective 850.5 kbit/s)...  
Hash of data verified.  
  
Leaving...  
Hard resetting via RTS pin...
```

Рисунок 3.20 – Відображення успішного завантаження коду в мікроконтролер

Після цього наш контролер переходить до стану активного пошуку користувача для парування через технологію Bluetooth. Перейдемо до смартфона і зайдемо в меню Bluetooth в налаштуваннях. В цьому меню переходимо до опції пошуку доступних для парування пристроїв і знаходимо там наш контролер (рис. 3.21). Натискаємо на пристрій з назвою ESP Controller і прив'язуємо пару. Після цього контролер починає працювати і виконувати усі описані вище функції.

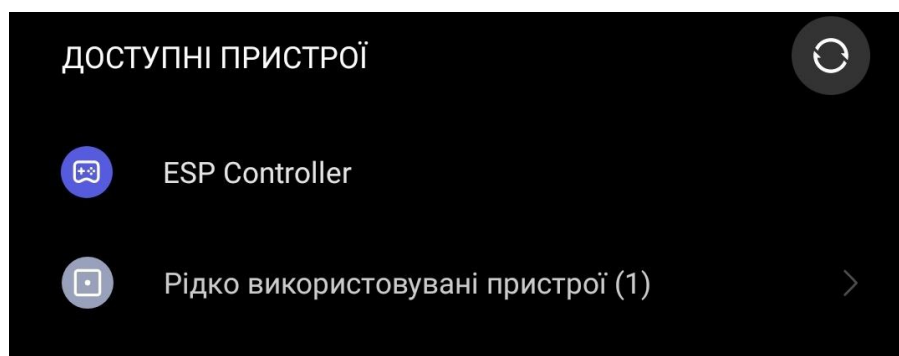


Рисунок 3.21 – Підключення контролеру до смартфона

Для тестування функцій було встановлене програмне забезпечення з назвою GamePad Tester, яке підтримує тестування контролерів які підключаються до Android пристроїв.

Ця програма є мостом між контролером і мікроконтролером, контролер не підключений безпосередньо до мікроконтролера (рис. 3.22). Для прямого підключення вам знадобиться використовувати мікроконтролер із підтримкою Bluetooth і знати, як писати власні стеки Bluetooth.



Рисунок 3.22 – Частина діаграми Activities класів ПЗ для тестування контролера

Ігровий контролер підключається до вашого смартфона через Bluetooth або USB. Android IDE (або Android Studio) має вбудовані функції для обробки ключових подій, таких як натискання кнопки або рух джойстика, та може розпізнавати натискання клавіатури та клацання мишею. Gamepad Tester приймає ці дані контролера та надсилає їх на ESP32 за мережевим SSID: «ESPTest». Це означає, що будемо використано метод Char-by-Char або Streaming для оцінки кожного байта в порядку його надходження.

ESP32 діє як сервер, тоді як смартфон є клієнтом (рис діаграма 3.23). Не потрібно підключатися до цього сервера вручну. Застосунок починає шукати пристрої WiFi з назвою "ESPTest". Після того, як смартфон під'єднається до ESP32, він увімкне режим потокової передачі. Цей режим дозволяє швидко надсилати UDP-пакети на ESP32. Звичайно, проблем, які виникли, багато. Оскільки UDP дуже швидкий, це може спричинити аварійне завершення роботи ESP32. Єдиним недоліком є те, що деякі пакети можуть бути втрачені, але оскільки це ігрові дані, то це не критично, тому що натискання кнопки людиною займає лише близько 10–20 мс, що навряд чи помітно під час трансляції.

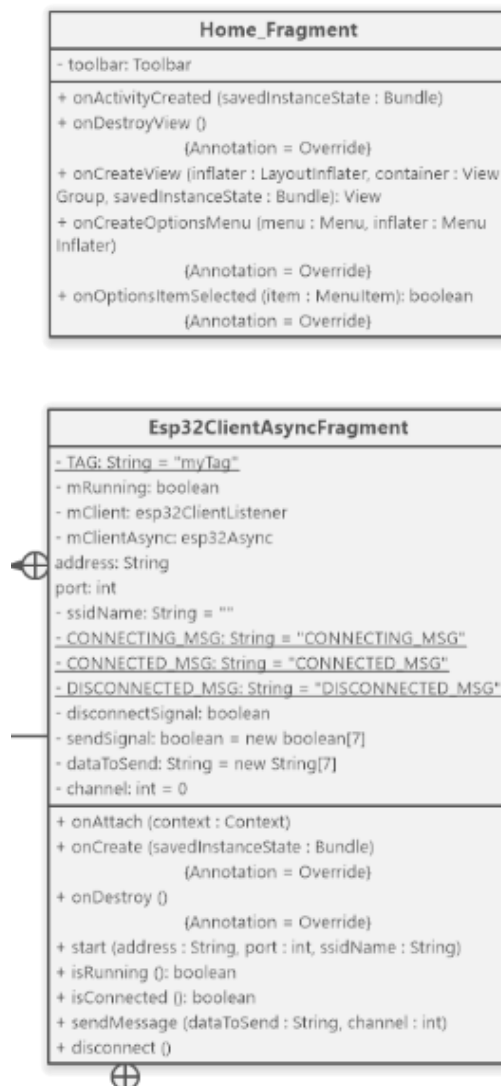


Рисунок 3.23 – Частина діаграми Fragments класів ПЗ для тестування контролера

Як видно з інтерфейсу застосунку (рис. 3.22), контролер посилає відповідні сигнали при натисканні відповідних кнопок, а смартфон ці сигнали опрацьовує і перетворює в дії. В застосунку видно що натиснута кнопка А, що свідчить про спрацьовування тактильної кнопки, і видно відхилення лівого стіка у напрямку вліво і вгору, що свідчить про відповідне зміщення осі механічного компоненту в тому самому напрямку.

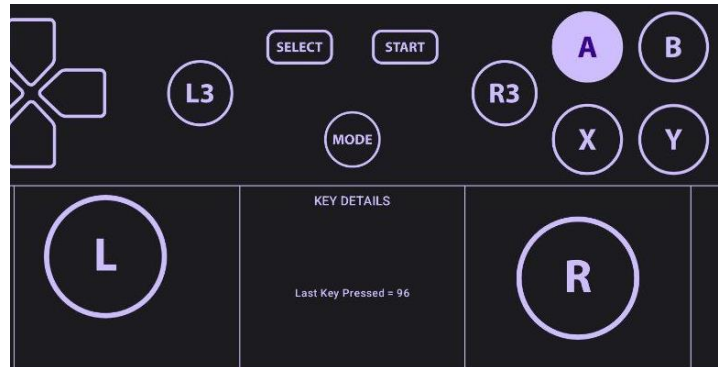


Рисунок 3.22 – Інтерфейс програми тестування

Виконавши відповідні тестування можна з впевненістю сказати що програмування контролера відбулося вдало, і розробка програмного забезпечення не потребує додаткових коригувань.

3.4 Методи покращення пристрою

Для покращення пристрою можна використовувати як програмні так і апаратні компоненти. На програмному рівні важливим стане розробка власного мобільного застосунку не тільки для тестування роботи контролера, але і для можливості прив'язувати певні кнопки до певних положень на екрані смартфона, тим самим додаючи більше функціоналу для контролера.

Важливим рішенням з апаратної частини є додавання більшої кількості різних кнопок, для виконання описаного функціоналу в пункті про програмне забезпечення. Гарним рішенням стане зміна використаного в проєкті стіку на більш дорогий і функціональну одиницю. Новий стік має бути більш чутливим до змін в нахилі осей і передавати найменші зміни в нахилі, що відкриє доступ до точної імітації водіння пальцем по екрану смартфона.

Непоганим рішенням може стати впровадження системи зворотнього зв'язку між користувачем і девайсом, шляхом додавання вібромоторчика, котрий буде реагувати на певні виконані дії на контролері і отримання певних дій в процесі гри на смартфоні.

Висновки до третього розділу

У цьому розділі було детально оглянуто можливі середовища програмування для мікроконтролерів сімейства ESP і конкретно для самого ESP32. Проведено аналіз розглянутих середовищ і обрано Arduino IDE. З використанням цього середовища не виникло проблем, а навпаки, простий і зручний інтерфейс дозволив з легкістю виконати усі кроки до підготовки ESP32 для роботи з цим середовищем.

Створили програмне забезпечення для апаратної частини контролера, з детальним описом коду що використовувався для програмування. Розглянуто процеси ініціалізації самого мікроконтролера і його Bluetooth модуля, обробки відповідних функціональних кнопок та зчитування даних з стіка.

Проаналізовано різні методи покращення готового пристрою шляхом додавання різного апаратного обладнання до уже готової конструкції, так і заміна деталей котрі використовувалися у рішені. Шляхом програмного обладнання було визначено що найкращим рішенням буде впровадження свого власного застосунку для тестування і прив'язування функціональних компонентів.

ВИСНОВКИ

Результатом кваліфікаційної бакалаврської роботи є пристрій, який являє собою бездротовий контролер для аркадних відеоігор. Розроблений пристрій поєднує в собі найкращі рішення уже готових контролерів, котрий покращує ефективність і комфорт при використанні, має ергономічну форму та виконує свої функції в повному обсязі.

В ході роботи було проаналізовано різні компоненти котрі можна впровадити в систему і обрано найефективніші компоненти, включаючи вибір мікроконтролеру, різних функціональних елементів керування, таких як кнопки і стіки. Розроблено забезпечення системи живлення для контролера при його використанні. Під час розробки апаратного забезпечення було детально описано кожен крок при підключенні різних компонентів до мікроконтролера і в загальну суцільну систему, включаючи мікроконтролер ESP32, тактильні кнопки і стік для емуляції натискання на екран і свайпання по екрану відповідно. Було налаштовано бездротове з'єднання між контролером і пристроєм на базі ОС Android, описані різні шляхи такого підключення і вибрано єдиний стандарт для використання в роботі.

Було розроблено програмне забезпечення для контролера на базі вже існуючих бібліотек і методів, яке використовує 623973 байт (47%) оперативної пам'яті. Максимум – 1310720 байт. Глобальні змінні використовують 31884 байти (9%) динамічної пам'яті, залишаючи 295796 байт для локальних змінних. Максимальний розмір – 327680 байт. Повністю описано код програмного забезпечення для всіх елементів апаратної частини, від ініціалізації компонентів до опрацювання певних аналогових сигналів і їх передачі на інший пристрій.

Розглянуто можливі зміни в використаних при проектуванні компонентах, так і додавання нових. З програмної сторони було розглянуто створення програмного забезпечення для ОС Android для детального налаштування контролера.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is a controller in computing? URL: <https://www.techtargеt.com/whatis/definition/controller> (Last accessed: 09.05.2024).
2. What is a Gamepad? – Definition from Techopedia. URL: <https://www.techopedia.com/definition/7363/gamepad> (Last accessed: 09.05.2024).
3. What is a Gamepad? URL: <https://www.computerhope.com/jargon/g/gamepad.htm> (Last accessed: 09.05.2024).
4. What is a Computer Keyboard? – Parts, Layout & Functions. URL: <https://study.com/academy/lesson/what-is-a-computer-keyboard-parts-layout-functions.html> (Last accessed: 09.05.2024)..
5. What is mouse? | Definition from TechTarget. URL: <https://www.techtargеt.com/whatis/definition/mouse> (Last accessed: 09.05.2024).
6. Детально про стандарти та специфікації USB. URL: <https://www.imena.ua/blog/usb-standards-and-specifications/> (Дата звернення: 09.05.2024).
7. Interpreting Analog Sticks URL: <https://blog.hypersect.com/interpreting-analog-sticks/> (Last accessed: 10.05.2024).
8. What is a Steering Wheel? URL: <https://www.computerhope.com/jargon/s/steewhee.htm> (Last accessed: 10.05.2024).
9. What is a Joystick? How to Use it in Gaming. URL: <https://www.lenovo.com/us/en/glossary/joystick/?orgRef=https%253A%252F%252Fwww.google.com%252F> (Last accessed: 10.05.2024).
10. What is a Touchpad. URL: <https://www.javatpoint.com/what-is-a-touchpad> (Last accessed: 10.05.2024).

11. A Beginner's Guide to Understanding Basic Circuit Board Components. URL: <https://vectorbluehub.com/circuit-board-components> (Last accessed: 10.05.2024).
12. DIY Game Controller – Instructables. URL: <https://www.instructables.com/DIY-Game-Controller/> (Last accessed: 10.05.2024).
13. Мікро вібромоторчик 3В (11 мм). URL: <https://arduino.ua/prod469-mikro-vibromotorchik-3v-11-mm> (Дата звернення: 10.05.2024).
14. Bluetooth technology overview. URL: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/> (Last accessed: 07.06.2024).
15. ESP32 Pinout Reference. URL: <https://lastminuteengineers.com/esp32-pinout-reference/> (Last accessed: 07.06.2024).
16. Wokwi – Online ESP32, STM32, Arduino Simulator. URL: <https://wokwi.com> (Last accessed: 07.06.2024).
17. Software: Arduino IDE. URL: <https://www.arduino.cc/en/software> (Last accessed: 07.06.2024).
18. Arduino ESP32. URL: https://docs.espressif.com/projects/arduino-esp32/en/latest/getting_started.html (Last accessed: 07.06.2024).
19. ESP32-BLE-Gamepad. URL: <https://github.com/lemmingDev/ESP32-BLE-Gamepad> (Last accessed: 07.06.2024).
20. Yunsun A. Hong, Soojin O. Peck, Ilgang M. Lee A Comparison of Input Devices for Gaming: Are Gamepads Still Useful in PC Environment? In book: Intelligent Human Computer Interaction (pp.403-416), Seoul, Republic of Korea, January 2022. URL: https://www.researchgate.net/publication/359338677_A_Comparison_of_Input_Devices_for_Gaming_Are_Gamepads_Still_Useful_in_PC_Environment (Last accessed: 13.06.2024).

21. Mehaboob Mujawar, Subuh Pramono Wireless Networking Fundamentals: An In-Depth Exploration Of Wireless Technologies, India, April 2024. URL: https://www.researchgate.net/publication/380318158_Wireless_Networking_Fundamentals_An_In-Depth_Exploration_Of_Wireless_Technologies (Last accessed: 13.06.2024).
22. Ignas Plauska, Agnius Liutkevičius, Audronė Janavičiūtė Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller, Kaunas, Lithuania, December 2022. URL: https://www.researchgate.net/publication/366663196_Performance_Evaluation_of_CC_MicroPython_Rust_and_TinyGo_Programming_Languages_on_ESP32_Microcontroller (Last accessed: 13.06.2024).

ДОДАТОК А

Довідка

про унікальність пояснювальної записки

бакалаврської кваліфікаційної роботи на тему:
«Бездротовий контролер аркадних відеоігор»

студента спеціальності 123 «Комп'ютерна інженерія» групи 405
Павленко Богдан Вікторович
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту бакалаврської кваліфікаційної роботи: схожість
складає 3,99 %

UNICHECK
by Turnitin

User name: **Іван Бурлаченко** Check ID: **1016366408**
Check date: **17.06.2024 00:09:18 EEST** Check type: **Doc vs Internet + Library**
Report date: **17.06.2024 00:13:45 EEST** User ID: **100000130**

File name: **Павленко_Б.В. Комп'ютерна інженерія КБР 2024**
Page count: **47** Word count: **10390** Character count: **78732** File size: **91.55 KB** File ID: **1016172656**

3.99% Matches
Highest match: **1.28%** with Internet source (<https://uk.wikipedia.org/wiki?curid=1625990>)

3.02% Internet sources 50 Page 49
1.15% Library sources 5 Page 49

0% Quotes
Exclusion of quotes is off
Exclusion of references is off

0% Exclusions
No exclusions

Modifind
Text modifications detected. Find more details in the online report.
Replaced characters 5

Здобувач:

студент 405 групи

підпис

Павленко Б. В.
ініціали, прізвище

Керівник:

ст. викладач

підпис

Бурлаченко І. С.
ініціали, прізвище

Дата: «___» _____ 2024 р.

ДОДАТОК Б

Повний код розробленого програмного забезпечення

```
#include <Arduino.h>
#include <BleGamepad.h>
BleGamepad bleGamepad("ESP Controller", "BohdanPavlenko");
#define A 33
#define LX 34
#define LY 35
#define B 32
void setup() {
  Serial.begin(115200);
  Serial.println("Starting BLE work!");
  bleGamepad.begin();
}
void loop() {
  if (bleGamepad.isConnected()) {
    if (digitalRead(A) == LOW) {
      Serial.println(F("A Button"));
      bleGamepad.press(BUTTON_1);
      bleGamepad.sendReport();
      delay(10);
      bleGamepad.release(BUTTON_1);
      bleGamepad.sendReport();
    }
    if (digitalRead(B) == LOW) {
      Serial.println(F("B Button"));
      bleGamepad.press(BUTTON_2);
      bleGamepad.sendReport();
      delay(10);
    }
  }
}
```

```
    bleGamepad.release(BUTTON_2);
    bleGamepad.sendReport();
}
// Map analog reading from 0 ~ 4095 to 32737 ~ -32737 for use as an axis
reading
int value1 = map(analogRead(LX), 0, 4095, 32767, -32767);
int value2 = map(analogRead(LY), 0, 4095, 32767, -32767);
bleGamepad.setLeftThumb(value1, value2);
bleGamepad.sendReport();
delay(1);
}
}
```

ДОДАТОК В

Графічні матеріали КБР

